

Herausgeber

T. LÄNGLE
F. PUENTE LEÓN
M. HEIZMANN

FORUM
BILDVERARBEITUNG 2018



Scientific
Publishing

T. Längle | F. Puente León | M. Heizmann (Hrsg.)

FORUM BILDVERARBEITUNG 2018

FORUM BILDVERARBEITUNG 2018

Herausgegeben von

T. Längle, F. Puente León und M. Heizmann

Impressum



Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark
of Karlsruhe Institute of Technology.
Reprint using the book cover is not allowed.

www.ksp.kit.edu



*This document – excluding the cover, pictures and graphs – is licensed
under a Creative Commons Attribution-Share Alike 4.0 International License
(CC BY-SA 4.0): <https://creativecommons.org/licenses/by-sa/4.0/deed.en>*



*The cover page is licensed under a Creative Commons
Attribution-No Derivatives 4.0 International License (CC BY-ND 4.0):
<https://creativecommons.org/licenses/by-nd/4.0/deed.en>*

Print on Demand 2018 – Gedruckt auf FSC-zertifiziertem Papier

ISBN 978-3-7315-0833-5

DOI 10.5445/KSP/1000085290

Vorwort

Bildverarbeitung spielt in vielen Bereichen der Technik zur schnellen und berührungslosen Datenerfassung eine Schlüsselrolle. Beispielsweise in der Qualitätssicherung industrieller Produktionsprozesse, in der Robotik und zur Fahrerassistenz haben sich Bildverarbeitungssysteme einen unverzichtbaren Platz erobert. Diese Entwicklung wird unterstützt durch die Verfügbarkeit qualitativ hochwertiger und günstiger Sensorsysteme sowie durch die Zunahme der Leistungsfähigkeit von Rechnersystemen.

Das „Forum Bildverarbeitung“ hat das Ziel, über aktuelle Trends zu allgemeinen und robusten Lösungen in der Bildverarbeitung zu berichten und zum fachlichen Austausch zwischen den Teilnehmern beizutragen. Die Beiträge wurden vom Programmausschuss ausgewählt. Sie umfassen die folgenden Schwerpunkte:

- Bildgewinnung,
- Computational Imaging,
- Bildsignalverarbeitung,
- Maschinelle Modelle und Verfahren,
- Mustererkennung sowie
- Deep Learning.

Das Forum Bildverarbeitung möchte einen Beitrag zur Weiterentwicklung der Bildverarbeitung als wichtige und zukunftssträchtige Fachdisziplin leisten. Es richtet sich an Fachleute, die sich in der industriellen Entwicklung, in der Forschung oder der Lehre mit Bildverarbeitungssystemen befassen, und bietet eine Plattform für den Wissens- und Erfahrungsaustausch zwischen Wissenschaftlern und Anwendern.

November 2018

T. Längle, F. Puente León und M. Heizmann

Wissenschaftliche Leitung

Prof. Dr.-Ing. T. Längle	Fraunhofer-IOSB Karlsruhe
Prof. Dr.-Ing. F. Puente León	Karlsruher Institut für Technologie
Prof. Dr.-Ing. M. Heizmann	Karlsruher Institut für Technologie

Programmausschuss

Prof. Dr. C. Bach	Buchs
Dr.-Ing. S. Bauer	Madison (Wisconsin)
Prof. Dr.-Ing. J. Beyerer	Karlsruhe
Dr. rer. nat. J. Burke	Karlsruhe
Prof. Dr. K. Donner	Passau
Dr. rer. nat. J. Eggert	Offenbach
Dr. T. Haist	Stuttgart
Prof. Dr. A. Heinrich	Aalen
Prof. Dr. B. Jähne	Heidelberg
Dipl.-Ing. M. Maurer	Wiesbaden
Prof. Dr. R. Neubecker	Darmstadt
Prof. Dr. W. Osten	Stuttgart
Prof. Dr. F. Salazar	Madrid
Dipl.-Ing. M. Stelzl	Mainz
Prof. Dr.-Ing. C. Stiller	Karlsruhe
Prof. Dr.-Ing. R. Tutsch	Braunschweig
Prof. Dr.-Ing. S. Werling	Mannheim
Dipl.-Ing. S. Wienand	Darmstadt
Dr.-Ing. V. Willert	Darmstadt

Inhaltsverzeichnis

Vorwort	v
---------------	---

Optische Messtechnik

Coding strategies for static patterns suitable for UV deflectometry <i>C. Kludt and J. Burke</i>	1
Photogrammetrische Bestimmung der Brechungseigenschaften von Flüssigkristallbildschirmen..... <i>H. Dierke, M. Petz und R. Tutsch</i>	13
Single pose camera calibration using a curved display screen <i>R. Acuna, R. Ziegler, and V. Willert</i>	25
100% Oberflächeninspektion in der Karosseriefertigung <i>F. Loibl</i>	37
Kalibrierfreie 3D-Rekonstruktion von spiegelnden Oberflächen mittels Orthogonalprojektionen unter Verwendung von Reflexionsmarkern <i>C. Negara und T. Längle</i>	47
Subtleties of extrinsic calibration of cameras with non- overlapping fields of view <i>Z. Li and V. Willert</i>	59

Bildgewinnung und Simulation

Extended noise equalisation for image compression considering linear and quadratically increasing noise sources..... <i>D. Kirchhöfer and B. Jähne</i>	71
--	----

A concept for a multi-spectral snap-shot camera based on single-chip RGB sensor	81
<i>R. Zirkler, T. Eckhard, and B. Jödicke</i>	
Visuelle Szenendiskretisierung mittels wahlfreien Bildrastern am Beispiel von Drei-, Vier- und Sechseckpixeln	95
<i>R. Manthey und D. Kowerko</i>	
Image processing for scatter-plate microscopy	105
<i>S. Ludwig, B. Le Teurnier, G. Pedrini, and W. Osten</i>	
Übersicht zur Übertragbarkeit der Leistungscharakteristik von Computer Vision Algorithmen bei synthetischen und realen Luftbildaufnahmen	117
<i>M. Krump, C. Hellert, G. Hummel und P. Stütz</i>	
Virtuelle Inbetriebnahme kamerabasierter Qualitätssicherungssysteme	129
<i>N. Piero</i>	

Mustererkennung

Defekterkennung an optischen Funktionsflächen	135
<i>M. Petz, M. Fischer und R. Tutsch</i>	
Komplexitätsbestimmung von Datensätzen als Grundlage für die Wahl eines geeigneten Klassifikationsverfahrens	147
<i>K. Anding, L. Haar, K. Trambitckii, D. Garten und G. Notni</i>	
Automated classification of imperfections and dust on small optical elements	161
<i>A. Schöch, P. Perez, S. Linz-Dittrich, C. Bach, and C. Ziolek</i>	
Stochastische Mustererkennung zur Bildsegmentierung	173
<i>T. Müller und B. Erdnüß</i>	

Maschinelles Lernen

Ein evolutionärer Ansatz für die automatische Ermittlung der Topologie neuronaler Netze	185
<i>N. Mitschke, M. Heizmann, K.-H. Noffz und R. Wittmann</i>	

Towards explaining deep learning networks to distinguish facial expressions of pain and emotions	197
<i>K. Weitz, T. Hassan, U. Schmid, and J. Garbas</i>	

Application of deep learning for crack segmentation on concrete surface	209
<i>O. Müller, A. Moghiseh, H. Stephani, N. Rottmayer, and F. Huang</i>	

Algorithmik

Der phenoTest – Ein automatisiertes CT-System zur Phänotypisierung von Zuckerrübenkeimlingen	217
<i>M. Rehak, U. Haßler, T. Grulich, N. Wörlein, F. Porsch, Y. Götz und A. Wolff</i>	

Algorithms for microlens center detection	229
<i>M. Schambach and F. Puente León</i>	

Schnelle und präzise Segmentierung von beweglichen und morphologisch variablen Objekten am Beispiel der Deflagrationsdetektion	241
<i>J. Krooß, F. Kümmerlen und A. Fay</i>	

The HCI flow metrics: A novel approach for benchmarking optical flow	253
<i>D. A. Brock and B. Jähne</i>	

Local histogram equalization and homomorphic filtering to improve wide-angle stereo disparity computation	265
<i>S. Metzner, R. Ramesh, and T. Hausotte</i>	

Umfeldwahrnehmung

Embedded Stereokamera-basierte reaktive Kollisionsvermeidung für UAVs	277
<i>M. Grinberg, B. Ruf, S. Monka, M. Kollmann, A. Buller und I. Tchouchenkov</i>	

Reliable multilane detection and classification using a compact encoder-decoder CNN	291
<i>S. V. Chougule, A. Ismail, G. Adam, V. Narayan, and M. Schulze</i>	

Adaptive Algorithmenauswahl für die Fahrzeugdetektion auf Luftbildaufnahmen	303
<i>C. Hellert und P. Stütz</i>	

Mensch-Maschine-System

A kalman filter with state constraints for model-based dynamic facial action unit estimation	315
<i>T. Hassan, D. Seuß, A. Ernst, and J. Garbas</i>	

Monokulare Kopfposenschätzung	327
<i>B. Jäschke und F. Puente León</i>	

Sensorik

Real-time stereo vision on FPGAs with SceneScan	339
<i>K. Schauwecker</i>	
Camera projector system for cost-effective 3D surface detection ...	351
<i>R. Shanthakumar, B. Yadaiah, and M. Kaufmann</i>	

Coding strategies for static patterns suitable for UV deflectometry

Codierungsstrategien für in der UV-Deflektometrie anwendbare Muster

Christian Kludt¹ and Jan Burke²

¹ Karlsruhe Institute of Technology,
Karlsruhe School of Optics and Photonics,
Kaiserstraße 12, 76131 Karlsruhe

² Fraunhofer IOSB,
Fraunhoferstraße 1, 76131 Karlsruhe

Abstract We introduce a method based on the deflectometry principle for the inspection of transparent, rotation-symmetric objects with steep angles. Examples are intraocular or strongly curved, aspheric lenses. Usually, an additional reflection at the lens' back side occurs and disturbs the signal. The solution is to use ultraviolet light which does not penetrate the material. As a consequence, static masks have to be used to generate the fringe pattern. A key feature of our approach is the specification of a spiral pattern, that meets the requirements for ultraviolet deflectometry with static masks. The spiral patterns are decoded by a multi-frequency phase shifting algorithm. We explain the principles of pattern design and present first evaluation results. Furthermore, we introduce a straight forward approach to obtain the absolute coordinates of the screen directly without any unwrapping. This facilitates the reconstruction of the three-dimensional shape of the lens in a subsequent step.

Keywords Optical Inspection, Lens Testing, Deflectometry, Fringe Analysis, Phase Measurement.

1 Introduction

In some medium-uncertainty applications, interferometry can today be replaced by phase measuring deflectometry (PMD) as a tool for

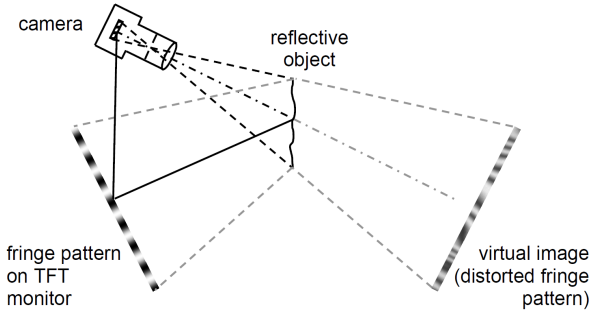


Figure 1: Deflectometry setup. A regular fringe pattern is displayed on a monitor, gets reflected and distorted by the object and is captured by the camera. From [3].

lens inspection [1], [2]. Deflectometric sensors provide often sufficient accuracy and benefit from their cost-effective components that consist only of a monitor and a camera. The monitor displays a regular reference pattern that is reflected by the object's surface and captured by the camera. If the object's surface shows topographic irregularities such as dents and bumps, the recorded pattern appears distorted. By analyzing the deformation and displacement compared to the reference pattern, and having a calibrated system, i.e. knowing the position and orientation of both monitor and camera, it is also possible to reconstruct the global surface shape. The basic setup is shown in Figure 1.

However, when inspecting transparent objects, one will observe an additional reflection from the object's back side. It is superimposed onto the front side reflection, generating an unwanted extra signal that cannot be processed by conventional evaluation procedures [4] and at some locations cannot be analyzed at all [5].

2 Existing approaches

A suitable solution is to apply light at a spectral range where the object's material is non-transparent. Most organic lenses are made from polymethyl methacrylate (PMMA), which is blended to be absorbing

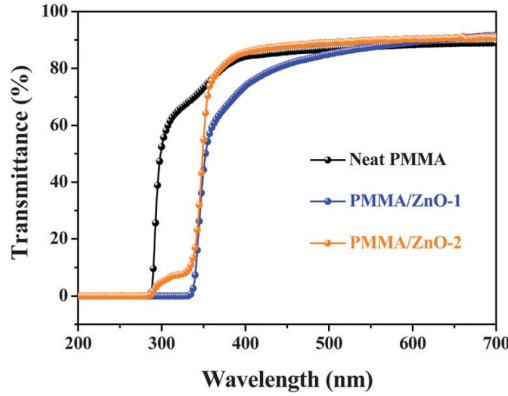


Figure 2: Transmission spectra of pure and blended PMMA. From [6].

for ultraviolet (UV) radiation [6]. Figure 2 shows the transmission spectra of pure and blended PMMA. Consequently, in the UV the reflection takes place only at its front side.

Problem 1: There is no spatial pattern generator available working in the UV range, as there is in the visual range (e.g. TFT monitors or OLED panels). So one key issue is how to generate the reference fringe pattern at the appropriate wavelength.

Problem 2: Fringe compression occurs at objects with steep edges or strongly curved surfaces. This is due to perspective shortening effects from the camera's line of sight and widening of its sight ray when reflected at convex surfaces. In Figure 3(b) this effect occurs at the periphery of a lens. In the area marked red, the fringes of the reflected pattern become very dense and eventually cannot be resolved anymore.

Solution to 1: Sprenger et al. [9] illuminate a slit mask with a UV light source to create a UV line lighting. However, the downside of this approach is its only one-dimensional extent. To acquire the second

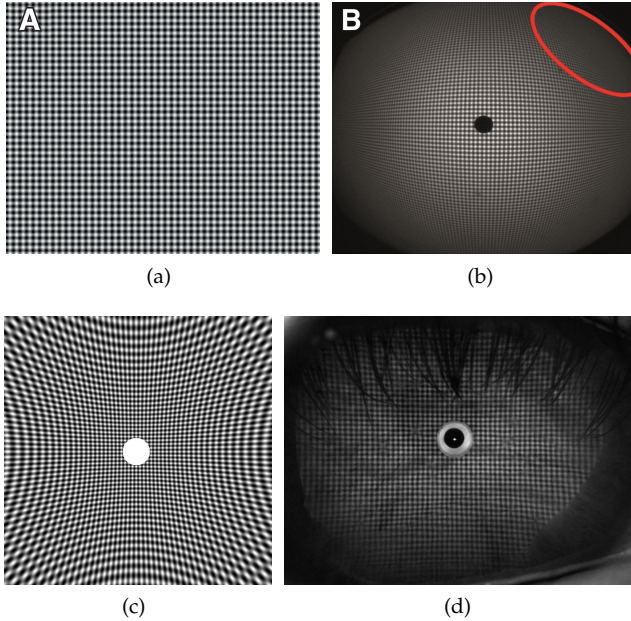


Figure 3: (a) Predistorted Cartesian fringe pattern. (b) Straight fringe pattern reflected from cornea. (c) Reference pattern with straight fringes. (d) Reflected reference pattern shows fringe compression at the edging of a lens. From [7], [8].

dimension, a time consuming scanning process is required to compile an image of the whole object covered by fringes. A straight forward approach is to create a metallic pattern and illuminate it with ultraviolet light or to place a two-dimensional mask transmissive in the UV in front of a UV light source. In any case, it will be a static pattern and hence has to be moved mechanically if phase shifting techniques are applied.

Solution to 2: Fringe compression by convex lenses can be compensated by using predistorted reference patterns that are matched to their rotational symmetry. Figure 3(c) shows a predistorted Cartesian

fringe pattern that compensates the distortion caused by the cornea of a human eye. As a result, the reflected pattern shows straight fringes and uniform fringe spacing, cf. Figure 3(d). This ensures constant fringe resolution throughout the image plane.

But temporal phase shifting (TPS) [10] is difficult with current predistorted, static patterns: As predistortion introduces a non-uniform scale, each point will be shifted by a different fraction of periods when the pattern is moved. So, if each point of the pattern is to be shifted by the same fraction of periods, they each have to be moved by a different distance or corrections for non-uniform phase shifts must be used. Hence, the pattern does not coincide with itself after shifting, and instead behaves like a dynamic pattern.

While solutions already exist for the individual problems, there still is no solution to the combined problem: There is no spatial light modulator available in the UV range that provides predistortion for convex, rotation symmetric objects (such as lenses) and stays congruent when it is shifted.

3 Proposed method

In order to obtain two phase maps with an orthonormal basis, modulation must be applied to a fringe pattern in two linearly independent directions. The basic approach is to shift a sinusoidal fringe pattern laterally in one direction, then rotate it by 90 degrees and shift it again. This requires N phase shifts per direction, so a total number of $2N$ phase shifts is required.

A more efficient method is demonstrated by Liu et al. [11]: A one-dimensional phase shift is applied to a crossed fringe pattern. The phase shifting direction must not be parallel to any of the fringes so as to create simultaneous phase modulation of both fringe orientations. At this, each point is shifted by a different integer number of periods for the x and y direction, respectively. In Figure 4, a red marker is drawn at a fixed phase to visualize the pattern translation. To decode the phase information, the phase shifting algorithm (PSA) is adjusted

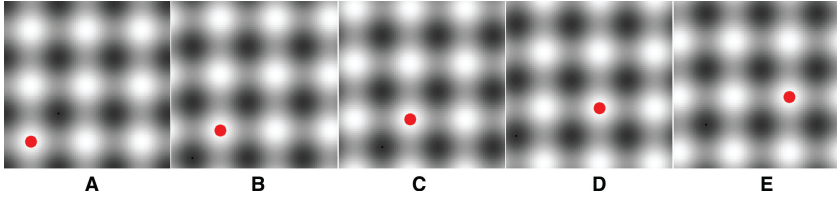


Figure 4: One-dimensional phase shift by two periods horizontally and one period vertically. From [11].

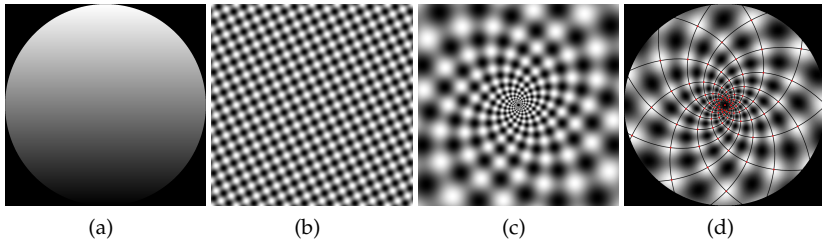


Figure 5: (a) Linear gray scale pattern. (b) Cartesian coordinate system with regular, straight fringes. (c) Log-polar coordinate system with spiral fringes. (d) Intersections of spiral arms, indicated as red dots. They provide an orthonormal basis in each point.

to the number of shifted periods, i.e. it is made frequency-selective. This corresponds to applying the discrete Fourier transform (DFT) to the phase shifting sequence and reduces the number of exposures to N .

An alternative approach is encoding the screen not by a cosine pattern but by a linear gray scale [12], cf. Figure 5(a). It is rotated around its center and due to its width predistortion is not required. For phase decoding, only $N = 3$ phase shifts are needed.

Ziebarth et al. [13] introduced a novel pattern design based on spinning spirals. It exploits the rotational symmetry of lenses and enables TPS techniques to be applied to predistorted, static patterns. Instead of Cartesian coordinates as in Figure 5(b), fringe patterns are generated in Log-polar coordinates, see Figure 5(c). The final crossed fringe pattern

depicted in Figure 5(d) is the superposition of two counter rotating spiral patterns. Phase shifting is not applied in lateral but in angular direction, i.e. the spirals are spinning around their center, therefore the name.

Crossed fringe patterns with logarithmic spirals offer the following characteristics:

- During rotation, the modulation in both spiral directions is achieved, which enables phase decoding in two linearly independent directions.
- They are rotationally symmetric and therefore coincide with themselves after being rotated by certain angles, i.e. the rotation is a congruence mapping. This allows for the use of static, mechanically moved patterns.
- A special property of logarithmic spirals is their constant slope and therefore intersection angle with the tangential and radial line. Hence, the spiral crossed fringe pattern has an orthonormal basis at each point, see the crossing points in Figure 5(d). This simplifies the calculation of the surface slope in a subsequent step.
- A substantial feature is their inherent predistortion: The width of the spiral arms increases with the radius, thereby compensating the fringe compression by strong positive lenses. Of course the opposite effect occurs when moving along the spirals to their origin. In the center, the fringes become too dense to be resolved by the camera. In practice, however, the camera will be placed here anyway, as in Figure 3(d).

4 Coding scheme

In conventional TPS in the visual range, each pattern is displayed and shifted separately. Since in the UV static masks have to be used, the individual patterns have to be superimposed to create a composite, crossed fringe pattern. In order to extract the contained patterns, we propose a *multi-frequency phase coding* technique.

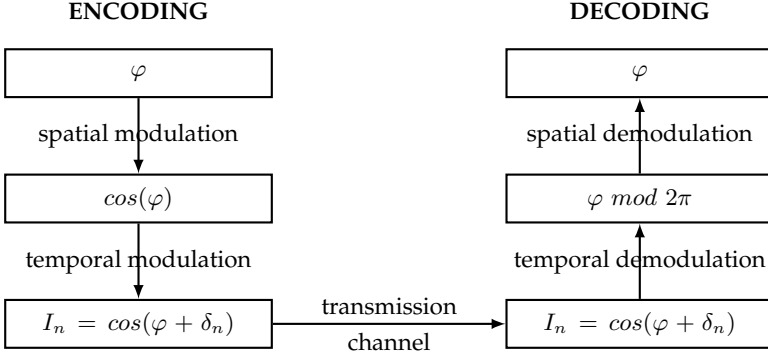


Figure 6: Signal model of a deflectometric coding scheme.

To understand the individual steps, its signal model is depicted in Figure 6. The screen coordinates are represented by the phases φ_x and φ_y and in polar coordinates by φ_r and φ_ϕ , respectively. For the sake of brevity we omit the location dependency and describe the coding scheme by the generic phase coordinate φ . To exploit the precision of phase shifting techniques, we introduce spatial modulation in the form of a cosine pattern. Now we apply TPS as we shift the pattern by the nominal phase increment δ_n and receive the phase shifting sequence I_n , which is displayed on the monitor. The reflection from the object is modeled as the transmission channel of the system and is affected by additional noise [14], but in the following we assume an ideal channel. The camera's irradiance readings are temporally demodulated by PSAs into the wrapped phase distribution $\varphi \bmod 2\pi$. Spatial demodulation is achieved by unwrapping algorithms. Finally, the absolute phase φ is obtained, which identifies the screen coordinate, where the light impinging on each camera pixel originated from.

The crucial task is the stable algorithmic separation of the two superimposed fringe patterns from the composite pattern displayed in Figure 5(d). Each pattern contains a different number of spiral arms f , hence we can adjust our algorithm to each number and construct a frequency selective PSA. For this purpose, we apply the DFT algorithm to the phase shifting sequence I_n and extract the phases at the defined

frequencies. For this to function best, the pattern has to be shifted by one full revolution and a minimum number of $N = 2f_{max} + 1$ incremental phase shifts is required. The results are the wrapped, but orthonormal phase distributions in Figure 7(a) and 7(b) that are only unambiguous within one fringe period. This is due to the 2π -periodicity of the trigonometric functions within the PSA.

The absolute phase can be spatially unwrapped by analyzing neighbouring pixels [15], [16]. This will work fine on optical, but not flawlessly at discontinuous surfaces. For proper unwrapping, additional information is required. Therefore, a total of four patterns are required, with always two of them forming parallel pairs [12]. If the number of spirals within such a pair are primes, e.g. $f_1 = 5$ and $f_2 = 13$ spirals, the phase distribution can be unwrapped by applying the Chinese remainder theorem [17] and an absolute phase map is obtained.

Unwrapping is not required if we don't introduce spatial modulation at all. Hence, the temporal demodulation of the linear gray scale pattern yields the absolute screen coordinates directly. The DFT algorithm requires only $N = 3$ phase shifts to compute at frequency $f = 1$ the phase in Figure 7(c), which constitutes the angular coordinate φ_ϕ . The radial coordinate φ_r in Figure 7(d) is obtained by computing the fringe modulation at the same frequency. This technique is more susceptible to noise because of its very low fringe frequency.

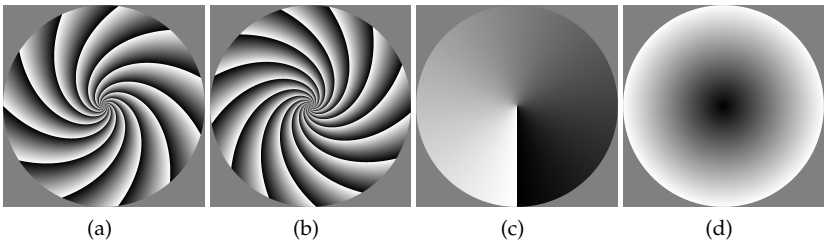


Figure 7: Phase Extraction. Temporal demodulation of the spiral pattern yields wrapped phases (a) and (b), whereas absolute phases (c) and (d) are obtained when the linear gray scale pattern is used.

5 Experiments

So far we assumed an error free transmission channel and simulated the decoding process, cf. Figure 7. To prove it to work in a real environment, we displayed the spiral crossed fringe pattern on a computer monitor and captured it directly with a camera. Although no reflecting object was in between and we were operating in the visual range, it resembles a deflectometric measurement as the setup adds noise to the signal. Figure 8(a) shows the rotated pattern. In Figure 8(b) and 8(c) the extracted, wrapped phases are depicted. The phase decoding works quite well and the technique appears stable under experimental conditions.

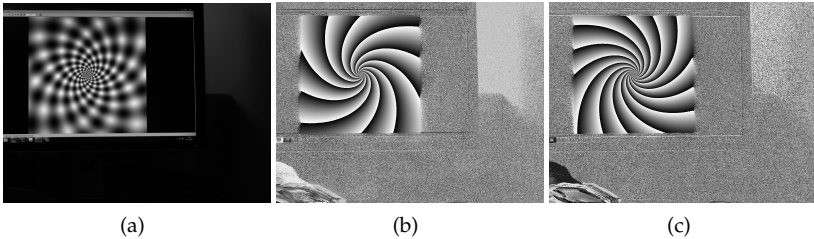


Figure 8: Deflectometric measurement. (a): Rotationally shifted spiral pattern. (b) and (c): Extracted phase maps.

6 Conclusion

We demonstrated the feasibility of spinning spirals, which allows for the use of static, but moving masks to extend the application of deflectometry outside the visual range. Their logarithmic scale compensates for fringe compression on strong convex lenses. The composite crossed fringe pattern is analyzed with the DFT algorithm, which extracts the phase maps at the prescribed frequencies. A simple but efficient approach utilizes a rotating gray scale and yields the absolute phase maps directly. Subject of current research is the reduction of the number N of phase shifts required to decode the phase shifting sequence appropriately on the basis of filters designed with Z-transform techniques [18].

References

1. G. Häusler, "Verfahren und Vorrichtung zur Ermittlung der Form oder der Abbildungseigenschaften von spiegelnden oder transparenten Objekten," Germany Patent DE000 019 944 354B4, April, 1999. [Online]. Available: <https://register.dpma.de/DPMAREgister/pat/PatSchrifteneinsicht?docId=DE19944354A1>
2. 3D-Shape GmbH. (2018) Optische Messtechnik für spiegelnde Oberflächen: SpecGAGE3D. <http://www.3d-shape.com/produkte/spiegelnde-oberflaechen.php>. 3D-Shape GmbH. [Online]. Available: http://www.3d-shape.com/produkte/spiegelnde_oberflaechen.php
3. P. Huke, J. Burke, and R. B. Bergmann, "A comparative study between deflectometry and shearography for detection of subsurface defects," pp. 9203 – 9203 – 12, 2014. [Online]. Available: <https://doi.org/10.1117/12.2063650>
4. C. Faber, M. C. Knauer, and G. Häusler, "Can deflectometry work in presence of parasitic reflections?" in *Proceedings of the 110th Annual DGaO Conference*, ser. DGaO Proceedings, vol. 110, DGaO, 2009, p. A10. [Online]. Available: <http://www.dgao-proceedings.de/download/110/110.a10.pdf>
5. C. Faber, E. Olesch, R. Krobot, and G. Häusler, "Deflectometry challenges interferometry: the competition gets tougher!" pp. 8493 – 8493 – 15, 2012. [Online]. Available: <https://doi.org/10.1117/12.957465>
6. Y. Zhang, X. Wang, Y. Liu, S. Song, and D. Liu, "Highly transparent bulk PMMA/ZnO nanocomposites with bright visible luminescence and efficient UV-shielding capability," *Journal of Materials Chemistry*, vol. 22, no. 24, p. 11971, 2012. [Online]. Available: <https://doi.org/10.1039/c2jm30672g>
7. H. Liang, E. Olesch, Z. Yang, and G. Häusler, "Single-shot phase-measuring deflectometry for cornea measurement," *Advanced Optical Technologies*, vol. 5, no. 5-6, pp. 1–6, jan 2016. [Online]. Available: <https://doi.org/10.1515%2Faot-2016-0049>
8. J. H. Massig, E. Lingelbach, and B. Lingelbach, "Videokeratoscope for accurate and detailed measurement of the cornea surface," *Appl. Opt.*, vol. 44, no. 12, pp. 2281–2287, 4 2005. [Online]. Available: <http://ao.osa.org/abstract.cfm?URI=ao-44-12-2281>
9. D. Sprenger, C. Faber, M. Seraphim, and G. Häusler, "Uv-deflectometry: No parasitic reflections," in *DGaO Proceedings*, vol. 111, 2010, p. A19.

10. H. Schreiber and J. H. Bruning, *Phase Shifting Interferometry*. Wiley-Blackwell, 2006, ch. 14, pp. 547–666. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470135976.ch14>
11. Y. Liu, E. Olesch, Z. Yang, and G. Häusler, “Fast and accurate deflectometry with crossed fringes,” *Advanced Optical Technologies*, vol. 3, pp. 441–445, Aug. 2014.
12. C. Kludt, “Spinning Spirals - Measuring Optical Surfaces in Polar Coordinates,” Master’s Thesis, Karlsruhe Institute of Technology, 2018.
13. M. Ziebarth, T. Stephan, S. Höfer, and J. Burke, “Creating patterns for phase shifting by rotation in deflectometry,” in *DGaO Proceedings*, vol. 118, 2017, p. B12. [Online]. Available: http://www.dgao-proceedings.de/abstract/abstract_only.php?id=2194
14. Y. Surrel, “Additive noise effect in digital phase detection,” *Appl. Opt.*, vol. 36, no. 1, pp. 271–276, Jan 1997. [Online]. Available: <http://ao.osa.org/abstract.cfm?URI=ao-36-1-271>
15. H. Lei, X. yu Chang, F. Wang, X.-T. Hu, and X.-D. Hu, “A novel algorithm based on histogram processing of reliability for two-dimensional phase unwrapping,” *Optik - International Journal for Light and Electron Optics*, vol. 126, no. 18, pp. 1640 – 1644, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0030402615003228>
16. M. A. Herráez, D. R. Burton, M. J. Lalor, and M. A. Gdeisat, “Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path,” *Appl. Opt.*, vol. 41, no. 35, pp. 7437–7444, Dec 2002. [Online]. Available: <http://ao.osa.org/abstract.cfm?URI=ao-41-35-7437>
17. K. Ireland and M. Rosen, “The Chinese remainder theorem,” *A Classical Introduction to Modern Number Theory*, pp. 34–38, 1990.
18. Y. Surrel, “Design of algorithms for phase measurements by the use of phase stepping,” *Appl. Opt.*, vol. 35, no. 1, pp. 51–60, Jan 1996. [Online]. Available: <http://ao.osa.org/abstract.cfm?URI=ao-35-1-51>

Photogrammetrische Bestimmung der Brechungseigenschaften von Flüssigkristallbildschirmen

Photogrammetric determination of the refractive properties of liquid crystal displays

Hanno Dierke, Marcus Petz und Rainer Tutsch

Technische Universität Braunschweig, Institut für Produktionsmesstechnik
Schleinitzstraße 20, 38106 Braunschweig

Zusammenfassung Bei der phasenmessenden Deflektometrie zur Messung spiegelnder und transparenter Oberflächen wird in der Regel ein mittels eines Flüssigkristallbildschirms dargestelltes kontrastreiches Muster optisch erfasst. Eine der hinsichtlich der erzielbaren Messunsicherheit bedeutendsten Nichtidealität von Flüssigkristallbildschirmen ist die Brechung im Glassubstrat, welche zu einem winkelabhängigen Versatz zwischen dem tatsächlichen Ort und dem aufgrund der Brechung beobachteten Ort eines auf dem Bildschirm angezeigten Merkmals führt. Aufgrund der typischen Dicken und Brechungsindizes der verwendeten Glassubstrate erreicht dieser Versatz selbst bei moderaten Betrachtungswinkeln mehrere zehntel Millimeter. In dem hier vorgestellten Ansatz wird der brechungsbedingte Versatz auf dem Bildschirm angezeigter Merkmale durch gleichzeitige Beobachtung angezeigter und auf den Bildschirm angebrachter Merkmale ermittelt. Das bislang verwendete Modell wurde verbessert, so dass außer einem vor der Auswertung zu bestimmenden Abstand der Merkmale keine weiteren Vorkenntnisse erforderlich sind. Im Rahmen der durchgeführten Untersuchungen konnten erstmals auch Aussagen zur Messunsicherheit der Methode getroffen und verifiziert werden.

Schlagwörter Deflektometrie, Flüssigkristalldisplay, LCD, Brechung.

Abstract In phase-measuring deflectometry for measuring reflecting and transparent surfaces, a high-contrast pattern displayed on a liquid crystal screen is usually optically detected. One of the most important non-idealities of liquid crystal displays with regard to the achievable measurement uncertainty is the refraction in the glass substrate, which leads to an angle-dependent offset between the actual location and the observed location of a feature displayed on the screen due to the refraction. Owing to the typical thicknesses and refractive indices of the glass substrates used, this offset reaches several tenths of a millimeter even at moderate viewing angles. In the approach presented here, the refraction-related offset of features displayed on the screen is determined by simultaneous observation of displayed features and features mounted on the screen. The model used so far was improved, so that apart from a distance of the features to be determined before the evaluation no further previous knowledge is necessary. Within the scope of the investigations carried out, statements on the measurement uncertainty of the method could also be made and verified for the first time.

Keywords Deflectometry, liquid crystal display, LCD, refraction.

1 Einleitung

Die deflektometrische Messung spiegelnder und transparenter Oberflächen, insbesondere in ihrer verbreiteten Form der phasenmessenden Deflektometrie, basiert in der Regel auf der optischen Erfassung eines im Raum verkörperten und aus unterschiedlichen Richtungen beobachtbaren kontrastreichen Musters. In der Mehrzahl der Anwendungsfälle kommen zur Realisierung dieser Referenzstruktur handelsübliche Flüssigkristallbildschirme zum Einsatz (vgl. Abbildung 1).

Da im Rahmen der deflektometrischen Rekonstruktion der Oberfläche des Prüflings die Eigenschaften des Bildschirms als bekannt vorausgesetzt werden, kommt einer exakten Kenntnis der Bildschirmeigenschaften eine hohe Bedeutung im Hinblick auf die erzielbare Messunsicherheit zu. Eine der in diesem Zusammenhang bedeutendsten Nichtidealität von Flüssigkristallbildschirmen stellt die optische Brechung im Glassubstrat dar. Diese führt zu einem winkelabhängigen Versatz zwischen dem tatsächlichen Ort eines auf dem Bildschirm angezeigten

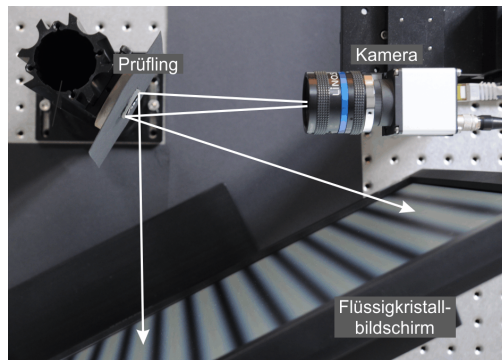


Abbildung 1: Beispielhafter deflektometrischer Aufbau.

Merkmals und jenem Ort, an dem es aufgrund der Brechung zu liegen scheint (Abbildung 2). Aufgrund der typischen Dicken und Brechungsindizes der verwendeten Glassubstrate erreicht dieser Versatz selbst bei moderaten Betrachtungswinkeln mehrere zehntel Millimeter.

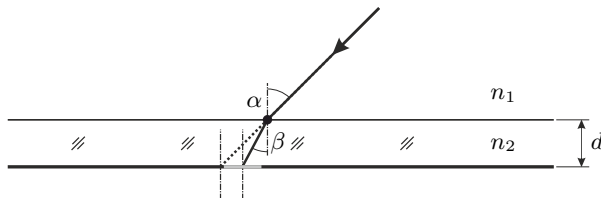


Abbildung 2: Grundlegendes Modell: Brechungsbedingter Versatz der gemessenen Position eines auf dem Display dargestellten und unter dem Winkel α betrachteten Merkmals.

Als grundsätzliche Möglichkeit, den brechungsbedingten scheinbaren Versatz angezeigter Merkmale an einem Flüssigkristallbildschirm zerstörungsfrei zu ermitteln, hat sich die simultane Beobachtung auf dem Bildschirm applizierter und angezeigter Kreismarken unter verschiedenen Beobachtungswinkeln bewährt [1,2] (vgl. Abbildung 4).

Die bislang angewendete Implementierung lässt jedoch Raum für Verbesserungen, sowohl hinsichtlich der experimentellen Durchführung, insbesondere jedoch hinsichtlich der Qualität des

für die Berechnung der Brechungseigenschaften verwendeten Modells sowie der hierfür erforderlichen Vorkenntnisse.

In der experimentellen Durchführung wurden bislang die applizierten Kreismarken auf dem Monitor aufgeklebt. Diese Vorgehensweise birgt jedoch die Gefahr der Beschädigung des Monitors. Weiterhin wurden zur Messung der Kreismarken zwei Kameras in Form eines stereophotogrammetrischen Systems verwendet. Durch die dabei auftretenden wechselnden Beobachtungswinkeln können zum einen systematische Abweichungen durch die Exzentrizität der Kreismarken auftreten. Zum anderen wird der nutzbare Winkelbereich eingeschränkt, da die Marken stets mit beiden Kameras erfasst werden müssen.

Für die Anwendung des für die Berechnung der Brechungseigenschaften verwendeten Modells war bislang die Kenntnis von Parametern oder die Annahme von Randbedingungen erforderlich. Beispielsweise musste die Dicke der applizierten Kreismarken zuvor gemessen werden. Weiterhin wurde der Einfluss von Ebenheitsabweichungen im betrachteten Bildschirmausschnitt vernachlässigt.

In dem hier vorgeschlagenen Ansatz werden die oben genannten Einschränkungen überwunden [3]. So genügt die Beobachtung der Kreismarkenanordnung mit nur einer Kamera. Die Kreismarken können zudem derart auf die Bildschirmoberfläche appliziert werden, dass zwar weiterhin ein sehr guter Kontakt während der Messung gewährleistet ist, die Marken im Anschluss an die Messung jedoch rückstandsfrei entfernt und für weitere Messungen an anderen Bildschirmen wiederverwendet werden können. Das verwendete Modell wurde dahingehend erweitert, dass eine vorherige Kenntnis der Dicke der applizierten Marken nicht mehr erforderlich ist, sondern diese im Rahmen des Ausgleichs mit hoher Genauigkeit mit bestimmt werden kann. Weiterhin wird nun der Einfluss von Ebenheitsabweichungen im betrachteten Bildschirmbereich berücksichtigt. Mit dem verbesserten Ansatz ist nun bei geringerem experimentellem Aufwand eine deutlich genauere Bestimmung der Brechungseigenschaften möglich.

2 Versuchsdurchführung

Der in den hier vorgestellten Untersuchungen verwendete Aufbau ist in Abbildung 3 gezeigt. Zur Beobachtung der Kreismarkenanordnung ist

nur noch eine Kamera erforderlich, welche auf einem Kreisbogen um das durch die Marken gebildete Muster bewegt wird. Damit steht für die Messungen ein deutlich größerer Winkelbereich zur Verfügung, was Vorteile für die robuste Anpassung des Brechungsmodells bietet, wie in Abschnitt 3 zu sehen sein wird.

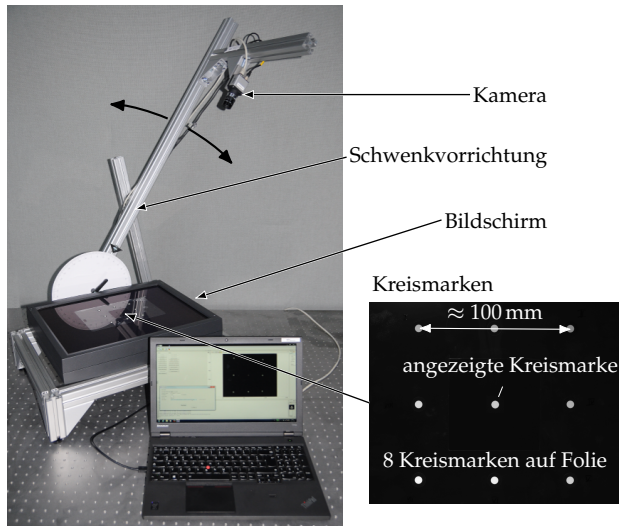


Abbildung 3: Versuchsaufbau, rechts die auf Adhäsionsfolie applizierten Referenzmarken sowie die auf dem Display angezeigte Kreismarke.

Die Kreismarken werden nicht mehr direkt auf dem Monitor, sondern auf einer Adhäsionsfolie aufgeklebt. Dies bietet den Vorteil, dass die Marken im Anschluss an die Messung rückstandsfrei entfernt und für weitere Messungen an anderen Bildschirmen wiederverwendet werden können, gleichzeitig jedoch während der Messung ein sehr guter Kontakt mit der Monitoroberfläche gewährleistet ist. Die Höhe der applizierten Marken oberhalb der Displayoberfläche ergibt sich beim verwendeten Modell zudem aus den Messdaten, so dass eine messtechnische Bestimmung der Markendicke nicht mehr erforderlich ist. Lediglich die Relativpositionen der Referenzmarken müssen zuvor gemessen werden.

Mit dem oben gezeigten Aufbau wird das Muster aus 8 mittels der

Adhäsionsfolie applizierten Referenzmarken sowie einer in deren Mitte auf dem Monitor dargestellten Kreismarke gleicher Größe aus verschiedenen Winkeln im Bereich $\pm 80^\circ$ zur Displaynormalen aufgenommen.¹ Für die folgende Auswertung werden zunächst die Kamerapositionen photogrammetrisch mittels der Referenzmarken bestimmt (Abbildung 4).

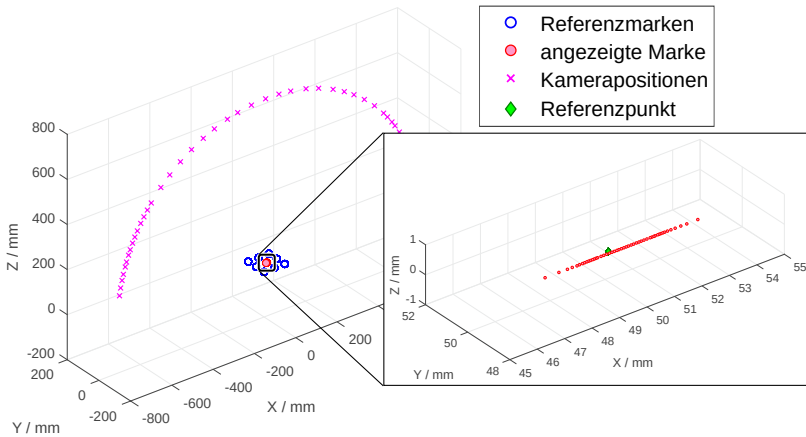


Abbildung 4: Räumliche Anordnung verwendeter Kamerapositionen und Kreismarken. Vergrößerter Ausschnitt: Brechungsbedingter Versatz der durch den Monitor dargestellten zentralen Kreismarke im Vergleich zum Schwerpunkt der 8 Referenzmarken für die dargestellten Kamerapositionen.

Für die auf dem Monitor dargestellte Kreismarke ergibt sich dann in Abhängigkeit vom Beobachtungswinkel ein brechungsbedingter Versatz zum Schwerpunkt der 8 Referenzmarken. (vergrößerter Ausschnitt in Abbildung 4).

3 Anpassung des Brechungsmodells

Der bisherige Ansatz zur Modellierung der Brechungseigenschaften ist in Abbildung 5(a) skizziert. Aus dem Versatz D der angezeigten Kreismarke bei Betrachtung unter dem Winkel α ergibt sich mit der als be-

¹ Im Bereich der auf dem Display angezeigten Marke wurde die Folie entfernt.

kannt angenommenen Höhe h der Referenzmarken die Dicke d und der Brechungsindex n_2 des Glassubstrats des Displays. Die Höhe h ist jedoch nicht exakt bekannt, sondern mit einer Messunsicherheit behaftet. Zudem werden weder die laterale Verschiebung der angezeigten Marke zum Schwerpunkt der Referenzmarken noch Ebenheitsabweichungen, z.B. aufgrund der Durchbiegung des Displays, berücksichtigt. So wird der tatsächliche Blickwinkel auf den untersuchten Displaybereich unter Umständen nicht durch den Winkel repräsentiert, der mittels der durch die Referenzmarken festgelegten Ebene bestimmt wurde.

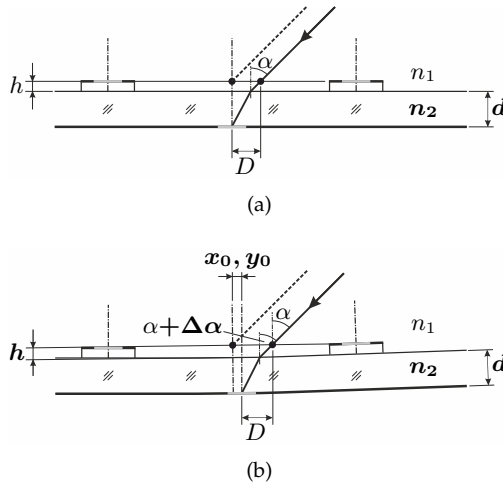


Abbildung 5: Bestimmung der Brechungseigenschaften des Displays mit Hilfe des brechungsbedingten Versatzes D der angezeigten Marke. (a) Bisheriger Modellansatz: applizierte Marken bekannter Höhe h auf (ebenem) Display. (b) Erweiterter Modellansatz: Referenzmarken unbekannter Höhe h , Berücksichtigung von Ebenheitsabweichungen (Änderung $\Delta\alpha$ des Einfallswinkels α) und Offsetkorrektur (x_0, y_0) .

Im hier vorgestellten Modellansatz werden die Höhe h der Referenzmarken sowie eine durch Ebenheitsabweichungen bedingte Veränderung $\Delta\alpha$ des Einfallswinkels α als zusätzliche Parameter in die Auswertung einbezogen, wie in Abbildung 5(b) dargestellt ist.

Vor Anpassung der Modellparameter ist es sinnvoll, die Verschie-

bung (x_0, y_0) der angezeigten Kreismarke zum Schwerpunkt der Referenzmarken zu korrigieren. Trägt man den Versatz der angezeigten Kreismarke zum Schwerpunkt der Referenzmarken über den Winkel auf, so ergibt sich der in Abbildung 6(a) dargestellte Verlauf (“Rohdaten”). In der Regel ergibt sich dabei für den Winkel 0° eine Verschiebung in x - und y -Richtung. Exemplarisch ist die Bestimmung der Verschiebung in x -Richtung gezeigt, die Vorgehensweise für die y -Richtung ist vergleichbar. Aus dem in guter Näherung linearen Bereich um 0° ergibt sich mittels linearer Regression die Verschiebung (x_0, y_0) , so dass die Messdaten entsprechend korrigiert werden können (Abbildung 6(b)).

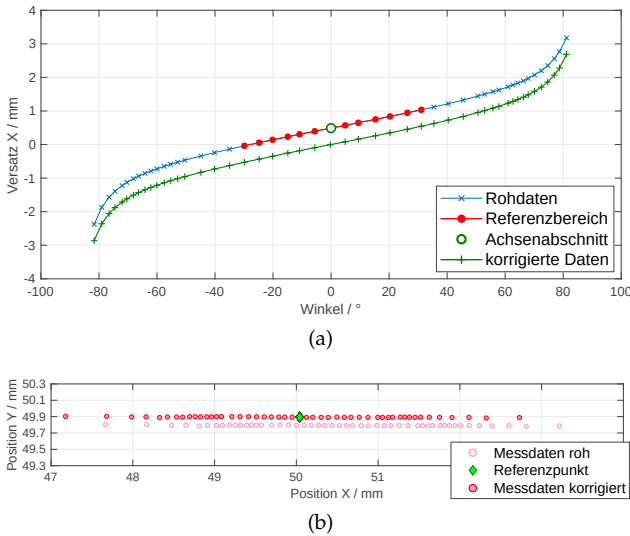


Abbildung 6: Bestimmung der Verschiebung in x -Richtung (a) und Korrektur (b) der Verschiebung (x_0, y_0) der angezeigten Marke zum Schwerpunkt der Referenzmarken.

An die so korrigierten Messdaten werden nun die Modellparameter angepasst. Der beobachtete Versatz D_{gesamt} der angezeigten Marke zum Schwerpunkt der Referenzmarken setzt sich zusammen aus einem Anteil D_{Glas} aufgrund der Brechung im Displaymaterial, sowie aus dem Versatz D_{Marker} , welcher sich geometrisch aufgrund des schrägen

Durchlaufens der Luftschicht der Dicke h (Höhe der Referenzebene) oberhalb der Displayoberfläche ergibt.

$D_{\text{gesamt}} = D_{\text{Glas}} + D_{\text{Marker}}$ mit:

$$D_{\text{Marker}} = h \cdot \tan(\alpha + \Delta\alpha)$$

$$D_{\text{Glas}} = d \cdot \tan \left(\underbrace{\arcsin \left(\sin(\alpha + \Delta\alpha) \cdot \frac{n_1}{n_2} \right)}_{\beta} \right)$$

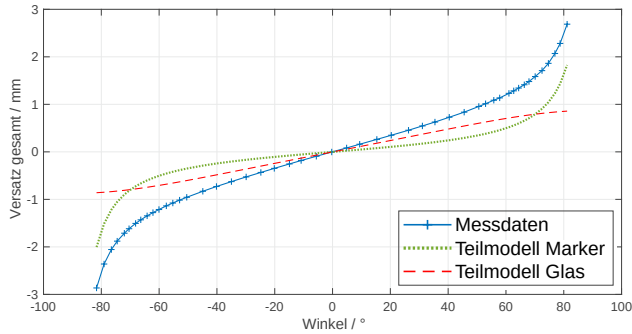
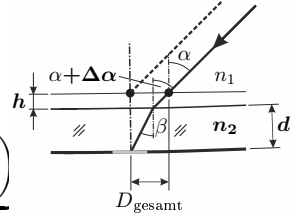


Abbildung 7: Ergebnis der Modellierung des gemessenen Versatzes D_{gesamt} : Beiträge der Teilmodelle Marker und Glas.

In Abbildung 7 sind neben dem gemessenen Versatz D_{gesamt} die Beiträge der Brechung im Glassubstrat sowie des geometrisch bedingten Versatzes in der Luftschicht zwischen Referenzebene und Displayoberfläche zum Gesamtmodell dargestellt. Obwohl beide Beiträge eine ähnliche Auswirkung auf den Versatz der angezeigten Marke haben, ist der gemeinsame Ausgleich beider Effekte möglich, da die Teilmodelle für größere Winkel ($> 60^\circ$) unterschiedliches Verhalten zeigen: Während der Einfluss der Brechung nachlässt, steigt der Anteil des geometriebedingten Versatzes stark an. Hier wird der Vorteil des vorgestellten Verfahrens besonders deutlich. Da durch die Verwendung nur einer Kamera ein größerer Winkelbereich abgedeckt werden kann, können die Einflüsse der beiden Teilmodelle klar unterschieden werden.

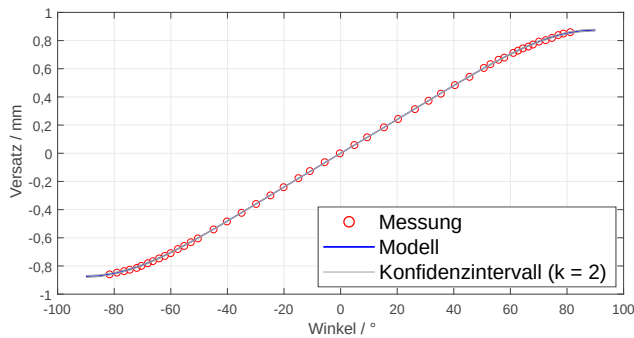


Abbildung 8: Messdaten und Modell des brechungsbedingten Markenversatzes.

Aus dem Teilmodell für die Brechung im Glassubstrat (Abbildung 8) ergeben sich dann die in Tab. 1 angegebenen Parameter.

Größe	Messwert	Standard-unsicherheit	relative Un-sicherheit
Brechungsindex n_2	1,6127	0,013257	0,82 %
Glasdicke d	1,1064 mm	0,011243 mm	1,02 %
Markendicke h	0,28894 mm	0,0006519 mm	0,23 %
Winkelkorrektur $\Delta\alpha$	$-0,14432^\circ$	$0,0062558^\circ$	4,3 %

Tabelle 1: Ergebnisse der Parameteranpassung im Brechungsmodell

Es ist zu beachten, dass die Gesamtunsicherheit des Brechungsmodells allein aus den angegebenen Unsicherheiten der relevanten Einflussgrößen d und n_2 überschätzt wird, da eben diese beiden Größen eine hohe Korrelation aufweisen. Bestimmt man die Gesamtunsicherheit für den brechungsbedingten Markenversatz unter Berücksichtigung dieser Korrelation, ergibt sich das in Abbildung 9 vergrößert dargestellte Konfidenzintervall aus Abbildung 8 für $k = 2$. Für Winkel im Bereich von bis zu etwa $\pm 50^\circ$ liegt danach die Gesamtunsicherheit für den brechungsbedingten Markenversatz unterhalb von $2 \mu\text{m}/0,33 \%$. Selbst für im praktischen Einsatz nicht relevante Winkel von $\pm 80^\circ$ liegt die Gesamtunsicherheit mit $5,9 \mu\text{m}/0,69 \%$ deutlich unterhalb der Unsicherheit der für die Berechnung des Brechungseinflusses wesentlichen Ein-

zelgrößen d und n_2 .

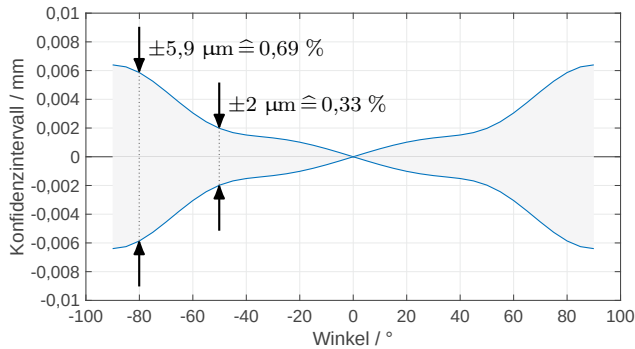


Abbildung 9: Konfidenzintervall ($k=2$) für den Markenversatz.

4 Zusammenfassung

In diesem Beitrag wurde ein weiterentwickeltes Verfahren zur Bestimmung der Brechungseigenschaften von Flüssigkristalldisplays vorgestellt. Wie bei der bisherigen Vorgehensweise wird dazu der brechungsbedingte Versatz einer auf dem Display angezeigten Kreismarke in Relation zu auf dem Monitor applizierten Referenzmarken ausgewertet. Bisher mussten dazu Referenzmarken bekannter Höhe direkt auf dem zu untersuchenden Display aufgebracht werden. Die Erfassung des Versatzes der angezeigten Kreismarke erfolgte dann durch zwei Kameras in Stereuanordnung. Beim weiterentwickelten Verfahren werden die Referenzmarken auf einer Adhäsionsfolie aufgeklebt, wodurch einerseits die Marken nach der Untersuchung rückstandsfrei und ohne Beschädigung des Displays entfernt werden können sowie für weitere Messungen wiederverwendet werden können, andererseits aber während der Messung in sehr gutem Kontakt zur Displayoberfläche stehen. Durch die Verwendung nur noch einer Kamera für die Erfassung der Kreismarken sind Messungen in einem deutlich größeren Winkelbereich (bis $\pm 80^\circ$ statt bisher $\pm 60^\circ$ zur Displaynormalen) möglich. Da gerade in den Randbereichen die den Versatz beeinflussenden Effekte

deutlich unterschiedliches Verhalten zeigen, ist so eine robuste Modellierung der Brechungseigenschaften möglich.

In dem erweiterten Modell wird die Höhe der Referenzmarken nicht mehr als bekannt vorausgesetzt, sondern als freier Parameter berechnet. Zudem wird der Einfluss von Ebenheitsabweichungen des betrachteten Bildausschnitts berücksichtigt. Der laterale Versatz der angezeigten Kreismarke relativ zum Referenzschwerpunkt wird im Vorfeld der Modellanpassung mittels linearer Regression automatisch kompensiert.

Im Zusammenwirken aller Maßnahmen ist mit dem neuen Ansatz bei geringerem experimentellem Aufwand eine deutlich genauere Bestimmung der Brechungseigenschaften möglich. Im Rahmen der durchgeführten Untersuchungen konnten erstmals auch Aussagen zur Messunsicherheit der Methode getroffen und verifiziert werden.

Danksagung

Die Autoren danken der Deutschen Forschungsgemeinschaft (DFG) für die Förderung dieser Arbeit im Rahmen des Projekts Pe1402/6-1.

Literatur

1. M. Petz, M. Fischer, and R. Tutsch, "Systematic errors in deflectometry induced by use of liquid crystal displays as reference structure," in *Proc. 21st IMEKO TC2 Symposium on Photonics in Measurement, 16-18 September 2013, Gdansk, Poland, 2013*, pp. 16–18.
2. M. Fischer, "Deflektometrie in Transmission - Ein neues Messverfahren zur Erfassung der Geometrie asphärischer refraktiver Optiken," Dissertation, TU Braunschweig, Schriftenreihe des Instituts für Produktionsmesstechnik, Band 12, Shaker Verlag Aachen, 2016.
3. M. Petz, H. Dierke, and R. Tutsch, "Photogrammetrische Bestimmung der Brechungseigenschaften von Flüssigkristallbildschirmen für die Deflektometrie." Vortrag anlässlich der 119. Jahrestagung der DGaO, Aalen, Mai 2018.

Single pose camera calibration using a curved display screen

Raul Acuna¹, Robin Ziegler¹, and Volker Willert¹

TU Darmstadt, Control Methods and Robotics Lab,
Landgraf-Georg Straße 4, 64283 Darmstadt

Abstract In this paper, a method for single pose camera calibration is presented. Dense point correspondences are obtained by displaying structured light in a non-flat display screen (i.e. a curved display screen) which then are used as an input in common calibration algorithms. Experimental results demonstrate that the depth information present in a common commercial curved monitor with a radius of curvature of 1800 millimeters is sufficient in order to obtain calibration results comparable to the standard checkerboard method. In contrast to the commonly used checkerboard based calibration methods, the proposed method does not require to move the camera and it is cheaper and easier to implement than other methods based on expensive 3D calibration rigs.

Keywords Camera calibration, structured light, display, curved screen.

1 Introduction

Camera calibration is the first step for optical based 3D measurements [1], e.g. measure the size of an object in world units or determine the location of the camera in world coordinates. Camera calibration methods require the use of known control points in world coordinates which can then be correlated to image points in the camera. These control points are defined by using objects or shapes with known dimensions in world units, therefore either three dimensional or two-dimensional calibration objects are used.

Three-dimensional calibration objects like the one used in [2] are usually complicated to manufacture (and measure) and it is generally hard to cover the whole image space with a good resolution of control points. To overcome these problems Zhang [1] proposed the nowadays most commonly used calibration technique which employs a checkerboard pattern on a flat surface. Since a single shot of a 2D calibration object cannot provide enough constraints to calibrate the camera, the checkerboard must be captured by the camera in at least three different poses. The amount and size of squares on the pattern define the number of points that can be obtained in a single image and the range of the checkerboard detection. Therefore in practice, more than three captures (recommended more than 10) that cover the whole camera image space are required with different orientations and positions.

During the image capture process, usually wrong measurements may be included due to the movement of the checkerboard (blurred images) or because a user with low experience in camera calibration may not understand how to properly place the checkerboard pattern to obtain good results. Additionally, the checkerboard is a fixed planar shape which has a limited optimal detection range, if the checkerboard is too close or too far away from the camera the measurements will have a worse quality than the ones taken at the optimal distance. Some calibration suites try to overcome this by using several calibration checkerboards of different sizes, which increases the complexity of the calibration process in favor of higher accuracy. Some camera calibration suites employ smart user interfaces that guide the user during the checkerboard image acquisition process and allow the removal of wrong captures.

Newer calibration methods use an LCD screen [3] that replaces the printed checkerboard. Using an LCD screen has several advantages, with the manufacturing capacity of nowadays LCD panels a resolution of over 1920×1080 pixels can be achieved with a constant pixel size less than 0.3 millimeters. This allows the precise presentation of any kind of calibration pattern, including a checkerboard, but also smaller patterns with more squares, phase-based shapes where more point correspondences can be obtained or other shapes robust to de-focusing [4]. However, using a flat screen requires still the capture of at least three different poses of the screen by the camera. Some single pose camera calibrations using flat screens involve the positioning of two or more flat screens in order to create a 3D calibration rig [5], but the relative

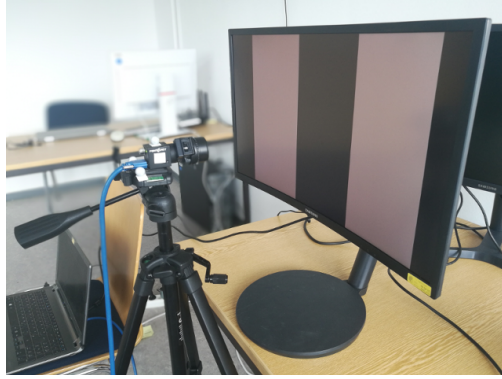


Figure 1: Example of our experimental setup for structured light based calibration of a camera using a curved display screen.

poses of the camera still have to be precisely calibrated incrementing the solution complexity.

In contrast, by using a curved screen, see Figure 1, the depth information may be added to every pixel without additional devices or additional motions. With a good point correspondence system established between pixels in the curved screen and the pixels in the camera image, the curved screen can be used as a 3-dimensional calibration rig with a high amount of control points. This could enable a calibration with only one camera pose reducing the number of user interactions to a minimum. The question to answer is if the radius of curvature of a common commercial curved panel (18000 mm) is enough to provide the required depth information of the control points for a proper calibration and if the standard calibration algorithms can be used with this data. On the following section, we propose and evaluate a calibration method which demonstrates that it is, in fact, possible to perform camera calibration with a curved screen with comparable performance to the state of the art.

2 Proposed calibration method

Our calibration work-flow is based on the flat screen approach with some modifications to suite the special case of a curved screen. It is shown in a general form in Figure 2.

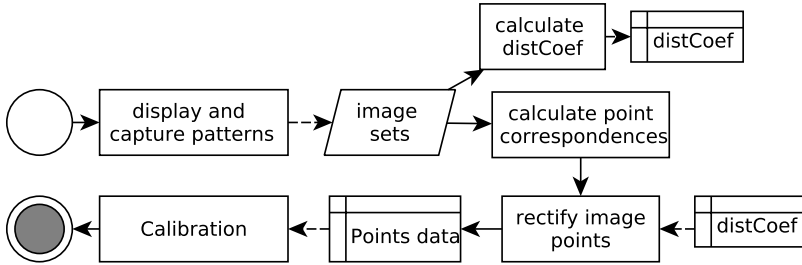


Figure 2: Proposed calibration method.

2.1 Extraction of point correspondences

A time-multiplexing gray coded structured light technique was chosen due to its simple implementation and decoding. This method consists of the successive projecting of patterns on the display. First, horizontal and then vertical ones. The Gray code pattern generation was implemented using standard methods from the *OpenCV* structured light library [6]. Example images corresponding to the third horizontal and third vertical pattern are displayed in Figures 3(a) and 3(b). Example images corresponding to the third horizontal and third vertical pattern are displayed in Figures 3(a) and 3(b).

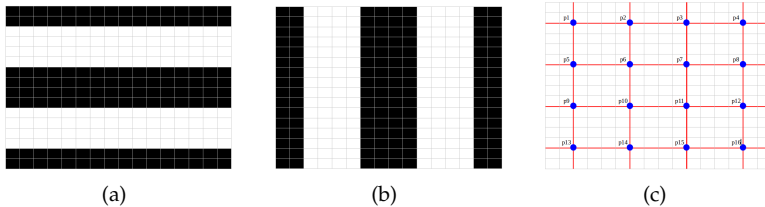


Figure 3: Process of point decoding. a) Third horizontal pattern, b) Third vertical pattern, c) Combined edge maps with marked intersection points.

This technique produces a mapping of several display pixels to a single pixel in the camera image due to the higher display screen resolution compared to the camera. The patterns can be detected only up to a specific resolution until the stripes cannot be differentiated, see Figure 5(a). This problematic effect depends both on the relative camera-display resolutions and on the camera-to-display distance.

Screen pixel coordinates are denoted by $[U, V]$ and camera image pixel coordinates as $[u, v]$. The origin of the world coordinate system is defined in the center of the screen. The total amount of gray coded horizontal and vertical images necessary to completely codify a screen with a resolution of $[S_{width}, S_{height}]$ pixels are:

$$n_u = \log_2(S_{width}), \quad n_v = \log_2(S_{height}). \quad (1)$$

In order to overcome the resolution problem instead of displaying all the n_u and n_v images we only show the patterns up to some maximum horizontal and vertical density parameters D_u and D_v . Each one defines the maximum value used in the gray coded patterns in the horizontal and vertical direction, respectively. This maximum density values assure that the camera will be capable of discerning the edges of the patterns. The currently displayed pattern number is denoted by d . For example, if we set $D_u = 3$ only the patterns $d = 1$, $d = 2$ and $d = 3$ are generated, displayed and captured in the horizontal direction.

For each horizontal or vertical pattern, the edge lines are extracted and for each one, the gray code information of the previous pattern is used to match the line on the image to its correspondent on the display screen. The intersection of the pattern edges plus the gray code information are used to find the camera image to screen point correspondences. This approach is similar to the one proposed by [7].

The width of the black and white columns in a Gray pattern d will be half of the width of the previous Gray pattern $d - 1$, see Figure 4. As consequence, the edge of a row/column in a pattern d will be located exactly on the middle of the row/column of the previous pattern $d - 1$. Hence, each edge on camera image for the current d can be tagged with the gray code of the previous pattern $d - 1$, the gray code of a given horizontal or vertical edge will be denoted by e_u or e_v . For example in Figure 4 the Gray code corresponding to the third edge from left to right is $e_u = 11_g$.



Figure 4: Column extraction process with $d = 3$ and a resolution of 16 pixels, Gray Code.

We can use this information to do the line correspondence between image captured edges and screen pattern edges and obtain its screen pixel coordinates by applying the following equations:

$$U = ((1 + G'(e_u)) - 0.5)w_u, \quad V = ((1 + G'(e_v)) - 0.5)w_v, \quad (2)$$

where the function $G(x), x_{dec} \mapsto x_{gray}$ represents the conversion from decimal values to Gray coded values and its inverse is denoted as $G'(x)$. For a given pattern number d , the width of a column is defined as w_u for horizontal patterns and the width of a row as w_v for vertical patterns. Both w_u and w_v are in screen pixel coordinates and calculated by the following equations:

$$w_u = \frac{2^{n_u}}{2^{d-1}}, \quad w_v = \frac{2^{n_v}}{2^{d-1}}. \quad (3)$$

Finally, we obtain the world coordinates of the point by using the radius of curvature ρ and the pixel pitch of the screen in x and y directions ϕ_x, ϕ_y (values provided by the screen manufacturer):

$$v = (U - S_{width}/2)\phi_x, \quad (4)$$

$$\varphi = (V - S_{height}/2)\phi_y, \quad (5)$$

$$[X, Y, Z] = [\sin(v/\rho) * \rho, -\varphi, \cos(v/\rho) * \rho - \rho]. \quad (6)$$

The accuracy of the point extraction process depends mainly on three factors: 1) external light sources, 2) camera placement and 3) performance of the edge detector, see Figure 5. It is recommended to place the

screen in a place where no reflections of other light sources can interfere with the camera capture. The ideal placement to cover the screen without degradation of the line detection is front-to-parallel, with the screen covering as much as possible on the camera image.

To improve the performance of our edge detection an upscaling of the raw images using the Gaussian pyramid method and then the application of a Gaussian convolution kernel to estimate the values of the new missing pixels is performed. This results in a high-resolution blurred image. A Canny edge detector is then applied to the resulting image. Additionally, the display of a full black followed by a full white image on the screen is used to calibrate the thresholding of the patterns which allows an accurate recognition of the contour of the panel and the definition of a region of interest in the camera image for the processing.

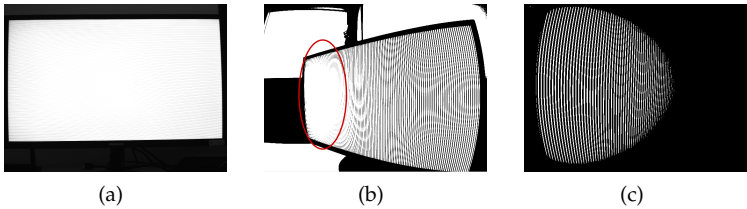


Figure 5: Potential problems in image acquisition. a) Gray Coded pattern with a resolution that cannot be resolved by the camera, b) Influence of other light sources, c) Effect of an inclined camera placement.

2.2 Non metric image rectification preprocessing

The obtained points produced by the point correspondence system can be used as input for a classical camera calibration algorithm (e.g. the OpenCV calibrate camera function). Empirical results with different lenses showed that for points obtained by placing the camera directly in front of the curved monitor the calibration process produces wrong results. It is our guess that the calibration method cannot differentiate between the lens distortion and the screen curvature. It is possible to obtain good calibration results by using a steeper camera angle relative to the screen, which increases the relative depth change. However, the more the camera is tilted, the fewer point correspondences are obtained

in the far field. Hence, our proposed solution consists of a preprocessing step, in the form of a nonmetric image rectification.

A nonmetric rectification method exploits the constraint that straight lines in the scene must remain so in the projected space [8]. A least squares approach is used to find the distortion parameters. First, the distortion error is defined as the sum of squared distances from the detected edge points to an approximated line distorted by some initial distortion parameters. These distortion parameters are then optimized iteratively by minimizing the error.

We used this approach in combination with the one parameter division model proposed by Fitzgibbon [9], where the distortion is described in camera coordinates as:

$$x_d = x \frac{1}{1 + k_1 r^2}, \quad y_d = y \frac{1}{1 + k_2 r^2}, \quad (7)$$

where x_d, y_d are the distorted camera coordinates, x, y the undistorted ones, $r = \sqrt{x^2 + y^2}$ and k_1, k_2 are the distortion parameters to be found. It is safe to assume that the distortion parameter is the same in each axis so $k = k_1 = k_2$. In order to not confuse this distortion parameter with the produced by OpenCV or Matlab calibration suites when performing the calibration we are going to denote k as γ . And then in camera pixel coordinates equation (7) becomes:

$$u_d = (u - u_0) \frac{1}{1 + \gamma_1 r^2} - u_0, \quad v_d = (v - v_0) \frac{1}{1 + \gamma_1 r^2} - v_0, \quad (8)$$

with $u_0 = C_{width}/2$ and $v_0 = C_{height}/2$ for a camera with image size $[C_{width}, C_{height}]$. For simplification reasons, the center of distortion is assumed to be the center of the image which is a good approximation.

Once γ is found, the edge intersections in camera coordinates are rectified and together with the world coordinates are used as inputs for the camera calibration functions of Matlab and OpenCV.

3 Results and discussion

To test the performance of our calibration method a curved screen with a radius of curvature of 1800 millimeters was used. The cameras used

was a *FLIR Blackfly BFLY-U3-13S2C-CS* from Point Grey with a resolution of 1288×964 pixels. Three different focal lengths were selected. For each focal length, the camera was placed in front of the screen so the whole display area was within the camera image and the point correspondences, nonmetric rectification and calibration with both Matlab and OpenCV was performed. As a comparison, we also calibrated the camera using the traditional checkerboard method and with a planar LCD screen (3 views of the planar screen).

An example of the point correspondences obtained with our method when using a density parameter $d = 7$ is shown in Figure 6. Notice the density of the acquired points and their regularity.

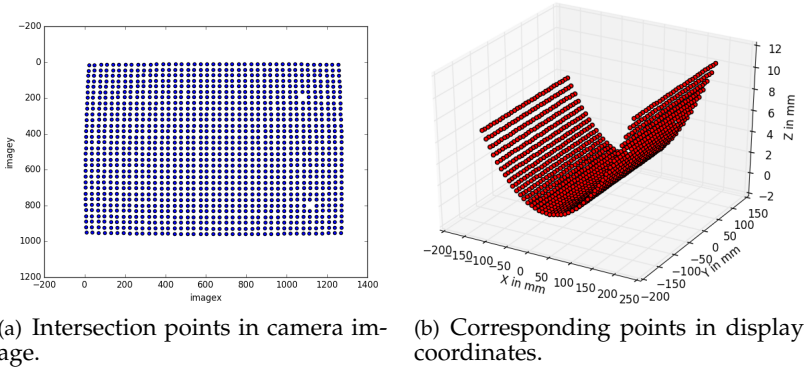


Figure 6: Point correspondences obtained from a curved screen with a camera perpendicular to the screen ($d = 7$).

The camera intrinsics results obtained for the 4mm focal length for all the compared methods are shown in Table 1. The error is the reprojection error, calculated using the following equation:

$$error = \sqrt{\frac{\sum_{i=1}^N (x_i - x'_i)^2 + (y_i - y'_i)^2}{N}}, \quad (9)$$

where x'_i, y'_i are the reprojected image point coordinates and N the total amount of points. This is the default error on OpenCV and was cal-

culated as well from the data of the Matlab Calibration Toolbox (which in contrast uses by default the standard deviation of the pixel error).

Parameter	Curved screen (<i>OpenCV</i>)	Curved screen (<i>MATLAB</i>)	Flat screen (<i>Matlab</i>)	Checkerboard (<i>OpenCV</i>)
fx	1071.912	1084.156	1065.406	1074.786
fy	1072.538	1084.777	1054.065	1074.958
cx	632.110	632.444	628.077	627.898
cy	508.721	508.295	495.022	493.513
k1	0.0312	0.02234	-0,35959	-0.39147
k2	-0.0655	-0.03323	0,13475	0.24112
k3	0.0051	0.00522	0	-0.63883
p1	-0.0045	-0.00452	-0,00014	-0.62106
p2	0.0316	0	0,00013	-0.10542
γ	-3.607e-07	-3,607e-07	0	0
Error	0.54828	0.55240	0.51886	0.33515

Table 1: Camera parameters obtained using the curved screen compared to other calibration methods.

A comparison of the calibration results for different focal lengths in terms of the reprojection error are show in Table 2 and the graphical calibration results of the *Matlab Calibration Toolbox* are shown in Figure 7.

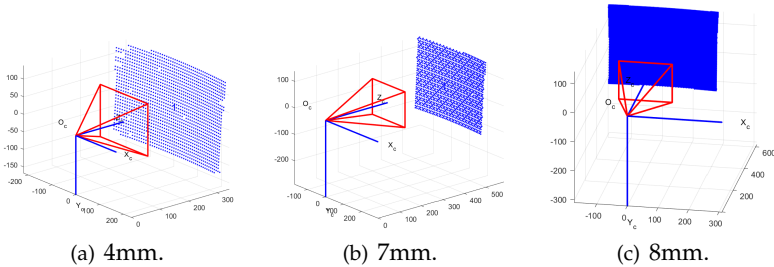


Figure 7: Point correspondences obtained from a curved screen with a camera perpendicular to the screen ($d = 7$).

The curved screen approach achieves sub-pixel reprojection errors, with values comparable to the 3-views flat screen approach and higher

Focal length	Curved OpenCV	Curved Matlab	checkerboard
4mm	0.5482	0,5524	0,3352
7,5mm	0.4670	0.4671	0,3341
8,4mm	0.4808	0.4812	0,3221

Table 2: Re-projection error in px units for different focal lengths using the curved screen method.

than the checkerboard method. This is remarkable considering that only one camera pose is needed. Comparing the camera intrinsics obtained with the curved screen to those of the checkerboard we can notice that the parameters corresponding to the x axis of the camera present better results than the ones on the y axis. This could be a product of the lower amount of depth present in that axis in the curved screen. More tests are required with different display orientations in order to overcome this dependency. It is possible to observe that the $k1$ parameter is reduced considerably in the curved screen results. This is a consequence of performing the one parameter division model rectification preprocessing step.

We notice in Table 2 that our method works better for larger focal lengths. We believe that it is due to the low radius of curvature (1800 mm) of our screen. It is possible that a screen with a higher radius of curvature will provide better results. In terms of speed, our method is the fastest. The whole process of placing the camera in front of the screen, displaying and acquiring the images and calibrating takes less than one minute with minimal user interaction which compared to the checkerboard method is an important advantage.

4 Conclusion

A single pose automatic camera calibration method was presented. The novelty of the approach lies in the use of a curved display screen which allows the capture of dense point correspondences without moving the camera. Since traditional methods will fail when directly using this data due to the screen curvature our pipeline introduces a preprocessing step in the form of a non-metric rectification using the single parameter division model. Experimental results confirm, that our method achieves sub-pixel reprojection errors and has comparable performance to the

standard checkerboard method with the added advantage of not having to move the camera. The whole process is made in an automatic way by the software with a minimum of user interaction. However, there is still a great margin for improvement. First, the refraction of the screen has not been considered which has been tackled in flat screens based methods with an increase in accuracy. Phase-based patterns may be used instead of the gray code approach which may give denser and more accurate points and finally, a more complex distortion model can be employed for the preprocessing rectification step.

References

1. Z. Zhang, "A Flexible New Technique for Camera Calibration (Technical Report)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2002.
2. J. Heikkila, "Geometric camera calibration using circular control points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1066–1077, 2000.
3. F. Forster, "Camera calibration: active versus passive targets," *Optical Engineering*, vol. 50, no. 11, p. 113601, 2011.
4. B. Cai, Y. Wang, K. Wang, M. Ma, and X. Chen, "Camera calibration robust to defocus using phase-shifting patterns," *Sensors (Switzerland)*, vol. 17, no. 10, 2017.
5. E. Grossmann, J. Woodfill, and G. Gordon, "Display screen for camera calibration," *US Patent*, 2011.
6. Q. Gu, K. Herakleous, and C. Poullis, "3DUNDERWORLD-SLS: An Open-Source Structured-Light Scanning System for Rapid Geometry Acquisition," *arXiv*, no. May 2015, 2014.
7. R. Sagawa, M. Takatsuji, T. Echigo, and Y. Yagi, "Calibration of lens distortion by structured-light scanning," *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, no. Lcd, pp. 1349–1354, 2005.
8. El-Melegy and Farag, "Nonmetric lens distortion calibration: closed-form solutions, robust estimation and model selection," in *Proceedings Ninth IEEE International Conference on Computer Vision*, no. Iccv. IEEE, 2003, pp. 554–559 vol.1.
9. A. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I-125–I-132, 2001.

100% Oberflächeninspektion in der Karosseriefertigung

100% surface inspection of press parts in the automobile industry

Florian Loibl¹

Daimler AG,
Mercedes-Benz Werk Bremen

Zusammenfassung Die 100%-Oberflächeninspektion stellt sowohl in Hinblick auf die Absicherung von Fehlern, als auch für die Analyse und den Zusammenhang zwischen Einstellung der Pressenlinie und Änderung von Fehlerbildern eine erforderliche Technologie dar. Zudem ist eine Regelung von Pressenlinien mit dem Ziel einer konstanten Qualität nur mittels einer Gesamtausgabe der Fertigteilqualität erreichbar. Für Folgegewerke stellt die 100%-Oberflächeninspektion eine Möglichkeit dar, komplette Karossen auf Oberflächenfehler zu überprüfen.

Schlagwörter Oberflächeninspektion, Qualität.

Abstract 100% surface inspection can be seen as an enabling technology for a complete information about the surface quality in detail. It is relevant for finding the defect in the press line as well as to analyze the root cause of it. With regard to paint or body shop the 100% surface control is a possibility to achieve a full check of the whole car.

Keywords Surface inspection, quality.

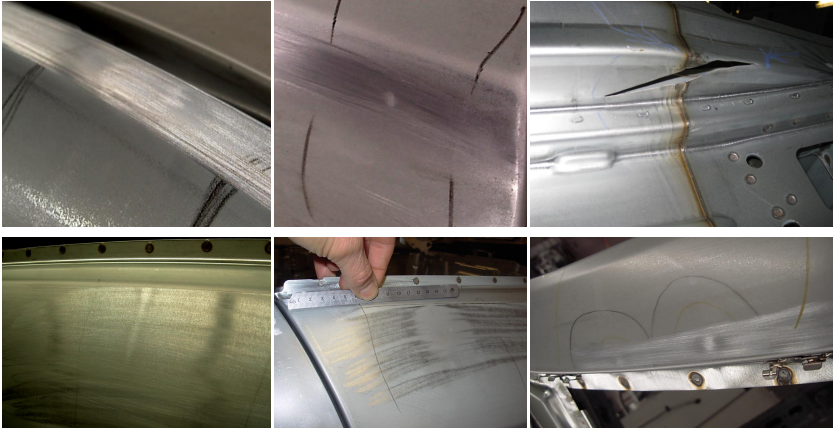


Abbildung 1: Fehlerarten auf Karosserieteilen: Ölbeule, Druckstelle, Riss, Abdruck, Einfallstelle, Pickel. [Quelle: Daimler AG]

1 Motivation für eine vollständige Prüfabdeckung der Pressteiloberfläche

1.1 Aufkommen u. Varianz von Karosseriefehlern

Die Fertigung von Pressteilen in der Automobilindustrie erfolgt im Presswerk unter Verwendung mehrerer Operationsstufen (max. 6), die durch Presswerkzeuge abgebildet werden. Aus der geschnittenen „Platine“ wird zunächst in der Ziehoperation ein Ziehteil hergestellt; dieses besitzt bereits die finale Außengeometrie des fertigen Bauteils. In bis zu fünf Folgeoperationen werden Beschnitte und Lochungen durchgeführt. Jeder dieser Prozessschritte hat einen Einfluss auf bestimmte Eigenschaften des Bauteils, die Ziehoperation jedoch die größte in Hinblick auf die Oberflächenqualität. Einige der dabei potentiell entstehenden Fehlerarten sind in Abbildung 1 dargestellt. Ausprägung und Form dieser Fehlerbilder hängen mit verschiedenen Einflussgrößen beim Ziehprozess wie Druckparametern und/oder Menge der verwendeten Zubeölung ab.

1.2 Auswirkung von Fehlern in Folgebereichen

Während einige dieser Fehler in denen dem Presswerk nachgelagerten Bereichen wie Rohbau oder Lackierung nachgearbeitet werden können, führen andere zu Ausschuss; Bsp. gerissenes Bauteil. Dennoch können auch Beulen oder Einfallstellen eine Fehlerschwere aufweisen, dass eine Nacharbeit sehr aufwendig oder nicht möglich ist.

1.3 Schwierigkeit der Maßstabsfindung durch den Menschen

Ein vollständig dem CAD-Zustand entsprechendes Bauteil, das keine Spuren des Umformprozesses aufweist, ist nicht herstellbar. An Bauteilbereichen mit hohem Umformgrad kommt es zu Welligkeiten, die in einer bestimmten Ausprägung dem Normalzustand entsprechen. Diese sind ein Musterzustand, nach dem eine Qualitätsprüfung in manueller Form stattfindet. Die Schwierigkeit für das menschliche Auge besteht hierbei darin, dass solche Fehlerbilder gleichsam nicht sichtbar sind, ohne entsprechende Werkzeuge wie „Abziehsteine“ einzusetzen, mit denen lokale Unebenheiten in Form eines Schleifbilds (vgl. Abbildung 1) herausgearbeitet werden. Bei einer Fertigungsgeschwindigkeit von mehr als 30 Bauteilen in der Minute ist das Abziehen der Bauteile praktisch nicht möglich; zumal diese nicht beschädigt werden dürfen.

1.4 Heutiger Stand der Qualitätsabsicherung im Presswerk

An den Servo-Pressenlinien der Daimler AG kommt aus diesem Grunde ein automatisiertes Oberflächeninspektionssystem der Fa. Zeiss (ehemals Steinbichler) in Verbindung mit vier Robotern zum Einsatz, das mittels Streifenlichtprojektion ein 3D-Modell der Oberfläche erzeugt, anhand dessen lokale Unebenheiten auf dem Bauteil bewertet und zur Anzeige gebracht werden können¹. Eine Kombination aus einer Bandkontrolle, bei der jedes Bauteil auf wenige **Fehlerschwerpunkte** kontrolliert werden kann, sowie einer Stichprobenprüfung, bei der durchschnittlich jedes zehnte Bauteil vollständig geprüft wird, sichern in Verbindung mit einem Operator an jedem der zwei Pressenauslaufbänder die Qualität der Fertigteile. Die Fehlerschwere wird auf einer frei

¹ [1]

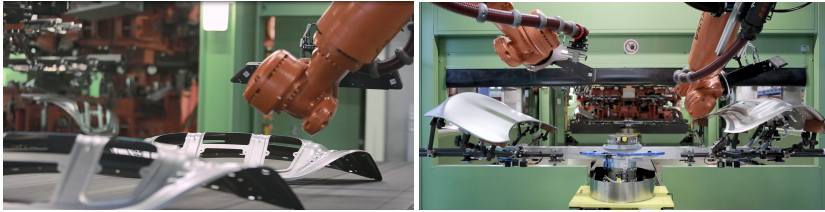


Abbildung 2: Derzeitige Kombination aus automatisierter Bandkontrolle und Stichprobenprüfung. [Quelle: Daimler AG]

wählbaren Skala dargestellt, sodass auch für technisch weniger versierte Mitarbeiter eine Einstellung des Systems möglich ist.

1.5 Nachteile und Fehlerschlupf

Für verschiedene Fehlerbilder an den Bauteilen können so genannte Prüfzonen z_n definiert werden. Diese beinhalten wiederum Prüfpunkte, z.B. Radlauf und Bordkante. Je nach aktuellem Zustand des Bauteils kann die Prüfzone durch den Operator geändert werden. Es ist darüber hinaus eine alternierende Prüfung möglich, d.h. bei Bauteil $[n]$ wird $\leftarrow z_1$, bei $[n + 1] \leftarrow z_2$ usw. geprüft. Dennoch ist heute keine Abdeckung aller Fehlerschwerpunkte bei jedem Bauteil darstellbar, sodass das Fertigungslos zum Teil sortiert werden muss.

2 Ansatz zur Lösung mit kooperierenden Industrierobotern

2.1 Systemische Verfügbarkeit (Automation und Messtechnik)

Für eine Erreichung einer vollständigen Abdeckung der Oberflächenschwerpunkte bei jedem einzelnen Bauteil ist eine Robotertechnik in der heute verwendeten Form (120kg Traglast) nicht geeignet. Der Fokus liegt vielmehr auf einer Erhöhung der Achsgeschwindigkeit, da die durchschnittliche Anzahl Prüfpositionen je Bauteil bei etwa $\sum_P = 50$ liegt, die in etwa 3.3sec durch die Robotik erreicht werden müssen. Da die Anzahl Positionen pro Sekunde nach dem erwähnten Beispiel bei ~ 15 läge, ist mit einer durchschnittlichen Bauteilgröße von

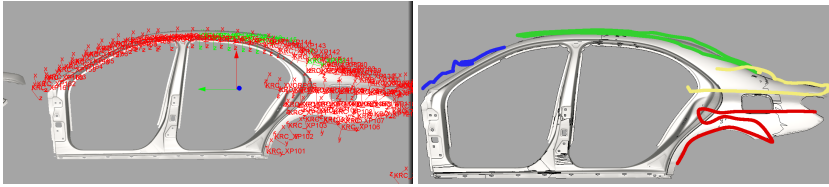


Abbildung 3: Zur Simulation der 100% Kontrolle herangezogene Punktwolke (1) sowie resultierende SPLINE-Roboterbahnen des KUKA Agilus. [Quelle: Daimler AG]

ca. 0.7m^2 sowie einem Messpositionsabstand von etwa 220mm diese Aufgabe aufgrund der zu erreichenden Achsgeschwindigkeiten nicht mit einem einzelnen Roboter lösbar. Das eingesetzte Bildverarbeitungssystem der Fa. Zeiss ist für eine Bildaufnahme in der geforderten Frequenz ebenfalls nicht geeignet.

2.2 Simulative Absicherung der Roboterbewegung anhand vorhandener Prüfprogramme

Somit ist eine Übertragung der Prüfprogramme auf eine Multi-Sensor sowie Multi-Roboterumgebung umzusetzen. Zu diesem Zweck lassen sich die Datensätze aus der Stichprobenprüfung der aktuellen Maschine heranziehen, da hier die vollständige Oberfläche mit Prüfpositionen abgedeckt ist. Zudem kommt die Fahrbewegung SPLINE zum Einsatz, die eine Funktion höherer Ordnung als Bewegungsform durch die Prüfpunkte berechnet, sodass keine Abweichung, sowohl lateral als auch in der Orientierung zum Sollpunkt vorliegt. Das Prüfsystem erlaubt Relativgeschwindigkeiten von $\sim 5 \frac{\text{m}}{\text{s}}$ zwischen Bauteil und Sensor, demnach kann die volle Verfahrensgeschwindigkeit der KUKA Agilus Roboter genutzt werden. (vgl. Abbildung 3). Aus der Simulation der relevanten Außenhaut-Bauteile ergab sich eine Mindestanzahl von **vier** Industrierobotern mit zusätzlichen Linearachsen, die für die Bauteilprüfung eingesetzt werden müssen (siehe Abbildung 4). Der Roboterhersteller bietet hierfür Multi-Roboterumgebungen (RoboTeam) an, mit denen sich diese koppeln lassen. Aufgrund der zeitkritischen Operation im Fertigungstakt der Pressenlinie ist darüber hinaus allerdings eine zeitlich homogene Aufteilung der Prüfpunkte zu den einzelnen

Prüfrobotern erforderlich, damit die zur Verfügung stehende Zeit optimal genutzt werden kann.

2.3 Optimierungsproblem bei der Programmierung

Die Programmierung kann anhand der Stichproben-Punktesätze erfolgen, wobei die Zuordnung Punkt p_i zum Roboter R_n nicht trivial ist. Grundsätzlich lässt sich die Thematik auf das *Traveling Salesman Problem* (TSP, [2]) zurückführen, bei der zu einer gegebenen Anzahl Orte (Kostenfunktion Ort_{*i*} zu Ort_{*j*}: c_{ij}) die kürzeste Route² zu finden ist, bei der alle Orte besucht werden, und die letzte gleich der ersten Station ist (z.B. eine HOME-Position des einzelnen Roboters). Das TSP wird definiert durch $\min \left(\sum_{i=1}^n \cdot \sum_{j \neq i, j=1}^n c_{ij} \cdot x_{ij} \right)$, wobei das Problem in diesem Fall multipel auftritt (mehrere Roboter), und nicht die Gesamtstrecke zu minimieren ist, sondern die Strecke der längsten Teilroute (mit dem Ziel der Homogenisierung). Damit erschwert sich die Fragestellung hin zu einem *minimum-maximum multiple depot vehicle routing problem*, bei der die Trajektorien der Roboter hinsichtlich des eigentlichen TSP sowie im Verbund miteinander (Kollision) optimiert werden müssen. Detailliert ist die Lösung des Problems sowie eine C++-Implementierung beschrieben in [3].

2.4 Integration in den Prozess, Offline-Programmierungsumgebung

Die optimierten Roboterbahnen können nun für eine Bildaufnahme (zwingend für die Programmierung des Bildverarbeitungssystems erforderlich) verwendet werden. Durch die geringe Verfügbarkeit der Pressenlinie muss diese Arbeit offline geschehen. Weiterhin ist eine Optimierung des RoboTeam-Verbunds zeitlich sinnvoller in einer Offline-Umgebung darstellbar. Die geometrische Anordnung der Roboter entspricht in diesem *Offline-Simulator*, Abbildung 5, exakt derjenigen in der Pressenlinie, sodass eine Kopie mit entsprechender Verschiebung der Roboter-Base durch eine Bauteilerkennung am Auslaufband möglich wird.

² binäre Variable $x_{ij} = 1$, wenn Route zwischen den Orten existiert und 0 sonst.

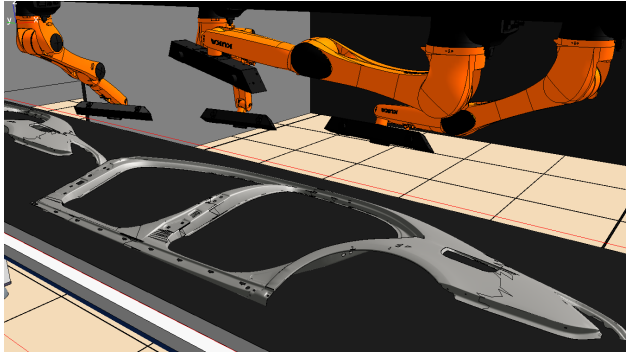


Abbildung 4: Finale Anordnung der Roboter, um das gesamte Bauteilspektrum des XL-Segments abzubilden. [Quelle: Daimler AG]

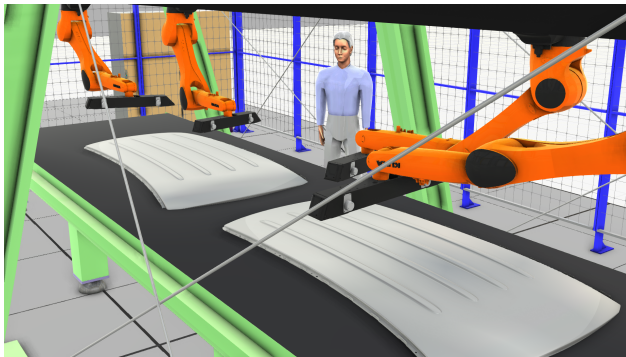


Abbildung 5: Konzeptionelle Darstellung des Offline-Simulators mit exaktem geometrischen Bezug zur Inline-Messtechnik. [Quelle: Daimler AG]

2.5 Verwendung der Daten für eine Regelung basierend auf Bauteilqualität

Aus der Oberflächenprüfung werden bereits heute mit der Band- und Stichprobenprüfung wichtige Qualitätsdaten gewonnen, die dazu beitragen, Einstellungsänderungen an der Pressenlinie durchzuführen. Viele der in Abschnitt 1 dargestellten Fehlerarten sind durch entsprechende Druck-, oder Beölungsänderungen zu beheben, bzw. in ihrer

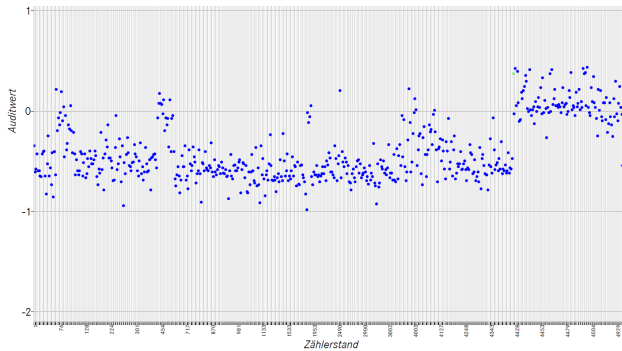


Abbildung 6: Darstellung des Verlaufs der Fehlerschwere eines gewählten Fehlerbilds. Größere Werte entsprechen einer Verschlechterung. Die Sprunghafte Änderung des Qualitätszustands beim Zählerstand 4428 ist deutlich zu erkennen. [Quelle: DatenbankanalyseTool, Daimler AG]

Ausprägung zu verbessern. Was dem Anlagenführer heute jedoch fehlt, ist eine quantitative Aussage über die Gesamtqualität jedes einzelnen Fertigteils. Bezogen auf einzelne Fehlerbilder lässt sich aus der automatisierten Bandkontrolle zwar eine Information ableiten, gleichwohl sind Fehler wie z.B. Risse und Welligkeiten allerdings genau gegensätzlich von Einstellungsänderungen betroffen (Welligkeit zu hoch → Ölmenge verringern → mehr „Auszug“ der Oberfläche, erhöhte Dehnung → Rissbildung wird begünstigt). Eine anspruchsvolle Aufgabenstellung für den Maschinenführer ist demnach in diesem Beispiel das Bestimmen eines Arbeitspunktes (Druck, Öl, etc.), in dem beide Fehlerbilder in Ordnung sind. Das ist wiederum übertragbar auf viele andere Fehler (Einzug, Einfallstellen, etc.). Somit lässt sich die Datengrundlage, die in einem Auszug in Abbildung 6 gezeigt ist, für eine Regelung dieses Arbeitspunktes nutzen. Erforderlich hierfür ist die Verwendung der Eingangsgröße (**Materialparameter**), der Pressenparametern (**Druck, Öl, etc.**) sowie der Fertigteilsqualität (**100% Kontrolle**). Dies ist im Gesamtprojekt im Presswerkverbund der Daimler AG als Arbeitspaket definiert. Zu diesem Thema wird ein Experiment für die beispielhafte Bestimmung eines Arbeitspunktes im Presswerk Bremen gemeinsam mit dem Fraunhofer IOSB stattfinden.

2.6 Technologietransfer

Die dargestellte Technologie zur Implementierung einer 100%-Oberflächeninspektion im Presswerk lässt sich bezogen auf die Robotik ebenfalls in Folgegewerken, wie Rohbau oder Lackierung einsetzen. Denkbar ist die Ausgestaltung der unter Abschnitt 2.4 erläuterten Offline-Programmierung auf die Weise, dass sowohl die Pressenlinien des Presswerksverbunds, als auch Inline-Prüfzellen in den anderen Gewerken hiermit abgedeckt werden können. Damit ist auf der einen Seite eine Effizienz bei der Programmierung, als auch eine Durchgängigkeit des zugrunde gelegten Maßstabs bei der Bewertung der Fehlerbilder gesichert. Somit stellt die Technologie den Grundstein für eine Komplett-Karosserienprüfung dar, bei der die Anforderungen hinsichtlich Geschwindigkeit und Genauigkeit noch wesentlich höher sind.

3 Zusammenfassung

Die 100%-Oberflächeninspektion stellt sowohl in Hinblick auf die Absicherung von Fehlern, als auch für die Analyse und den Zusammenhang zwischen Einstellung der Pressenlinie und Änderung von Fehlerbildern eine erforderliche Technologie dar. Zudem ist eine Regelung von Pressenlinien mit dem Ziel einer konstanten Qualität nur mittels einer Gesamtaussage der Fertigteilqualität erreichbar. Für Folgegewerke stellt die 100%-Oberflächeninspektion eine Möglichkeit dar, komplette Karossen auf Oberflächenfehler zu überprüfen.

Literatur

1. F. Loibl, "Wirtschaftlichkeit und prozessintegration eines automatisierten oberflächeninspektionssystems," 2013.
2. M. u. Held, "Aspects of the traveling salesman problem," *IBM Journal of Research and Development* 28, vol. 4, pp. 476–486, 2004.
3. T. N. Wissel, "Bestimmung und optimierung von kooperierenden roboterbahnen," pp. 46–53, 2017.

Kalibrierfreie 3D-Rekonstruktion von spiegelnden Oberflächen mittels Orthogonalprojektionen unter Verwendung von Reflexionsmarkern

Calibration-free 3D reconstruction of specular surfaces
with orthogonal projections by using reflection markers

Christian Negara¹ und Thomas Längle¹

Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung
Fraunhoferstraße 1, D-76131 Karlsruhe

Zusammenfassung In diesem Artikel wird ein Verfahren zur 3D-Rekonstruktion von Freiformflächen vorgestellt, das sich besonders in Laborumgebungen zur einfachen Erfassung der 3D-Form mit einer telezentrischen Optik und telezentrischem, ko-axialen Beleuchtung eignet. Die untersuchte Oberfläche des Prüflings kann dabei eben oder gekrümmt und transparent oder spiegelnd sein und wird durch auf der Oberfläche aufgebrachten retroreflektierenden Markern erfasst.

Schlagwörter 3D-Rekonstruktion, spiegelnde Oberflächen, B-Spline-Oberfläche, Retroreflexion.

Abstract A method for 3d reconstruction of free-form surfaces is proposed, which is especially suited for laboratory applications to easily estimate the 3d form of a specimen with a telecentric optic and a telecentric, co-axial illumination. The surface under inspection can be flat or curved and transparent or reflective. The 3d-reconstruction of the surface is performed by acquiring images of retroreflective markers which are put on the surface.

Keywords 3D-reconstruction, specular surfaces, b-spline surface, retroreflection.

1 Einleitung

Das hier beschriebene Verfahren zur 3D-Rekonstruktion schätzt die Oberflächenform mit einer telezentrischen Optik/Kamera und telezentrischer, koaxialer Beleuchtung ohne Verwendung von zusätzlicher Sensorik bzw. heterogenen Sensordaten. Dabei wird die Probe mit retroreflektierenden Markern versehen und Bilder in Form von Orthogonalprojektionen aus unterschiedlichen Beobachtungsrichtungen aufgenommen. Die Marker sind kleine Schnipsel aus retroreflektierendem Material. Das schwierige Problem der Korrespondenzsuche von Pixeln in mehreren Bildern bei (teil-)spiegelnden Oberflächen (vgl. [1]) wird durch Verwendung von Reflexionsmarkern stark vereinfacht. Das Korrespondenzproblem wird mit einem Algorithmus zur Berechnung des dichten optischen Flusses für nicht-starre Szenen aus mehreren Kamerabildern gelöst, auf den in Abschnitt 2.1 eingegangen wird. In Abschnitt 2.2 wird ein Algorithmus zur simultanen Schätzung der Kameraposen und der 3D-Koordinaten der Marker aus Orthogonalprojektionen behandelt. Die Regionen der Oberfläche, die nicht durch Marker bedeckt und im Bild sichtbar sind, können z.B. durch Fitting einer B-Spline-Oberfläche interpoliert werden. Hierfür kommt ein probabilistischer Algorithmus zur Oberflächen-Rekonstruktion zum Einsatz, auf den in Abschnitt 2.3 eingegangen wird. Der Artikel schließt mit einer Zusammenfassung in Abschnitt 4 ab.

2 Stand der Technik

Das in diesem Artikel vorgestellte Verfahren kombiniert verschiedene Algorithmen der Bildverarbeitung zu einer Datenverarbeitungskette, um aus den aufgenommenen Orthogonalprojektionen einer Oberfläche zunächst den optischen Fluss, dann eine 3D-Punktwolke und schließlich eine parametrischen Oberflächenrepräsentation zu erhalten.

2.1 Lösung des Korrespondenzproblems bei mehreren unbekannten Kameraposen

Bei bekannten Kameraposen wird die Suche von korrespondierenden Kamerapixeln in zwei Bildern über die Epipolarbedingung gelöst, wo-

mit sich die zweidimensionale Suche auf eine eindimensionale Suche reduziert [1]. Bei unbekannten Kameraposen hingegen müssen die Punkt-korrespondenzen über eine zweidimensionale Suche auf der Projektionsebene der Kamera erfolgen. Für die Korrespondenzanalyse wurde ein sehr allgemeiner Algorithmus zur Berechnung des optischen Flusses bei dynamischen, nicht-starren Szenen mittels mehreren Kamerabil-dern verwendet [2]. Eine Implementierung des Algorithmus ist zudem öffentlich zugänglich¹. Sei $I(\mathbf{x}; i) : \Omega \times \{1, \dots, m\} \rightarrow \mathbb{R}$ eine Sequenz von m Bildern und $\Omega \subset \mathbb{R}^2$ der Ortsbereich der Bildinformation. Die 2D-Trajektorie eines Pixels $\mathbf{x}$, die den optischen Fluss des Pixels durch alle Bilder hindurch repräsentiert, wird durch die Funktion $\mathbf{u}(\mathbf{x}; i)$ beschrieben:

$$\mathbf{u}(\mathbf{x}; i) = \begin{pmatrix} u_1(\mathbf{x}; i) \\ u_2(\mathbf{x}; i) \end{pmatrix} : \Omega \times \{1, \dots, m\} \rightarrow \mathbb{R} \quad (1)$$

Weiter sei $i_0 \in \{1, \dots, m\}$ ein Referenzbild. Für dieses gilt $\mathbf{u}(\mathbf{x}; i_0) = \mathbf{0}$. Der Grundgedanke des verwendeten Algorithmus besteht darin, die zulässigen Bewegungstrajektorien auf einen Unterraum aller möglichen Trajektorien zu beschränken. Für ein festes Pixel hat eine beliebige Trajektorie $\mathbf{u}(\mathbf{x}; i)$ $2m$ Dimensionen. Diese wird nun mittels $R \leq 2m$ vorgegebener Basisfunktionen $\mathbf{q}_k(i)$ mit $k \in \{1, \dots, R\}$ auf folgende Art approximiert:

$$\mathbf{u}(\mathbf{x}; i) = \sum_{k=1}^R \mathbf{q}_k(i) L_k(\mathbf{x}) + \varepsilon(\mathbf{x}; i) \quad (2)$$

Hierbei sind $L_k(\mathbf{x})$ die gesuchten Gewichtungskoeffizienten für die Basisfunktionen und $\varepsilon(\mathbf{x}; i)$ der Modellierungsfehler.

Der Algorithmus berechnet nun simultan den optischen Fluss $\mathbf{u}(\mathbf{x}; i)$ und die Koeffizienten $L_i(\mathbf{x})$ durch Minimierung eines Energiefunktional-s bestehend aus drei Termen mit Gewichtungsfaktoren α und β :

$$E(\mathbf{u}(\mathbf{x}; i), \mathbf{L}(\mathbf{x})) = \alpha E_{data} + \beta E_{link} + E_{reg} \quad (3)$$

Der Datenterm E_{data} bewirkt die Konsistenz der Intensitäten zwischen Referenzbild und allen anderen Bildern, wobei diese mittels des dichten optischen Flusses zunächst verzerrt werden. Durch E_{data} wird die

¹ Der Matlab-Quellcode ist unter <https://bitbucket.org/troussos/mfsf/downloads/> erhältlich (aufgerufen am 27.09.18)

Intensitätsdifferenz $|I(\mathbf{x} + \mathbf{u}(\mathbf{x}; i), i) - I(\mathbf{x}, i_0)|$ minimiert. Der Term E_{link} bewirkt eine Glattheit des optischen Flusses in zeitlicher Dimension indem die Abweichungen der Trajektorien $\mathbf{u}(\mathbf{x}; i)$ zum Unterraum, der durch die Basisfunktionen $\mathbf{q}_i(n)$ aufgespannt wird, minimiert wird. Dies entspricht einer Minimierung des Modellierungsfehlers $|\varepsilon(\mathbf{x}; i)|$ aus Gleichung (2). Der Term E_{reg} schließlich bewirkt eine Glattheit in örtlicher Dimension durch Minimierung von $|\nabla L(\mathbf{x})|$. Dabei beschreibt $\nabla L(\mathbf{x})$ die örtliche Änderung der Koeffizienten bzgl. der Basistrajektorien.

Für die Optimierung von $\mathbf{u}(\mathbf{x}; i)$ und $L(\mathbf{x})$ kommen hierbei verschiedene Techniken der Variationsrechnung und konvexen Optimierung zur Berechnung des optischen Flusses zum Einsatz [3, 4]. Eine grobe Abschätzung des optischen Flusses wird zunächst mit den Basisfunktionen der diskreten Kosinus-Transformation durchgeführt. Anschließend wird eine Hauptkomponentenanalyse der Trajektorien berechnet und der Algorithmus wird nochmals mit den Hauptkomponenten als neue Basis ausgeführt. Daraus resultiert eine genauere Abschätzung des optischen Flusses.

2.2 Schätzung der 3D-Struktur und Kamerabewegung

In diesem Abschnitt wird aufgezeigt, wie die vorher berechneten Pixelkorrespondenzen zur simultanen Schätzung der Kamerabewegung und der Form des beobachteten Objekts genutzt werden können. Bei gewöhnlichen Kameras mit perspektivischer Abbildung muss hierzu ein großes nichtlineares Gleichungssystem gelöst werden [1]. Bei Aufnahmen mittels einer telezentrischen Optik kann das Verfahren aus [5] angewendet werden, das mit Methoden aus der (numerischen) linearen Algebra auskommt.

Seien $\mathbf{s}_j \in \mathbb{R}^3$ die 3D-Koordinaten im Weltkoordinatensystem von n beobachteten Objektpunkten mit $j \in \{1, \dots, n\}$. Die Struktur des beobachteten Objektes wird als 3D-Punktwolke anhand der Matrix $\mathbf{S}^W \in \mathbb{R}^{3 \times n}$ mit $\mathbf{S}^W := (\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n)$ beschrieben.

Die 3D-Punktwolke $\mathbf{S}^W$ lässt sich mittels einer Rotationsmatrix $\mathbf{R}_i$ und einem Translationsvektor $\mathbf{t}_i$ in das Kamerakoordinatensystem der Kamera i überführen:

$$\mathbf{S}_i^C := \mathbf{R}_i \cdot \mathbf{S}^W + \mathbf{t}_i$$

Nun wird angenommen, dass die intrinsischen Parameter der Kamera bekannt sind. Bei einer Orthogonalprojektion lässt sich die Abbildung der 3D-Punktwolke $\mathbf{S}_i^C$ im Kamerakoordinatensystem der Kamera i auf 2D-Pixelkoordinaten $\mathbf{x}_i \in \mathbb{R}^{2 \times n}$ über die Matrix $\mathbf{A} \in \mathbb{R}^{3 \times 2}$ und dem Offset $\mathbf{o}$ auf folgende Weise berechnen:

$$\mathbf{x}_i = \mathbf{A} \cdot \mathbf{S}_i^C + \mathbf{o} = \mathbf{A} \cdot \mathbf{R}_i \cdot \mathbf{S}^W + \mathbf{A} \cdot \mathbf{t}_i + \mathbf{o} \quad (4)$$

Die Matrix $\mathbf{A}$ hat bei einer telezentrischen Optik folgende Form:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \end{pmatrix} \quad (5)$$

Wird nun der Mittelwert der Pixelkoordinaten über alle Objektpunkte j abgezogen, erhält man:

$$\underbrace{\mathbf{x}_i - \bar{\mathbf{x}}_i}_{=: \mathbf{x}_i^*} = \mathbf{A} \cdot \mathbf{R}_i \cdot \underbrace{(\mathbf{S}^W - \bar{\mathbf{S}}^W)}_{=: \mathbf{S}^*} \quad (6)$$

Die Abbildung der Punktwolke $\mathbf{S}^W$ auf die Pixelkoordinaten $\mathbf{X}^* \in \mathbb{R}^{m \times n}$ aller Kameras lässt sich nun kompakt darstellen als:

$$\mathbf{X}^* := \begin{pmatrix} \mathbf{x}_1^* \\ \mathbf{x}_2^* \\ \vdots \\ \mathbf{x}_m^* \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{A} \cdot \mathbf{R}_1 \\ \mathbf{A} \cdot \mathbf{R}_2 \\ \vdots \\ \mathbf{A} \cdot \mathbf{R}_m \end{pmatrix}}_{=: \mathbf{M}^*} \cdot \mathbf{S}^* \quad (7)$$

Hierbei sind $\mathbf{X}^*$ und $\mathbf{A}$ bekannt und $\mathbf{R}_i$ mit $i \in \{1, \dots, m\}$ und $\mathbf{S}^*$ die gesuchten Rotationen der Kameras bzw. die gesuchte 3D-Punktwolke. Die Singulärwertzerlegung von $\mathbf{X}^*$ ergibt:

$$\mathbf{X}^* = \mathbf{U} \cdot \mathbf{W} \cdot \mathbf{V}^\top \quad (8)$$

mit den Matrizen $\mathbf{U} \in \mathbb{R}^{m \times m}$, $\mathbf{W} \in \mathbb{R}^{m \times n}$ und $\mathbf{V}^\top \in \mathbb{R}^{n \times n}$. $\mathbf{W} = (w_{kl})$ ist eine Rechtecksmatrix mit $w_{kl} = 0$ für $k \neq l$ und $w_{kk} \geq 0$. Falls $\text{rang}(\mathbf{S}^*) = \text{rang}(\mathbf{M}^*) = 3$ so gilt nach den Rangungleichungen von Sylvester auch $\text{rang}(\mathbf{X}^*) = 3$. In diesem Fall kann Gleichung (8) auch kompakter dargestellt werden, da $w_{kk} = 0$ für $k > 3$:

$$\mathbf{X}^* = \mathbf{U}_3 \cdot \mathbf{W}_3 \cdot \mathbf{V}_3^\top \quad (9)$$

Dabei ist $U_3 \in \mathbb{R}^{m \times 3}$ eine Matrix, die die ersten drei Spaltenvektoren von U enthält, $V_3^T \in \mathbb{R}^{3 \times n}$ eine Matrix, die die ersten drei Zeilenvektoren von V_3^T enthält und $W_3 \in \mathbb{R}^{3 \times 3}$ die Diagonalmatrix der ersten drei Spalten- und Zeilenvektoren von W . Weiter seien M und S zwei Matrizen mit:

$$\begin{aligned} M &:= U_3 \cdot \sqrt{W_3} \\ S &:= \sqrt{W_3} \cdot V_3^T \end{aligned} \quad (10)$$

Somit gilt: $X^* = M \cdot S$. Die berechneten Matrizen M und S sind nicht eindeutig. Sei $C \in \mathbb{R}^{3 \times 3}$ eine beliebige reguläre Matrix, so wäre auch $X^* = M' \cdot S'$ mit $M' = M \cdot C$ und $S' = C^{-1} \cdot S$ eine mögliche Zerlegung von X^* . Es ist also eine Matrix C gesucht, so dass gilt:

$$\begin{aligned} M^* &= M \cdot C \\ S^* &= C^{-1} \cdot S \end{aligned} \quad (11)$$

O.B.d.A. wird nun angenommen, dass A folgende Form hat:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (12)$$

Ist dies nicht der Fall, so kann nach Gleichung (5) eine Matrix B gefunden werden, damit $B \cdot A$ die gewünschte Form erhält. Hierfür werden die Matrizen auf der linken und rechten Seite von Gleichung (7) blockweise mit B multipliziert.

Sei $M_i := (m_{kl})_{k \in \{2i-1, 2i\}, l \in \{1, \dots, 3\}}$ eine Untermatrix der bekannten Matrix $M = (m_{kl})$. Da R_i orthogonal ist, erhält man mit Gleichung (7), (11) und (12) folgende Beziehung zur Einheitsmatrix I :

$$\forall i : A \cdot R_i \cdot R_i^T \cdot A^T = M_i \cdot \underbrace{C \cdot C^T}_{=: L} \cdot M_i^T = I \quad (13)$$

Die Matrix L kann über ein lineares Gleichungssystem bestimmt werden, das sich aus Gleichung (13) ableiten lässt. Da L positiv definit ist, lässt sich C aus L über die Cholesky-Faktorisierung bestimmen. Nach Berechnung von C ergeben sich schließlich aus Gleichung (12) die gesuchten Matrizen M^* und S^* . Aus M^* lassen sich nur die ersten beiden Zeilen für alle R_i bestimmen. Mithilfe des Kreuzproduktes erhält man jedoch auch die dritte Zeile.

2.3 Probabilistische Oberflächenrekonstruktion mit B-Splines

In einem früheren Artikel [6] wurde ein probabilistisches Verfahren zur 3D-Rekonstruktion einer B-Spline-Oberfläche aus einer gemessenen 3D-Punktwolke vorgestellt, das auch hier genutzt wird. Bei B-Spline-Oberflächen wird ein zweidimensionaler Parameterraum $U \times V$, in diesem Fall $[0, 1]^2$, auf eine dreidimensionale Freiformfläche abgebildet. Die Abbildungsfunktion hat folgende Form:

$$S(u, v) = \sum_{k=0}^{n^u} \sum_{l=0}^{n^v} N_k^u(u) N_l^v(v) \mathbf{p}_{k,l} \quad (14)$$

Hierbei sind N_k^u und N_l^v univariate B-Spline-Basisfunktionen und $\mathbf{p}_{k,l}$ die Kontrollpunkte der B-Spline-Oberfläche. Es gibt $n^u + 1$ und $n^v + 1$ Kontrollpunkte in u - bzw. v -Richtung. Eine detaillierte Behandlung von B-Spline-Oberflächen und deren Eigenschaften findet sich in [7]. Sei $S(u, v)$ eine gegebene B-Spline-Oberfläche und $\mathbf{s}_i$ ein gemessener 3D-Punkt. Weiter sei (u_i, v_i) der Parameter im Parameterraum der B-Spline-Oberfläche, der den Abstand $|\mathbf{s}_i - S(u_i, v_i)|$ minimiert. Es kann eine lineare Messabbildung zwischen $\mathbf{s}_i$ und den Kontrollpunkten $\mathbf{p}_{k,l}$ aufgestellt werden, indem die Kontrollpunkte zu einem Vektor $\mathbf{p}$ zusammengefasst werden:

$$\mathbf{s}_i = \mathbf{H}_i(u_i, v_i) \cdot \underbrace{\begin{pmatrix} \mathbf{p}_{0,0} \\ \vdots \\ \mathbf{p}_{n^u, n^v} \end{pmatrix}}_{=: \mathbf{p}} + \mathbf{r}_i \quad (15)$$

Die Matrix $\mathbf{H}_i(u_i, v_i)$ für die Parameter u_i und v_i ergibt sich aus Gleichung (14) und $\mathbf{r}_i$ sei ein additiver Rauschterm. Ist eine ganze Punktwolke $\{\mathbf{s}_1, \dots, \mathbf{s}_n\}$ mit n Messungen gegeben und wird dafür eine Parametrierung berechnet, also jedem $\mathbf{s}_i$ ein Parameter (u_i, v_i) zugewiesen, so hat die Messabbildung die Form:

$$\underbrace{\begin{pmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_n \end{pmatrix}}_{=: \mathbf{S}} = \underbrace{\begin{pmatrix} \mathbf{H}_1(u_1, v_1) \\ \vdots \\ \mathbf{H}_n(u_n, v_n) \end{pmatrix}}_{=: \mathbf{H}} \cdot \mathbf{p} + \underbrace{\begin{pmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_n \end{pmatrix}}_{=: \mathbf{r}} \quad (16)$$

Bei dem probabilistischen Verfahren werden die Kontrollpunkte der B-Spline-Oberfläche durch eine a-priori Wahrscheinlichkeitsverteilung in Form einer multivariaten Normalverteilung beschrieben. Für diese Verteilung sei $\hat{p}_0$ der Erwartungswert und C_0^p die dazugehörigen Kovarianzmatrix. Es seien außerdem Messwerte durch einen Erwartungswert $\hat{S}$ und der Kovarianzmatrix C^S gegeben. Aus den Gleichungen des Kalman-Filters kann nun die a-posteriori Wahrscheinlichkeitsverteilung gegeben den Messwerten berechnet werden. Der Erwartungswert der a-posteriori Wahrscheinlichkeitsverteilung sei $\hat{p}_1$ und die Kovarianzmatrix C_1^p . Für die weitere Berechnung ist meist nur die Form der B-Spline-Oberfläche von Interesse, die durch die Kontrollpunkte $p_{k,l}$ beschrieben wird. Um die Kontrollpunkte zu erhalten, muss nun der Erwartungswert $\hat{p}_1$ entsprechend Gleichung (15) umgeformt werden.

3 Evaluation

Das beschriebene Verfahren zur 3D-Rekonstruktion von Oberflächen wurde anhand einer aufgenommenen Glühbirne evaluiert, die in Abbildung 1(a) dargestellt ist. Auf der Oberfläche der Glühbirne wurden kleine retroreflektierende Marker aufgebracht (siehe Abbildung 1(b)). Die Marker reflektieren nur von einer Seite. Somit sind nur ca. 50% der Marker als helle Flecken im Bild erkennbar. Mit dem beschriebenen Verfahren zur 3D-Rekonstruktion können nur die Oberflächenregionen erfasst werden, wo die Marker von der Probe nicht herunterfallen. Außerdem dürfen sich diese während der Messung nicht verschieben. Um mehrere Aufnahmen zu erhalten kann entweder die Probe oder der Sensor bewegt werden. Die Glühbirne wurde mit einem Goniometer in 0.3° -Schritten von -8° bis 8° senkrecht zur optischen Achse gedreht.

Mit einer telezentrischen Optik und einer coaxialen telezentrischen Beleuchtung wurde für jede Winkellage des Goniometers eine Aufnahme durchgeführt. Diese wurden dann mit einem Schwellwertfilter binarisiert. Der Algorithmus aus Abschnitt 2.1 wurde mit den Parametern $\alpha = \beta = 15$ auf den binarisierten Bildern ausgeführt, um den optischen Fluss zu berechnen. In Abbildung 2(a)- 2(c) sind exemplarisch drei binarisierte Aufnahmen unter drei verschiedenen Winkellagen des Goniometers abgebildet. Mithilfe des errechneten dichten op-

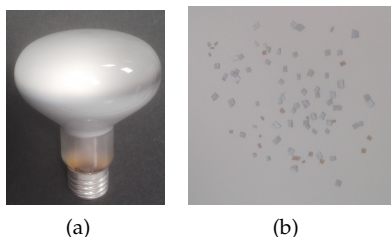


Abbildung 1: Auf den oberen Teil der Glühlampe (a) wurden Reflexionsmarker (b) aufgebracht und anschließend die Oberfläche mit dem beschriebenen Verfahren rekonstruiert.

tischen Flusses wurden die Abbildungen rücktransformiert. Die resultierenden Ergebnisbilder aus Abbildung 2(d)- 2(f) sind nach der Rücktransformation pixelgenau zueinander registriert.

Nach Berechnung des optischen Flusses wurde der Algorithmus aus Abschnitt 2.1 zur Schätzung der Punktwolke und der Kamerapositionen angewendet. Für die 3D-Rekonstruktion wurden nicht alle Pixel der Marker verwendet, sondern nur die Pixel in der Nähe der Kanten, wie in Abbildung 3(a) dargestellt. Der Grund liegt darin, dass der optische Fluss in den schwarzen und weißen Regionen des untexturierten Bereichs im Prinzip beliebig sein kann. Lediglich an den Kanten der Marker sind die berechneten Werte vertrauenswürdig. Das Ergebnis der simultanen Schätzung der Punktwolke und der Kamerapositionen ist für vier Winkel in Abbildung 3(b) dargestellt.

Anhand der Punktwolke lässt sich nun mit dem Verfahren aus Abschnitt 2.3 eine analytische Repräsentation der Oberfläche in Form einer B-Spline-Oberfläche gewinnen. Hierfür muss zunächst für $\hat{p}_0$ eine initiale Oberfläche definiert werden. Beispielsweise kann eine Ebene in die Daten der Punktwolke gefittet werden. Abbildung 4 zeigt das Ergebnis der 3D-Rekonstruktion. Während man in der Mitte der Heat-map in Abbildung 4(a) gut die Rotationssymmetrie der Oberfläche der Glühlampe erkennen kann, ist diese an den Ecken der Oberfläche, wo keine Marker aufgenommen wurden, nicht erkennbar. Aus der analytischen Repräsentation als B-Spline-Oberfläche lassen sich für jedes Kamera-Pixel sowohl die 3D-Koordinaten als auch der Gradient des entsprechenden Oberflächenpunkts berechnen. Hierfür muss für jedes

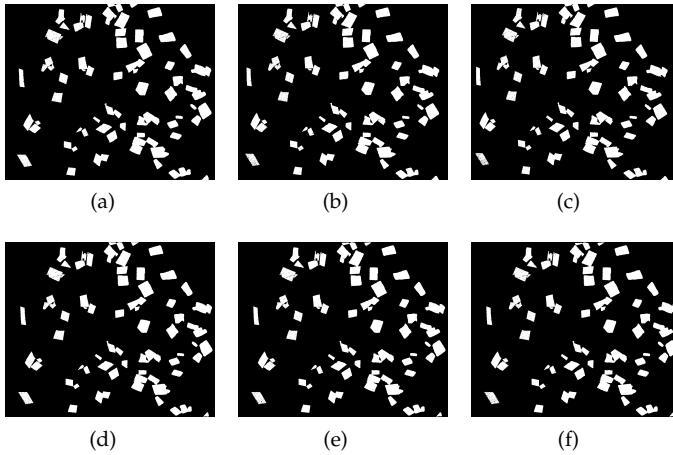


Abbildung 2: Die Abbildungen (a)-(c) wurden unter drei verschiedenen Winkellagen aufgenommen. Die Rücktransformation mithilfe des optischen Flusses ist in (d)-(f) dargestellt.

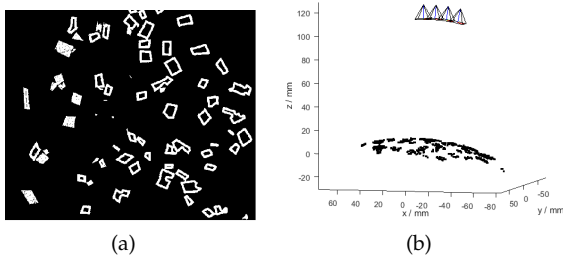


Abbildung 3: In (a) sind die für die 3D-Rekonstruktion verwendeten Pixel in weiß markiert. In (b) ist das Ergebnis der simultanen Schätzung der Punktwolke und Kamerabewegung dargestellt.

Pixel der koaxiale Lichtstrahl entlang der optischen Achse mit der B-Spline-Oberfläche geschnitten werden, um den dazugehörigen Parameter (u, v) im Parameterraum zu bestimmen. Diese nichtlineare Optimierung wurde mit einer performanten Implementierung aus [8] durchgeführt.

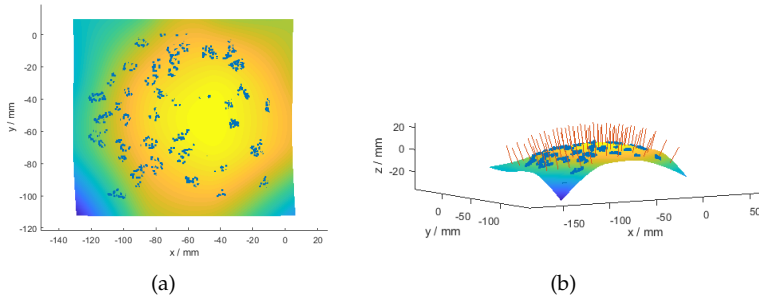


Abbildung 4: In (a) sind die 3D-Punktwolke und die rekonstruierte B-Spline-Oberfläche abgebildet. In (b) sind die B-Spline-Oberfläche und die Oberflächennormale dargestellt.

4 Zusammenfassung

Im diesem Artikel wurde ein Verfahren zur 3D-Rekonstruktion von glatten Oberflächen vorgestellt, das sich besonders für eine einfache 3D-Erfassung der Oberflächenform z.B. in Laborumgebungen eignet. In Anwendungen wo eine bewegliche telezentrische Kamera mit koaxialer Beleuchtung für die Vermessung von Prüfkörpern bereits vorhanden ist, kann die 3D-Form relativ zur Kamera ohne Verwendung von zusätzlicher Sensorik oder einer Kalibrierung der extrinsischen Kameraparameter mitgeschätzt werden. Eine Registrierung der 3D-Daten von unterschiedlichen Sensoren entfällt. Das Verfahren kann überall dort eingesetzt werden, wo die retroreflektierenden Marker aufgrund der Schwerkraft an die zu untersuchende Oberfläche haften bleiben. Außerdem ist das Verfahren bei beliebigen Oberflächen einsetzbar, unabhängig von der Absorption oder Reflektanz der Oberfläche.

Literatur

1. J. Beyerer, F. Puente León, and C. Frese, *Automatische Sichtprüfung: Grundlagen, Methoden und Praxis der Bildgewinnung und Bildauswertung*, 2nd ed. Berlin and Heidelberg: Springer Vieweg, 2016. [Online]. Available: <http://www.springer.com/de/book/9783662477854>
2. R. Garg, A. Roussos, and L. Agapito, "A variational approach to video registration with subspace constraints," *International journal of computer vision*, vol. 104, pp. 286–314, 2013.
3. C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Pattern recognition*, ser. Lecture Notes in Computer Science, F. A. Hamprecht, C. Schnörr, and B. Jähne, Eds. Berlin: Springer, 2007, vol. 4713, pp. 214–223.
4. T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Berlin and Heidelberg: Springer, 2004, vol. 3024, pp. 25–36.
5. C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International journal of computer vision*, vol. 9, no. 2, pp. 137–154, 1992.
6. C. Negara and M. Roschani, "Effiziente probabilistische b-spline-oberflächenrekonstruktion durch verwendung eines informationsfilters," in *Forum Bildverarbeitung 2012*, F. Puente León, Ed. Hannover and Karlsruhe: Technische Informationsbibliothek u. Universitätsbibliothek and KIT Scientific Publishing, 2012, pp. 131–142.
7. W. Piegl, Les A. ; Tiller, *The NURBS book*, 2nd ed., ser. Monographs in visual communications. Berlin: Springer, 1997, literaturverz. S. 629 - 638.
8. C. Negara, "Probabilistische Oberflächenrekonstruktion durch Planungsbaasierte Kameraaufnahmen," Diplomarbeit, Karlsruher Institut für Technologie (KIT), 2011.

Subtleties of extrinsic calibration of cameras with non-overlapping fields of view

Zaijuan Li¹ and Volker Willert¹

TU Darmstadt, Control Methods and Robotics Lab,
Landgraf-Georg Straße 4, 64283 Darmstadt

Abstract The calibration of the relative pose between rigidly connected cameras with non-overlapping fields of view (FOV) is a prerequisite for many applications. In this paper, we focus on the subtleties of experimental realization of such a calibration optimization method presented in [1]. We evaluate two strategies to adapt a given optimization process to find better local minima. The first strategy is the introduction of a quality measure for the image data used for calibration, which is based on the projection size of known planar calibration patterns on the image. We show, that introducing an additional weighting to the optimization objective chosen as a function of that quality measure improves calibration accuracy and increases robustness against noise. The second strategy to further improve accuracy is a careful data acquisition of pose pairs used for the calibration. We integrate the above strategies into different setups and demonstrate the improvement both in simulation and real-world experiment.

Keywords Quality measure, extrinsic calibration.

1 Introduction

Extrinsic camera calibration comprises the estimation of the relative pose between cameras with non-overlapping FOV. Especially, in the car industry, the calibration of cameras mounted on the car with vastly different FOV is ubiquitous [2], [3].

Different classifications of the existing calibration methods, together with detailed analyses and discussion of the methods could be found in [1], [4]. In principle, our proposed strategies are suitable for any

setup that builds an objective function based on the reprojection error of 3D-2D point correspondences constrained by 3D-3D closed-loop pose transformation $\mathbf{AX} = \mathbf{YB}$. Here, we restrict ourselves to the following two setups given in Fig. 1(a) and Fig.1(b).

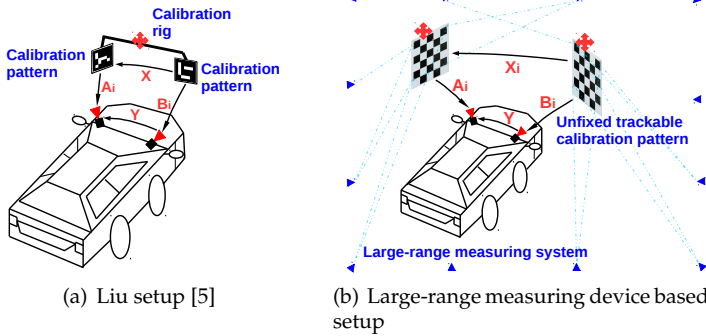


Figure 1: Illustration of two setups that could apply our proposed strategies. The red arrows appearing in the above figures indicate that the objects with red arrows overlaid have to be moved or placed into different positions during the calibration procedure.

The first setup presented in [5] uses a movable calibration rig that rigidly links two planar calibration patterns whose relative pose is unknown (see Fig.1(a)). By changing the pose of the calibration rig relative to the camera, the initial estimation of $\mathbf{X}$ and $\mathbf{Y}$ can be recovered by solving $\mathbf{AX} = \mathbf{YB}$. The initial values are then applied to minimize the objective function based on the reprojection errors from all measurements. Similar to the first setup, the second setup in [1] (see Fig.1(b)) introduces high accuracy tracking system ‘OptiTrack’, so the two pattern boards used for recovering the relative pose to camera pairs could be accurately localized within the tracking system after aligning their coordinate frames with that of the tracking targets attached to them. In this case, the 3D-3D closed-loop pose transformation is formulated as $\mathbf{AX}_i = \mathbf{YB}$, where $\mathbf{X}_i$ is obtained from the tracking system. The extrinsic could thus be solved and optimized using the reprojection error based objective.

Though Liu’s setup needs extra infrastructure and additional interaction, the calibration patterns could be detected reliably with sub-pixel

accuracy, which provides true scale information and could be further included in the optimization process. Meanwhile, the camera rig does not have to be moved during calibration, which is a big advantage, especially for mobile vehicles. However, the limited pose change of the calibration targets could result in instability [4]. Applying a large-range measuring system in the second setup is generally more accurate but the setup complexity and the costs are high.

2 Problem statement and optimization strategies

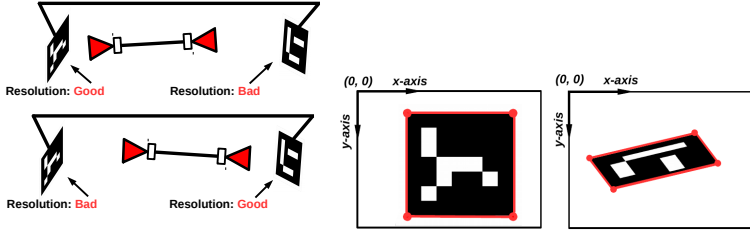
In [1], the optimization problem of Liu's setup is formulated as follows:

$$(\hat{\mathbf{R}}_X, \hat{\mathbf{t}}_X, \hat{\mathbf{R}}_Y, \hat{\mathbf{t}}_Y) = \arg \min_{\mathbf{R}_X, \mathbf{t}_X, \mathbf{R}_Y, \mathbf{t}_Y} \sum_{i=1}^n \left(\sum_{j=1}^m \|\epsilon_{ij}^A\|_2^2 + \sum_{l=1}^o \|\epsilon_{il}^B\|_2^2 \right), \quad (1)$$

where $\mathbf{R}_X, \mathbf{t}_X, \mathbf{R}_Y, \mathbf{t}_Y$ are the unknown rotational and translational matrix to be optimized, i.e. the relative pose between two calibration patterns $\mathbf{X}$ and the camera pair $\mathbf{Y}$. m and o stand for the amount of the fiducial features from corresponding patterns, and n is the number of the collected pose pairs. ϵ_{ij}^A and ϵ_{il}^B are the reprojection errors from different camera frames. This optimization problem is non-convex, so the iterative optimization can only guarantee to converge to a local minimum and a proper initialization is needed in order to reach a good estimation. In this paper, we use initial values for $\mathbf{X}$ and $\mathbf{Y}$ applying the method in [6] beforehand.

For each measurement pair $(\mathbf{A}_i, \mathbf{B}_i)$, both markers have to be in the FOV of the cameras such that all the coordinates of the fiducial feature projections can be extracted without outliers and with a certain accuracy. In practice, collecting a proper set of measurement pairs is challenging because of the rigid coupling of the patterns. With the assistance of the customized calibration device, a minor change in pose $\mathbf{A}_i$ or $\mathbf{B}_i$ would lead to an unpredictable change in the other, which indicates the hardness of capturing both calibration patterns with high resolution.

Fig.2(a) demonstrates the relationship between the calibration rig pose relative to the camera pair and the captured resolution quality. In Fig.2(b) an example is given to further explain this problem, which shows very *imbalanced* projection sizes of two different projec-



(a) Relationship between the pose and the resolution quality. (b) An example of two fiducial patterns captured from different cameras.

Figure 2: When the calibration pattern from one side of the rig is placed near to the camera, an image with high resolution will be captured, while the calibration pattern from the other side would be captured with comparatively lower resolution and vice versa. In this example, the left image gives a larger projection size of the pattern, hence higher resolution than the right image. When both images are corrupted by the same level noise, the left one is less sensitive to noise and produces a better pose estimation.

tions within one pose pair measurement. This discourages the objective function from including all the measurements and treating them equally.

Considering the unpleasant *imbalance* of the measurement quality which leads to the diversity of the projection size within one measurement pair, we introduce the additional weightings $\lambda_i^{A/B}$ to the objective (1) that are proportional to the projection size of the planar patterns, which leads to:

$$(\hat{\mathbf{R}}_X, \hat{\mathbf{t}}_X, \hat{\mathbf{R}}_Y, \hat{\mathbf{t}}_Y) = \arg \min_{\mathbf{R}_X, \mathbf{t}_X, \mathbf{R}_Y, \mathbf{t}_Y} \sum_{i=1}^n (\lambda_i^A \sum_{j=1}^m \|\epsilon_{ij}^A\|_2^2 + \lambda_i^B \sum_{l=1}^o \|\epsilon_{il}^B\|_2^2). \quad (2)$$

The weighting λ_i^A used for the reprojection error related with $\mathbf{A}_i$ is chosen to be the square root of the projection size $S(\mathbf{B}_i)$ related with reprojection $\mathbf{B}_i$ normalized by the full image size S_{max} and vice versa:

$$\lambda_i^A = \sqrt{S(\mathbf{B}_i)/S_{max}}, \quad (3)$$

$$\lambda_i^B = \sqrt{S(\mathbf{A}_i)/S_{max}}, \quad (4)$$

The reason for choosing such weighting lies in the replacement of $\mathbf{A}_i$ with $\mathbf{Y}\mathbf{B}_i\mathbf{X}^{-1}$. The reprojection error produced from $\mathbf{A}_i$ now depends

on its replacement $\mathbf{YB}_i\mathbf{X}^{-1}$ which has the measurement $\mathbf{B}_i$ inside, so the quality of $\mathbf{B}_i$ influences the reprojection error of $\mathbf{A}_i$: $\mathbf{B}_i$ with good-quality should have more influence on the optimization and this leads to a higher weight λ_i^A . The same happens with the replacement of $\mathbf{B}_i$.

Except for the measurement quality, another major practical issue to reach accurate calibration results is a proper set of measurement pairs covering the six degrees of freedom of the poses $\mathbf{X}$ and $\mathbf{Y}$, which indicates the spatial distribution of pose pairs also influences the calibration results. This requires that images of the calibration objects should be taken from as many different poses as possible. However, generating a pose pair set with good-quality and comparatively scattered spatial distribution which are essential for accurate calibration is subtly tricky. Therefore, instead of including all collected pose pairs, our second strategy picks out a subset with comparatively scattered pose pairs and good measurements. In this case, compromises have to be made between pose change variety and measurement quality.

3 validation on simulated dataset

3.1 Synthetic Dataset

As illustrated in Fig.1(a), a customized calibration device is introduced to assist the calibration procedure, except that all the true transforms are exactly known in the simulation.

To generate the synthetic dataset, an exhaustive searching program is first run based on known ground truth and camera intrinsic parameters to produce a ‘pose pair bank’ which consists of over 12,000 pose-pairs. All pose pairs in the bank meet the following requirements: Each pose pair in the bank is different from the rest both in translation and rotation so that the pose pairs in the bank span the whole possible measurement space; The projection size of the calibration object generated by the corresponding pose pair must exceed a certain threshold, which guarantees the minimum quality of the measurements. In this experiment, the threshold is set to 0.14 of the full image plane. The synthetic measurements are then generated based on ‘pose pair bank’: First, the true pose pairs are randomly extracted from the bank; The noise-free 2D coordinates obtained through the projection process are then corrupted with Gaussian noise; In the end, the noise-corrupted 2D coordinates are used

to recover the noisy pose pairs which will be taken as the measurements of $\mathbf{A}_i$ and $\mathbf{B}_i$.

3.2 Error metric

$\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ represent the estimated solutions which are calculated by applying different calibration methods. The ground truth of $\mathbf{X}$ and $\mathbf{Y}$ is known in the simulation environment, so the estimated $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ could be directly compared based on the following error metrics. Since the error metric calculation of $\mathbf{X}$ and $\mathbf{Y}$ is the same, only the $\mathbf{X}$ is taken as the example.

We apply the method of Wunsch et al. in [7] to define the rotation error. $\hat{\mathbf{q}}_X$ denotes the estimated quaternion of $\mathbf{X}$ and $\mathbf{q}_X$ the ground truth quaternion. The rotation error $\mathbf{e}_X^R$ is then defined as:

$$\mathbf{e}_X^R = \min\{\arccos(\mathbf{q}_X \cdot \hat{\mathbf{q}}_X), \pi - \arccos(\mathbf{q}_X \cdot \hat{\mathbf{q}}_X)\}, \quad (5)$$

in which ' $\cdot$ ' denotes the inner product of two quaternion vectors. Here the rotation error is represented by the angles returned by $\arccos$ and mapped to $[0, 90^\circ]$.

The estimated translation vector is described as $\hat{\mathbf{t}}_X$, and the ground truth is $\mathbf{t}_X$. The translation error is computed as follows,

$$\mathbf{e}_X^t = \|\mathbf{t}_X - \hat{\mathbf{t}}_X\|. \quad (6)$$

3.3 Simulation results

The results from the method proposed in [5] as well as after applying our strategies will be compared. In addition, the calibration results of the method in [6] are also presented since the above methods take its estimation of $\mathbf{X}$ and $\mathbf{Y}$ as initial values. Considering the improvement brought by the second strategy is not noticeable and would cover the other results, in what follows we demonstrate the benefits of strategies individually. For the validation of the proposed quality measure factor, those methods are labeled as Liu, 'Weighted Liu's method' (Wght-Liu), and Wang.

In the first setting, the measurement number changes from 10 to 60 with the fixed Gaussian noise of 1.0 pixel; In the second one, the added Gaussian noise varies from 0.2 to 1.2 pixels with a fixed number of 40

measurement pairs. The results shown below are taken the average of 100 iteration runs. For each iteration, the measurements are extracted from the ‘pose pair bank’ and processed applying each method. The calibration results are the average of overall calibration errors. The code for the calibration model as well as the optimization method is available online¹.

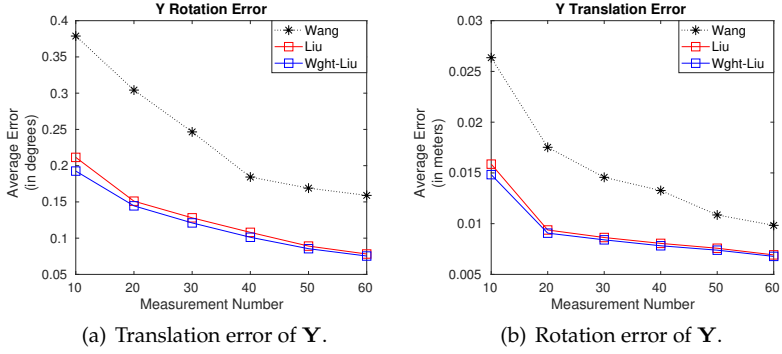


Figure 3: Estimation error with regard to increased number of measurements and different methods.

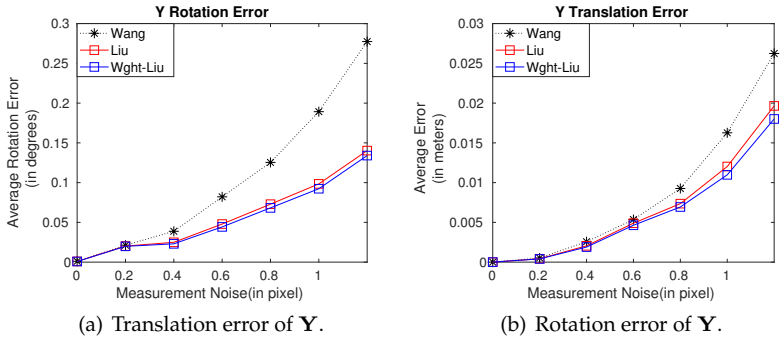


Figure 4: Estimation error with regard to increased image Gaussian noise and different methods.

Fig.3 and Fig.4 demonstrate the estimation error of $\mathbf{Y}$ under different

¹ <https://github.com/zaijuan/eye-to-eye-calibration.git>

settings. Since the comparison results of $\mathbf{X}$ are similar to $\mathbf{Y}$ in magnitude and pattern, it is unnecessary to present repetitive work.

The above experiment results validate the accuracy and robustness of our weighting factor strategy: Wght-Liu after applying the weighting factor gives the best results regardless of different settings; With the increase of noise level, the benefits from applying the weighting factor become more noticeable.

To demonstrate the importance of the spatial distribution of pose pairs to the calibration results, measurement sets with the following characteristics could be generated from ‘pose pair bank’: a). Spatially scattered pose pair set with larger projection size. b). Spatially clustered pose pair set with larger projection size. c). Scattered distributed pose pair set with smaller projection size. d). Clustered distributed pose pair set with smaller projection size. The ‘scattered’ and ‘clustered’ spatial distributions are comparative concepts. Because all generated pose pairs are extracted from the bank, each pose pair in the clustered set has at least the same minimum translational and rotational difference as the ones in the bank. Since each calibration object has to be placed in the FOV of its corresponding camera, the measurement space is reduced, so the pose pairs in the scattered set have comparatively larger while still limited differences in translation and rotation. Same for the measurement quality. The bad measurements are somewhat bad only when compared to the good ones, they are still guaranteed the minimum required quality.

We use those four extreme types of measurement sets to emphasize the improvement after choosing comparatively scattered pose pair distribution. For each configuration, the measurement number is set to be 40, and the noise level is 1.0. Fig.5 demonstrates the calibration results of different methods with different types of measurement sets.

Same as before, a final bundle adjustment with weighting factors is applied to refine calibration results. Since the relative pose $\mathbf{X}_i$ between the calibration patterns for each measurement is known, the replacement of $\mathbf{A}_i$ and $\mathbf{B}_i$ now becomes:

$$\mathbf{A}_i = \mathbf{Y}\mathbf{B}_i\mathbf{X}_i^{-1}, \quad (7)$$

$$\mathbf{B}_i = \mathbf{Y}^{-1}\mathbf{A}_i\mathbf{X}_i. \quad (8)$$

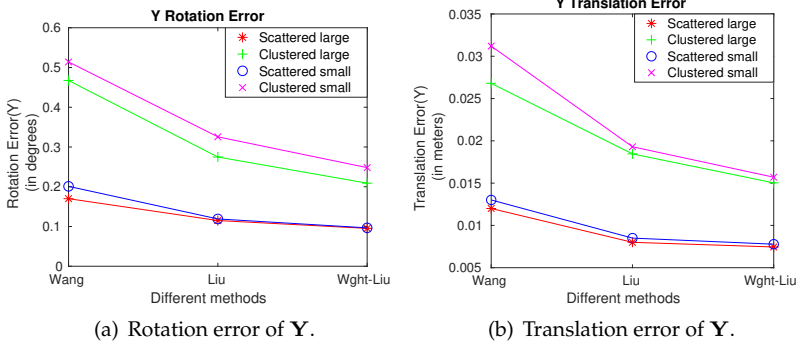


Figure 5: Estimation error with regard to different pose pair configurations and different methods.

For all methods, spatially scattered pose pairs with better measurement quality (larger projection size) generate the best accurate estimation, while the clustered spatially pose pairs with smaller projection size result in the worst estimation results. When the measurements have the same quality level, scattered pose pairs always produce better results than clustered ones, which implies the calibration methods are more demanding on the distribution of pose pairs than their measurement quality.

We give a short summary of our optimization strategies. The above calibration results bring some insights into the tradeoff between the spatial distribution of pose pairs and their generated measurement quality: The combination of scattered pose distribution and larger projection size produces the best calibration results. However, these two factors are mutually restricted: Scattered pose distribution implies the diversity of the projection size; While the demand for large projection size limits the spatial distribution of the pose pair. This further explains why our introduced weighting factor is important during the optimization process: First, it increases the pose change space by allowing larger varying range of measurement quality; Second, the increased measurement space helps to provide more accurate initial values from solving $\mathbf{AX} = \mathbf{YB}$, which further reduces the calibration error after non-linear optimization.

4 Real experimental results

4.1 Experiment setup

In the real experiment, both Liu's setup and the setup in [1] are implemented. Those experiments are carried out as follows:

In the first experiment, the calibration rig with two rigidly linked calibration patterns is placed to a variety of poses relative to the camera pairs and the corresponding pictures containing the calibration pattern are taken to recover the relative pose between them. In the end, a measurement set containing different pose pairs $\mathbf{A}_i$ and $\mathbf{B}_i$ is generated and used for different calibration methods.

With the assistance of the tracking system, the unknown relative pose $\mathbf{X}$ between the calibration objects could be accurately recovered. We name this configuration 'fixed trackable pattern'. In this case, the collected $\mathbf{A}_i$, $\mathbf{B}_i$, and the recovered $\mathbf{X}$ are used to run a final bundle adjustment (9) including weighting factors similar to (2) to refine calibration results:

$$(\hat{\mathbf{R}}_Y, \hat{\mathbf{t}}_Y) = \arg \min_{\mathbf{R}_Y, \mathbf{t}_Y} \sum_{i=1}^n (\lambda_i^A \sum_{j=1}^m \|\epsilon_{ij}^A\|_2^2 + \lambda_i^B \sum_{l=1}^o \|\epsilon_{il}^B\|_2^2). \quad (9)$$

The further improvement brought by the tracking system is that the two calibration patterns do not have to be rigidly linked. This extra flexibility facilitates the improvement of measurement quality since each calibration pattern could be placed into positions relative to corresponding cameras which generate the best possible estimates. So in the second experiment (setup), two pattern boards are independently placed to different poses relative to cameras. We describe this configuration as 'unfixed trackable pattern'. A set of $\mathbf{A}_i$ and $\mathbf{B}_i$ with better measurement quality together with the corresponding ground truth of $\mathbf{X}_i$ obtained from the tracking system is collected.

Same as before, a final bundle adjustment with weighting factors is applied to refine calibration results. Since the relative pose $\mathbf{X}_i$ between the calibration patterns for each measurement is known, the replacement of $\mathbf{A}_i$ and $\mathbf{B}_i$ now becomes:

$$\mathbf{A}_i = \mathbf{Y} \mathbf{B}_i \mathbf{X}_i^{-1}, \quad (10)$$

$$\mathbf{B}_i = \mathbf{Y}^{-1} \mathbf{A}_i \mathbf{X}_i. \quad (11)$$

4.2 Experimental results

We use the same error criteria to evaluate the calibration difference of different methods. The benchmark is set as the weighted estimation of configuration ‘unfixed trackable pattern’. We use the term ‘difference’ instead of ‘error’ to indicate that although the ground truth of $\mathbf{Y}$ is unknown, it could be estimated with the highest accuracy applying the ‘unfixed trackable pattern’ configuration.

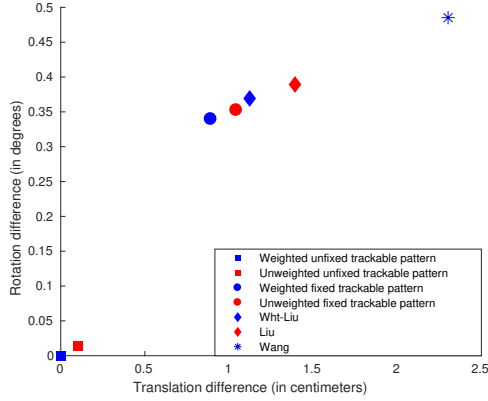


Figure 6: Calibration difference of $\mathbf{Y}$ with regard to different configurations and different methods.

Fig. 6 shows the calibration differences of different setups and different methods. Although in ‘unfixed trackable pattern’ configuration, the difference between weighted and unweighted estimation is minor since all patterns are captured with relatively high resolution, we could still verify that applying weighting factor generates less deviation compared to benchmark regardless of the configuration.

5 Conclusion and future work

In this paper, we discussed the subtleties of certain calibration methods and proposed two optimization strategies applicable to calibration setups that could minimize the reprojection error of 3D-2D point correspondences constrained by rigid 3D-3D closed-loop pose transforma-

tions $\mathbf{AX} = \mathbf{YB}$. First, we introduce an additional quality measure factor to the objective function, which helps enlarge the measurement space and improves the calibration accuracy. Hence, instability could be alleviated and the robustness could be safely guaranteed. Besides, by carefully choosing a measurement subset the possibility of getting trapped in a worse local minimum is reduced. In future work, we will focus on refining the weighting factor, which is now simply based on the projection area of the calibration pattern.

References

1. Z. Li and V. Willert, "Eye-to-eye calibration for cameras with disjoint fields of view(in press)," in *Intelligent Transportation Systems (ITSC)*. IEEE, 2018.
2. S. Dabral, S. Kamath, V. Appia, M. Mody, B. Zhang, and U. Batur, "Trends in camera based automotive driver assistance systems (adas)," in *Circuits and Systems (MWSCAS), 2014 IEEE 57th International Midwest Symposium on*. IEEE, 2014, pp. 1110–1115.
3. D. Cheda, D. Ponsa, and A. Lóz, "Camera egomotion estimation in the adas context," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 2010, pp. 1415–1420.
4. R. Xia, M. Hu, J. Zhao, S. Chen, Y. Chen, and S. Fu, "Global calibration of non-overlapping cameras: state of the art," *Optik-International Journal for Light and Electron Optics*, vol. 158, pp. 951–961, 2018.
5. Z. Liu, G. Zhang, Z. Wei, and J. Sun, "A global calibration method for multiple vision sensors based on multiple targets," *Measurement Science and Technology*, vol. 22, no. 12, p. 125102, 2011.
6. J. Wang, L. Wu, M. Q.-H. Meng, and H. Ren, "Towards simultaneous coordinate calibrations for cooperative multiple robots," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 410–415.
7. P. Wunsch, S. Winkler, and G. Hirzinger, "Real-time pose estimation of 3d objects from camera images using neural networks," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4. IEEE, 1997, pp. 3232–3237.

Extended noise equalisation for image compression considering linear and quadratically increasing noise sources

Daniel Kirchhöfer¹ and Bernd Jähne²

¹ PCO AG

Donaupark 11, 93309 Kelheim

² Heidelberg University,

Heidelberg Collaboratory for Image Processing (HCI)

Berliner Straße 43, 69120 Heidelberg

Abstract Today's camera systems used for machine vision and scientific applications have intra-scene dynamics up to 16 bit and therefore, A/D converters with up to 16 bit resolution per pixel. Unfortunately a linear amplification of electrons is not well suited, because then the noise of the image increases in first or second order with the signal. Based on a method, published in 2016 [1], this paper describes a more general nonlinear transformation which equalizes the combined effect of temporal noise and photoresponse non-uniformity (PRNU) and/or temporal noise in the illumination system of an image sensor.

Keywords Compression, image sensor, noise, EMVA1288.

Zusammenfassung Heutige Kamerasysteme, die für maschinelles Sehen oder wissenschaftliche Anwendungen eingesetzt werden, haben aufgrund ihrer hohen Intra-Szenen Dynamik eine digitale Auflösung von bis zu 16 Bit pro Pixel. Leider eignet sich eine lineare Verstärkung der Elektronen nicht besonders, da hier das Rauschen mit dem Signal ansteigt. Basierend auf einer in 2016 veröffentlichten Methode [1] beschreibt dieser Artikel eine nicht-lineare Transformation, die den kombinierten Effekt von zeitlichem Rauschen, PRNU und/oder zeitlichem Rauschen im Beleuchtungssystem eines Bildsensors equilibriert.

Schlagwörter Kompression, Bildsensor, Rauschen, EMVA1288.

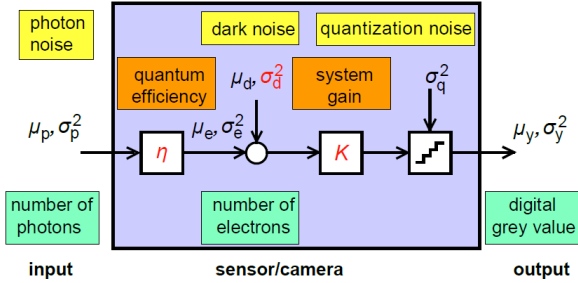


Figure 1: Linear Camera: Model of a single Pixel showing the processing steps and their noise sources [3].

1 Introduction

Today's (s)CMOS image sensors have due to their large intra-scene dynamic range a digital resolution up to 16 bit per pixel and it might get even more in the future. Additionally the high quality of these sensors and respectively the very good cameras that arise from those have very small internal noise sources, like for example only around $1e^-$ readout noise covering a dynamic range of over 35000:1 [2]. Furthermore they follow quite exactly the model of a linear camera as described in the EMVA1288 standard [3] and thus the variance of the images increases linearly with the grey value. This is because of the photon noise that is dominating, having a poisson distribution. Figure 1 shows a processing chain of each pixel including all factors with their noise sources and how they affect the grey value at the end.

The quantization, which is shown at the end of this process, is done linearly in the 16 Bit range. Unfortunately, due to the light-dependent photon noise, this is not very well suited. Because most compression algorithms are much more efficient having a slow and constant noise, since only noise cannot be compressed in the classical way. Additionally also image processing is much easier with a low and constant noise.

According to these informations Jähne and Schwarzbauer developed a nonlinear transformation to equalize the temporal noise to a constant and grey value independent level. This enables a compression of 16 bit images to 8 bit without losing information [1].

However this method only considers temporal variances that influence each pixel. Since a camera has also spatial varieties or noise sources, DSNU and PRNU for dark and bright images respectively [4], the noise in the compressed images is not really constant, because the variance of the PRNU increases quadratically with the light signal. Furthermore the noise of the light sources isn't considered as well. For standard light sources like LED's this doesn't matter, because here the noise is negligible. But many scientific applications use laser light where the RMS noise is in the range of several percent. Here the noise does matter. This paper describes a method, based on the previous approach, which also considers the described spatial noise sources and thus leads to a more constant noise and accurate compression. Furthermore also the variation of the light source can easily be integrated.

2 Noise model

According to the model of a linear camera as described above, a general noise model can be created. This mainly includes photon noise (σ_p), dark noise (σ_d) and quantisation noise (σ_q) [4-7]. Since in noise propagations not standard deviations but variances are added, the variance of the signal (σ_y^2) results in the following equation:

$$\sigma_y^2 = \sigma_p^2 + \sigma_d^2 + \sigma_q^2 \quad (1)$$

While dark noise and quantization noise are constant values, for example measured according to the EMVA1288 standard [3], the photon noise is poisson distributed and thus has a variance equal to the incoming photons. Since the quantum efficiency is a probabilistic function, it doesn't affect the photon noise. Thus $\sigma_p^2 = \mu_e$, where μ_e represents the generated electrons. All these electrons are amplified with a certain factor, called the system gain K. This factor can be estimated using the so called photon transfer method [3, 4] (for an example see figure 2). Since amplification affects noise in the same way as the signal, the above equation can be written as ($\mu_y = K \cdot \mu_e$)

$$\sigma_y^2 = K \cdot \mu_y + K^2 \cdot \sigma_d^2 + \sigma_q^2 \quad (2)$$

Equation (2) is used in [1] to compute the non-linear transformation for the noise equalization. However, if we look deeper into the proper-

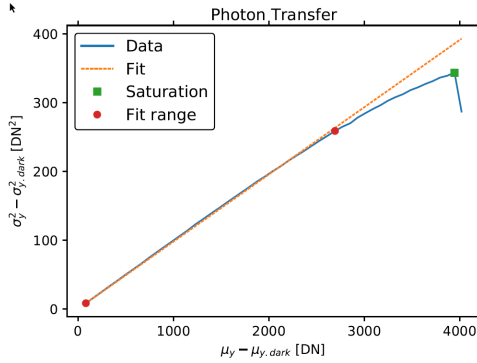


Figure 2: Example of a photon transfer curve [3].

ties of a linear camera, there are not only the described, temporal noise sources, but also spatial ones that influence the image [3,4]. These are called DSNU and PRNU and can be estimated with a normal distribution.

Looking at the field of applications of scientific cameras, especially in the area of microscopy [8–11], lasers are frequently used as light sources. Compared to standard LED's, the RMS noise here is already in the percentage range and thus also has to be considered as additional noise source.

This variation is a temporal noise source, but since, like for the PRNU, its variance increases quadratically with the light, it can just be added to the PRNU term. With $\sigma_{\text{PRNU}} = \text{PRNU} \cdot \mu_e$, $\sigma_{\text{light}} = \text{RMS}_{\text{light}} \cdot \mu_e$ and $\sigma_{\text{DSNU}} = \text{DSNU}$, the following noise component should be added.

$$\sigma_s^2 = (\text{PRNU}^2 + \text{RMS}_{\text{light}}^2) \cdot \mu_y^2 + K^2 \cdot \text{DSNU}^2 \quad (3)$$

Adding up (2) and (3) and using abbreviation terms, the resulting equation will be

$$\sigma_y^2 = s^2 \cdot \mu_y^2 + K \cdot \mu_y + \sigma_0^2 \quad (4)$$

where

$$s = \sqrt{\text{PRNU}^2 + \text{RMS}_{\text{light}}^2}$$

$$\sigma_0 = \sqrt{K^2 \cdot (\text{DSNU}^2 + \sigma_d^2) + \sigma_q^2}$$

3 Noise equalisation

Due to the laws of error propagation [4, 12], for any function $h(\mu_y)$ the noise can be expressed in the following way:

$$\sigma_h^2 = \left(\frac{dh}{d\mu_y}\right)^2 \cdot \sigma_{\mu_y}^2 (\mu_y) \quad (5)$$

Now with assuming σ_h to be constant, the required function $h(g)$ can be computed by [1]

$$h(g) = \sigma_h \cdot \int_0^g \frac{d\mu'_y}{\sqrt{\sigma^2(\mu'_y)}} \quad (6)$$

Applying the above equation to the noise computed in (4), this results in the following integral that has to be solved:

$$h(g) = \sigma_h \cdot \int_0^g \frac{d\mu'_y}{\sqrt{s^2 \cdot \mu_y'^2 + K \cdot \mu'_y + \sigma_0^2}} \quad (7)$$

Fortunately an algebraic solution can be found. The resulting function $h(g)$ is shown below

$$h(g) = \frac{\sigma_h}{s} \cdot \ln(|2 \cdot s \cdot (\sqrt{s^2 \cdot \mu_y^2 + K \cdot \mu_y + \sigma_0^2} + s \cdot \mu_y) + K|) + C \quad (8)$$

The constant C has to be chosen in such a way that $h(g=0) = 0$. So the final equation of this non-linear transformation is the following (the norm can be ignored since the result inside the logarithm will always be positive)

$$h(g) = \frac{\sigma_h}{s} \cdot (\ln(2 \cdot s \cdot (\sqrt{s^2 \cdot \mu_y^2 + K \cdot \mu_y + \sigma_0^2} + s \cdot \mu_y) + K) - \ln(2 \cdot s \cdot \sigma_0 + K)) \quad (9)$$

All required parameters of the equation can be determined by performing EMVA1288 measurements with the camera, except the resulting constant noise. This value has to be chosen in the right way. Jähne and Schwarzbauer [1] used Monte Carlo simulations based on the theoretical results discussed in [13] to show that the values for σ_h have to be at least 0.5 or greater in order to not make any significant error.

In many cases the condition need not be the smallest possible noise, but it is sufficient to chose σ_h in such a way that the 16 bit signal can just be mapped to 8 bit values. According to this requirement equation (9) can be rewritten in order to get a noise value for a given maximum h value (h_m) and a maximum μ_y value (μ_m , normally $2^{16} - 1$ for 16 bit image sensors)

$$\sigma_h = \frac{s \cdot h_m}{\ln(2 \cdot s \cdot (\sqrt{\mu_m \cdot (s^2 \cdot \mu_m + K) + \sigma_0^2} + s \cdot \mu_m)) + K) - \ln(2 \cdot s \cdot \sigma_0 + K)} \quad (10)$$

4 Comparison with the previous method

The nonlinear transformation in [1] is a specialization of equation (9) for DSNU = 0 and $s = 0$. This can easily be proven by just setting DSNU to zero and calculating $\lim_{s \rightarrow 0} h(\mu_y)$.

For the comparison a simulated set of images has been created with the following example values.

- $\sigma_h = 0.73$ [DN]
- $K = 2.2$ [DN/ e^-]
- $\sigma_d = 1.82$ [e^-]
- DSNU = 1.36 [e^-]
- PRNU = 1.20 [%]

First of all the actual results of the two methods are compared, especially looking at the difference. The top graph of figure 3 displays both transfer functions. The bottom graph shows their difference. It can clearly be seen that the difference between the two curves increases with the signal. This is due to the increasing effect of the PRNU. Additionally the difference curve also shows a minimum at a certain grey value and again increasing differences for smaller values. The reason for this behaviour is due to the DSNU. Because of small grey values, both the PRNU and the photon noise do not influence the signal significantly and therefore the DSNU-noise has a stronger relative impact.

The graphs in figure 4 show the standard deviation over the mean value of the transformed images for the two methods. The advanced

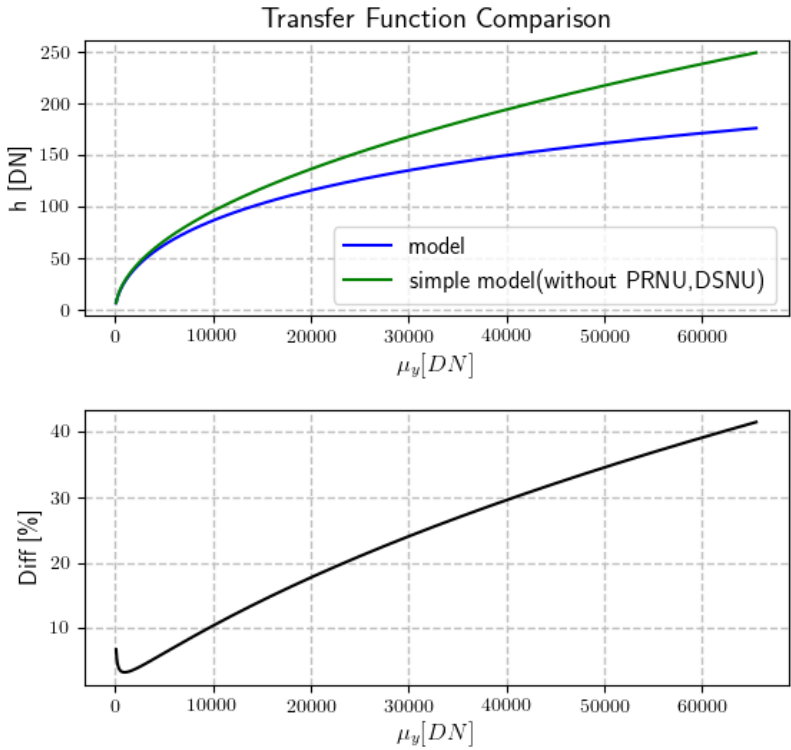


Figure 3: Top: Comparison of the nonlinear transformation functions Bottom: Difference of the two functions.

transformation considering also PRNU and DSNU values (blue), has a constant noise³, while in the other curve (green) the noise increases with larger grey values due to the quadratic influence of the PRNU on the signal. In addition also a significant difference in the dark image (first value) can be seen, this is due to the effect of the DSNU as described above. Furthermore the graphs have different maximum mean

³ The resulting standard deviation is a little bit higher than σ_h because of the additional quantisation noise that arises from the nonlinear transform

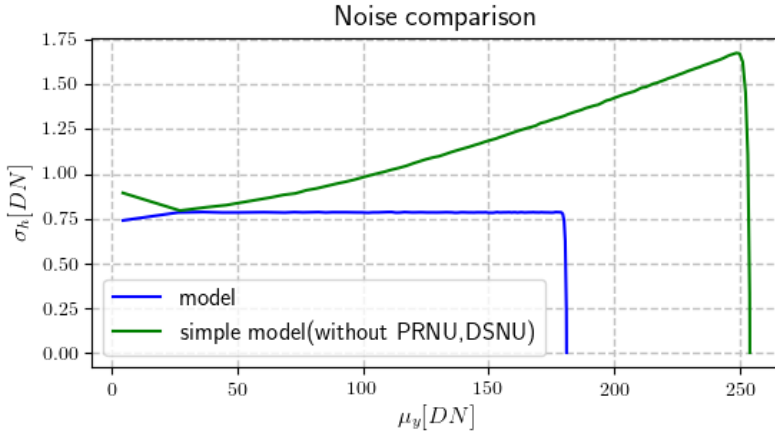


Figure 4: Comparison of the compressed noise level σ_h with and without considering DSNU and PRNU.

values, this is because the same σ_h is used for both transfer functions. If the requirement is only that the transformed image should fit in a 8 Bit range, a higher σ_h could be applied for the advanced transformation (see equation (10)).

As last comparison, different standard compression algorithms have been applied to transformed images. Therefore an example image stack from microscopy was used, where also the laser light was considered as additional noise source. For this specific example the RMS noise of the laser was 5 %. A σ_h of 0.74 was chosen so that for both methods the 8 bit value range is sufficient. Further parameters are $K = 2.26 \text{ DN}/e^-$, $\text{PRNU} = 0.42 \%$, $\text{DSNU} = 0.93 e^-$ and $\sigma_d = 1.05 e^-$.

The following table shows the file size after a nonlinear transform (and additional compression) for the method of [1], the transformation with DSNU and PRNU, as well as the transformation considering also the light source noise. As expected, the advanced method achieves better compression rates after transformation, especially if the light noise of the laser is considered as well.

Table 1: File sizes in MB for different transformation and compression methods using a set of microscopic images.

Method	No Compression	JPEG(lossless)	LZW	DEFLATE
Without PRNU, without light	869	498	498	405
With PRNU, without light	869	490	490	398
With PRNU, with light	869	332	332	269

5 Conclusions

It can be summarized, that for modern high dynamic range sCMOS cameras, with DSNU values smaller than 1 electron and PRNU values below 1 %, [1] is a good approximation of the more complex nonlinear transformation described here. At least as long as only the “raw” camera is considered.

Once looking also at the applications of such cameras, things might be different. As mentioned in chapter 2 many applications for scientific high end cameras use lasers for illumination. Here the additional noise of the light source lies in the percentage range and thus really influences the signal. Since the noise distribution of light sources follows the same rules as the one for the PRNU, here the advanced nonlinear transformation should be used in order to achieve good results.

References

1. B. Jähne and M. Schwarzbauer, “Noise equalisation and quasi loss-less image data compression—or how many bits needs an image sensor?” *tm-Technisches Messen*, vol. 83, no. 1, pp. 16–24, 2016.
2. PCO AG, “pco.edge 4.2 lt datasheet.” [Online]. Available: https://www.pco.de/fileadmin/user_upload/pco-product_sheets/pco.edge_42.lt.data_sheet.pdf
3. EMVA 1288 Working Group, “Emva standard 1288: Standard for characterization of image sensors and cameras: Release 3.1.” [Online]. Available: <http://www.emva.org/wp-content/uploads/EMVA1288-3.1a.pdf>

4. B. Jähne, *Digitale Bildverarbeitung*. Springer-Verlag, 2013.
5. F. Dierks, "Sensitivity and image quality of digital cameras," *Image_Quality_of_Digital_Cameras.pdf*, 2004.
6. J. Janesick, K. Klaasen, and T. Elliott, "Ccd charge collection efficiency and the photon transfer technique," in *Solid-State Imaging Arrays*, vol. 570. International Society for Optics and Photonics, 1985, pp. 7–20.
7. G. C. Holst, "Cmos/ccd sensors and camera systems," "CMOS/CCD sensors and camera systems/Gerald C. Holst, Terrence S. Lomheim. Published:, 2007.
8. C. Drobek, R. Mau, and H. Seitz, "Tomographic particle image velocimetry of a water-jet for low volume harvesting of fat tissue for regenerative medicine," *Current Directions in Biomedical Engineering*, vol. 1, no. 1, pp. 345–348, 2015.
9. J. Icha, M. Weber, J. C. Waters, and C. Norden, "Phototoxicity in live fluorescence microscopy, and how to avoid it," *Bioessays*, vol. 39, no. 8, p. 1700003, 2017.
10. M. B. Sikkel, S. Kumar, V. Maioli, C. Rowlands, F. Gordon, S. E. Harding, A. R. Lyon, K. T. MacLeod, and C. Dunsby, "High speed scmos-based oblique plane microscopy applied to the study of calcium dynamics in cardiac myocytes," *Journal of biophotonics*, vol. 9, no. 3, pp. 311–323, 2016.
11. M. J. Taormina, M. Jemielita, W. Z. Stephens, A. R. Burns, J. V. Troll, R. Parthasarathy, and K. Guillemin, "Investigating bacterial-animal symbioses with light sheet microscopy," *The Biological Bulletin*, vol. 223, no. 1, pp. 7–20, 2012.
12. A. Papoulis, "Probability, random variables, and stochastic processes," 1965.
13. B. Widrow and I. Kollar, "Roundoff error in digital computation, signal processing, control, and communications," *Cambridge University Press*, pp. 61–108, 2008.

A concept for a multi-spectral snap-shot camera based on single-chip RGB sensor

Robin Zirkler^{1,2}, Timo Eckhard², and Bernd Jödicke¹

¹ HTWG Konstanz,
Alfred-Wachtel-Str. 8, 78462 Konstanz

² Chromasens GmbH,
Max-Stromeier-Str. 116, 78467 Konstanz

Abstract Multi-spectral imaging with 12 spectral sensitive channels on a trichromatic sensor is proposed in this paper. The 12 channels allow for a high quality surface reflectance measurement of the acquired images. Here, we present a method for performant image registration of the 12 channels and a method for device independent color measurement. The spectral reconstruction algorithm is based on a linear least-square model.

Keywords Multi-spectral imaging, snap-shot, pseudo-inverse, image registration.

1 Introduction

The main goal of multi-spectral imaging in perspective of this work is to reconstruct the reflectance of a given object scene. A conventional RGB camera uses three spectral sensitive channels to digitalize the spatial content of an image scene with respect to the visible part of the spectral light. In case of estimating surface reflectance, the complexity of a visible scene needs more than three channels for a high quality reflectance estimation.

In the field of multi-spectral imaging there are several methods to acquire the spectral reflectance from a visible scene [1]. A snap-shot imaging approach collects the whole dataset (datacube) in one acquisition. *Chromasens GmbH* developed a multi-spectral camera based on a line-scan camera. In contrast to this method, there are multi-spectral cameras with different sensor architectures, as Monno et al [2] show.

In this paper, we propose a method for acquiring a 12 channel multi-spectral image with modified optics on a conventional three channel RGB camera. After describing the camera system and the used components (chap. 2) we will introduce the imaging process including image registration and spectral estimation (chap. 3). Furthermore we will verify the proposed methods (chap. 4) and conclude the work (chap. 5).

2 Multi-spectral camera

The multi-spectral camera is based on a camera with trichromatic area sensor from *Mikrotron GmbH*. The proposed system uses four lenses with distinct spectral color filters to create four separate images (sub-images) of the scene on a 25 megapixel CMOS sensor (*Mikrotron EoSens 25CXP*). A schematic view of the setup is given in figure 1. Every sub-image contains three spectral channels, each corresponding to a spectral sensitivity of the area sensor color filter (R, G, or B) plus the transmission of the color filter in front of the attached lenses. This configuration allows to acquire 12 district spectral channels in one shot. The transmittances of the attached color glass filters are given in figure 2.

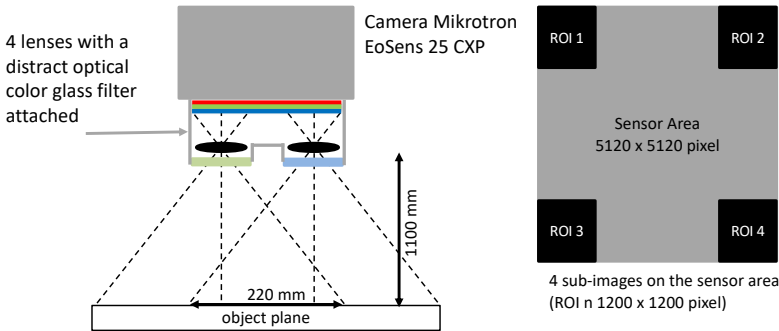


Figure 1: Schematic illustration of the working principle of the multi-spectral camera. On the left side is the the camera with the attached color glass filters and on the right side is the sensor area and the illustration of the occupied sub-images.

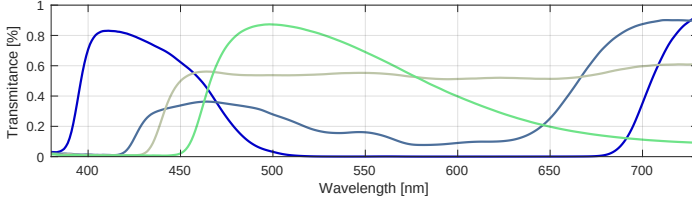


Figure 2: Transmittances of attached color glass filters.

3 Overview of imaging process workflow

An illuminated scene is first acquired by the multi-spectral camera. Then a conventional radiometric 2-point correction model [3] for shading and dark current correction is applied. Afterwards, a sub-pixel accurate registration of every image channel with respect to the first one is performed, which is followed by an estimation of the spectral reflectance. In figure 3 the imaging process is illustrated. During image acquisition we have used a fixed lighting setup, which employed Corona II from *Chromasens GmbH* [4]. The illumination angle is 45° and the observation angle is 0° ($45^\circ \times 0^\circ$).

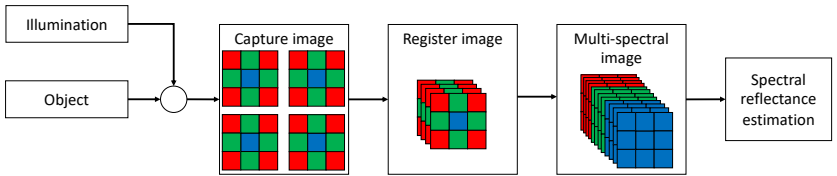


Figure 3: Schematic view of image processing workflow. After image acquisition a registration of the sub-images is performed. Afterwards spectral reflectances are estimated.

3.1 Image registration

The unregistered image is defined as an image cube ζ with the image data matrices $\{I_\lambda \in \mathbb{R}^{r \times c} | \lambda = 1, 2, \dots, 12\}$, where r , c and λ are

the rows, columns and the image channels. The image coordinates are $\Omega = \{(x, y) \in \mathbb{Z}^2 | 1 \leq x \leq c, 1 \leq y \leq r\}$. We register the images to a virtual reference cube ζ_0 with $\{I_{0,\lambda} \in \mathbb{R}^{r \times c} | \lambda = 1, 2, \dots, 12\}$ for which the image planes are spatially aligned and

$$I_{0,\lambda}(x, y) = I_\lambda(x + z_{u,\lambda}(x, y), y + z_{v,\lambda}(x, y)) , \quad (1)$$

respectively the misalignments in horizontal direction are described by the functions $z_{u,\lambda}(x, y)$ and in vertical direction by $z_{v,\lambda}(x, y)$ at any pixel location $(x, y) \in \Omega$ for image channel λ [5]. In this study the reference channel is sub-image one ($k = 1$). This sub-image includes three spectral sensitive channels, namely the R, G and B channel of the sensor. A set of n key-points P on a calibration target must be detected over the sub-image $k = 1, 2, \dots, 4$ to determine the misalignments. The location of the key-points is defined by $\{(x_{i,k}, y_{i,k}) \in \mathbb{R}^2 | x_{i,k}, y_{i,k} \in \Omega, i = 1, 2, \dots, n, k = 1, 2, \dots, 4\}$. The coordinates of the reference sub-image are $(x_{i,1}, y_{i,1})$. The horizontal and vertical misalignments can be calculated by

$$\begin{aligned} \mathbf{z}_{u,k} &= [x_{i,k} - x_{i,1}]_{n \times 1} , \\ \mathbf{z}_{v,k} &= [y_{i,k} - y_{i,1}]_{n \times 1} , \end{aligned} \quad (2)$$

for the key-points ($i = 1, 2, \dots, n$). These misalignments are calculated for each sub-image, so $\mathbf{z}_{u,k}$ and $\mathbf{z}_{v,k}$ are the horizontal and vertical displacement vectors for the sub-images ($k = 1, 2, \dots, 4$). The corresponding model can be described by:

$$\begin{aligned} \tilde{z}_{u,k}(x, y) &= \sum_{i=0}^d \sum_{j=0}^d a_{ij,k} x^i y^j , \\ \tilde{z}_{v,k}(x, y) &= \sum_{i=0}^d \sum_{j=0}^d b_{ij,k} x^i y^j , \end{aligned} \quad (3)$$

where $(x, y) \in \Omega$, d is the number of polynomial terms, and the respective polynomial coefficients are given by $a_{ij,k}$ and $b_{ij,k}$. Specifically for the image registration, we designed an adequate geometrical calibration target, to determine the displacement of each sub-image with respect to the reference, shown in figure 4. The target is composed by one ArUco-Marker and a regular grid of n_p dots, which are placed around the

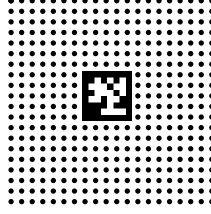


Figure 4: Illustration of the geometric calibration target. For a performant detection, an ArUco-Marker is centered upon a grid of dots.

Marker. The ArUco-Marker is detected by the OpenCV library, which has been implemented into MATLAB [6]. The ArUco-Marker allows for performant detection of the calibration target at arbitrary orientation in the images scene. The marker is defined as key-point P_m with the coordinates $(x_{i,k}, y_{i,k})$. According to eq. (2) we calculate $\mathbf{z}_{u,k}$ and $\mathbf{z}_{v,k}$ as the displacement of the sub-images k . Afterwards we calculate a linear transformation, where the sub-images ($k = 2, 3, 4$) are in line with the reference sub-image ($k = 1$). To get the image $I_{a,k}$ we define

$$I_{a,k}(x, y) = I_k(x + \mathbf{z}_{u,k}, y + \mathbf{z}_{v,k}) . \quad (4)$$

The points around the ArUco-Marker are the key-points P_p . These key-points were used for the displacement correction of the lenses. For this purpose we solve eq (3). The coefficients $a_{ij,k}$ and $b_{ij,k}$ are calculated with the MATLAB function *fitgeotrans* (f_g) which is based on ref. [7]. The function generates the following coefficients with respect to the reference key-points in sub-image one and the moving key-points in sub-image ($k = 2, 3, 4$):

$$a_{ij,k}, b_{ij,k} = f_g(x_{i,k}, y_{i,k}) . \quad (5)$$

The image transformation process can be described by

$$I_{k,0}(x, y) = I_k(x + \tilde{z}_{u,k}(x, y) + \mathbf{z}_{u,k}, y + \tilde{z}_{v,k}(x, y) + \mathbf{z}_{v,k}) , \quad (6)$$

where the four sub-images are registered. For a 12 channel image we separate the three spectral channels of the area sensor color filter (R,G and B) by demosaicing each registered sub-images.

3.2 Spectral reflectance estimation

The spectral reflectance estimation problem is to reconstruct high-dimensional spectral reflectance data from a low-dimensional camera response data. There exist two established strategies, physical based and empirical based estimation. We used an empirical based method to solve the spectral reflectance estimation problem. Therefore, we calibrated the camera by training the system by the use of a color calibration chart. For this training *Chromasens GmbH* has developed a special chart. The calibration chart is illustrated in figure 5 (a) and the color coordinates of the chart in CIE-L*a*b* coordinates is shown in (b). This chart includes $n_c \in \{1, 2, \dots, 660\}$ different color patches. Every patch is measured with a spectrodensitometer of type Koncia Minolta FD7 [8]. We define the spectrum of the spectrodensitometer as the reference spec-

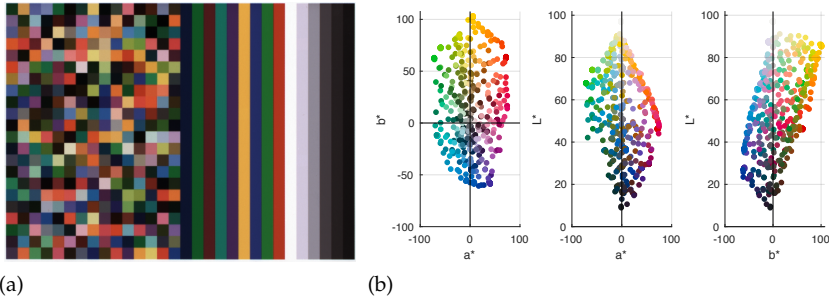


Figure 5: (a) shows the calibration chart and (b) illustrates the CIE-L*a*b* color coordinates of the calibration chart.

trum in the matrix $R_{\text{ref}} \in \mathbb{R}^{m \times n_c}$ in a wavelength range 380 – 730 nm and in steps of 10 nm, such that $m \in \{1, 2, \dots, 36\}$. The camera-response matrix $K \in \mathbb{R}^{\lambda \times n_c}$ includes the average intensity of the color patches n_c , where λ denotes the image channel. We assume that there is a linear relation between camera-responses K and reflectances R_{ref} , which we model as

$$R_{\text{ref}} = K \times A, \quad (7)$$

where $A \in \mathbb{R}^{m \times \lambda}$. Matrix A can be determined from training data using linear least-square regression.

$$A = R_{\text{ref}} \times K'^+, \quad (8)$$

where K^+ is the pseudoinverse according to *Moore Penrose*. The pseudoinverse is defined as

$$K^+ = (K' \times K)^{-1} \times K'. \quad (9)$$

4 Experiments and results

4.1 Image registration

We evaluated the image-registration for three cases:

- **unregistered** (displacement of the sub-images without registration)
- **poly-2** ($d = 2$ in formula 3; corresponding to 2nd degree polynomial model for image registration)
- **poly-4** ($d = 4$ in formula 3; corresponding to 4th degree polynomial model for image registration)

In the first experiment (**unregistered**), we detect the ArUco-Marker and calculate the image cube $I_{a,k}$ according to eq. (4). Afterwards we detect the key-points P_p and calculate the misalignment of the sub-images according to the eq. (2). Figure 6 shows the horizontal and vertical displacement $z_{u,k}$ and $z_{v,k}$ of every key-point n_p for every sub-image k across the sensor plane in pixel (px). This figure indicates two significant features of the lenses. The shape of point error in the plot corresponds roughly to a linear plane. The orientation of the plane is different in every sub-image. The plane inclinations are attributed to the varying magnification of the lenses, which can be explained by inaccurate matching of the 4 lenses with respect to their distance from the sensor. It also can be seen, that the plane shapes of points are slightly curved, which is an indication of barrel distortion and is attributed to the lenses [3].

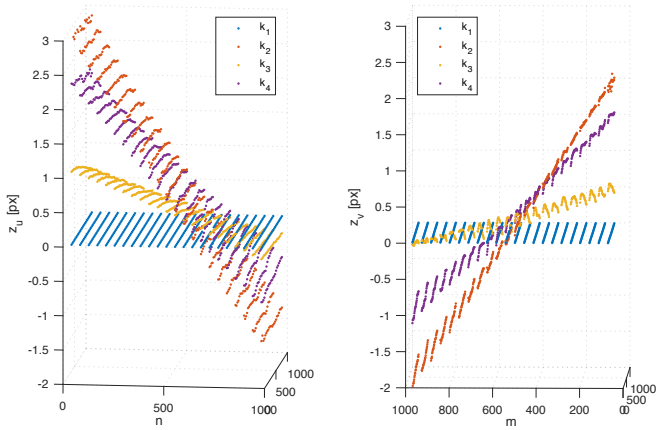


Figure 6: Displacement of image points for the unregistered channels with respect to sub-image 1.

In the **poly-4** and **poly-2** experiment we calculate the polynomial coefficients $a_{ij,k}$ and $b_{ij,k}$ according to eq. (5). With this model we register the images by eq. (6) and calculate the residual displacement according to eq. (2). Figure 7 illustrates the results of the experiment **poly-4**. It shows the displacement of the key-points in the registered image across the n columns and m rows in the sub-images k . The residual displacement for every single channel seems to oscillate along the corresponding image dimension. This effect is probably due to a limitation of the point-finder accuracy. The order of magnitude of the residual displacement is rather small ($\sim \pm 0.1$ pixel). We believe to be close to noise level of the point extraction used for generating the calibration data for the registration.

The statistics of the pixel displacement for the experiments are shown in table 1. The maximum displacement of the unregistered images is found in sub-image two with a value of 3.03 pixel. With a polynomial registration model of 2nd degree, we achieved a maximum displacement of 0.10 pixel and a mean displacement of 0.03 pixel. The registration model of 4th degree achieved a mean value of 0.02 pixel and a maximum value of 0.08 pixel.

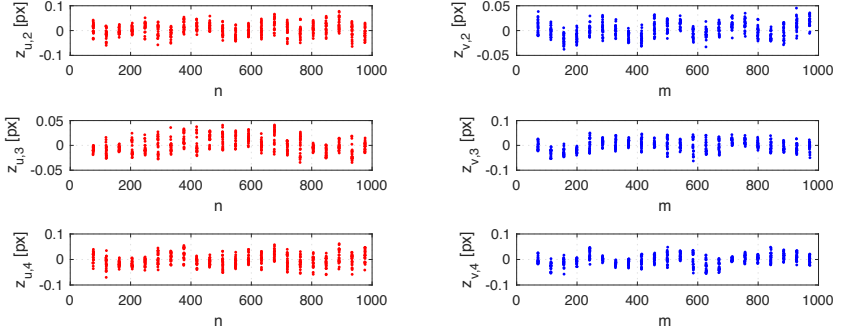


Figure 7: Residual displacement of image points from the calibration target for the registered image **poly-4**.

Table 1: Displacement of the experiment **unregistered** and the residual displacement of the experiment **poly-2** and **poly-4**. k represents the sub-images and x is the error in column and y in row direction.

	unregistered				poly-4				poly-2			
	mean		max		mean		max		mean		max	
k	x	y	x	y	x	y	x	y	x	y	x	y
2	1.24	1.21	3.03	2.26	0.02	0.02	0.08	0.05	0.03	0.02	0.08	0.05
3	0.34	0.26	0.94	0.79	0.01	0.02	0.04	0.06	0.02	0.03	0.05	0.10
4	0.84	0.81	2.13	1.71	0.02	0.02	0.07	0.06	0.03	0.03	0.10	0.08

4.2 Spectral reflectance estimation

We evaluated spectral reflectance estimation for the following three potential camera operation modes:

- **Single-Snapshot** (one acquisition of the calibration chart)
- **Multi-Snapshot** (100 times acquisition of the calibration chart and then averaging over images, reduction of noise)
- **Multi-frame** (every image channel (R,G,B) acquired with an optimized exposure time, optimal use of dynamic range for every channel)

We computed the calibration matrix $A_{\text{cal}} \in \mathbb{R}^{m \times \lambda}$ for every experiment according to eq. (8) and estimate spectral reflectance $R_{\text{est}} \in \mathbb{R}^{m \times n_c}$:

$$R_{\text{est}} = A_{\text{cal}} \times K. \quad (10)$$

We used CIE-Lab as well as spectral Root-Mean-Square-Error (hereafter RMSE) as criteria to evaluate the measurement performance. For a quantitative evaluation of the color calibration we computed a self-consistency. The self-consistency test calculates the CIE2000 ΔE_{00} [9] of the reference spectrum r and the estimated spectrum $\tilde{r}$ as well as the RMSE for all patches of the color calibration chart. The RMSE is defined [5] as:

$$RMSE(r, \tilde{r}) = \sqrt{\frac{1}{m} \sum_{i=1}^m (r - \tilde{r})^2} \quad (11)$$

The statistics of the self-consistency experiment are illustrated in table 2. With the **Single-Snapshot** experiment, we achieved a mean RMSE of 0.011 and a ΔE_{00} of 0.75. The **Multi-Snapshot** experiment resulted in the lowest mean RMSE, namely 0.009 and in ΔE_{00} of 0.64. This is the result of noise reduction by averaging the 100 acquired images. The **Multi-Frame** setting, which uses the full dynamic range on a channel by channel basis, shows the lowest maximum-value ΔE_{00} when compared to the other experiments. The **Multi-Frame** approach reduces the influence of the sensor noise especially when using color patches that have a low reflectance.

Table 2: Spectral estimation results of the 3 experiments in ΔE_{00} and RMSE.

	ΔE_{00}			RMSE		
	mean	max	p95	mean	max	p95
Single-Snapshot	0.75	5.37	1.91	0.011	0.034	0.018
Multi-Snapshot	0.64	6.98	1.57	0.009	0.023	0.016
Multi-Frame	0.79	4.54	1.83	0.011	0.033	0.020

In figure 8 and figure 9, two sets of estimated spectra from the experiment **Multi-Frame** are plotted. The solid lines represent the estimated spectrum $\tilde{r}$ and the dotted lines represent the reference spectrum r . Figure 8 shows 10 spectra with the lowest ΔE_{00} , which also happen to be spectra with a high reflectance factor. The errors are small overshoots in spectra with a saddle point.

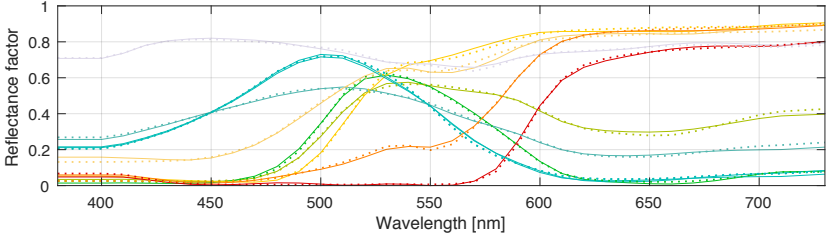
**Figure 8:** Estimate reflectances of the **Multi-Frame** experiment with low ΔE_{00} values. The solid lines represent the estimated reflectance and the dotted lines represent the reference reflectance.

Figure 9 shows five spectra with a high ΔE_{00} . It can be seen that the reflectance factors of these samples are low in comparison to spectra shown in figure 8. We believe that samples with low camera response values (i.e. dark patches) are more influenced by noise, resulting in lower spectral estimation performance. There is one spectrum with negative reflectance values which obviously is not feasible from a physical perspective.

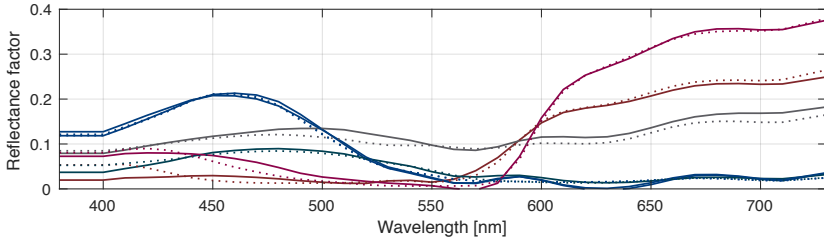


Figure 9: Estimate reflectances of the **Multi-Frame** experiment with high ΔE_{00} values. The solid lines represent the estimated reflectance and the dotted lines represent the reference reflectance.

5 Summary and conclusions

We propose a 12 channel multi-spectral camera system based on a RGB area sensor with four attached lenses and individual color filters. A color calibration is performed empirically, by means of a *Chromasens GmbH* proprietary color calibration chart. Once established, spectral reflectance estimation based on a linear least-square regression model can be performed. Device-independent color-difference measurements in CIE-L*a*b* color space are deduced from spectral reflectance measurements. We have used ΔE_{00} color difference as well as spectral RMSE as criteria to evaluate the measurement performance. On average, a ΔE_{00} error of 0.75 and a RMSE of 0.011 was found. The maximum errors were 5.37 for ΔE_{00} and 0.034 for RMSE. Furthermore, an adequate geometrical calibration target was designed. The image registration was done with a fitted polynomial curve model. Using the calibration target we also determined the residual displacement of the image registration at specific key-points. The best performance was achieved using a 4th degree model, resulting in residual displacements smaller than 0.08 pixel.

In conclusion, it was identified that the image registration with a polynomial of the 4th degree is more accurate than a polynomial of the 2nd degree. The spectral reflectance estimation is improved by averaging images to reduce the influence of sensor noise. Also by using the full dynamic range of the sensor for each image channel the influence of sensor noise can be decreased. Even though it was found that the color measurement performance is influenced negatively by dark color

patches, we still believe that such a multi-spectral imaging system can be of much use for device-independent color measurements.

References

1. N. Hagen and M. W. Kudenov, "Review of snapshot spectral imaging technologies," *Optical Engineering*, vol. 52, no. 9, p. 090901, 2013.
2. Y. Monno, T. Kitao, M. Tanaka, and M. Okutomi, "Optimal spectral sensitivity functions for a single-camera one-shot multispectral imaging system," *Proceedings - International Conference on Image Processing, ICIP*, pp. 2137–2140, 2012.
3. B. Jähne, *Digitale Bildverarbeitung und Bildgewinnung*. Heidelberg: Springer Vieweg, 2013.
4. Chromasens Corona - <https://www.chromasens.de/en/product/corona-ii-led-line-scan-light>, 7 2018.
5. T. Eckhard, *Design considerations for line-scan multi-spectral imaging systems Application in reflectance and color measurements*, 2015.
6. <http://kyamagu.github.io/mexopencv/>, 2 2018.
7. A. Goshtasby, "Piecewise linear mapping function for image registration," *Pattern Recognition*, vol. 19, pp. 459–466, 1986.
8. <https://www.konicaminolta.eu/de/messgeraete/produkte/grafische-industrie/fd7/einfuehrung.html>, 7 2018.
9. CIE, "Technical Report: Improvement To Industrial Colour- Difference Evaluation," *Comission Internationale de L'Éclairage*, 2001.

Visuelle Szenendiskretisierung mittels wahlfreien Bildrastern am Beispiel von Drei-, Vier- und Sechseckpixeln

**Visual discretisation of scenes by random grids using the
example of triangular, rectangular and hexagonal pixels**

Robert Manthey und Danny Kowerko

robert.manthey@informatik.tu-chemnitz.de

Juniorprofessur Media Computing,

Technische Universität Chemnitz, 09107 Chemnitz

Zusammenfassung Die Abbildung einer kontinuierlichen dreidimensionalen Szene auf die in verarbeitbare Elemente unterteilte zweidimensionale Fläche der Netzhaut eines Auges oder dem Aufnahmeelementen einer Digitalkamera erfordert eine Diskretisierung des empfangenen Lichtes, wofür häufig Rechteckraster zum Einsatz kommen obwohl in der Biologie Hexagonale vorherrschen. Verschiedene Arbeiten bestätigen die Vorteile von Hexagonen, offenbaren aber auch die Problematik der nicht vorhandenen Aufnahmesysteme. In dieser Arbeit wird eine Lösung für dieses Problem vorgestellt, welche die Möglichkeit bietet nahezu beliebige Szenen mit potenziell beliebigen Rasterungen in zweidimensionale Bilddaten zu überführen und somit deren Eigenschaften sowie mögliche Anwendungen umfangreicher untersuchen und evaluieren zu können.

Schlagwörter Rasterung, Hexagon, Bildgewinnung, Bilderzeugung, Bildverarbeitung.

Abstract Projecting the continuous three-dimensional scene to a two-dimensional plane be composed of processing elements requires discretization of the incoming light like in retina or digital camera. This is almost entirely done by rectangle grids although biological systems be dominated by hexagonal grids. Various articles confirm the advantages of hexagons, but also the difficulty to capture images with non-rectangle grid. This work present a solution to the problem and allow the rasterization of scenes with nearly any grid structure. It simplifies more detailed analyses and evaluations of the hexagonal grid as well as other tessellations.

Keywords Discretisation, hexagon, image acquisition, image capture, image Processing.

1 Einleitung

Augen und Kameras stellen als optische Wahrnehmungssysteme eine wichtige Informationsquelle für biologische und technische Systeme dar um z.B. die Essbarkeit von Früchten, die Entfernung zu Beutetieren, die Qualität von Bauteilen oder geografische Gegebenheiten, Abbildung 1 zu beurteilen. Hierzu werden über größere Entfernungen Form, Struktur und bestimmte Eigenschaften der Oberfläche der Objekte durch Analyse der Farbe und Intensität des von der Oberfläche reflektierten sichtbaren Lichts, der abgegebenen Infrarotstrahlung oder auch deren räumliche und zeitliche Veränderung erfasst und von nachfolgenden Verarbeitungs- und Analyseschritten entsprechend ausgewertet. Dabei kommt der zur technischen Weiterverarbeitung notwendigen Diskretisierung der Szenenprojektion als erstem Glied der Verarbeitungskette eine besondere Bedeutung zu, da die unterschiedliche Abbildungseffizienzen verschiedener Diskretisierungsraster mit einer Reduzierung der Menge an Informationen einhergehen [1].

Untersuchungen möglicher Diskretisierungsraster von [2], [3], [4] u.a. zeigten bereits beim Aufkommen des Bereiches der digitalen Bildverarbeitung, dass eine Verwendung von dreieckigen, rechteckigen und sechseckigen Grundstrukturen für die Konstruktion ebener, flächenfüllender, lückenloser, symmetrischer Raster für digitalen Aufnahmesystemen geeignet sind. Den dabei ebenfalls ermittelten deutlichen Unterschieden im Bereich der Abbildungseffizienzen [4, 5], der für

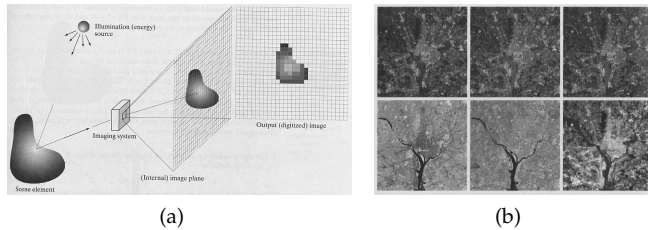


Abbildung 1: (a) Schematischer Bildaufnahmeprozess [7, S.51] mit Beleuchtung, Objekt, Aufnahmesystem und diskretisiertem Bild mit Rechteckraster. (b) Aufnahmen eines Flusses im sichtbaren blauen, grünen und roten, sowie dem nah-, mittel- und langwelligem infrarotem Spektralbereichen [7, S.846].

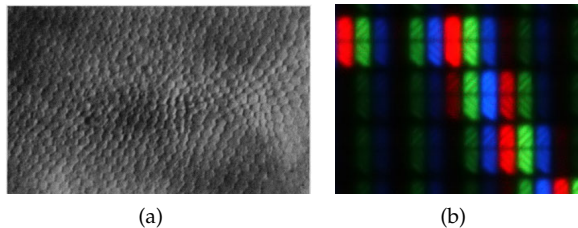


Abbildung 2: (a) Detailaufnahme der Netzhautstruktur eines menschlichen Auges mit ihren meist sechseckigen Fotorezeptoren [8] und (b) eine technisches Anzeigesystems mit nahezu rechteckigen Elementen.

hexagonbasierte Systeme exemplarisch zeigten um über 40% besseren Ausführungsgeschwindigkeit [6] sowie weiterer Vorteile steht allerdings das Fehlen nativer Aufnahmesysteme mit dreieckigen, sechseckigen o.a. Strukturen entgegen. Im Gegensatz dazu weisen biologische, evolutionär über jahrmillionen entwickelte und optimierte Aufnahmesysteme wie Facetten- oder Linsenaugen eine hexagonale Rasterung auf, wie in Abbildung 2 zu sehen.

Im Rahmen dieser Arbeit wurde ein System entwickelt, welches beliebige virtuelle Szenen mit wahlfreier Rasterung aufnehmen kann, somit die Problematik der fehlenden nativen Aufnahmesysteme behebt und Daten für weitere Analysen zur Verfügung stellt.

2 Grundlagen

Um Informationen über Formen und Strukturen einer Szene speichern, analysieren und verstehen zu können, besteht die Notwendigkeit die von dieser ausgehenden Signale wie Licht, infrarote Wärmestrahlung oder Radiowellen und deren Eigenschaften in eine geeignete Form zu überführen. Die Eigenschaften, wie räumliche, zeitliche und Intensitätsauflösung, dieser Abbildung beeinflussen maßgeblich die Leistungsfähigkeit aller nachfolgenden Analysen und des Verständnisses der Szene. [3,9]

Um die auf einer Sensorfläche wie der Netzhaut oder einem Bildsensor auftreffende informationstragenden Strahlung weiterverarbeiten zu können muss diese in kleiner Einheiten wie beispielsweise Fotorezeptoren oder Fotodioden unterteilt werden, welche die Überführung der aufgetroffenen Strahlung in diskrete Werte der zugehörigen Pixel durchführen [9]. Die Form und Anordnung dieser grundlegenden Strukturen beeinflusst zum Einen den Umfang und die Vollständigkeit der Strahlungsabbildung, wie an Abbildung 3 in deren diskrete Repräsentation, zum Anderen aber auch deren Effizienz [4, 7]. Auf dieser im Allgemeinen als eben betrachtete Fläche erfolgt die Verteilung der Grundstrukturen sowohl möglichst lückenlos, um keinen Verlust der dort auftreffenden Informationen zu erleiden, als auch überdeckungslos, um Doppelungen zu vermeiden. Eine Vereinheitlichung und Vereinfachung von Größen und Formen reduziert den Aufwand und die Komplexität nachfolgende Berechnungen. Dies reduziert die möglichen Grundstrukturen auf Dreiecke, Rechtecke und Sechsecke und führt zu unterschiedlichen möglichen Abbildungen der Szene auf ihre diskreten Repräsentationen ähnlich Abbildung 4. [3]

3 Verfahren

Um den in Abbildung 1(a) dargestellten Prozess nachzustellen und wahlfrei Rasterungen zu realisieren wird die Raytracing-Software POV-Ray¹ eingesetzt. Diese erlaubt die Nutzung von Kameras mit Bildsensoren bestehend aus einer Menge beliebig definierter Dreiecke, was in

¹ <http://www.povray.org>

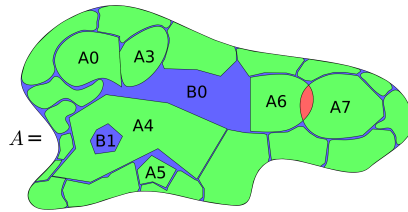


Abbildung 3: Beispielhafte Unterteilung der Sensorfläche $\mathcal{A}$ in Grundstrukturen (grün) und Lücken (blau) mit jeweils unterschiedlicher Form und Größe. Die Informationen der Sensoren $A0$ und $A3$ sind aufgrund der unterschiedlichen Formen trotz ähnlicher Größe nur begrenzt vergleichbar, was nachfolgende Berechnungen und Interpretationen erschwert. Die Sensoren $A6$ und $A7$ sind nicht Überdeckungsfrei (rot) sodass Doppelungen auftreten können. Sensor $A5$ ist von nahezu drei Seiten vom stark unregelmäßigen $A4$ umgeben wodurch die Bestimmung von Kanten o.ä. beeinträchtigt oder der Nutzen von $A5$ eingeschränkt ist. $A4$ umschließt das Loch $B1$ was Flächenbestimmungen erschwert und wie $B0$ zum Verlust von Informationen führt.

Kombination mit nachfolgenden Verarbeitungsschritten wahlfreie Raster ermöglicht.

Für die Erstellung der Bilddaten mit den gewünschten Rastern, wie in Abbildung 5 dargestellt, werden die zu untersuchenden Szenen unabhängig von der endgültigen Rasterstruktur definiert. Die hierbei eingesetzten geometrischen Elementen setzen sich bei Bedarf zu größere Objekte zusammen und erlauben somit die Erzeugung beliebig komplexer Szeneobjekte und Szenen. Mittels eines wahlfrei definierbaren Dreiecksnetzes (*Mesh*) erfolgt die Festlegung der Position, Größe und Ausrichtung der für die virtuelle Kamera genutzten Sensorfläche innerhalb jeder Szene. Das auf jedes einzelne Dreiecke eingestrahlt Licht der Szene bildet dabei auf jeweils ein virtuell Pixel ab und erzeugt das Bild einer Kamera mit dreieckigen Pixeln. Im nachfolgenden Verarbeitungsschritt erfolgt die Kombination der Pixel, deren Dreiecke zur gleichen Grundstruktur in der gewünschten Rasterung gehören, mittels Durchschnittsbildung.

Gleichzeitig erstellt die Standardkamera mit quadratischer Rasterung eine Aufnahme der Szene mit gleicher Position, gleicher Ausrichtung und gleicher Sensorfläche, wodurch beide vergleichbar sind.

4 Ergebnisse

Zur Untersuchung der Funktionsfähigkeit der vorgestellten Lösung wurden verschiedene Szenen mit geometrischen Objekten und komplexeren Szenen mit bekannten Testobjekten wie Abbildung 6 erstellt. Einige Szenen zeigen Oberflächen mit hochaufgelöste Bilder welche mit deutlich niedrigerer Auflösung aufgenommen wurden, sodass diese möglichst fehlerfreie Bilder mit dem gewünschten Raster erzeugen, ähnlich 7.

Die erzeugten Beispiele zeigen bereits den Vorteil der Abbildung runder Kanten bei drei- bzw sechseckiger Rasterung, welche in natürlichen Umgebungen häufig vorkommen. Auch Rauschen wie in (a) scheint beim Sechseckraster weniger ausgeprägt zu sein.

5 Zusammenfassung

Im Gegensatz zu biologischen Sehsystemen verwenden technische Lösungen fast ausschließlich rechteckige Raster bei der Bilderzeugung, welche einige Nachteile aufweisen. Der damit einhergehende Nachteil fehlender technischer Aufnahmesysteme stellte bisher ein erhebliches Problem bei Untersuchungen zu anderen Rastern dar und kann mit der hier vorgestellten Lösung überwunden werden. Die dargestellten Beispiele zeigen bereits einige Unterschiede zwischen rechteckbasierten, dreieckbasierten und sechseckbasierten Bilderaufnahmen, sowie deren Vorteile gegenüber der konventionellen technischen Lösung.

Danksagung

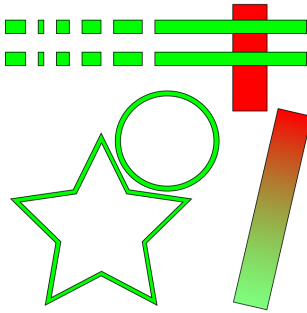
Diese Arbeit wurde vom BMBF im Rahmen des Unternehmen Region geförderten InnoProfile-Transfer-Initiative *localizeIT* (Förderkennzeichen: 03IPT608) durchgeführt.

² <http://sipi.usc.edu/database/download.php?vol=misc&img=4.2.04>

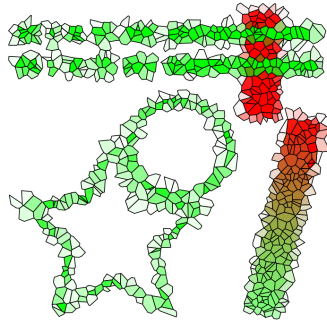
³ <http://graphics.stanford.edu/data/3Dscanrep/#bunny>

Literatur

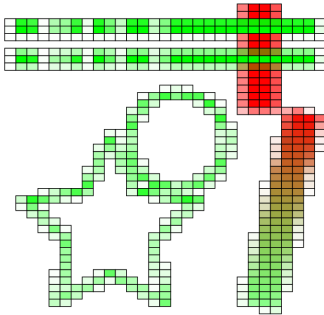
1. X. He and W. Jia, "Hexagonal structure for intelligent vision," in *International Conference on Information and Communication Technologies*. IEEE, August 2005, pp. 52–64.
2. M. J. Golay, "Hexagonal parallel pattern transformations," *Transactions on Computers*, vol. C-18, no. 8, pp. 733–740, August 1969.
3. E. Deutsch, "Thinning algorithms on rectangular, hexagonal, and triangular arrays," *Communications of the ACM*, vol. 15, no. 9, pp. 827–837, September 1972. [Online]. Available: <http://doi.acm.org/10.1145/361573.361583>
4. C. Wüthrich and P. Stucki, "An algorithmic comparison between square- and hexagonal-based grids," *Graphical Models and Image Processing*, vol. 53, no. 4, pp. 324–339, July 1991.
5. A. L. Nel, "Hexagonal image processing," in *Southern African Conference on Communications and Signal Processing*. IEEE, June 1989, pp. 109–113.
6. R. C. Staunton, "The design of hexagonal sampling structures for image digitization and their use with local operators," *Image and Vision Computing*, vol. 7, no. 3, pp. 162–166, August 1989. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0262885689900401>
7. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Pearson, 2008.
8. C. A. Curcio, J. Kenneth R. Sloan, O. Packer, A. E. Hendrickson, and R. E. Kalina, "Distribution of cones in human and monkey retina: individual variability and radial asymmetry." *Science*, vol. 236, no. 4801, pp. 579–582, May 1987. [Online]. Available: <http://www.jstor.org/stable/1699166>
9. W. Burger and M. J. Burge, *Digitale Bildverarbeitung - Eine Einführung mit Java und ImageJ*. Berlin, Heidelberg: Springer-Verlag, August 2005.



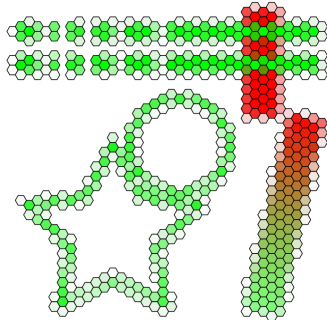
(a)



(b)



(c)



(d)

Abbildung 4: Darstellung der informationshaltigen Pixel die nach einer beispielhafte Kachelung der Szene aus (a) mit Pixeln durchschnittlich gleicher Fläche aber ungleicher Form (b), mit rechteckigen Pixeln gleicher Form und Größe (c) und sechseckigen Pixeln gleicher Form und Größe entsteht (d). Die glatten Kanten der Objekte werden in (b) deutlich unregelmäßig und uneinheitlich abgebildet, wohingegen dies in (c) und (d) strukturierter erfolgt. Die Umrisse des Sterns und des Kreises sind ebenso deutlicher erkennbar wie die parallele Anordnung der grünen Rechtecke links oben.

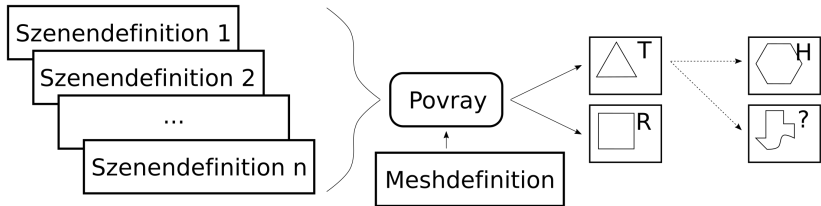


Abbildung 5: Schematischer Ablauf der Erstellung von Bildern mit wahlfreien Bildrastern. Eine abstrakte, rasterunabhängige Szenendefinition wird mittels POV-Ray in eine Aufnahme mit Dreieck- und Rechteckrasterung umgewandelt. Nachfolgende Verarbeitungsschritte kombinieren das entsprechend der Meshdefinition erstellten Dreiecksraster zum Zielraster.

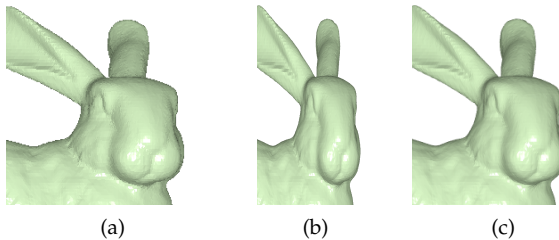


Abbildung 6: Kopfbereich des Stanford Bunny² gerastert mit (a) 256×256 rechteckigen, (b) 476×824 dreieckigen und (c) 238×275 sechseckigen Pixeln. Deutlich treten die Treppenstufenartefakte im Rechteckrasten an den Kanten und den Flächen der Schulter hervor. Letztere sind auch im Dreiecksraster noch auffälliger als im Sechseckraster.



Abbildung 7: Testbild Lena^3 gerastert mit (a) 64×64 rechteckigen und (b) 59×68 sechseckigen Pixeln bei gleicher Sensorfläche. Runde Kanten weisen im Rechteckraster deutlich sichtbarere Treppenstufenartefakte und mehr Rauschen auf.

Image processing for scatter-plate microscopy

Stephan Ludwig, Benjamin Le Teurnier, Giancarlo Pedrini, and
Wolfgang Osten

Universität Stuttgart, Institut für Technische Optik,
Pfaffenwaldring 9, 70569 Stuttgart

Abstract While light-scattering media are usually regarded as an obstacle in imaging a new technique called scatter-plate microscopy uses the diffusing property of such media for imaging micro structures. Using an ordinary ground glass diffuser instead of a microscope objective the method can be used for both technical as well as for biological microscopic investigations with both variable magnification and variable numerical aperture (NA). The simple replacement of an objective in an imaging system by a random diffuser generates naturally not an image of the sample but a random pattern. An image is eventually acquired by the calculation of the cross-correlation of this pattern with the previously recorded point spread function (PSF) of the diffuser. In the whole imaging process the autocorrelation of the PSF can be regarded as the impulse response of the microscope. The knowledge of this impulse response can be used for deconvolution. This image processing method is able to reduce the occurring blur and to improve both contrast and resolution of the image.

Keywords Speckle, scatter-plate, microscopy, deconvolution, memory effect, scattering.

1 Introduction

Imaging through light scattering media is of high interest for both technical and biological investigation. Most approaches in this field regard the scattering media as obstacle screening the region of interest from the actual imaging elements. On the other hand many traditional imaging techniques were recently combined with scattering elements to improve

their performance. Our approach however to realize imaging through scattering media is quite different. We regard the scattering element itself as an imaging element able to replace a whole microscope objective built of several lenses [1, 2]. In scatter-plate microscopy an image is acquired by cross-correlating the speckle pattern caused by illuminating the scattering element with a point source with the pattern recorded when the light emitted by an incoherently illuminated sample is scattered [3]. Sharp imaging with optimized resolution is realised when the point source and the sample are positioned at the same distance to the scatterer and with only a small lateral displacement. The magnification achieved is given by the distance of the scatterer from the recording sensor over the distance of the scatterer from the sample. Since these distances can be selected in wide sense arbitrarily, with a single diffuser different magnifications can be realised. In addition to this our scatter-plate microscope reveals some more indisputable advantages in comparison to conventional microscopes: the scatter-plate is thin and light, easy to fabricate and low-cost. Not only the magnification but also the NA, the focal length and the field of view (FOV) are variable. And finally the working distance is extendable as desired.

2 Methods

The so called point spread function (PSF) of an imaging system is the intensity distribution (in the case of incoherent imaging) or the amplitude distribution (in the case of coherent imaging) in the image plane generated by a point source emitting in the object plane. The PSF carries all information about the image formation capabilities of the system. Since scatter-plate microscopy is a incoherent imaging method in this article the PSF $S(\mathbf{r}, \hat{\mathbf{r}}, z, \hat{z})$ always denotes an intensity distribution, with the image plane at distance z from the imaging system and the object plane at distance $\hat{z}$ being perpendicular to the optical axis (see Fig. (1)). $\mathbf{r}$ describes the lateral position in the recorded intensity distribution whereas $\hat{\mathbf{r}}$ means the lateral position in the object plane. The image formation follows the rule

$$I_{ob}(\mathbf{r}, z, \hat{z}) = \iint S(\mathbf{r}, \hat{\mathbf{r}}, z, \hat{z}) O(\hat{\mathbf{r}}, \hat{z}) d\hat{\mathbf{r}} \quad (1)$$

where I_{ob} is the recorded intensity distribution in the image plane

whereas O means the intensity distribution emitted by the object. To obtain the full PSF in principle the intensity distribution has to be recorded for all possible point source positions $\hat{\mathbf{r}}$ in the object plane. However for conventional lens based imaging systems the PSF is extensively shift invariant over a wide area in the object plane meaning that $S(\mathbf{r}, \hat{\mathbf{r}}, z, \hat{z}) = S(\mathbf{r} + \Delta\mathbf{r}, \hat{\mathbf{r}} + \Delta\hat{\mathbf{r}}, z, \hat{z})$ with $\Delta\hat{\mathbf{r}}$ being the lateral shift in the position of the point source and $\Delta\mathbf{r} = -(z/\hat{z})\Delta\hat{\mathbf{r}}$. Therefore the imaging properties are characterized by a single intensity distribution $S(\mathbf{r}, z, \hat{z})$ recorded with the point source positioned at the intersection of the optical axis with the object plane and Eq. 1 simplifies to a convolution integral (since the distances z and $\hat{z}$ are considered to be fixed they will be dropped in the following):

$$I_{ob}(\mathbf{r}) = \int \int S(\mathbf{r} - \hat{\mathbf{r}})O(\hat{\mathbf{r}})d\hat{\mathbf{r}} = S(\mathbf{r}) * O(\mathbf{r}) \quad (2)$$

For imaging with a scatter plate Eq. (1) still holds true however the PSF of the scatter plate is a speckle pattern and its shift-invariance is restricted to a small area in which the so called memory effect holds true. Nevertheless if the field of view (FOV) in the object plane is part of this memory effect area also Eq. (2) holds true and the recorded intensity distribution in the image plane is a convolution of the object's intensity distribution and the PSF. With a scatter-plate of course the intensity distribution I_{ob} we record in the image plane is not anything we could call an image but a random intensity pattern. Cross-correlating ($\star$) now this intensity pattern with the PSF however will lead to an image if the field of view is restricted to the memory effect range:

$$I(\mathbf{r}) = I_{ob}(\mathbf{r}) \star S(\mathbf{r}) = [O(\mathbf{r}) * S(\mathbf{r})] \star S(\mathbf{r}) = O(\mathbf{r}) * [S(\mathbf{r}) \star S(\mathbf{r})] \quad (3)$$

$C(\mathbf{r}, z) = S(\mathbf{r}) \star S(\mathbf{r})$ the autocorrelation of a random speckle pattern is a sharply peaked function similar to the PSF of an ordinary imaging lens (airy disc), but with a constant offset C . Approximating this autocorrelation with a Dirac delta distribution $C(\mathbf{r}) \approx \delta(\mathbf{r}) + C$ reveals the imaging capability of the method:

$$I(\mathbf{r}) = O(\mathbf{r}) * [S(\mathbf{r}) \star S(\mathbf{r})] \approx O(\mathbf{r}) * [\delta(\mathbf{r}) + C] = O(\mathbf{r}) + \tilde{C} \quad (4)$$

Now the principle of image acquisition with a scatter-plate microscope is obvious: in a first step the PSF of the scatter plate is recorded. Afterwards the point source is removed from the object plane and a sample in its extend restricted to the memory effect range of the scatter-plate is placed in the object plane and illuminated incoherently. The resulting random pattern is recorded. To reduce computational effort the cross-correlation of the patterns are calculated in Fourier space turning the deconvolution integration into a simple multiplication:

$$I(\mathbf{r}) = \mathcal{F}^{-1} \left(\hat{S}(\mathbf{p})^* \hat{I}_{ob}(\mathbf{p}) \right), \quad (5)$$

with $\hat{\cdot}$ denoting the two-dimensional Fourier transform, $\mathcal{F}^{-1}$ the operator of the inverse Fourier transform, $\mathbf{p}$ the spacial frequency vector and $*$ denoting the complex conjugate. To compute the Fourier transform and its inversion we use the FFT-algorithm.

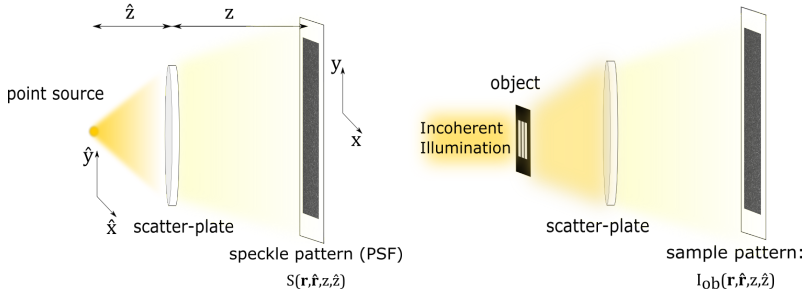


Figure 1: Experimental setup for image acquisition. Two random patterns are recorded. For image acquisition the speckle pattern of a point source (left) is cross-correlated with the random pattern caused by an incoherently illuminated sample.

2.1 Deconvolution in scatter-plate microscopy

Deconvolution uses the knowledge of the transfer function of shift-invariant imaging systems to improve resolution and contrast and to reduce the blur caused by aberrations in the imaging system. The basic

ideas follows from Eq. (1). After transferring the equation into Fourier space resulting in a multiplication, the equation can be changed after the Fourier transform of the object:

$$\hat{I}_{ob}(\mathbf{p}) = \hat{S}(\mathbf{p})\hat{O}(\mathbf{p}) \Rightarrow \hat{O}(\mathbf{p}) = \frac{\hat{I}_{ob}(\mathbf{p})}{\hat{S}(\mathbf{p})} \quad (6)$$

However the application of this naive approach is not recommended since the division through values close to zero leads to unwanted features in the result. Therefore many more sophisticated deconvolution approaches were developed to overcome this problem. In scatter-plate microscopy the application of deconvolution on the random sample pattern $I_{ob}(\mathbf{r})$ using the recorded PSF $S(\mathbf{r})$ does not lead to satisfying results. However Eq. (3) is also a convolution and therefore deconvolution can be applied on the acquired image regarding the autocorrelation of the PSF $C = S(\mathbf{r}) \star S(\mathbf{r})$ as the known linear filter function [4]. Here we present three different deconvolution methods proved to be useful to improve the quality of the images acquired with our scatter-plate microscope.

Wiener filter with constant regularization

The Wiener filter [5] approach estimates the object by minimization of the mean square error. The filter is most easily described in the frequency domain:

$$O_{est}(\mathbf{r}) = \mathcal{F}^{-1} \left(\frac{\hat{C}^*(\mathbf{p})\hat{I}(\mathbf{p})}{|\hat{C}(\mathbf{p})|^2 + N(\mathbf{p})} \right). \quad (7)$$

$N(\mathbf{p})$ is the mean power spectral density of the noise. We approached this regularization function with a constant parameter α .

Gold algorithm

The Gold algorithm [5, 6] develops in an iterative process a solution, which, when convolved by the known filter function, provides the ac-

quired image. The iteration follows the rule

$$O_{est}^{i+1}(\mathbf{r}) = O_{est}^i(\mathbf{r}) \frac{I(\mathbf{r})}{C(\mathbf{r}) * O_{est}^i(\mathbf{r})}. \quad (8)$$

For an initial estimate $O_{est}^1(\mathbf{r})$ of the object we normalized the acquired image. The algorithm always ensures a non-negativity condition ($O_{est}^i(\mathbf{r}) > 0$) for optical objects (i.e. $O_{est}^1(\mathbf{r})$ and $C(\mathbf{r})$ need to be positive). Disadvantages of this method are that first it gives poor results when the initial estimate $O_{est}^1(\mathbf{r})$ is rather noisy (noise is not taken into account by the algorithm) and second it suffers from slow convergence when the pure autocorrelation C is used. However we achieve a remarkable improvement in convergence by subtracting from C the mean value of its noisy background. This step leads to negative values in the filter function but as long as there do not occur negative values in the estimated object structures O_{est}^i this does not make any problems.

Jansson Van-Cittert algorithm

Like the Gold algorithm the Jansson Van-Cittert algorithm [7, 8] develops iteratively a solution which, when convolved by the known filter function, provides the acquired image:

$$O_{est}^{i+1}(\mathbf{r}) = O_{est}^i(\mathbf{r}) + r(O_{est}^i(\mathbf{r})) [I(\mathbf{r}) - C(\mathbf{r}) * O_{est}^i(\mathbf{r})], \quad (9)$$

with

$$r(O_{est}^i(\mathbf{r})) = r_0 \left[1 - 2 \left| O_{est}^i(\mathbf{r}) - \frac{1}{2} \right| \right]. \quad (10)$$

For the application of the Jansson Van-Cittert algorithm in scatter-plate microscopy it turned out to be useful to normalize both the acquired image $I(\mathbf{r})$ and the subtrahend $C(\mathbf{r}) * O_{est}^i(\mathbf{r})$ in each step of the iteration. Normalization is realized by subtracting the minimum pixel value followed by a division through the maximum pixel value.

3 Experimental setup and image acquisition

The experimental setup is shown in Fig. 2. We demonstrate the possibilities of scattering-plate microscopy by imaging a 1951 USAF high res-

olution test target and by imaging onion cells. The smallest element of the test target (group 9 element 3) indicates a resolution of 645.1 lp/mm or $0.78\mu\text{m}$. As a light source we used a frequency-doubled Nd:YAG-laser with a wavelength of 532nm. To illuminate the sample the laser beam was let through a rotating ground glass diffuser to destroy the spatial coherence of the illumination. In the case of the onion cells this rotating diffuser was located very close to the sample. For imaging the USAF-target a condenser consisting of a collimating and a light focusing lens (focal length 5cm and 3cm) were installed between the rotating ground glass and the sample to improve the sample illumination. To approximate a point source we focused the laser beam with a microscope objective with 40X magnification into the object plane. As scatter-plate we used a DG-600 ground glass diffuser from THORLABS. The ground glass was mounted on a step rotator stage from PI GmbH & Co.KG enabling imaging for different orientations of the diffuser. An averaging over images acquired with different orientations (five in the case of the USAF-target, ten in the case of the onion cells) improved significantly the image quality. The patterns were recorded with a CCD-Camera from VISTEK-GMBH with 3280×4896 pixels and a pixel size of $7.45\mu\text{m} \times 7.45\mu\text{m}$. The FOV was restricted to the memory effect range by a circular pinhole with a diameter of $400\mu\text{m}$. A diaphragm between the scatter-plate and the object enabled the specification of the NA.

It turns out that even small trends in the intensity distribution e.g. due to a slight misalignment of the camera lead to a bright cross-like feature in the center of the calculated cross-correlation. To overcome this image disturbing behaviour it proves advantageous to apply a shading correction to the acquired patterns. This shading correction was realized by dividing the patterns through low-pass filtered versions of them (Gaussian low-pass filter with a standard deviation of 100 pixels). The images obtained by calculating the cross-correlation were normalized by subtracting the minimum pixel value followed by a division through the maximum pixel value. It might happen that also after the shading correction due to dust on the CCD or due to destroyed pixels a bright spot-like feature remains in the center of the image. However this spot does not really disturb most images and just has not to be taken into account for the determination of the maximum in the normalization procedure.

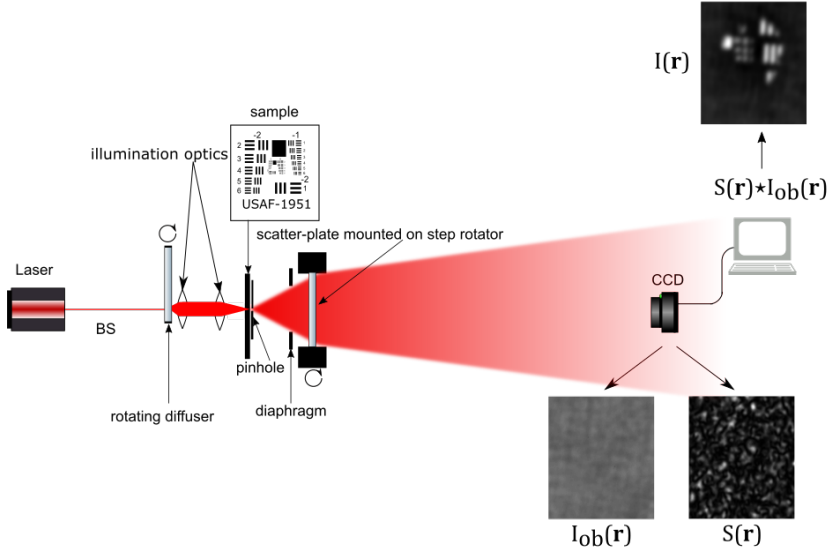


Figure 2: Experimental setup to record the pattern caused by the USAF-target according to the description in section 3. The pattern of the onion cells were recorded without the two lenses between the rotating diffuser and the scatter-plate. Instead the rotating diffuser was located very close to the sample. To record the point spread function the sample and the rotating diffuser (and the illumination optic) were removed and the microscope objective focuses the illumination beam into the object plane.

4 Experimental result

4.1 USAF-target

The USAF target was positioned at a distance of 5cm to the scatter-plate, the recording CCD in a distance of 100cm leading to a magnification of 20X. Images were acquired with an NA of 0.3 for five different orientations of the scatter-plate. The normalized average is shown in Fig. (3 a). The deconvolution methods were applied according to the descriptions in 2 on the five acquired images and afterwards averaging

and normalization were applied. For the Wiener filter (Fig. (3 b)) the noise-parameter was set to $\alpha = 5 \times 10^{13}$, for the Gold algorithm (Fig. (3 c)) we applied 100 iterations, for the Jansson van-Cittert algorithm (Fig. (3 d)) we applied 40 iterations. The plotted intensity distribution in the bottom of Fig. (3) demonstrate the influence of deconvolution. All the three applied methods improve the contrast and the resolution. Without deconvolution group 8 element 4 (indicating a resolution of $1.38\mu\text{m}$) of the USAF target is resolved. With the gold algorithm group 8 element 5 ($1.23\mu\text{m}$), with Wiener filter group 8 element 6 ($1.1\mu\text{m}$) and with the Jansson van-Cittert method even group 9 element 1 ($0.98\mu\text{m}$) is resolved. The diffraction limit of optical systems with $\text{NA}=0.3$ is a resolution of $0.89\mu\text{m}$.

4.2 Onion cells

The slide with a piece of onion skin was positioned in the object plane. The normalized average over 10 images acquired with 20X magnification and an NA of 0.3 is shown in Fig. (4 a). We applied a Wiener filter with a regularization parameter $\alpha = 10^{15}$ (Fig. (4 b)) and both the Gold algorithm (Fig. (4 c)) and the Jansson van-Cittert algorithm (Fig. (4 d)) with 10 iterations. For all the three deconvolution methods the details appear more defined than in the pure image. Nevertheless by applying deconvolution on this biological sample the great drawback of these methods also become obvious: a remarkable noise amplification can accompany these methods of image processing.

Fig. (4 e) shows a part of the sample imaged with a conventional microscope objective with the same $\text{NA}=0.3$ and the same magnification (20X). Compared to the result recorded with the scatter-plate microscope this conventional image is better resolved and less noisy but on the other hand the conventional image shows less contrast.

5 Conclusions

We have shown the successful application of deconvolution methods in scatter-plate microscopy. We applied the Wiener filter with a constant regularization parameter the Gold algorithm and the Jansson van-Cittert algorithm on both technical and biological sample. Each method

led to a remarkable improvement of the image contrast and resolution. With the Jansson van-Cittert-algorithm we were able to resolve group 9 element 1 of the USAF target referring to a resolution of $0.98\mu\text{m}$.

6 Acknowledgements

This work was supported by the German Research Foundation (DFG) under Grant no. OS111/49-1.

References

1. A. K. Singh, G. Pedrini, M. Takeda, and W. Osten, "Scatter-plate microscope for lensless microscopy with diffraction limited resolution," *Scientific Reports*, vol. 7, no. 1, p. 10687, September 2017. [Online]. Available: <https://www.nature.com/articles/s41598-017-10767-3>
2. W. Osten, K. Frenner, G. Pedrini, A. K. Singh, J. Schindler, and T. Mitsuo, "Shaping the light for the investigation of depth-extended scattering media," *Proc. SPIE 10503, Quantitative Phase Imaging IV*, p. 1050318, 2018. [Online]. Available: <https://doi.org/10.1117/12.2296582>
3. I. Freund, "Looking through walls and around corners," *Physica A: Statistical Mechanics and its Applications*, vol. 168, no. 1, pp. 49 – 65, 1990. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/037843719090357X>
4. S. Ludwig, A. K. Singh, G. Pedrini, and W. Osten, "Scatter-plate microscope: improved image acquisition," *Proc. SPIE*, vol. 10677, pp. 10 677 – 10 677 – 11, 2018. [Online]. Available: <https://doi.org/10.1117/12.2306252>
5. J.-B. Sibarita, "Deconvolution microscopy," in *Microscopy Techniques*, 1st ed., ser. Advances in Biochemical Engineering, J. Rietdorf, Ed. Heidelberg Berlin: Springer, 2005, vol. 95, ch. 6, pp. 201–243.
6. R. Gold, "An iterative unfolding method for response matrice," *Argonne National Lab. Report ANL-6984*, 1964.
7. P. A. Jansson, Ed., *Deconvolution of Images and Spectra (2Nd Ed.)*. Orlando, FL, USA: Academic Press, Inc., 1996.
8. P. H. van Cittert, "Zum einfluß der spaltbreite auf die intensitätsverteilung in spektrallinien. ii," *Zeitschrift für Physik*, vol. 69, no. 5, pp. 298–308, May 1931. [Online]. Available: <https://doi.org/10.1007/BF01391351>

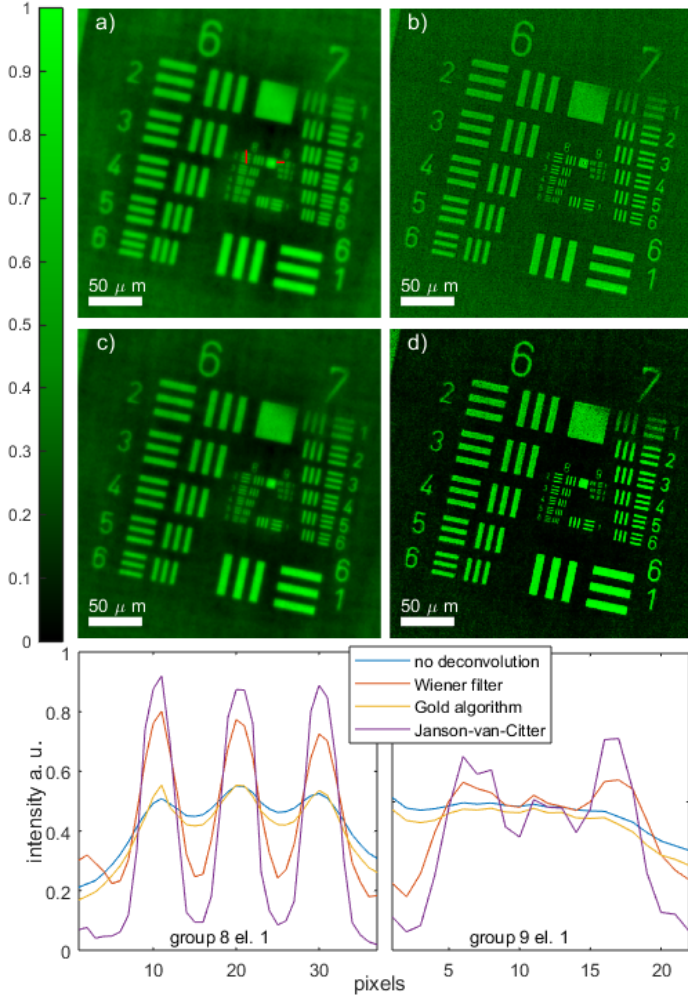


Figure 3: Central part of the USAF-target imaged with the scatter-plate microscope. Average of five acquisitions. a) without deconvolution, b) Wiener filter with $\alpha = 5 \times 10^{13}$, c) Gold algorithm after 100 iterations, d) Jansson Van-Cittert algorithm after 40 iterations. The plots below show the behaviour of intensity across the element 1 of group 8 and 9 as indicated by the red lines in image a).

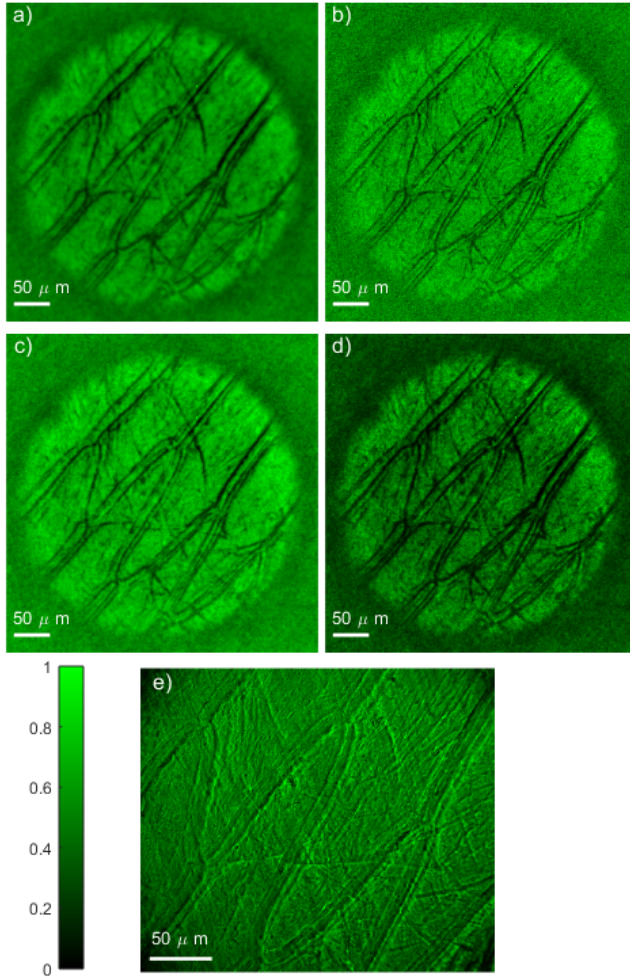


Figure 4: Onion cells imaged with the scatter-plate microscope. Average of ten acquisitions. a) without deconvolution, b) Wiener filter with $\alpha = 10^{15}$, c) Gold algorithm after 10 iterations, d) Jansson Van-Cittert-algorithm after 10 iterations. e) part of the sample imaged with a conventional microscope objective (magnification 20X, NA=0.3) and a CCD-Camera from XIMEA with 1280×1024 pixels and a pixel size of $5.3\mu\text{m} \times 5.3\mu\text{m}$.

Übersicht zur Übertragbarkeit der Leistungscharakteristik von Computer Vision Algorithmen bei synthetischen und realen Luftbildaufnahmen

Review on the transferability of algorithm performance on synthetic generated airborne imagery to real-world computer vision applications

Michael Krump, Christian Hellert, Georg Hummel und Peter Stütz

Universität der Bundeswehr München, Institut für Flugsysteme,
Werner-Heisenberg-Weg 39, 85577 Neubiberg

Zusammenfassung Virtuelle Umgebungen, wie die *Presagis* Suite, können die Datengenerierung für Methoden aus dem Bereich des maschinellen Lernens beschleunigen. Diese Arbeit gibt einen Überblick über verschiedene Arten und Anwendungsfelder und betrachtet die Anwendbarkeit von *Computer Vision* Algorithmen, welche mit synthetischen Daten trainiert und getestet wurden, in der Realität. Außerdem wird ein Konzept vorgestellt, um die Diskrepanz zwischen synthetischen und realen Daten zu quantifizieren und mit Hilfe von *Domain Adaptation* zu minimieren.

Schlagwörter Simulation, virtuelle Realität, Presagis, UAV, Reality Gap, Computer Vision Algorithmen, Domain Adaptation.

Abstract Virtual environments, such as *Presagis*, can accelerate the data generation for machine learning methods. On that point, this paper gives an overview of different types and applications. The applicability of *computer vision* algorithms, which have been trained and tested with synthetic data, in reality is considered. Furthermore, a concept is presented to quantify the gap between synthetic and real data and to minimize it by *domain adaptation*.

Keywords Simulation, virtual Reality, presagis, UAV, reality gap, computer vision algorithm, domain adaptation.

1 Einleitung

Sensornutzlastsysteme sind wichtiger Bestandteil bemannter und unbemannter Flugsysteme für die Gewinnung von Aufklärungs- und Überwachungsinformationen. Rechnerbasierte Datenverarbeitungsalgorithmen erhöhen den Automatisierungsgrad und tragen damit zur Unterstützung des Sensoroperators bei. Häufig werden dabei Methoden aus dem Bereich des maschinellen Lernens verwendet. Dabei wird die spätere Leistungsfähigkeit dieser Methoden in hohem Maß von der Verfügbarkeit und der Menge passender Test- bzw. Trainingsdatensätze beeinflusst. Ein vielversprechender Ansatz zur Kompensation dieser Problematik ist die Verwendung virtueller Simulationsumgebungen zur Erzeugung synthetischer Sensoraufnahmen. Dabei stellt sich die Frage, inwieweit diese im Hinblick auf die Leistungscharakteristik von *Computer Vision* (CV) Algorithmen mit realen Aufnahmen vergleichbar sind und welche Randbedingungen es zu beachten gibt. Im Folgenden soll ein Überblick über den Stand der Technik und über aktuelle Anwendungsfälle gegeben werden. Anschließend werden mögliche Vorgehenskonzepte vorgestellt, um die Differenzen zwischen realen und synthetischen Sensordaten und deren Auswirkungen auf mögliche Verarbeitungsalgorithmen zu minimieren.

2 Simulationen und virtuelle Umgebungen

Simulationsumgebungen mit Sensormodellen zur Rohdatengenerierung bieten unter anderem im Automobilbereich und der Luftfahrt-technik viele Vorteile. Der vollständige Test von Fahrerassistenz- und Sicherheitssystemen (ADAS, engl.: *Advanced Driving Assistance System*), wie z.B. Spurhalteassistent, Fußgängererkennung, Abstandsregelung und Notbremsassistent, in Verbindung mit den übrigen aktiven Fahrzeugsystemen ist bei klassischen Testfahrten aufwendig und aus Sicherheitsgründen oftmals nicht möglich. Fahrzeugsimulatoren ermöglichen die Durchführung der nötigen Testzyklen in kritischen Situationen und unter vielfältigen Umgebungsbedingungen [1, 2]. Dies führt zu kürzeren Produktentwicklungszyklen, Kostenreduktion und nicht zuletzt zu zuverlässigeren Systemen [2–4].

Noch größer ist der Gewinn bei luftgestützten CV-Algorithmen, denn

die Planung und Durchführung von entsprechenden Flugmissionen ist aufwendig und kostspielig. Außerdem sind Flüge generell durch gesetzliche Vorgaben stark eingeschränkt [5, 6]. Zusätzlich wird dabei nur einen Bruchteil der möglichen szenarischen Varianz (z.B. Jahreszeiten, geografische Regionen, aber auch Blickwinkel) und der beim späteren Einsatz auftretenden Umwelteffekte (z.B. Schnee, Regen, Nebel, Bewölkungsgrad) erfasst. Dies kann zu Algorithmen mit geringer Robustheit gegenüber schwankenden Umgebungsbedingungen führen. Ein Test von Kontroll- und Steueralgorithmen [7–9] oder von Methoden zur Hindernisvermeidung bei UAVs (engl.: *Unmanned Aerial Vehicle*) [10–12] ist nur in der Simulation risikofrei möglich. Dies trifft vor allem dann zu, wenn der Lernprozess auf einem *trial-and-error* basierten Verfahren in Verbindung mit *Reinforcement Learning* beruht. Hier werden Kollisionen im Laufe des Trainings bewusst provoziert, damit das Modell lernt, sie in der späteren Anwendung zu vermeiden [8, 12, 13]. Ein weiterer Vorteil in diesem Zusammenhang ergibt sich aus der exakten Wiederholbarkeit der Experimente.

Datenverfügbarkeit und Datengenerierung

Die Anwendung von Verfahren des maschinellen Lernens erfordert große Mengen annotierter Daten. Ein entscheidender Vorteil von virtuellen Umgebungen ist die Möglichkeit synthetische Sensorrohdaten mit zugehöriger *Ground Truth* zu generieren [3, 14–16]. Richter et al. [14] haben in diesem Zusammenhang ein Verfahren vorgestellt, um aus der Kommunikation zwischen modernen Computerspielen und der Grafikhardware gleichzeitig Bildmaterial und pixel-genaue semantische Annotationen zu erhalten.

Eine weitere Möglichkeit ist die Verwendung existierender Datensätze bzw. bereits aufgezeichneter Sensordaten. Im Bereich der Luftbildaufnahmen stehen nur wenige Datensätze zur Verfügung, die zudem häufig auf spezielle Anwendungen zugeschnitten sind und z.T. nur sehr beschränkte Anteile der späteren Operationsbedingungen erfassen. Die Daten müssen jedoch aufwendig von Hand annotiert werden, was den Umfang der Datensätze noch weiter reduziert. Außerdem ist im Nachhinein kein Einfluss auf die Aufzeichnungs- und Sensorparameter möglich, was den Test von geschlossenen Regelschleifen oder Mensch-Maschine-Interaktionen erschwert [2, 16].

Ein vor allem zum Training von Objektdetektoren interessantes Vor-

gehen wurde von Rozantsev et al. [17] vorgestellt. Dabei werden Hintergrundbilder mit einem vereinfachten 3D-Modell des zu identifizierenden Objekts überlagert. Durch Vergleich mit einer realen Aufnahme passt das System die Renderparameter derart an, dass im Hinblick auf die vom Detektor verwendeten Features eine möglichst hohe Korrelation zwischen synthetischen und realen Daten erzielt wird. Diese Parameter können nun verwendet werden, um eine große Anzahl an Trainingsdaten mit unterschiedlichen Blickwinkeln zu erzeugen. Gängige Objektdetektoren wurden mit den synthetischen Daten und mit realen Daten, mit und ohne nachträgliche Störeffekte, trainiert. Es wurde gezeigt, dass die Detektionsraten von UAVs auf Luftbildern durch die vorgestellte Methode signifikant gesteigert werden konnten [17]. Ansätze dieser Art werden allgemein auch als *Randomization* bezeichnet und wurden z.B. in [13] und [18] erfolgreich angewandt.

Realismus und Reality Gap

Der Begriff *Reality Gap* stammt ursprünglich aus der Robotik und bezeichnet die Differenz des Systemverhaltens zwischen Simulation und Realität [7, 13]. Ferwerda [19] stellte ein Konzept vor, das Simulationsdaten in drei hierarchisch angeordnete Stufen gliedert:

- **Physikalischer Realismus** Die Simulation bietet durch physikalisch korrekte Zusammenhänge die gleiche visuelle Stimulation wie die reale Szenerie
- **Photo-Realismus** Die Simulation erzeugt die identische visuelle Reaktion und ist optisch nicht unterscheidbar
- **Funktionaler Realismus** Die Simulation liefert die für den jeweiligen Anwendungsfall notwendige visuelle Information

Ziel der Sensorsimulation ist es, die Differenzen bei vertretbarem Modellierungsaufwand so gering zu halten, dass sie für den jeweiligen Beobachter bzw. Algorithmus vernachlässigbar sind.

3 Anwendungsfälle und Vergleiche mit der Realität

Im nachfolgenden Kapitel werden einige Anwendungsfelder von virtuellen Umgebungen aufgezeigt.

Objekterkennung und Hindernisvermeidung

Während in [20] und [10] Algorithmen ausschließlich mit synthetischen Daten trainiert und getestet werden, wird in [11, 12, 21, 22] auch der Leistungsunterschied zur Realität ermittelt. Hummel et al. [21] untersuchten eine visuelle Methode zum *Multi-Objekt-Tracking* anhand von simulierten und realen Luftbildaufnahmen einer vielbefahrenen Straße. Trotz der Leistungsunterschiede, die auf fehlendes Rauschen, reduzierte Störeffekte und prägnantere Merkmale in den simulierten Daten zurückgeführt wurden, zeigen die Ergebnisse aus Tabelle 1 die generelle Vergleichbarkeit in diesem Bereich.

Tabelle 1: Leistung des Objekterkennungs-
algorithmus (nach [21]).

	Erkennungs- rate	Precision	Recall
Realität	84 %	77 %	84 %
Simulation	90 %	93 %	85 %

Tabelle 2: Leistung der Hindernis-
vermeidung (nach [12]).

	Precision	IoU (freier Raum)	IoU (Hindernis)
Realität	80,3 %	63,4 %	63,3 %
Simulation	90,4 %	57,0 %	87,3 %

Neueste Ansätze von Sadeghi et al. [12] zeigen, dass mit einem UAV autonome Indoorflüge möglich sind, nachdem das dabei zur Kollisionsvermeidung verwendete neuronale Netz ausschließlich in einer virtuellen Umgebung trainiert wurde. In Tabelle 2 wird die Leistung anhand der *Intersection over Union* (IoU) verglichen.

Positionsschätzung

Bei autonomen Andockvorgängen im Weltraum [23] und Szenarien der Luftbetankung [6] ist eine präzise Positionsermittlung zwingend nötig. Kelsey et al. [23] präsentierten ein Vorgehen zur modellbasierten Lage-schätzung und testeten dieses unter anderem an einem simulierten und einem maßstabsgetreuen 3D-Modell der Magellan-Sonde. Die Auswertung zeigt eine hohe Übereinstimmung, wobei die Leistung in der Simulation aus den bereits genannten Gründen [21] leicht höher ist.

Tracking

Das Tracking von bewegten Objekten mit niedrig fliegenden UAVs ist ein vielversprechender Ansatz, da das Kamerasystem das Ziel aktiv verfolgen und damit die Tracking-Performance optimieren kann. Aufgrund fehlender bzw. ungeeigneter Datensätze für die Evaluierung von

UAV-basiertem Tracking haben Müller et al. [24, 25] einen synthetisch generierten Benchmark-Datensatz erstellt. Dieser enthält wichtige Faktoren, wie Kamerabewegungen, Blickpunktänderungen, Verdeckungen und Größenänderungen des Ziels. Umfassende online und offline Evaluierungen (d.h. mit und ohne Einfluss des Trackers auf die UAV-Steuerung) machten deutlich, dass unter anderem Größenänderungen des Zielobjekts zu den Haupteinflussfaktoren zählen.

Semantische Segmentierung

Experimente von Richter et al. [14] zur semantischen Segmentierung realer Bilddaten haben bewiesen, dass die Genauigkeit der Modelle durch Hinzunahme von synthetischen Daten aus 3D-Spiele-Engines deutlich gesteigert werden konnte. Wie Tabelle 3 zeigt, konnte dadurch die benötigte Menge an aufwendig von Hand annotierten Realdaten bei gleicher Leistung auf ein Drittel vermindert werden.

Tabelle 3: Segmentierungsleistung anhand der mittleren IoU bei unterschiedlicher Zusammensetzung der Trainingsdaten (nach [14]).

reale Daten (CamVid)	100 %	-	25 %	33 %	50 %	100 %
synthetische Daten (Spiel)	-	100 %	✓	✓	✓	✓
mittlere IoU	65,0	43,6	63,9	65,2	66,5	68,9

Shafaei et al. [15] kamen zu vergleichbaren Ergebnissen und haben darüber hinaus durch Versuche mit mehreren Datensätzen gezeigt, dass die mit synthetischen Daten trainierten neuronalen Netze eine ähnliche Generalisationsfähigkeit aufweisen wie die mit Realdaten trainierten.

Merkmalsdetektoren und Bildbeschreiber

Da viele höherwertige Bildverarbeitungsalgorithmen mit Merkmalen wie Gradienten oder Histogrammen arbeiten, bilden diese eine Grundlage für den Vergleich zwischen synthetischen und realen Daten. Hummel [5, 20, 26] untersuchte dazu unter anderem, inwiefern sich verschiedene Konfigurationen einer Simulationsumgebung in verschiedenen Szenen auf die Leistung von Merkmalsdetektoren auswirken. Um einen Bezug zwischen Algorithmenleistung und Bildunterschieden herzustellen, wurden dabei sog. Bildbeschreiber verwendet. Einige Einflussfaktoren, die untersucht wurden, sind Schattierungseffekte, Detailgrade der Texturen und der Satellitenaufnahmen, Anti-Aliasing-Filter

und die Modellierung verschiedener Kameraeffekte. Daraus konnten Designvorschläge abgeleitet werden, die die Leistungsunterschiede von Algorithmen minimieren.

4 Konzepte zum Vergleich von Simulation und Realität

In diesem Abschnitt wird ein generelles Konzept zur Untersuchung der Übertragbarkeit von Sensorsimulationen vorgestellt. In Abbildung 1 ist das allgemeine Vorgehen zur Datengenerierung veranschaulicht.

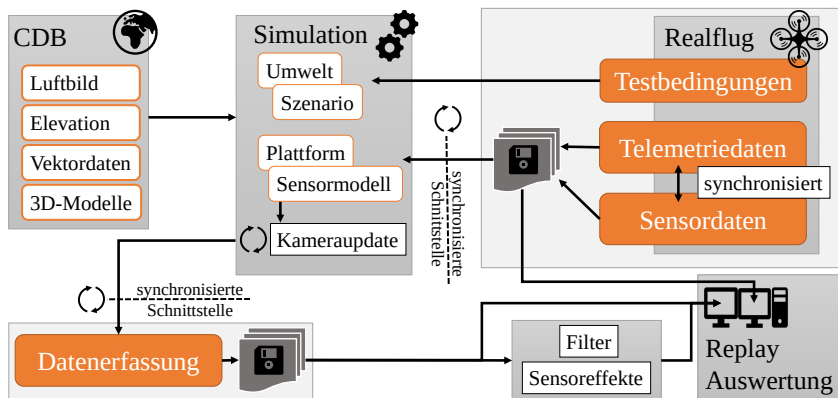


Abbildung 1: Schema zur Datengenerierung in Simulation und Realflug.

Die synchronisierten Telemetrie- und Sensordaten eines Realflugs werden aufgezeichnet und abgespeichert. Zusätzlich werden die Testbedingungen, unter denen der Realflug stattgefunden hat, quantisiert und später zur Konfiguration der Simulationsumgebung genutzt. Dazu zählen z.B. atmosphärische Effekte wie Bewölkungsgrad, Niederschlag, Nebel oder Schnee. Das modellierte Testfluggelände wird on-demand aus der *Common Database* (CDB) der Simulation zur Verfügung gestellt. Für zukünftige Untersuchungen ist der Einsatz der *Presagis* Simulationsumgebung vorgesehen. Diese stellt eine technisch ausgereifte und physikalisch basierte Sensorsimulation mit den benötigten atmosphärischen und sensorspezifischen Effekten bereit und verfügt außerdem über eine komplette Werkzeugkette zur Modellierung. Zu je-

dem diskreten Zeitschritt wird die Position der Sensorplattform und die Ausrichtung der Kamera anhand der aufgezeichneten Telemetriedaten des Realdatensatzes in der Simulation aktualisiert. Anschließend werden die gerenderten Bilddaten abgespeichert und wahlweise mit Filtern bzw. Sensoreffekten überlagert. Das Bildmaterial kann zum visuellen Vergleich und zur Justierung der Simulationsparameter parallel zu den Realflugaufnahmen abgespielt und an die Auswertung übergeben werden. Abbildung 2 zeigt ein auf [5] basierendes Analysekonzept. Dieses wurde durch die zusätzliche Erzeugung eines synthetischen Trainingsdatensatzes auf Verfahren des maschinellen Lernens ausgeweitet und mit optionalen Methoden zur *Domain Adaptation* (DA) ergänzt. DA bezeichnet im Allgemeinen eine Optimierung der Übertragbarkeit von Methoden, die in der Quelldomäne trainiert wurden, auf eine dazu vergleichbare, aber dennoch unterschiedliche Zieldomäne.

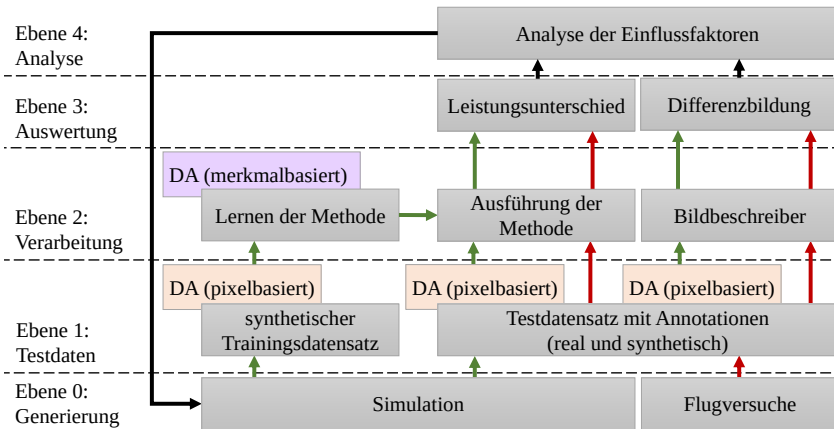


Abbildung 2: Konzept zur Evaluierung von Verfahren des maschinellen Lernens (auf Basis von [5]). Grün: synthetischer Bildfluss, rot: realer Bildfluss, DA: Domain Adaptation, optional eingezeichnet sind die Einsatzmöglichkeiten pixelbasierter oder merkmalsbasierter DA.

Nach Generierung der Bilddaten in Ebene 0 (s. Abbildung 1) werden die simulierten Daten in Ebene 1 in Trainings- und Testdatensatz aufgeteilt. Die Verwendung von ausschließlich synthetischen Trainingsdaten kann zu Modellen mit geringer Fähigkeit zur Generalisierung führen.

Hier besteht die Möglichkeit die Unterschiede zwischen Quelldomäne (=Simulation) und Zieldomäne (=Realität) mit Methoden der *Domain Adaptation* zu verringern und somit den Leistungsverlust der Modelle in der Zieldomäne abzuschwächen. Ein Überblick über das Thema und weiterführende Literaturverweise sind in [18], [13] und [27] zu finden. Bei der in Orange dargestellten pixelbasierten DA liegt der Fokus auf Anpassung der Bilddaten der Quelldomäne an die Daten der Zieldomäne. Um die Vergleichbarkeit aufrechtzuerhalten werden alle synthetischen Datensätze mit identischen Parametern an die Realdaten in der Zieldomäne angeglichen. Erst im nächsten Schritt wird das Modell trainiert und mit Hilfe des realen und synthetischen Testdatensatzes evaluiert. Optional kann eine merkmalsbasierte DA (in Abbildung 2 lila hervorgehoben) verwendet werden, um den Lernprozess zu steuern und das Lernen von domäneninvarianten Merkmalen anzustreben. In diesem Zusammenhang ist auch das Lernen von Merkmalen beider Domänen möglich, wodurch die Fähigkeit zur Generalisierung erhöht werden soll. Beide Ausprägungen versuchen die Leistungscharakteristik zwischen Quell- und Zieldomäne durch Einflussnahme auf die Modellbildung anzugleichen. Außerdem werden in dieser Ebene Bildbeschreiber, wie z.B. die MPEG7 Deskriptoren, verwendet, um parallel dazu die Bildinformationen aus den jeweiligen Datensätzen zu extrahieren. In Ebene 3 wird schließlich die Diskrepanz zwischen synthetischen und realen Testdaten anhand von Leistungsunterschieden der Algorithmen und Differenzen der Bildbeschreiber quantifiziert. Aus diesen Informationen werden mit statistischen Methoden die Einflussfaktoren bestimmt und anschließend das Rendering entsprechend der daraus abgeleiteten Designvorschläge optimiert.

5 Zusammenfassung

Virtuelle Umgebungen sind unter anderem in der Luftfahrttechnik ein wichtiges Hilfsmittel, um schnelle und effektive Entwicklungs- und Testzyklen zu ermöglichen. Die vorgestellten Untersuchungen haben gezeigt, dass mit synthetischen Daten trainierte CV-Algorithmen in vielen Anwendungsfällen ähnliche Leistungscharakteristiken ermöglichen wie mit realen Sensordaten trainierte. Durch Hinzunahme von synthetischen Daten zum realen Trainingsdatensatz, kann eine gesteiger-

te Fähigkeit zur Generalisierung erzielt werden. Die Haupteinflussfaktoren sind dabei Größe, Varianz und Zusammensetzung des annotierten Trainingsdatensatzes. Das vorgestellte Konzept erläutert ein Schema zur Datengenerierung in Simulation und Realflug und liefert eine Vorgehensweise zur Untersuchung der Übertragbarkeit von Sensorsimulationen in die Realität. Diese kann helfen, Einflussparameter auf den *Reality Gap* zu identifizieren und diesen z.B. durch *Domain Adaptation* zu minimieren.

Literatur

1. M. Nentwig, M. Miegler, and M. Stamminger, "Möglichkeiten und Grenzen der simulationsbasierten Validierung von kamerabasierten Fahrerassistenzfunktionen," in *VDI-Berichte 2106*, 15. VDI-Tagung, 2010.
2. M. Nentwig and M. Stamminger, "Hardware-in-the-Loop Testing of Computer Vision Based Driver Assistance Systems," *2011 IEEE Intell. Veh. Symp.*, pp. 339–344, 2011.
3. J. Bossu, D. Gruyer, J. C. Smal, and J. M. Bosseville, "Validation and Benchmarking for Pedestrian Video Detection based on a Sensors Simulation Platform," *IEEE Intell. Veh. Symp. Proc.*, pp. 115–122, 2010.
4. M. Nentwig and M. Stamminger, "A method for the reproduction of vehicle test drives for the simulation based evaluation of image processing algorithms," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, pp. 1307–1312, 2010.
5. G. Hummel, "On synthetic datasets for development of computer vision algorithms in airborne reconnaissance applications," Dissertation, Universität der Bundeswehr München, 2017.
6. C. Parsons, "Improving Automated Aerial Refueling Stereo Vision Pose Estimation Using A Shelled Reference Model," Master Thesis, Air Force Institute of Technology, 2017.
7. F. Orjales, F. Lopez Peña, A. Paz-Lopez, A. Deibe, and R. J. Duro, "On the use of mixed reality for setting up control and coordination strategies for teams of autonomous UAV," *Adv. Intell. Syst. Comput.*, vol. 693, pp. 529–540, 2018.
8. R. Polvara, M. Patacchiola, J. Wan, A. Manning, R. Sutton, and A. Cangelosi, "Toward End-to-End Control for UAV Autonomous Landing via Deep Reinforcement Learning," in *Int. Conf. Unmanned Aircr. Syst.*, 2018.

9. G. Cai, B. M. Chen, T. H. Lee, and M. Dong, "Design and implementation of a hardware-in-the-loop simulation system for small-scale UAV helicopters," *Mechatronics*, vol. 19, no. 7, pp. 1057–1066, 2009.
10. B. Call, R. Beard, C. Taylor, and D. Barber, "Obstacle Avoidance for Unmanned Air Vehicles Using Image Feature Tracking," *AIAA Guid. Navig. Control Conf. Exhib.*, pp. 1–9, 2006.
11. S. Hrabar, "3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs," *2008 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS*, pp. 807–814, 2008.
12. F. Sadeghi and S. Levine, "CAD2RL: Real Single-Image Flight without a Single Real Image," in *Robot. Sci. Syst.* 2017, 2017.
13. J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World," *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2017.
14. S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for Data: Ground Truth from Computer Games," in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2016, vol. 9906 LNCS, pp. 102–118.
15. A. Shafaei, J. J. Little, and M. Schmidt, "Play and Learn: Using Video Games to Train Computer Vision Models," *Br. Mach. Vis. Conf.*, 2016.
16. K. A. Redmill and J. I. Martin, "Virtual Environment Simulation for Image Processing Sensor Evaluation," *Electr. Eng.*, 2000.
17. A. Rozantsev, V. Lepetit, and P. Fua, "On Rendering Synthetic Images for Training an Object Detector," *Comput. Vis. Image Underst.*, vol. 137, no. 0, pp. 24–37, 2015.
18. K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, "Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping," 2017.
19. J. A. Ferwerda, "Three Varieties of Realism in Computer Graphics," in *Proc. SPIE Hum. Vis. Electron.*, B. E. Rogowitz and T. N. Pappas, Eds., vol. SPIE 5007, 2003.
20. G. Hummel and P. Stütz, "Using Virtual Simulation Environments for Development and Qualification of UAV Perceptive Capabilities: Comparison of Real and Rendered Imagery with MPEG7 Image Descriptors," *Model. Simul. Auton. Syst. First Int. Work. MESAS 2014*, vol. 8906, pp. 27–43, 2014.

21. G. Hummel, L. Kovács, P. Stütz, and T. Szirányi, "Data simulation and testing of visual algorithms in synthetic environments for security sensor networks," *Commun. Comput. Inf. Sci.*, vol. 318 CCIS, pp. 212–215, 2012.
22. M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, and R. Vasudevan, "Driving in the Matrix: Can Virtual Worlds Replace Human-Generated Annotations for Real World Tasks?" in *IEEE Int. Conf. Robot. Autom.*, 2017, pp. 746–753.
23. J. M. Kelsey, J. Byrne, M. Cosgrove, S. Seereeram, and R. K. Mehra, "Vision-based relative pose estimation for autonomous rendezvous and docking," *IEEE Aerosp. Conf.*, 2006.
24. M. Müller, N. Smith, and B. Gahem, "A Benchmark and Simulator for UAV Tracking," in *Comput. Vis. - ECCV 2016*, 2016, pp. 445–461.
25. M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4CV: A Photo-Realistic Simulator for Computer Vision Applications," *Int. J. Comput. Vis.*, vol. 126, no. 9, pp. 902–919, 2018.
26. G. Hummel and P. Stütz, "Evaluation of Synthetically Generated Airborne Image Datasets using Feature Detectors as Performance Metric," *Ipcv 2015*, pp. 231–237, 2015.
27. V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual Domain Adaptation: A survey of recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 53–69, 2015.

Virtuelle Inbetriebnahme kamerabasierter Qualitätssicherungssysteme

Virtual commissioning of camera-based quality assurance systems

Nils Piero^{1,2}

¹ Universität Stuttgart, GSaME,
Allmandring 35, 70569 Stuttgart

² Daimler AG, Produktionsplanung Montage,
Benzstraße 10, 71063 Sindelfingen

Zusammenfassung Die variantenreiche Produktion bestimmt zunehmend das Bild der Automobilendmontage, wodurch sich ein anhaltender Wandel in dieser ergibt. Qualitätssicherungssysteme müssen diesen Wandel und eine Einsatzbereitschaft zum Start-Of-Production ermöglichen. Eine frühe Einsatzbereitschaft und eine Wandlungsfähigkeit kann durch die Strategie der Virtuellen Inbetriebnahme erreicht werden. In dieser Arbeit wird eine Methode zur virtuellen Inbetriebnahme von kamerabasierten Qualitätssicherungssystemen vorgestellt und im Anschluss anhand eines Szenarios und der Auswertung definierter Merkmale validiert.

Schlagwörter Qualitätsprüfung, Qualitätssicherung, CAD-basiertes Tracking, Virtuelle Inbetriebnahme.

Abstract The wide range of production variants increasingly determines the picture of final automobile assembly, resulting in a continuous change in this area. Quality assurance systems must enable this change and a readiness for start-of-production. Early operational readiness and adaptability can be achieved through the strategy of virtual commissioning. In this paper, a method for the virtual commissioning of camera-based quality assurance systems is presented and subsequently validated on the basis of a scenario and the evaluation of defined attributes.

Keywords Quality inspection, quality assurance, CAD-based tracking, virtual commissioning.

1 Einleitung

Die Komplexität heutiger Produkte [1] und die Produktion der Varianten dieser mit einer Tendenz zu kundenindividualisierten Einzelprodukten ist eine anhaltende Herausforderung für die Industrie [2]. Aktuelle Anwendungen der Bildverarbeitung zur Qualitätssicherung sind für die Serienfertigung ausgelegt. Eine Inbetriebnahme dieser Anwendungen findet während des Ramp-Up der Produktionslinie statt. Während dieser Inbetriebnahme Phase kann das Qualitätssicherungssystem (QSS) nicht genutzt werden und ausweichende Qualitätssicherungsmethoden müssen parallel eingesetzt werden. Änderungen in der Produktionslinie, am Produkt oder das Hinzufügen von neuen Produkten oder Varianten in die Produktionslinie führt zu erneuten Anpassungen und Systemausfällen. Heutige kamerabasierte QSS bringen nicht die geforderte Flexibilität für den benötigten frühen Einsatz - idealerweise des ersten produzierten Produktes – mit sich. Die Forschungsfrage, welche diese Arbeit beantworten soll lautet:

Wie können kamerabasierte Qualitätssicherungssysteme schon in der Ramp-Up Phase der Produktion eingesetzt werden?

Ziel dieser Arbeit ist es Inbetriebnahmetätigkeiten des bildverarbeitenden QSS vor die Ramp-Up Phase der Produktion, in den virtuellen Raum, zu verlagern.

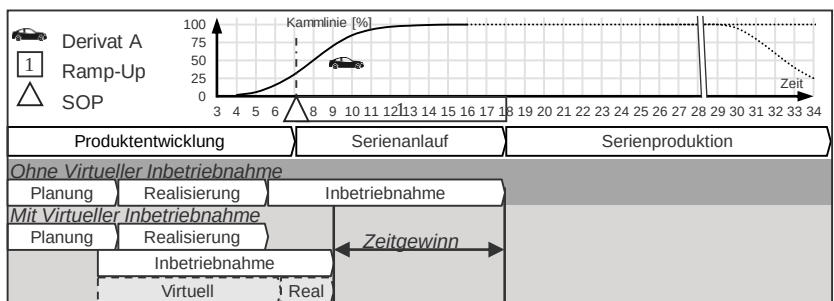


Abbildung 1: Reale und virtuelle Inbetriebnahme in der Hochlauf der Endmontage.

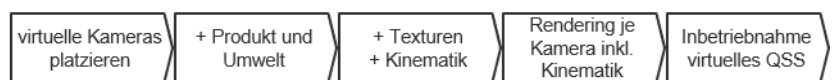


Abbildung 2: Prozess zur Erstellung eines virtuellen kamerabasierten QSS.

2 Methode

Das Ziel der Nutzbarkeit des QSS schon während des Ramp-Up zu erreichen, soll methodisch mit der virtuellen Inbetriebnahme gelingen. Ziel bei der virtuellen Inbetriebnahme ist es, Tätigkeiten der Inbetriebnahme in den virtuellen Raum zu verlagern, wodurch sich der Zeitbedarf in der realen Inbetriebnahme verkürzt (vgl. Abbildung 1). [3]

Für die virtuelle Inbetriebnahme wird ein digitales Planungsmodell des kamerabasierten QSS erstellt und mit Planungsmodellen der unmittelbaren Fabrikumwelt angereichert. Dieses Modell wird im Anschluss mit Textur- und Kinematikeigenschaften erweitert. Die entstandene Szenerie wird über die positionierten Kameras gerendert. Die simulierten Bilder werden für die virtuelle Inbetriebnahme genutzt und unterstützen bei der Optimierung des QSS. Der Prozess ist in Abbildung 2 einsehbar.

3 Szenarien- und Versuchsbeschreibung

Die virtuelle Inbetriebnahme des kamerabasierten QSS wird auf ein Szenario in der Automobilendmontage, die Vorderachsvormontage, angewendet. Die Vorderachsvormontage zeichnet sich durch einen annähernd 100 % manuellen Montagevorgang aus. Zusätzlich ergeben sich eine Vielzahl der möglichen Produktvarianten. Das Ende der Station wird durch eine Qualitätssicherung ausgestattet um **Vorhandensein**, **Variante** und **Position** der montierten Umfänge zu kontrollieren. In einem Versuch soll validiert werden, ob eine virtuelle Inbetriebnahme eines QSS am Ende der Vorderachsvormontage möglich ist. Um den Produktionsbetrieb nicht zu beeinflussen, wurde eine solche Station, entkoppelt von der Produktion, in einem Laborumfeld nachgebildet. Dieses Laborumfeld wurde nach der oben beschriebenen Lösung virtuell nachgebildet. Anschließend wurden je vier Kameras in der virtuellen und in der realen Szenerie angeordnet.

4 Versuchsdurchführung

Die intrinsischen und extrinsischen Kalibrierparameter der Kameras, sowie die Bewegungstrajektorie der zu prüfenden Vorderachse wurden aufgenommen und auf das virtuelle Modell übertragen. Drei Prüfmerkmale wurden ausgewählt um auf Position und Anwesenheit überprüft zu werden. Für die Prüfung wurde die Bildverarbeitungsmethode **CAD-basiertes Tracking** ausgewählt (siehe [4, 5]). In dem virtuellen Modell wird entlang der Bewegungstrajektorie des Produktes alle 13,33 mm ein Bild je Kamera gerendert. Die so entstandenen Bilder werden für die Einrichtung des virtuellen QSS genutzt. Die Parameter des CAD-basierten Trackings wurde an Hand der virtuellen Bilder optimiert. Die gefundenen optimierten Parameter wurden auf das reale System übertragen und das Tracking wurde auf dem realen System wiederholt.

Abbildung 3 zeigt das Tracking von dem Bauteil „Grundplatte“ an dem virtuellen und realen Modell. Die Inlier Ratio (Überdeckung in % von CAD-Kanten mit Modell-Kanten) wird für jedes Frame in Abbildung 5 (links) dargestellt. Analog zur Grundplatte wurde der Integralträger (Abbildung 4, Abbildung 5) getrackt.

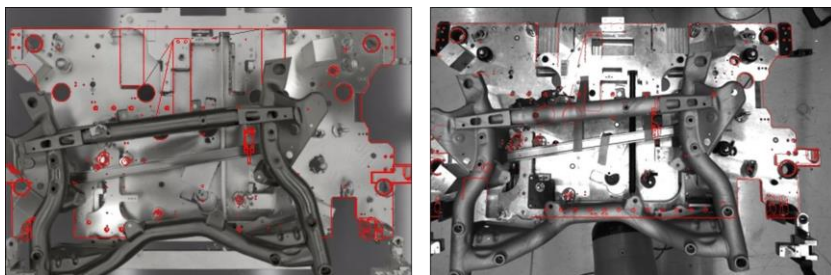


Abbildung 3: Tracking Grundplatte – virtuelles Modell (li), Tracking Grundplatte reales Modell (re).

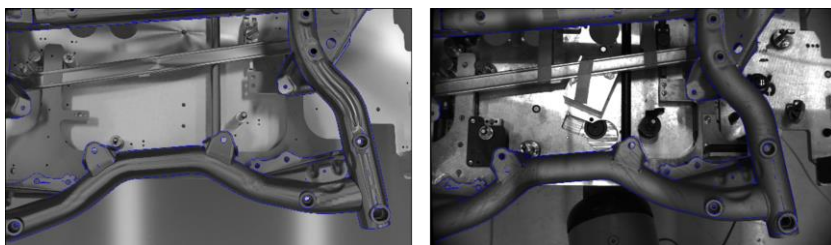


Abbildung 4: Tracking Integralträger – virtuelles Modell (li), Tracking Integralträger reales Modell (re).

5 Ergebnis und Diskussion

Die Ergebnisse zeigen, dass ein Qualitätssicherungssystem mittels CAD-basiertem Tracking über die angeführte Methode virtuell in Betrieb genommen werden kann. Die Optimierung der Trackingparameter im virtuellen Raum lassen sich auf das reale Szenario übertragen. In der Abbildung 5, linkes Diagramm, lässt sich erkennen, dass eine Schwankung der Inlier Ratio im virtuellen Modell zwischen den Frames 1-20 stattfindet. Auch im realen Modell findet eine Schwankung in diesem Framebereich statt. In Abbildung 5, rechtes Diagramm, ist sichtbar, dass sich die Inlier Ratio im virtuellen und im realen Modell geometrisch annähert. Jedoch ergibt sich ein Offset zwischen den Kurven. Dieser Offset ist auch im Bereich 25-125 im linken Diagramm er-

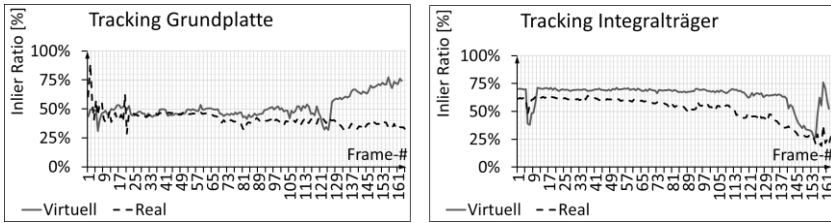


Abbildung 5: Tracking Ergebnis Grundplatte (li), Tracking Ergebnis Integralträger (re).

kennbar. Durch die Optimierung der Trackingparameter im virtuellen Raum und die Übertragbarkeit dieser auf das reale Modell lässt sich die Theorie aufstellen, dass Inbetriebnahmetätigkeiten kamerabasierter QSS vom Realen ins Virtuelle verlagern lassen und sich somit die reale Inbetriebnahmedauer verkürzen lässt. Die gestellte Forschungsfrage lässt sich für das angegebene Szenario also mit der gezeigten Methode beantworten. Grundsätzlich wird darauf hingewiesen, dass das virtuelle Modell eine höhere Inlier Ratio aufweist, was auf die Unterschiede zwischen realem und virtuellem QSS zurückzuführen ist. Zur weiteren Validierung der Methode sollte diese in der realen Produktionsumgebung angewendet werden.

Literatur

1. G. Schuh, *Produktkomplexität managen: Strategien-Methoden-Tools*. Carl Hanser Verlag GmbH Co KG, 2014.
2. T. Bauernhansl, M. ten Hompel, and B. Vogel-Heuser, *Industrie 4.0 in Produktion, Automatisierung und Logistik*. Springer, 2014.
3. Georg Wünsch, *Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme*. Utz, 2008.
4. N. Piero and M. Schmitt, "Virtual Commissioning of Camera-based Quality Assurance Systems for Mixed Model Assembly Lines," *Procedia Manufacturing*, vol. 11, pp. 914–921, 2017.
5. H. Wuest, D. Stricker, and J. Herder, "Tracking of industrial objects by using cad models," *Journal of Virtual Reality and Broadcasting*, vol. 4, no. 1, pp. 1–9, 2007.

Defekterkennung an optischen Funktionsflächen

Defect detection on optical functional surfaces

Marcus Petz, Marc Fischer und Rainer Tutsch

Technische Universität Braunschweig, Institut für Produktionsmesstechnik,
Schleinitzstraße 20, 38106 Braunschweig

Zusammenfassung Die Identifizierung von Oberflächendefekten an optischen Funktionsflächen erfolgt in vielen Fällen noch immer durch visuelle Sichtprüfung. Im vorliegenden Beitrag wird ein neuartiges, kamerabasiertes Messprinzip zur Detektion lokaler Oberflächendefekte vorgestellt, welches sowohl an spiegelnden wie auch an transparenten Oberflächen anwendbar ist und keine geometrische Einmessung der Systemkomponenten erfordert. Im Unterschied zu den verbreiteten Anordnungen mit Dunkelfeldbeleuchtung basiert das Verfahren auf einer Beobachtung im Hellfeld. Hinsichtlich des Anwendungsspektrums des Verfahrens liegt der Schwerpunkt in diesem Beitrag auf der Prüfung von Linsen in Transmission.

Schlagwörter Prüfung optischer Funktionsflächen, Oberflächenunvollkommenheiten, Deflektometrie, Bildverarbeitung.

Abstract The identification of surface imperfections on optical functional surfaces is in many cases still performed by visual inspection. In this contribution a novel, camera-based measuring principle for the detection of local surface defects is presented. This principle is applicable to both specular and transparent surfaces and does not require a geometric calibration of the system components. In contrast to widespread configurations with dark field illumination, the presented approach is based on a bright field arrangement. Regarding the range of applications, the focus of this contribution lies on the testing of lenses in transmission.

Keywords Testing of optical functional surfaces, surface imperfections, deflectometry, image processing.

1 Einleitung

Spiegelnd reflektierende oder transparente Oberflächen sind an einer Vielzahl technischer Erzeugnisse vorzufinden. Ihre produktspezifische Funktion reicht dabei von der optischen Abbildung, wie etwa bei Oberflächen von Spiegeln und Linsen, bis zur Steigerung der Produktästhetik von Konsumgütern. In vielen dieser Anwendungsfälle ist es zur Sicherstellung der Produktqualität erforderlich, auch Oberflächenimperfectionen geringer Ausdehnung zu identifizieren und zu klassifizieren.

Sofern bei dieser Prüfung nicht die exakten geometrischen Abmessungen der Imperfectionen im Vordergrund stehen, sondern deren optische Wirkung oder Wahrnehmbarkeit, erfolgt in der Regel eine visuelle Sichtprüfung. Entsprechende Prüfverfahren sind etwa für optische Funktionsflächen Gegenstand der Normung [1]. Das Grundprinzip der Prüfung besteht in einer manuellen Neigung und Drehung des Prüflings unter einer geeigneten Dunkelfeldbeleuchtung und einer Bewertung anhand der visuellen Wahrnehmung des Prüfers.

Aufgrund der erforderlichen Bewegung des Prüflings sowie wegen des Auftretens direkter Reflexe insbesondere an gekrümmten Oberflächen gestaltet sich die Automatisierung einer entsprechenden Prüfung als technisch anspruchsvoll [2] und ist bislang im industriellen Umfeld in der Regel nicht anzutreffen. Wird die zu prüfende Oberfläche hingegen unter Verwendung einer technisch einfacher realisierbaren Hellfeldbeleuchtung beobachtet, sind Imperfectionen geringer Ausdehnung meist weder für einen menschlichen Betrachter noch für eine elektronische Kamera wahrnehmbar.

Einen Mittelweg zwischen klassischer Dunkel- und Hellfeldanordnung beschreiten Meyer et al. [3] mit dem Ansatz der Lichttransportmatrix. Bei diesem ermöglicht es eine aus vielen einzeln ansteuerbaren Lichtquellen bestehende flächenhafte Beleuchtungseinheit, den von Oberflächenimperfectionen verursachten Streulichtanteil zu detektieren, ohne dass eine Relativbewegung zwischen Prüfling und Beleuchtungseinheit erfolgen muss. Das Wirkprinzip besteht darin, dass an Oberflächenimperfectionen aufgrund der veränderten Streueigenschaften andere und/oder eine größere Anzahl von Einzellichtquellen zur Bildentstehung beitragen, als diese für intakte Oberflächenbereich der Fall ist.

Der im vorliegenden Beitrag beschriebene Ansatz greift in Teilen den Grundgedanken der Lichttransportmatrizen auf, verzichtet jedoch auf die hinsichtlich der Messdauer nachteilige sequentielle Ansteuerung einzelner Orte der flächenhaften Lichtquelle. Basierend auf einer kurzen Sequenz örtlich modulierter Streifenmuster ermöglicht das Verfahren, wie bereits am Beispiel spiegelnder Oberflächen gezeigt werden konnte [4–6], für jedes beobachtete Oberflächenelement die zur Bildentstehung beitragenden Flächenanteile der Lichtquelle und damit das lokale Streuverhalten der zu prüfenden Oberfläche zu charakterisieren.

2 Messprinzip

Das im Folgenden dargestellte Messprinzip zur Identifizierung und Charakterisierung von Oberflächenimperfektionen basiert auf Techniken, die aus dem Umfeld der klassischen Deflektometrie zur geometrischen Messung spiegelnder und transparenter Oberflächen grundsätzlich bekannt sind [7–9]. In Abgrenzung zur klassischen Deflektometrie gilt es bei der hier betrachteten Detektion kurzweiliger Oberflächenimperfektionen jedoch nicht, geometrische Information über die Oberfläche, wie etwa Form, Neigung oder Krümmung, zu ermitteln. Daher gestaltet sich nicht nur die algorithmische Auswertung deutlich einfacher, sondern es kann insbesondere auf die anspruchsvolle und aufwändige geometrische Einmessung der Systemkomponenten verzichtet werden.

Da eine direkte Beobachtung ideal spiegelnder oder transparenter Oberflächen prinzipiell nicht möglich ist, basiert deren deflektometrische Messung darauf, den Prüfling selbst zum Teil eines optisch abbildenden Systems zu machen. Hierdurch kann die Prüflingsoberfläche anhand ihrer optischen Wirkung, also der durch Spiegelung oder Brechung verursachten Strahlablenkung, charakterisiert werden. Hierzu blicken eine oder mehrere elektronische Kameras auf den Prüfling und beobachtet dabei jedoch nicht die Prüflingsoberfläche selbst, sondern ein durch Spiegelung oder Brechung verändertes Abbild einer als bekannt angenommenen Referenzstruktur. Als Referenzstruktur kommen heute meistens Flüssigkristallbildschirme zur Anwendung, deren Anzeigebereich durch Mustersequenzen in zwei Dimensionen kodiert wird.

Primäre Messgröße der klassischen Deflektometrie ist die lokale Neigung der Oberfläche, aus welcher in einer Nachlaufrechnung durch Integration grundsätzlich die absolute Form der Prüflingsoberfläche ermittelt werden kann. Hierzu sind jedoch eine exakte Kenntnis der geometrischen Eigenschaften des Messsystems sowie zusätzliche Informationen oder Annahmen zur Lösung des in der Deflektometrie bestehenden Mehrdeutigkeitsproblems erforderlich. Steht hingegen nicht die absolute Formmessung im Vordergrund, sondern die Detektion unerwünschter Oberflächeneigenschaften wie Welligkeit oder Dellen, wird häufig die lokale Variation der Neigung, also die durch Ableitung der Neigungsdaten bestimmbare Krümmung, ausgewertet. Durch den Verzicht auf eine absolute Formmessung zugunsten einer reinen Krümmungsmessung reduzieren sich die Anforderungen an die geometrische Einmessung der Systemkomponenten in der Regel erheblich.

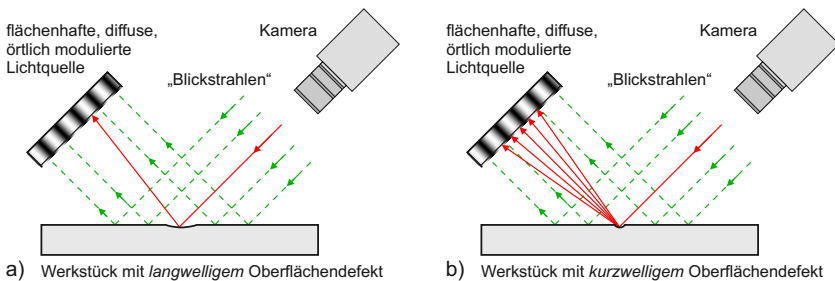


Abbildung 1: a) Prinzip der klassischen deflektometrischen Oberflächencharakterisierung durch Bestimmung der lokalen Neigung. b) Prinzip der neuartigen deflektometrischen Oberflächencharakterisierung durch Bestimmung des lokalen Streuverhaltens.

Wie in Abbildung 1 a) veranschaulicht, liegt der klassischen Deflektometrie die Annahme zugrunde, dass ebenso wie in intakten Oberflächenbereichen auch im Bereich unerwünschter Oberflächenstrukturen die Strahlablenkung stark gerichtet erfolgt. Es ist somit primär die unerwartete oder inhomogene Strahlablenkung, anhand welcher etwaige Oberflächenunvollkommenheiten erkennbar sind.

Im Unterschied hierzu basiert das vorgestellte neuartige Verfahren zur Detektion kleiner Oberflächendefekte auf der in Abbildung 1 b)

dargestellten Strahlauffächerung. Ein vorhandener Oberflächendefekt führt hiernach nicht zu einer gerichteten Ablenkung des Strahls, sondern zu einer Erhöhung des Anteils diffuser Strahlablenkung.

Die Detektion des Ausmaßes der Strahlauffächerung erfolgt bei dem vorgeschlagenen Verfahren gemäß Abbildung 1 b) durch Verwendung eines Flüssigkristallbildschirms als flächenhaft ausgedehnter, diffus abstrahlender Lichtquelle, auf welchem amplitudenmodulierte Muster dargestellt werden. Die im Kamerabild lokal detektierbare Modulation stellt somit ein qualitatives Maß für den Grad gerichteter beziehungsweise diffuser Strahlablenkung dar. Der primäre Messeffekt besteht also darin, dass kurzzeitige Oberflächendefekte als Verminderung der beobachtbaren Modulation in Erscheinung treten.

Das hier aus Gründen der Übersichtlichkeit am Beispiel einer spiegelnden Oberfläche veranschaulichte Messprinzip ist gleichermaßen an transparenten Prüflingen wie etwa Glas- oder Kunststofflinen in Transmission anwendbar. In diesem Fall führen grundsätzlich sowohl auf der Vorder- als auch auf der Rückseite etwaig vorhandene Oberflächenimperfectionen zu einer detektierbaren Strahlauffächerung. Ebenso ist das Verfahren grundsätzlich empfindlich gegenüber Inhomogenitäten im Inneren des Prüflings, wie etwa eingeschlossene Poren oder Fremdkörper.

3 Messdatenerfassung und -auswertung

Wie auch die klassische Deflektometrie nutzt das vorgestellte Verfahren zur Defekterkennung etablierte Kodierungsverfahren auf Basis sinusförmiger Streifenmuster in Verbindung mit Phasenschiebetechniken. Obgleich die hier betrachtete Defekterkennung im Unterschied zur klassischen Deflektometrie keine eindeutige, zweidimensionale Ortskodierung des Bildschirms erfordert, ist die Verwendung zweier orthogonaler Streifenorientierung dennoch zweckmäßig. Sie gewährleistet eine näherungsweise homogene Empfindlichkeit des Verfahrens ungeachtet der jeweiligen Ausdehnungsrichtung eines Defekts. Dies ist insbesondere von Bedeutung für solche Defektarten, die eine Strahlauffächerung hauptsächlich in nur einer Achse bewirken, was etwa auf längliche Kratzer zutrifft, welche das Licht in erster Linie quer zu ihrer Verlaufsrichtung streuen. Um eine derartige einachsige Strahlauffächerung de-

tektieren zu können, muss das Referenzmuster in der entsprechenden Achse eine nennenswerte Modulation aufweisen.

Bei Verwendung sinusförmiger Streifenmuster kann die in einem Punkt des Kamerabildes für das Muster i einer Sequenz beobachtete Intensität I_i kann als Summe eines konstanten und eines modulierten Signalanteils gemäß Gleichung 1 beschrieben werden.

$$I_i = I' + I'' \cos(\phi + \psi_i) \quad (1)$$

Hierin kennzeichnet der Index $i = 1, \dots, M$ die einzelnen Muster einer Sequenz von insgesamt M um den jeweiligen Phasenoffset ψ_i verschobenen Streifenmustern. Die am weitesten verbreitete Klasse von Phasenschiebe-Algorithmen im Bereich der strukturierten Beleuchtung bilden die symmetrischen M -Schritt Algorithmen. Diese zeichnen sich durch gleichmäßig über eine volle Periode verteilte Phasenverschiebungen ψ_i aus. Für $M = 4$ ergibt sich der gebräuchliche 4-Schritt Algorithmus mit den Phasenwinkeln $\psi_1 = 0^\circ$, $\psi_2 = 90^\circ$, $\psi_3 = 180^\circ$ und $\psi_4 = 270^\circ$, welcher auch im vorliegenden Fall zum Einsatz kommt. Da die vorgeschlagene Defekterkennung im Unterschied zur klassischen Deflektometrie keine Ortskodierung erfordert, kann auf ergänzende Kodierungsverfahren zum Zweck der Phasenfaltung verzichtet werden, was sich vorteilhaft auf die Bilderfassungsdauer auswirkt.

Im Fall der vorgeschlagenen Defekterkennung werden anstelle des Phasenwinkels die aus Gleichung 1 zu ersehenden Parameter I' und I'' betrachtet, welche die mittlere Helligkeit und die Modulation der beobachteten Mustersequenz charakterisieren. Um eine von absoluten Intensitätsgrößen unabhängige Darstellung zu erhalten, ist es zweckmäßig, die absoluten Größen I' und I'' durch zwei relative Größen β und γ gemäß Gleichungen 2 und 3 zu substituieren.

$$I' = \beta I_{Sat} \quad (2)$$

$$I'' = \gamma I' \quad (3)$$

Hierin steht I_{Sat} für die kameraabhängige Sättigungsgrenze eines Bildpunktes, β für die im Weiteren als Belichtung bezeichnete relative mittlere Intensität und γ für den im Weiteren als Sichtbarkeit bezeichneten relativen Streifenkontrast. Mit dieser Substitution kann die in absoluten Größen formulierte Gleichung 1 gemäß nachfolgender Gleichung 4

ausgedrückt werden.

$$I_i = I_{Sat}\beta(1 + \gamma \cos(\phi + \psi_i)) \quad (4)$$

Die Parameter β und γ beschreiben folglich die Helligkeits- und Kontrastverhältnisse in einem Bildpunkt unabhängig von der verwendeten Kamera. Eine Veranschaulichung des Erscheinungsbildes sinusförmiger Streifenmuster in dem durch die Größen β und γ aufgespannten Parameterraum ist Abbildung 2 zu entnehmen. Die relativen Parameter β und γ weisen beide einen Wertebereich von 0 bis 1 auf, wobei die Nebenbedingung $\beta \leq 1/(1 + \gamma)$ gilt, da ansonsten die absolute Intensität I_i im Bildpunkt höher als die Sättigungsintensität I_{Sat} wäre.

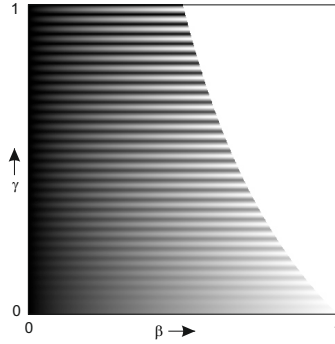


Abbildung 2: Kontrastverhältnisse eines sinusförmigen Streifenmusters in Abhängigkeit von Belichtung β und Sichtbarkeit γ [10].

Für eine vollständige Messsequenz nach dem vorgeschlagenen Ansatz zur Defekterkennung werden jeweils vier phasenverschobene Sinusmuster in zwei zueinander orthogonalen Orientierungen aufgezeichnet. Um die effektiv nutzbare Empfindlichkeit des Verfahrens zu steigern, ist es jedoch empfehlenswert, den Einfluss des Bildrauschens durch Mittelung mehrerer Einzelbelichtungen jedes der acht registrierten Zustände zu reduzieren.

Aus den für jeden auswertbaren Bildpunkt pro Streifenorientierung aufgezeichneten Intensitäten I_1 bis I_4 werden gemäß Gleichungen 5 und 6 die Größen Belichtung β und Sichtbarkeit γ berechnet, zunächst getrennt für die beiden im Weiteren mit x und y bezeichneten Kodie-

rungsrichtungen.

$$\beta = \frac{I_1 + I_2 + I_3 + I_4}{4 \cdot I_{Sat}} \quad (5)$$

$$\gamma = \frac{\sqrt{(I_1 - I_3)^2 + (I_2 - I_4)^2}}{2 \cdot \beta \cdot I_{Sat}} \quad (6)$$

Werden die so für beide Orientierungen x und y separat ermittelten Beleuchtungen β_x und β_y sowie Sichtbarkeiten γ_x und γ_y als Komponenten von Vektoren $\vec{\beta}$ und $\vec{\gamma}$ interpretiert, bietet sich die Bestimmung richtungsunabhängiger Größen in Gestalt der Vektorbeträge $\beta_{abs} = |\vec{\beta}|$ und $\gamma_{abs} = |\vec{\gamma}|$ an. Im Unterschied zu den Einzelgrößen weisen die so ermittelten kombinierten Größen folglich einen Wertebereich im Intervall $[0, \sqrt{2}]$ auf.

Die wie beschrieben ermittelten Größen β_{abs} und γ_{abs} sind gemäß dem postulierten Messeffekt abhängig von den lokalen Streueigenschaften der untersuchten Oberfläche. Darüber hinaus unterliegen die Größen jedoch auch dem Einfluss von Systemgeometrie, Form des Prüflings und Abstrahleigenschaften des Flüssigkristallbildschirms. Es ist jedoch festzustellen, dass die gesuchten defektbedingten Änderungen von β_{abs} und γ_{abs} sich durch deutliche höhere Ortsfrequenzen von den durch den Aufbau verursachten, niederfrequenten Variationen abheben. Es wird daher eine Hochpassfilterung der Daten durchgeführt, welche sich beispielsweise durch Differenzbildung mit einer durch Medianfilterung geglätteten Kopie des Originaldatensatzes realisieren lässt. In den Differenzsignalen $\Delta\beta_{abs}$ und $\Delta\gamma_{abs}$ zeigen sich Oberflächenimperfectionen wie Kratzer, Schrammen, Löcher oder Verunreinigungen als signifikante Abweichungen von Null.

4 Messaufbau

Der für die im Weiteren vorgestellten Messungen an Linsen in Transmission verwendete Messaufbau ist in Abbildung 3 dargestellt. Der Aufbau verfügt über eine Kamera vom Typ IDS μ Eye UI-5240SE-M mit CMOS-Sensor und einer Auflösung von 1280×1024 Pixel. Die Kamera ist im dargestellten System mit einem Objektiv vom Typ Kowa LM16XC mit einer Brennweite von $f = 16$ mm ausgestattet, welches für die Messung

hier auf einen Blendenwert von $f/5,6$ abgeblendet wurde. Der zur Musterdarstellung verwendete Bildschirm vom Typ Samsung S24E650BW verfügt über ein PLS-Panel mit einer Auflösung von 1920×1200 Pixel. Im Rahmen vorangegangener Untersuchungen zur Messung ebener und sphärischer spiegelnder Oberflächen aufgebaute Konfigurationen nutzten mit Ausnahme des Objektivs dieselben Komponenten, lediglich in veränderter räumlicher Anordnung [4, 5].

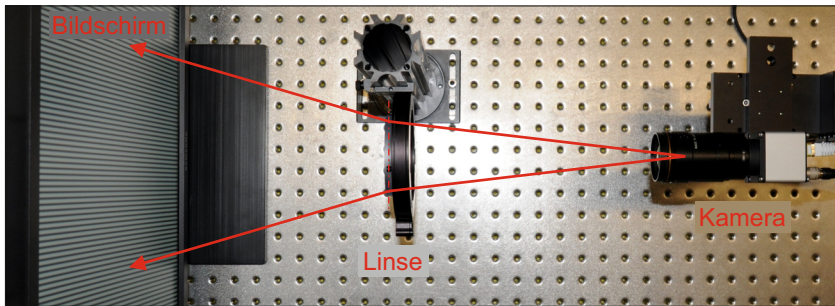


Abbildung 3: Messaufbau zur Defekterkennung an Linsen.

Die Komponenten des Systems sind auf einem Leicht-Breadboard angeordnet. Während Kamera und Prüfling mittels eines Profilsystems fest aufgespannt sind, ist der Bildschirm lediglich mittels seines normalen Standfußes auf das Breadboard gestellt. Als Linsenhalter kommt ein Thorlabs SCL04 zum Einsatz. Position und Orientierung der Komponenten sowie die Einstellung des Objektivs können flexibel je nach Beschaffenheit des Prüfobjekts verändert werden, da das vorgeschlagene Messprinzip keinerlei geometrische Einmessung der Systemkomponenten erfordert.

5 Ergebnisse

Im Rahmen vorangegangener Untersuchungen an spiegelnden Oberflächen konnte gezeigt werden, dass das beschriebene Verfahren anhand einer signifikanten Verminderung der lokalen Sichtbarkeit γ Defekte identifizieren kann, deren Strukturbreite deutlich unterhalb des Auflösungsvermögens des zur Beobachtung eingesetzten Kamerasys-

tems liegen [4]. Ebenfalls konnte festgestellt werden, dass unterschiedliche Defektarten mit charakteristischen Änderungen von Belichtung und Sichtbarkeit korrelieren, wodurch die Möglichkeit einer automatischen Defektklassifikation eröffnet wird [4]. Durch Vergleich der in β und γ beobachteten Abweichungen von Null mit dem Rauschniveau der jeweiligen Messdaten sowie durch Berücksichtigung der lateralen Ausdehnung können zudem Aussagen zur Signifikanz der Beobachtung sowie zur Ausprägung des Defekts getroffen werden [5].

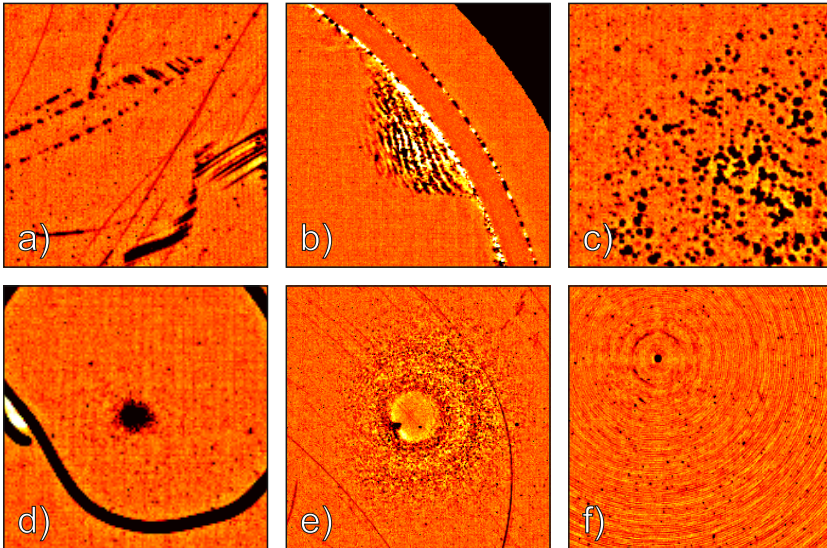


Abbildung 4: Differenzsignal $\Delta\gamma_{abs}$ für exemplarische Defekttypen an Linsen. a) Kratzer und Schrammen, b) Verunreinigung im Randbereich durch Fingerabdruck, c) Beschichtungsfehler, d) punktförmige Krümmungsabweichung im Zentrum¹, e) zentrale Eintrübung durch Werkzeugeinfluss, f) Drehspuren.

Für den Anwendungsfall der Messung in Transmission wurde eine Reihe von Linsen untersucht, deren Rohform durch zerspanende Bearbeitung erzeugt wurde und die anschließend durch Polieren und Beschichten finalisiert wurden. Eine exemplarische Auswahl der da-

¹ Der umgebende dunkle Ring in Teilbild d) wird von einer mit Filzschreiber auf die Linse aufgetragenen Markierung verursacht.

bei anhand des Differenzsignals der Sichtbarkeit $\Delta\gamma_{abs}$ nachgewiesenen Defektarten gibt Abbildung 4. Die Teilbilder zeigen im einzelnen a) Kratzer und Schrammen auf der Linsenoberfläche, b) Verunreinigung durch einen Fingerabdruck im Randbereich der Linse, c) Fehlstellen in der aufgetragenen Beschichtung der Linse, d) punktförmige Krümmungsabweichung im Zentrum der Linse durch Fehler bei der spanenden Bearbeitung, e) Aufrauung und dadurch Eintrübung im Zentrum der Linse durch Werkzeugverschleiß und f) konzentrische Drehspuren auf der gesamten Linsenoberfläche. Eine eindeutige Zuordnung der Defekte zur Vorder- oder Rückseite der Linse ist in der genutzten Konfiguration aufgrund der gegebenen Schärfentiefe der Abbildung nicht möglich. Erweiterungen des Verfahrens zur Überwindung dieser Einschränkung beispielsweise durch Multilateration oder Variation der Fokuslage erscheinen jedoch grundsätzlich möglich und erfolgversprechend.

6 Zusammenfassung

Das im vorliegenden Beitrag vorgestellte Messprinzip eröffnet die Möglichkeit, basierend auf deflektometrischen Grundtechniken mit geringem apparativem Aufwand und bei flexibler Systemgeometrie Spiegel und Linsen sowie vergleichbare Oberflächen anderer Produktgruppen mit hoher Empfindlichkeit auf verschiedene kurzwellige Oberflächenimperfectionen zu prüfen. Das in Hellfeldanordnung wirkende Verfahren ist grundsätzlich in Reflexion und in Transmission sowie auf Oberflächen unterschiedlicher Krümmung anwendbar. Das Messprinzip ermöglicht es, Defekte zu detektieren, deren Abmessungen deutlich unterhalb des optischen Auflösungsvermögens des Aufzeichnungssystems liegen. Zugleich zeigt das Verfahren bei Vergleich mit visuellen und mikroskopischen Prüfungen eine hohe Signifikanz.

Danksagung

Diese Arbeit basiert in Teilen auf einem vorherigen Forschungsprojekt (Pe1402/2-2), welches von der Deutschen Forschungsgemeinschaft DFG gefördert wurde. Die Autoren bedanken sich für die Förderung.

Literatur

1. DIN ISO 14997:2018-05 - *Optik und Photonik - Prüfverfahren für Oberflächenunvollkommenheiten optischer Elemente (ISO 14997:2017)*, Deutsches Institut für Normung e.V., 2018.
2. A. Schöch, P. Perez, S. Linz-Dittrich, C. Bach, and C. Ziolk, "Automated surface inspection of small customer-specific optical elements," *tm - Technisches Messen*, vol. 84, no. 7-8, pp. 502–511, 2017.
3. J. Meyer, T. Längle, and J. Beyerer, "Erfassung und Verarbeitung von Lichttransportmatrizen zur automatischen Sichtprüfung transparenter Objekte," in *Forum Bildverarbeitung 2016*, M. Heizmann, T. Längle, and F. Puente León, Eds. Karlsruhe: KIT Scientific Publishing, 2016, pp. 75–86.
4. M. Petz, M. Fischer, and R. Tutsch, "Defekterkennung an spiegelnden und transparenten Oberflächen durch Abbildung einer örtlich modulierbaren Lichtquelle," *tm - Technisches Messen*, vol. 85, no. 2, pp. 79–87, 2018.
5. —, "Optische Inspektion spiegelnder und transparenter Oberflächen," in *VDI-Berichte 2326, 6. VDI-Fachtagung „Optische Messung von Funktionsflächen 2018“*, VDI Wissensforum GmbH, Ed. Düsseldorf: VDI, 2018, pp. 107–118.
6. —, "Defekterkennung an spiegelnden und transparenten Oberflächen durch Abbildung einer flächenhaft modulierbaren Lichtquelle," in *DGaO (Deutsche Gesellschaft für angewandte Optik), 118. Jahrestagung (Tagungsband)*, 2017, p. 46.
7. M. Knauer, "Absolute phasenmessende Deflektometrie," Dissertation, Friedrich-Alexander Universität Nürnberg-Erlangen, 2006.
8. M. Fischer, "Deflektometrie in Transmission – Ein neues Messverfahren zur Erfassung der Geometrie asphärischer refraktiver Optiken," Dissertation, Technische Universität Braunschweig, 2016, Schriftenreihe des Instituts für Produktionsmesstechnik, Band 12, Aachen: Shaker.
9. M. Petz, "Rasterreflexions-Photogrammetrie – Ein neues Verfahren zur geometrischen Messung spiegelnder Oberflächen," Dissertation, Technische Universität Braunschweig, 2006, Schriftenreihe des Instituts für Produktionsmesstechnik, Band 1, Aachen: Shaker.
10. M. Fischer, M. Petz, and R. Tutsch, "Modellbasierte Rauschvorhersage für Streifenprojektionssysteme – Ein Werkzeug zur statistischen Analyse von Auswertalgorithmen," *tm - Technisches Messen*, vol. 84, no. 2, pp. 111–122, 2017.

Komplexitätsbestimmung von Datensätzen als Grundlage für die Wahl eines geeigneten Klassifikationsverfahrens

Complexity determination of datasets as basis for the choice of a suitable classifier

Katharina Anding¹, Lilli Haar¹, Konstantin Trambitckii¹, Daniel Garten² und Gunther Notni¹

¹ Technische Universität Ilmenau, Fachgebiet Qualitätssicherung und Industrielle Bildverarbeitung,
Gustav-Kirchhoff-Platz 2, 98693 Ilmenau

² GFE Schmalkalden e.V.,
Näherstiller Straße 10, 98574 Schmalkalden

Zusammenfassung Bei der Lösung einer Bildverarbeitungsaufgabe mit Hilfe maschineller Lernverfahren ist neben der optimalen Segmentierung und der Berechnung relevanter Merkmale die Wahl eines geeigneten Klassifikationsalgorithmus ausschlaggebend für das Erreichen hoher Erkennungsleistungen. Es erweist sich jedoch nicht als einfach, eine derartige Wahl zu treffen. Sie muss sich nach der Struktur und den Eigenschaften des vorliegenden Datensatzes bzw. Merkmalsraumes richten. Datensätze mit einfach zu trennenden Klassen bedürfen keiner komplexen Klassifikatoren, welche einen hohen Zeit- und Rechenaufwand erfordern. Bei derartigen Datensätzen lassen sich bereits mit einfachen Klassifikationsalgorithmen sehr gute Erkennungsleistungen erzielen. Demnach muss zunächst die Komplexität des Datensatzes selbst abgeschätzt werden. Auf der Grundlage der Charakteristik des gegebenen Merkmalsraumes kann danach entschieden werden, ob ein einfacher oder ein komplexer Klassifikator verwendet wird. Für die Komplexitätsabschätzung eines Datensatzes werden ausgewählte Verfahren vorgestellt.

Schlagwörter Überwachtes Lernen, unüberwachtes Lernen, Klassifikatorwahl.

Abstract The proper solution of an image processing task, using machine learning, is dependent of the choice of an optimal segmentation, a calculation of relevant features and especially a suitable classification algorithm. Such a choice is not easy to perform. It has to be based on the structure and the properties of the existing dataset or feature space. Datasets with classes that are easy to separate do not require complex classifiers, which require a great amount of computational time. By using such datasets very good recognition performance can be achieved even with simple classification algorithms. Accordingly, first the complexity of a dataset has to be estimated. Based on the characteristic of the feature space, it is possible to decide whether a simple or a complex classifier should be used. Different methods are presented, which can estimate the complexity of a dataset.

Keywords Supervised learning, unsupervised learning, choice of classifier.

1 Einleitung

Die Auswahl eines geeigneten Klassifikationsverfahrens erweist sich oftmals als nicht trivial. Hierbei müssen die Struktur des Datensatzes, die Charakteristik des Merkmalsraumes, die Voraussetzungen der Aufgabenstellung und die benötigte Rechenzeit, welche für die Klassifikation zur Verfügung steht, bedacht werden [1], [2]. In der Literatur werden unterschiedliche Vorgehensweisen und Verfahren, welche die Klassifikatorwahl unterstützen, beschrieben. Diese reichen vom Testen mehrerer Klassifikatoren [3], über Cheat-Sheets [4], [5], [6] bis hin zu Systemen, die diese Wahl gänzlich automatisieren sollen [7], [8], [9], [10], [11].

Allen Ansätzen ist gleich, dass der Klassifikator auf Grundlage der vorliegenden Datensätze gewählt wird. Die Charakteristik des gegebenen Merkmalsraumes spielt hierbei eine ausschlaggebende Rolle, in [2] werden erste Ansätze zur Ermittlung geeigneter Klassifikatoren allein auf Basis der Analyse des entstehenden Merkmalsraumes beschrieben.

Es existieren verschiedene Methoden zur Analyse des Merkmalsraumes. Grundsätzlich spricht man hier von sogenannten Merkmalsselektionsverfahren, weil im Ergebnis des analysierten Merkmalsraumes eine Auswahl der relevanten, diskriminanzfähigen Merkmale erfolgen soll. Zur Erzielung einer hohen Erkennungsleistung, ist die Verwendung

von Merkmalen, welche eine Separation der Klassen ermöglichen von essentieller Bedeutung. Auch bei einer zu geringen Menge eingesetzter Merkmale sind keine zufriedenstellenden Ergebnisse erreichbar. Durch eine Hinzunahme weiterer Merkmale steigt die Erkennungsleistung an, bis der Informationsgehalt ausgeschöpft ist und eine Erweiterung des Merkmalsvektors zu einer Stagnation ggf. sogar zu einem Rückgang der Erkennungsleistung führen würde (als Peaking-Phänomen bezeichnet) [12]. Zudem unterliegt ein Merkmalsraum mit einer sehr hohen Anzahl an Merkmalen in Relation zu einer zu kleinen Anzahl an Datensatzobjekten dem sog. curse of dimensionality [13], welcher die Erkennungsleistung ebenfalls negativ beeinflussen kann.

Ziel dieser Untersuchungen war es, die Charakteristik des gegebenen Merkmalsraumes hinsichtlich Redundanzen und Irrelevanzen zu analysieren und auf dieser Basis abzuschätzen, ob ein einfacher oder ein komplexer Klassifikator verwendet werden sollte.

2 Stand der Technik

Die **Hauptkomponentenanalyse** (englisch: **Principal Component Analysis**, kurz: **PCA**) ist eine lineare Transformation, welche sich zur Datenreduktion bzw. Datenkompression und ebenfalls hervorragend zur Visualisierung hochdimensionaler Daten im entstehenden 3-dimensionalen Merkmalsraum der ersten drei Hauptkomponenten (kurz HK; englisch: Principal Components, kurz: PC) mit größter Summenvarianz eignet. Unter Verwendung einer PCA wird das vorliegende Koordinatensystem derart transformiert, dass es aufgrund einer Orientierung in Richtung der größten Varianz besser geeignet ist, die gegebenen Daten zu beschreiben [14].

Eine erfolgversprechende Methode zur Beurteilung des Merkmalsraums bietet auch die Analyse von Gruppierungstendenzen des **Clusterings**. Das Ziel der Cluster-Bildung ist es, Objekte mit ähnlichen Merkmalen in Gruppen, sog. **Cluster**, zu teilen, wobei keine Klassenzugehörigkeiten bekannt sind. Diese Einordnung erfolgt nach dem Grundsatz, dass Objekte innerhalb eines Clusters in starkem Maße Ähnlichkeiten aufweisen, während sie besonders unähnlich zu den Objekten anderer Cluster sein sollen (Homogenität innerhalb der Cluster und Heterogenität zwischen den Clustern) [15], [16]. Beim Clustering-

Prozess können fehlerbehaftete bzw. irrelevante Merkmale, fehlerbehaftete Objekte oder auch Ausreißer zu unbefriedigenden Ergebnissen führen.

Das Ziel des **Expectation-Maximization-Algorithmus** (kurz: **EM-Algorithmus**) ist es, jene Modellparameter der Wahrscheinlichkeitsverteilungen zu finden, welche die vorliegenden Daten am besten repräsentieren [17], und somit die Anpassung des mathematischen Modells an diese Daten zu optimieren [15]. Der EM-Algorithmus ist aufgrund seiner einfachen Implementierung und Flexibilität [15] sehr beliebt. Probabilistische Verfahren übertreffen häufig andere Clustering-Methoden und können in vielen Bereichen eingesetzt werden [17].

Im Rahmen der Untersuchungen wurden ausgewählte Klassifikatoren, welche eine möglichst einfache Parametrierung aufweisen, verwendet, um den Einfluss der Klassifikatorparameter zu verringern. Als einer der leistungsfähigsten Klassifikatoren wurde der **Random-Forest-Klassifikator** verwendet, welcher aus einer Vielzahl an Entscheidungsbäumen, einem Entscheidungswald besteht, und damit zu den Ensemble-Methoden zählt [18]. Er erzielt auch auf sehr komplexen, hochgradig nichtlinearen Erkennungsaufgaben in kurzer Zeit sehr gute Ergebnisse. Der verwendete **J48-Klassifikator** ist eine Implementierung einer leicht verbesserten Version des C4.5-Algorithmus in Weka, welcher zu den Entscheidungsbaumverfahren gehört [19]. Der **k-Nearest Neighbor** (deutsch: *k*-nächste-Nachbarn-Klassifikator, kurz: *k*NN) bedient sich verschiedener Abstandsmaße (z. B.: euklidischer Abstand) im Merkmalsraum zur Ermittlung der Klassenzugehörigkeit. Der **Naive-Bayes-Klassifikator** gehört zur Gruppe der statistischen bzw. wahrscheinlichkeitsbasierten Klassifikatoren.

3 Vorgehen

Im Rahmen dieser Untersuchungen lag der Fokus auf der Beurteilung der mit dem Datensatz und der erfolgten Merkmalsextraktion gegebenen Merkmalsräume und der Beschreibung der vorhandenen Komplexität unter Ableitung von empirischen Regeln zur Klassifikatorauswahl.

Bei Feststellung einer gegebenen hohen Komplexität des Merkmalsraums mit großer Redundanz der gegebenen Merkmale bei gleichzeitig relativ geringer Relevanz der Merkmale zur Klassentrennung, ist neben

der Verwendung eines komplexen nichtlinearen Klassifikators, um das Erkennungsproblem noch hinreichend gut zu lösen, eine andere Option die Erweiterung des Datensatzes um die Klassenvarianz wiedergebende Datensatzbeispiele sowie eine Erweiterung der Merkmalsextraktion um weitere signifikante Merkmale mit hoher Diskriminanzfähigkeit.

Datensätze mit Klassen, welche sich im Merkmalsraum nicht bzw. nur geringfügig überlappen und somit einfach voneinander abgegrenzt werden können, weisen eine einfache Struktur auf. Einfache Klassifikationsverfahren stehen beim Vorliegen derartiger Datensätze den komplexen Algorithmen in ihrer Leistung nicht nach und sind in der Lage, sehr gute Ergebnisse zu liefern. Diese benötigen im Allgemeinen weniger Rechenzeit und Rechenaufwand. Zudem muss im Vorfeld keine aufwändige Parameteroptimierung erfolgen.

Die grundlegende Annahme ist die, dass eine hohe Komplexität des Merkmalsraums begründet ist in einer großen Redundanz und geringen Relevanz der gegebenen Merkmale. Zur Beurteilung der Redundanzen können z. B. Ergebnisse der Hauptkomponentenanalyse betrachtet werden. Da die PCA die Kovarianz des gesamten Datensatzes untersucht, ermöglicht sie eine Beurteilung der Korrelation der Merkmale im gegebenen Datensatz und damit der Redundanzen sowie ggf. eine Datenreduktion, jedoch keine Aussage zur Klassenzugehörigkeit bzw. Trennbarkeit des Datensatzes. Die ersten HK beinhalten den größten Teil der Informationen, weshalb die Anzahl der benötigten HK zum Erreichen einer Summenvarianz von 99% im Kontext zur Gesamtzahl der HK bzw. Ursprungsdimensionen des Merkmalsraums Rückschlüsse auf Redundanzen eines Datensatzes erlaubt.

Eine andere Möglichkeit ist die Anwendung des unüberwachten Maschinellen Lernens (Clustering), bei dem auf der Basis mathematischer Gesetzmäßigkeiten im Merkmalsraum natürliche Gruppen gebildet werden, um Informationen zur Struktur des Merkmalsraumes und somit zur Komplexität der Erkennungsaufgabe zu gewinnen. Hintergrund ist hier, dass beim Clustering-Prozess fehlerbehaftete bzw. irrelevante Merkmale, fehlerbehaftete Objekte oder auch Ausreißer zu unbefriedigenden Ergebnissen führen können [16]. Dieser Umstand wurde im Rahmen dieser Untersuchungen genutzt, um auf Basis der bekannten Objektcluster (Klassenzugehörigkeit als Apriori-Wissen) und den mittels unüberwachten Clusteringverfahren gefundenen Clustern Rückschlüsse auf die Komplexität vorliegender Datensätze sowie die

Relevanz des Informationsgehalts im Merkmalsraum ziehen zu können. Es werden Ergebnisse der Komplexitätsbestimmung von Datensätzen als Grundlage für die Wahl eines geeigneten Klassifikationsverfahrens unter Verwendung verschieden komplexer Bilddatensätze aus der industriellen Praxis aufgezeigt und Empfehlungen abgeleitet.

4 Charakteristik verwendeter realer Datensätze

Für die nachfolgenden Untersuchungen wurden 11 reale Datensätze numerischer Art aus der Praxis mit unterschiedlichen Eigenschaften verwendet, welche in Tabelle 1 detaillierter dargestellt sind.

Tabelle 1: Übersicht reale verwendete Datensätze.

Datensatz	Anzahl Objekte	Anzahl Merkmale	Anzahl Klassen
Iris	150	4	3
Wine	178	13	3
Streulicht_DS1_3Kl	300	182	3
Streulicht_DS1_2Kl	300	182	2
Streulicht_DS2_4Kl	719	182	4
Streulicht_DS2_2Kl	719	182	2
Streulicht_DS3_4Kl	1193	182	4
Streulicht_DS3_2Kl	1193	182	2
Autopetrographie	18596	234	4
Gestein	11493	199	8
Metalloberflächen	273	123	3

5 Ergebnisse der Untersuchungen

Da die erzielbare Erkennungsleistung eines Klassifikators stets von der Relevanz und Redundanz der Merkmale im Merkmalsraum abhängt, wurde zunächst zur Quantifizierung der Redundanz im Merkmalsraum auf Basis der PCA das Redundanzmaß R definiert und auf Basis realer Datensätze untersucht:

$$R [\%] = 100\% - \frac{n_{PC_{99\%}}}{n_{PC_{all}}}, \text{ wobei } n_{PC_{99\%}} \text{ die Anzahl der zum Errei-}$$

chen einer Summenvarianz von 99% benötigten HK ist, welche bei der Berechnung der PCA ermittelt wurden, und $n_{PC_{all}}$ der Gesamtanzahl

der HK und damit der Dimension des Ursprungsmerkmalsraums entspricht. Die Ergebnisse der Berechnung des Redundanzmaßes für die realen Datensätze sind in Abbildung 1 dargestellt.

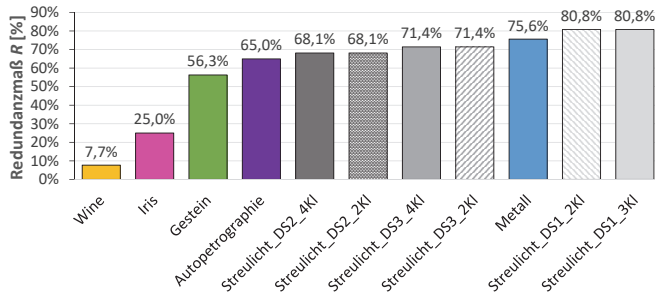


Abbildung 1: Redundanzanalyse auf Basis der PCA.

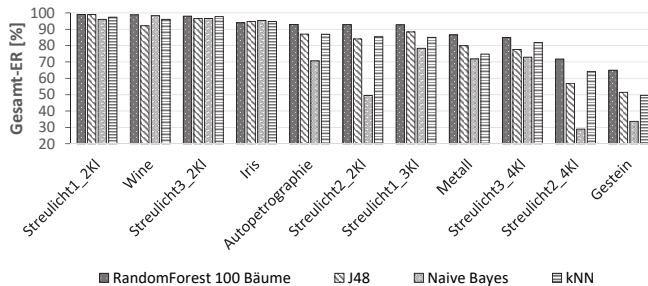


Abbildung 2: Gesamterkennungsraten ausgewählter Klassifikatoren für reale Datensätze.

Eine Beurteilung der erhaltenen Werte für R wurde empirisch auf Basis der untersuchten Datensätze durchgeführt. Die empirisch bewerteten Werte für R sollen zukünftig unter Verwendung weiterer realer Datensätze aus der Praxis noch verfeinert werden. Man kann jedoch davon ausgehen, dass ein Wert mit $R \leq 25\%$ für Merkmalsräume mit geringen Redundanzen steht. Des Weiteren kann man beim gleichzeitigen Vorhandensein einer ausreichenden Anzahl an relevanten Merkmalen dann auch von einer Anwendbarkeit selbst Redundanz-empfindlicher

einfacher Klassifikatoren, wie dem Naive-Bayes, ausgehen. In Abbildung 2, die die Gesamtübersicht der Erkennungsraten der 4 ausgewählten Klassifikatoren enthält, wird deutlich, dass der Klassifikator Naive-Bayes insbesondere für die beiden Datensätze Wine und Iris, deren Merkmale niedrige Redundanz aufweisen, hervorragende Ergebnisse liefert (auf dem Iris-Datensatz sogar die höchste erreichbare Erkennungsrate im Vergleich zu den anderen Klassifikatoren), für alle anderen Datensätze mit großen Redundanzen $R > 25\%$ jedoch stets am schlechtesten von allen untersuchten Klassifikatoren abschneidet. Die starke Empfindlichkeit des Naive-Bayes lässt sich durch die Annahme erklären, dass die vorliegenden Merkmale unabhängig seien, welche in Bezug auf reale Datensätze i.d.R. zu optimistisch ist [15], [19]. Wird z. B. ein bereits vorhandenes Merkmal dem Merkmalsvektor nochmals hinzugefügt, vervielfacht sich der Einfluss dieses einen, im ungünstigen Fall auch noch unwichtigen, Merkmals, da die bedingte Wahrscheinlichkeit quadratisch in die Ermittlung des Ergebnisses eingeht. Redundante Merkmale tragen damit zur Entscheidung deutlich mehr bei als andere. Die Erkennungsleistung mit einfachen Redundanz-empfindlichen Klassifikatoren kann man verbessern, indem man vor der Klassifikation eine Merkmalsselektion, welche vorhandene Korrelationen erkennt und den Merkmalsvektor entsprechend einkürzt, durchführt. Alternativ kann man auch Klassifikatoren, die eine höhere Robustheit gegenüber redundanten Merkmalen aufweisen, wie z. B. Random-Forest, verwenden. Dieser zeigt auf den untersuchten Datensätzen mit großen Redundanzen die beste Erkennungsleistung im direkten Vergleich. Random-Forest ist als sehr leistungsstarker Klassifikator bekannt, weil er insbesondere mit verrauschten Daten gut umgehen kann und zudem robust gegenüber Ausreißern ist [18]. Der Hintergrund des geringen Einflusses irrelevanter oder redundanter Merkmale liegt hier im Aufbau der Bäume und einer internen Selektion der wichtigsten Merkmale begründet.

Eine qualitative nicht numerische Beurteilung der Komplexität des Merkmalsraums kann auch auf Basis der Visualisierung der Ergebnisse der PCA in den ersten 3 HK unter gleichzeitiger Nutzung und Darstellung des Apriori-Wissens zur Klassenzugehörigkeit der Trainingsinstanzen erreicht werden. Hier kann man sehr anschaulich eine Einschätzung zur linearen / nichtlinearen Klassentrennbarkeit, zu Überlappungen bzw. Durchdringungen sowie Homogenität / Inhomo-

genität der visualisierten Cluster (Klassen) im PCA-Raum gewinnen.

Eine Möglichkeit der Bewertung der Komplexität und Relevanz eines gegebenen Datensatzes und des ihn beschreibenden Merkmalsraums besteht in der Anwendung des unüberwachten Maschinellen Lernens (Clustering), bei dem auf der Basis mathematischer Gesetzmäßigkeiten im Merkmalsraum natürliche Gruppen gebildet werden, um Informationen zur Struktur des Merkmalsraumes und somit zur Komplexität der Erkennungsaufgabe zu gewinnen. Während Klassen von Experten vorgegeben sind, werden Cluster ausschließlich anhand der mathematischen Gesetzmäßigkeiten in den Daten gebildet. Somit ist die Anzahl von Klassen bei der Klassifikation bekannt, wobei die Anzahl natürlicher Gruppen beim Clustering grundsätzlich abgeschätzt werden muss. Sind die Klassenzugehörigkeiten der Objekte jedoch gegeben, besteht die Möglichkeit, das Clustering-Ergebnis mit der Klassifikation zu vergleichen und zu überprüfen, inwieweit sich diese entsprechen.

Im Allgemeinen kann davon ausgegangen werden, dass die Klassifikation leistungstärker als das Clustering ist [15]. Sie führt zu einer besseren Gruppierung der Objekte, da sie auf den, vom Menschen vorgegebenen gewünschten Klassen basiert. Was ein Clustering-Verfahren lernt, hängt jedoch alleinig von den übergebenen Merkmalen und ihren Ausprägungen ab. Eben dieser Umstand kann für die Evaluation eines gegebenen Datensatzes genutzt werden. Sind die Merkmalsvektoren derart gewählt, dass Clustering-Verfahren im Stande sind eine Gruppierung zu erstellen, welche den ursprünglichen Klassen entspricht, so kann davon ausgegangen werden, dass eine gute Separation der Klassen möglich ist und ein einfacher Klassifikator verwendet werden kann. Entsprechen die durch Clustering gefundenen Gruppen jedoch nicht den eigentlichen Klassen, weist das auf einen für die Klassentrennbarkeit unzureichenden Merkmalsvektor und somit auf eine komplexe Klassifikationsaufgabe hin, welche eines komplexen Klassifikators bedarf.

Für diese Untersuchungen wurde der unüberwachte Expectation-Maximization-Algorithmus verwendet. Um die Cluster-Bildung mit den tatsächlichen Klassen vergleichen zu können, ist es wichtig ein Verfahren zu wählen, welches die manuelle Angabe der Anzahl zu bildender Cluster ermöglicht. Anschließend wurde ein der Erkennungsrate entsprechender Koeffizient, welcher als ER_{Clust} bezeichnet wird,

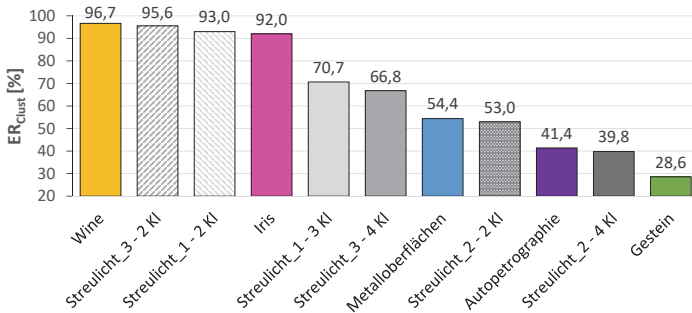


Abbildung 3: ER_{Clust} auf Basis des EM-Algorithmus und der tatsächlichen Klassen für reale Datensätze.

wie folgt bestimmt: $ER_{Clust} = \frac{n_{Richtig}}{n}$, wobei $n_{Richtig}$ die Anzahl an Objekten beschreibt, welche im Anschluss an das Clustering der richtigen Klasse zugeordnet wurden und n der Anzahl gesamt vorhandener Objekte entspricht. Abbildung 3 enthält die erhaltenen Ergebnisse der untersuchten Daten. Iris, Streulicht_3 und Streulicht_1 mit jeweils 2 Klassen sowie Wine stellen Datensätze mit geeigneten Merkmalsvektoren und somit einfache Klassifikationsaufgaben dar. In allen vier Fällen übersteigt ER_{Clust} 90%. Streulicht_1 und Streulicht_3 mit 3 bzw. 4 Klassen sind den mittelschweren Problemstellungen zuzuordnen. ER_{Clust} liegt hier über 60%. Die restlichen fünf Datensätze besitzen Merkmalsvektoren, welche die Separation der Klassen aus Sicht des EM-Algorithmus lediglich in geringer Weise widerspiegeln. Die natürlichen Gruppierungstendenzen entsprechen kaum den gegebenen Klassen, wodurch eine Klassifikation deutlich erschwert wird und komplexe Algorithmen benötigt werden. Die empirisch bestimmten Werte für ER_{Clust} sollen zukünftig unter Verwendung weiterer realer Datensätze aus der Praxis noch verfeinert werden.

6 Zusammenfassung der Ergebnisse

Die vorgestellten Verfahren können zur Analyse des Merkmalsraums eines gegebenen Erkennungsproblems und als Entscheidungshilfe zur Wahl eines Klassifikators angesehen werden. Deuten diese auf eine

einfache Struktur des Datensatzes und somit auf eine günstige Lage der Klassen zueinander hin, so sollten zunächst einfache Klassifikatoren verwendet und keine Zeit für aufwändige Parameteroptimierungen aufgebracht werden. Andernfalls sollten gleich komplexe Verfahren herangezogen werden, ohne Zeitverlust für das vorherige Probieren einfacher Algorithmen.

In den Untersuchungen konnte gezeigt werden, dass die erzielbare Erkennungsleistung eines Klassifikators stets von der Relevanz und Redundanz der Merkmale im Merkmalsraum abhängt und dass es möglich ist, mittels eines, auf der Basis der PCA berechneten Redundanzmaßes R eine Beurteilung der Redundanzen im Merkmalsraum durchzuführen. Die Ergebnisse der PCA können später zur Bereinigung des Merkmalsraums verwendet werden, indem redundante Merkmale entfernt werden, um eine bessere Erkennungsleistung sowie einen geringeren Rechenaufwand zu ermöglichen. Die Ergebnisse der PCA können ebenfalls gemeinsam mit dem Apriori-Wissen zur Klassenzugehörigkeit der Trainingsinstanzen dazu genutzt werden, die Komplexität des Datensatzes sowie die Klassentrennbarkeit im visualisierten 3D-Merkmalraum der ersten 3 HK zu beurteilen.

Untersuchungen zeigten weiterhin, dass die unüberwacht gefundenen Cluster bei Datensätzen mit einfacher Struktur in hohem Maße den bekannten Klassen des vorgegebenen Experten-Apriori-Wissens entsprechen. Derart große Übereinstimmungen konnten bei Datensätzen mit schwer zu trennenden bzw. überlappenden Klassen nicht erreicht werden. Weiterhin zeigten diese Untersuchungen, dass die Variation der Erkennungsraten zwischen einfachen und komplexen Klassifikationsalgorithmen bei Datensätzen mit klar zu trennenden Klassen deutlich geringer war. Das bedeutet, dass einfache Verfahren hier ebenfalls in der Lage waren gute Ergebnisse zu liefern. Liegen Datensätze mit komplexen Strukturen vor, sind einfache Klassifikatoren nicht mehr in der Lage, zufriedenstellende Ergebnisse zu liefern, wohingegen komplexe Algorithmen zu hohen Erkennungsraten führen.

7 Ausblick

Die empirisch ermittelten Werte des Redundanzmaßes R sowie der Cluster-Erkennungsrate sollen im Rahmen weiterführender Untersu-

chungen mit einer größeren Anzahl an realen Datensätzen sowie synthetisch erzeugten Datensätzen mit definiertem Redundanzanteil sowie Rauschanteil weiter untersucht und gegebenenfalls angepasst werden.

Literatur

1. R. Langenbach, "Methoden der farbbildverarbeitung für die automatische oberflächeninspektion," Dissertation, Siegen, 2002.
2. K. Anding, "Ein Beitrag zur qualitätssichernden Sichtprüfung von Produkten mit Bildverarbeitungstechnologien," Habilitationsschrift, Technische Universität Ilmenau, Februar 2016.
3. R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence*, 1995, pp. 1137–1145.
4. scikit-learn developers, "Choosing the right estimator," 2017. [Online]. Available: http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html
5. DLIB, "Dlib c++ library machine learning guide." [Online]. Available: <http://dlib.net/ml.html>
6. Microsoft, "Machine learning – cheat sheet für algorithmen für microsoft azure machine learning studio," 2017. [Online]. Available: <https://docs.microsoft.com/de-de/azure/machine-learning/studio/algorithm-cheat-sheet>
7. M. Zorrilla and D. García-Saiz, "Meta-learning based framework for helping non-expert miners to choose a suitable classification algorithm: An application for the educational field," in *Computational Collective Intelligence*, 2015, pp. 432–440.
8. D. García-Saiz and M. Zorrilla, "A meta-learning based framework for building algorithm recommenders," *Journal of Intelligent & Fuzzy Systems*, vol. 32.2, pp. 1449–1459, 2017.
9. M. Reif, F. Shafait, M. Goldstein, T. Breuel, and A. Dengel, "Automatic classifier selection for non-experts," *Pattern Analysis and Applications*, vol. 17(1), pp. 83–96, 2014.
10. C. Soares and Brazdil, "Zoomed ranking: Selection of classification algorithms based on relevant performance information." in *European Conference on Principles of Data Mining and Knowledge Discovery*. Berlin, Heidelberg: Springer, 2000, pp. 126–135.

11. L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka," *Journal of Machine Learning Research*, vol. 17, pp. 1–5, 2016.
12. S. Theodoridis and K. Koutroumbas, *Pattern recognition*, 4th ed. Amsterdam: Elsevier/Acad. Press, 2009.
13. R. Bellman, *Adaptive Control Processes: A Guided Tour*, ser. Princeton legacy library. Princeton, N.J: Princeton University Press, 1961.
14. E. Alpaydin, *Introduction to Machine Learning*, 2nd ed., ser. Adaptive computation and machine learning. Cambridge: MIT Press, 2010.
15. C. C. Aggarwal, *Data Mining: The Textbook*. Cham: Springer, 2015.
16. J. Bacher, A. Pöge, and K. Wenzig, *Clusteranalyse: Anwendungsorientierte Einführung in Klassifikationsverfahren*, 3rd ed. München: Oldenbourg, 2010.
17. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm." *Journal of the royal statistical society*, vol. Series B (methodological), pp. 1–38, 1977.
18. L. Breiman, "Random forests," *Machine Learning*, vol. 45(1), pp. 5–32, 2001.
19. I. H. Witten, E. Frank, M. A. Hall, and Pal Christopher J., *Data mining: Practical Machine Learning Tools and Techniques*, 4th ed. Cambridge, MA: Morgan Kaufmann, 2017.

Automated classification of imperfections and dust on small optical elements

Alexander Schöch¹, Patric Perez¹, Sabine Linz-Dittrich²,
Carlo Bach¹, and Carsten Ziolek²
{alexander.schoech, patric.perez, sabine.linz,
carlo.bach, carsten.ziolek}@ntb.ch

- ¹ NTB Interstaatliche Hochschule für Technik Buchs, Institute for Production
Metrology, Materials and Optics, Machine Vision Group
Werdenbergstrasse 4, CH-9471 Buchs
- ² NTB Interstaatliche Hochschule für Technik Buchs, Institute for Production
Metrology, Materials and Optics, Technical Optics Group
Werdenbergstrasse 4, CH-9471 Buchs

Abstract This article discusses a specifically designed machine vision setup to assess surface imperfections on optical elements according to the DIN ISO 14997 standard is discussed. By emulating a human testers inspection process, imperfections that are not trivially detectable by a simple camera/light setup are made visible to the system. A complete removal of dust before the inspection process is a requirement if dust particles shall not be detected as surface defects. Practical testing of the system revealed that a complete removal is economically not feasible and thus the need for a method to discriminate between dust particles and other surface imperfections was identified. This article describes an approach to discriminate between dust particles and imperfections in the surface of interest.

Keywords Quality control, automated surface inspection, optical elements, optical industry.

1 Introduction

Manufacturing industries aim to minimize defects in their products for maximum customer satisfaction. Manufacturers of basic optical elements are no exception here. The production of lenses, mirrors or

prisms becomes more and more challenging due to continuously decreasing geometrical dimensions and tolerances, e.g. for application in medical devices. Additionally, customer specific requirements may result in small batch sizes. As a consequence, demands on quality control such as surface inspection increase in terms of accuracy and flexibility. Surface imperfections may be of cosmetic interest only [1–3], i.e. they do not significantly impair the optical performance, however, customers often request minimal surface imperfections nevertheless.

The definition of surface imperfections and appropriate inspection methods are addressed in the standards DIN ISO 10110-7 [4] and DIN ISO 14997 [5] respectively. The American standard MIL-O-13830A uses a different approach and cannot be compared directly to its ISO counterpart [1]. This article is focused on inspection methods regarding the ISO standard.

Surface inspection is typically done by human operators. Combined with increasing throughput requirements, a higher number of skilled personnel becomes necessary to retain product quality. By the decreasing size of imperfections and varying customer-specific geometries, the time until fatigue may be reduced. Clearly, manual inspection is inevitably accompanied by variation in the results due to subjective perception [1–3]. Motivated by these shortcomings, several attempts for automated systems were made which are shortly reviewed in the next section.

A survey of quality control processes at six Swiss optics manufacturers revealed that basic optical components are assessed by means of manual visual inspection. During observation, it was ascertained that experienced testers detect and classify imperfections of lateral extensions down to $16\text{ }\mu\text{m}^2$ in a matter of seconds. They achieve high accuracy by use of various magnification utensils, a dedicated comparison artefact and proper variation of viewpoint and light source position.

Specification and tolerancing of surface imperfections and appropriate inspection methods are addressed in the standards DIN ISO 10110-7 [4] and DIN ISO 14997 [5] respectively. The American standard MIL-O-13830A takes a different approach and cannot be easily compared with its ISO counterpart [1]. This article is focused on inspection methods regarding the DIN ISO 14997 standard.

The DIN ISO 10110-7 defines several classes of surface imperfections: “Edge chips”, “Long scratches”, “Coating blemishes” and “Localized

surface imperfections". The latter class comprises several regional defects (scratches, scuffs, etc.) and dust particles. Each class can be tolerated in technical drawings in terms of imperfection size and the maximal permissible number of imperfections. E.g. $5/3 \times 0.004$; specifies that 3 defects of class localized surface imperfections of *grade* 0.004 are permissible. The *grade* number represents the square root of the surface defects area in millimeter. For a more detailed discussion of defects classes the reader is referred to [6].

Inspection methods as described in the current DIN ISO 14997 standard are limited to manual visual inspection. The draft version of the upcoming E DIN ISO 14997:2016-08 [7] does include automated inspection systems. However, the need for agreement between manufacturer and customer on measurement instruments and conditions is stated. Thus, the involved parties have to agree upon a specific system. It would be preferable to agree on a maximal permissible error of the measurement instrument, similar to the methods described in dimensional measurement standards. Regarding automated systems, three commercial products to evaluate surface imperfections on optical elements are currently available on the market.

The SavvyInspector [8] from Savvy Optics Corp. employs a dark-field setup and a matrix sensor, allowing for a $1 \text{ mm} \times 1 \text{ mm}$ inspection area. Besides its applicability for flat parts, also "mild concave" surfaces are advertised for. The system was originally designed to assess conformance to the American standard MIL-O-13830A but provides also a DIN ISO 10110-7 mode. For this, the minimal specified grade is $5/1 \times 0.025$ for a single imperfection. Imperfection classes are limited to general defects and coating imperfections.

Redlux, an UK based company, builds and distributes an automated system to inspect flat optical elements called OptiLux SD. They apply a similar approach as the SavvyInspector, namely a diffuse dome with a camera at its North Pole to generate a darkfield setup augmented by an xy-stage to scan large surfaces. Curved elements can be measured by "manual measurement", i.e. tilting the object multiple times such that surface normals become sufficiently orthogonal to the cameras optical axis.

A different approach is found in the Dioptic ARGOS system [2]. It employs a line sensor and a rotation stage. A holistic representation of an optical elements surface is constructed by rotating the element

during the acquisition phase and transforming the line sensor data accordingly. Their darkfield setup enables assessment of curved elements such as lenses by preliminary manual adjustment. It prevents the occurrence of specular reflections from curved surfaces and provides high-resolution images up to 256 Megapixel. As a minimal quantifiable grade, $5/1 \times 0.0025$ for single imperfections is stated and a repeatability between $0.4 \mu\text{m}$ and $1.3 \mu\text{m}$ based on 30 repetitions is reported. Etzold et al. [2] indicate that the system discriminates digs, scratches and edge chips.

For a more detailed literature review of systems to inspect surfaces on specular reflecting, transparent parts please see [6]. In this article, an alternative system, still with darkfield setup though, is presented. The system renders mechanical actuators redundant by replicating viewports and light source, i.e. usage of multiple cameras and LEDs. Appropriate illumination and viewing directions can be chosen dynamically based on image data while preserving a quick acquisition process without movement. The system thus emulates the behaviour of a human inspector having multiple viewpoints and lighting directions. The system classifies a variety of imperfection and dust particles.

2 Acquisition system

The ISO standard proposes the use of a white light darkfield setting. First experiments were conducted with a diffuse light dome in a darkfield setup. Such a system has the potential to make typical surface defects visible especially on flat surfaces. However, with one single camera, it is hard to holistically detect imperfections on a curved surface of an optical component [1,2].

In order to test for a multitude of camera poses and keep the experiments economically feasible, a surrogate setup was designed that allows emulating multiple camera positions by one rotation. As shown in figure 1, the dome can be rotated around two orthogonal axes – this is equivalent to rotations of the camera around the domes centre and was preferred due to practical reasons. Clearly, camera positions are only feasible where the line of sight is not obstructed by the hemispheres geometry, i.e. at the North Pole and between strands.

The dome consists of 16 strands, with five or six LEDs attached to

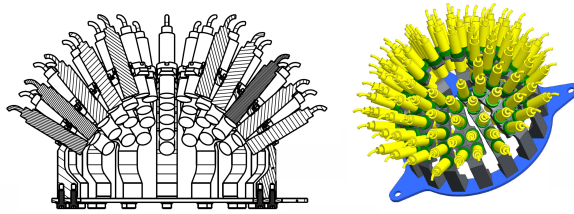


Figure 1: CAD drawing of the surrogate setup. Left: Section from front view. Right: 3D view.

each of them resulting in a total of 88 light sources and directions (figure 1). Each LED can be controlled separately in intensity. This allows for discrete changes of the illumination angle, creating a brightfield or darkfield setting, as well as lighting scenarios with multiple sources enabled. To address the potentially significantly curved surfaces of lenses, multiple viewpoints become necessary. Instead of having multiple cameras, a rotary table was used to emulate multiple viewpoints for economical reasons. By employing a rotary table and arranging LEDs rotational symmetric, multiple viewpoints can be emulated by rotation of the optical element (figure 2). The current setup uses a telecentric lens with a magnification ratio of 1 and a monochrome 5 megapixel CMOS camera. For results of preliminary experiments regarding the visibility of different defects classes, the reader is referred to [6].

3 Image processing

Based on multiple images from different viewpoints and distinct lighting conditions, the position and grade of surface imperfections on the optical element are to be evaluated. This process consists of preprocessing, specular reflex reduction, dust suppression, data fusion, segmentation, and classification. These steps will be discussed in the following.

In order to constrain data processing to the optical element under consideration, a preprocessing step lets the operator crop images and specify an appropriate mask, e.g. an ellipse. Note that these bounding box and mask parameters have to be requested only once per optical

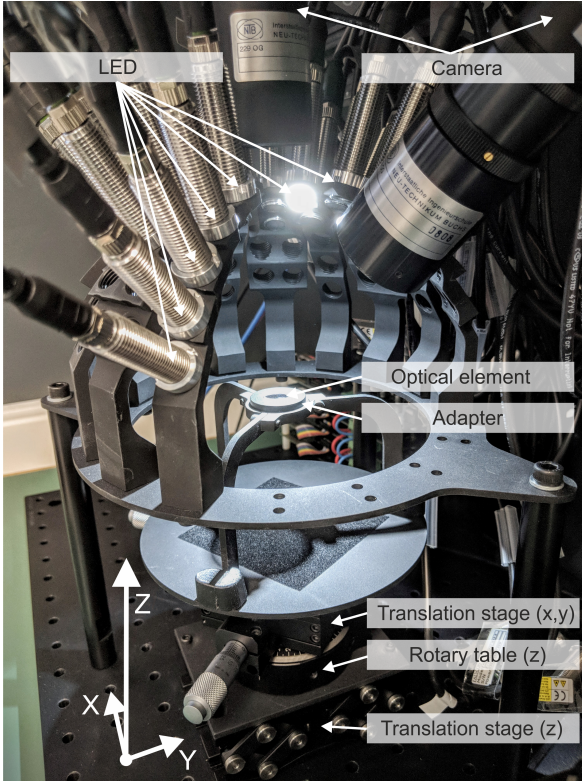


Figure 2: Photograph of the surrogate setup, only a small portion of the light sources are mounted for better clarity. The telecentric lenses indicate the positions two cameras. By employing a rotary table and arranging LEDs rotational symmetric, multiple viewpoints can be emulated by rotation of the optical element. The element is held in place by a geometry specific adapter.

element geometry from the operator.

Specular reflex reduction shall be discussed explicitly here since it poses a non-trivial problem. On curved surfaces, specular reflexes from the involved LEDs are clearly visible in the images (figure 3). The current approaches include (i) extraction and damping of circular reflexes without a priori knowledge; (ii) damping with a grayscale mask (fig-

ure 3, bottom left) which consists only of reflexes, created by recording multiple elements of the same geometry and appropriate aggregation; (iii) creation of a grayscale mask as in (ii) but by knowledge of the setup and element geometry and means of raytracing; (iiii) simply reject images with reflexes. By several experiments, it was found that (ii) yields the best classification results if sufficient optical elements are used for the mask creation.

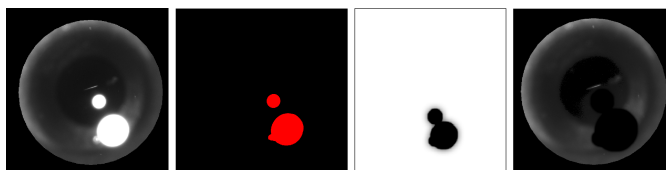


Figure 3: From left to right: an example of specular reflexes on a lens with one LED enabled; generated binary mask; processed mask with blurred edges; resulting image after masking.

To obtain a holistic representation of the elements surface, the data from multiple viewpoints and lighting conditions is to be fused into a single dataset. A pair of viewpoint and lighting condition can be considered as one sensor. With such a definition, the utilised sensor configuration is complimentary. Complementary configurations aim to combine information to give a more complete representation of a phenomenon and do not directly depend on each other [9].

The applied approach fuses the datasets on pixel-level by e.g. a maximum criterion: the resulting intensity at pixel coordinate (x, y) is the maximum intensity at (x, y) over images of all lighting conditions for a fixed viewpoint. By this choice of aggregation, defects that result in relatively low intensities are retained. However, also specular reflexes appear in the result what motivated the prior removal step. The aggregated dataset (example in figure 5) is then used for segmentation of defects. Note that datasets from multiple viewpoints are registered to each other by reversing lens distortion and applying a homography transformation. Thus, from data of several lighting conditions and viewpoints, one fused image is generated.

At the step of transforming images from various viewpoints into one viewpoint, there exists an opportunity to suppress the appearance of

non-defect artefacts in the fused result. Essentially, locations of light scattering which are not in the plane of interest are transformed to different positions in the resulting image as shown in figure 4. By appropriate post-processing, non-overlapping intensities from different viewpoints can be suppressed, thus removing artefacts which are not in the plane of interest. This effect can be exploited to remove e.g. scattering from side faces or dust particles (zoom in figure 4). It should be noted that the resulting positions in the fused image must be non-overlapping, thus requiring a sufficiently high angle between the different viewpoints and relatively small artefacts. This approach could be extended to curved surfaces by considering uv-coordinates on e.g. a lens instead of plane coordinates. In single viewpoint systems, such a discrimination of particles can be achieved by approaches such as the one discussed by Li et al. [10].

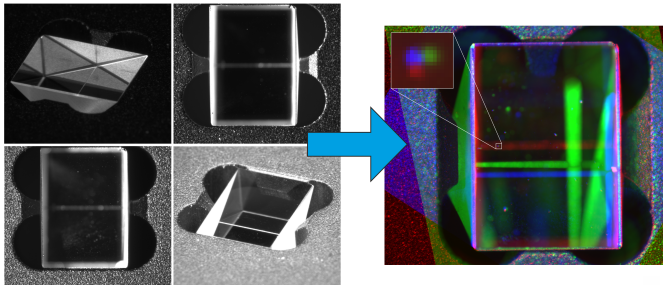


Figure 4: Aggregation of images from multiple viewpoints (left) by homography transformations into one image (right, three images shown as RGB channels for demonstration). By sufficiently accurate calibration and adequate setup, objects that are not in the plane of interest are projected to different positions (zoom on the right image).

Different methods were tested to segment defects in the fused image, i.e. static and adaptive thresholding and application-specific approaches such as from Zhang et al. [11]. It was found that Bradleys method for adaptive thresholding [12] works consistently well for considered optical elements.

The last step classifies the segmented regions according to the DIN ISO 10110-7 and DIN ISO 14997 standards. First, the class of a defect is

evaluated by its position and eccentricity, e.g. edge chips are connected to the border of the optical element and scratches have a higher eccentricity as digs. The grade is then determined by geometrical means, according to the previously evaluated class. However, not only the number of pixels and the magnification ratio is evaluated to assign a grade to a defect but also the pixel intensity. To tweak the “intensity awareness” of the system, a factor k was introduced to assign a grade A_{px} in pixels as

$$A_{px} = \sum_{i \in \mathcal{I}} (k \cdot i + (1 - k)). \quad (1)$$

Where $\mathcal{I}$ represents the set of intensities of the defect. i is one of those intensities and k is to be chosen between 0 and 1. Dependent on this choice, pixels which were segmented are simply counted ($k = 0$) or their intensities are aggregated ($k = 1$). Clearly, $k = 0$ results in higher grades, and thus in a more strict operation of the system. Note that typically, a high value of k is preferable not only due to economic reasons but also since it replicates the behaviour of operators who take the defects intensity into account. For a metric result, the area in pixels A_{px} is converted under consideration of camera and optics parameters into square millimeters. Eventually, a resulting *grade* is rounded according to Annex E in DIN ISO 14997 [5]. An exemplary result is given in figure 5.

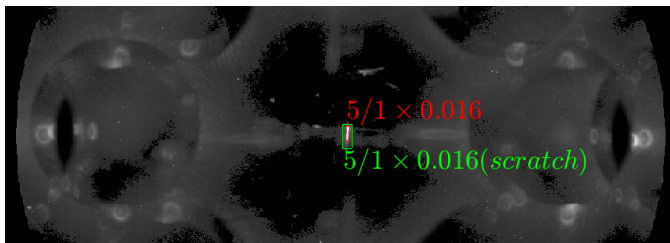


Figure 5: Example of a fused dataset from 48 lighting conditions and the classified defect at the centre of the lens.

4 Comparison to manual visual inspection

Due to several project partners from industry, the authors gained access to badges of optical elements of the same geometry. Moreover, they have inspection personnel to test these batches at an industrial grade. This allows for testing of different geometries with known and viable manual inspection results.

For each batch, the first step is a rigorous manual inspection by partners from industry according to the DIN ISO 14997 standard. They generate a protocol for each geometry where all found imperfection are documented in type, size, and location. As a second step, the same elements are inspected by the proposed system, keeping them as dust-free as possible. In the last step, results are compared and visualized. This comparison is presented in the following for one batch of 20 biconvex lenses, having a total of 19 different defects on 16 of these lenses. Four lenses did not have any defect at all.

During experiments, dust particles were deposited on the surface caused by the non-cleanroom environment in the laboratory. Thus, evaluation naturally result in additional defects. As they are technically correct detections, they are caused by handling and were obviously not assessed by manual visual inspection beforehand. Note that the ISO standards do include dust as one class of surface imperfection, however, such specifications were not encountered in industrial application during surveys.

Presence of dust usually results in a miscount due to the additional and only partially valid defect occurrences. To check for the main objective, true-positives and false-negatives were assessed, as these results of real imperfections have a strong impact on project viability.

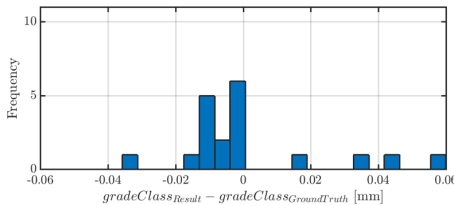


Figure 6: Defect grade classification performance for 20 biconvex lenses.

Within the considered batch, only one defect has not been detected by the proposed setup. This defect was neither visible for humans in the acquired images. We suppose that the lack of sufficient lighting positions causes this invisibility. Additionally to the detection rate, the differences in the reported grade were evaluated and are given in figure 6 that shows a histogram of the difference between automatic and manual detection. The results indicate that the system currently underestimates defect grades from manual inspection by maximally $60\text{ }\mu\text{m}$, whereas 14 results deviated by less than $20\text{ }\mu\text{m}$. The proposed system was not intended to fully resolve structures smaller than $4\text{ }\mu\text{m}$ but is able to detect smaller defects.

5 Conclusions

The proposed setup allows mimicking the manual inspection process according to the DIN ISO 14996 standard for surface imperfections in an automated way.

The system demonstrates certain advantages over other, commercially available systems, due to the use of multiple viewpoints and lighting directions. It enables acquisition of a sparse pseudo BRDF that can be utilized to discriminate between specific defects, i.e. scratches and scuffs. It was shown that artefacts that do not belong to the surface to be inspected can be discriminated by a stereo approach and appropriate data processing.

Automatic detection and classification of defects according to the DIN ISO 10110-7 and DIN ISO 14997 standards can be achieved with the proposed setup. A detection rate of 18 out of 19 defects on a batch of 20 biconvex lenses w.r.t. manual inspection was demonstrated.

The gained understanding of most defect types under different illumination and camera angles leads to possible designs for a motionless setup, equipped with multiple physical cameras. This next iteration of the system is to be tested and assessed under industrial conditions.

References

1. R. Neubecker and J. E. Hon, "Automatic inspection for surface imperfections: requirements, potentials and limits," in *Third European Seminar on*

- Precision Optics Manufacturing*, ser. SPIE Proceedings, R. Rascher, O. Föhnle, C. Wünsche, and C. Schopf, Eds. SPIE, 2016, p. 1000907.
2. F. Etzold, D. Kiefhaber, A. F. Warken, P. Würtz, J. Hon, and J.-M. Asfour, "A novel approach towards standardizing surface quality inspection," in *Third European Seminar on Precision Optics Manufacturing*, ser. SPIE Proceedings, R. Rascher, O. Föhnle, C. Wünsche, and C. Schopf, Eds. SPIE, 2016, p. 1000908.
3. Q. Turchette and T. Turner, "Developing a more useful surface quality metric for laser optics," in *SPIE LASE*, ser. SPIE Proceedings, W. A. Clarkson, N. Hodgson, and R. Shori, Eds. SPIE, 2011, p. 791213.
4. DIN ISO 10110-7, "Optik und Photonik – Erstellung von Zeichnungen für optische Elemente und Systeme – Teil 7: Oberflächenunvollkommenheiten: Surface imperfection tolerances (ISO 10110-7:2008)," 2009.
5. DIN ISO 14997, "Optik und Photonik – Prüfverfahren für Oberflächenunvollkommenheiten optischer Komponenten (ISO 14997:2011)," 2013.
6. A. Schöch, P. Perez, S. Linz-Dittrich, C. Bach, and C. Ziolek, "Automated surface inspection of small customer-specific optical elements," in *Forum Bildverarbeitung 2016*, M. Heizmann, T. Laengle, and F. Puente León, Eds. Karlsruhe: KIT Scientific Publishing, 2016, pp. 37–48. [Online]. Available: <https://publikationen.bibliothek.kit.edu/1000059899>
7. E DIN ISO 14997, "Optik und Photonik – Prüfverfahren für Oberflächenunvollkommenheiten optischer Elemente (ISO/DIS 14997:2016)," 2016.
8. D. M. Aikens, "Objective Measurement of Scratch and Dig," in *Applied industrial optics: Spectroscopy, imaging and metrology ; part of Imaging and applied optics ; 24-28 June 2012, Monterey, California, United States*, ser. OSA technical digest (online). OSA, The Optical Society, 2012.
9. A. Weckenmann, X. Jiang, K.-D. Sommer, U. Neuschaefer-Rube, J. Seewig, L. Shaw, and T. Estler, "Multisensor data fusion in dimensional metrology," *CIRP Annals - Manufacturing Technology*, vol. 58, no. 2, pp. 701–721, 2009.
10. L. Li, D. Liu, P. Cao, S. Xie, Y. Li, Y. Chen, and Y. Yang, "Automated discrimination between digs and dust particles on optical surfaces with dark-field scattering microscopy," *Applied Optics*, vol. 53, no. 23, pp. 5131–5140, 2014.
11. Z. Zhang and C. Li, "Defect Inspection for Curved Surface with Highly Specular Reflection," in *Integrated Imaging and Vision Techniques for Industrial Inspection*, ser. Advances in Computer Vision and Pattern Recognition, Z. Liu, H. Ukida, P. Ramuhalli, and K. Niel, Eds. London: Springer London, 2015, pp. 251–317.
12. D. Bradley and G. Roth, "Adaptive Thresholding using the Integral Image," *Journal of Graphics Tools*, vol. 12, no. 2, pp. 13–21, 2011.

Stochastische Mustererkennung zur Bildsegmentierung

Stochastic pattern recognition for image segmentation

Thomas Müller¹ und Bastian Erdnütz²

¹ Fraunhofer IOSB - Institut für Optronik, Systemtechnik u. Bildauswertung,
Fraunhoferstr. 1, 76131 Karlsruhe

² Karlsruher Institut für Technologie,
Institut für Photogrammetrie und Fernerkundung,
Englerstr. 7, 76131 Karlsruhe

Zusammenfassung Luftbilder zeigen häufig weite homogene Bereiche mit inhomogener Textur. Das sind Bereiche, die zwar lokal eine hohe Schwankung in den Farbwerten aufweisen, die jedoch stochastisch ein augenscheinlich so regelmäßiges Muster zeigen, dass man sie als Mensch leicht als zusammengehörige Fläche wie einen Wald, ein Feld, einen Weg oder ein Gewässer erkennt. Mit der vorliegenden Arbeit sollen solche Bereiche erkannt werden. Dazu werden zunächst Texturmodelle zu verschiedenen Bildbereichen generiert, die strukturelle Ähnlichkeit ermittelt und hinreichend unterschiedliche Texturmodelle gespeichert. Anschließend wird jedes Pixel des Bildes via Struktur-Metrik einem der gefundenen Texturmodelle zugeordnet und entsprechend charakteristisch eingefärbt. Dabei werden drei unterschiedliche Ansätze zur Texturmodellierung untersucht und miteinander verglichen im Hinblick auf das Ziel, möglichst viele Bildbereiche als gleichartig zu segmentieren, die von einem Menschen als ähnlich angesehen werden (z.B. alle Baumkronen im Bild, Wege oder Äcker) und Segmentgrenzen dort zu erzielen, wo augenscheinliche Übergänge sind (z.B. Waldgrenzen, Auto auf einer Wiese, Fahrzeugspuren im Gelände, Vegetationsveränderungen).

Schlagwörter Bildtextur, Bildsegmentierung, Objekterkennung, Landnutzung.

Abstract Aerial image often show wide homogeneous regions with inhomogeneous texture. Such regions may contain high local color dynamics but nevertheless show apparently regular stochastic patterns which make it easy for a human observer to recognize them as associated structures like forests, acres, pathways, or waterbodies. This paper intends to recognize such regions. To do this, first, texture models are generated for different image regions, structural similarities are determined and sufficiently different texture models are stored. Afterwards, each image pixel is assigned to one of those texture models with a structural metric and is then colored characteristically for the assigned texture model. In this procedure, three different methods of texture modelling are examined and compared with each other in view of the goal to reach a segmentation result in which as many regions as possible are found to be similar that are seen as similar also by a human observer (e.g., all treetops in the image, pathways, or acres) and in which segment margins correspond to apparent transitions (e.g., boundary of a forest, a car on a grassland, vehicle tracks in the field, vegetation changes).

Keywords Image texture, image segmentation, object recognition, land use.

1 Einleitung

In dieser Arbeit werden drei unterschiedliche Textur-Verfahren untersucht und verglichen, welche die lokale stochastische Struktur von Bildbereichen modellieren. Ziel ist es, strukturell ähnliche und unterschiedliche Regionen in einem Bild möglichst gut zu erkennen, um mit diesen später weiterarbeiten zu können. Dabei wird auf kein vorab trainiertes Modellwissen zurückgegriffen. Die erfolgende Nutzung zur texturbasierten Bildsegmentierung bedeutet zum Einen eine Anwendung an sich, aber zum Anderen auch unmittelbar eine günstige Visualisierung der Ergebnisse zur Einschätzung der Leistung der Verfahren bzw. der qualitativen Eigenschaften und Unterschiede.

Das Basisvorgehen ist bei allen drei Ansätzen das gleiche. Zunächst werden gleichmäßig über das Bild verteilt Struktur-Beschreibungen berechnet und eine Teilmenge hinreichend unähnlicher Beschreibungen daraus ermittelt. Anschließend wird für jedes Bildpixel über ei-

ne Struktur-Metrik berechnet, ob und zu welcher dieser Struktur-Beschreibungen es passt. Dadurch ergibt sich die Segmentierung des Bildes, die farblich visualisiert wird.

Im Folgenden werden drei Texturmodelle mit ihren Struktur-Metriken untersucht: Eine basierend auf Auto-Regressiven Filtern (AR-Filtern), eine basierend auf einer Erweiterung der Schablonenanpassung (engl. Template Matching) und eine basierend auf Histogrammen von Grauwerten/Farben und Farbübergängen (vgl. HoG-Features, HoG = Histogram of oriented Gradients).

2 Auto-Regressiver Filter (AR-Filter)

2.1 Texturmodell

Um ein Test-Pixel x herum wird eine Umgebung A_x betrachtet. Für jedes Pixel $y \in A_x$ in der Umgebung werden die Intensitätswerte der wiederum y umgebenden Pixel $z \in B_y \setminus \{y\}$ zu einem Vektor v_y aufgelistet. Diese Vektoren werden als Zeilen einer Matrix $M_x := (v_y^T)_{y \in A_x}$ zusammengefasst. Die dabei ausgesparten Intensitätswerte der Zentralpixel y bilden die entsprechenden Komponenten eines Vektors b .

Für jedes Pixel $y \in A_x$ entsteht so eine Zeile in M_x bzw. b . M_x und b haben $m := |A_x|$ Zeilen, und M_x hat $n := |B_y \setminus \{y\}|$ Spalten, wobei $|\cdot|$ die Kardinalität einer Menge ist. Jede Zeile im linearen Gleichungssystem

$$M_x p = b \tag{1}$$

bedeutet anschaulich, dass ein Zentralpixel y (in b) als Linearkombination p der Umgebung $B_y \setminus \{y\}$ (Spalten der Zeile in M_x) beschrieben wird. Wenn also das Testpixel x Teil einer stochastisch einheitlichen Textur ist, so ist die Umgebung B_x um x herum ein „Sample“ dieser Textur, ebenso wie auch alle Umgebungen B_y um die benachbarten Pixel $y \in A_x$. Grundidee des Ansatzes ist, dass dieses Sample durch Lösen des Gleichungssystems 1 nach der kleinste Quadrate-Methode gefunden wird.

Grundsätzlich können A_x und B_y von beliebiger (aber fixer) Form sein. Die Reihenfolge bei der Abarbeitung der Pixel in den beiden Umgebungen entscheidet über die Reihenfolge der Zeilen und Spalten in M_x bzw. b . Sie ist grundsätzlich beliebig, muss aber bei allen Berechnungen fix bleiben. In dieser Arbeit werden die Umgebungen A_x und B_y kreisförmig gewählt mit den Radien r_A bzw. r_B Pixel. Die Reihenfol-

Tabelle 1: Größen m und n in Abhängigkeit der Radien r_A bzw. r_B .

r_A bzw. r_B	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
m bzw. $n + 1$	5	13	29	49	81	113	149	197	253	317	377	441	529	613	709
r_A bzw. r_B	16	17	18	19	20	21	22	23	24	25	26				
m bzw. $n + 1$	797	901	1009	1129	1257	1373	1517	1653	1793	1961	2121				

ge bei der Abarbeitung erfolgt ähnlich wie bei Pixelbildern zuerst zeilenweise und dann spaltenweise. Tabelle 1 illustriert die Anzahl Pixel m bzw. n in den Umgebungen A_x bzw. B_y abhängig von r_A bzw. r_B .

Die Umgebung A_x muss mindestens so groß wie B_y sein, d.h. $r_A \geq r_B$, sodass $m \geq n$ und das lineare Gleichungssystem 1 bei geeigneten Eingabedaten gelöst werden kann. Wird keine Lösung gefunden, so existiert keine hinreichende Widerspruchsfreiheit in der Umgebung um x und somit befindet sich dort auch keine via AR-Filter beschreibbare Textur. Kann eine Lösung berechnet werden, so ist der ermittelte n -dimensionale Vektor p eine Beschreibung der Textur um den Punkt x , d.h. *Texturmodell* zu x .

Die bisherige Beschreibung mit Intensitätswerten geht von Grauwertbildern aus. Bei Farbbildern werden drei Gleichungssysteme 1 (für jeden Farbkanal eines) aufgestellt und gelöst. Alternativ wurde auch untersucht, nur ein Gleichungssystem mit $3m$ Zeilen und $3n$ Unbekannten zu lösen. Dies führte jedoch zu etwas schlechteren Endergebnissen.

Die erhaltenen Ergebnisse lassen sich verbessern, wenn statt des Gleichungssystems 1 das entsprechende mit mittelwertbereinigten Spalten gelöst wird. D.h. vor Lösung des Gleichungssystems wird zu jeder Spalte in M_x und b der Mittelwert berechnet und von jedem Spaltenelement subtrahiert.

Beim zu lösenden Gleichungssystem kommt es nur darauf an, ob sich Pixel unterscheiden oder nicht. Große Unterschiede sind nicht gravierender als kleine und müssen daher auch nicht stärker gewichtet werden. Eine weitere Verbesserung der Endergebnisse stellt man daher fest, wenn das Gleichungssystem 1 so gelöst wird, dass die L1-Norm des Restfehlers minimiert wird anstatt der Euklidnorm (Huber-Schätzer, Reweighting-Methode) [1,2].

Werden r_A und r_B so gewählt, dass sich m und n zu wenig unterscheiden, stellt man schlechte Endergebnisse fest. Ebenso wirkt es sich auch ungünstig aus, wenn m viel größer als n ist. Oft (aber nicht immer)

wurden die besten Endergebnisse bei $r_A = r_B + 1$ beobachtet. Letztendlich hat am besten funktioniert, r_B vorzugeben und für r_A den kleinsten Wert zu wählen, bei dem M_x mindestens 50 Prozent mehr Zeilen als Spalten hat, d.h. $m \geq 1,5 \cdot n$ (vgl. Tabelle 1).

2.2 Evaluation

Zur Quantifizierung, inwieweit ein Pixel x einem zuvor (i.A. zu einem völlig anderen Pixel t) berechneten Texturmodell p folgt, wird zunächst M_x und b um x sowie die Differenz $d := M_{xp} - b$ berechnet und die *Struktur-Metrik* definiert als

$$\delta := \sum_{i=1}^m \left(\begin{cases} 1, & \text{falls } |d_i| > \epsilon \\ 0, & \text{sonst} \end{cases} \right) + 10^{-6} \cdot |d_i| \quad , \quad (2)$$

wobei d_i die i -te Komponente von d bezeichnet. Der erste Summand nach dem Summenzeichen ist der wesentliche, er zählt die Pixel in A_x , die dem Texturmodell widersprechen. Falls bei zwei Texturmodellen das gleiche Zählergebnis entsteht, sorgt der zweite Summand dafür, dass das Modell mit der kleineren Abweichung in d zu einem kleineren Wert δ führt. ϵ ist eine einstellbare Konstante, die angibt, bis zu welcher Differenz die Intensitätswerte zweier Pixel nicht unterschieden werden sollen, z.B. $\epsilon = 15$. Bei Berechnung von d wird analog zu Abschnitt 2.1 mit mittelwertbereinigten Spalten gerechnet.

x wird als *kompatibel zum Modell p* angesehen, wenn $\delta \leq c \cdot m$, wobei c ein voreinstellbarer Prozentsatz ist. In dieser Arbeit wird der relativ kleine Wert $c = 30\%$ verwendet. Dies ermöglicht die Zuordnung von Texturen zueinander auch bei starken örtlich konzentrierten oder verteilten Unterschieden.

2.3 Bildsegmentierung

Über das Bild wird ein regelmäßiges Raster gelegt und an den Rasterpunkten die Textur untersucht. In einer Liste von Texturmodellen werden die gefundenen, unterschiedlichen Texturmodelle gespeichert. Dazu wird für jeden Rasterpunkt zunächst geprüft, ob er kompatibel ist zu einem bereits gespeicherten Texturmodell der Liste. Falls das nicht der Fall ist, wird p für den Rasterpunkt berechnet und in die Liste aufgenommen.

Zur Bildsegmentierung wird für jedes Bildpixel das Texturmodell der Liste mit dem kleinsten δ bestimmt. Ist das Pixel kompatibel zum gefundenen Texturmodell, wird das Pixel mit einer Farbe coloriert, die für dieses Texturmodell steht.

3 Schablonenanpassung (Template Matching)

3.1 Texturmodell

Wie beim AR-Filter wird die kreisförmige Umgebung A_x um ein Test-Pixel x mit m Elementen und wohldefinierter Reihenfolge der Pixelpositionen betrachtet. Die $k := m$ Intensitätswerte bei Grauwertbildern bzw. $k := 3m$ Farbwerte bei 3-Kanal-Farbbildern in der Umgebung A_x um x seien bezeichnet mit s_i^x , $i = 1, \dots, k$. s_*^x heisst *Texturmodell* zu x und dient als Schablone (engl. Template) für Texturvergleiche.

3.2 Evaluation

Ein Texturmodell s_*^t korrespondiert nach Definition mit einem Pixel t , um welchen es erzeugt wurde. Zur Quantifizierung, inwieweit ein anderes Pixel x dem gleichen Texturmodell folgt, werden zunächst die absoluten Differenzen

$$d_i := |s_i^x - s_i^t| \quad (3)$$

zwischen beiden Texturmodellen berechnet, $i = 1, \dots, k$. Die *Struktur-Metrik* wird definiert als

$$\delta := \sum_{i=1}^k \begin{cases} 1, & \text{falls } d_i > \epsilon \\ 10^{-6} \cdot d_i, & \text{sonst} \end{cases}, \quad (4)$$

wobei ϵ wieder eine einstellbare Konstante ist, die angibt, bis zu welcher Differenz die Intensitäts- bzw. Farbkanäle zweier Pixel nicht unterschieden werden sollen. In dieser Arbeit wird $\epsilon = 15$ verwendet. In der Formel wird im Wesentlichen gezählt, wieviele Elemente in s_*^x und s_*^t unterschiedlich sind. Die Fallunterscheidung in der Formel dient dazu, ähnliche Texturmodelle mit gleicher Zählung unterscheiden zu können.

x wird als *kompatibel zum Modell* s_*^t angesehen, wenn $\delta \leq c \cdot m$, wobei c ein voreinstellbarer Prozentsatz ist.

3.3 Bildsegmentierung

Bei der Bildsegmentierung wird in einer ersten untersuchten Verfahrensvariante genauso vorgegangen wie in Abschnitt 2.3 beschrieben, wobei zur Erstellung der Liste von Texturmodellen $c = 50\%$ verwendet wird und beim Einfärben der Pixel $c = 40\%$.

Im Gegensatz zum AR-Filter, bei dem kleine Verschiebungen von Bildtexturen nach Konstruktion zu keiner bedeutenden Veränderung des Texturmodells führen, hat dieses Vorgehen den potenziell entscheidenden Nachteil, dass sich ein Texturmodell s_*^{At} bei einer Verschiebung von Bildtexturen (oder bei Veränderung der Pixelposition t) stark ändern kann. Solche Veränderungen werden zwar über das Konstrukt mit dem Parameter c kompensiert, dieses sollte sich jedoch besser nur auf inhaltlich-strukturelle Unterschiede beziehen. Daher wird in einer zweiten untersuchten Variante des Verfahrens eine Verschiebungsinvarianz integriert: Einem Pixel x wird bei der Bildsegmentierung nicht das Texturmodell zugeordnet, das nur genau bei x zur kleinsten Strukturmetrik δ führt, sondern das Texturmodell der Liste, das in der Umgebung A_x von x zum kleinsten δ führt.

4 Histogramme von Farben und Farbübergängen

4.1 Texturmodell

Wiederum wird die kreisförmige Umgebung A_x um ein Test-Pixel x mit m Elementen betrachtet (vgl. Abschnitte 2.1 und 3.1). Es sollen Histogramme berechnet werden. Dazu werden die Grauwerte bzw. Farben reduziert. r sei eine benutzerdefinierte Farbreduktionskonstante und $k := \lfloor 255/r \rfloor + 1$. Der Grauwert eines Grauwertbildes bzw. ein Farbkanal eines Farbbildes zu Pixel $y \in A_x$ sei bezeichnet mit $g_y \in \{0, \dots, 255\}$ und die Farbreduktion sei $\bar{g}_y := \lfloor g_y/r \rfloor \in K$, $K := \{0, \dots, k-1\}$. Als *Texturmodell* zu x wird das Histogramm h_* reduzierter Farben verwendet, welches wie folgt fünf Teilhistogramme konkateniert:

$$h_{i+0k} := |\{y \in A_x \mid \bar{g}_y = i\}|, \quad (5)$$

$$h_{i+1k} := |\{y \in A_x \mid \bar{g}_{y_r} - \bar{g}_y = i\}|, \quad (6)$$

$$h_{i+2k} := |\{y \in A_x \mid \bar{g}_y - \bar{g}_{y_r} = i, i \neq 0\}|, \quad (7)$$

$$h_{i+3k} := |\{y \in A_x \mid \bar{g}_{y_u} - \bar{g}_y = i\}|, \quad (8)$$

$$h_{i+4k} := |\{y \in A_x \mid \bar{g}_y - \bar{g}_{y_u} = i, i \neq 0\}|, \quad (9)$$

$i = 0, \dots, k-1$. y_r bezeichne dabei das Nachbapixel rechts von y und y_u das Nachbapixel darunter.

Bei Farbbildern wird das für jeden Farbkanal gemacht und die drei sich ergebenden Teile ebenfalls konkateniert. Bei Grauwertbildern hat das Texturmodell h_* insgesamt $j := 5k$ und bei Farbbildern $j := 15k$ Einträge. In dieser Arbeit wird $r = 8$ verwendet, d.h. $k = 32$.

Anschaulich speichert h_{i+ak} , welche Intensitäten wie oft ($a = 0$) bzw. welche nach rechts/unten zunehmenden ($a = 1, 3$) und abnehmenden Gradienten ($a = 2, 4$) wie oft zwischen horizontalen ($a = 1, 2$) und vertikalen Nachbapixeln ($a = 3, 4$) auftreten, in Form eines Vektors $h \in K^j$.

4.2 Evaluation

Ein Texturmodell $h = h_*^t$ korrespondiert nach Konstruktion mit einem Pixel t , zu dessen Umgebung A_t es berechnet wurde. Zur Quantifizierung, inwieweit ein anderes Pixel x dem gleichen Texturmodell folgt, wird zunächst h_*^x zu x berechnet. Es sei $U_i := \{i-1, i+1\} \cap K$ und

$$\beta := \sum_{\substack{i=0, \dots, k-1 \\ a=0, \dots, 4}} \begin{cases} \gamma_{i+ak}, & \text{falls } \max_{u \in U_i} h_{u+ak}^x < h_{i+ak}^t \wedge \gamma_{i+ak} \geq \rho \\ 0, & \text{sonst} \end{cases} \quad (10)$$

mit dem Histogrammunterschied $\gamma_{i+ak} := h_{i+ak}^t - h_{i+ak}^x$, der wertet, wie viel in h_*^t von h_*^x unerreicht bleibt. Er wird in der Formel jedoch nicht gewertet, wenn er sich rein durch Bildrauschen erklären lässt, d.h. unter einer Rauschschwelle $\rho := 3$ bleibt. Durch die Farbreduktion können zudem bei Rauscheinflüssen an sich gleiche Bildstrukturen leicht in unterschiedlichen (aber benachbarten) Bins von h_* landen. Daher wird der Histogrammunterschied γ_{i+ak} in der Formel nur gewertet, wenn auch die Nachbarn h_{u+ak}^x innerhalb der betreffenden Teilhistogramme den Mangel bei h_{i+ak}^x nicht kompensieren können. Bei Farbbildern summiert β auch über die drei Farbkanäle.

Die *Struktur-Metrik* wird definiert als $\delta := \beta \cdot m / \sum_{i=0}^{j-1} h_i^t$. x wird als *kompatibel zum Modell* h_*^t angesehen, wenn $\delta \leq c \cdot m$, wobei c ein voreinstellbarer Prozentsatz ist. In dieser Arbeit wird $c = 30\%$ verwendet.

4.3 Bildsegmentierung

Die Bildsegmentierung erfolgt wie beim AR-Filter, siehe Abschnitt 2.3.

5 Ergebnisse

Abb 1 zeigt oben die verwendeten Originalbilder und darunter für $r_A = 10$ die erhaltenen Ergebnisse: erst zum AR-Filter ($r_B = 8$), darunter die Varianten 1 und 2 zur Schablonenanpassung und ganz unten zum Histogrammansatz. Wie man erkennen kann, eignen sich prinzipiell alle Verfahren/Varianten: insbesondere Wege, Bewuchs und Vegetationsunterschiede werden gut herauspräpariert.

Die beiden AR-Filter-Ergebnisse wirken unruhiger als die anderen, der AR-Filter wirkt also feingranularer. Die beiden Template-Varianten unterscheiden sich nicht substantiell, Variante 2 vermag vorteilhafterweise jedoch deutlich mehr und dichter zu segmentieren. Interessant ist, dass alle Verfahren/Varianten beim rechten Bild Übergänge im unteren linken Bereich feststellen, die man mit bloßem Auge nicht ohne Weiteres erkennen kann. Der Histogramm-Ansatz ist klar am besten: Er ist der einzige, der die Baumkronen geschlossen erkennt und sauber segmentiert. Auch der Pkw-Parkbereich im rechten Teil des linken Bildes wird am besten erfasst. Er segmentiert zudem die meisten Bildpixel.

Abschließend illustriert Abb. 2 das Verhalten des AR-Filter-Ansatzes bei unterschiedlichen Werten für r_B (von oben nach unten 2, 4, 6, 10 und 11) und Abb. 3 das des Histogramm-Ansatzes bei unterschiedlichen Werten für r_A (von oben nach unten 3, 5, 7, 9 und 12).

6 Zusammenfassung

Es wurden drei Ansätze zur Texturmodellierung bzw. Bildsegmentierung vorgestellt und verglichen. Alle zeigen sich prinzipiell als geeignet, jedoch liefert der Histogrammansatz klar die besten Ergebnisse. Er zeigt sich als vielversprechend für weitere geplante Optimierungen und Weiterverarbeitungsmöglichkeiten.

Literatur

1. A. Ruckstuhl, "Einführung in die robusten Schätzmethoden," Juni 2008.
2. F. Wicki, "Robuste Schätzverfahren für die Parameterschätzung in geodätischen Netzen," *Dissertation an der Eidgenössischen Hochschule (ETH) Zürich, Schweiz*, 1998.

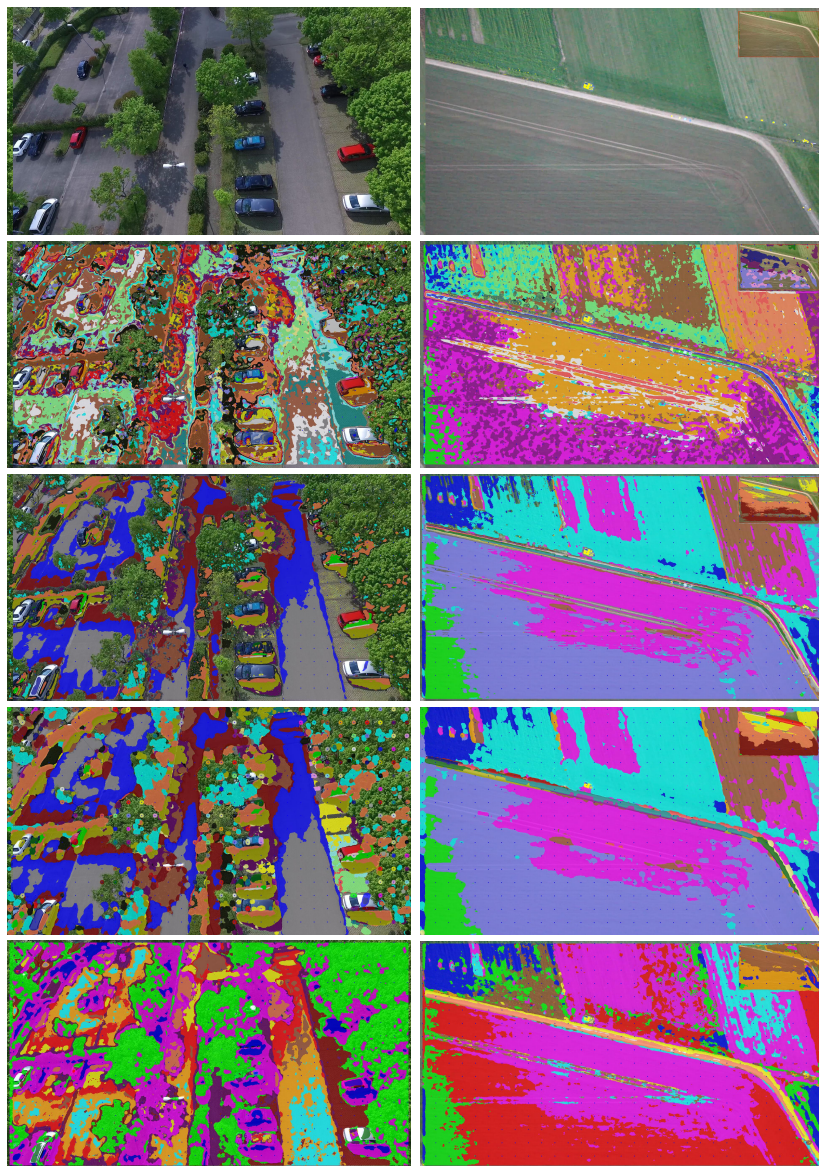


Abbildung 1: Testbilder (oben) und Ergebnisse für $r_A = 10$ (siehe Text).

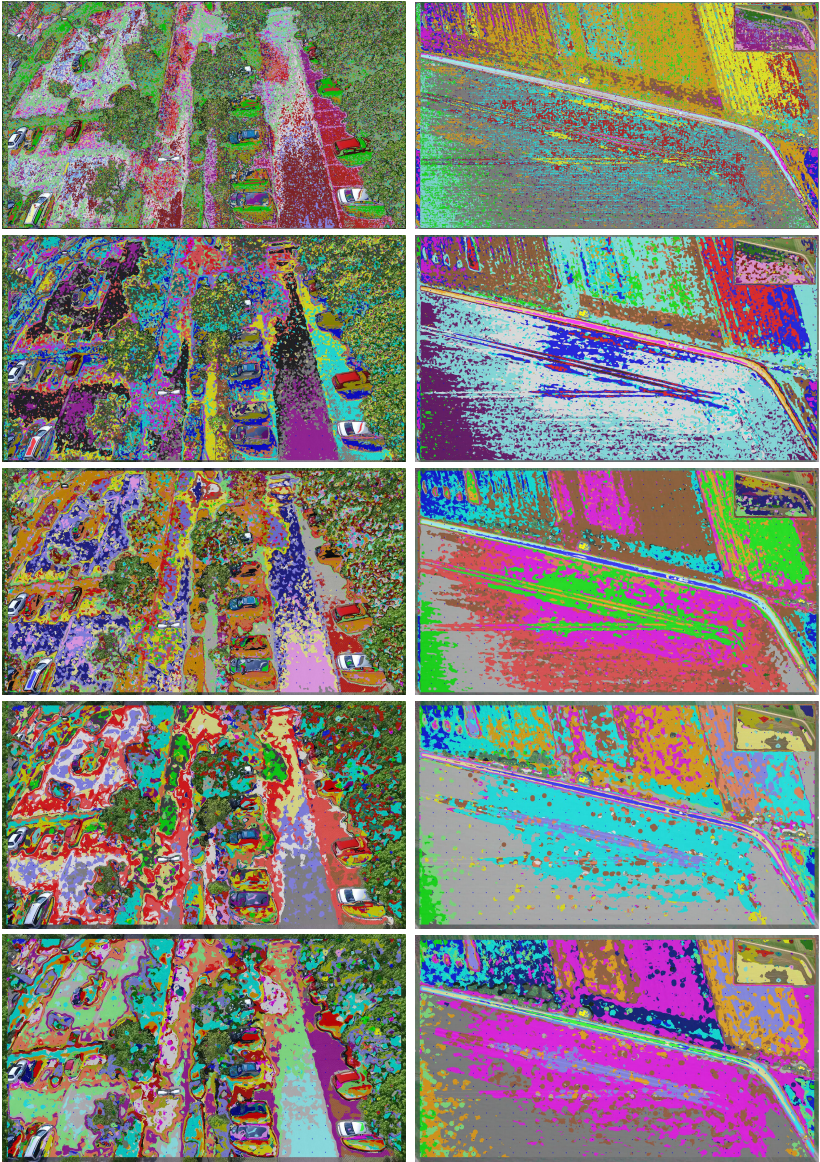


Abbildung 2: Ergebnisse zum AR-Filter für verschiedene r_B .

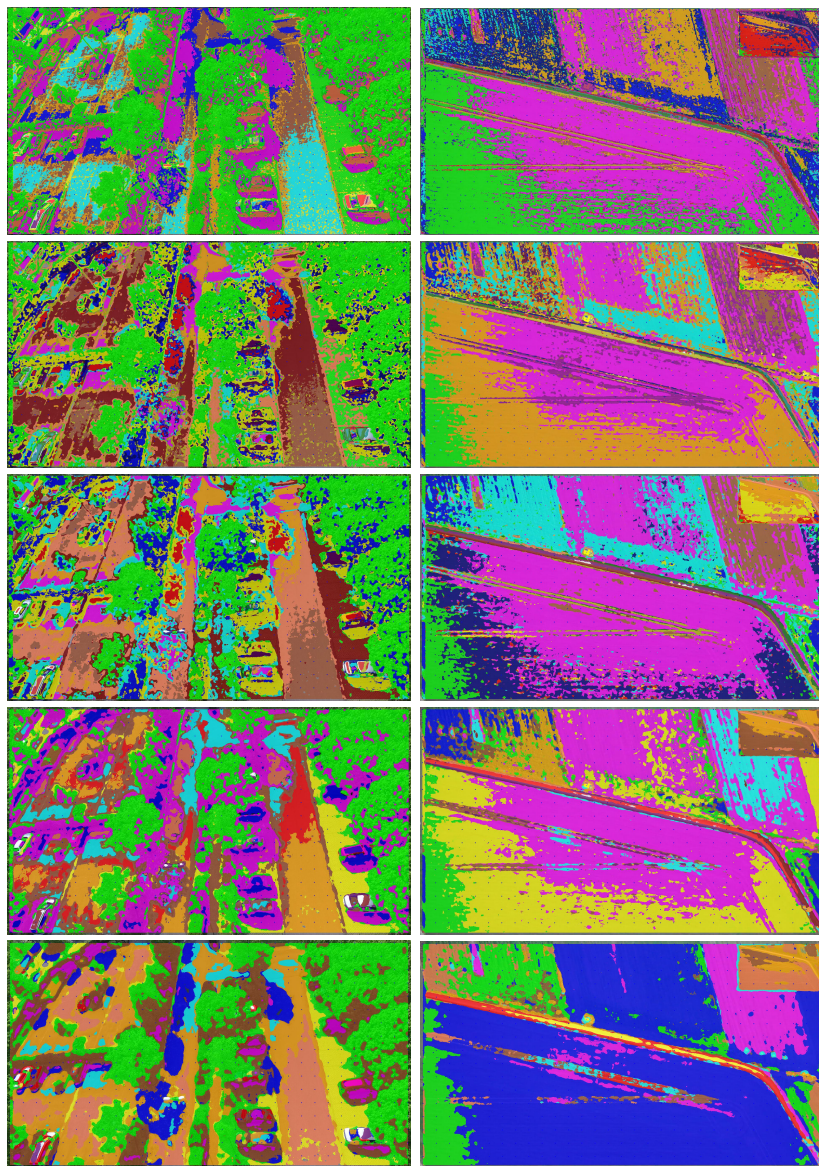


Abbildung 3: Ergebnisse zum Histogramm-Ansatz für verschiedene r_A .

Ein evolutionärer Ansatz für die automatische Ermittlung der Topologie neuronaler Netze

An evolutionary approach to automatically determine the topology of a neural network

Norbert Mitschke¹, Michael Heizmann¹, Klaus-Henning Noffz² und Ralf Wittmann²

¹ Karlsruher Institut für Technologie,
Institut für Industrielle Informationstechnik,
Hertzstraße 16, 76187 Karlsruhe

² Silicon Software GmbH
Konrad-Zuse-Ring 28, 68163 Mannheim

Zusammenfassung Infolge von sinkenden Hardware-Preisen und der zunehmenden Automatisierung wird maschinelles Lernen für industrielle Anwendungen wie klassischen Sichtprüfungsaufgaben immer attraktiver. In diesem Artikel wird ein metaheuristischer Ansatz für die automatische Ermittlung der Topologie eines neuronalen Netzes präsentiert, der auf differentieller Evolution basiert. Dieser ist in der Lage, anhand eines gegebenen Datensatzes und ohne zusätzliches Vorwissen einen geeigneten Klassifikator zu entwerfen. Gleichzeitig wird durch die Wahl einer geeigneten Fitnessfunktion der Ressourcenbedarf der Inferenz des neuronalen Netzes begrenzt bzw. minimiert. Für typische industrielle Datensätze kann mit dem Ansatz eine Topologie gefunden werden, die eine Genauigkeit von im Mittel über 98 % erreicht, während die Rechendauer relativ kurz bleibt.

Schlagwörter Neuronale Netze, Differentielle Evolution, Neuroevolution, Netzwerktopologie, Maschinelles Lernen.

Abstract As a result of falling hardware prices and increasing automation, machine learning is becoming increasingly attractive for industrial applications such as automatic visual inspection. This article presents a metaheuristic approach to automatically determine the topology of a neural network based on differential evolution for various applications on the basis of a given data set and without additional previous knowledge. Another design goal is to limit or minimize the resource requirements of the inference of the neural network. For typical industrial data sets, a topology can be found with this approach that achieves an accuracy of more than 98 % on average, while the processing time remains relatively short.

Keywords Neural networks, differential evolution, neuroevolution, structure learning, deep learning.

1 Einleitung

Neuronale Faltungsnetze (engl. *Convolutional Neural Network*, kurz: CNN) stellen ein heutzutage viel genutztes Verfahren zur Klassifikation von Bilddaten dar. Seit dem Durchbruch von AlexNet [1] beim ImageNet Wettbewerb 2012 wurden erhebliche Verbesserungen durch immer tiefere Netze in der Klassifikation von komplexen Szenen erreicht.

Die Suche nach einer geeigneten Netztopologie stellt dabei eine Herausforderung dar. Einerseits wird eine hohe Testgenauigkeit verlangt und andererseits sind kleine CNNs vorteilhaft für die Implementierung. In den 1990ern und 2000ern wurden relativ langsame Verfahren entworfen, die nach den optimalen Verbindungen und Gewichten zwischen Neuronen mittels Heuristiken suchen [2–4]. Moderne Verfahren setzen auf das aufwändigere Reinforcement-Learning [5], um tiefe CNNs für komplexere Anwendungen zu finden. Jedoch werden bei diesem Ansatz enorm hohe Rechendauern benötigt, da sehr viele CNNs hierfür mehrere Epochen trainiert werden.

Der Einsatz von CNNs in industriellen Anwendungen spielt bisher eine untergeordnete Rolle. Es werden immer noch meist klassische Methoden verwendet, obwohl der Einsatz von CNNs in ähnlichen Anwendungen bereits erfolgreich untersucht wurde.

Für die Oberflächeninspektion reichen kleine CNNs bereits aus. So wird in [6] ein Verfahren beschrieben, bei dem ein vortrainiertes CNN

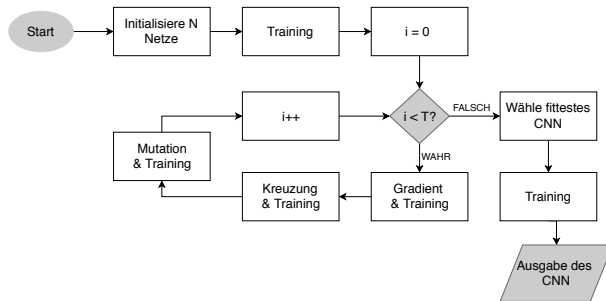


Abbildung 1: Ablauf der Metaheuristik.

mit fünf Schichten für die Inspection von Holz, Lötstellen und Stahl verwendet wird. In [7] werden Defekte wie Verschmutzung, Kratzer und Grate ebenfalls mit CNNs inspiziert. Dabei wird die Topologie durch manuelles Variieren von Neuronen und Schichten bestimmt. In [8] wird die Topologie eines CNN durch die beiden Parameter *Tiefe* und *Breite* beschrieben. Anschließend wird der Einfluss der beiden Hyperparameter auf die Genauigkeit bei der Defektdetektion auf einem künstlichen Datensatz untersucht.

Jedoch fehlt in diesen Beiträgen ein Automatismus für das Auffinden einer geeigneten Netztopologie. In diesem Beitrag wird ein metaheuristisches Verfahren vorgestellt, das aus einem gegebenen Datensatz ein neuronales Netz automatisch entwirft und trainiert. Der Beitrag ist folgendermaßen aufgebaut: In Kapitel 2 wird der Ansatz des Verfahrens vorgestellt. Kapitel 3 thematisiert den Testaufbau und die verwendeten Datensätze. Anschließend werden die Ergebnisse in Kapitel 4 und das Fazit in Kapitel 5 behandelt.

2 Methodik

Der Ansatz basiert auf differentieller Evolution [9], welcher nach [10] im Allgemeinen mindestens genauso gute Ergebnisse liefert wie eine Partikelschwarmoptimierung oder andere genetische Evolutionsstrategien. Ein CNN wird grundsätzlich durch die Anzahl und Art seiner Schichten, die zugehörigen Parameter und die Gewichte sowie Biases

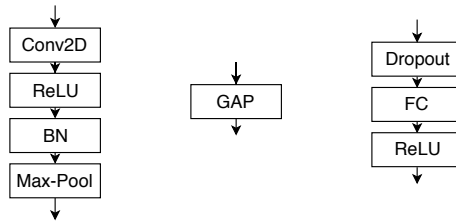


Abbildung 2: Die Abbildung zeigt die drei Bausteine des hier verwendeten CNN. Links sind der Block mit einer Faltungsschicht, in der Mitte die GAP-Schicht und rechts der Block um die FC-Schicht dargestellt. Die letzte FC-Schicht besitzt anstelle einer ReLU eine Softmax-Aktivierung.

beschrieben. Da die Anzahl und Art der Schichten variiert, variiert auch die Anzahl der Parameter, die das CNN beschreiben. Daher wird ein zweistufiger Ansatz gewählt, der die Schichten und seine Parameter getrennt innerhalb eines Iterationsschrittes bezüglich einer Fitnessfunktion optimiert. Der Algorithmus besteht aus einer Initialisierung, T Iterationsschritten und im Training des ausgewählten CNN am Ende. Der Ablauf ist in Abb. 1 illustriert.

In diesem Kapitel wird zunächst die allgemeine Topologie sowie die Initialisierung des vorgestellten Algorithmus dargestellt. Anschließend werden die Fitnessfunktion und dann die Rechenschritte während der Iteration vorgestellt.

2.1 Allgemeine Topologie und Initialisierung

Um die Konvergenz des Algorithmus zu beschleunigen, muss der Suchraum klein gehalten werden. Dies führt zu Einschränkungen in der Topologie des Netzes: Es werden lediglich drei verschiedene Blöcke für die Topologie verwendet. Im Kern eines Blocks steht eine vollständig verbundene Schichten (engl. *fully connected layer*, kurz: FC), eine 2D-Faltungsschichten (engl. *convolutional layer*, kurz: Conv2D) oder eine globale Mittelungsschicht (engl. *global averaging pooling layer*, kurz: GAP). Faltungsschichten werden um eine Bündel-Normalisierungsschicht (engl. *batch normalization layer*, kurz: BN) [11], eine Maximum-Pooling-Schicht (kurz: Max-Pool) und um eine ReLU-

Aktivierungsfunktion ergänzt. Der FC-Baustein besteht aus jeweils einer Ausfallschicht (engl. *dropout layer*), einer FC-Schicht und einer ReLU. Diese drei Blöcke sind in Abb. 2 abgebildet. Beim Aufbau werden zunächst die Faltungsblöcke, deren Anzahl auf maximal 10 beschränkt ist, hintereinander geschaltet, bevor die GAP-Schicht den Ausgabensor in einen Vektor formt. Anschließend werden maximal vier FC-Blöcke hinzugefügt.

Für die Optimierung können in FC-Schichten die Anzahl der Neuronen und in den Faltungsbausteinen die Anzahl der Kernel, die Schrittweite der Filter und das Maximum-Pooling verändert werden. Die Filterkernelgröße (3×3) und andere Parameter sind hingegen fest. Ferner wird eine L_2 -Norm der Gewichte als zusätzliche Regularisierung verwendet.

Bei der Initialisierung werden zufällig $\frac{N}{2}$ Topologien aus einer Liste mit Initialtopologien ausgewählt. Anschließend werden die CNNs mit dem Datensatz b_0 Epochen trainiert, die Gewichte gespeichert und die CNNs dupliziert, sodass die Population aus N Individuen besteht.

2.2 Fitnessfunktion

Das Ziel der Metaheuristik ist es, die Fitnessfunktion zu optimieren. Wie zuvor beschrieben bildet die Fitnessfunktion die Validierungsgenauigkeit und den Ressourcenbedarf einer Topologie ab. Dieser setzt sich aus dem benötigten Speicher s für Gewichte, Biases und Aktivierungen und den MAC-Operationen m der Inferenz zusammen.

Empirisch wurde als Fitnessfunktion die Funktion $F(a, \eta, m, s)$ mit

$$F(a, \eta, m, s) = \frac{1}{1 - a + \epsilon} + C \cdot a + \frac{1}{\eta + \epsilon} - \frac{m}{M} \quad \text{für } s \leq s_{\max} \quad (1)$$

ermittelt. Hierbei stellt a die Klassifikationsgenauigkeit des Validierungsdatensatzes, η die Kreuzentropie, M sowie C Skalierungsfaktoren und $s_{\max}$ den maximal zur Verfügung stehenden Speicherplatz dar. ϵ dient zur numerischen Stabilität. Die Fitness von CNNs mit $s > s_{\max}$ wird formal zu $F(s > s_{\max}) = \infty$ definiert, was durch den begrenzten Speicherplatz auf mobilen Geräten oder FPGAs motiviert wird.

Die einzelnen Terme der Fitnessfunktion haben die Aufgabe zu verhindern, dass die gesamte Fitness nur durch Verkleinerung des Netzes,

d. h. durch Reduktion von s und m , erhöht wird, während die Validierungsgenauigkeit bei niedrigen Werten verharret. Mit der inversen Validierungsgenauigkeit wird erreicht, dass bei hohen absoluten Validierungsgenauigkeiten noch ein Anstieg in der Fitness verzeichnet werden kann. Bei geringer Genauigkeit sorgt hierfür der lineare Term $C \cdot a$. Falls die Klassen stark unterschiedlich viele Objekte beinhalten, dauert es u. U. mehrere Trainingsdurchläufe, bis sich die Genauigkeit erhöht. Daher wird ebenfalls die inverse Kreuzentropie η verwendet.

2.3 Iteration

Der Übergang einer Population $\mathfrak{P}_t$ in die nächste Generation $\mathfrak{P}_{t+1}$ besteht drei Schritten: Im ersten Schritt wird innerhalb der Population ein Gradient der Fitness bezüglich der Schichten gebildet. Danach folgt eine Kreuzung innerhalb der CNNs mit gleichen Schichten. Der dritte Schritt ist eine zufällige Mutation, um die Diversität der Population zu erhöhen. Nach jedem der drei Schritte wird das sich neu ergebende CNN für b_t Epochen trainiert, um die Gewichte an die Veränderung anzupassen und die Klassifizierungsgenauigkeit zu erhöhen. Die Gewichte werden während des Prozesses für jede Topologie gespeichert und vor dem Training in das veränderte CNNs geladen, damit der erreichte Fortschritt erhalten bleibt.

Die Gewichte neuer Schichten werden zufällig initialisiert, während die Gewichte entfernter Schichten ebenfalls gelöscht werden. Falls sich die Anzahl an Neuronen bzw. an Filterkernen ändert, werden aus der Gewichtsmatrix zufällig Zeilen entfernt bzw. Zeilen mit zufälligen Einträgen hinzugefügt. Darüber hinaus ist es nötig, in der darauf folgenden Schicht sowohl die Anzahl der Spalten der Gewichtsmatrix als auch den Bias-Vektor entsprechend anzupassen. Die Anzahl der Neuronen in der letzten FC-Schicht kann nicht verändert werden.

Individuen mit der gleichen Anzahl an Faltungsschichten c und FC-Schichten d innerhalb der Population $\mathfrak{P}_t$ werden zur Gattung $\mathfrak{G}_t^{(c,d)}$ zusammengefasst.

Im ersten Schritt wird für jede Gattung $\mathfrak{G}_t^{(c,d)}$ ein Individuum i ausgewählt und der Gradient $\vec{g}_i$ seiner Fitness $F(i)$ bezüglich der Anzahl der Schichten gebildet. Als Referenz wird eine zufällig ausgewählte Gruppe $\tilde{\mathfrak{P}}_t$ von $\tilde{N}$ Individuen aus $\mathfrak{P}_t$ herangezogen. Der Gradient wird

gewählt zu:

$$\vec{g}_i = \frac{1}{\tilde{N}} \sum_{j \in \mathfrak{P}_t} (F(i) - F(j)) \cdot \left(\begin{bmatrix} c_i - c_j \\ d_i - d_j \end{bmatrix} \right). \quad (2)$$

Anschließend wird das Individuum i angepasst:

$$\begin{bmatrix} c_i \\ d_i \end{bmatrix} \leftarrow \text{srd} \left(\begin{bmatrix} c_i \\ d_i \end{bmatrix} + r \cdot \vec{g}_i \right), \quad (3)$$

wobei $\text{srd}(\cdot)$ die stochastische Rundungsoperation meint und $r \sim U(0, \frac{1}{2})$ eine gleichverteilte Zufallsvariable ist.

Im zweiten Schritt werden jeweils zwei Individuen aus der gleichen Gattung $\mathfrak{G}_t^{(c,d)}$ – falls vorhanden – durch Roulette-Wheel-Selektion [12] ausgewählt und gekreuzt. Die Anzahl an Filterkernen und Neuronen des Nachkommen wird entsprechend den Gleichungen 2 und 3 berechnet. Die Schrittweiten von Faltungen und von Max-Pooling werden durch eine Zwei-Punkte-Kreuzung bestimmt, d. h. die Schrittweiten werden abschnittsweise von den Eltern übernommen. Der Nachkomme ersetzt das am wenigsten fitte Individuum aus seiner Gattung, wenn seine Fitness nach den b_t Trainingsepochen höher ist.

Eine zufällige Mutation stellt den letzten Schritt dar. Hierbei werden die besten zwei Individuen von $\mathfrak{P}_t$ ausgewählt. Anschließend werden die optimierbaren Werte zufällig verändert. Die so generierten Nachkommen ersetzen die schlechtesten Individuen der Population.

3 Versuchsaufbau

Die vorgestellte Metaheuristik wird an den Datensätzen, die in diesem Kapitel vorgestellt werden, fünfmal evaluiert. Der Algorithmus wird jeweils mit $\frac{N}{2} = 15$ Individuen initialisiert, die $b_0 = 8$ Epochen trainiert werden. Die Parameter für die Fitnessfunktion aus Gl. 1 sind $C = 10$, $M = 10^6$ und $s_{\max} = 27$ MB in Anlehnung an die verfügbaren Ressourcen moderner FPGAs. Ferner ist $\tilde{N} = \lfloor \frac{N}{3} \rfloor$. Die Bündelgröße (engl. *batch size*) wird zu 64 gesetzt. Um die Konvergenz zu beschleunigen, werden am Ende jeder der $T = 10$ Iterationen die beiden schlechtesten Individuen aus der Population entfernt (vgl. [10]) und die Anzahl der Trainingsdurchläufe linear von $b_1 = 4$ zu Beginn auf $b_{10} = 6$ erhöht.

Der NEU-Oberflächendefekt-Datensatz [13] besteht aus sechs Klassen von Oberflächendefekten auf Stahl mit jeweils 300 Grauwertbildern der Größe 200×200 Pixeln. Die Bilder werden auf 192×192 Bildpunkte interpoliert und es wird ein Testdatensatz von 180 und ein Validierungsdatensatz von 360 Bildern angelegt. Die restlichen 1.260 Bilder werden durch Rotation, Translation, Rauschen und Variation in der Beleuchtung um den Faktor 4 augmentiert und als Trainingsdatensatz verwendet.

Im RSDDs-Datensatz [14] sind Aufnahmen von Defekten auf Eisenbahnschienen enthalten. Die 128 verwendeten Bilder haben eine Größe von 55×1250 Pixeln. Für die Zwecke der Defekterkennung werden Ausschnitte von 55×55 Pixeln aus den Bildern extrahiert und entsprechend der Ground-Truth klassifiziert. Damit ergibt sich ein Trainingsdatensatz von 2.342, ein Validierungsdatensatz von 563 und ein Testdatensatz von 302 Bildausschnitten, von denen jeweils ca. 12 % einen Defekt enthalten.

Die optische Erkennung von Defekten an Löt- und Schweißstellen ist die Klassifikationsaufgabe des Welds-Datensatzes [15], welcher sowohl Röntgenbilder der Lötstellen als auch die Ground-Truth der Defekte enthält. Ähnlich wie beim RSDDs-Datensatz werden insgesamt etwa 9.000 Ausschnitte der Größe 64×64 aus den Röntgenbildern als Grauwertbild extrahiert und klassifiziert, bevor sie als Datensatz benutzt werden können.

Der MNIST-Datensatz [16] besteht aus 60.000 Trainings- und 10.000 Testbildern von handgeschriebenen Ziffern. Der Trainingsdatensatz wird unterteilt in 47.000 Bilder zum Trainieren und 13.000 Bilder für die Validierung. Jedes Bild wird von 28×28 auf 32×32 hochskaliert.

Als weiterer Datensatz wird der Fashion-MNIST [17] verwendet. Dieser stellt zwar keine industrielle Anwendung dar, gilt aber als ein Benchmarkdatensatz und ist mit industriellen Datensätzen vergleichbar. Der Datensatz zeigt Grauwertbilder von zehn verschiedenen Klassen von Kleidung. Der Datensatz enthält 70.000 Bilder der Größe 28×28 , die identisch wie der MNIST-Datensatz aufgeteilt und hochskaliert werden.

Zuletzt wird noch ein künstlicher Datensatz der DAGM untersucht. Dieser ist der fünfte Datensatz aus dem Wettbewerb für die optische Inspektion aus dem Jahr 2007 [18]. Es stehen 1.150 Grauwertbilder einer synthetischen Textur zu Verfügung, wobei 150 Bilder einen Defekt enthalten. Die Größe der Bilder beträgt 512×512 Pixel.

4 Ergebnisse

Datensatz	NEU	RSDDs	Welds	MNIST	Fashion	DAGM
Max. Genauigkeit	>0,999	0,989	0,974	0,994	0,923	>0,999
Mittlere Genauigkeit	0,997	0,985	0,966	0,992	0,913	0,997
Mittlere Laufzeit	251 min	240 min	159 min	151 min	176 min	142 min
Anzahl Parameter	12.868	26.561	25.266	133.980	134.162	9.724
MAC-Operationen	2,87 Mio.	0,78 Mio	2,19 Mio.	0,72 Mio.	1,01 Mio.	0,86 Mio.

Tabelle 1: Zusammenfassung der Ergebnisse des Verfahrens für die sechs Datensätze zusammen. Aus den fünf Durchläufen sind der minimale und der mittlere Klassifizierungsfehler bzgl. des Testdatensatzes, die mittlere Laufzeit und die im Mittel benötigten Ressourcen dargestellt.

Der vorgestellte Algorithmus wurde in Tensorflow (Python) implementiert. Das Training wurde auf einer *GeForce 1080 Ti* durchgeführt.

Die Ergebnisse sind in Tabelle 1 zusammengefasst und der Verlauf der Median-Genauigkeit in Abb. 3. Auffällig hierbei ist, dass die Median-Fitness beim NEU- und beim DAGM-Datensatz stärker steigt als bei den anderen und dass CNNs für Datensätze mit größeren Bildern tendenziell mehr MAC-Operationen und weniger Parameter benötigen, da die zugehörige Inferenz einen höheren Ressourcenverbrauch aufweist. Die Fitness von Welds und Fashion-MNIST steigt nur sehr langsam, da die erreichbare Genauigkeit deutlich geringer ist (s. Tab. 1).

Beim NEU-Datensatz werden im Median Genauigkeiten von 99,7 % erreichbar, bei relativ hohen Laufzeiten der Heuristik von im Mittel 251 min. Die Anzahl der MAC-Operationen variiert zwischen 1,7 Mio. und 4,6 Mio und die der Parameter zwischen 7.158 und 57.034, wobei der letzte Wert einen Ausreißer darstellt. Das korrespondierende CNN erreicht als einziges nicht eine Genauigkeit von 1.

Beim RSDD-Datensatz werden Genauigkeiten von 98 % bis 99 % erreicht. Die resultierenden Netze haben 5 bis 6 Faltungs- und 1-2 FC-Schichten. Der Welds-Datensatz verhält sich ähnlich, erreicht aber eine etwas geringere Testgenauigkeit.

Mit 99,38 % Testgenauigkeit erzielt das beste CNN bzgl. des MNIST-Datensatz eine Genauigkeit, die sich im Bereich des Standes der Technik bewegt. Das CNN besitzt acht Faltungs- und zwei FC-Schichten, hat 133.980 Parameter und die Inferenz benötigt ca. 0,72 Mio. MAC-

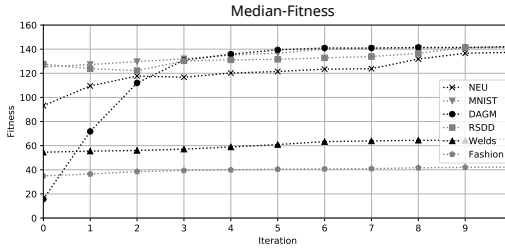


Abbildung 3: Verlauf der mittleren Median-Fitness bzgl. der fünf Durchläufe für jeden der sechs Datensätze.

Operationen. Damit stellt dieses Netz einen Ausreißer bzgl. der benötigten Ressourcen und der Genauigkeit dar. Die übrigen Netze besitzen lediglich 4 oder 5 Faltungsschichten.

Ebenso stellen Topologien, die für den Fashion-MNIST-Datensatz Genauigkeiten von über 91 % erreichen, einen Erfolg dar. Alle gefundenen Netze haben 3-6 Faltungen und eine FC-Schicht.

Die Defekterkennung im DAGM-Datensatz ist eine vergleichsweise einfache Herausforderung, bei der weniger Parameter notwendig sind (s. Tabelle 1). Nach Abb. 3 steigt die mediane Fitness stark und kontinuierlich an, während die mittlere Maximalfitness stufenförmig ansteigt. Dies liegt daran, dass der zu detektierende Fehler ein rotierter und verschobener Patch ist, dessen jeweilige Erscheinungsform sukzessive durch den Klassifikator erlernt wird.

Für ein weiteres Experiment werden die besten CNNs extrahiert, die Gewichte neu initialisiert und anschließend 100 Epochen trainiert. Für kein CNN kann die Genauigkeit aus der Metaheuristik erreicht werden. Beispielsweise erreicht das beste CNN des Fashion-MNIST Datensatzes zwar eine Trainingsgenauigkeit von über 99,95 %, aber in der Validierung und beim Testen können keine Genauigkeiten von über 89 % erreicht werden. In diesem Fall neigt die Topologie bei einer Neuinitialisierung zu starker Überanpassung gegenüber dem Resultat aus der Heuristik. Dies zeigt, dass die Metaheuristik nicht nur nach einer Topologie sucht, sondern implizit auch nach einer geeigneten Initialisierung der Gewichte (vgl. [19]), da durch die evolutionären Operationen das jeweilige *basin of attraction* aufgrund des Hinzufügens bzw. Entfernens

von Gewichten verlassen wird.

5 Fazit

In diesem Beitrag wird eine neue Metaheuristik, die auf differentieller Evolution beruht, für die Optimierung einer CNN-Topologie vorgestellt. Für relativ einfache Datensätze, wie sie in industriellen Anwendungen oft vorliegen, können Topologien mit hohen Genauigkeiten, die nahe am Stand der Technik liegen, mit relativ kurzer Rechenzeit gefunden werden. Desweiteren garantiert diese Methode einen geringen Ressourcenbedarf, sodass das Ergebnis auf mobilen Geräten oder FPGAs implementiert werden kann.

Für komplexere Aufgaben muss zwischen zusätzlicher Rechendauer und langsamerer Konvergenz abgewogen werden. Beispielsweise kann die Anzahl der Trainingsepochen, die Größe der Population oder die Anzahl an Iterationen erhöht werden. Ferner kann durch Verwenden vortrainierter Netze oder Transferlernen Rechenzeit eingespart werden.

Literatur

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.
2. K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.
3. F. H.-F. Leung, H.-K. Lam, S.-H. Ling, and P. K.-S. Tam, "Tuning of the Structure and Parameters of a Neural Network Using an Improved Genetic Algorithm," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 79–88, 2003.
4. D. Floreano, P. Dürr, and C. Mattiussi, "Neuroevolution: from Architectures to Learning," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 47–62, Mar 2008. [Online]. Available: <https://doi.org/10.1007/s12065-007-0002-4>
5. B. Zoph and Q. V. Le, "Neural Architecture Search with Reinforcement Learning," *arXiv preprint arXiv:1611.01578*, 2016.
6. R. Ren, T. Hung, and K. C. Tan, "A Generic Deep-Learning-based Approach for Automated Surface Inspection," *IEEE Transactions on Cybernetics*, 2017.

7. J.-K. Park, B.-K. Kwon, J.-H. Park, and D.-J. Kang, "Machine Learning-based Imaging System for Surface Defect Inspection," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 3, no. 3, pp. 303–310, 2016.
8. D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of Deep Convolutional Neural Network Architectures for Automated Feature Extraction in Industrial Inspection," *CIRP Annals*, vol. 65, no. 1, pp. 417–420, 2016.
9. R. Kuo and F. E. Zulvia, "The Gradient Evolution Algorithm: A new Metaheuristic," *Information Sciences*, vol. 316, pp. 246–265, 2015.
10. K.-L. Du and M. Swamy, *Search and Optimization by Metaheuristics*. Springer, 2016.
11. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.
12. J. H. Holland, *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT press, 1992.
13. K. Song, S. Hu, and Y. Yan, "Automatic Recognition of Surface Defects on Hot-Rolled Steel Strip using Scattering Convolution Network," *Journal of Computational Information Systems*, vol. 10, no. 7, pp. 3049–3055, 2014.
14. J. Gan, Q. Li, J. Wang, and H. Yu, "A Hierarchical Extractor-Based Visual Rail Surface Inspection System," *IEEE Sensors Journal*, vol. 17, pp. 7935–7944, 2017.
15. D. Mery, V. Rizzo, U. Zscherpel, G. Mondragón, I. Lillo, I. Zuccar, H. Lobel, and M. Carrasco, "GDXray: The Database of X-ray Images for Nondestructive Testing," *Journal of Nondestructive Evaluation*, vol. 34, no. 4, p. 42, 2015.
16. Y. LeCun, C. Cortes, and C. Burges, "MNIST Handwritten Digit Database," AT&T Labs [Online]. Available: <http://yann.lecun.com/exdb/mnist>, vol. 2, 2010.
17. H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
18. Deutsche Arbeitsgemeinschaft für Mustererkennung. (2007) Weakly Supervised Learning for Industrial Optical Inspection. [Online]. Available: <https://resources.mpi-inf.mpg.de/conference/dagm/2007/prizes.html#industry>
19. D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why Does Unsupervised Pre-Training Help Deep Learning?" *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 625–660, 2010.

Towards explaining deep learning networks to distinguish facial expressions of pain and emotions

Katharina Weitz^{1,2}, Teena Hassan¹, Ute Schmid², and Jens Garbas¹

¹ Fraunhofer IIS, Intelligent Systems Group,
Am Wolfsmantel 33, 91058 Erlangen

² University of Bamberg, Cognitive Systems Group,
An der Weberei 5, 96047 Bamberg

Abstract Deep learning networks are successfully used for object and face recognition in images and videos. In order to be able to apply such networks in practice, for example in hospitals as a pain recognition tool, the current procedures are only suitable to a limited extent. The advantage of deep learning methods is that they can learn complex non-linear relationships between raw data and target classes without limiting themselves to a set of hand-crafted features provided by humans. However, the disadvantage is that due to the complexity of these networks, it is not possible to interpret the knowledge that is stored inside the network. It is a black-box learning procedure. Explainable Artificial Intelligence (AI) approaches mitigate this problem by extracting explanations for decisions and representing them in a human-interpretable form. The aim of this paper is to investigate the explainable AI method Layer-wise Relevance Propagation (LRP) and apply it to explain how a deep learning network distinguishes facial expressions of pain from facial expressions of emotions such as happiness and disgust.

Keywords Explainable artificial intelligence, deep learning, emotion recognition, pain recognition.

1 Introduction

Facial expressions are one of the most important human nonverbal signals in interacting with other people and thus contribute to the emer-

gence and maintenance of social relationships [1]. One of the tasks of facial expressions is to communicate emotions [2,3]. This is of particular importance when people are unable to express themselves verbally (e. g., because of illness, accidents or congenital disabilities). For this reason, nursing staff in clinics and care facilities are required to observe patients closely in order to be able to read their emotions and take action, if necessary. Due to the already significantly high number of patients, especially in nursing homes, and the prognosis that more and more people will be cared for in such facilities in the future, it would be beneficial to deploy a system to support the nursing staff to monitor a patient's facial expressions and alert them when a pain episode is detected. Additionally, humans often have problems in differentiating between pain and other facial expressions [4,5]. Therefore, in addition to the (classical) exploration of emotions in a psychological context, research into a technical solution for distinguishing emotions and pain has gained greater importance in the last decade. A system, which uses explainable AI methods to describe how pain differs from other emotions, can be used to train nursing staff to improve their ability to recognise pain correctly.

Towards this goal, in this paper we examine and apply the LRP method [6] to explain the decisions made by a deep Convolutional Neural Network (CNN) that is trained to distinguish facial expressions of pain, happiness, and disgust.

2 Related work

One of the deep learning architectures that has been successfully applied to image processing applications is CNN [7], which processes images in a hierarchical manner [8]. Compared to approaches based on explicit facial activity descriptors [9], the features of deep learning do not have to be handcrafted. Instead, the system learns the features by itself by projecting information from bitmaps into so-called convolutional layers. They can learn non-linear relationships to model dependencies among the features [10]. One disadvantage is that deep learning approaches require a lot of sample data to extract features [11]. This problem can be reduced by data augmentation methods such as flipping or rotation. The other disadvantage is that, due to its complexity, it is no

longer comprehensible for humans, what the network has learned and what it bases its predictions on. In a practical application, it could be shown that these systems were not going to be accepted, because people do not blindly trust a system which they do not understand [12]. Therefore, techniques that make the black-box learning comprehensible to humans are necessary. One of these techniques for explaining the black-box deep learning is called Layer-wise Relevance Propagation (LRP), which explains the network's decisions by pixel-wise decomposition [6]. In facial image analysis, LRP can be used to explain which pixels were important for the decision of the network. For this, LRP uses the model's prediction strategies in the given pre-processing conditions and visualises the results of this computation as heatmaps [13]. When applying LRP, different parameters can be set to improve the resulting heatmap. In some cases, the resulting relevance scores for each pixel generated by LRP can take on unbounded values [6]. To adjust and stabilize the relevance scores, an ε value can be used. Additionally, α and β values can be applied for stabilization. Besides the stabilizing effect, α and β values can be used to visualize positive and negative activations of pixels [6]. With different values for α and β , the strength of the influence of positive (α) and negative (β) portions can be controlled [6, 14]. Besides these parameters, [15] showed that a 'preset' variant of the LRP algorithm achieves optimal results in the calculation of relevance maps. Using the preset approach, the relevance scores for all neurons of the lowest (first) layer are uniformly distributed to the input neuron instead of using the α and β values [13]. To control the resolution of the heatmaps generated by LRP, [16] describes an approach for 'mapping influence cut-off point'. This point describes the moment from which the forward mapping function of the classifier no longer influences relevance propagation, since only the receptive field of the classifier is relevant. The cut-off at this point is called the 'flat' rule. The reference of a receptive field is adapted from neuroscience [17]. In a CNN, the convolutional and pooling layers are inspired by the biological receptive field [11].

3 Research questions

Answers must be found for the following research questions for automatic pain classification becoming applicable in real-life settings:

- **Predictive performance:** How well can facial expressions of pain be automatically distinguished from those of disgust and happiness using self-learned spatial features?
- **Decision interpretation:** How can the decisions made by the model be presented to people in a comprehensible and transparent way?
- **Feature explanation:** How do the self-learned features differ for the facial expressions of pain and those of disgust and happiness?

This paper³ would like to provide answers of the questions above. For this, a pre-trained VGG-Face model [18] implemented with the Keras framework was finetuned to distinguish pain from happiness and disgust. For the fine-tuning images of the BioVid dataset⁴ [19] were used. Then, the Keras implementation of the LRP approach⁵ [20] was used to generate heatmaps at pixel level to illustrate which pixels were relevant for the classification by the VGG-Face model.

4 Material and procedure

The procedure of this study consisted of the following steps: First, data preparation was done on the BioVid dataset [19]. After that, the VGG-Face architecture was used to fine-tune the CNN for the three-class problem of distinguishing pain from happiness and disgust.

The BioVid dataset [19] was used for finetuning the VGG-Face model. Frames were extracted from the video sequences of pain, happiness, and disgust. For the class pain, part A (pain stimulation without facial EMG) was used. For the class disgust and happiness, frames from

³ This paper is based on the master's thesis of the first author submitted on August, 31, 2018 to the University of Bamberg. Online link: <https://www.uni-bamberg.de/en/cogsys/research/theses/advised-theses/>

⁴ <http://www.iikt.ovgu.de/BioVid.html>

⁵ <https://github.com/albermax/investigate>

video sequences in part D (posed pain & basic emotions) were used. The video sequences for the pain condition are each 5 seconds long (24,012 frames), the video sequences for the emotions are each 1 minute long (114,076 frames for disgust and 112,575 frames for happiness). The dataset was balanced by manually selecting 3×10^7 frames from each of the happiness and disgust sequences. One subject in the condition ‘disgust’ turned away from the camera and talked to the study leaders and showed no disgust expression. This subject was removed from the dataset. In Table 1, the amount of frames for each class after the data cleaning steps is provided. The extracted BioVid dataset was then used

Table 1: Extracted BioVid data after data cleaning steps.

Part	Name	Subjects	Frames
Part A	Pain intensity 3	87	12,006
	Pain intensity 4	87	12,006
Part D	Disgust	75	24,075
	Happiness	75	24,075

to fine-tune VGG-Face. For the implementation, Tensorflow (version 1.8) and Keras (version 2.2.0) were used. After that, the explainable AI method LRP was applied. For LRP, the Keras implementation from [20] was adapted.

5 Results

The VGG-Face CNN was fine-tuned and tested using 5-fold cross-validation. Here, 4 folds were used for training the model and the remaining fold was used to test the model. The best performing fold had an accuracy of 0.67 on the test data, and was used for generating explanations using the LRP method. In Table 2, the class-wise performance of the best fold is presented. When looking at the confusion matrix (see Figure 1), it becomes clear that the CNN had problems to classify happy faces as happy. 28% of the happy images were classified as pain. To take a closer look at this problem, the LRP approach was used. It was used to get an insight of the pixel-related areas of the image which were important for decisions of pain and happiness. To gain this insight, two

Table 2: Class-wise results of the best performing fold.

	Precision	Recall	F1-score	#Images
Pain	0.62	0.69	0.66	4692
Disgust	0.70	0.73	0.71	4815
Happiness	0.67	0.57	0.62	4815
Average/Total	0.67	0.66	0.66	14322

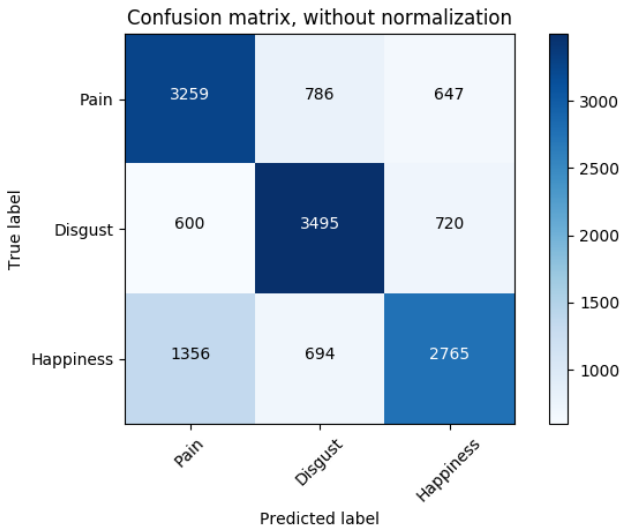


Figure 1: Confusion matrix without normalization for the best fold of the 5-fold cross-validation.

test images from the pain category were selected from the BioVid test fold. In Figure 2, the first image displays the subject experiencing pain intensity 3, and in the second image the same person experiencing pain intensity 4. The first label above the image refers to the true class, and the second label to the predicted class.

In Figure 3, the visualization generated using LRP with different parameters are shown. In LRP-Z, the basic LRP approach without stabi-



Figure 2: Original image from the test fold of the BioVid dataset. First label indicates the true class, second label indicates the predicted class.

lizers is applied. Here, a face is roughly recognizable. The noise due to the absence of stabilizers is present. In LRP-Epsilon, an ε value of $1e-7$ is used. In comparison to the basic LRP approach (LRP-Z), much less noise is represented here. For the LRP-PresetAFlat, an α value of 1, a β value of 0, and an ε value of $1e-1$ are used. The flat rule is also applied. For LRP-PresetBFlat, an α value of 2, a β value of 1, and an ε value of $1e-1$ are used. Again, the flat rule was applied. In both visualizations, red pixel indicate a positive contribution to the predicted class, and blue pixel indicate a negative contribution. In comparison to the LRP-Z heatmap, the visualization of preset-flat variants are much more detailed and clearer. In the two preset variants, it can be observed that highly positive pixel values are more important for the decision of the CNN. It becomes apparent that with the increase of the α value (LRP-PresetAFlat), the positive pixel values become more prominent. With the increase of the β value (LRP-PresetBFlat), the negative pixel values are highlighted more clearly. When looking at the LRP PresetAFlat visualization, it can be seen that mostly the same areas in the face, namely the eyes, the nose and the mouth contribute to the classification of happiness and pain. This could be an indication why the accuracy of the CNN is not very high. When looking at the LRP PresetBFlat, slight differences in the contribution of negative pixels for the classification are visible. For pain, more negative pixels around the nostrils and on the lower side of the eyebrows are detectable on the heatmaps.

Besides test images from the BioVid dataset, images from the UNBC-McMaster shoulder pain expression archive database [21] for pain were used for visualization using LRP methods (see left part of Figure 4). For

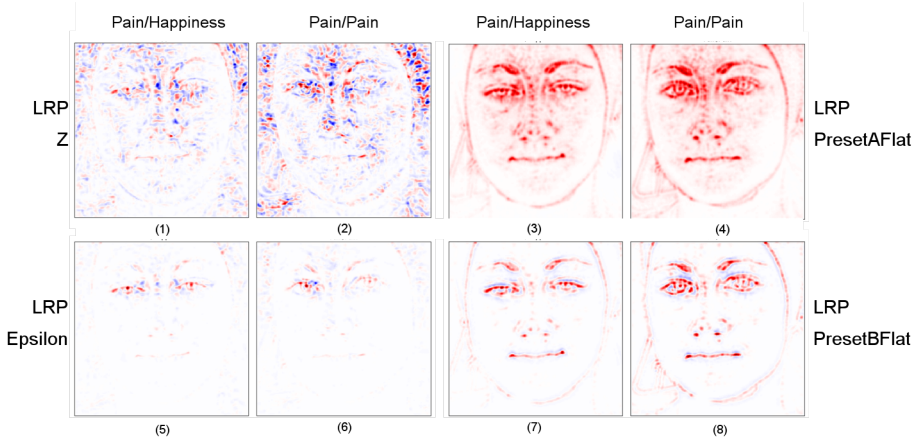


Figure 3: Visualizations for applying LRP method with different parameters on two pain images. First label indicates the true class, second label indicates the predicted class. The heatmap generated with the basic LRP approach (LRP-Z) is displayed in subimages (1) and (2). Subimages (3) and (4) display the heatmap generated with the LRP-PresetAFlat Variant. Subimages (5) and (6) display the heatmap generated with an ε stabilizer. Subimages (7) and (8) are the results of applying LRP-PresetBFlat. The visualizations display the predicted class.

happiness and disgust, images from the Actorstudy dataset⁶ were used. The visualizations using the LRP-PresetAFlat approach are shown on the right part in Figure 4. Here it can be seen that for happiness, the eyes and the mouth are important areas for the classification. For disgust, the focus lays on the nose and the eyes. This could be a reason that the pain image (subimage 6) was misclassified as disgust. For pain, the nostrils seems to be important.

6 Discussion

For the topic of **predictive performance**, this paper shows that the CNN could distinguish images of pain, disgust, and happiness only with an accuracy of 67%. Above all, happy faces were often misclassified as

⁶ Unpublished dataset from Intelligent Systems Group, Fraunhofer IIS, Erlangen

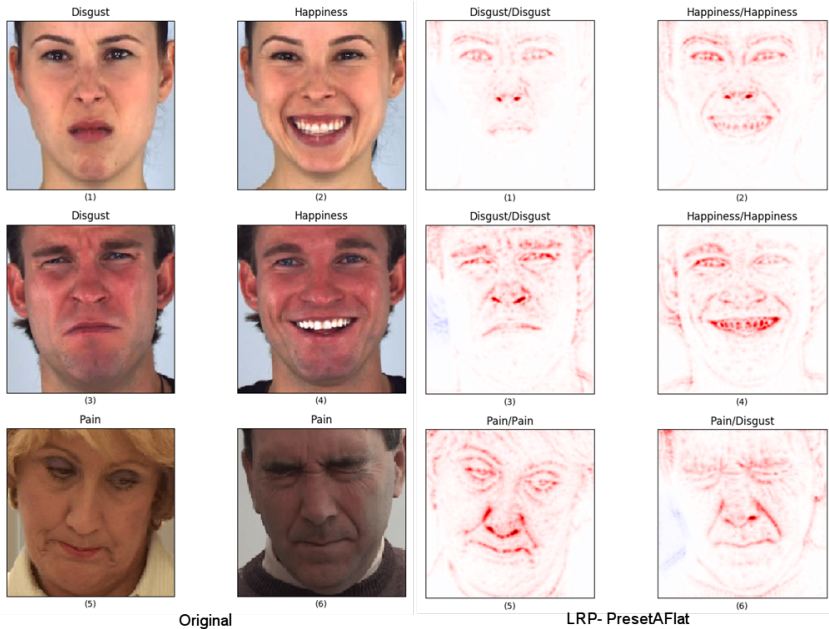


Figure 4: Left: Input images 1-4 from Actorstudy dataset (©Fraunhofer IIS) and images 5 & 6 from the UNBC-McMaster shoulder pain expression archive database (©Jeffrey Cohn) to visualize LRP approach. Right: Visualizations for applying LRP PresetAFlat method. The visualizations display the predicted class. First label refers to the true class, second label refers to the predicted class.

faces of pain.

For the part of **decision interpretation**, LRP is a helpful tool to generate a fine-granular heatmap of relevant pixels. The usage of LRP with its various parameters allows a wide range of adjustments. The results presented here for the categorization of pain, disgust, and happiness represent only an initial step into the research of making decisions of black-box systems comprehensible for humans. Lapuschkin et al. [13] already investigated the application of LRP for the recognition of age and gender from images of faces. They could show that the visualizations of relevant pixels allow an interpretation of the relevant facial ar-

eas to classify age and gender. However, when looking at facial expressions of happiness, pain, and disgust it becomes clear that pixel activation alone cannot yet provide a clear difference between the predicted classes for the human eye. Therefore, for the topic of **feature explanation**, the relevant features for the classification are not easily detectable by humans. This is partly due to the classification capabilities of the VGG-Face CNN. The classification accuracy of 67% must be taken into account when looking at the visualizations.

Montavon et al. [14] describe some practical recommendations to improve the visualizations generated by the LRP method: using dropout as regularization technique, preferring sum pooling, instead of max pooling and not to use too many fully connected layers in the network (whereas no definition is given for what is meant by ‘many’). Nevertheless, additional information is needed for a clearer interpretation of the results. Future research approaches may focus on the implementation of such additional information sources. Additional sources of information could, for example, take the form of linguistic information (e.g., ‘In this image, the eyes are important for the classification of happiness’) or the form of uncertainty formulations (e.g., pixel activations for happiness have an uncertainty value of 20 out of 100, while pixel activations for pain have an uncertainty value of 90 out of 100). Only when a result that is informative and interpretable for humans is achieved, a comprehensive application in real-life can be considered.

References

1. C. Frith, “Role of facial expressions in social interactions,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1535, pp. 3453–3458, 2009.
2. N. Ambady and R. Rosenthal, “Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis,” *Psychological Bulletin*, vol. 111, no. 2, pp. 256–274, 1992.
3. P. Ekman and E. L. Rosenberg, *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997.
4. H. Aviezer, Y. Trope, and A. Todorov, “Body cues, not facial expressions, discriminate between intense positive and negative emotions,” *Science*, vol. 338, no. 6111, pp. 1225–1229, 2012.

5. S. Brahnam, C.-F. Chuang, F. Y. Shih, and M. R. Slack, "Machine recognition and representation of neonatal facial displays of acute pain," *Artificial Intelligence in Medicine*, vol. 36, no. 3, pp. 211–222, 2006.
6. S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.
7. Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems*, 1990, pp. 396–404.
8. C. Ravat and S. A. Solanki, "Survey on different methods to improve accuracy of the facial expression recognition using artificial neural networks," in *Proceedings of the National Conference on Advanced Research Trends in Information and Computing Technologies*, vol. 4, no. 2, 2018.
9. P. Werner, A. Al-Hamadi, K. Limbrecht-Ecklundt, S. Walter, S. Gruss, and H. C. Traue, "Automatic pain assessment with facial activity descriptors," *IEEE Transactions on Affective Computing*, vol. 8, no. 3, pp. 286–299, 2017.
10. J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.
11. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
12. H. C. Lane, M. G. Core, M. Van Lent, S. Solomon, and D. Gomboc, "Explainable artificial intelligence for training and tutoring," University of Southern California Marina del Rey CA Institute for Creative Technologies, Tech. Rep., 2005.
13. S. Lapschkin, A. Binder, K.-R. Müller, and W. Samek, "Understanding and comparing deep neural networks for age and gender classification," in *Proceedings of the International Conference on Computer Vision*, 2017, pp. 1629–1638.
14. G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2017.
15. M. H. Kohlbrenner, "On the stability of neural network explanations," Apr 2017, bachelor's Thesis.
16. S. Bach, A. Binder, K.-R. Müller, and W. Samek, "Controlling explanatory heatmap resolution and semantics via decomposition depth," in *Proceedings of the International Conference on Image Processing*. IEEE, 2016, pp. 2271–2275.

17. D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex." *Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.
18. O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *BMVC*, vol. 1, no. 3, 2015, pp. 1–12.
19. S. Walter, S. Gruss, H. Ehleiter, J. Tan, H. C. Traue, P. Werner, A. Al-Hamadi, S. Crawcour, A. O. Andrade, and G. M. da Silva, "The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system," in *Proceedings of the International Conference on Cybernetics*. IEEE, 2013, pp. 128–131.
20. M. Alber, S. Lapuschkin, P. Seegerer, M. Hägele, K. T. Schütt, G. Montavon, W. Samek, K.-R. Müller, S. Dähne, and P.-J. Kindermans, "investigate neural networks!" *arXiv preprint arXiv:1808.04260*, 2018.
21. P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, and I. Matthews, "Painful data: The unbc-mcmaster shoulder pain expression archive database," in *Proceedings of the International Conference on Automatic Face & Gesture Recognition and Workshops*. IEEE, 2011, pp. 57–64.

Application of deep learning for crack segmentation on concrete surface

Oliver Müller¹, Ali Moghiseh², Henrike Stephani², Niklas Rottmayer¹,
and Feibai Huang¹

¹ TU Kaiserslautern

² Fraunhofer ITWM

Abstract In this paper we present a method for pixel-wise crack segmentation on concrete surfaces. For this purpose we are using the architecture of the SegNet model [1] and combine it with several ways of data augmentation. By this approach we can benefit from the extreme effectiveness of the SegNet approach, while avoiding the shortcomings of industrial applications, especially lack and imbalance of data.

Keywords Crack, deep convolutional neural networks, semantic pixel-wise segmentation, SegNet, CrackNet.

1 Introduction

The segmentation of cracks is an important step during the inspection, diagnosis and maintenance of concrete surfaces like pavements or walls. Most real world applications in that area are still relying on manual quality inspection. Due to the wide spectrum of applications there are several approaches for automated solutions, e.g., see [2, 3]. However, it remains a difficult task, since the concrete surface contains various types of irregularities due to different causes such as the high variance of the surface itself, the fine structure of the cracks and the possibility of concrete bubbles, stains, insufficient contrast and shading. Therefore, current methods either have difficulty detecting all defects or are very sensitive to finding false positives.

The main contribution of this paper is the adaption of SegNet [1] to this typical industrial application, where data is scarce and class representation is usually not balanced. A direct application of deep learning

methods can hardly be granted in these scenarios. For this reason we propose the application of various types of data augmentation to use the available annotated samples more efficiently. By this approach the dataset of crack images can be sufficiently enlarged and imbalance of classes can be counteracted such that the architecture gives very satisfying results. From this point on we will call this approach CrackNet

2 Model

2.1 Architecture

Our model CrackNet is a deep fully convolutional neural network architecture for semantic pixel-wise segmentation. This architecture has been very successfully applied to various computer vision tasks such as scene understanding and object detection. See [1].

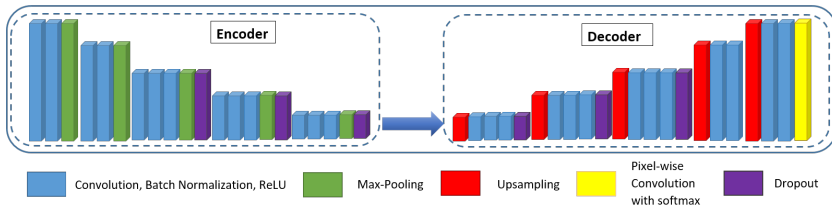


Figure 1: Architecture of our CrackNet Model.

The network consists of a sequence of encoder layers and a corresponding set of decoder layers completed by a pixel-wise convolutional layer, with a softmax activation function to generate classification probabilities between 0 and 1.

Each encoder in the encoder network performs convolution with a filter bank to produce a set of feature maps. These are then batch normalized [4], which normalizes the previous layer, allowing the usage of higher learning rates and an element-wise rectifier ReLU, defined as the positive part of its argument ($\max(0, x)$), is applied. It is followed by a non-overlapping 2×2 max-pooling with stride 2, which takes from each 2×2 square the maximum value (i.e. the neuron with the most activity) and discards all other values, thus diminishing the image resolution, i.e.

reducing the number of parameters and amount of computation in the network. We also perform a zero padding layer, which adds rows and columns of zeros around an image before each convolution in order to avoid problems with dimensions and boundaries.

The appropriate decoder in the decoder network upsamples its input feature maps using the memorized max-pooling indices from the corresponding encoder feature map. These feature maps are then convoluted with a trainable decoder filter bank to produce dense feature maps. A batch normalization step is then applied to each feature map.

To avoid overfitting we use a total of six dropout layers, placed after the last three max-pooling layers in the encoder-network and after the first three decoders in the decoder-network, each randomly removing 10% of the nodes in the neural net.

2.2 Training

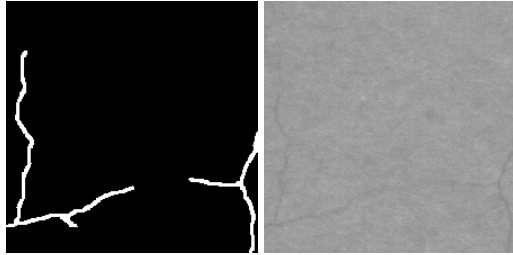


Figure 2: Label and original Image from the dataset.

We choose to work with an industrial dataset which contains a total of 353 gray-scale images with the size of 224×224 pixels of concrete surface cracks with corresponding label images. As mentioned this is quite small for a machine learning application. Furthermore the cracks are just a small part of the image, leading to the problem of class imbalance that can lead to overfitting and makes it difficult to compute the right class weights. Therefore, we augmented the data and labels in a simultaneous way by rotating and flipping both images.

Doing so, we generated from one single image eight different images, getting a total of 2824 images. To ensure model stability during lighting

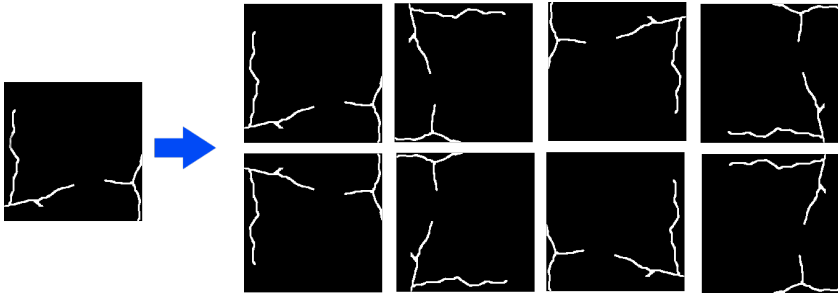


Figure 3: Example of our augmentation.

changes, we also apply a histogram equalization [5] on each picture. We use the cross-entropy loss [6] as the objective function for training the network, which is summed up over all pixels in an image.

Since we have a large variation in the number of pixels in each class in the training set, i.e. much more no crack pixels than crack pixels, we need to weight the loss differently based on the true class. Because a model without weighing, especially in its earlier stages, would learn never to predict the class crack and by doing that, it would achieve lower losses.

We use *median frequency balancing*, where each class is assigned a frequency defined as the number of pixels that belong to this class divided by the number of image that contain this class [1]. The weight of a class is its frequency divided by the median frequency over all classes. This implies that larger classes have a weight smaller than 1 and the smallest classes have the highest weight. After computing the right class weights, we now can train our model appropriately using the ADADELTA method [7].

3 Results

The proposed model for automatic crack segmentation is evaluated using test data containing 424 images. The test data does not contain augmented train data. The ground truth is used to evaluate the performance of the model (Table 2), by computing a global error-rate, Precision, Recall and the F1-score (F1), reflecting the overall classifier performance as follow:

$$\begin{aligned} \text{G. error-rate} &= 1 - \frac{\text{correctly pred. crack} + \text{correctly pred. background}}{\text{total number of pixels}} \\ \text{Precision} &= \frac{\text{correctly pred. crack}}{\text{correctly pred. crack} + \text{background pred. as crack}} \\ \text{Recall} &= \frac{\text{correctly pred. crack}}{\text{correctly pred. crack} + \text{crack pred. as background}} \\ \text{F1} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned}$$

In the evaluation with the test data our proposed model gets a global accuracy of 98,98%, which is a very good result. The performance can be shown using the confusion matrix where pixels are categorized by their actual and predicted class. We use a tolerance of 2 pixels, which means a predicted crack pixel is considered correctly predicted as long as the actual crack is within a range of 2 pixels.

No. of Pixels	Actual Crack	Actual Background
Pred. Crack	42685	2311
Pred. Background	4600	2609732

Table 1: Confusion matrix of the proposed model on the test data.

For comparison purposes, the two bottom lines of Table2 show the results of native methods without weights or without augmentation with the CrackNet architecture. The results of Table2 show that the proposed methodology achieves overall better results than the native techniques. The model without augmentation has significantly worse results in Recall (90,27% against 81,47%) and in the F1-score (92,51% against 86,21%) and is also lacking in Precision (94,86% against 91,53%). Although the results of the model without weights seems similar to our proposed

model, the gain in system robustness, which can be seen in the following images, leads to the conclusion that the proposed methodology is good.

Strategy	G. Error-rate	Precision	Recall	F1
proposed model	1,02%	94,86%	90,27%	92,51%
model without weights	1,30%	94,61%	89,73%	92,11%
model without augmentation	3,28%	91,53%	81,47%	86,21%

Table 2: Results for the proposed model compared to a model without weights and a model without augmentation.

If we compare the following images, we see that our proposed model is more stable than the other models, since they are prone to finding false negatives and the model without augmentation is also sensitive to finding false positives.

Our model predicts the cracks almost completely, independent of the direction of the crack. The model isn't affected by the noisy images, change of lighting or the small concrete bubbles, so it can be used for all different types of concrete crack images without big loss of accuracy.

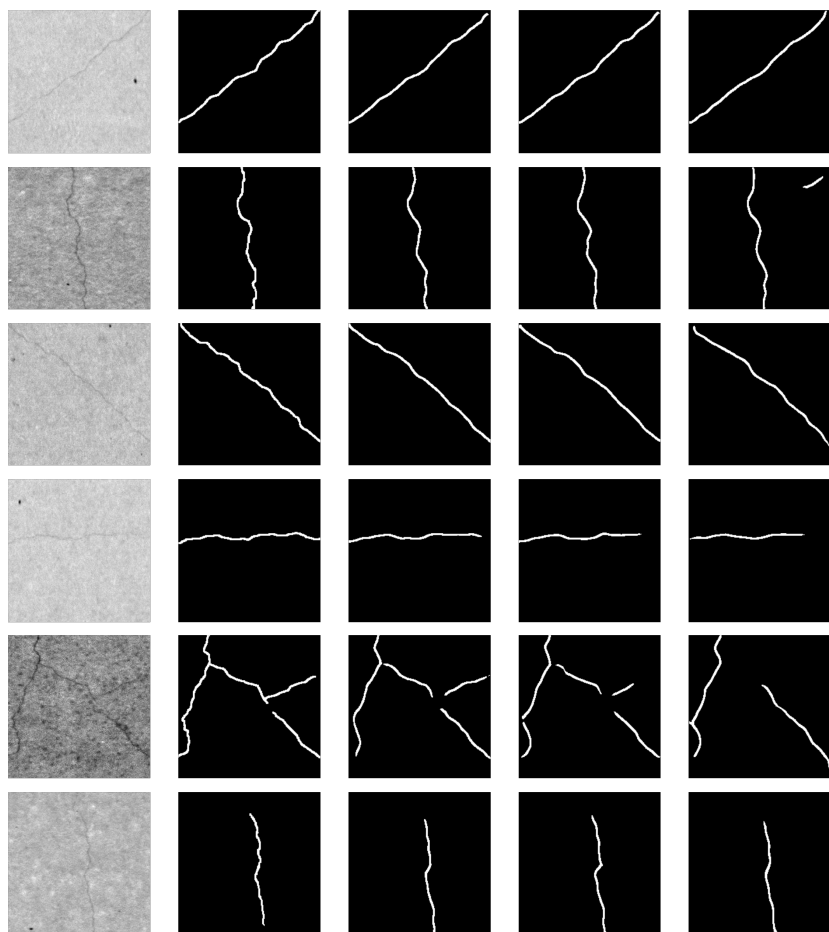


Figure 4: Results of CrackNet for the crack segmentation. From left to right: Original, Ground Truth, Proposed Model, Model without Weights, Model without Augmentation.

4 Conclusions

In the past, deep learning models have increased their success, due to fast and simple structures and the availability of big datasets. However, in practice, even if we have large input data, we often lack of obligatory labels that are required for the training and which are tedious to generate. Thus our goal has to be to use the given, possibly small data as efficient as achievable.

In this paper a simple convolutional neural network model with different types of data augmentation for crack segmentation is proposed, achieving promising performance results. It was demonstrated that the proposal is able to deal with crack images, independent of lighting, noise and direction of the crack.

The main motivation was the need to design a model for concrete crack segmentation which is efficient both in accuracy of segmentation and lack of false positives, despite using a quite small dataset. Having this efficient structure and outstanding performance our model is well suited for all crack segmentation problems.

References

1. V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2015.
2. T. Yamaguchi and S. Hashimoto, "Fast crack detection method for large-size concrete surface images using percolation-based image processing," *Machine Vision and Applications*, vol. 21, no. 5, pp. 797–809, feb 2009.
3. P. Subirats, J. Dumoulin, V. Legeay, and D. Barba, "Automation of pavement surface crack detection using the continuous wavelet transform," in *2006 International Conference on Image Processing*. IEEE, oct 2006.
4. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.
5. R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.
6. C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.
7. M. D. Zeiler, "Adadelta: An adaptive learning rate method," 2012.

Der phenoTest – Ein automatisiertes CT-System zur Phänotypisierung von Zuckerrübenkeimlingen

The phenoTest – An automated CT system for phenotyping of sugar beet seedlings

Markus Rehak¹, Ulf Haßler¹, Tobias Grulich¹, Norbert Wörlein¹, Felix Porsch², Yvonne Götz³ und Antje Wolff³

¹ Fraunhofer-Entwicklungszentrum Röntgentechnik EZRT,
Flugplatzstr. 75, 90768 Fürth

² Fraunhofer-Institut für Zerstörungsfreie Prüfverfahren IZFP,
Campus E3.1, 66123 Saarbrücken

³ Strube D&S GmbH
Hauptstr. 1, 38387 Söllingen

Zusammenfassung Die Keimfähigkeit von Zuckerrübensamen wird im Allgemeinen nach den International Rules for Seed Testing (ISTA) bestimmt. Diese visuelle Prüfmethode ist sehr personal- und kostenintensiv sowie subjektiv und schwer standardisierbar. Mit dem phenoTest wurde ein vollautomatisches Phänotypisierungs-System entwickelt, das mittels 3D-Computertomographie eine objektive, zerstörungsfreie, und quantitative Bestimmung von Keimfähigkeitsparametern erlaubt. Die automatisierte Bildauswertung muss hierbei mit der für den hohen Durchsatz notwendigen geringen Auflösung der CT-Aufnahme sowie der Tatsache, dass zu trennende Strukturen keinen Kontrastunterschied aufweisen, zurechtkommen. Ein Vergleich der automatisch ermittelten Keimfähigkeitswerte vom phenoTest mit den visuell ermittelten Werten der ISTA-Methode zeigte eine hohe Korrelation.

Schlagwörter Bildverarbeitung, Zuckerrübensamen, Keimfähigkeit, Phänotypisierung, Computertomographie.



Abbildung 1: Visuelle Klassifikation von Zuckerrübenkeimlingen nach ISTA-Test.

Abstract Germination capacity of sugar beet seeds is conventionally assessed by means of the International Rules for Seed Testing (ISTA). This visually, very time-consuming and cost-intensive test has subjective results and is therefore difficult to standardize. With the phenoTest, a fully automated phenotyping system has been developed. This test allows for an objective, non-destructive and quantitative determination of the germination parameters. The automated image processing has to cope with the limited resolution of the CT image required for achieving the high throughput, as well as with the lack of contrast in-between the structures to be separated. A comparison of automatically determined germination parameters from the phenoTest with the visually determined parameters of the ISTA method showed a high correlation.

Keywords Image processing, sugar beet seeds, germination capacity, computed tomography.

1 Einleitung

Wer Saatgut in Verkehr bringt muss die Regelungen des Saatgutverkehrsgesetzes [1] beachten. Unter anderem muss die Keimfähigkeit des Saatgutes geprüft sein. Als offizieller, weltweit üblicher Standardtest zur Beurteilung der Saatgutqualität dient der ISTA-Keimfähigkeitstest [2].

Dieser ISTA-Test hat einige ganz wesentliche Nachteile. Die Klassifikation der Keimpflanzen nach normal, anomal und nicht gekeimten Samen wird visuell durchgeführt und ist damit subjektiv und schwer zu standardisieren (Abb. 1). Die Aussagen sind rein pauschal, es gibt keine Messwerte zur Qualität der einzelnen Pflanzen, wie Größe, Länge der Wurzel, Stärke des Hypokotyls usw. Diese Qualitäten sind aber wichtige Kenngrößen für die Triebkraft. Zudem ist der Test personalintensiv und die Ergebnisse sind nach dem Test nicht mehr nachvollziehbar, da die Pflanzen verworfen werden und es keine visualisierbare Dokumentation der gekeimten Samen gibt. Im Zweifel muss der Test erneut durchgeführt werden, um die Auswertung zu überprüfen.

Deshalb hat die Firma Strube D&S in Zusammenarbeit mit der Fraunhofer Gesellschaft den phenoTest entwickelt. Ein wesentlicher Teil des Systems ist ein vollautomatisches 3D-Computertomographiesystem (CT) mit integrierter Volumenbilddauswertung für die digitale Hochdurchsatz-Phänotypisierung von keimenden Zuckerrübensamen. Die CT ermöglicht als zerstörungsfreies und berührungsloses Prüfverfahren die Verfolgung des Entwicklungsprozesses jedes einzelnen Keimlings einer Stichprobe innerhalb des geschlossenen Keimbehälters. Die für die Bestimmung der Keimfähigkeit und Triebkraft relevanten Bestandteile Wurzel, Hypokotyl (Keimstengel) und Kotyledonen (Keimblätter) werden durch die automatisierte 3D-Bildanalyse aus dem Datenvolumen extrahiert und anschließend vermessen. Damit ist eine quantitative und damit objektive Beurteilung der Qualität eines jedes einzelnen Keimlings und seiner Bestandteile möglich [3].

2 Beschreibung des CT-Systems

Um den geforderten hohen Proben-Durchsatz zu erreichen muss zum einen die CT-Aufnahme möglichst schnell erfolgen, ohne dabei zu große Einbußen in der Qualität der CT-Aufnahme in Kauf zu nehmen. Zum anderen muss auch der Probenwechsel automatisiert werden. Um ausreichend Probenkapazität und damit autonome Laufzeit für das System zu erreichen, ist der Probenwechsel zweistufig ausgelegt. Je 32 Proben mit jeweils 50 Keimlingen stehen auf einem speziellen Probenträger. Bis zu drei solcher Probenträger können in ein Fördersystem eingestellt



Abbildung 2: Vollautomatischer CT-Scanner für Keimproben. (Quelle: Strube)

werden. Das Fördersystem bringt die Probenträger nacheinander zur Entnahmestation. Dort entnimmt ein kartesischer Roboter nacheinander die einzelnen Proben und führt sie dem CT-Scanner zu. Die Taktzeit für die CT-Aufnahme liegt bei zwei Minuten einschließlich Probenwechsel. Unmittelbar nach der CT-Aufnahme erfolgt die Bildverarbeitung. Sie wird umlaufend auf insgesamt fünf Bildverarbeitungsrechner verteilt, damit die Auswertung schritthaltend mit der CT-Aufnahme ist. Nach etwa drei Stunden sind drei Probenträger mit insgesamt knapp einhundert Proben abgearbeitet (Abb. 2).

Als Ergebnis wird für jede Probe eine Liste mit den verschiedenen Qualitätsmerkmalen für jede einzelne Keimpflanze erstellt, sowie eine statistische Zusammenfassung der Ergebnisse. Darüber hinaus werden die bewerteten Volumenbilder für jede einzelne Keimpflanze zur Dokumentation hinterlegt.

3 Beschreibung der Messung

Für die Messung der Keimlinge mittels CT wurde ein 15 cm hoher, konisch zulaufender und verschließbarer Keimbehälter speziell angefertigt. Die konische Form ist zur Stabilisierung des Filterpapiers notwendig, da anders als bei der ISTA-Methode ein auf den Kanten stehendes

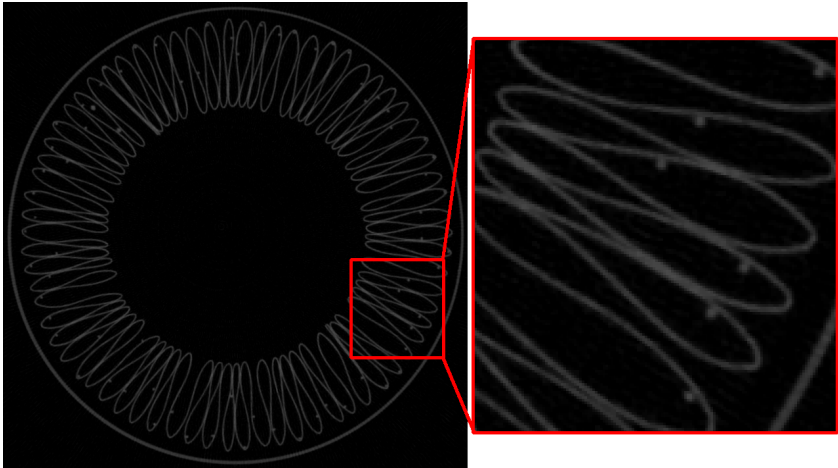


Abbildung 3: XY-Schicht einer CT-Rekonstruktion. Die Wurzeln liegen am Keimpapier an.

Filterpapier verwendet wird. Die Höhe des Behälters führt in Verbindung mit der einzuhaltenden Taktzeit von 2 Minuten zu einer realisierbaren Voxelgröße von etwa $140\ \mu\text{m}$.

Die Samen werden zu Beginn des Keimversuchs ca. 3 cm unterhalb der Filteroberkante in die äußeren Schlaufen des Faltenbalgs eingelegt. Die nach innen zeigenden Schlaufen bleiben leer. Während der Keimung wächst das Hypokotyl nach oben, während sich die Wurzel nach unten ausbreitet. Die Wurzeln legen sich hierbei an das Filterpapier. Durch den konisch zulaufenden Faltenbalg befinden sich die Wurzeln am unteren Ende häufig in der Spitze der Falte (Abb. 3).

4 Segmentierung der Keimlinge

Das nur 0,5 mm dicke Papier sowie ein Wurzeldurchmesser von teils deutlich unter 0,5 mm führen dazu, dass während der Bildverarbeitung Strukturen erkannt und voneinander getrennt werden müssen, die nur ein bis drei Voxel breit sind. Erschwerend kommt hinzu, dass sowohl die Keimlinge als auch das Filterpapier Röntgenstrahlung ähnlich stark

schwächen und somit kein für die Bildverarbeitung nutzbarer Kontrastunterschied besteht.

Die Bildverarbeitung besteht deshalb aus vier Phasen, in denen größtenteils auf Morphologie und Vorwissen basierende Algorithmen eingesetzt werden. Nach der initialen Entfernung des Keimbehälters wird das Filterpapier von den Keimlingen abgetrennt, wobei auch Teile der Keimlinge als Filterpapier interpretiert werden könnten. Die dritte Phase erzeugt anschließend zusammenhängende Keimlinge, an denen möglichst kein Filterpapier mehr anhaftet. Im letzten Schritt werden die Keimlinge segmentiert und die notwendigen Kennzahlen berechnet.

4.1 Entfernung des Keimbehälters

Bei den bis zu 14 Tagen dauernden Keimversuchen bilden sich mit großer Wahrscheinlichkeit Tropfen aus Kondenswasser an den Wänden des Keimgefäßes. Ein Abstandshalter sorgt dafür, dass kein Kontakt zwischen Filterpapier und Wand besteht. Mit Hilfe einer Distanztransformation [4] zur Messung der Distanz von der Behälteraußenseite lässt sich die mit sehr geringer Toleranz gefertigte konische Wand des Keimbehälters ausblenden. Sowohl der Behälterboden als auch der Deckel sind ohnehin nicht Teil des rekonstruierten Datenvolumens. Die übrigbleibenden Kondenswassertropfen lassen sich anschließend mittels einer Zusammenhangsanalyse entfernen.

4.2 Entfernung des Filterpapiers

Der nächste Schritt ist die Abtrennung des Filterpapiers von den Keimlingen. Durch den fehlenden Kontrastunterschied bedingt sind die einzigen Konstanten, auf die sich die Bildverarbeitung stützen kann, die bekannte Dicke des Filterpapiers sowie die Tatsache, dass sich Keimlinge immer auf der Innenseite des Filterpapiers und damit in den äußeren Schlaufen befinden.

Die grundsätzliche Idee ist deshalb, die Außenseite des Filterpapiers zu identifizieren und gewissermaßen abzuschälen. Dies lässt sich durch eine Segmentierung des Hintergrunds erreichen: Werden den Bereichen auf der Innen- und Außenseite des Filterpapiers jeweils unterschiedliche Werte zugewiesen, so lässt sich anschließend ein Distanzfeld zur Außenseite berechnen. Alle Distanzen, die maximal der Papierdicke in

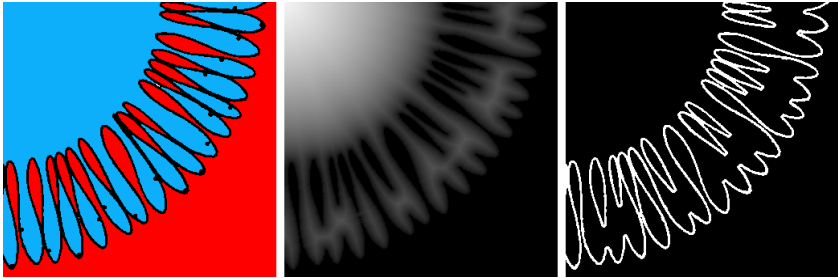


Abbildung 4: Erster Schritt der Papierentfernung. Links: Rote Regionen liegen auf der Außenseite des Papiers, blaue Regionen entsprechend innen. Mitte: Distanztransformation bezüglich der Außenseite. Recht: Entferntes Filterpapier.

Voxeln entsprechen, gehören damit zum Filterpapier und werden aus dem Datenvolumen entfernt (Abb. 4).

Für die dazu notwendige Segmentierung wird eine schichtweise Zusammenhangsanalyse [5] des Hintergrunds durchgeführt, wodurch der komplette Hintergrundbereich in eine Vielzahl von Segmenten zerfällt. Durch Analyse von Lage und Form dieser Segmente können diese jeweils der Außen- bzw. Innenseite zugeordnet werden.

Im Idealfall bleiben nach diesem Schritt zwei Objekttypen im Datenvolumen übrig. Einerseits handelt es sich um Fragmente der gesuchten Keimlinge, andererseits bleiben genau die Teile des Filterpapiers übrig, an denen sich dessen Außenseiten berühren. Dies ist für die jeweils benachbarten äußeren Falten häufig nicht zu vermeiden. Betrachtet man die Papierreste schichtweise, so handelt es sich hierbei im Prinzip um je zwei parallel verlaufende, schwach gekrümmte Linien. Eine Möglichkeit, die hier anhaftenden Keimlinge abzuspalten ist deshalb die Ermittlung einer den Papierrest modellierenden Splinefunktion, aus der sich anschließend ein Papierrest ohne anhaftende Keimlinge mittels Dilatation rekonstruieren lässt.

Zu diesem Zweck werden die Papierreste schichtweise skelettiert, wodurch je Papierrest eine linienhafte Darstellung mit mindestens zwei Endpunkten entsteht. Zwei dieser Endpunkte entsprechen den Kontaktpunkten zum bereits entfernten Filterpapier, so dass die dazwischen verlaufenden Linien näherungsweise den tatsächlichen Verlauf des Filterpapiers abbilden. Alle eventuellen weiteren Enden sowie die

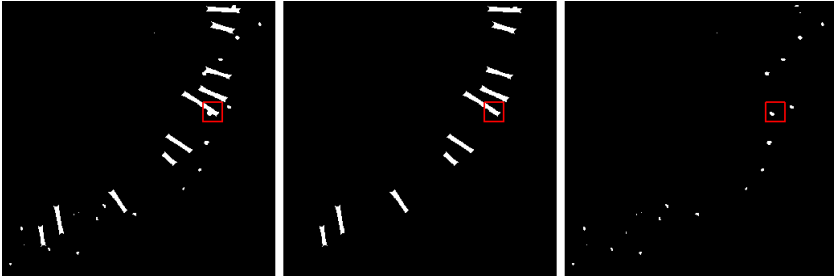


Abbildung 5: Links: An Papierresten anhaftende Keimlingsfragmente. Mitte: Synthetisch erzeugte Papierreste. Rechts: Keimlingsfragmente nach Subtraktion der synthetischen erzeugten Papierreste.

dadurch bedingten Verzweigungen müssen durch anhaftende Keimlingsfragmente bedingt sein und können eliminiert werden. Die Punkte der verbleibenden Skelettierung jedes Papierrests dient als Eingangsdaten für die Interpolation mittels einer offenen kubischen Splinefunktion mit fixem Start- und Endpunkt (Abb. 5).

Die durch die Dilatation mittels einer der Papierdicke entsprechenden Dilatationsmaske rekonstruierten Papierfragmente können abschließend von den real vorhandenen Fragmenten subtrahiert werden, wodurch nur noch die Keimlingsfragmente übrigbleiben.

4.3 Rekonstruktion der freigestellten Keimlinge

Bedingt durch die geringe Auflösung lassen sich nach der Entfernung des Filterpapiers in der Regel nicht nur komplette unverfälschte Keimlinge finden. Insbesondere im Bereich der Wurzeln, wo auch der Wassergehalt des Papiers höher ist, können durch die dadurch verursachte leicht höhere Papierdicke noch Reste des Filterpapiers an den Keimlingen anhaften. Gleichzeitig kann nicht garantiert werden, dass alle Keimlinge komplett zusammenhängend vom Filterpapier gelöst werden konnten. Aufgrund der schichtweisen Verarbeitung können wiederum im Bereich der feinen Wurzeln Teile der selben fehlen.

Deshalb muss im nächsten Schritt dafür gesorgt werden, dass die verschiedenen Fragmente wieder korrekt zu Keimlingen zusammengesetzt werden. Nebenbei muss eventuell noch anhaftendes Filterpapier iden-

tifiziert und isoliert werden können.

Hierbei hilft wiederum das Wissen über den grundsätzlichen Aufbau eines Zuckerrübenkeimlings. Ausgehend vom Samen breitet sich das Hypokotyl ohne Verzeigungen nach oben aus und mündet in der Regel in genau zwei Keimblätter. Hierbei kann es mitunter zur Bildung einer Schlaufe kommen. Gleichzeitig wächst genau ein Wurzelstrang ohne Verzweigungen nach unten. Ein dem Keimling entsprechendes Modell kann somit davon ausgehen, dass sich ausgehend von der Wurzelspitze ein eindeutiger Weg zu den Keimblättern finden lassen muss.

Somit wird zuerst eine dreidimensionale Skelettierung [6] der Keimlingsfragmente erzeugt. Diese haben besonders an den Stellen, an denen noch Reste des Filterpapiers anhaftend sind, meist zahlreiche Verzweigungen. Die Skelettierungen werden deshalb weiter zerlegt in Liniensegmente und Kreuzungspunkte. Ausgehend von dem am weitesten untenliegenden Punkt eines Liniensegments wird der Keimling anschließend iterativ wieder zusammengesetzt. Der Algorithmus versucht den kürzesten Weg von der Wurzelspitze zu den Keimblättern zu finden und achtet hierbei darauf, dass einzelne Liniensegmente nur dann verschmolzen werden dürfen, wenn dies bezüglich Abstand und Winkel zueinander schlüssig ist und wenn sie sich innerhalb der gleichen Falte befinden.

Dies führt einerseits dazu, dass sich die einzelnen Keimlingsfragmente wieder korrekt zusammensetzen lassen, und andererseits können Liniensegmente, die durch anhaftendes Filterpapier bedingt sind, aussortiert werden, da sie nicht auf dem kürzesten Weg von der Wurzelspitze zu den Keimblättern liegen. Es entsteht somit ein dem Keimling entsprechendes Gesamtskelett.

Hierbei wird genau eine Verzweigung mit in das Modell übernommen. Diese muss sich an einer Stelle am oberen Ende befinden, die innerhalb eines Intervalls liegt, dessen Unter- und Obergrenze der minimalen und maximalen Länge eines Keimblatts entspricht. Die maximale Dicke eines Keimlings ist eine weitere bekannte Größe, weshalb der finale Keimling mittels Dilatation rekonstruiert werden kann. Lücken zwischen den ursprünglichen Fragmenten werden hierbei interpoliert und anhaftendes Filterpaper entfernt.

4.4 Segmentierung der Keimlinge

Die letzte Phase der Auswertung erzeugt zuerst eine Segmentierung des Keimlings, um anschließend die für eine Bewertung relevanten Merkmale berechnen zu können. Gleichzeitig dient die Segmentierung auch der Visualisierung des Endergebnisses. Die zu segmentierenden Bestandteile sind die Keimblätter, das Hypokotyl, die Wurzel, die Reste des ursprünglichen Samens sowie alle nicht zum Keimling gehörigen Bestandteile.

Die Segmentierung beginnt an der Wurzelspitze und hat zuerst die Aufgabe, den Übergang in das Hypokotyl zu erkennen. Hierbei kann nicht davon ausgegangen werden, dass sich dieser Übergang genau dort befindet, wo auch die Reste des Samens zu finden sind. Stattdessen geht der Übergang in das Hypokotyl mit einer lokal starken Änderung der Dicke einher. Unglücklicherweise bedeutet dies bei der vorhandenen Auflösung von nur 140 μm , dass dies gleichbedeutend ist mit einer Dickenänderung von gerade einmal ein bis zwei Voxel, d.h. die Wahrscheinlichkeit ist groß, dass allein das Rauschen in den Ursprungsdaten schon ausreichend ist, dass dieser Übergang nicht mehr klar erkennbar ist. Für diese Fälle wird der Übergang an der ursprünglichen Lage des Samens angenommen (Abb. 6).

Das Hypokotyl mündet schließlich an Stelle der einzig zulässigen Verzweigung in die Keimblätter. Somit bleiben nur noch Samenreste und Papierreste übrig. Diese lassen sich dadurch unterscheiden, dass Samenreste meist deutlich dicker als Papierreste sind.

Letztlich lassen sich für alle Segmente Merkmale berechnen. Wichtig für eine Bewertung der Keimfähigkeit und Triebkraft sind beispielsweise das Längenverhältnis von Hypokotyl und Wurzel sowie deren Volumina.

5 Ergebnisse

Im ersten Schritt zeigt der direkte visuelle Vergleich zwischen der fotografischen Aufnahme und den freigestellten segmentierten Keimlingen in der CT-Aufnahme sehr gute Übereinstimmungen. Bei den geprüften Referenzproben gibt es nur eine geringe Fehlerquote (Abb. 6). Deshalb kann über die Länge und das Volumen der einzelnen Pflanzenorgane eine Klassifizierung der Keimlinge in die von ISTA geforderten Klassen

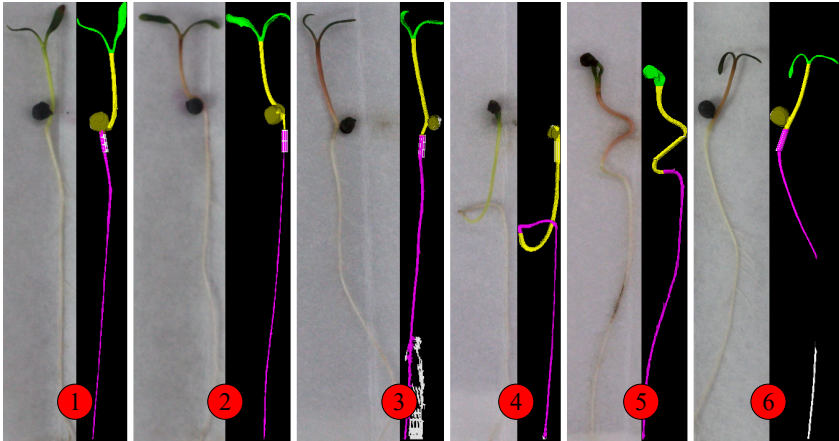


Abbildung 6: Segmentierungsergebnisse: 1, 2) Fehlerfreie Segmentierung. 3) Gute Segmentierung trotz Papieranhaftung. 4) Gute Segmentierung trotz Schlaufenbildung. 5) Problem bei der Trennung von Pille und Keimblättern. 6) Nicht vollständig segmentierte Wurzel.

normal, anormal und nicht gekeimt erfolgen. Die Unterscheidung der Pflanzen ist vollständig anhand festgelegter Schwellenwerte standardisiert.

Im zweiten Schritt weist der Vergleich der Ergebnisse der ISTA-Keimversuche und der automatischen Klassifikation über die Bildverarbeitung eine starke Korrelation auf. Hierbei wurden Samenchargen unterschiedlicher Genetik, Herkunft und Verarbeitungsstufen verglichen. Gleichzeitig zeigt die quantitative Vermessung der einzelnen Keimpflanzen eine sehr große Variabilität im Hinblick auf die Pflanzenqualität und Homogenität zwischen und innerhalb der Saatgutpartien. Erste Ergebnisse deuten darauf hin, dass die Vermessung der individuellen Keimlinge mittels CT nicht nur sehr gut mit der Triebfähigkeit im Feld korreliert, sondern sogar besser als die mittels ISTA-Methode ermittelte reine Keimfähigkeit.

Literatur

1. BMJV, "Saatgutverkehrsgesetz," <https://www.gesetze-im-internet.de/saatverkg.1985/BJNR016330985.html>, 12 2016.
2. ISTA, "International rules for seed testing," <http://doi.org/10.15258/istarules.2015.05>, 2015, vol. 2015, Chapter 5, i-5-56 (60).
3. A. Wolff and I. Götz, "The phenotest - 4d phenotyping to automatically measure germination and vigor," in *Sugar Industry*, vol. 141, no. 9. Verlag Dr. Albert Bartens KG, Berlin, 2016, pp. 571–574.
4. T. Lee, R. Kashyap, and C. Chu, "Building skeleton models via 3-d medial surface/axis thinning algorithms," *Computer Vision, Graphics, and Image Processing*, vol. 56, no. 6, pp. 462–478, 1994.
5. A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," *J. ACM*, vol. 13, no. 4, pp. 471–494, Oct. 1966. [Online]. Available: <http://doi.acm.org/10.1145/321356.321357>
6. A. Meijster, J. B. T. M. Roerdink, and W. H. Hesselink, "A general algorithm for computing distance transforms in linear time," 2000.

Algorithms for microlens center detection

Maximilian Schambach and Fernando Puente León

Karlsruhe Institute of Technology
Institute of Industrial Information Technology
Hertzstr. 16, 76187 Karlsruhe

Abstract We investigate four algorithms for microlens center detection, two of which have not been previously discussed in the literature. Using a physical approach, we create a set of 81 synthetic white images with known microlens center coordinates. Applying the different detection algorithms to the synthetic white images, we are able to quantitatively evaluate their respective performance in terms of accuracy, precision and recall. Overall, the proposed methods outperform the ones that have been previously published.

Keywords Image processing, computational imaging, microlens array.

1 Introduction

In the scope of geometrical optics, light is described as rays whose optical path lengths, according to Fermat's principle, attain an extremum. Geometrical optics is well suited to describe the imaging process of cameras but as such is lacking the description of many essential properties of light. By means of an extension to geometrical optics, properties such as color and intensity, which can only be described within wave optics (or higher order theories such as quantum electrodynamics), can be heuristically incorporated into geometrical optics: The *light field* (LF) or *plenoptic function* $L_{\lambda,t}(x, y, z, \phi, \theta)$ describes the optical radiance at point (x, y, z) in direction (ϕ, θ) of wavelength λ at time t in units $\text{Wm}^{-2}\text{sr}^{-1}$. In homogeneous media that are free of occluders, the radiance along a ray is constant. The spatial dependency of the LF can hence be reduced by one dimension, resulting in the so-called *4D light field* $L_{\lambda,t}(u, v, a, b)$,

where the coordinates (u, v, a, b) correspond to a certain parametrization of the spatial dependency of the LF of which there are numerous. For LF cameras, one usually uses the plane-plane parametrization: A light ray inside a camera is uniquely described by the intersection points (u, v) and (a, b) of two parallel planes, the main lens plane and the sensor plane.

At every sensor coordinate (a, b) , conventional cameras perform an integration of the LF over the time t , the wavelength λ as well as the coordinates (u, v) which encode the direction of the incident light ray. The aim of LF cameras is to measure this angular dependency that is lost in conventional camera systems. LF cameras have been part of active research over the past two decades and open new possibilities to measure spatial information, in particular the depth, of a scene. There are different implementations of LF cameras such as camera arrays, gantry or microlens array (MLA) based LF cameras. Camera arrays and gantry based LF cameras are the optically most straightforward way to construct a LF camera. They offer very good spatial and angular resolution but require sophisticated calibration schemes, are bulky and mechanically sensitive. MLA based cameras [1, 2] have recently gained in popularity due to their compact monocular design. By placing a MLA in front of the imaging sensor, the direction (u, v) of incident light rays is coded into the sensor image: Light rays hitting the sensor plane perpendicularly will be imaged onto the central pixel underneath a microlens (ML), slanted rays will be imaged onto pixels deviating from the center. Thus, the relative position of the image point w. r. t. the ML center codes the (u, v) coordinate of the incident light field, the position of the ML itself the (a, b) coordinate.

The foundation of all calibration and decoding schemes needed in the case of MLA based LF cameras is the exact detection of the ML centers with subpixel precision. The ML centers are detected using so-called *white images* (WIs) – images of a white scene for example taken using an optical diffuser. The calibration and decoding quality is dependent on the accuracy and robustness of the ML center detection. In spite of the importance of the ML center detection, to our knowledge there is no literature investigating the quality of the ML center detection algorithms. This is the main scope of this article. The results presented in this paper analogously apply for MLA based computational cameras other than LF cameras with little or no adjustment, e. g. MLA based spectral cameras.

The paper is organized as follows. Following this short introduction, we outline two existing and propose two new ML center detection algorithms in Section 2. In Section 3, we first present the methods used to create reference data and the used evaluation metrics. Using the synthetic ground truth data, we present a performance evaluation of the different ML center detection algorithms. Section 4 concludes the paper with a brief summary.

2 Microlens center detection

Challenges in the detection of MLs and their centers are versatile: On the one hand, the sheer amount of MLs in MLAs used in practice (in the case of the Lytro Illum camera about 150.000 MLs) limits the algorithm's complexity. On the other hand, the geometry of the MLA is not trivial and usually slightly irregular: Ideally, the MLs are circular, arranged in a perfect rectangular or hexagonal grid and perfectly aligned with the sensor. In practice, the MLA will be translated, rotated and scaled¹ w. r. t. the sensor and the lattice will be slightly irregular due to manufacturing tolerances. Furthermore, main lens and ML vignetting influences the form and brightness of the ML images, particularly of those that are close to the sensor edge.

There are two ML detection methods proposed in the literature: Cho et al. [3] first perform a greyscale erosion and clustering of the WI. To estimate the ML centers, they use a parabolic least squares (LS) regression of the clustered MLs. Dansereau et al. [4] propose a decoding pipeline for the Lytro light field camera implemented, as the de-facto non-proprietary standard, in the MATLAB *Light Field Toolbox*. As a pre-processing step, the WI is convolved with a filter kernel. In the case of the MATLAB *Light Field Toolbox*, a disk kernel with a fixed radius of $1/3$ of the expected ML spacing is used. The ML centers are then estimated by finding the local maxima in the filtered image². We propose two detection algorithms that join parts of the aforementioned detection pipelines: First, to reduce noise and other high frequencies, we convolve the image with a filter kernel. We investigate different kernel types and

¹ That is, ML centers will not be imaged onto pixel centers.

² This does not result in subpixel precision. However, in the succeeding decoding pipeline of the toolbox, a ML grid model is built with subpixel precision.

sizes. In a second step, the images are clustered using local thresholding (by local Gaussian weighted mean with a block size of 17 px) to find areas around local peaks and a standard cluster labeling algorithm. Each cluster represents exactly one ML. The thresholding has to be performed locally, as the main lens vignetting results in different local maximum values for the MLs close to the sensor edge. Finally, we either perform a parabolic LS estimation to obtain the ML centers or calculate the center of mass (CoM) of the individual clusters. Both methods yield ML centers with subpixel precision but perform differently in terms of accuracy and runtime. The different ML center detection pipelines are depicted in Figure 1.

3 Evaluation

3.1 Reference data

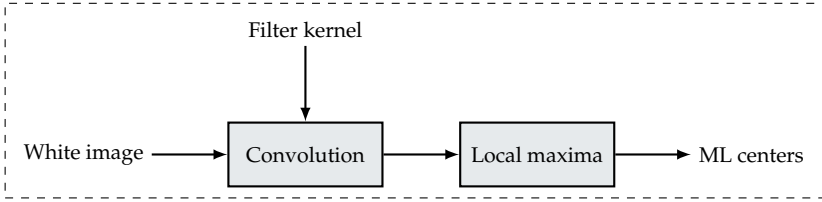
In order to objectively evaluate the performance of the ML detection algorithms, appropriate reference data is needed. Of course, real WIs, as for example provided by the Lytro cameras, are unsuited since the actual ML centers are unknown. Therefore, the reference data has to be synthesized: In general, the irradiance at a sensor coordinate (a, b) for a camera with one lens of radius r at distance d to the sensor is given by [2]

$$E(a, b) = \frac{1}{d^2} \int_0^T \int_{\Lambda} \iint_{\mathcal{A}} L_{\lambda, t}(u, v, a, b) \cos^4 \phi \, du \, dv \, d\lambda \, dt.$$

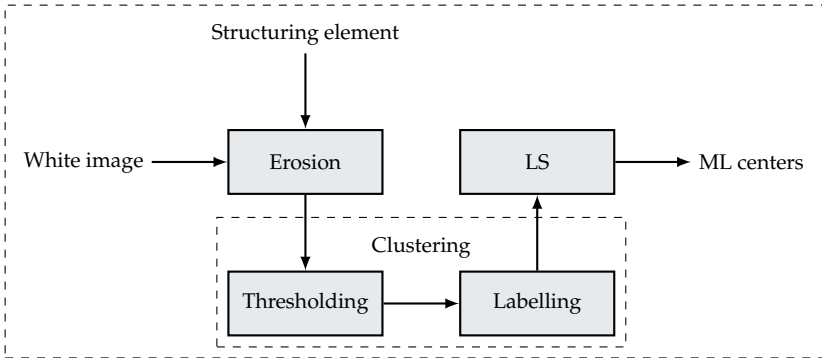
Here, $\mathcal{A} = \{\mathbf{u} = (u, v) \in \mathbb{R}^2 : \|\mathbf{u}\| < r\}$, T denotes the exposure time, Λ a wavelength interval and $\phi = \arctan(d^{-1} \cdot |\sqrt{u^2 + v^2} - \sqrt{a^2 + b^2}|)$ the angle between the incident light field and the sensor normal. For WIs, we assume that the incident light field is constant inside the camera at the MLA plane, hence the time and wavelength integration yields a constant factor. We obtain

$$E(a, b) \propto \frac{1}{2d^2} \left(\frac{A_-}{A_-^2 + 1} - \frac{A_+}{A_+^2 + 1} + \arctan A_- - \arctan A_+ \right), \quad (1)$$

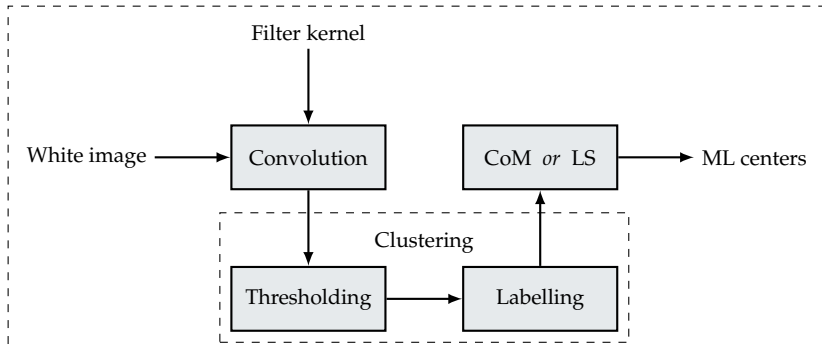
where $A_{\pm} = d^{-1} (\sqrt{a^2 + b^2} \pm r)$. Using (1), we calculate the irradiance underneath each ML. These ML illuminations are then arranged



(a) Dansereau et al. [4].



(b) Cho et al. [3].



(c) Proposed (CoM) and (LS).

Figure 1: ML center detection pipelines.

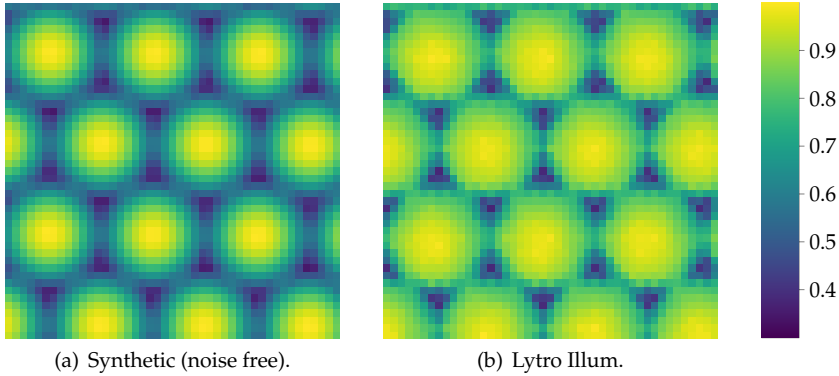


Figure 2: Image sections of normalized WIs with ML radius $r = 7$ px. The shown Lytro WI is the mean of all 34 Lytro WIs.

in a hexagonal grid to compose a WI. Main lens vignetting is added to the WI in an analogous fashion. To simulate irregularities of the grid, we add bivariate Gaussian noise of different variances to the ideal grid point coordinates. Furthermore, we add different levels of Gaussian image noise to the WIs. Finally, the synthesized images are downsampled to 16 bit. The used parameters (ML and main lens radius, focus distance, pixel pitch, grid spacing) are taken from a metadata file of a Lytro Illum camera. The remaining parameters, such as the grid rotation and subpixel offset, are varied throughout the investigation. A comparison of a synthesized and a Lytro WI are shown in Figure 2.

Each Lytro Illum camera provides 34 white images for calibration, taken at different focus and zoom settings. These white images differ mostly in the severity of vignetting: With different focal lengths of the main lens but with unchanged entry pupil, the ML images will be crescent shaped instead of circular at the sensor edges. To be able to detect the true geometric ML centers, a preprocessing of the white images is necessary, for example by averaging all available white images. This shall not be part of this paper, as we focus on the actual ML center detection.

Using the synthetic WIs, the performance of the ML center detection algorithms can be investigated: The coordinates of the true ML centers

are known and can be compared with the estimated ones. We investigate a total of 81 different synthesized WIs of size 1200×1800 px containing around 10 000 MLs each.

3.2 Metrics

For every WI, we calculate the estimated ML centers using the four different ML center detection pipelines for varying parameters such as the convolution filter kernel and kernel radius³ or the size of the structuring element. For every detected center, using the ground truth data, we calculate the distance to the closest known grid point. If the distance is larger than 4 px, the detection is dismissed and regarded as a false positive. Otherwise, the measured distances are collected and the mean and variance of these measured distances are saved along with the corresponding WI and algorithm parameters. If two or more detections occur within a 4 px radius around the same grid point, only the closest one is considered a true positive. We then can calculate the accuracy Q and its standard deviation as the mean and standard deviation of all measured absolute distances of detected to actual ML centers, the precision P , the recall R and the F-measure F ,

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F = 2 \frac{P \cdot R}{P + R},$$

where TP (FP), FN denote the detected true (false) positives and false negatives respectively.

3.3 Results

To be able to compare the performances of every detection pipeline, we first determine the free parameters of each method with the highest detection accuracy. For this, we investigate the mean accuracy over all WIs for different sizes of the kernel respectively the structuring element. The results are shown in Figure 3.

The accuracy in case of the method proposed by Dansereau et al. shows only little dependence on the size of the used filter kernel as well

³ We only investigate kernels of odd size with size = $2 \cdot \text{radius} + 1$.

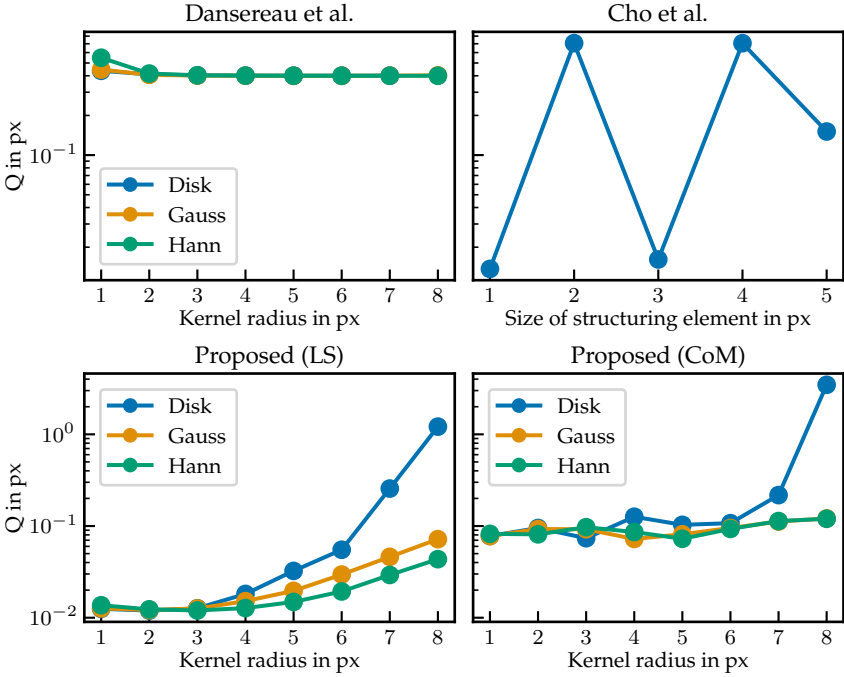


Figure 3: Measured accuracy Q for different kernel radii respectively different sizes of the structuring element for the various detection pipelines.

as the kernel shape. Except for a kernel radius of 1 px, the algorithm performs with an mean accuracy $Q \approx 0.4$ px with a minimum at a kernel radius of 6 px. The algorithm proposed by Cho et al. shows alternating performance depending on the size of the structuring element. A minimum of $Q = 0.0137$ px is reached for a size of 1 px, corresponding to no erosion performed at all. For even-sized structuring elements, the algorithm performs significantly worse than for odd-sized ones. This is caused by shift of the picture due to the fact that even-sized structuring elements do not possess a central pixel: The shift causes a systematic error in the estimation of the ML centers. Overall, we conclude that erosion is not well suited as part of the image preprocessing in the case of

ML center detection. The proposed method using a parabolic LS estimator shows a dependence on the kernel size similar for all kernel types. A minimum is reached for kernel sizes larger than one, but the algorithm performs poorly for large kernel sizes. For large kernel sizes, the disk kernel performs the poorest. This is unsurprising as the disk kernel is not suited as a high frequency filter as it contains high frequencies itself. Overall, an optimum of $Q = 0.0120$ px is reached for a disk kernel with 2 px radius, but similar results are obtained also with the Gauss and Hann kernel with radii of 2 and 3 px. Finally, for the proposed method using a CoM calculation, only slight dependence on the kernel size is observed, with an exception again being a large disk kernel. A minimum is obtained for a Gauss kernel of radius 4 px with $Q = 0.0723$ px. For further comparison, we choose the two best performing parameters for every algorithm.

Using these best performing parameters, detailed performance results are shown in Figure 4. Here, each scatter point represents the result calculated from one WI. Overall, the results cluster in three groups: The pipeline proposed by Dansereau et al. runs fast but has a poor performance w. r. t. the detection accuracy. The method by Cho et al. and the proposed LS method result in very accurate measurements with small standard deviations but with significantly longer runtime. Last, the proposed CoM method performs very fast with reasonable accuracy below 0.1 px. All methods reach a very high F-measure in general. Looking more closely at the recall and precision, we observe that the LS-based methods perform poorly w. r. t. precision compared to the remaining methods. The problem is inherent to the used non-linear LS estimator: Depending on the detected ML clusters, the used optimization algorithm might not converge within the desired tolerance or a given maximum number of iterations yielding an invalid ML center resulting in a FP and thus in a suboptimal precision value⁴. The proposed CoM based method and the one proposed by Dansereau et al. always perform with a precision of exactly one, meaning that no false positives are detected. This can be of importance for applications where the centers are directly used for further calculations. All methods show reasonable recall performance. Further analysis showed that suboptimal recall val-

⁴ Using a linear LS estimator showed even worse numerical stability resulting in a lower precision performance with only a slight improvement of runtime.

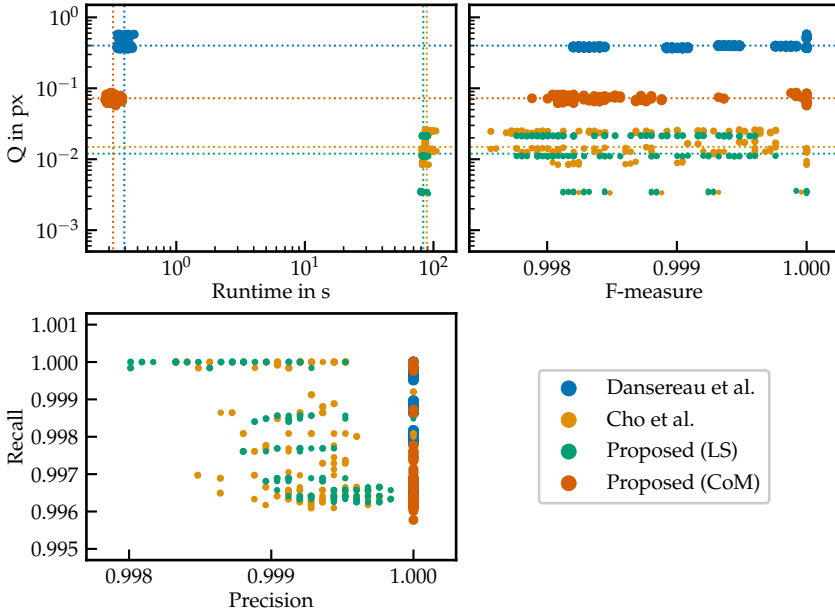


Figure 4: Overall performance comparison of the different detection pipelines. The size of each scatter point is proportional to its measurement’s standard deviation. The shown lines correspond to each dataset’s mean value.

ues, meaning that not all ML are detected, occur exclusively at image edges. Due to cut off ML images, in real applications, one would neglect the image edges anyway.

Finally, we have a closer look at the performance dependence on the WI properties such as rotation, grid and image noise. The results are shown in Figure 5. Overall, we again observe that the LS methods are the most accurate, with the proposed method performing slightly better than the one by Cho et al. [3]. The method by Dansereau performs worst and the proposed CoM method’s performance lies in between. For grid rotation and grid noise, there is no influence on the detection accuracy for either one of the pipelines. This is not a surprise since the detection pipelines are all operating locally and every ML is detected and evaluated separately. Rotation and grid noise do not have a systematic influ-

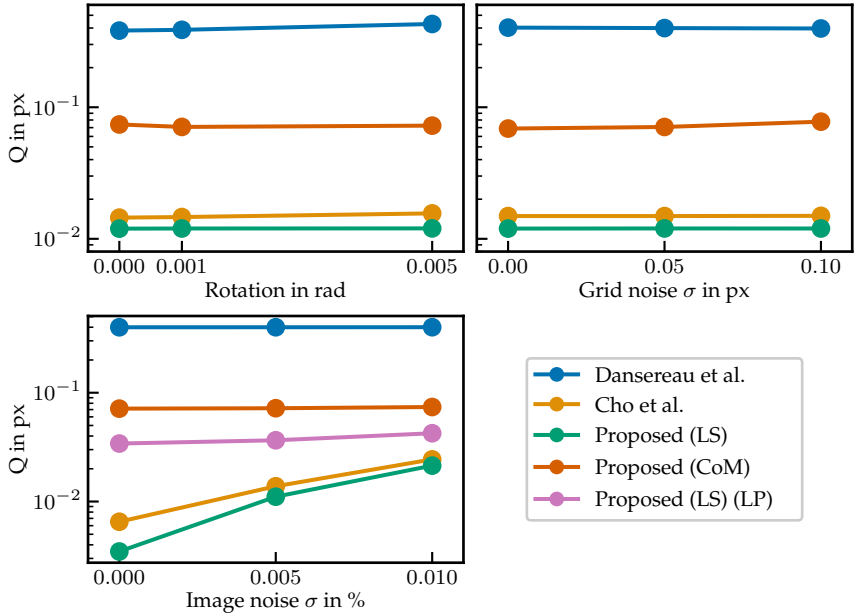


Figure 5: Performace comparison of the different detection pipelines for different WI properties.

ence on the local ML image. Image noise, on the other hand, does have a local impact on each ML image. We observe a significant influence on the LS based methods. Both the proposed LS and the method by Cho et al. perform considerably worse with increasing image noise while the methods by Dansereau et al. and the proposed CoM method are invariant to image noise. The latter perform a low-pass filtering with kernel radii of about 7 px whereas the former either perform no filtering (Cho et al.) or use very small filters (proposed LS method). Increasing the filter radius, and hence decreasing the cutoff frequency, for the proposed LS method to 7 px (depicted in Figure 5 by the (LP) label) confirms that the filtering successfully suppresses image noise. In doing so, the accuracy of the proposed LS pipeline drops significantly, resulting in an accuracy close to the proposed CoM method.

4 Conclusions

We have investigated four ML center detection pipelines, two of which have not been previously discussed in the literature. Overall, the proposed methods perform better than those having previously been proposed in the literature: The method proposed by Dansereau et al. [4] shows a fast runtime, high precision, but poor accuracy. The proposed CoM based method outperforms the previous in terms of accuracy and runtime while performing similarly in terms of precision and recall. On the other hand, the method proposed by Cho et al. [3] performs with a high accuracy but long runtimes and suboptimal precision. The proposed LS based method again outperforms the former in terms of accuracy while performing similarly in the remaining metrics. With respect to WI parameters and noise, the method by Dansereau et al. and the proposed CoM method are the most robust. Concluding, the proposed CoM method is well suited for most applications.

References

1. E. H. Adelson and J. Y. Wang, "Single lens stereo with a plenoptic camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 99–106, 1992.
2. R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report*, vol. 2, no. 11, pp. 1–11, 2005.
3. D. Cho, M. Lee, S. Kim, and Y.-W. Tai, "Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3280–3287.
4. D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1027–1034.

Schnelle und präzise Segmentierung von beweglichen und morphologisch variablen Objekten am Beispiel der Deflagrationsdetektion

Fast and precise segmentation of moving and morphologically variant objects by the example of deflagration detection

Jakob Krooß¹, Felix Kümmerlen² und Alexander Fay¹

¹ Helmut-Schmidt-Universität/Universität der Bundeswehr,
Institut für Automatisierungstechnik,
Holstenhofweg 85, 22043 Hamburg

² Wehrwissenschaftliches Institut für Schutztechnologien,
ABC-Schutz Munster
Humboldtstraße 100, 29633 Munster

Zusammenfassung Bei der Entwicklung von Segmentierungsalgorithmen sind Sensitivität und Präzision oftmals zwei sich widersprechende Ziele. In diesem Beitrag wird eine Methode vorgestellt, die durch die Berechnung zweier wechselwirkender Segmentierungen diese Limitierung aufhebt. Die Methode wird am Beispiel der Deflagrationsdetektion erläutert und mit einem klassischen Algorithmus verglichen.

Schlagwörter Deflagration, Detektion, Segmentierung.

Abstract For segmentation algorithms, sensitivity and precision are often limiting each other. In this paper, a new method that cancels this limitation by calculation of two different, interactive segmentations, is introduced. It is explained by example of deflagration detection and compared to a classic algorithm.

Keywords Deflagration, detection, segmentation.

1 Einleitung

Bei der Segmentierung sich schnell bewogender und potentiell in der Form variabler Objekte ist eine hohe Sensitivität entscheidend, um diese rechtzeitig und zuverlässig zu erfassen. Unter der hohen Sensitivität kann allerdings die Qualität der Segmentierung leiden, was insbesondere eine anschließende Lokalisierung im Bild erschwert. Eine deutlich präzisere Segmentierung wäre hier wünschenswert. Am Beispiel der Deflagrationsdetektion soll in diesem Beitrag eine Methode vorgestellt werden, die durch zwei miteinander wechselwirkende Segmentierungen die Vorteile einer sensitiven und einer präzisen Auswertung miteinander vereint.

Bei der Vermengung von exotherm oxidierbaren (brennbaren) und feinkörnigen, bzw. flüssigen oder gasförmigen Materialien und einem Oxidationsmittel wie Sauerstoff können explosionsfähige Atmosphären entstehen [1]. Entzünden sich diese, kommt es zu explosionsartigen Verbrennungsreaktionen. Diese werden als Deflagrationen bzw. Verpuffungen bezeichnet. Insbesondere in geschlossenen Räumen besitzen Deflagrationen ein gewaltiges Zerstörungspotential. Deshalb gilt es, Deflagrationen bereits in ihrer Entstehungsphase zu detektieren und zu löschen, um die durch sie entstehenden Schäden an Mensch und Material zu minimieren. Anwendungsgebiete sind dabei z.B. die Besatzungsräume militärischer Fahrzeuge, aber auch verschiedene verfahrenstechnische Anlagen. Mit dem Verbot des in der Vergangenheit zum Löschen genutzten, hoch wirksam reaktionshemmenden, aber auch stark umweltschädlichen Gases Halon muss künftig vermehrt auf weniger wirksame Löschsysteme wie Hochdrucksysteme auf Wasserbasis zurückgegriffen werden [2]. Diese können ihre Löschwirkung nur auf einen begrenzten Bereich ausüben und dementsprechend eine entstehende Deflagration nur dann effektiv hemmen, wenn auch deren Position bekannt ist.

Vor diesem Hintergrund wurde in vorangegangenen Arbeiten [3, 4] bereits ein Algorithmus entwickelt, um Deflagrationen in den Aufnahmen mehrerer Hochgeschwindigkeitskameras zu detektieren und lokalisieren. Ein Fokus lag dabei darauf, die benötigte Rechenzeit minimal zu halten und so eine Echtzeitüberwachung zu ermöglichen. Der zweistufige Algorithmus klassifiziert zuerst jedes Pixel anhand seiner Intensität und deren Anstieg im Vergleich zum letzten Bild bezüglich seiner

Deflagrationsähnlichkeit und nutzt die resultierende Information (Anteil deflagrationsähnlicher Pixel und deren Anstieg) anschließend, um jedes Kamerabild den Klassen „Deflagration“ bzw. „keine Deflagration“ zuzuordnen. Durch die Lage der detektierten Pixel im Bild lässt sich zusätzlich eine Lokalisierung der Deflagration durchführen [4].

Der in [3,4] gezeigte Ansatz bringt allerdings genau das oben genannte Problem mit sich: Bei einer hohen Sensitivität der pixelweisen Auswertung kann die Deflagration schon in einem sehr frühen Stadium erkannt werden, es besteht jedoch ein höheres Risiko der Fehlklassifikation von Pixeln und dementsprechend auch von Fehlalarmen bei der Gesamtauswertung. Insbesondere die Lokalisierung wird durch fehlerklassifizierte Pixel deutlich ungenauer. Eine geringe Sensitivität hingegen ermöglicht eine deutlich präzisere Lokalisierung, erschwert aber eine frühzeitige Detektion.

Im Folgenden wird zunächst der Stand der Forschung im Bereich Deflagrations-Detektionstechnik dargestellt. Anschließend wird der auf [3,4] basierende, nun weiterentwickelte Ansatz zur Segmentierung von deflagrationsartigen Verbrennungen detailliert vorgestellt. Die Vorteile im Vergleich zum Vorgängeralgorithmus werden an Beispielen demonstriert. Abschließend werden neben der Anwendung zur Deflagrationsdetektion auch Ansätze zur Anwendung in anderen Bereichen diskutiert.

2 Stand der Forschung

Die Verhinderung, Detektion und Bekämpfung von Deflagrationen ist bereits seit mehreren Jahrzehnten Gegenstand der Forschung. Dies ist nicht zuletzt auf einige größere Unfälle in der Prozessindustrie zurückzuführen, weshalb diverse Methoden für diesen Bereich entwickelt wurden. In [1] wird ein guter Überblick über baulichen und anlagentechnischen Brandschutz sowie die gängigsten Brandbekämpfungsmittel gegeben.

Die Detektion von Deflagrationen muss dabei klar abgegrenzt werden von klassischen Brandmeldeanlagen, also der Detektion von in gemäßigter Geschwindigkeit ablaufenden Verbrennungen bzw. Feuer. Brandmeldeanlagen basieren auf einer großen Bandbreite an Sensoren für Rauch, Temperatur und Licht [5]. Die meisten dieser Sensoren sind

jedoch aufgrund der Verzögerungszeiten bei der Detektion von teilweise mehreren Sekunden nicht geeignet, um die schlagartigen Verbrennungsprozesse von Deflagrationen rechtzeitig zu erfassen. Lediglich auf Fotodioden basierende Lichtsensoren sind mit ihrer schnellen Messung in der Lage, Deflagrationen rechtzeitig zu erkennen. Sie konzentrieren sich dabei zumeist auf bestimmte Wellenlängen im Infrarotbereich, im UV-Bereich oder einer Kombination aus beidem [5].

Allen Sensoren ist gemeinsam, dass sie keine zuverlässige Lokalisierung eines Feuers oder einer Deflagration ermöglichen. Nicht zuletzt deswegen gibt es zahlreiche Arbeiten zur Erkennung oder Segmentierung von Feuer, bzw. Flammen, auf Basis von Überwachungskameras und Bildverarbeitung. Wesentlicher Vorteil dieses Ansatzes ist die Gewinnung zusätzlicher Informationen über die Verbrennung, insbesondere des Ortes. Ein guter Überblick wird z.B. in [6] gegeben. Entsprechende Algorithmen werten das Bild unter anderem anhand von spektralen Eigenschaften (z.B. [7–10]), einer Analyse der Flickerfrequenzen (z.B. [9]) und/oder eines Hintergrundmodells (z.B. [7, 10–12]) aus. Die meisten publizierten Verfahren sind wegen Ihres Fokus auf in moderater Geschwindigkeit ablaufende Verbrennungen zu langsam und/oder nicht in der Lage, schlagartig ablaufende Verbrennungen rechtzeitig zu detektieren. Einige Ansätze lassen sich jedoch auf das Problem der Deflagrationsdetektion übertragen. So bilden z.B. Teile der Algorithmen in [7, 12] einen Bestandteil des Algorithmus in [4] und somit auch der im nächsten Kapitel vorgestellten Methode.

Die Erkennung von Deflagrationen in Fahrzeugen ist insbesondere von militärischem Interesse und wurde lange Zeit durch Lichtsensoren für den Infrarot- und UV-Bereich bewerkstelligt [2]. Bildverarbeitungs-basierte Lösungen zur Deflagrationsdetektion wurden in den Arbeiten [3, 4] vorgestellt, auf denen der im Folgenden vorgestellte Algorithmus basiert.

3 Methoden

Der in [3, 4] vorgestellte Algorithmus lässt sich im Wesentlichen in vier Phasen unterteilen: Die pixelweise Segmentierung, die Detektion, die kameraweise 2D-Lokalisierung und die globale 3D-Lokalisierung.

Die bereits gegebene, sensitive Segmentierung wird nur in wenigen

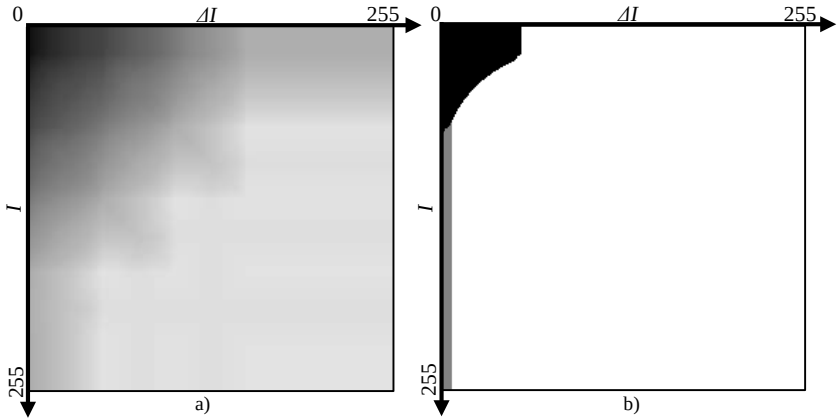


Abbildung 1: (a) Die auf einer Fuzzy-Logik basierende Look-Up-Tabelle. Die Zugehörigkeit zur Klasse der deflagrationsähnlichen Pixel wird auf einer Skala von 0 (schwarz) bis 1 (weiß) bewertet. Der maximale auftretende Wert ist 0,8827. (b) Die Elemente der Look-Up-Tabelle, deren zugehörige Pixel durch den Schwellwert $\theta = 0,3$ als deflagrationsähnlich eingestuft werden (weiß). Der graue Streifen entspricht Bereichen im Bild, die im Vergleich zum vorherigen Bild nur marginal heller geworden sind und wird im Gegensatz zu [4] nicht berücksichtigt.

Punkten ergänzt und weiterhin für die Detektion eingesetzt. Neu vorgestellt wird eine weitere, auf dieser basierende und deutlich spezifischere Segmentierung, die anschließend die Grundlage für die 2D-Lokalisierung bildet. Zur besseren Unterscheidung werden die beiden Segmentierungen im Folgenden als Detektionssegmentierung map^{det} und Lokalisierungssegmentierung map^{loc} bezeichnet. Beide Segmentierungen und ihre Wechselwirkungen untereinander werden im folgenden detailliert beschrieben. Auf die Detektion sowie die Lokalisierung selbst hingegen wird hier nicht weiter eingegangen werden, Details können diesbezüglich z.B. in [4] gefunden werden.

Die Detektionssegmentierung map^{det} macht sich mit dem kontinuierlichen Anstieg der Helligkeit die wesentliche Eigenschaft einer Deflagration zu Nutze: Für jedes Pixel an der Position x, y im Eingangsbild f wird über eine heuristische, auf einer Fuzzy-Logik basierende Look-Up-Tabelle die Zugehörigkeit zur Klasse der deflagrationsähnlichen Pi-

xel bestimmt. Die beiden Eingabeparameter sind durch die aktuelle Intensität $I^{x,y}$ sowie den Anstieg der Intensität im Vergleich zum letzten Bild $\Delta I^{x,y} = I_i^{x,y} - I_{i-1}^{x,y}$ gegeben. Vereinfacht dargestellt steigt die Zugehörigkeit mit zunehmenden Werten für $I^{x,y}$ und $\Delta I^{x,y}$. Eine Darstellung der aktuell verwendeten Look-Up-Tabelle ist in Abb. 1 (a) zu finden. Über den Schwellwert θ wird anschließend zwischen deflagrationsähnlichen und nicht deflagrationsähnlichen Pixeln unterschieden (siehe Abb. 1 (b)).

Da dieses einfache, aber im Vergleich zur benötigten Rechenzeit sehr effektive Modell in dieser Form zu einer großen Anzahl an Fehlklassifikationen führt, werden weitere Ausschlusskriterien geprüft. Nur wenn alle der nachfolgenden Bedingungen erfüllt werden, wird ein entsprechendes Pixel in die Detektionssegmentierung map^{det} aufgenommen:

1. **Spektrale Filterung:** Insbesondere in der Entstehungsphase einer Deflagration ist der Rotkanal überproportional ausgeprägt. Dementsprechend werden nur Pixel klassifiziert, die dies widerspiegeln:

$$S_i^{RGB} = \begin{cases} 1 & r > b \cap r > g \cap r > \bar{r} \\ 0 & \text{sonst} \end{cases} \quad (1)$$

Dabei bezeichnet r den Rotkanalanteil des Eingangsbildes f , g den Grünkanalanteil, b den Blaukanalanteil und $\bar{r}$ den über das gesamte Bild gemittelten Wert für den Rotkanal. Zusätzlich zu dieser spektralen Auswertung im RGB-Raum wird das Ursprungsbild in den YCbCr-Farbraum übertragen. Die spektrale Filterung berechnet sich anschließend leicht modifiziert nach [7] durch

$$S_i^{YCbCr} = \begin{cases} 1 & \bar{Y} > 17 \cap Y > \bar{Y} \cap Cb < \bar{Cb} \cap Cr > \bar{Cr} \\ 0 & \text{sonst} \end{cases}, \quad (2)$$

wobei Y die Grundhelligkeit, Cb die Blau-Gelb-Chrominanz, Cr die Rot-Grün-Chrominanz und $\bar{Y}$, $\bar{Cb}$, $\bar{Cr}$ deren Mittelwerte über das gesamte Bild beschreiben.

2. **Vordergrund/Hintergrund-Modell:** Ein großer Teil der fehlklassifizierten Pixel entsteht durch helle und sich im Vergleich zu Deflagrationen langsam bewegende und nur unwesentlich heller werdende Objekte wie weiße Autos oder rote Warnwesten. Um deren

Segmentierung zu vermeiden lässt die Look-Up-Tabelle keine positive Klassifikation unterhalb einer gewissen Intensitätsänderung zu (siehe Abb. 1 b)). Die Klassifikation kann aber zusätzlich durch die Berechnung eines auf Teilen von [12] basierenden Hintergrundmodells verbessert werden: Im ersten Bild wird der Hintergrund B_0 mit dem Rotkanalbild r_0 initialisiert. Der Hintergrund jedes folgenden Bildes berechnet sich durch

$$B_i = 0.4 B_{i-1} + 0.6 r_i, \quad (3)$$

wobei r_i wie bei der Initialisierung vor der Berechnung leicht geglättet wird. Über einen variablen Schwellwert wird anschließend, ähnlich wie in [13], der Vordergrund bestimmt, in dem sich nur die Pixel des Bildes befinden, in denen eine deutliche Änderung stattgefunden hat:

$$F_i = \begin{cases} 1 & |r_i - B_i| > \Delta F^{x,y} \\ 0 & \text{sonst} \end{cases} \quad \text{mit } \Delta F^{x,y} = \begin{cases} 12 & r_i^{x,y} > 130 \\ 28 & r_i^{x,y} > 85 \\ 56 & \text{sonst} \end{cases} . \quad (4)$$

Die ausschließliche Verwendung des Rotkanalbildes erklärt sich dadurch, dass wesentliche Bildinformationen bezüglich Flammen und Deflagrationen im Rotkanal zu finden sind, und erhöht die Robustheit gegenüber anderen Bildsignalquellen. Sowohl die angepasste Look-Up-Tabelle als auch das Hintergrundmodell haben allerdings auch einen Nebeneffekt: Pixel im Kernbereich einer Deflagration, die den maximalen Helligkeitswert erreicht haben und dementsprechend nicht mehr heller werden können, werden nicht mehr berücksichtigt. Es entstehen „Löcher“ in der Segmentierung.

map^{det} enthält also alle Pixel, die den Schwellwert in der Look-Up-Tabelle überschritten haben und deren Einträge in den Matrizen S^{RGB} , S^{YCbCr} und F jeweils auf 1 gesetzt sind. Eine optionale Opening-Operation mit kleinem Kernel kann bei schwierigen Lichtverhältnissen die Qualität der Segmentierung verbessern.

In die Lokalisierungssegmentierung map^{loc} werden anschließend nur die Pixel aus map^{det} übernommen, deren Rotkanal r bereits den maximalen Wert angenommen hat. Die Stabilität des Algorithmus kann

dabei deutlich erhöht werden, wenn nicht nur die aktuelle Detektionssegmentierung, sondern auch die aus den n vorherigen Bildern berücksichtigt werden.

Enthält die Lokalisierungssegmentierung aus der vorherigen Iteration bereits Pixel, so werden diese auch bei fehlender Detektion in der aktuellen Iteration gesetzt, wenn der Rotkanal dort noch immer den Maximalwert angenommen hat. Auf diese Weise bleiben auch die Kernbereiche einer Deflagration segmentiert. Wird ein Bildbereich wieder dunkler, werden die Pixel entsprechend wieder auf null gesetzt:

$$map_i^{loc} = \begin{cases} 1 & (map_{i-1}^{loc} \cup map_i^{det} \cup map_{i-1}^{det} \cup \dots \cup map_{i-n}^{det}) \cap r = 255 \\ 0 & \text{sonst} \end{cases} \quad (5)$$

Anschließend kann auf das Ergebnis optional eine Closing-Operation mit einem vergleichsweise großen, runden Kernel angewendet werden. Dies kann die Qualität der Segmentierungen teilweise erhöhen, sollte aber aufgrund der zusätzlichen Berechnungszeit nur angewandt werden, wenn die Umgebungsbedingungen es erzwingen.

Um die Vermeidung von Löchern im Kern der Deflagration auch in der Detektionssegmentierung zu nutzen, werden alle Pixel, die in map^{loc} gesetzt sind, abschließend auch in map^{det} gesetzt.

4 Ergebnisse

Abb. 2 zeigt die vorgestellten Segmentierungen beispielhaft für drei unterschiedliche, unter Versuchsbedingungen erzeugte Bilder von Deflagrationen.

Abb. 2 a) zeigt eine Deflagration im Inneren eines abgedunkelten Raumes. Für das Bild der Deflagration in Abb. 2 b) wurde der Versuchsaufbau durch einen zusätzlichen Leuchtstrahler ergänzt. Die Deflagration ist zudem in ihrer Ausbreitung weiter fortgeschritten und teilweise verdeckt. Die Deflagration in Abb. 2 c) wurde im Freien aufgenommen und stellt aufgrund des starken Umgebungslichts im Vergleich zur eher schwach ausgeprägten Deflagration selbst und zahlreicher Hintergrundobjekte bei der Segmentierung eine besonders große

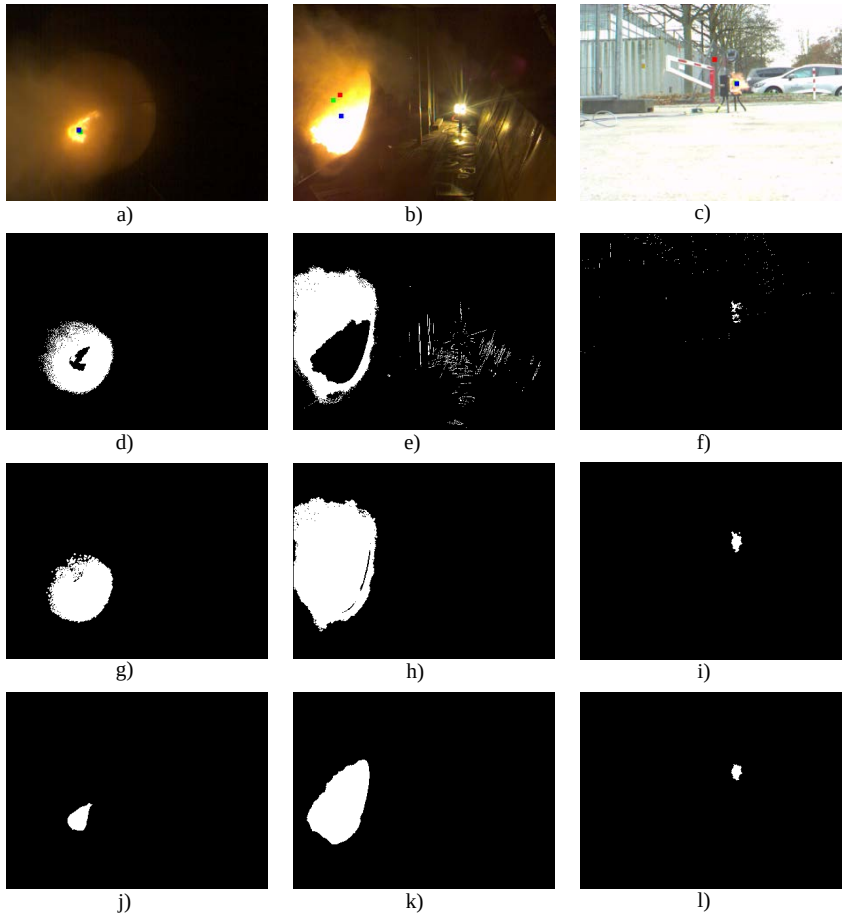


Abbildung 2: Die Segmentierungen von drei beispielhaften Deflagrationen. (a-c) Eingangsbilder f . (d-f) Zugehörige Segmentierungen nach [4]. (g-i) Zugehörige Detektionssegmentierungen map^{det} . (j-l) Zugehörige Lokalisierungssegmentierungen map^{loc} .

Herausforderung dar. Für alle Bilder wurden die gleichen Parameter, Faltungskernel und Schwellwerte verwendet.

Abb. 2 d-f) zeigen die zugehörigen, „klassisch“ berechneten Segmen-

tierungen nach [4]. Gut zu erkennen sind insbesondere die Detektions-„Löcher“ in den Zentren der größeren Deflagrationen.

Abb. 2 g-h) zeigen die modifizierte Detektionssegmentierungen map^{det} . Im direkten Vergleich fällt auf, dass in den Zentren der Deflagrationen keine Löcher mehr auftreten. Zusätzlich wurde der Anteil an fehlklassifizierten Pixeln durch Reflexionen und Hintergrundobjekte stark reduziert. Dies ist zum einen auf die im Zuge des Hintergrundmodells angepasste Look-Up-Tabelle, zum anderen auf die zusätzlich eingeführte Opening-Operation zurückzuführen.

In Abb. 2 i-k) sind die neu eingeführten Lokalisierungssegmentierungen map^{loc} dargestellt. Im Gegensatz zu den Detektionssegmentierungen beschreiben sie nur den hellen Kern der Deflagration.

Die roten Punkte in Abb. 2 a-c) stellen die Schwerpunkte der Detektionssegmentierungen nach [4] dar, die grünen die Schwerpunkte der modifizierten Detektionssegmentierungen und die blauen die der Lokalisierungssegmentierungen.

5 Diskussion

Die Ergebnisse des vorangehenden Abschnitts veranschaulichen, dass die Generierung zweier unterschiedlicher Segmentierungen es ermöglicht, eine deutlich genauere Lokalisierung der Deflagration durchzuführen, ohne dabei auf die hohe Sensitivität des Algorithmus in [3, 4] zu verzichten; die Schwerpunkte der Lokalisierungssegmentierungen geben die tatsächliche Position der Deflagrationen teilweise deutlich besser wieder, die für die Sensitivität der Detektion entscheidende Anzahl detektierter Pixel in der Detektionssegmentierung verändert sich aber nur unwesentlich. Das Vorhandensein einer gewissen Menge an Pixeln in der Lokalisierungssegmentierung, die den hellen Kern einer Deflagration beschreibt, kann zusätzlich eine notwendige Bedingung für die Detektion sein und so die Häufigkeit von Fehlalarmen reduzieren. Die Lokalisierungssegmentierung bildet zudem eine deutlich geeignetere Grundlage für eine Abschätzung des Volumens oder weiterer Parameter wie der Ausbreitungsrichtung der Deflagration. Die dafür zusätzlich benötigte Abstandsinformation wird durch das Multi-Kamera-System bestimmt.

Die beiden Segmentierungen sind also für die Lösung verschiedener

Aufgaben optimiert, verlieren aber durch ihre starke Wechselwirkung nicht den Bezug zueinander. Die Lokalisierungssegmentierung bietet sich auch zur Segmentierung von Feuer bzw. Flammen an. Wesentlicher Vorteil gegenüber vielen anderen Algorithmen zur Flammenerkennung ist dabei die relativ hohe Segmentierungsqualität bei extrem kurzer Berechnungszeit, was auch zeitkritische Anwendungen ermöglicht.

Der Ansatz, zwei wechselwirkende Segmentierungen mit unterschiedlicher Zielsetzung zu nutzen, lässt sich auf weitere Anwendungsgebiete in der Bildverarbeitung übertragen, beispielsweise das Verhindern von Kollisionen mit Fußgängern beim autonomen Fahren [14] oder allgemein das Erkennen von aus Kamerasicht schnell größer werdenden, in der Form variablen Objekten. Für das zu erkennende Objekt müssen sensitive Eigenschaften gefunden werden, deren Änderung eine Bewegung/Ausweitung des Objektes in entsprechende Bereiche des Bildes impliziert. Zur anschließenden Berechnung der Lokalisierungssegmentierung werden dann weitere, deutlich spezifischere Eigenschaften benötigt, die das Vorhandensein des Objektes im entsprechenden Bildbereich bestätigen. Die genaue Art der Eigenschaften sowie der Wechselwirkungen zwischen den beiden Segmentierungen kann sich dabei je nach Anwendung unterscheiden.

6 Zusammenfassung und Ausblick

Im vorliegenden Beitrag wurde beispielhaft an der Deflagrationsdetektion eine neue Methode vorgestellt, die durch die Verwendung zweier wechselwirkender Segmentierungen die gegenseitige Einschränkung von Sensitivität und Präzision aufhebt. Die Vorteile wurden anhand von Beispielen demonstriert.

Durch die zusätzlichen Berechnungen steigt jedoch auch der Rechenaufwand. Es bleibt zu zeigen, dass der vorgestellte Algorithmus trotzdem die zeitlichen Anforderungen erfüllt, die an einen Algorithmus zur Deflagrationsdetektion gestellt werden. Weiterhin ist zu prüfen, auf welche Teile des Algorithmus wegen Redundanz verzichtet werden kann, ohne die Segmentierungsqualität wesentlich einzuschränken.

Neben der Segmentierung werden auch die Detektion und die Lokalisierung von Deflagrationen wesentliche Gegenstände weiterer Forschungen sein.

Literatur

1. VDI-Richtlinie 2263: Staubbrände und Staubexplosionen; Gefahren, Beurteilung, Schutzmaßnahmen. Verein deutscher Ingenieure, 2017.
2. A. Kim, Z. Liu, and G. Crampton, "Study of explosion protection in a small compartment," *Fire Technology*, vol. 43, no. 2, pp. 145–172, 2007.
3. T. Schröder, K. Krüger, and F. Kümmerlen, "Image processing based deflagration detection using fuzzy logic classification," *Fire Safety Journal*, vol. 65, pp. 1–10, 2014.
4. T. Ernst, F. Kümmerlen, and A. Fay, "Real-time image processing based deflagration detection and localization," *Future Security*, vol. 11, 2016.
5. D. P. Nolan, *Handbook of Fire and Explosion Protection Engineering Principles for Oil Gas, Chemical and Related Facilities*, 2nd ed. Elsevier, 2011.
6. A. E. e. a. Cetin, "Video fire detection - review," *Digital Signal Processing*, vol. 23, no. 6, pp. 1827–1843, 2013.
7. J. Seo, M. Kang, C. H. Kim, and J.-M. Kim, "An optimal many-core model-based supercomputing for accelerating video-equipped fire detection," *The Journal of Supercomputing*, vol. 71, no. 6, pp. 2275–2308, 2015.
8. T.-H. Chen, P.-H. Wu, and Y.-C. Chiou, "An early fire-detection method based on image processing," *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 1707–1710, 2004.
9. X. Qi and J. Ebert, "A computer vision-based method for fire detection in color videos," *International Journal of Imaging*, vol. 2, no. S09, pp. 22–34, 2009.
10. T. Celik, "Fast and efficient method for fire detection using image processing," *ETRI Journal*, vol. 32, no. 6, pp. 881–890, 2010.
11. B. U. Töreyn, Y. Dedeoğlu, U. Güdükbay, and A. E. Cetin, "Computer vision based method for real-time fire and flame detection," *Pattern Recognition Letters*, vol. 27, no. 1, pp. 49–58, 2006.
12. S. Verstockt, I. Kypraios, P. De Potter, C. Poppe, and R. Van de Walle, "Wavelet-based multi-modal fire detection," in *Signal Processing Conference, 2011 19th European*. IEEE, 2011, pp. 903–907.
13. B. Jiang, Y. Lu, X. Li, and L. Lin, "Towards a solid solution of real-time fire and flame detection," *Multimedia Tools and Applications*, vol. 74, no. 3, pp. 689–705, 2015.
14. T. Gandhi and M. M. Trivedi, "Pedestrian collision avoidance systems: A survey of computer vision based recent studies," in *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*. IEEE, 2006, pp. 976–981.

The HCI flow metrics: A novel approach for benchmarking optical flow

Daniel Alexander Brock¹ and Bernd Jähne²

Universität Heidelberg, Heidelberg Collaboratory for Image Processing (HCI),
Berliner Straße 43, 69120 Heidelberg

Abstract When choosing an optical flow algorithm or its parameters it is essential to measure the performance in a meaningful way depending on the application. We show that the widely used mean endpoint error and badpix error metrics lead to the development of algorithms performing poor at fine structures and that the latter are omnipresent. We propose methods to extract challenging areas from dense ground truth data and a set of metrics measuring performance at these areas. We show that these metrics lead to different rankings than the commonly used metrics.

Keywords Optical flow, ground truth, performance evaluation.

1 Introduction

Optical flow is a crucial part of machine vision and therefore subject to broad scientific research. For a pair of images the corresponding optical flow can be described as the vector field of apparent motion [1]. It can be used in many applications, for example in motion estimation [2] [3] and object tracking [4]. For automatically measuring the precision of flow algorithms, sets of image pairs and associated exact optical flow fields are required. The latter are called ground truth or reference data. In this paper we introduce metrics for benchmarking specific properties of flow algorithms like behaviour near object edges and thin objects.

2 Related work

Benchmarking of flow algorithms is usually done by computing pixel-wise metrics where the difference between the reference data and the algorithm's result is measured by some loss function, e.g. the Euclidean norm [5], angular error [6] or a combination of those [7] are taken and then this local measure is averaged over all pixels. McCane et. al [8] introduce two error metrics addressing the problem that areas with large flow vectors have a larger influence on the mean endpoint error than areas with small flow vectors, but not the problem that different areas of an image may pose different challenges to algorithms.

For the task of benchmarking optical flow algorithms there are several public ground truth datasets and corresponding benchmarks available, for example the Middlebury Flow Dataset [9], the MPI Sintel Dataset [10], the KITTI Vision Benchmark Suite [11], the HCI Benchmark Suite [12].

Some benchmarks compute separate metrics for different image regions: Middlebury and HCI compute flow metrics near flow discontinuities, Sintel shows results for different ranges of ground truth flow vector length and KITTI separates background and foreground.

We show that these metrics are not sufficient for determining the most suitable algorithm for each application. Honauer et. al. [13] show that pixel-wise error metrics are not sufficient for comparing stereo algorithm and introduce a set of stereo metrics to address this problem. We extend their approach to the field of optical flow.

3 Geometry-aware metrics for optical flow evaluation

In order to select an optical flow algorithm or a set of parameters for such an algorithm it is crucial to measure and compare the performance of different algorithms w.r.t. the specific application.

Applying this approach to optical flow requires modification for two reasons: First, in optical flow the movement of the camera is unknown and has 6 degrees of freedom, whereas in stereo it is known and has 1 degree of freedom. Second, in stereo both images show the exact same scene with all objects stationary while in flow every object might move independently. Therefore we develop corresponding metrics for mea-

asuring the performance of flow algorithms.

Our contribution in this paper is twofold:

- We provide simple methods to automatically identify areas near discontinuities, planar surfaces, smooth surfaces and fine structures. We only use a (semi-) dense ground truth flow field and we do not require any human annotation in the process.
- For each computed subset we introduce metrics measuring specific details of algorithm results like edge fattening, surface bumpiness / smoothing and fine structure fattening / thinning.

We use a set of publicly available optical flow algorithms which we run on the training data of the Middlebury flow dataset to show that ranking depends on the chosen metric. Therefore our metrics are suitable to reveal specific strengths and weaknesses of individual flow algorithms.

In this section we introduce our new metrics, explain how the motion discontinuities, fine structures and smooth surfaces are extracted from the ground truth data and show how the metrics are computed. For each metric zero is the best possible result and higher values mean worse algorithm performance.

3.1 Flow discontinuities

Discontinuities in the flow field occur due to two reasons: First, static objects in the scene which are at different distances to the (moving) camera. Second, different objects move in different directions or with different speed. We don't distinguish between these cases since it would require additional information about the scene and we want our metrics to be easily applicable given only an algorithm result and the ground truth data.

We compute flow discontinuities in three steps: First we compute the Frobenius norm of the Jacobi matrix of the ground truth flow field. Second, we use adaptive thresholding to identify pixels where the gradient is stronger than the gradients in their neighborhood. Third we discard small 8-connected components of detected edge pixels.

For each pixel adjacent to a flow edge we identify the opposite side of the edge and propagate the ground truth flow, see figure 1 b. This

allows us to measure edge fattening where an algorithm extends the flow of a highly textured region into an adjacent region with lower texture. For each pixel in the identified regions we measure the euclidean distance between algorithm result and ground truth as well as between algorithm result and extrapolated ground truth. We count the number of pixels where the algorithm is closer to the propagated ground truth than to the original ground truth and calculate the percentage of bad pixels.

Figure 1 d shows a visualization of the edge fattening metric. Note that the area of evaluation is smaller than the area of propagated ground truth (b) since we only evaluate at pixels where foreground and background are sufficiently different.

Honauer et.al. [13] differentiate foreground-fattening and foreground-thinning (= background-fattening). In the case of optical flow it is not obvious from the ground truth data which area shows the foreground object so we would need additional information about which object is the foreground object, therefore we measure both effects combined.

3.2 Smooth surfaces

Motion estimation is considered easier at continuous surfaces than near motion edges. We argue that this notion stems from the fact that in most benchmarks only some kind of mean error of all pixels is calculated which is sensitive to outliers near edges. In some applications like industrial surface inspection via 3D reconstruction another aspect is important: Reconstructed surfaces should be as smooth or bumpy as the actual object. Therefore we propose metrics to measure bumpiness and smoothing. We define the set $\mathcal{M}_s$ as all pixels which have a euclidean distance of more than five pixels to the detected motion discontinuities. On this subset we calculate the Jacobi matrices J_{gt} and J_{algo} of ground truth and flow result and compare their Frobenius norms. This gives us the metric S_{smooth} which measures how much smoother the algorithm result is than it should be and S_{bumpy} which measures how much

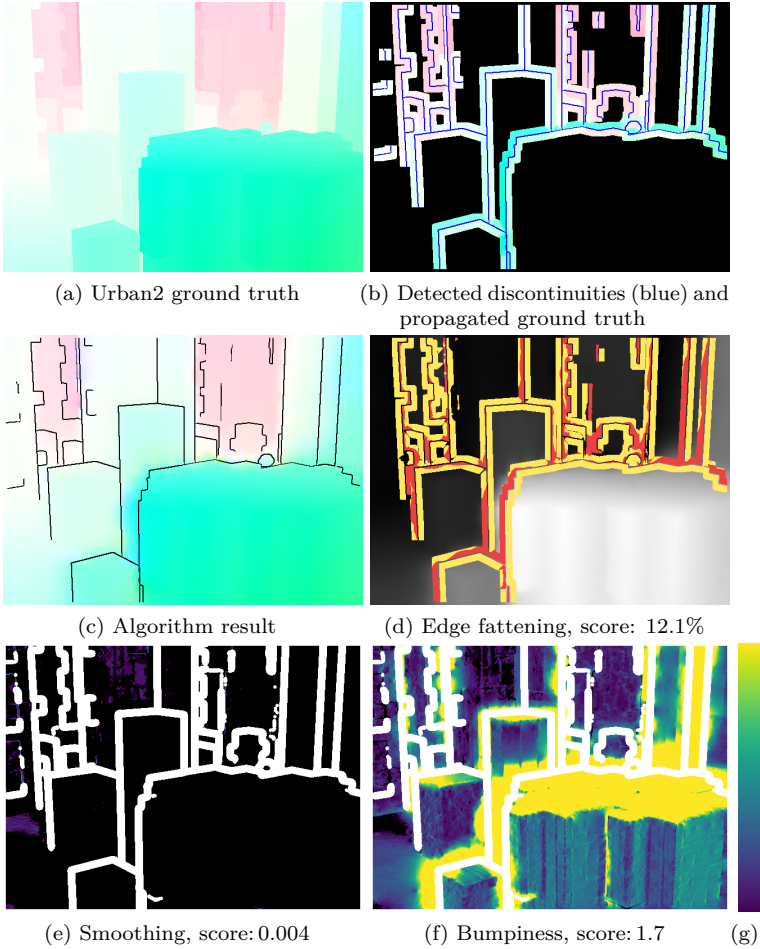


Figure 1: a) the ground truth for frame 10 of the Urban2 sequence from the Middlebury optical flow dataset. Flow vectors are color coded using the Middlebury visualization code [9]. For input images see <http://vision.middlebury.edu/flow/data/> b) our evaluation mask with ground truth propagated across flow discontinuities. c) the result of a Horn&Schunck algorithm with a multi-scale strategy [14]. d) visualization of our fattening metric: yellow = no fattening detected, red = fattening detected, gray: length of algorithm flow vector, darker = shorter. e) and f): Visualization of S_{smooth} and S_{bumpy} normalized by the maximum gradient magnitude of the ground truth (2.5) and colored using the Viridis color scale (g).

bumpier the algorithm result is than it should be.

$$S_{\text{smooth}} := \frac{1}{|\mathcal{M}_s|} \sum_{(x,y) \in \mathcal{M}_s} \max(0, \|J_{\text{gt}}\|_F - \|J_{\text{algo}}\|_F)$$

$$S_{\text{bumpy}} := \frac{1}{|\mathcal{M}_s|} \sum_{(x,y) \in \mathcal{M}_s} \max(0, \|J_{\text{algo}}\|_F - \|J_{\text{gt}}\|_F)$$

3.3 Fine structures

Thin objects which are mapped to fine structures with few pixels diameter are omnipresent in image processing. In autonomous / assisted driving we find bicycles, bollards, chains, traffic sign posts, all kinds of vegetation and fences. In medical imaging there are blood vessels, sutures and instruments. Detecting these is essential in applications like collision avoidance. As of October 2018 out of the top 20 optical flow algorithms in the KITTI leaderboard not a single one reliably detects the chains connecting the bollards and many even struggle with the bollards. Figure 2 d-f shows an example of the phenomenon. The problem is not restricted to the KITTI dataset, out of the top 20 algorithms in the Sintel leaderboard not a single one detects the walking staff of the protagonist running through the forest in test image #5 and 8 algorithms detect only parts or nothing of the weapon in test image #4.

We argue that this is due to three facts: First, the chains only make up about 0.6% of the total image so high errors don't have a strong influence on metrics averaging over large parts of the image. Second, algorithms sensitive to fine structures are likely to over-estimate the size of the fine structure, leading to high errors in an area near the structure which is larger than the structure itself. Third, new algorithms are required to perform at least comparable to the state of the art which is defined by the evaluation leaderboards This means that the current evaluation practice is preventing the development of algorithms which reliably detect fine structures.

We conclude that the development of metrics measuring algorithm performance only at fine structures is a necessary precondition for the development of algorithms which detect them reliably. Therefore we introduce a method to detect fine structures given only a ground truth

flow field. We start with the detected edges we already used in the edge fattening metric. For every pixel which does not belong to an edge we search in opposing directions for edges within a few pixels range in two directions. We repeat this with 4 different angles to find edges independent of orientation. If this first step succeeds we compute a segmentation of a circle around the center pixels as shown in figure 2 a. We accept the pixel if the segment with the center pixel has at least 2 different neighbor segments which are not direct neighbors themselves. We call this set of pixels $\mathcal{M}_{\text{thin}}$ and use it to compute a fine thinning metric F_{thin} as percentage of pixels where the algorithm result is closer to the propagated ground truth than to the ground truth analogue to the edge fattening metrics. We also compute a corresponding fine fattening metric F_{fat} on all pixels in a 8-px radius around $\mathcal{M}_{\text{thin}}$.

4 Experiment

For our experiment we use all image pairs of the Middlebury flow dataset for which public ground truth is available. We run four different flow algorithms, compute a set of metrics and get a score by taking the median of all images. Since these scores have different units and scales, they are not easily comparable, therefore we normalize each metric by the score of the worst algorithm. Figure 3 shows the results. We included commonly used error metrics like BadPix and mean endpoint error (MEE) for comparison. Since D_{edge} , F_{fat} and F_{thin} all restrict evaluation to areas near flow discontinuities we also add variations of the common metrics which only evaluate on pixels with a distance of at least 5px to the nearest flow discontinuity. Finally we measure the 25th percentile of the endpoint errors as a measure of the algorithms precision in the best case.

The resulting rankings show several important aspects of our metrics. First, the pairs $(S_{\text{bumpy}}, S_{\text{smooth}})$ and $(F_{\text{fat}}, S_{\text{smooth}})$ are orthogonal in the sense that algorithms performing well in one metric perform worse in the other. To a slightly lesser extend this also holds for the pairs $(F_{\text{fat}}, F_{\text{thin}})$ and $(D_{\text{edge}}, S_{\text{smooth}})$.

We conclude that our new metrics in practice measure aspects of algorithm performance not covered by existing metrics which average over the whole image.

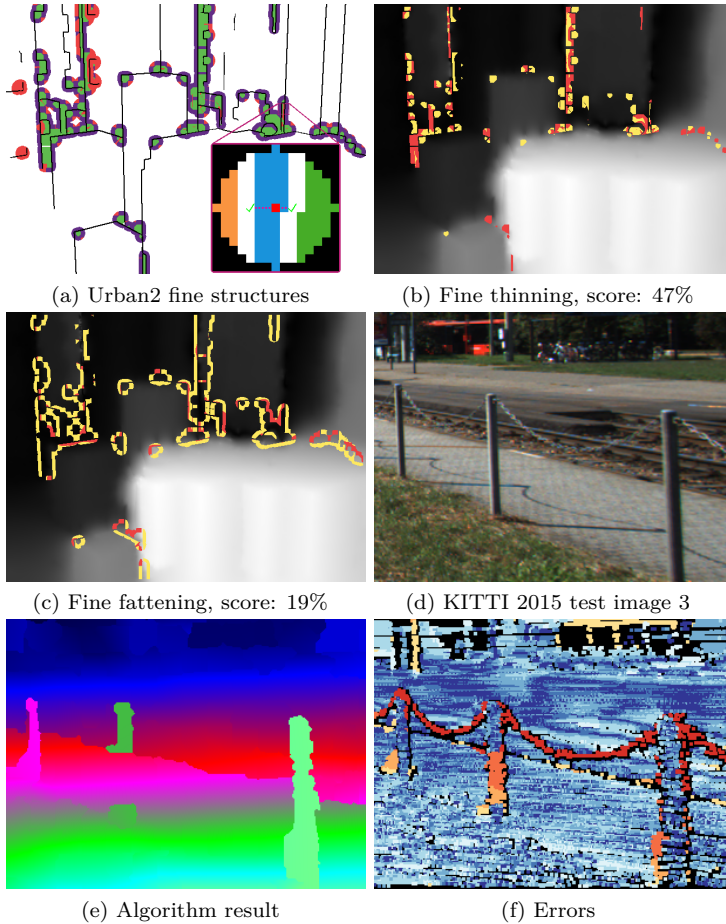


Figure 2: a) Fine structure detection. Green: fine structure, purple: adjacent to a fine structure, red: no evaluation possible, black: flow edge. Zoomed section: visualization of the detection method. b) and c) visualization of fine thinning and fine fattening analogue to sub-figure (d) of figure 1. d) image section of the KITTI 2015 flow dataset [11] e) result of the best published algorithm [15] according to the Middlebury leaderboard f) visualization of the errors in (e). It clearly shows fine thinning, a 3D reconstruction algorithm using this result would miss the chains and a large part of the bollard.

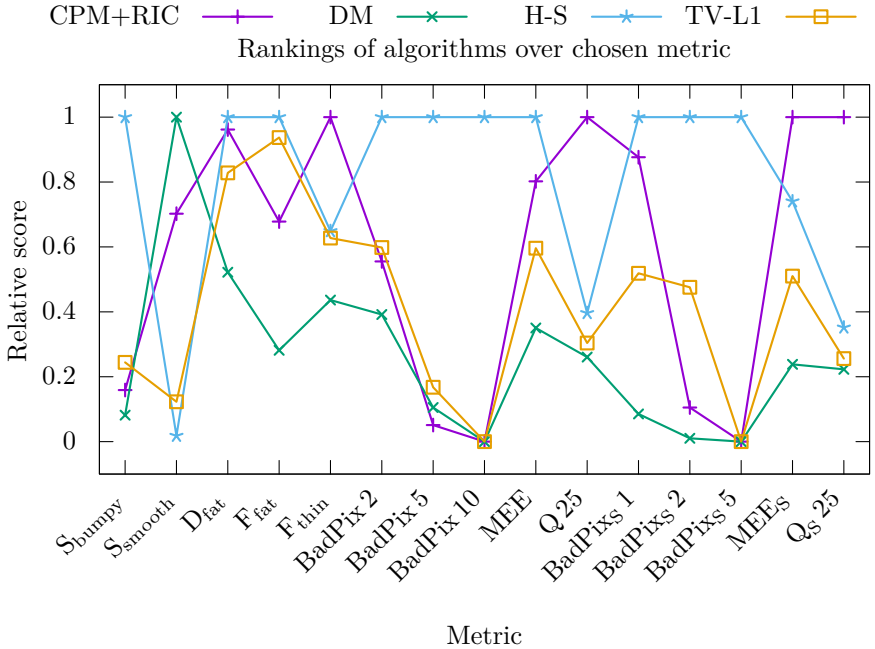


Figure 3: CPM+RIC: [16] using a patch match algorithm [17], DM: DeepFlow2 with deep matching [18], H-S: Horn&Schunck algorithm with a multi-scale strategy [14], TV-L1: [19]. The x-axis shows the median of the metric across all image pairs in the Middlebury dataset for which the ground truth is public. The scores vary wildly between metrics and are therefore normalized by the value of the worst algorithm. The Euclidean distance between algorithm result vector and ground truth is called endpoint error. MEE: Mean endpoint error, BadPix X: percentage of pixels with an endpoint error larger than X, Q25: 25th percentile of the endpoint errors. All metrics with a “S” subscript: Evaluation restricted to the pixels at smooth surfaces $\mathcal{M}_s$. The lines connecting the points are a guide for the eye, their crossings show when rankings change.

5 Conclusion and outlook

We showed that current top algorithms for optical flow consistently fail at fine structures. We argued that the reason is that current evaluation methods weight all pixels of an image uniformly, since fine structures cover only a very small part of most test images, which makes it cheap to ignore them in favour of more accurate results in their adjacent regions.

We extended the concept of geometry-aware performance metrics introduced by Honauer et. al. [13] to the evaluation of dense optical flow algorithms. We developed simple methods to identify challenging regions given only dense ground truth data. Moreover we showed how to compute different kinds of metrics on these regions.

We hope that both the region identification and the corresponding metrics helps researchers and algorithm developers to identify weaknesses in algorithms and adjust the parameters or the training setup for methods using neural networks.

Our future research will focus on the development of methods for automatically identifying correspondences between algorithm parameterization and metrics.

The authors would like to thank Katrin Honauer and Hendrik Schilling for constructive criticism of the manuscript and for inspiring discussions about algorithms and their weaknesses. The authors gratefully acknowledge partial financial support for this research by the Heidelberg Collaboratory for Image Processing (HCI) within the Institutional Strategy ZUK49 “Heidelberg: Realizing the Potential of a Comprehensive University”, Measure 6.4 including matching funds from the industry partners of the HCI.

References

1. B. Jähne, *Digitale Bildverarbeitung*, 7th ed. Springer-Verlag, 2012.
2. K. Palaniappan, C. Kambhamettu, A. F. Hasler, and D. B. Goldgof, “Structure and semi-fluid motion analysis of stereoscopic satellite images for cloud tracking,” in *Computer Vision, 1995. Proceedings., Fifth International Conference on.* IEEE, 1995, pp. 659–665.
3. D. H. Ballard and O. Kimball, “Rigid body motion from depth and optical

- flow," *Computer Vision, Graphics, and Image Processing*, vol. 22, no. 1, pp. 95–115, 1983.
4. C. L. Fennema and W. B. Thompson, "Velocity determination in scenes containing several moving objects," *Computer graphics and image processing*, vol. 9, no. 4, pp. 301–315, 1979.
5. M. Otte and H.-H. Nagel, "Estimation of optical flow based on higher-order spatiotemporal derivatives in interlaced and non-interlaced image sequences," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 5–43, 1995.
6. D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information," *International journal of computer vision*, vol. 5, no. 1, pp. 77–104, 1990.
7. B. McCane, K. Novins, D. Crannitch, and B. Galvin, "On benchmarking optical flow," *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 126–143, 2001.
8. —, "On benchmarking optical flow," <http://of-eval.sourceforge.net/>, pp. 126–143, 2001.
9. S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," in *Proc. of the 11th International Conference of Computer Vision, (ICCV07)*. IEEE, 2007, pp. 1–8.
10. D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Computer Vision, ECCV 2012*, ser. Lecture Notes in Computer Science, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., vol. 7577. Springer Berlin Heidelberg, Oct. 2012, pp. 611–625. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33783-3_44
11. M. Menze, C. Heipke, and A. Geiger, "Joint 3d estimation of vehicles and scene flow," in *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.
12. D. Kondermann, R. Nair, K. Honauer, K. Krispin, J. Andrulis, A. Brock, B. Gusefeld, M. Rahimimoghaddam, S. Hofmann, C. Brenner *et al.*, "The HCI benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 19–28.
13. K. Honauer, L. Maier-Hein, and D. Kondermann, "The HCI stereo metrics: Geometry-aware performance analysis of stereo algorithms," in *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015.
14. E. Meinhardt-Llopis, J. S. Pérez, and D. Kondermann, "Horn-schunck optical flow with a multi-scale strategy," *Image Processing On Line*, vol. 3, pp. 151–172, 2013. [Online]. Available: <http://dx.doi.org/10.5201/ipol.2013.20>

15. A. Behl, O. H. Jafari, S. K. Mustikovela, H. A. Alhaija, C. Rother, and A. Geiger, "Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios?" in *International Conference on Computer Vision (ICCV)*, 2017.
16. Y. Hu, Y. Li, and R. Song, "Robust interpolation of correspondences for large displacement optical flow," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.
17. Y. Li, Y. Hu, R. Song, P. Rao, and Y. Wang, "Coarse-to-fine patchmatch for dense correspondence," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, pp. 1–1, 2017.
18. P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, Dec. 2013. [Online]. Available: <http://hal.inria.fr/hal-00873592>
19. J. Sánchez Pérez, E. Meinhardt-Llopis, and G. Facciolo, "TV-L1 Optical Flow Estimation," *Image Processing On Line*, vol. 3, pp. 137–150, 2013.

Local histogram equalization and homomorphic filtering to improve wide-angle stereo disparity computation

Sebastian Metzner, Rahul Ramesh, and Tino Hausotte

Friedrich-Alexander-University Erlangen-Nürnberg (FAU), Institute of Manufacturing Metrology, Naegelsbachstr. 25, 91052 Erlangen

Abstract Dimensional measurements of reflective surfaces are often challenging for optical measurement systems. The use of structured light causes overexposure of the reflective regions, resulting in an incomplete measurement. Stereo vision is an effective alternate to optical measurements. Image rectification and disparity map computation steps are used to generate a 3D point cloud which can be used for dimensional measurements. When the stereo vision setup requires cameras at wide angles to the surface normal, the image pairs are affected by varying illumination, once again resulting in bad disparity. In this paper different image processing possibilities are examined at the example of a sheet-bulk metal formed workpiece to obtain an improved disparity map of the overexposed areas of the data set. A newly developed local histogram equalization method improves the disparity map for stereo vision calculations with wide-angle camera setups.

Keywords Wide-angle stereo vision, image processing, disparity computation.

1 Introduction

Sheet-bulk-metal forming is a new production technology. It combines sheet metal forming with solid forming and leads to workpieces with varying size and shortened process chains through the combination of process steps [1]. This process creates different forming zones based on the different reflective properties of the workpiece. Figure 1(a) shows

an exemplary workpiece with various forming zones. The point cloud obtained using the fringe projection system GOM ATOS Compact Scan 2M, shows the missing data points at directional reflection areas of the workpiece (Figure 1(b)). These missing points are caused by the directional reflection on the surface and the geometrical placement of the camera and the projector [2].

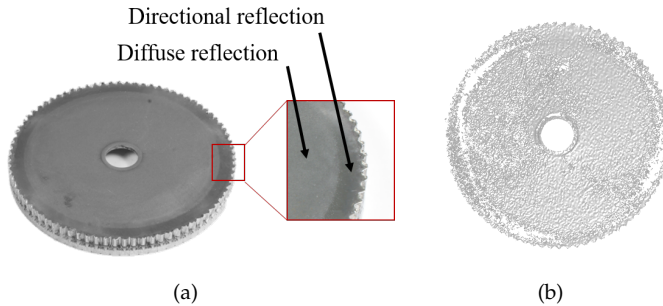


Figure 1: (a) Workpiece with directional and diffuse reflection, (b) Point cloud captured with the fringe projection system with missing points in the directional reflection area.

The results obtained using a stereo vision setup is comparable to the results of the fringe projection system. In order to obtain an improved point-cloud of the object using stereo-vision setup, a standard pipeline is used. This involves stereo camera calibration, image acquisition, rectification and disparity map calculation. The disparity map along with the intrinsic and extrinsic camera parameters are used for the point cloud reconstruction. In a standard stereo vision setup, cameras with a similar angle to the vertical axis of 25 degrees are applied. This angle also corresponds to the angle of the fringe projection system. A disadvantage of large angles and long distance of the cameras is the often imprecise calibration of the exact relative orientation and position of the cameras. In addition, the identical areas are exposed differently due to the different viewing angles which makes stereo matching more difficult [3]. Several image processing steps are used to overcome the challenge posed by the reflective surfaces with varying illumination under wide-angle setup.

2 Related works

Semi global block matching (SGBM) is a standard stereo matching algorithm used for disparity map computation [4]. The use of histogram equalization prior to disparity computation improves the disparity map. Correal et al. mention the use of histogram matching and homomorphic filtering for intensity correction [5]. Further, Aditya et al. suggest the use of gaussian and median filtering for the disparity improvements [6]. In these previous works, the entire image is considered for the intensity corrections between the stereo image pair. Therefore the size of blocks used in SGBM is used to correct the illumination differences locally.

3 Disparity map computation using stereo vision

3.1 Measurement setup

For a flexible measurement setup, the placement of the fringe projection system is reconstructed. Figure 2 shows the measurement setup with two cameras, which are mounted at an angle of 25 degrees to the vertical axis identical to the GOM System.

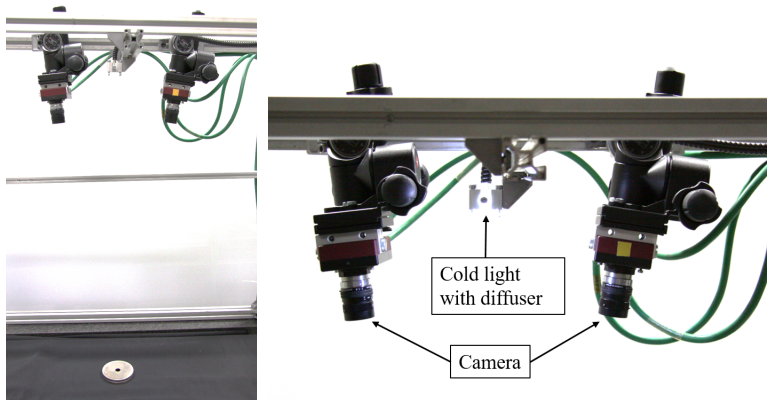


Figure 2: Measurement setup with two cameras and a cold light source with diffuser.

In addition, a cold light source positioned in the middle of the two cameras with a diffuser is implemented to generate defined light conditions. Any additional external light source was switched off during image capturing. The distance to the measuring object is 570 mm, which is identical to the ideal distance for fringe projection measurement. The Prosilica GS2450 cameras are Sony ICX625 CCD cameras with a resolution of 5 MP. The cameras are equipped with Ricoh Pentax FL-BC2514D-V objectives with a focal length of 25 mm. The optimal exposure time is selected and optimized as described in the next chapter.

3.2 Initial disparity map computation

The methodology employed involves three steps. First, the optimum light setting and exposure value for the setup is calculated through exposure analysis. Second, for the selected exposure value, the binary mask of reflective regions in the workpiece is computed. Third, an initial estimate of disparities is constructed.

To get repeatable results, the light setting and the exposure time have to be adapted and optimized to the measurement setup. The position of the workpiece should produce overexposed areas on the workpiece, so that the contra lateral sides in the stereo image pair are reflective. A light source with diffuser is used to maintain a constant inhomogeneous light condition. For exposure analysis, stereo images are captured with increasing exposure time ranging from 5000 ms to 50000 ms in steps of 5000 ms. The number of pixels that reach saturation (pixel intensity > 200 grey value) in the left image is plotted against each exposure value in Figure 3(a). Further, the number of valid disparity points from the disparity maps corresponding to each image pair is calculated using Equation 1. The threshold values used here are obtained by a trial and error method and may vary with respect to the set up.

$$N_{valid} = \sum (I_{DisaprityMap}(x, y) > 1480 \quad \& \quad I_{LeftImage}(x, y) > 30) \quad (1)$$

with $I_{DisaprityMap}(x, y)$ = disparity values for each pixel and $I_{LeftImage}(x, y)$ = grey values of the reference (left) image. The plot of number of valid disparity points versus the exposure time is shown in Figure 3(b). A comparison of the plots in Figure 3 shows that the number of valid of disparities decreases with an increase in the number of

pixels reaching saturation. An exposure time of 10000 ms is the optimal exposure time in this setup as it results in a high number of valid disparities.

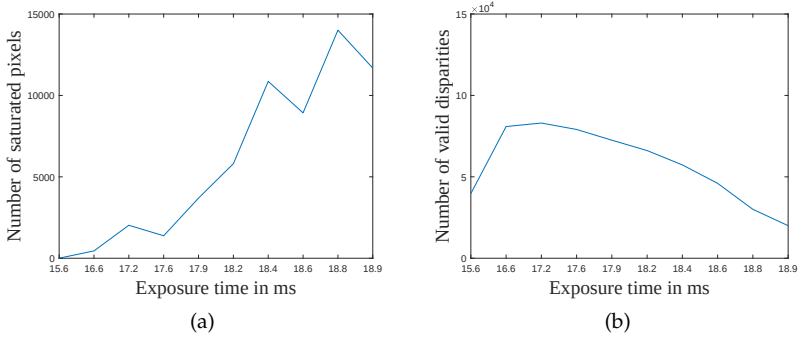


Figure 3: (a) Exposure values versus saturated pixels; (b) Exposure values versus number of valid disparities.

The next step after calculating the optimal exposure time is to generate a binary map outlining the reflective regions in the left-right image pair obtained. An increasing gain analysis using standard deviations and area under the gain curve is used. The procedure is described in [7] and is used to obtain the mask regions corresponding to the left and the right images. The binary points in the right mask are shifted to the left by a previously computed disparity value of 1470. Thus, a final mask region that outlines the reflective regions is obtained as seen in Figure 4(a).

The computation of the initial disparity map corresponding to the selected image pair is achieved using the standard steps. This includes rectification, a global histogram equalization followed by the disparity calculation. The global histogram equalisation (GHE) will re-assign the intensity values of pixels to make the intensity distribution uniform [8]. Therefore, the grayscale transformation T was minimized. The obtained disparity map is shown in Figure 4(b). It can be observed from the Figure 4 that the areas missing in the disparity map corresponds approximately to the reflecting region outline (left).

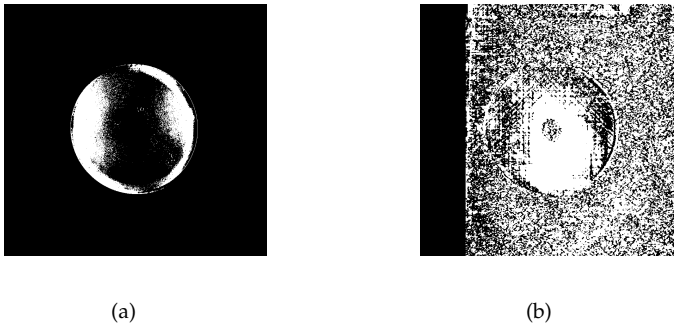


Figure 4: (a) Mask of reflective regions; (b) Initial Disparity Map.

4 Improvements to disparity computation

To improve the results in wide-angle disparity computation, common image processing techniques such as histogram equalization, gaussian and median filter or the homomorphic filter are used. This section compares the previously used filters to an adapted method of local histogram equalization.

4.1 Gaussian filter and median filter

The method described in [6] uses a gaussian filter followed by a median filter on global histogram equalised and rectified images. The disparity map is computed using MATLAB with the standard settings. The missing points in initial disparity map (Figure 4(b)) corresponding to the white regions in ROI (Figure 4(a)) are replaced by the disparity points obtained using gaussian and median filtering. Figure 5 shows the new disparity map from the gaussian and median filtered images. The result is an approximately seven percent lower number of valid disparity values in comparison to initial disparity map.

4.2 Homomorphic filter

Another possibility to improve the disparity map is to compute homomorphic filtering on global histogram equalized images. Homomorphic

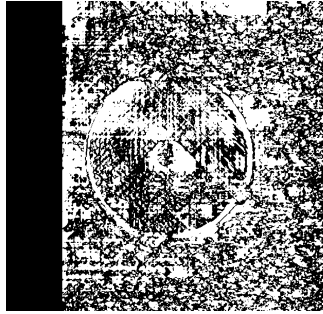


Figure 5: Disparity map using gaussian and median filtered images.

filtering uses illumination-reflectance models to correct non-uniform illumination in the images. This model considers the image to be characterized by the reflectance component of the objects on the scene $r(x, y)$ and the amount of source illumination incident $i(x, y)$ [9]. Equation 2 describes the image with the reflectance and illumination component. The aim is to remove the illumination component and force an uniform intensity.

$$f(x, y) = i(x, y)r(x, y) \quad (2)$$

The filter is implemented in several steps using natural logarithm, Fourier transformation, high pass filtering and inverse Fourier transformation [10]. The improved disparity map is calculated similar to the disparity map described in section 4.1. Figure 6 shows the improved disparity map.

4.3 Local histogram equalization

Local histogram matching is a simple and straightforward method proposed in this work. The goal of intensity correction is to minimize the difference between the intensity values of corresponding pairs of pixels in stereo image pair. Advantages result from an local implementation

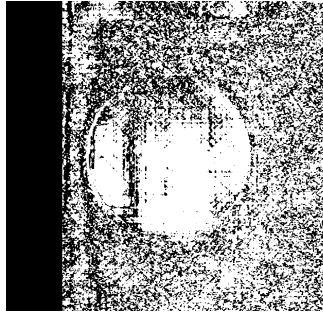
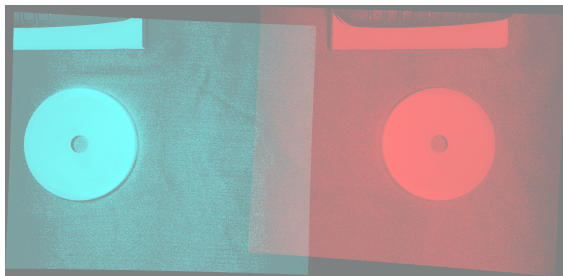


Figure 6: Disparity map improvement using homomorphic filtered images.

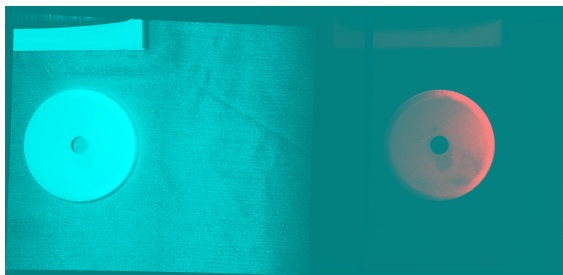
to a small window. The block size used for disparity map computation is the same as the window size. The image from the right camera is the reference image. The reference image is global histogram equalized. For every window $W(x, y)$ in the reference image, the corresponding window in the other image lies in a range $W(x, y(d_{min} : d_{max}))$, where d_{min} is the minimum disparity value and d_{max} is the maximum disparity value. This range is defined as R . Since the images are rectified, the matching is limited to one direction, similar to the case in SGBM. Hence for every $W(x, y)$ in the reference image, R different windows in the left image have to be matched to the histogram of $W(x, y)$. This results in R different local histogram equalized images. Afterwards the disparity maps corresponding to each reference-local histogram equalized image pair are computed. The image with the most number of valid disparity points is selected. Similar to previous sections, the improved disparity maps using ROI are computed. The final selected local equalized image pair is shown in Figure 7(c).



(a)



(b)



(c)

Figure 7: (a) Input image pairs, (b) Global histogram equalized pair, (c) Local histogram equalized pair.

Figure 8 shows the improved disparity map. With local histogram equalization, the number of valid points increased from 229121 to 261942 (14 percent). Compared to Figure 6 it can be seen that the structures on the surface like the hole in the middle of the workpiece are still visible. In addition, the overexposed areas at the edge of the workpiece are partially filled with valid disparity values.

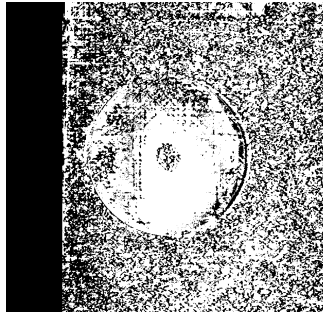


Figure 8: Disparity map using local histogram equalized images.

5 Conclusion

Disparity map computation from wide angle camera images of reflective objects suffer from local illumination variations. Histogram equalization is a standard preprocessing step applied on the rectified images prior to the disparity computation. The histogram of the right camera image is matched to the histogram profile of the left image. This is a global process that is applied over the whole image and fails to correct the local intensity variations. As a corrective measure for the illumination variations, popular filters such as homomorphic filters and gaussian-median filters have been employed previously. A method of local histogram matching is proposed that will reduce the intensity differences of corresponding pixels in the left and right images. Table 1 lists the valid disparity values of the individual disparity maps using different filters and histogram equalizations.

Table 1: Number of valid disparity points using the described filters and histogram equalisations.

Filter/Histogram equalisation	Number of valid disparity points
Global Histogram Equalisation	229112
Homomorphic filtering	234943
Local histogram Equalisation	261942
Median and gaussiang filtering	212934

The maximum number of valid disparity values is reached with the developed local histogram equalization. The study conducted shows that the proposed local histogram equalization method improves the disparity map by a factor that is better than global histogram equalization or its combination with the aforementioned filters.

References

1. M. Merklein, J. Allwood, B.-A. Behrens, A. Brosius, H. Hagenah, K. Kuzman, K. Mori, A. Tekkaya, and A. Weckenmann, "Bulk forming of sheet metal," *CIRP Annals*, vol. 61, no. 2, pp. 725–745, 2012.
2. S. Metzner, T. Hausotte, and A. Loderer, "Ganzheitliche dimensionelle mesung von blechmassivumgeformten bauteilen," *tm - Technisches Messen*, vol. 84, no. 9, jan 2017.
3. C. Olson, H. Abi-Rached, M. Ye, and J. Hendrich, "Wide-baseline stereo vision for mars rovers," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*. IEEE, 2003.
4. H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, feb 2008.
5. R. Correal, G. Pajares, and J. Ruz, "Automatic expert system for 3d terrain reconstruction based on stereo vision and histogram matching," *Expert Systems with Applications*, vol. 41, no. 4, pp. 2043–2051, mar 2014.
6. K. Aditya, V. K. Reddy, and H. Ramasangu, "Enhancement technique for improving the reliability of disparity map under low light condition," *Procedia Technology*, vol. 14, pp. 236–243, 2014.
7. S. Metzner, R. Ramesh, A. Müller, and T. Hausotte, "Evaluation of reflection properties of sheet bulk metal formed parts by pixel wise analysis of camera

images provided for triangulation measurement," in *Proceedings of the 19th ITG/GMA-Fachtagung Sensoren und Messsysteme*, 2018.

8. H. Cheng and X. Shi, "A simple and effective histogram equalization approach to image enhancement," *Digital Signal Processing*, vol. 14, no. 2, pp. 158–170, mar 2004.
9. R. E. W. Rafael C. Gonzalez, *Digital Image Processing*. Pearson Education (US), 2017. [Online]. Available: https://www.ebook.de/de/product/28253161/rafael_c_gonzalez_richard_e_woods_digital_image_processing.html
10. S. A. M. Saleh, "Mathematical equations for homomorphic filtering in frequency domain: A literature survey," 01 2012.

Embedded Stereokamera-basierte reaktive Kollisionsvermeidung für UAVs

Embedded stereo vision based reactive collision avoidance for UAVs

Michael Grinberg¹, Boitumelo Ruf^{1,2}, Sebastian Monka¹, Matthias Kollman¹, Aleksej Buller¹ und Igor Tchouchenkov¹

¹ Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung IOSB,
Fraunhoferstr. 1, Karlsruhe

² Institut für Photogrammetrie und Fernerkundung, KIT,
Englerstr. 7, Karlsruhe

Zusammenfassung Dieser Beitrag beschäftigt sich mit der automatischen Kollisionsvermeidung für UAVs basierend auf einer Stereokamera. Dazu haben wir ein echtzeitfähiges und energieeffizientes System realisiert, das mit beschränkten Ressourcen zu-rechtkommt. Zur Hinderniserkennung werden aus den Bildern der Stereokamera Disparitätskarten berechnet. Diese werden in die sog. U- bzw. V-Maps konvertiert, mit deren Hilfe mit Methoden der Mustererkennung Objekte erkannt werden. Die Kollisionsvermeidung basiert auf einem reaktiven Ansatz, welcher den kürzesten Weg zum Umfliegen eines sich in kritischen Entfernung befindlichen Hindernisses sucht.

Wir haben unser System auf einer Xilinx Zynq Ultrascale+ Prozessoreinheit mit einer ARM Cortex-A53 Quadcore-CPU und einem FinFET+ FPGA umgesetzt. Die Disparitätsschätzung wurde auf dem FPGA implementiert, während die weniger rechenintensive Kollisionsvermeidung zur Ausführung auf der CPU umgesetzt wurde. Die Portierung der in C/C++ geschriebenen Algorithmen zur Disparitätsschätzung auf den FPGA erfolgte mittels High-Level-Synthese, welche die Lücke zwischen Anwendungs-entwicklung und Hardware-Optimierung schließt und damit eine Reduktion der Entwicklungszeit und -Kosten ermöglicht.

Schlagwörter UAV-Navigation, Hindernisdetektion, Kollisionsvermeidung, embedded, Echtzeit-Stereoverarbeitung.

Abstract This contribution deals with obstacle avoidance for UAVs based on a stereo camera setup. We implemented a real-time-capable and energy-efficient system based on disparity map estimation for obstacle detection and a reactive approach for collision avoidance. The basic functionality has been written in C/C++ and optimized for Xilinx Zynq Ultrascale+ with an ARM Cortex-A53 Quad-Core CPU and a FinFET+ FPGA. We aimed to use High-Level Synthesis for porting parts of our code to FPGA in order to close the gap between application development and hardware optimization and reduce development time and costs. We evaluated our implementation of the disparity estimation on the KITTI Stereo 2015 benchmark. The integrity of the overall real-time collision avoidance system has been evaluated by using Hardware-in-the-Loop testing in conjunction with two flight simulators.

Keywords UAV navigation, obstacle detection, collision avoidance, embedded, real-time stereo processing.

1 Einleitung

Um die Benutzerfreundlichkeit und Sicherheit von unbemannten Luftfahrzeugen (UAVs) zu verbessern, können UAVs mit Sensoren, wie z.B. Ultraschallsensoren, Laserscannern und Kameras, ausgestattet werden. Eine wichtige Aufgabe dabei ist die Detektion von Hindernissen und die Vermeidung von Kollisionen. Da ein Großteil der Verbraucher-UAVs unter engen Gewichts- und Leistungseinschränkungen leidet, konzentriert sich unsere Arbeit auf die Hindernisdetektion und Kollisionsvermeidung auf der Grundlage eines leichten Stereokamera-Setups. Dabei sind die Kameras in Flugrichtung ausgerichtet. Die aus den Kamerabildern berechnete Disparitätskarten verwenden wir um Hindernisse im Flugpfad des UAVs zu detektieren und Ausweichmanöver einzuleiten. Dabei erfolgt die gesamte Verarbeitung, welche die Berechnung der Disparitätskarten und Planung des Ausweichmanövers einschließt, an Bord und in Echtzeit.

Die grundsätzliche Eignung von Stereo-Bildverarbeitung für die On-Board Kollisionsvermeidung für UAVs wurde durch zahlreiche Studien gezeigt [1,2]. Für die Berechnung von Disparitäts- bzw. Tiefenkarten auf embedded Systemen hat sich das SGM-Verfahren [3] etabliert. Gründe

dafür liegen nicht nur in dessen Leistungsfähigkeit, sondern insbesondere auch darin, dass sich dieses Verfahren gut für die Parallelisierung auf verschiedenen Rechnerarchitekturen eignet. Zudem zeichnet sich das SGM-Verfahren durch ein gutes Trade-Off zwischen Genauigkeit und Rechenzeit aus. Durch eine Optimierung des Verfahrens für eine CPU wurden für die VGA-Bildauflösung jeweils Verarbeitungsraten von 12 fps [4] und 14 fps [5] erreicht. Durch die Verwendung von General Purpose Computation on GPUs (GPGPU) können deutlich größere Verarbeitungsraten bei höheren Bildauflösungen erreicht werden [6,7].

Im Hinblick auf ihre Energieeffizienz sind FPGAs für ein embedded System jedoch deutlich besser geeignet. So wurde in [1] ein Stereo-Video-Sensor vorgestellt, bei dem eine FPGA-basierte SGM-Implementierung für die Bildauflösung von 320×240 Pixel eine Framerate von 120 fps lieferte. In [8] wurde eine Portierung des SGM-Verfahrens auf ein Xilinx Zynq 7000 FPGA beschrieben. Dabei wurde eine Framerate von 32 fps für eine Bildauflösung von 1280×720 Pixel erreicht. Wie in [9] gezeigt, können je nach FPGA für diese Bildauflösung gar Frameraten von bis zu 197 fps erreicht werden. Um solch hohe Frameraten erzielen zu können, ist allerdings eine ausgeklügelte Optimierung notwendig, die typischerweise Verwendung von Hardware-spezifischen Schnittstellen wie VHDL oder Verilog erfordert.

Eine solche Optimierung erfordert spezielles Wissen sowohl aus dem Bereich der Bildverarbeitung als auch aus dem Bereich der Hardware-Optimierung. Zudem ist sie sehr kostspielig und verlängert die Time-To-Market von Anwendungen erheblich. Daher haben wir uns im Rahmen des TULIPP-Projektes³ [10] darauf beschränkt, die Bildverarbeitungsverfahren, die in C/C++ entwickelt wurden, alleine mittels High-Level-Synthese (HLS) auf ein FPGA zu portieren. Dabei muss naturgemäß mit einigen Leistungseinbußen gerechnet werden. Wie jedoch in diesem Beitrag gezeigt, reicht die erzielte Genauigkeit und Leistung für den anvisierten Anwendungsfall aus.

Der Rest dieses Artikels gliedert sich wie folgt: Kapitel 2 enthält eine detaillierte Übersicht über den verfolgten Ansatz und dessen Implementierung. Experimentelle Ergebnisse werden im Kapitel 3 vorgestellt. Kapitel 6 gibt eine kurze Zusammenfassung und einen Ausblick.

³ <http://tulipp.eu/>

2 Systemrealisierung

2.1 Echtzeit-Disparitätsschätzung

Die Disparitätsschätzung besteht aus den folgenden Schritten:

- Rektifizierung
- Pixel-Matching
- SGM-Optimierung (Regularisierung)
- Links-Rechts-Konsistenzcheck
- Median-Filter

Diese Schritte werden in einer Pipeline angeordnet um eine effiziente Verarbeitung der Pixelströme der Kameras zu ermöglichen. Für ein speichereffizientes Verarbeiten der Daten verwenden wir FIFO-Puffer zwischen den einzelnen Pipelineinstufen. Wie bereits erwähnt, wurden alle Verfahren in C/C++ implementiert, woraus anschließend mittels HLS entsprechender VHDL-Code für die Ausführung auf dem FPGA erzeugt wurde.

Rektifizierung

Für die Disparitätsschätzung müssen die Bildpaare der Stereokamera zunächst rektifiziert werden. Wir verwenden die Standard-Kalibrieremethode nach [11] um die Rektifizierungs-Lookup-Tabellen für die beiden Bilder zu berechnen. Diese enthalten für jeden Pixel im rektifizierten Bild die Koordinaten des Pixels im Originalbild. Für die Reduktion der Zugriffszeit haben wir zur Berechnung des rektifizierten Bildes einen Zeilenpuffer für die Zwischenspeicherung der benötigten Zeilen des Eingang-Pixelstroms realisiert.

Pixel-Matching

Im nächsten Schritt wird für jeden Pixel in einem der Bilder ein Ähnlichkeitsmaß mit den in Frage kommenden Pixeln des anderen Bildes berechnet. Dadurch entsteht für ein Bild der Größe $W \times H$ ein dreidimensionales Kostenvolumen $\mathcal{C} \in \mathbb{N}_0^{W \times H \times d_{\max}}$. Dabei gibt $d_{\max}$ die maximal mögliche Disparität an. Für jeden Pixel p und jede Disparität d enthält dieses Kostenvolumen das entsprechende Ähnlichkeitsmaß.

In der Literatur wurden mehrere Kostenfunktionen zur Ermittlung der Ähnlichkeitsmaßes zweier Bildpixel vorgeschlagen. Wir haben eine einfache Summe der pixelweise absoluten Differenzen sowie die nicht-parametrische Hamming-Distanz der Census Transformation [12] ausgewählt, da diese in vielen Anwendungen recht beliebt sind. Auch hier werden Zeilenpuffer verwendet, um auf die Pixeldaten aus der unmittelbaren Umgebung des betrachteten Pixels schnell zugreifen zu können.

SGM-Optimierung (Regularisierung)

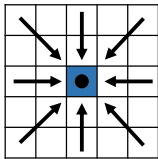
Das vorberechnete Kostenvolumen $\mathcal{C}$ enthält für jeden Pixel p des linken Bildes und für jede mögliche Disparität entsprechende Plausibilitätswerte. Dies könnte bereits zur Erstellung einer Disparitätskarte genutzt werden, indem man den Winner-Takes-it-All-Ansatz verwendet, bei dem für jeden Pixel die Disparität mit den geringsten Kosten ausgewählt wird. Allerdings liefert dieser Ansatz aufgrund von Mehrdeutigkeiten in der Praxis keine zufriedenstellenden Ergebnisse. Daher wird zur Regularisierung des Kostenvolumens eine Optimierungsstrategie angewandt, die die meisten Mehrdeutigkeiten eliminieren soll. Im Falle des SGM-Verfahrens werden für jeden Pixel die Kosten benachbarten Pixel entlang mehrerer konzentrischen Pfade aggregiert (s. Abbildung 1). Dabei werden die Kosten für die benachbarten Pixel q eines Pixels p entlang eines Pfades jeweils mit dem kleinen Strafterm P_1 bestraft, wenn die Differenz der Disparitäten beider Pixel p und $q \pm 1$ beträgt, und mit einem größeren Strafterm P_2 , wenn der Betrag der Differenz größer 1 ist. Aus dem aggregierten Kostenvolumen $\mathcal{C}_{\text{aggr}}$ können dann die endgültigen Disparitäten durch die Anwendung des Winner-Takes-it-All-Ansatzes extrahiert werden.

Die Aggregation entlang jedes Pfades stellt eine unabhängige Aufgabe dar, was eine starke Parallelisierung des Verfahrens für diverse Rechnerarchitekturen erlaubt. Wir haben den SGM-Algorithmus für eine Echtzeit-Anwendung auf dem FPGA optimiert. Dabei wurde die in TULIPP entwickelte Hardware-Instanziierung basierend auf einem embedded Xilinx Zynq Ultrascale+ MPSoC zugrunde gelegt.

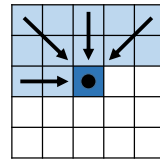
In der ursprünglichen SGM-Formulierung [3] wird die Aggregation entlang von 16 konzentrischen Pfaden durchgeführt. Um eine Berechnung der Zwischenwerte für subpixelgenaue Positionen zu vermeiden,

verwenden die meisten Implementierungen jedoch lediglich 8 statt 16 Pfade. Aufgrund der Speicherbeschränkungen von embedded FPGAs, verbunden mit deren Stärken bei gleichzeitiger Verarbeitung mehrerer Datenströme, ist es üblich, bei Portierung des Algorithmus auf FPGAs lediglich 4 Pfade zu verwenden. Wie in [13] gezeigt, führt eine Reduktion von 8 auf 4 Pfade zu einem relativ geringen Genauigkeitsverlust von 1.7% bei gleichzeitiger Reduktion der Laufzeit.

Die Aggregation kann effizient durch eine Zwischenpufferung und Akkumulation der vorberechneten Kosten für jeden Pixel durchgeführt werden. Wie in [3] vorgeschlagen, werden nach Prozessierung jedes Pixels bei der Kostenaggregation die bisherigen Mindestkosten des Pfades subtrahiert um einen numerischen Überlauf zu vermeiden.



(a)



(b)

Abbildung 1: Kostenaggregation (4 vs. 8 Pfade). Während die Aggregation entlang von 8 Pfaden (a) den Zugriff auf das komplette Bild für die Kostenberechnung einzelner Pixel erfordert, ist bei der Verwendung von 4 Pfaden (b) eine effiziente Implementierung durch Akkumulation der bereits berechneten Kosten des Pixelstroms (hellblau) möglich.

Links-Rechts-Konsistenzcheck

Zur Eliminierung von Ausreißern wird ein Links-Rechts-Konsistenzcheck durchgeführt. Dazu muss neben der Disparitätskarte für das linke Bild eine Disparitätskarte für das rechte Bild berechnet werden. Bei korrekter Schätzung sollen die für die Pixeln im linken Bild geschätzten Disparitäten mit den für das rechte Bild geschätzten Disparitäten an den jeweils entsprechend versetzten Pixelpositionen übereinstimmen. Ist dies nicht der Fall, wird die geschätzte Disparität als ungültig markiert.

Median-Filter

Um weitere Ausreißer zu eliminieren, wird abschließend eine Median-Filterung durchgeführt. Ähnlich zur Berechnung der Kostenfunktion, sind für die FPGA-Implementierung eines $k \times k$ -Median-Filters k Zeilenpuffern notwendig.

2.2 Hindernisdetektion und Kollisionsvermeidung

Unser Ansatz zur UAV-Navigation basiert auf einer reaktiven Kollisionsvermeidung. Dabei werden die wie beschrieben errechneten Disparitätskarten verwendet, um die Umgebung vor dem UAV zu erfassen. Diese werden zunächst jeweils in ein horizontales und vertikales Disparitätshistogramm konvertiert (U- bzw. V-Map), wodurch die Komplexität deutlich verringert und die Robustheit gegenüber Messungenauigkeiten verbessert wird [2, 14, 15]. Bei einer U-Map wird jede Spalte der Disparitätskarte durch das zweidimensionale Histogramm der Größe $W \times d_{\max}$ repräsentiert. Entsprechend wird bei einer V-Map jede Zeile durch ein Histogramm der Größe $d_{\max} \times H$ repräsentiert. Mittels einer anschließenden Dilatation, sowie einer Gauß-Filterung, werden Messungenauigkeiten und Rauschen unterdrückt. Im nächsten Schritt werden beide Maps mittels Schwellwertbildung in ein Binärbild umgewandelt. Das Ergebnis der Berechnung sowie seine Interpretation sieht man in der Abbildung 2. Die V-Map wird dazu verwendet die Lage der Bodenebene und die Höhe der Objekte in einer bestimmten Entfernung zu schätzen. Die U-Map kann als eine Draufsicht auf die Szene interpretiert werden. Dabei können größere Objekte in der Umgebung anhand dominanteren Strukturen in der U-Map ermittelt werden.

Zur Navigation werden potentielle Hindernisse gemäß dem Ansatz in [16] anhand ihrer Konturen in der U-Map erkannt. Anschließend werden deren Schwerpunkte berechnet und sie in Ellipsen eingeschlossen. In der V-Map kann für die gegebene Disparität die vertikale Ausdehnung der Objekte berechnet werden. Dadurch werden Objekte durch ein zylindrisches Modell repräsentiert. Wird ein Objekt in einem kritischen Abstand zum UAV entdeckt, wird der kürzeste Weg nach links, rechts, oben oder unten gefunden, sodass das Hindernis umflogen werden kann. Durch eine periodische Ausführung des Algorithmus können falsche Entscheidungen erkannt und revidiert werden.

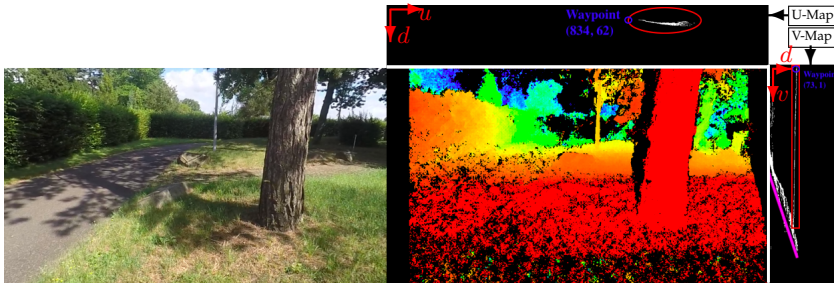


Abbildung 2: U-/V-Map-basierte Hinderniserkennung. Das linke Bild ist das Referenzbild der linken Kamera. Das rechte Bild zeigt die farbig kodierte Disparitätskarte. Dabei steht Blau für eine geringe Disparität (große Bildtiefe bzw. Entfernung zur Kamera) und Rot für eine große Disparität (kleine Bildtiefe bzw. Entfernung). Über der Disparitätskarte, sowie rechts daneben, befinden sich jeweils die U- und die V-Map. Die weiße Kontur in der U-Map entspricht der sichtbaren Seite des Baums in der Draufsicht, die interne Repräsentation des Baums geschieht durch die rot gefärbte Ellipse. In der V-Map ist der Baum durch das rot gefärbte Rechteck repräsentiert. Die pinkfarbene Linie dient zur Hervorhebung der Bodenebene (diagonale weiße Kontur). Blaue Kreise zeigen den Wegpunkt, der genutzt wird um dem Hindernis auszuweichen.

3 Evaluation

Wir haben unser System zur Kollisionsvermeidung auf einem embedded Xilinx Zynq Ultrascale+ MPSoC mit einer ARM Cortex-A53 Quadcore-CPU und einem FinFET+ FPGA implementiert. Während die Disparitätskarten auf dem FPGA berechnet werden, wird der Algorithmus zur Kollisionsvermeidung auf der CPU ausgeführt. Der Ultrascale+ wird als System-on-a-Module (SOM) auf dem EMC²-DP Trägerboard von Sundance betrieben. Für den Bildeinzug wurden zwei Industriekameras von Sentech verwendet, die über die CameraLink-Schnittstelle mit der Trägerplatine verbunden sind.

3.1 Evaluation der Disparitätsschätzung

Sowohl die Rektifizierung der Eingangsbilder als auch die Berechnung der Disparitätskarte wurde in C/C++ implementiert. Dabei haben wir unseren Algorithmus wie folgt konfiguriert: Bei Berechnung der Sum-

me der absoluten Differenzen (SAD), der Census-Transformation (CT) sowie der Median-Filterung wird eine Nachbarschaft der Größe 5×5 verwendet. Die maximale Disparität wird auf 59 Pixel gesetzt. Die SGM-Strafterme wurden auf $P_1 = 200$ und $P_2 = 800$ für die SAD-Kostenfunktion bzw. auf $P_1 = 8$ und $P_2 = 32$ bei Verwendung der Hamming-Distanz der Census-Transformation gesetzt.

Zur automatischen Generierung des VHDL-Codes aus dem C/C++-Code für die Prozessierung auf dem FPGA haben wir Xilinx Vivado High-Level Synthesis verwendet. Die damit erzielten Ergebnisse werden in den nachfolgenden Tabellen dargestellt. Tabelle 1 zeigt den Ressourcenverbrauch des Algorithmus zur Berechnung der Disparitätskarte auf dem Ultrascale+, dessen FPGA 154.000 Logikzellen hat. Tabelle 2 zeigt die Laufzeitmessungen der einzelnen Systemkomponenten und des Gesamtsystems bei einer FPGA-Taktung von 200 MHz.

Tabelle 1: Ressourcenverbrauch des Algorithmus zur Disparitätsschätzung auf dem Zynq Ultrascale+.

Ressource	Insgesamt verfügbar	Verwendet	Ausnutzung
DSP48E	360	22	6 %
BRAM_18K	432	132	31 %
FF	141120	12561	9 %
LUT	70560	27063	38 %

Tabelle 2: Laufzeitmessungen des Systems auf dem Xilinx Zynq Ultrascale+ bei einer FPGA-Taktung von 200 MHz.

Verfahren	Laufzeit (s)	Mittlere Laufzeit (s)
Schätzung der Disparitätskarte	0,0314 - 0,0361	0,0345
Kollisionsvermeidung	0,0027 - 0,0110	0,0042
Σ	0,0341 - 0,0471	0,0387

Die komplette Pipeline zur Disparitätsschätzung erreicht somit eine mittlere Verarbeitungsgeschwindigkeit von 29 fps bei einer Bildgröße von 640×360 Pixeln. Die Latenz der Verarbeitung liegt bei 28,5 ms. Eine Reduktion der Taktrate des FPGAs auf 100 MHz reduziert die Verarbeitungsgeschwindigkeit auf ca. 20 fps und erhöht die Latenz auf 53,2 ms.

Unser Algorithmus zur Kollisionsvermeidung läuft lediglich auf der ARM-CPU. Aufgrund der geringen Komplexität des Algorithmus liegt dessen Laufzeit weit unter der Laufzeit zur Berechnung der Disparitätskarte.

Eine quantitative Evaluation der Implementierung der Disparitätsschätzung erfolgte anhand des KITTI Stereo 2015 Benchmark-Datensatzes [17]. Für die Evaluation haben wir die Bilder auf eine 640×360 Pixel große ROI beschnitten. Tabelle 3 zeigt die Ergebnisse für die Benchmark. Unter anderem kann man daraus entnehmen, dass die Verwendung der CT ein robusteres Matching gegenüber der Verwendung der SAD Kostenfunktion erlaubt. Dies führt zu genaueren Ergebnissen und dichteren Disparitätskarten. Abbildung 3 zeigt exemplarische Ergebnisse der Disparitätsschätzung, die diese Schlussfolgerungen veranschaulichen. Eine Demonstration der Ergebnisse in Form einer Videosequenz befindet sich unter <https://youtu.be/gzIFqUmQM7g>.

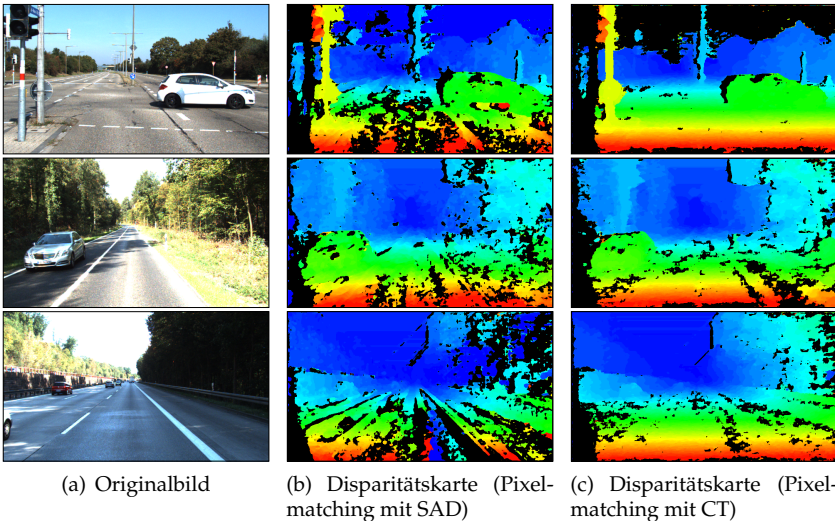


Abbildung 3: Exemplarische Visualisierung der mit unterschiedlichen Pixel-Matching-Verfahren geschätzten Disparitätskarten für drei Stereo-Bildpaare aus dem KITTI Stereo 2015 Benchmark-Datensatzes [17].

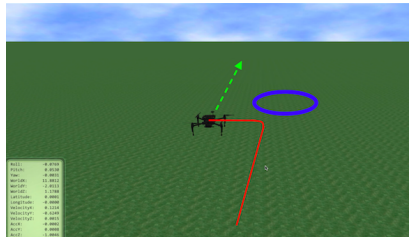
Tabelle 3: Ergebnisse der Disparitätsschätzung bei Verwendung unterschiedlicher Kostenfunktionen für die KITTI Stereo 2015 Benchmark. Die Dichte bezeichnet den Prozentsatz der Pixel, für welche eine gültige Disparität berechnet werden konnte. Die Disparitätsschätzung eines Pixels wird als korrekt betrachtet, wenn ihre Differenz zur Groundtruth weniger als 3 Pixel ist.

Kostenfunktion	Dichte	Anzahl korrekter Pixel
SAD	64,1 %	76,0 %
CT	74,2 %	95,4 %

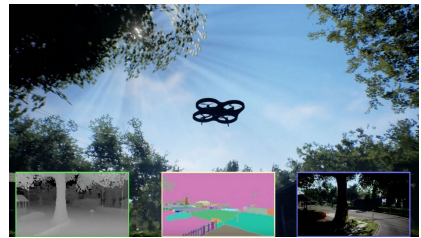
Insgesamt zeigt sich somit, dass die erzielte Verarbeitungsgeschwindigkeit und Genauigkeit für unseren Anwendungsfall ausreichen.

3.2 Evaluation der Hindernisdetektion und Kollisionsvermeidung

Die Funktionalität der Kollisionsvermeidung wurde anhand von Hardware-in-the-Loop-Tests für zwei unterschiedliche UAV-Systeme validiert, nämlich für die DJI Matrice 100 und für ein Pixhawk-basiertes System. Dazu haben wir die UAV-Flugsimulator-Software von DJI sowie Microsoft AirSim [18] verwendet. Abbildung 4 zeigt Screenshots der beiden Flugsimulatoren.



(a) Screenshot der Simulationssoftware von DJI



(b) Screenshot der AirSim Simulationssoftware von Microsoft

Abbildung 4: Screenshots von zwei Flugsimulatoren, die für das Hardware-in-the-Loop-Testen verwendet wurden. Das Hauptfenster in (b) zeigt die Sicht eines externen Beobachters, während die drei kleine Fenster am unteren Rand die Disparitätskarte, die Objektsegmentierung und das Kamerabild des UAV beinhalten.

Beide Flugcontroller waren über die USB-Schnittstelle mit dem PC verbunden, auf dem die Simulationssoftware ausgeführt wurde. Ausgehend von den aus der Simulations-Software erhaltenen Daten, sollte unser System eine Flugplanung durchführen und die entsprechenden Steuerbefehle an die Simulations-Software zurückliefern. Durch diese Hardware-in-the-Loop-Anbindung konnte eine automatische Navigation des UAV simuliert und getestet werden. Die Tests wurden im autonomen Flugmodus durchgeführt. Dabei haben wir das folgende Verhalten des UAV zugrunde gelegt: Nach dem Start steigt das UAV in 1m Höhe und fliegt dann geradeaus. Befindet sich ein Hindernis im Flugpfad, findet das System den kürzesten Weg zu seinem Rand und bewegt sich entsprechend nach links, rechts, oben oder unten. Sobald das Hindernis aus dem Sichtfeld verschwindet, bewegt sich das UAV wieder geradeaus.

Für die Tests der grundlegenden Funktionalität der Kollisionsvermeidung haben wir den proprietären Flugsimulator von DJI verwendet. Dieser ist zwar sehr minimalistisch, enthält jedoch die für die ersten Versuche notwendige Funktionalität. Eine Demonstration der durchgeführten Simulation befindet sich unter <https://youtu.be/pMDMTUCVwKc>.

Für realistischere Tests haben wir den Pixhawk-Flugcontroller verwendet, welches durch den Microsoft AirSim-Flugsimulator unterstützt wird. Der AirSim-Simulator basiert auf der Unreal Engine, welche fotorealistische Modellierung der Umgebung bietet. Auch erlaubt AirSim Tiefen- und damit auch Disparitätskarten der simulierten Umgebung zu generieren. Dadurch ist es möglich die Leistungsfähigkeit der Hindernisausweichung zu testen und sein Verhalten in verschiedenen Szenarien zu analysieren. Zu beachten ist allerdings, dass die durch den Simulator generierten Disparitätskarten keine in realen Szenarien auftretenden Artefakte wie Rauschen und Ausreißer enthalten, so dass die in der Simulation erreichten Ergebnisse in ähnlichen Szenarien in der realen Umgebung überprüft werden müssen.

4 Fazit & Ausblick

In diesem Beitrag haben wir uns mit der Entwicklung eines Systems zur Detektion von Hindernissen und Kollisionsvermeidung für UAVs

beschäftigt. Dazu haben wir eine Optimierung des bekannten SGM-Verfahrens für die Ausführung auf embedded Hardware durchgeführt und einen reaktiven Ansatz zur Kollisionsvermeidung umgesetzt. Unsere Experimente zeigen, dass mit unserem System eine echtzeitfähige Detektion von Hindernissen und eine Kollisionsvermeidung für UAVs, welche starken Gewichts- und Leistungseinschränkungen unterliegen, möglich ist.

Unsere aktuellen Arbeiten konzentrieren sich auf die Durchführung der Experimente in einer realen Umgebung. Parallel dazu sind wir dabei, eine Evaluation unserer Implementierung der Disparitätsschätzung in Bezug auf ihre Energieeffizienz auf unterschiedlichen Plattformen, nämlich FPGA, CPU und GPU, durchzuführen.

Literatur

1. A. J. Barry, H. Oleynikova, D. Honegger, M. Pollefeys, and R. Tedrake, "FPGA vs. Pushbroom Stereo Vision for MAVs," in *IROS Workshop on Vision-based Control and Navigation of Small Lightweight UAVs*, 2015.
2. H. Oleynikova, D. Honegger, and M. Pollefeys, "Reactive Avoidance Using Embedded Stereo Vision for MAV Flight," in *Proc. IEEE International Conference on Robotics and Automation*, 2015, pp. 50–56.
3. H. Hirschmueller, "Stereo Processing by Semi-Global Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.
4. R. Spangenberg, T. Langner, S. Adfeldt, and R. Rojas, "Large Scale Semi-Global Matching on the CPU," in *Proc. IEEE Intelligent Vehicles Symposium*, 2014, pp. 195–201.
5. S. K. Gehrig and C. Rabe, "Real-Time Semi-Global Matching on the CPU," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2010, pp. 85–92.
6. C. Banz, H. Blume, and P. Pirsch, "Real-Time Semi-Global Matching Disparity Estimation on the GPU," in *Proc. IEEE International Conference on Computer Vision Workshops*, 2011, pp. 514–521.
7. I. Haller and S. Nedeveschi, "GPU Optimization of the SGM Stereo Algorithm," in *Proc. IEEE International Conference on Intelligent Computer Communication and Processing*, 2010, pp. 197–202.

8. J. Hofmann, J. Korinth, and A. Koch, "A Scalable High-Performance Hardware Architecture for Real-Time Stereo Vision by Semi-Global Matching," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 27–35.
9. Y. Li, Z. Li, C. Yang, W. Zhong, and S. Chen, "High Throughput Hardware Architecture for Accurate Semi-Global Matching," *Integration*, 2017.
10. T. Kalb, L. Kalms, D. Göhringer, C. Pons, F. Marty, A. Muddukrishna, M. Jahre, P. G. Kjeldsberg, B. Ruf, T. Schuchert, I. Tchouchenkov, C. Ehrenstrahle, F. Christensen, A. Paolillo, C. Lemer, G. Bernard, F. Duhem, and P. Millet, "TULIPP: Towards Ubiquitous Low-power Image Processing Platforms," in *Proc. International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation*, July 2016, pp. 306–311.
11. Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330–1334, 2000.
12. R. Zabih and J. Woodfill, "Non-parametric Local Transforms for Computing Visual Correspondence," in *Proc. European Conference on Computer Vision*, 1994, pp. 151–158.
13. C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch, "Real-Time Stereo Vision System using Semi-Global Matching Disparity Estimation: Architecture and FPGA-Implementation," in *Proc. International Conference on Embedded Computer Systems*, 2010, pp. 93–101.
14. R. Labayrade, D. Aubert, and J. P. Tarel, "Real Time Obstacle Detection in Stereovision on Non Flat Road Geometry Through „V-disparity“ Representation," in *Proc. IEEE Intelligent Vehicle Symposium*, vol. 2, Jun. 2002, pp. 646–651 vol.2.
15. Y. Li and Y. Ruichek, "Occupancy Grid Mapping in Urban Environments from a Moving On-Board Stereo-Vision System," *Sensors*, vol. 14, no. 6, pp. 10 454–10 478, 2014.
16. S. Suzuki and K. Abe, "New Fusion Operations for Digitized Binary Images and Their Applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 6, pp. 638–651, Nov. 1985.
17. M. Menze and A. Geiger, "Object Scene Flow for Autonomous Vehicles," in *Proc. Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3061–3070.
18. S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," in *Proc. Field and service robotics*, 2018, pp. 621–635.

Reliable multilane detection and classification using a compact encoder-decoder CNN

Shriyash V. Chougule¹, Asad Ismail², Ganesh Adam¹, Vikram Narayan², and Matthias Schulze²

¹ Visteon Corporation, Pune, India

² Visteon Corporation, Karlsruhe, Germany

Abstract Reliable lane detection is crucial functionality for autonomous driving. Additionally positional information of ego lanes and side lanes is pivotal for critical tasks like overtaking assistants and path planning. In this work we present a CNN based segmentation approach for detecting multiple lanes as well as positionally classifying them. Wherein we optimize an encoder-decoder architecture for speed and accuracy. We evaluate our network against ENet and SegNet, and found it performing notably better in terms of accuracy. Our network achieves a promising 23 FPS performance on Nvidia's PX2 platform and it has been validated on our test vehicle in highway driving conditions.

Keywords Multilane detection, CNN, segmentation.

1 Introduction

Advanced Driver Assistance Systems (ADAS) is experiencing rapid adoption and growth in automotive industry [1]. A recent study [2] estimated that ADAS functions are effective in reducing head-on and single-vehicle crashes, and minimizing the driver injury risks. Among these safety-critical functionality, capability of autonomous navigation is most sought after function in ADAS. The ability to accurately and reliably detect ego lane and side lanes lie at the core of autonomous driving, and it also serves other critical driving assistance tasks like lane keeping, lane departure warning, and path planning [3].

A vision based lane detection approach provides a low cost solution, but is required to perform robustly in complex driving scenarios. Extracting lane signature from road surface becomes challenging due to



Figure 1: Example of multilane detection and classification by our network, where the lane boundaries are classified as *leftside*, *leftego*, *rightego* and *rightside* lane boundaries and are shown by orange, green, red and blue colors respectively.

varying road appearance. Low light situations like sunset, dawn and night add to the challenge. Varying road texture due to shadows casted by trees, vehicles and buildings demands extreme robustness. Other inevitable situations like worn-out lane markings, weakly marked lane boundaries (using periodically placed reflectors) and occlusions due to vehicles is very common driving scenario. Also, due to the road geometry, the side lane signatures are relatively feeble than ego lane in a given image. Thus estimating side lane boundaries is relatively difficult and rarely explored problem. Model based approaches use traditional computer vision techniques to devise specialized hand-crafted features. Such solutions usually works under a controlled environment and are likely to fail in complex driving scenarios. Convolutional neural networks (CNN) have demonstrated a superior performance in pattern recognition. A computer vision approach, devised around a CNN, has the potential to bring about a robust solution to autonomous driving.

In this work, we intent to exploit the CNN's ability of recognizing complex patterns for gaining robustness in challenging driving situations. Towards that direction we investigate architectures of renowned semantic segmentation networks (ENet [4] and SegNet [5]) to assess their potential in segmenting lane boundaries. We observed that these networks have inherent low sensitivity to objects that occupy relatively few pixels in a image (like lane markings and poles). With these acquired insights, an encoder-decoder CNN architecture was chosen for optimization to segment-out lane boundaries as well as classify them positionally. The network and the dataset used for training is described in Section 3. In Section 4, we compare our network against the ENet and SegNet where we demonstrate that our network is notably more reliable than the rest. Examples of multilane detection and classification by our network are shown in Fig. 1

2 Related work

Recent approaches incorporating CNNs for the lane marker detection [6] have proven to be more robust than model based methods. A CNN used as a preprocessing step in a lane detection system [7], helps in enhancing edge information by noise suppression. DeepLanes [8] detects immediate sideward lane markings with laterally mounted camera system. Although DeepLanes achieve real time performance with high accuracy, it cannot detect road turning ahead. Multilane detection method in [9] makes use of CNN and regression to identify line segments that approximate a lane boundary effectively, it requires high resolution images to work with which hampers the speed. SegNet [5] based multilane detection network [10] segment out lanes, though promising the segmented mask are not accurate at road turnings. VPGNet [11] detects and classify lane markings along with road informative markings, it is inspired by the human intuition of identifying lane layout from global context like road structure and traffic flow. VPGNet trains a baseline network for task of vanishing point detection and further fine tune it for lane and road marking detection task, which helps to improve the overall accuracy. To improve segmentation accuracy of thin and elongated objects like poles and lanes boundaries, Spatial CNN (SCNN) [12] replaces the conventional layer-by-layer convolution with slice-by-slice convolution within feature maps. Such an arrangement provides information flow between pixels across rows and columns, which is hypothesized to be effective for summarizing the global context. A GAN framework is utilized in [13] for lane boundary segmentation, where the discriminator network is trained using an “embedding loss” function which iteratively helps the generator network in learning of higher structural semantics of lanes. LaneNet [14] is a instance segmentation network which makes use of a branched structure to output binary lane segmentation mask and pixel localization mask, which is further used to infer lane instance by clustering process. Apart from LaneNet, a separate network is trained for obtaining parametric lane information from lane instance segmentation mask.

In this work we optimize an encoder-decoder architecture [15], along with borrowing design choices from SCNN and ENet to improve accuracy and speed respectively.

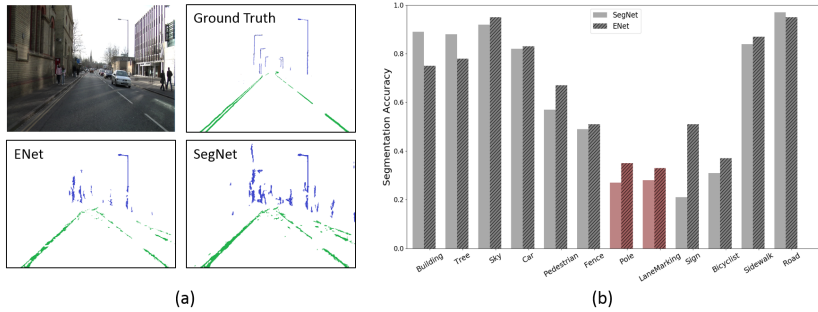


Figure 2: Performance of SegNet and ENet on Camvid dataset (a) Fragmented lane markings and poles in the prediction mask (b) Segmentation accuracy.

3 Multilane detection and classification network

Lane markings and poles have peculiar shape of being an elongated and thin objects, usually occupying less than 5% of the total pixels in a driving scene image. We analyze two prominent semantic segmentation networks like SegNet [5] and ENet [4], to evaluate their potential for segmenting lane boundaries. We summarize their performance on Camvid dataset, which contain examples of an urban driving scenario. As can be seen in Fig. 2 (b), the segmentation accuracy for lane markings (road markings) class and pole class is low in comparison with remaining classes. As a consequence of poor segmentation accuracy, the segmented lane markings appear fragmented due to many false positive and false negative as shown in Fig. 2 (a). Inferring lane boundaries from these fragmented lane markings demands designing of a tedious post-processing stage, followed by the use of a tracking framework. Fragmentation also hinders the prospect of estimating turns and curvature of the road ahead, which is a key information for critical driving assistance tasks (such as path planning and lane keeping). In the following subsections we elaborate our network design decisions to boost the segmentation accuracy of lane boundaries and minimize their fragmentation.

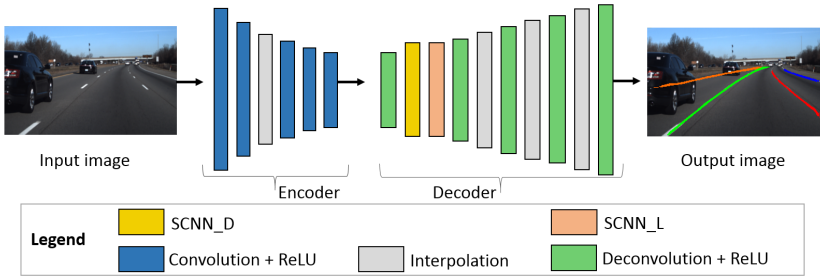


Figure 3: Network architecture.

3.1 Use of encoder-decoder architecture

Use of gradually graded deconvolution layers instead of a single deconvolution layer is known to construct denser and much precise segmentation mask [16]. Thus the segmentation accuracy of lane boundaries can greatly benefit by these graded deconvolutional layers. So we choose to optimize a symmetrical encoder-decoder architecture where the encoder section (or convolutional network) is a sequence of convolutional layers, followed by the decoder section (or deconvolutional network) involving deconvolutional layers. The encoder section behaves as a feature extractor whereas the decoder section constructs a segmentation map from the extracted features. We reduce the original 39 layers encoder-decoder network [15] to 10 layers network i.e. 5 convolutional and 5 deconvolution layers (see Fig. 3). Such compact structure of encoder-decoder network significantly cut down the computations, and yet is observed to be sufficiently accurate for segmenting lane boundaries as demonstrated in Section 4. Also we add interpolation layers [17] at appropriate positions, which helps in resizing the feature map to match input dimensions of a successive layer. We also implement the *SCNN_D* and *SCNN_L* layer of Spatial-CNN [12] as explained in Section 3.3, in order to improve segmentation accuracy.

3.2 No pooling operations

It is well known in the computer vision domain that for any given method, lowering the input image resolution provides execution speed

up at the cost of accuracy. ENet leverages this fact and achieves impressive speedup by employing a strategy of early down sampling, done by its first two bottleneck layers. Instead of going for early down sampling strategy, we choose to work directly with a low resolution input image to uplift the execution speed. And to counter the accuracy loss because of lowering the input image resolution, we choose to remove the max pooling layers and corresponding unpooling layers from the original deconvolution network. Max-pooling is ubiquitous procedure used to retain most active neurons, which translates in decreased features dimensionality as well as introduces some degree of translation invariance. However, in our problem we found that losing spatial information by using max pooling layer is indeed not desirable. Without pooling layers our network produces much finer segmentation masks, and saves the computation time as well.

3.3 Spatial information flow

We implement the *SCNN_D* and *SCNN_L* spatial information flow modules/layers of SCNN to further enhance the segmentation accuracy. As described in the original work, the *SCNN_D* layer establishes an information flow from top to bottom within a feature map, whereas *SCNN_L* establishes information flow from left to right. Other two spatial information modules/layers (*SCNN_U* and *SCNN_R*) were ignored in order to save computations. All the modules use a general strategy to slice the 3D feature map along the information flow directions under consideration. These feature slices are used generate a new set of feature slices, using successive convolution and element-wise addition operations given by below equation:

$$slice_i^{new} = \begin{cases} slice_i^{old}, & \text{if } i = 1 \\ slice_i^{old} + f(convolution(slice_{i-1}^{new})) & \end{cases} \quad (1)$$

where f is nonlinear activation function like ReLU, i indicates the slice sequence observed w.r.t. the information flow direction. The new slices are concatenated along the information flow direction to create a new 3D feature map. Network implementation details along with the convolution operation in above equation is given in the Table 1.

Layer parameters
conv{output, padding, kernel, group, stride }
deconv{output, padding, kernel, group, stride }
interpolation{height, width }
scnn{direction, conv{output, padding, kernel, group, stride } }
Encoder
conv{13, 1, 3, 1, 2 }
conv{96, 0, 3, 1, 1 }
interpolation{23, 40 }
conv{200, 1, 3, 4, 1 }
conv{200, 1, 3, 4, 1 }
conv{128, 1, 3, 2, 2 }
Decoder
deconv{96, 1, 3, 2, 1 }
scnn{top to bottom, conv{96, 1, 3, 1, 1 } }
scnn{left to right, conv{96, 1, 3, 1, 1 } }
deconv{128, 1, 3, 2, 2 }
interpolation{23, 40 }
deconv{96, 1, 3, 2, 1 }
interpolation{89, 157 }
deconv{16, 1, 3, 1, 1 }
interpolation{184, 320 }
deconv{5, 1, 3, 1, 1 }

Table 1: Network description (Caffe deeplearning framework context).

3.4 Dataset generation

To train our network for lane segmentation task, we gathered a dataset containing 33000 images of highway driving scenarios and manually annotated the lane boundaries. Images from TuSimple lane detection challenge dataset were included as well, after adapting its ground truth by fitting curves on the list of lane boundary points. An image in the dataset may have 1 to 4 annotated lane boundaries depending upon the scene. The ground truth information for each lane boundary is represented by a gray scale image where the lane boundary pixels are given distinct integer values which act as labels to the four lane types i.e. *rightego*, *leftego*, *rightside* and *leftside* and the background. All images are of

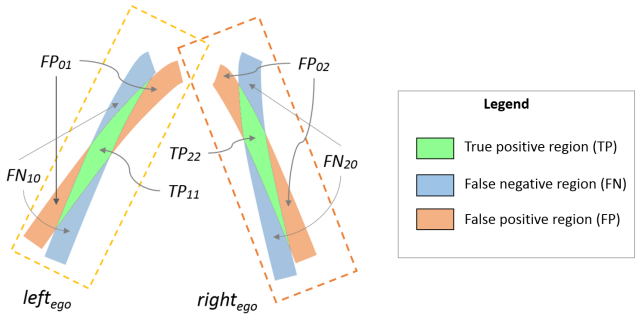


Figure 4: Illustration of MIoU metric components, where the background, *left_ego* boundary and *right_ego* boundary have class id as 0, 1 and 2 respectively.

184 x 320 resolution which were recorded in normal weather conditions, during day and evening time.

3.5 Network training

Caffe deeplearning framework was used for training our network as well as for ENet and SegNet training. The dataset was split into 32000 training and 1000 testing images. On an average, out of the total pixels in a training image, the background label accounted for 97.25% pixels whereas *left_side*, *left_ego*, *right_ego* and *right_side* labels accounted for 0.59%, 0.91%, 0.82% and 0.43% pixels respectively. Because of the uneven sample counts, a weighted softmax loss function was used in training all the networks where the class loss weights were based on label percentage i.e. $(1 - percentage_{label})$. All the networks were trained till convergence to 92% training accuracy using ADAM optimizer.

Network	MIoU
SegNet	52.48
ENet	64.23
Ours	76.39

Table 2: Lane detection accuracy measured using MIoU metric.

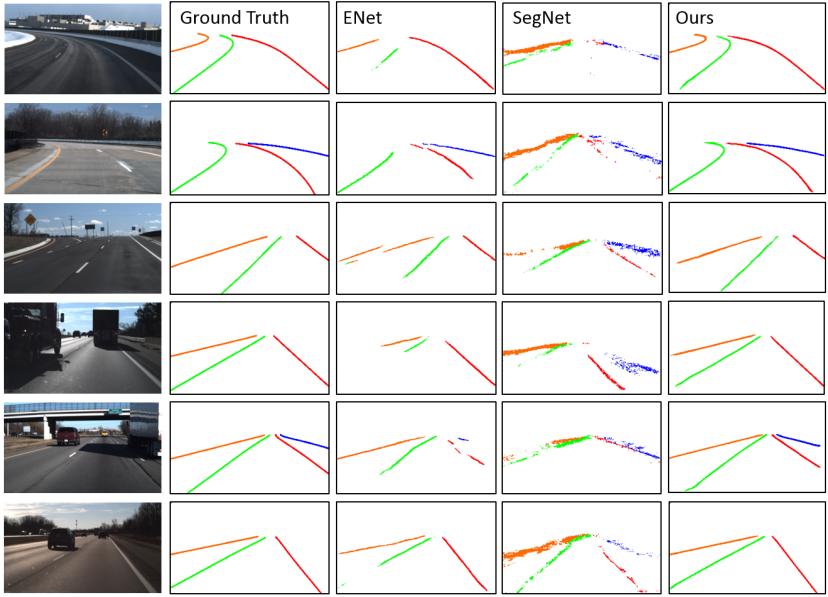


Figure 5: Performance in challenging scenarios.

4 Evaluation

Most common metrics for evaluating segmentation accuracy are based on similarity measures between predicted and ground truth pixels. The predictions of all the networks were obtained on test dataset and then compared to the ground truth (grayscale images) using the MIoU (mean intersection over union) metric. The MIoU metric for a single test image is defined as follows:

$$MIoU = \frac{1}{1+k} \sum_{i=0}^k \frac{TP_{ii}}{\sum_{j=0}^k FN_{ij} + \sum_{j=0}^k FP_{ji} - TP_{ii}} \quad (2)$$

where k is number of classes ($k=5$ in our case i.e. *rightego*, *leftego*, *rightside*, *leftside* and *background*). TP , FN and FP are pixel counts in the true positive, false negative and false positive regions respectively as illustrated in Fig. 4. Performance of networks is summarized in Table 2.



Figure 6: Scenarios where network performance dropped.

The predictions on challenging scenarios is shown in Fig. 5 for visual comparison, where it can be observed that our network predictions are relatively more accurate than that of ENet and SegNet. Particularly our network is more reliable in capturing the road curvature at turnings ahead, during occlusions due to vehicles, and at approaching road split where ENet tends to confuse. Fig. 6 shows cases where our network performance dropped. It includes case where over-bridge shadow is covering more than 50% of scene, multiple lane markings, and when the vehicle is changing lane. Our network has been validated on our test vehicle using Nvidia's PX2 platform where a consistent and stable detection was observed, with a promising performance of 23 FPS.

5 Conclusions

We presented a CNN based segmentation approach for detecting multiple lane boundaries and classifying them based on position. Our network demonstrated reliable performance in challenging driving scenarios like lane occlusions due to vehicle, poorly lane marked roads and in low evening light, when compared to ENet and SegNet. Also it accurately captured the road turnings compared to the rest. A performance drop was observed during lane changing, which we intend to address either by tracking or by use of feedback framework analogous to RNN. We also intend to modify our network to directly output parametric lane boundaries, which is a more convenient format to work with.

References

1. K. Bengler, K. Dietmayer, M. Maurer, and H. Winner, "Three Decades of Driver Assistance Systems Review and Future Perspectives," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. October, pp. 6–22, 2014.

2. S. Sternlund, J. Strandroth, M. Rizzi, A. Lie, and C. Tingvall, "The effectiveness of lane departure warning systems—A reduction in real-world passenger car injury crashes," *Traffic Injury Prevention*, vol. 18, no. 2, pp. 225–229, feb 2017.
3. A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: A survey," pp. 727–745, 2014.
4. A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet : A Deep Neural Network Architecture for Real-Time Semantic Segmentation arXiv : 1606 . 02147v1 [cs . CV] 7 Jun 2016," pp. 1–10, 2016.
5. H. A. C. R. Badrinarayanan V., "A deep convolutional encoder-decoder architecture for robust semantic pixelwise labelling," *CoRR*, vol., "abs/1505.07293, 2015.
6. J. Li, X. Mei, D. Prokhorov, and D. Tao, "Deep Neural Network for Structural Prediction and Lane Detection in Traffic Scene," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 690–703, mar 2017.
7. J. Kim, J. Kim, G. J. Jang, and M. Lee, "Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection," *Neural Networks*, vol. 87, pp. 109–121, mar 2017.
8. A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali, "DeepLanes: End-To-End Lane Position Estimation Using Deep Neural Networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, jun 2016, pp. 38–45.
9. A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: A survey," pp. 727–745, apr 2014.
10. J. Kim and C. Park, "End-To-End Ego Lane Estimation Based on Sequential Transfer Learning for Self-Driving Cars," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2017-July. IEEE, jul 2017, pp. 1194–1202.
11. S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T. H. Lee, H. S. Hong, S. H. Han, and I. S. Kweon, "VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, oct 2017, pp. 1965–1973.
12. X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial As Deep: Spatial CNN for Traffic Scene Understanding," dec 2017.
13. M. Ghafoorian, C. Nugteren, N. Baka, O. Booij, and M. Hofmann, "EL-GAN: Embedding Loss Driven Generative Adversarial Networks for Lane Detection," jun 2018.

14. D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards End-to-End Lane Detection: an Instance Segmentation Approach," 2018.
15. H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, may 2015, pp. 1520–1528.
16. E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, apr 2017.
17. L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

Adaptive Algorithmenauswahl für die Fahrzeugdetektion auf Luftbilddaufnahmen

Adaptive algorithm selection for vehicle detection in aerial imagery

Christian Hellert und Peter Stütz

Universität der Bundeswehr München, Institut für Flugsysteme,
Werner-Heisenberg-Weg 39, 85577 München

Zusammenfassung Durch den steigenden Automatisierungsgrad unbemannter Luftfahrtsysteme stellen sich neue Herausforderungen an die maschinelle Aufklärungsfunktion. Dabei ist deren Leistungsverhalten abhängig von bestimmten Umgebungsbedingungen, wie etwa Wetterbedingungen oder Tageszeiten. Um diesen Limitierungen zu entgegen, stellt diese Arbeit ein adaptives Auswahlverfahren vor, dass je nach Umgebungszustand den geeignetsten Algorithmus für eine bestimmte Aufklärungsaufgabe einsetzt. Hierbei konnte in Simulations- und Flugexperimenten gezeigt werden, dass die Detektionsgenauigkeit insgesamt gesteigert werden konnte.

Schlagwörter Algorithmenauswahl, UAV, Fahrzeugerkennung, Umgebungsbeschreibung, Neuronale Netze.

Abstract The increasing degree of automation of unmanned aerial systems poses new challenges to automated perception of the environment. Their performance depends on certain environmental conditions, for example weather conditions or times of day. To counter these constraints, this work presents an adaptive selection method that applies the most suitable algorithm for a certain reconnaissance task, depending on the environmental conditions. Simulation and flight experiments have shown that the overall detection accuracy could be increased.

Keywords Algorithm selection, UAV, vehicle detection, environmental description, neural networks.

1 Einleitung

Zur Fernerkundung und Aufklärung werden mit Sensoren ausgerüstete bemannte und unbemannte Luftfahrzeuge (UAV, engl. *Unmanned Aerial Vehicle*) eingesetzt. Da sie gegenüber konventionellen Luftfahrzeugen Vorteile hinsichtlich der Einsatz- und Personalkosten, der Sicherheitsaspekte und der durchführbaren Flugmanöver bieten, werden sie zunehmend verwendet. Getrieben durch zukünftige Szenarien, wie die parallele Bedienung von mehreren UAVs durch einen Operateur [1], zeigt sich ein Bedarf an einen erhöhten Automatisierungsgrad. In dieser Arbeit wird die Automatisierung des Sensoreinsatzes behandelt.

Eine besondere Herausforderung beim automatisierten Sensoreinsatz stellt die erdgerichtete Umwelterfassung mittels bildgebender Sensorik da. Denn die Leistungsfähigkeit der Sensoren als auch der Umwelterfassung sind durch spezifische Umweltbedingungen beeinflussbar. So können bei der hier betrachteten Applikation der Fahrzeugerkennung die Detektionsgenauigkeiten eingesetzter Algorithmen fluktuieren, wenn z.B. photogrammetrische, atmosphärische oder topographische Veränderungen eintreten. Um dem entgegenzuwirken wird eine *adaptive Algorithmenauswahl* vorgestellt, welche die Leistung der Algorithmen auf Basis der Umweltbedingungen prädiziert und den geeignetsten auswählt und einsetzt.

Der folgende Abschnitt behandelt den Stand der Technik und im anschließenden Abschnitt 3 wird das Konzept und die Methodik der Algorithmenauswahl vorgestellt. Die Experimente zur Evaluierung des Systems werden in Abschnitt 4 beschrieben und die Ergebnisse in Abschnitt 4 präsentiert. Abschließend gibt der Abschnitt 6 ein Fazit.

2 Stand der Technik

Smith-Miles [2] präsentiert eine theoretische Lösung zum Problem der Algorithmenauswahl, wobei die Selektion der Algorithmen über Methoden des maschinellen Lernens bestimmt wird. Formal stellt sich das Problem wie folgt dar: Für eine gegebene Problem Instanz $x \in P$, mit dem Merkmal $f(x) \in F$, ist die Zuordnung $s(f(x))$ gesucht, die in der Menge der Algorithmen A den Algorithmus α auswählt, so dass die Leistung $y(\alpha(x)) \in Y$ maximiert wird. Diese formale Beschreibung

stellt die Grundlage des *adaptiven Auswahlverfahrens* dar.

Ein adaptives System zur Analyse von Luftbildern wird in [3] vorgestellt, welches in Abhängigkeit der Umgebungsparameter die Leistung von Segmentierungsverfahren evaluiert. Zur Modellierung der Leistungscharakteristik der Algorithmen wurde dabei ein wissensbasierter Ansatz gewählt. Chau et al. [4] präsentieren eine adaptive Objektverfolgung. Mit Hilfe eines Fuzzy-Inferenzsystems (FIS) werden die Parameter eines Tracking-Algorithmus über Kontextmerkmale des zu verfolgenden Objektes justiert, wobei die Fuzzy-Regeln gelernt wurden. Ein ähnlicher Ansatz wird in [5] vorgeschlagen. Dabei werden die Leistungen von Fahrzeugdetektoren über FIS bestimmt. Weiterhin kommen künstliche neuronale Netze (KNN) [6] zur Leistungsbestimmung zum Einsatz. In dieser Arbeit sollen KNN verwendet werden, da diese nicht-lineare Probleme geeigneter lösen und tendenziell eine höhere Genauigkeit erzielen können als FIS.

Zur Erkennung von Fahrzeugen auf Luftbilddaufnahmen wird in [7] ein Template Matching (TM) Ansatz vorgestellt. Um perspektivischen Veränderungen entgegenzuwirken werden mehrere Fahrzeug-Templates herangezogen und diese in ein 3D-Modell überführt. In [8] wird ein Verfahren vorgeschlagen, um Fahrzeuge auf LWIR (engl. *long-wave infrared*) Sensordaten zu detektieren. Dabei wird eine BLOB (engl. *Binary Large Object*) Detektion eingesetzt, welche geringe Objektauflösungen ($< 10px$) handhaben kann. Hummel et al. [9] präsentiert eine Multi-Klassifikationskaskade (CC, engl. *cascading classifier*), die mittels Haar-Merkmalen Fahrzeuge in unterschiedlichen Perspektiven und Orientierungen erkennen kann. Des Weiteren wird in [10] der DPM-Detektor (engl. *Deformable Part Model*) verwendet, um eine Objekterkennung auf Luftbilddaufnahmen zu realisieren. Diese vier Klassifikatoren TM, BLOB, CC und DPM, werden für die hier betrachtete Anwendung der Fahrzeugdetektion als konkurrierende Alternativen verwendet.

Zur Erzeugung der Leistungsmodelle und der Objektklassifikatoren werden Trainingsdaten benötigt. Gängige Datensätze [11, 12] beinhalten Bilddaten und Annotationen für begrenzte Umgebungsbedingungen, welche ungeeignet für die Leistungsbestimmung sind. Da die Akquise realer Bilddaten und deren Annotation zeitaufwendig ist, wird eine Simulation zur Trainingsdatengenerierung herangezogen. Zur Überprüfung des Systems werden zusätzlich reale Sensordaten genutzt.

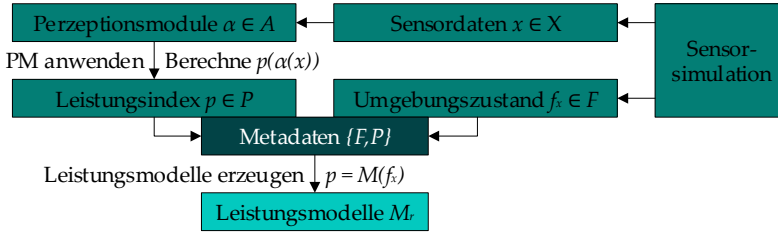


Abbildung 1: Konzeptuelle Darstellung der Leistungsmodellerzeugung.

3 Adaptive Algorithmenauswahl

Die Grundlage für das adaptive Auswahlverfahren ist die Algorithmenauswahl nach Smith-Miles [2]. Zur Realisierung dieser, müssen die betrachteten Algorithmen zur Fahrzeugerkennung hinsichtlich deren Leistung in Abhängigkeit des Umgebungszustandes bewertet werden. Daher wird für jeden Algorithmus ein Leistungsmodell generiert. Die Erzeugung der Leistungsmodelle ist in Abbildung 1 konzeptuell dargestellt. Hierbei wird über eine Sensorsimulation der Modelldatensatz $\{X, F\}$ erzeugt, welcher Sensordaten x mit zugehörigen Annotationen der Fahrzeugpositionen und den Umgebungszustand f_x enthält. Die Algorithmen zur Fahrzeugerkennung werden im folgenden Perzeptionsmodule (PM) genannt. Für jedes dieser PM wird der Leistungsindex p für jede Instanz von X ermittelt. Dabei wird als Leistungsindex das F_1 Maß verwendet. Daraus resultiert pro Algorithmus die Leistungsmatrix P . Zusammen mit den zugehörigen Umgebungszuständen f_x entsteht über die Metadaten $\{F, P\}$ eine Zuordnung zwischen Algorithmenerleistung und Umgebungszustand. Aus den Metadaten können schließlich durch Regression die Leistungsmodelle M_r erzeugt werden.

Mittels der Leistungsmodelle pro PM wird ein Ranking der Algorithmen durchgeführt. Hierbei kann eine Fahrzeugerkennung aus mehreren Perzeptionsmodulen bestehen. Eine valide PM-Kombination wird dabei Perzeptionskette (PK) c genannt. Um die Gesamtheit aller zulässigen Perzeptionsketten auszudrücken, wird das Konstrukt des Perzeptionsgraphen G eingeführt. Der hier konstruierte Graph mit den entsprechenden Bildverarbeitungsalgorithmen ist in Abbildung 2 dargestellt. Ziel der Algorithmenauswahl ist es, die geeignetste PK c_j zu

finden, welche den höchsten über die PM gemittelten Leistungsindex in G besitzt.

Für die Bestimmung des Leistungsindex durch die Leistungsmodelle werden Einflussgrößen zur Beschreibung des Umweltzustandes herangezogen. Die verwendete Sensorsimulation¹ bietet dabei die Möglichkeit 5 atmosphärische Bedingungen einzustellen, welche Teil von f_x sind. Drei davon betreffen die Witterung: Die Niederschlagsmenge Q_p , die Nebeldichte ρ_f und der Bewölkungsgrad W_o , welche einen Wertebereich von $[0, 1]$ besitzen, wobei ein Wert von 0 bedeutet, dass der entsprechende atmosphärische Effekt nicht dargestellt wird und ein Wert von 1, dass der Effekt voll dargestellt wird. Die anderen beiden atmosphärischen Parameter sind die Tageszeit t und der Monat M , welche Rückschlüsse auf den Sonnenstand erlauben.

Weiterhin können 2 photogrammetrische Parameter, welche die geometrische Abbildung der Szenerie auf den Sensor beschreiben, angegeben werden. Diese sind die Bodenpixelauflösung g in $[m/px]$ und der Sensorelevationswinkel θ in $[\circ]$. Zuletzt kann die Bodenbedeckung und die Geländeform ermittelt werden. Dabei setzt sich die Bodenbedeckung aus den Kategorien Ebene t_{os} , Vegetation t_v , Wasser t_w , Gebäude t_b und Straßen t_s zusammen. Diese werden als prozentualer Bedeckungsanteil relativ zur abgetasteten Bodenfläche A des Sensors angegeben. Über die Höhendaten kann die Geländeform bestimmt werden. Hierzu wird der Reliefwert r eingeführt, welcher über die Magnituden der Höhengradienten M_h eine Abtastung der Höhendaten in A mit

$$r = \tanh \left(r_{max} \frac{\sum_{i,j} M_h(i, j)}{A} \right) \quad (1)$$

berechnet. Dabei ist r_{max} ein Skalierungsparameter, der das Maximum von r in $\{X, F\}$ ist. r hat damit den Wertebereich $[0, 1]$. Zur Erzeugung der Leistungsmodelle müssen alle Einflussgrößen normiert werden, damit keiner dieser überbewertet wird und die Regression schneller konvergiert. Die Bodenpixelauflösung wird normiert mit,

$$g = \tanh \left(g_{max} \sqrt{\frac{A}{H \cdot W}} \right) \quad (2)$$

¹ VBS3 (engl. *Virtual Battlespace 3*), Herstellerseite: <https://bisimulation.com/> [Letzter Zugriff: 05.09.2018]

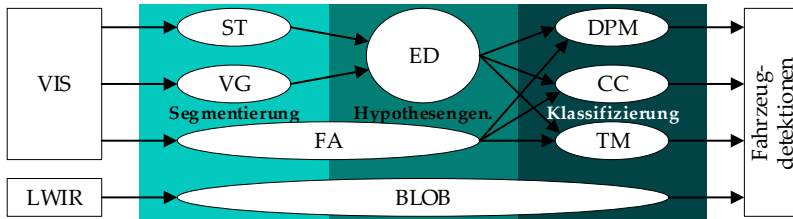


Abbildung 2: Perzeptionsgraph der Fahrzeugdetektion.

wobei g_{max} ein Skalierungsparameter (analog zu r_{max}), H die Höhe und W die Breite des Sensorbildes in Pixeln ist. Außerdem wird θ durch 90° geteilt und die Tageszeit und der Monat als Polarkoordinaten interpretiert und in normierte kartesische Koordinaten (x_t, y_t) und (x_M, y_M) umgewandelt.

Der Prozess der Erzeugung der Leistungsmodelle besteht aus der *Datenaufbereitung*, der *Clusterbildung* und der *Regression*. Die Datenaufbereitung normiert die Einflussfaktoren der Rohdaten $\tilde{D}_r = \{F, P\}$. Die anschließende Clusterbildung fasst identische Metamerkmalsvektoren zusammen und weist diesen über das F_1 Maß einen Leistungsindex zu. Abschließend werden zur Regression neuronale Netze verwendet, die mittels Fehlerrückführung trainiert werden.

Zur Evaluierung der Algorithmenauswahl wird die Applikation der Fahrzeugdetektionen herangezogen. Dabei zeigt Abbildung 2 schematisch den Perzeptionsgraphen mit den verwendeten Bildverarbeitungsalgorithmen. Dabei bilden die 4 ausgewählten Klassifikatoren aus Abschnitt 2 und 4 Vorverarbeitungsalgorithmen, Straßensegmentierung (ST), Vordergrundsegmentierung (VG), Farbraumanalyse (FA) und Eckpunktdetektor (ED), den Graphen.

Die Straßensegmentierung konvertiert das Eingangsbild in den HSV-Farbraum und maskiert dort asphaltierte Flächen mit den Schwellwertbereichen $H = [0\%, 100\%]$, $S = [0\%, 27\%]$ und $V = [60^\circ, 300^\circ]$. Um zu kleine Segmente auszusortieren, erfolgt die morphologische Operation *Schließen* und eine Konturendetektion. Damit bei flachen Sensorelevationswinkeln Himmelregionen von der weiteren Verarbeitung ausgeschlossen werden, nutzt die Vordergrundsegmentierung das OTSU-Verfahren. Dabei wird jeder Farbkanal einzeln behandelt und die

Binärbilder anschließend vereinigt und durch die Operation *Schließen* geglättet. Der Eckpunktdetektor generiert aus den vorherigen Segmentierungsschritten durch den Harris-Corner-Detektor Hypothesen für die Klassifikatoren. Die Farbraumanalyse hat das Ziel, typische Fahrzeugfarben zu segmentieren. Dabei erzeugt jeder eingesetzte Farbfilter ein Binärbild, welche vereinigt werden. Abschließend findet eine Filterung der segmentierten Regionen anhand deren Fläche statt, um die Hypothesen festzulegen.

4 Experimente und Evaluierungsstrategie

Zur Evaluierung der Algorithmenauswahl wurden zwei Experimente durchgeführt. Das erste Experiment wurde anhand einer *Observationsmission* in der Simulation durchgeführt. In dieser Mission wird ein Fahrzeug 24 Stunden beobachtet, welches von einer Hafenanlage zu einer Einrichtung und anschließend, in der Morgendämmerung, zu einem Gebäude in einer Stadt fährt. Primäres Ziel dieser Mission ist die Untersuchung der vier Fahrzeugdetektoren ohne Vorverarbeitung unter wechselnden Witterungen. Die aufgezeichneten veränderlichen Einflussgrößen sind in 3 über die Missionszeit dargestellt. Die Mission fand fiktiv im November statt, wobei g konstant zu $0,00694m/px$ gewählt wurde.

Das zweite Experiment dient der Evaluierung des adaptiven Perzeptionssystems während eines Flugversuches zur *Gebietsaufklärung*. Verwendet wurde ein Flächenflugzeug, welches mit einem schwenkbaren VIS Sensor ausgestattet ist und von einem Piloten gesteuert wird. Das Perzeptionssystem wird von einer Bodenkontrollstation aus kommandiert, indem das Zielgebiet und der Perzeptionsauftrag *Fahrzeugerkennung* an dieses übermittelt wird. Abbildung 4 zeigt die Einflussgrößen, die sich über die Missionszeit verändern.

Zur Evaluierung der Leistungsmodelle wird eine Referenzbestimmung der Algorithmenleistung verwendet. Die Referenzbestimmung ermittelt aus den Evaluierungsdaten $\tilde{\epsilon}$ (Missionsdaten) für jede PK den Leistungsindex anhand des F_1 Maßes. Für die Ermittlung der Leistungsindizes der Modelle werden die Einflussgrößen in $\tilde{\epsilon}$ mit der Datenaufbereitung, wie oben beschrieben, normiert und jedem Modell M_i präsentiert. Über den Perzeptionsgraphen können die Leistungsindizes

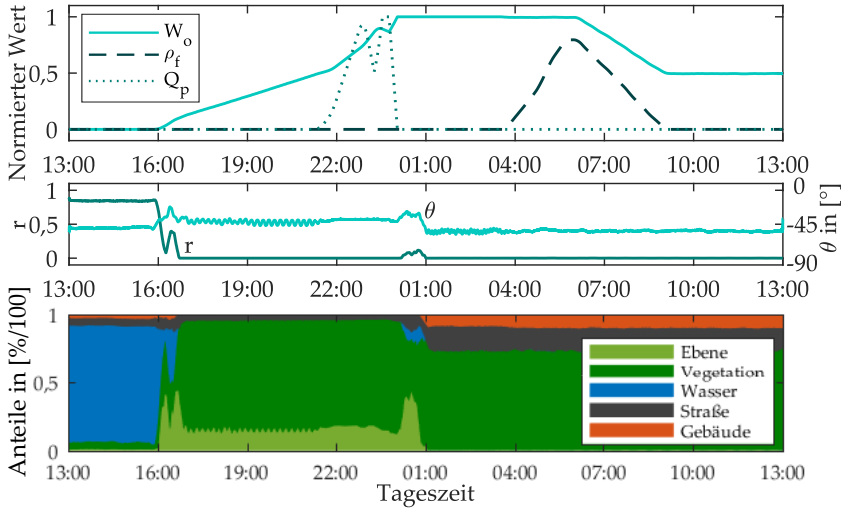


Abbildung 3: Einflussgrößen der Mission *Observierung*.

akkumuliert werden und der Leistungsindex für jede PK bestimmt werden. Für die Evaluierung wird das Qualitätsmaß q eingeführt, welches als Vergleichsmaß zwischen den Perzeptionsketten einer Mission dient. Berechnet wird q mit Formel 3, wobei i ein Zeitschritt in einer Mission und N die Zahl der Zeitschritte angibt.

$$q = 1 - \sum_{i=1}^N (p_i - 1)^2 \quad (3)$$

5 Ergebnisse und Diskussion

Die Ergebnisse für die *Observierungsmission* sind in Abbildung 5 dargestellt, wobei der obere Graph die Referenzbestimmung, der mittlere Graph die Leistungsindizes der Leistungsmodelle und der untere Graph abschließend die Algorithmenauswahl zeigt. Hierbei kann festgestellt werden, dass die Kurvenverläufe der einzelnen Leistungsmodelle mit denen der Referenzbestimmung über die Missionszeit

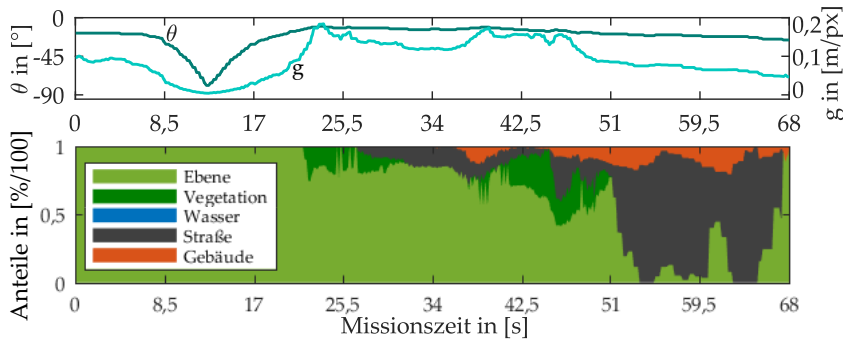


Abbildung 4: Einflussgrößen der Mission *Gebietsaufklärung*.

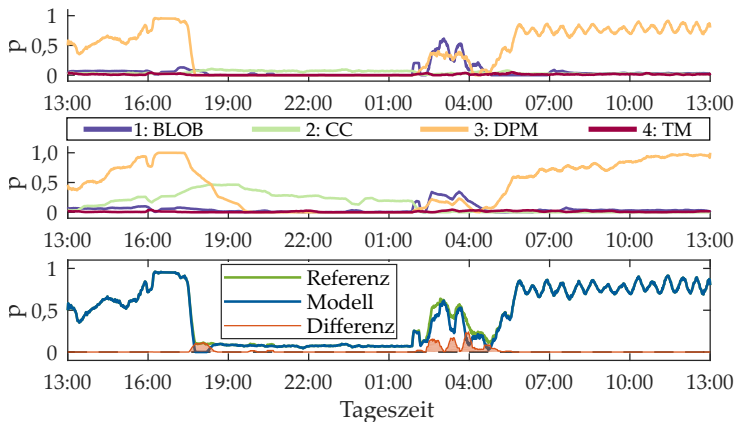


Abbildung 5: Leistungsindizes der Mission *Observierung*.

vergleichbar sind. Weiterhin lässt sich festhalten, dass ein steigender Bewölkungsgrad zu einer höheren DPM und CC Leistung führt (siehe ab 16:00 Uhr Missionszeit). Allerdings führt ein zu hoher Bewölkungsgrad $W_o > 0,5$ mit zusätzlichem Nebelaufkommen $\rho_f > 0,5$ zu verringerter Leistung. Außerdem liefert das CC Leistungsmodell höhere Leistungsindizes als die Referenzmessung zeigt, was auf die Generalisierungseigenschaften des verwendeten Regressionsverfahrens zurückzuführen ist.

Tabelle 1 zeigt die Auswertung der Algorithmenauswahl für die *Observierungsmission*. Als *bester* Algorithmus über die gesamte Missionszeit hat sich der DPM Detektor behauptet. Die Referenzbestimmung zeigt, dass mit der adaptiven Algorithmenauswahl maximal ein Plus von 15,63% möglich ist und durch die Leistungsmodelle 11,17% erreicht wurde, mit einer Genauigkeit von 96,15%. Dabei ist die Genauigkeit das Verhältnis der Fläche unter der Referenzkurve zur der Modellkurve im unteren Graphen in Abbildung 5.

	Referenz	Modell	Abweichung	Genauigkeit
Bester Algorithmus	0,495	—	—	—
Algorithmenauswahl	0,572	0,550	-3,86 %	96,15 %
Verbesserung	15,63 %	11,17 %	-4,47 %	—

Tabelle 1: Auswertung der Mission *Observierung*.

Analog zur vorherigen Mission sind in Abbildung 6 die Leistungsindizes über die Missionzeit der *Gebietsaufklärungsmission* dargestellt. Hierbei ist der BLOB Detektor und das TM nicht gezeigt, da diese von der Algorithmenauswahl nicht selektiert wurden. In großen Teilen der Mission wählt die adaptive Algorithmenauswahl die gleichen Algorithmen wie die Referenz, wie in untersten Graphen zu erkennen ist. Dabei fallen drei Bereich auf, die eine Differenz aufweisen. Grund hierfür sind Situationen wie z.B. partielle Verdeckungen und Bewegungsunschärfe im realen Sensorbild. Da diese von den Leistungsmodellen nicht erfasst werden können, führt dies im Gegensatz zur Referenzbestimmung zu unterschiedlichen Leistungsindizes der Algorithmen und damit zu einer suboptimalen Auswahl.

Die Auswertung der Gebietsaufklärungsmission zeigt, dass gegenüber dem *besten* Algorithmus (DPM-ST) maximal eine Verbesserung von 7,79% möglich wäre und die adaptive Algorithmenauswahl ein Plus von 1,23% erreichen konnte. Dabei lag die Genauigkeit bei 93,84%.

	Referenz	Modell	Abweichung	Genauigkeit
Bester Algorithmus	0,872	—	—	—
Algorithmenauswahl	0,941	0,883	-6,09 %	93,84 %
Verbesserung	7,79 %	1,23 %	-6,56 %	—

Tabelle 2: Auswertung der Mission *Gebietsaufklärung*.

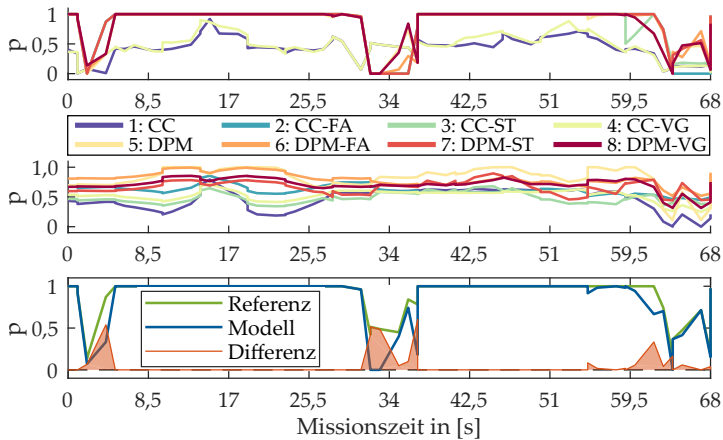


Abbildung 6: Leistungszindizes der Mission Gebietsaufklrung.

6 Fazit

Das vorgestellte adaptive Auswahlverfahren konnte mit den gewhlten Einflussgroen zur Beschreibung des Umgebungszustandes die Detektionsgenauigkeit erhhen. Hierbei wurden Leistungsmodelle eingesetzt, die aus synthetischen Sensordaten mittels neuronaler Netze generiert wurden, um die Leistungscharakteristik der bildverarbeitenden Algorithmen zu prdizieren. Fr Multi-UAV Anwendungen mit heterogenen UAVs, knnen so z.B. UAVs identifiziert werden, die fr eine gegebene perzeptive Aufgabe am geeignetsten sind [13]. Weiterhin lsst sich ber die Leistungsmodelle beurteilen, ob die Bildanalyse sinnvoll durch die Automatisierung durchgefhrt werden kann oder ob ein Operateur diese bernehmen sollte.

Literatur

1. M. Cummings, S. Bruni, S. Mercier, and P. Mitchell, "Automation architecture for single operator, multiple UAV command and control," *The International C2 Journal*, vol. 1, pp. 1–24, 2007.

2. K. A. Smith-Miles, "Cross-disciplinary perspectives on meta-learning for algorithm selection," *ACM Comput. Surv.*, vol. 41, no. 1, pp. 1–25, 2008.
3. P. Robertson and J. Brady, "Adaptive image analysis for aerial surveillance," *IEEE Intelligent Systems and their Applications*, vol. 14, no. 3, 1999.
4. D. P. Chau, K. Subramanian, and F. Bremond, "Adaptive Neuro-Fuzzy Controller for Multi-Object Tracker," *International Conference on Computer Vision Systems*, pp. 466—476, 2015.
5. C. Hellert and P. Stütz, "Performance Prediction and Selection of Aerial Perception Functions during UAV Missions," in *IEEE Aerospace Conf.*, 2017.
6. M. A. Muñoz, M. Kirley, and S. K. Halgamuge, "A Meta-Learning Prediction Model of Algorithm Performance for Continuous Optimization Problems," *International Conference on Parallel Problem Solving from Nature*, vol. 3242, pp. 226—235, 2012.
7. B. L. Saux and M. Sanfourche, "Robust vehicle categorization from aerial images by 3D-template matching and multiple classifier system," *Image and Signal Processing and Analysis (ISPA)*, pp. 466–470, 2011.
8. S. Hinz and U. Stilla, "Car Detection in Aerial Thermal Imagery by Local and Global Evidence Accumulation," *Pattern Recognition Letters*, vol. 27, no. 4, pp. 308—315, 2006.
9. G. Hummel, D. Smirnov, A. Kronenberg, and P. Stütz, "Prototyping and training of computer vision algorithms in a synthetic UAV mission test bed," in *AIAA SciTech 2014*, 2014, pp. 1–10.
10. G. Cheng, J. Han, L. Guo, X. Qian, P. Zhou, X. Yao, and X. Hu, "Object detection in remote sensing imagery using a discriminatively trained mixture model," *ISPRS*, vol. 85, pp. 32–43, 2013.
11. R. Collins, X. Zhou, and S. Teh, "An open source tracking testbed and evaluation web site," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 17–24, 2005.
12. S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery : A small target detection benchmark," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203, 2016.
13. M. Schmitt and P. Stuetz, "Perception-Oriented Cooperation for Multiple UAVs in a Perception Management Framework: System Concept and First Results," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016.

A kalman filter with state constraints for model-based dynamic facial action unit estimation

Teena Hassan¹, Dominik Seuß¹, Andreas Ernst¹, and Jens Garbas¹

Fraunhofer Institute for Integrated Circuits IIS,
Am Wolfsmantel 33, 91058 Erlangen

Abstract This paper describes a dynamic, model-based approach for estimating intensities of 22 out of 44 different basic facial muscle movements. These movements are defined as Action Units (AU) in the Facial Action Coding System (FACS) [1]. The maximum facial shape deformations that can be caused by the 22 AUs are represented as vectors in an anatomically based, deformable, point-based face model. The amount of deformation along these vectors represent the AU intensities, and its valid range is $[0, 1]$. An Extended Kalman Filter (EKF) with state constraints is used to estimate the AU intensities. The focus of this paper is on the modeling of constraints in order to impose the anatomically valid AU intensity range of $[0, 1]$. Two *process models* are considered, namely constant velocity and driven mass-spring-damper. The results show the temporal smoothing and disambiguation effect of the constrained EKF approach, when compared to the frame-by-frame model fitting approach ‘Regularized Landmark Mean-Shift (RLMS)’ [2]. This effect led to more than 35% increase in performance on a database of posed facial expressions.

Keywords Kalman filter, state constraints, action unit recognition, facial expression analysis.

1 Introduction

Facial expressions are one of the key modalities through which emotions are expressed by humans [3]. Psychologists and behavioural scientists, who study emotions, use FACS to encode facial expressions

in videos and images in terms of the observed AUs and their intensities. However, manual annotation of a 1-minute long facial expression video often takes more than an hour. Moreover, becoming a certified FACS coder requires paid training involving 50-100 hours [4]. Automatic recognition of AUs would therefore accelerate emotion and behavioural research by making it feasible to annotate large volumes of facial videos in less time. Automatic facial expression analysis based on FACS would also promote applications in the fields of market research, gaming, social robotics, and healthcare, where a broad range of subtle and spontaneous facial expressions come into play.

Psychologists identify AUs in video segments based on the facial muscles (observable as facial features or regions) that moved, the direction of the movement (shape), and the characteristic transient textural patterns (appearance) that are displayed [1]. The identification of AUs are therefore based on spatial (shape and appearance) and temporal aspects¹. Most of the research on automatic AU recognition have employed data-driven, supervised machine learning approaches with hand-crafted or self-learned spatial or spatiotemporal features [5]. Data-driven approaches enable learning from high amounts of data, but provide limited possibilities to incorporate domain knowledge. Moreover, interpretation and comparison of the decisions are either not possible or are difficult. In contrast, model-based approaches (e.g. [6]) enable incorporation of domain knowledge about the anatomy of the face and facial muscles. They also enable the definition of an anatomically valid range of AU intensities. Consequently, the intensities can be directly interpreted and compared across different facial expression videos.

In [7], we presented a dynamic, model-based approach for estimating intensities of 22 AUs, which were identified by psychologists as the most important for emotion research. The approach uses an Extended Kalman Filter (EKF) to combine an anatomically-inspired temporal model of facial muscle motion with an anatomically-based AU shape model. In [7], the focus was laid on the fusion of shape and appearance based observations of AUs within the EKF framework. In this paper, we describe the modeling and working of constraints on the estimated AU intensities to ensure that they adhere to the anatomically

¹ For example, AU43 (eyes closed) and AU45 (eye blink) are distinguished by observers based on the temporal characteristics [1].

valid range defined by the AU shape model.

2 Related work

Model-based approaches (e.g. [7], [8], [6]) for automatic AU intensity estimation use a parameterized, point-based face model that describes how a mean neutral face is deformed when different AUs are displayed. The CANDIDE-3 face model [9] is one such model that has been used for AU intensity estimation. It consists of 113 facial landmarks, 11 animation units, and 12 person-dependent, hand-crafted facial Shape Units (SU). The model-based approaches use Kalman filter, particle filter, or their variants to estimate AU intensities that are modelled as animation unit parameters of the CANDIDE-3 model. Dong et al. [8] used an EKF with Gaussian random walk as *process model*, and ten facial feature point positions as observation, to estimate head pose, one animation unit and nine SU parameters. All facial feature points were configured with identical noise. Dornaika and Davoine [10] used a particle filter based framework with adaptive transition and observation models to track the head pose and six animation unit parameters. In [6], Dornaika and Davoine proposed a two-stage approach involving particle filter to estimate six animation unit parameters and seven prototypic expressions of emotions. Empirically-determined Gaussian auto-regressive process models described the temporal changes in animation unit parameters, for each expression. Other works (e.g. [11], [12], [13]) used Kalman or particle filters to estimate different parameters of statistical face models.

Most of the works neither defined nor imposed any range validity checks on the estimated parameters. Statistical face models usually define a standard deviation for the shape parameters. However, face models like CANDIDE-3 that encode semantic information are often not accompanied by valid range definitions for the shape deformations. The lack of valid range specifications for AU intensities makes it impossible to reliably verify and compare the values across video sequences. It also leads to ambiguities, because an AU could be estimated as another AU in the reverse direction—for example, AU01.Inner.Brow.Raiser and AU04.Brow.Lowerer are performed in nearly opposite directions. Negative estimates are, however, inconsistent with the AU definitions in

FACS. Definition and enforcement of a valid range for AU intensity parameters is therefore necessary in order to ensure semantic integrity, FACS conformance, and comparability of estimated AU intensities. In order to achieve this, we use an anatomically-based face model comprising 22 AUs and 61 SUs, which was designed by psychologists [14]. The AU deformation vectors in this model correspond to the maximum humanly possible deformations. Therefore, the valid AU intensities belong to the range $[0, 1]$. We introduce these range constraints within an EKF framework by applying one of the standard methods for enforcing state constraints in a Kalman filter. We adapt the modeling of constraints to the *process model* used. This paper discusses the modeling and working of these constraints on the estimated AU intensities.

3 Approach

Our approach uses a constrained EKF to estimate the intensities of 22 AUs, as shown in Figure 1. This enables the combination of dynamic models of facial muscle motion with static observations of facial shape, which is defined by the positions of 68 facial landmarks². In the *predict* step, EKF applies a dynamic *process model* to predict the *a priori* estimates based on the *aposteriori* estimates of the previous frame. The dynamics of AUs can be represented using a Constant Velocity (CVel) model or a driven Mass-Spring-Damper (MSD) model. In the *correct* step, EKF updates the predicted *a priori* estimates based on the detected facial landmark positions, while enforcing the range constraints on the AU intensity estimates. Facial landmark positions are detected by fitting a statistical face model using the RLMS method, which is a static, iterative refinement method for model fitting. The AU intensities to be estimated are modeled as parameters of an anatomically-based, deformable facial shape model. This enables interpretability and comparability of the AU intensities estimated by our approach. Head pose and person-dependent facial shape are also estimated simultaneously, to account for global head motion and interpersonal differences in facial

² Our previous publication [7] used appearance-based AU classifiers as an additional observation. It focused on fusion of shape and appearance information in the EKF. Since the focus of this paper is on constraint modeling, for simplicity, we exclude the appearance-based observations.

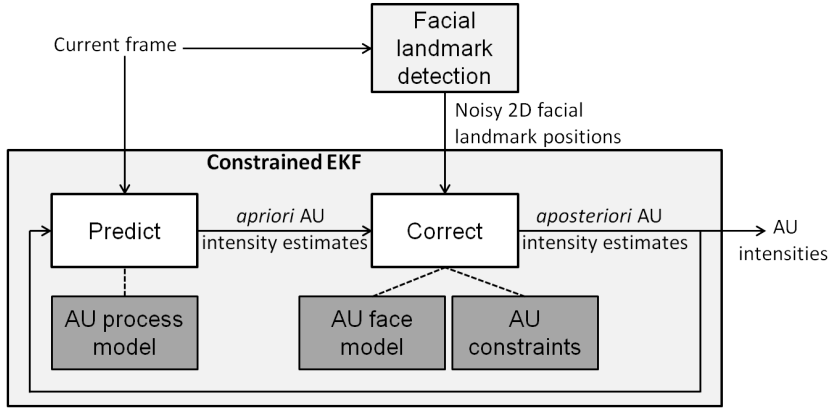


Figure 1: Overview of the constrained extended Kalman filter framework for AU intensity estimation.

shape. The noise in the *process model* is set empirically and its cumulative effect over the time interval between consecutive frames is modeled mathematically. The noise in the observed landmark positions is computed empirically on the annotated CK+ dataset [15]. The AU intensities estimated by the EKF could be thresholded to transform them into binary AU detections.

3.1 AU constraint modelling

The AU intensities in the face model have a valid range of $[0, 1]$. In order to enforce these range constraints, we adopted the method discussed in section 2.2 of [16]. The state constraints are introduced in the measurement model. The constraints are modeled as a push towards zero that is applied to each AU intensity variable in the *correct* step of the EKF. To create this push, the observation is set to zero, and a small zero-mean error variance σ^2 is attributed to it, so that the constraints are treated as soft constraints. This was done to provide the system flexibility in adapting itself to modeling errors and to avoid singularity issues. Each *apriori* AU intensity estimate x_{a_i} is linearly transformed into $x'_{a_i} = cx_{a_i} + d$. The new variable x'_{a_i} has a variance c^2 times greater than that of x_{a_i} . c , d and σ^2 are the constraint parameters. c is called

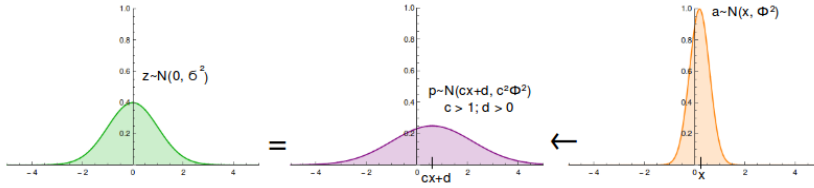


Figure 2: Modeling of soft state constraints in the measurement model. The *a priori* estimate x is linearly mapped and the observation is set to zero with a small variance σ^2 . ‘N’ denotes a normal distribution.

constraint coefficient, d is called *constraint offset* and σ^2 is called *constraint variance*. The strength of the constraint is controlled by adjusting these three constraint parameters. Figure 2 illustrates this modeling of the soft constraints in the measurement model.

The soft constraints work by controlling the amount of update applied to the *a priori* AU intensity estimates in the *correct* step. The update involves two factors, namely *Kalman gain* and *innovation*. The *Kalman gain*, which is computed as $c\phi^2/(c^2\phi^2 + \sigma^2)$, is influenced by the *constraint coefficient* c and the *constraint variance* σ^2 . The *innovation*, which is computed as $0 - cx_{a_i} - d = -cx_{a_i} - d$, is influenced by the *constraint coefficient* c and *constraint offset* d . The larger the value of c , the lower the confidence in x'_{a_i} and higher the confidence in the observation 0, given that $c > 1$. The *innovation* contributed by the constraint increases in magnitude as c increases. Thus, higher values of c have the overall effect of increasing the magnitude of the constraint-based update applied to the *a priori* AU intensity estimates. In other words, the greater the value of c , the greater is the push towards the observation 0. The direction of the push or the update is determined by the sign of the *innovation*. Given that c and d are positive, the computed *innovations* are negative, when $x_{a_i} > -d/c$, and positive, when $x_{a_i} < -d/c$. This is illustrated in Figure 3. As can be seen, $x_{a_i} = -d/c$ defines the *zero-line of operation* of the soft constraint. The positive AU intensity estimates x_{a_i} are reduced in value by the negative *innovations*, and the negative AU estimates x_{a_i} that are $< -d/c$ are increased in value by the positive *innovations*.

Now, let's look at the effect of *constraint variance* σ^2 and *constraint off-*

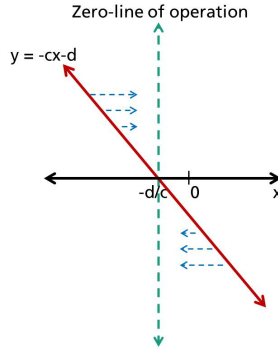


Figure 3: The working of soft constraints with $c > 1$ and $d > 0$. $y = -cx - d$ is the innovation contributed by the constraints. The zero-line of operation is defined by $x = -d/c$. The soft constraints create a force that pushes the *a priori* estimate x to this line.

set d . The lower the value of σ^2 , the higher the *Kalman gain*. The higher the *Kalman gain*, the greater the amount of update that is applied to the *a priori* AU intensity estimate. In other words, the trust in the observation 0 is greater. The contribution of *constraint offset* d is towards defining the *zero-line of operation* of the soft constraint as shown in Figure 3. This line determines when the *innovation* changes sign or equivalently, when the constraint changes the direction of operation. It also determines the range of negative AU intensity estimates that are tolerated, which is $(0, -d/c]$. Therefore, the higher the value of d , the farther the *zero-line of operation* from actual 0, or equivalently, the larger the band of negative AU intensity estimates that are tolerated, or the softer the constraint. The *constraint coefficient* also influences the location of the *zero-line of operation*. Being in the denominator, the higher the value of c , the smaller the band of negative AU intensity estimates that are tolerated, or equivalently, the harder the constraint.

Negative AU intensity estimates are not FACS-conform, since they represent activation of the AUs in the reverse direction. Therefore, negative estimates should be suppressed more strongly than non-negative estimates. In order to achieve this, the *constraint coefficient* c that is used for the negative estimates is one order of magintude higher than that

used for the non-negative estimates. The *constraint offset* d and *constraint variance* σ^2 for negative and non-negative AU intensity estimates are set identically. Moreover, the same constraint parameter configuration is used for all 22 AU intensities. The final values used for the constraint parameters c , d , and σ^2 were determined experimentally.

Depending on the used *process model*, soft constraints may need to be enforced on additional state variables in order to be consistent with the definition of valid AU intensities. If a CVel model is used, it is sufficient to apply soft constraints on the AU intensity estimates. If a driven MSD is used, it is necessary to apply soft constraints also to the *driving force* F_{a_i} , so that its direction of application on the muscle(s) is along the direction of the AU deformation vector. In the driven MSD model, each facial muscle or muscle group that causes an AU is modeled as a mass attached to a damped spring. The AUs are activated by applying external force to displace these springs. To ensure positive AU intensity estimates, the spring should be displaced in the positive direction of the AU deformation vector. To achieve this, the driving force should be positive. The soft constraints therefore suppress the negative range strongly, whereas the non-negative range is suppressed only lightly. The values of the constraint parameters for the *driving force* are not necessarily identical to those for the AU intensities. The constraint parameters identified experimentally are listed in Table 1.

Process Models	Parameter	c	d	σ^2
CVel, driven MSD	$x_{a_i} \geq 0$	2	0.2	1
CVel, driven MSD	$x_{a_i} < 0$	20	0.2	1
Driven MSD	$F_{a_i} \geq 0$	0.1	0	1
Driven MSD	$F_{a_i} < 0$	20	0	1

Table 1: Constraint parameters determined experimentally for different *process models*. Note: The constraints on F_{a_i} effectively work on the acceleration caused by the force, since mass is constant.

4 Experiments and results

In this section, the qualitative and quantitative results of constraint modeling and its effects are presented. For this, an internal, yet to be

published dataset of AUs posed by male and female Caucasian actors was used. From this dataset, 431 sequences from 20 actors were used to compare the performance of the proposed constrained EKF approach with the frame-by-frame RLMS approach. The EKF used CVel *process model*. The RLMS method estimated AU intensities by fitting the AU face model. An extended set of 740 sequences was later used to examine the performance of the driven MSD *process model*.

Figure 4 shows the qualitative impact of the constraint coefficient c on negative as well as non-negative AU intensity estimates produced by the constrained EKF approach with CVel model. It can be seen that higher values for c created a stronger push towards zero. One order of magnitude higher values had to be used for c for the negative range than for the non-negative range, so that the negative AU estimates are sufficiently suppressed and brought closer to zero. Table 2 shows the distribution of the AU intensity parameters estimated by the proposed constrained EKF method and the frame-by-frame RLMS approach. It can be seen that the soft state constraints in the proposed approach considerably reduced the number of AU intensity estimates that were outside the valid range of $[0, 1]$ defined by the AU face model.

The proposed approach allows the modeling of the dynamics of AUs, which leads to temporal smoothing of the produced AU intensity estimates. This is visible in Figure 5. The estimates from the frame-by-frame approach is agnostic to the temporal characteristics of AU intensity changes. Therefore, the estimates produced by it are very noisy. The driven MSD model produced smoother estimates than the CVel model, since it models the viscoelastic properties of facial muscles.

The modeling of temporal aspects and the enforcement of range constraints contributed towards resolving ambiguities in the spatially correlated AUs. This disambiguation effect resulted in improvement in overall performance over the RLMS approach. The overall performance is computed as the mean (in percentage) of the area under receiver operating characteristic curves for the 15 AUs that were annotated in the 431 sequences. RLMS achieved 52.63%, and CVel achieved 71.35%, which corresponds to an improvement of 35.57% over RLMS. The overall performance of the constrained EKF with the driven MSD model was 67.18% on the extended set of 740 sequences covering 22 AUs.

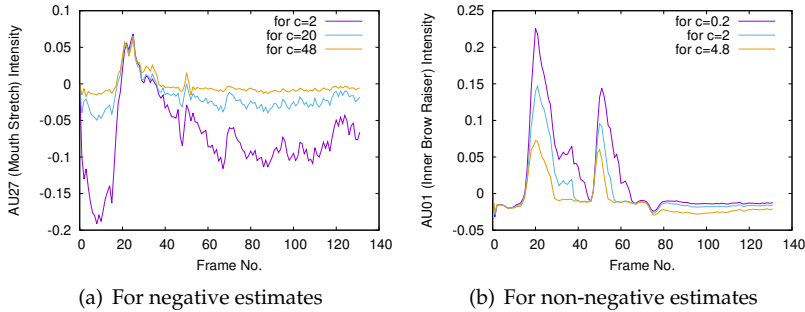


Figure 4: The effect of constraint parameter c on AU intensity estimates. The larger the value of c , the stronger the constraint. This is observable in Subfigure (a) as an upward push on negative estimates, and in Subfigure (b) as a downward push on non-negative estimates.

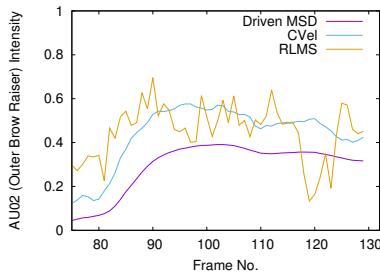


Figure 5: Temporal smoothing achieved by the proposed constrained EKF approach using two different *process models*.

5 Conclusions

In this work, we illustrated the modeling of range constraints on FACS AU intensities estimated by an EKF-based approach. The range constraints were defined by an anatomically based face model. The constraints were introduced in the *correct* step of the EKF, and created the effect of pushing the estimates towards zero. Additional parameters were also constrained depending on the *process model* used. The results showed that values farther away from the valid range of $[0, 1]$ were sig-

Approach	< -0.2	$[-0.2, 1.2)$	≥ 1.2
RLMS	25.87	73.61	0.52
Constrained EKF: CVel	0.0035	99.95	0.049
Constrained EKF: Driven MSD*	0.002	99.95	0.046

Table 2: Distribution of AU intensity estimates produced by RLMS approach and the proposed constrained EKF approach. The values are in percentage.*The results for MSD were computed on an extended set of 740 sequences.

nificantly suppressed. The disambiguation effect improved the overall performance of the constrained EKF compared to the model-fitting approach of RLMS.

References

1. P. Ekman, W. V. Friesen, and J. C. Hager, *The Facial Action Coding System*. Salt Lake City, UT: Research Nexus eBook, 2002.
2. J. M. Saragih, S. Lucey, and J. F. Cohn, "Deformable model fitting by regularized landmark mean-shift," *International Journal of Computer Vision*, vol. 91, no. 2, pp. 200–215, 2011.
3. C. Darwin, *The expression of the emotions in man and animals*. Oxford University Press, 1998.
4. P. E. Group, "Facial Action Coding System," <https://www.paulekman.com/product-category/facs/>, [Online; accessed 14-Sep-2018].
5. E. Sariyanidi, H. Gunes, and A. Cavallaro, "Automatic analysis of facial affect: A survey of registration, representation, and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1113–1133, 2015.
6. F. Dornaika and F. Davoine, "Simultaneous facial action tracking and expression recognition in the presence of head motion," *International Journal of Computer Vision*, vol. 76, no. 3, pp. 257–281, 2008.
7. T. Hassan, D. Seuss, J. Wollenberg, J. Garbas, and U. Schmid, "A practical approach to fuse shape and appearance information in a gaussian facial action estimation framework," in *ECAI 2016*, ser. Frontiers in Artificial Intelligence and Applications. Amsterdam: IOS Press, 2016, pp. 1812–1817.
8. Y. Dong, Z. Hu, Y. Zhou, K. Uchimura, and N. Murayama, "A robust and efficient face tracker for driver inattention monitoring system," in *Intelligent*

Control and Automation (WCICA), 2011 9th World Congress on, June 2011, pp. 1212–1217.

9. J. Ahlberg, "CANDIDE-3 – an updated parameterized face," Department of Electrical Engineering, Linköping University, Sweden, Tech. Rep. LiTH-ISY-R-2326, 2001.
10. F. Dornaika and F. Davoine, "On appearance based face and facial action tracking," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, no. 9, pp. 1107–1124, 2006.
11. A. M. Baumberg and D. C. Hogg, "An efficient method for contour tracking using active shape models," in *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, Nov 1994, pp. 194–199.
12. M. D. Cordea, E. M. Petriu, and D. C. Petriu, "Three-dimensional head tracking and facial expression recovery using an anthropometric muscle-based active appearance model," *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, no. 8, pp. 1578–1588, Aug 2008.
13. U. Prabhu, K. Seshadri, and M. Savvides, "Automatic facial landmark tracking in video sequences using Kalman filter assisted active shape models," in *Trends and Topics in Computer Vision*, ser. Lecture Notes in Computer Science, K. Kutulakos, Ed. Springer Berlin Heidelberg, 2012, vol. 6553, pp. 86–99. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-35749-7_7
14. E. G. Krumhuber, L. Tamarit, E. B. Roesch, and K. R. Scherer, "FACSGen 2.0 animation software: Generating three-dimensional FACS-valid facial expressions for emotion research," *Emotion*, vol. 12, no. 2, pp. 351–363, 2012.
15. P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, 2010, pp. 94–101.
16. D. Simon, "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms," *Control Theory Applications, IET*, vol. 4, no. 8, pp. 1303–1318, 2010.

Monokulare Kopfposenschätzung

Monocular head pose estimation

Benjamin Jäschke und Fernando Puente León

Karlsruher Institut für Technologie,
Institut für Industrielle Informationstechnik,
Hertzstr. 16, 76187 Karlsruhe

Zusammenfassung Es wird ein Verfahren zur Verfolgung der Kopfpose vorgestellt, welches die aus der photometrischen Stereometrie gewonnenen Oberflächennormalen für den Gesichtsbereich nutzt. Insbesondere wird hierbei auf die Wahl eines geeigneten Trackinggebiets sowie die Ableitung der Kopfposenänderung aus Normalenvektoren eingegangen. Der Vorteil dieses Verfahrens, welches mit mehr Aufwand verbunden ist als 2D-Verfahren, ist die leistungsfähigere Verfolgung der Pose bei starken Rotationsbewegungen.

Schlagwörter Kopfposenschätzung, Tracking, photometrisches Stereo.

Abstract We present an approach to head pose tracking by using surface normals which are gathered from photometric stereo. In particular, we focus on choosing an appropriate area for tracking and deriving the postural change from normal vectors. Despite causing more computational cost, the method presented is advantageous for tracking the head pose in case of large rotational movements.

Keywords Head pose estimation, tracking, photometric stereo.

1 Einleitung

Die Schätzung der Kopfpose ist für eine Vielzahl von Anwendungen von Interesse, bei denen eine Interaktion zwischen Mensch und

Maschine erfolgt. Oftmals wird die Kopfposenschätzung auch als Teillösung für das Problem der Blickrichtungsschätzung benötigt, bei dem zusätzlich die Augenbewegungen bestimmt werden müssen. Im hier beschriebenen Ansatz zur Kopfposenschätzung sollen lediglich Bildaufnahmen des Kopfes herangezogen werden, ohne die beobachtete Umgebung mit Markern oder anderen Referenzpunkten zu beeinflussen. Aufgrund von Mehrdeutigkeitsproblemen bei 2D-Verfahren wird in diesem Artikel ein Ansatz verwendet, der 3D-Information erfassen und berücksichtigen kann, sodass Mehrdeutigkeiten reduziert werden können.

Die nachfolgenden Abschnitte behandeln verwandte Methoden aus der Literatur, den Ansatz der photometrischen Stereometrie, die Ableitung der Kopfpose aus den gewonnenen Normalen, eine Vorstellung erster Ergebnisse, sowie die Planung für künftige Arbeiten.

2 Stand der Technik

In der Literatur wird eine Vielzahl verschiedener Ansätze zur Kopfposenschätzung beschrieben. Die Mehrheit dieser Ansätze nutzt lediglich 2D-Information. Es kann eine grobe Einteilung dieser Verfahren in pixelbasierte Methoden und merkmalsbasierte Methoden vorgenommen werden. Für einen umfassenden Überblick zu bekannten Verfahren der Kopfposenschätzung sei auf [1] verwiesen.

Pixelbasierte Methoden zeichnen sich dadurch aus, die als Bild aufgenommenen Pixel direkt heranzuziehen und zu vergleichen. Beispielsweise sei der optische Fluss [2] genannt, aus dem eine Ableitung der Kopfpose möglich ist. Es ergeben sich jedoch Mehrdeutigkeitsprobleme, die sich in einer schlechten Unterscheidbarkeit von Rotations- und Translationsbewegungen äußern. Diese Mehrdeutigkeiten lassen sich zwar reduzieren [3], sodass das Verfahren robuster bei starken Rotationen wird, jedoch lässt sich das Problem der Mehrdeutigkeiten nicht vollständig lösen.

Merkmalsbasierte Methoden beruhen auf der Erkennung signifikanter Punkte des Gesichts und nutzen diese zur Berechnung einer Kopfpose. Ein typischer Vertreter eines solchen Ansatzes sind die Active-Appearance-Modelle [4]. Auch hier zeichnet sich jedoch ab, dass die Auswertung von Abständen zwischen Merkmalen unter Mehrdeutig-

keiten leiden. Zudem müssen sämtliche Merkmale zunächst erkannt werden, was speziell bei starken Rotationen mit Problemen verbunden ist, da stets Merkmale durch Verdeckung das Bild verlassen oder neu hinzukommen.

Sämtliche dieser Verfahren bringen den Nachteil mit sich, die Kopfpose aus Daten abzuleiten, die einer Projektion unterzogen worden sind. Die fehlende Tiefeninformation führt bei pixelbasierten Ansätzen zu den im vorherigen Abschnitt beschriebenen Mehrdeutigkeiten. Bei merkmalsbasierten Verfahren führt die fehlende Tiefeninformation zur Verfälschung von Nachbarschaftsbeziehungen zwischen Merkmalen. Die veränderlichen Abstände zwischen einzelnen Merkmalen können zwar in beschränktem Rahmen zur Schätzung der Kopfpose genutzt werden, jedoch wird es zwangsläufig bei starken Rotationen zu Problemen kommen, die sich aus der Verdeckung und dem veränderten Blickwinkel auf die Merkmale ergeben. Besonders deutlich wird dies bei Active-Appearance-Modellen im Wangenbereich: Wird ein Merkmal am „Rand des Gesichts“ aus frontaler Ansicht festgelegt, so ist bei Drehung des Kopfes dieses Merkmal zunächst wiederzufinden und anschließend mit dem Merkmal vor der Kopfdrehung zu vergleichen. Bereits die Festlegung des Merkmalsorts im Wangenbereich ist in diesem Fall mit einer großen Unsicherheit behaftet und wird anschließend in die Posenberechnung eingehen.

3 Photometrisches Stereo

Als Vorbetrachtung sollen zunächst kurz die Grundlagen der photometrischen Stereometrie vorgestellt werden. Bei diesem Verfahren werden mehrere Aufnahmen unter unterschiedlichen Beleuchtungsbedingungen genutzt und hieraus die Normalenrichtungen der Oberfläche berechnet. Es werden folgende Annahmen getroffen:

- Es liegt eine Lambert'sche Oberfläche vor [5].
- Die Beleuchtungsrichtung für jede der Aufnahmen muss bekannt sein. Die Beleuchtungsrichtung wird als Vektor $\mathbf{l}_k$ bezeichnet und beschreibt die Richtung, unter der sich von der zu untersuchenden Oberfläche aus betrachtet die k -te punktförmige Lichtquelle befindet.

Sind diese Annahmen erfüllt, so ergibt sich für den hier betrachteten Fall mit vier Beleuchtungsrichtungen der folgende Zusammenhang zwischen den Intensitäten $i_k(\mathbf{x})$ an den Pixelorten $\mathbf{x} = (x, y)^\top$ und dem Oberflächennormalenvektor $\mathbf{n}(\mathbf{x})$:

$$\begin{pmatrix} i_1(\mathbf{x}) \\ i_2(\mathbf{x}) \\ i_3(\mathbf{x}) \\ i_4(\mathbf{x}) \end{pmatrix} = \rho(\mathbf{x}) \begin{pmatrix} \mathbf{l}_1^\top \\ \mathbf{l}_2^\top \\ \mathbf{l}_3^\top \\ \mathbf{l}_4^\top \end{pmatrix} \mathbf{n}(\mathbf{x}). \quad (1)$$

Hierbei beschreibt die Albedo $\rho(\mathbf{x})$ das Rückstrahlvermögen der Oberfläche, welches neben der Oberflächenausrichtung die Intensitäten $i_k(\mathbf{x})$ beeinflusst. Dies ist jedoch unproblematisch, da die Normalenvektoren die Länge 1 aufweisen, sodass sich sowohl die skalaren Werte $\rho(\mathbf{x})$, als auch die Vektoren $\mathbf{n}(\mathbf{x})$ berechnen lassen. Dieses Verfahren lässt sich erfolgreich auf das Gesicht anwenden [6]. Im Anschluss an die Normalenberechnung erfolgt üblicherweise eine Rekonstruktion der Oberfläche. Dieser Schritt ist für das hier vorgestellte Verfahren nicht nötig, da die Normalen bereits die gewünschte 3D-Information über den Kopf liefern.

4 Ableitung der Kopfpose

Es sollen vorab die Bezeichnungen der Rotationen eingeführt werden, auf die in den folgenden Abschnitten zurückgegriffen wird. Bei Festlegung der Koordinatenrichtungen gemäß Abbildung 1 wird eine Rotation um die x -Achse als „Nicken“, eine Rotation um die y -Achse als „Gieren“ und eine Rotation um die z -Achse als „Rollen“ bezeichnet.

4.1 Vorauswahl Trackingregion

Die Anwendung der photometrischen Stereometrie liefert Normalenvektoren für jedes Pixel. Die Ableitung der Kopfpose hieraus erfordert folglich zunächst eine Auswahl derjenigen Normalenvektoren, die zum Tracking geeignet sind. Dies geschieht zunächst in Form einer Vorbeurteilung, die nur den Zweck hat, zum Tracking verwendbare Gesichtsbereiche zu finden. Das Verfahren selbst entscheidet im Rahmen des

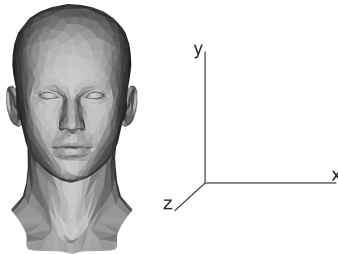


Abbildung 1: Festlegung der Koordinatenrichtungen.

Trackings, welche Gesichtsregion und welche Normalen tatsächlich verwendet werden. Beide Aspekte werden nachfolgend im Detail erläutert.

Für die Auswahl eines Gesichtsbereichs zum Tracking müssen vor allem die eventuell auftretenden Schwierigkeiten betrachtet werden, die bei der Anwendung der photometrischen Stereometrie oder dem Tracking entstehen können. Die photometrische Stereometrie nutzt Intensitäten zur Berechnung der Oberflächenausrichtung. Schattenwurf ist hierbei problematisch, da dieser gleichbedeutend damit ist, dass der betroffene Bildbereich nicht von der aktiven Beleuchtung erreicht wird. Die Folge ist eine Fehlinterpretation der Intensitäten und folglich eine falsche Normalenrichtung. Des Weiteren ist es möglich, dass die Forderung nach einer Lambert'schen Oberfläche verletzt wird. Dies ist insbesondere bei Spiegelungen der Fall, die im Bereich der Augen auftreten. Zudem sei im Sinne der Vorauswahl geeigneter Gesichtsregionen die Stirn als problematischer Bereich erwähnt, da hier Spiegelungen auftreten können, falls die beobachtete Person schwitzt.

Ferner kann eine Vorbetrachtung bezüglich der Geometrie und der Ausrichtung der zu trackenden Fläche erfolgen. Da Rotationen ebenso erfasst werden sollen wie Translationen, sind rotationssymmetrische Teilbereiche ungeeignet, was speziell die Nasenspitze betrifft. Für die Ausrichtung sind Flächen vorzuziehen, die in Richtung der Kamera – und folglich in Richtung der Lichtquellen – ausgerichtet sind. Dies bringt den Vorteil mit sich, dass beim Tracking von Rotationsbewegungen sichergestellt ist, dass auch nach der Rotation die Fläche noch von allen Lichtquellen beleuchtet werden kann. Während größerer Rotationsbewegungen ist dies nicht mit einer einzigen Gesichtsregion

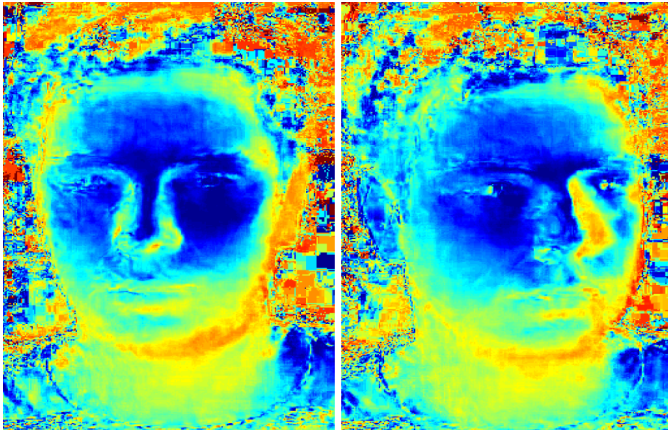


Abbildung 2: Exemplarische Darstellung zweier Nutzbarkeitskarten.

erfüllbar, weshalb das Trackinggebiet fortwährend angepasst werden muss, was in Abschnitt 4.3 näher erläutert wird.

Unter Betrachtung der beschriebenen Anforderungen lassen sich Nutzbarkeitskarten erstellen, wie in Abbildung 2 dargestellt. Hier sind diejenigen Bereiche blau dargestellt, die sämtliche Anforderungen erfüllen. Türkisfarbene bis grüne Regionen weisen auf eine Ausrichtung der Fläche hin, die nicht in Richtung der Kamera zeigt. Gelbe bis orangefarbene Bereiche kennzeichnen Regionen, in denen Schattenwurf vorliegt und die folglich keine Anwendung der photometrischen Stereometrie erlauben. Diese Karten sind für verschiedene Kopfhaltungen berechnet worden, von denen zwei hier exemplarisch dargestellt sind.

Die Karten zeigen direkt auf, dass der Schattenwurf durch die Nase die hintere Wange als Trackingregion nahezu unbrauchbar macht. Es bleiben letztlich die der Kamera zugewandte Wange und die Stirn als größere zusammenhängende Flächen, die sich zum Tracking eignen. Wie oben erwähnt, ist die Stirnregion jedoch mit dem Nachteil behaftet, dass eine schwitzende Person aufgrund von Spiegelungen nicht mehr sinnvoll zu interpretierende Oberflächennormalen liefern würde. Daher werden die Wangen als beste Option betrachtet, sofern sichergestellt ist, dass die der Kamera zugewandte Wange getrackt wird. Die Einzelheiten des hierfür notwendigen Vorgehens folgen im nächsten Abschnitt.

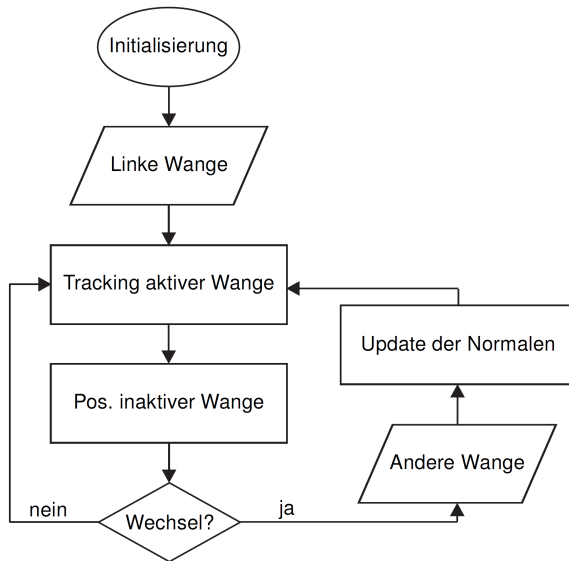


Abbildung 3: Ablaufplan des Wangenwechsel-Algorithmus.

4.2 Wangenwechsel-Algorithmus

Das Verfahren zur Auswahl der zum Tracking genutzten Wange ist in Abbildung 3 illustriert. Nach der Initialisierung, bei der mittels Gesichts- und Augendetektion die Bildregionen der beiden Wangen detektiert werden, erfolgt erstmalig eine Berechnung der Oberflächennormalen. Für die Initialisierung dürfen keine großen Rotationen des Kopfes vorliegen. Nach der Festlegung der möglichen Trackingregionen erfolgt die Auswahl der „aktiven Wange“, die tatsächlich zum Tracking genutzt wird. Diese wird zunächst auf die linke Wange festgelegt.

Anschließend beginnt das Verfahren, die Trackingschleife zu durchlaufen. Diese umfasst in jeder Iteration die folgenden Schritte:

- Das eigentliche Tracking erfolgt gemäß der im nachfolgenden Abschnitt beschriebenen Methode.
- Es wird beurteilt, ob die inaktive Wange bei der vorliegenden Pose

sichtbar ist. Ist dies der Fall, wird die entsprechende Bildregion geschätzt.

- Es wird beurteilt, ob ein Wechsel der aktiven Wange sinnvoll ist. Dies geschieht aktuell durch Festlegung eines Schwellenwertes für die Gierbewegung des Kopfes. Eine Entscheidung anhand der Qualität der Trackingregion ist derzeit in Entwicklung und soll vor allem bei größeren translatorischen Bewegungen bessere Entscheidungen bei der Wahl der aktiven Wange ermöglichen.
- Wird ein Wechsel der aktiven Wange nicht für sinnvoll erachtet, ist die Iteration beendet und das Tracking wird in der nächsten Iteration mit derselben Wange vollzogen.
- Ist ein Wechsel der aktiven Wange nötig, so erfolgt zunächst ein Update der Normalenvektoren. Dies geschieht mit Hilfe der im zweiten Schritt geschätzten Bildregion. Für diese werden die Normalen bestimmt, die in der nächsten Iteration als Referenz für die Bestimmung der Posenänderung dienen werden.

4.3 Adaption der Trackingregion

Die adaptive Veränderung des Trackingbereichs betrifft nicht die Auswahl der zum Tracking genutzten Wange, sondern bewirkt lediglich eine leichte Verschiebung der Trackingregion. Dies soll sicherstellen, dass auch bei starken Drehungen des Kopfes Normalen genutzt werden, die in Richtung der Kamera zeigen. Somit wird einerseits verhindert, dass bei weiterer Drehung das Trackinggebiet nicht mehr von allen Lichtquellen beleuchtet wird, und andererseits unterliegt die „Normalendichte“ weniger starken Veränderungen. Die Normalendichte entsteht durch die Berechnung eines Normalenvektors für jedes Pixel der aufgenommenen Bilder. Eine Gesichtsregion, die von der Kamera frontal erfasst wird, weist ein anderes Verhältnis von Pixelanzahl zu Gesichtsfläche auf, als dies bei einer nach hinten geneigten Gesichtsregion der Fall ist. Diese Pixeldichte überträgt sich direkt auf die berechneten Normalen. Da zum Zweck des Trackings ein Optimierungsproblem formuliert wird, soll anhand der Adaption des Trackingbereichs ein negativer Einfluss durch die Veränderung der Normalendichte vermieden werden.

Das Verfahren durchläuft folgende Schritte:

- Aus dem vorhergehenden Iterationsschritt ist ein Referenzgebiet mit Referenznormalen bekannt. Dieses Gebiet wird zunächst erneut genutzt und es wird überprüft, ob eine Rotation stattgefunden hat. Das Ergebnis dieses Schritts sind Winkel um die drei Rotationsachsen.
- Es folgt ein Schritt zur Kompensation des bei Kopfdrehungen auftretenden translatorischen Anteils der Gesichtsbewegung. Zugrunde liegt hier das Problem, dass die Rotationsachse des Kopfes im Bereich des Atlas liegt, bei Fehlhaltungen der beobachteten Person liegt die Rotationsachse unter Umständen sogar hinter dem Kopf. Anhand der im vorangegangenen Schritt bestimmten Winkel kann nun eine Schätzung erfolgen, wie weit sich die tatsächlich zum Tracking zu verwendende Region translatorisch bewegt hat. Das Ergebnis dieses Schritts ist eine neue Bildregion und folglich eine neue Normalenmenge.
- Mit Hilfe des neuen Bereichs erfolgt eine erneute Winkelberechnung, die nun exaktere Ergebnisse für die Rotationswinkel um alle Achsen liefert.
- Eine rein translatorische Kopfbewegung wurde zu diesem Zeitpunkt noch nicht erfasst und erfolgt durch Verschiebung des Trackingbereichs. Hierbei soll die Kosinus-Ähnlichkeit der N im Trackingbereich befindlichen Normalen $\mathbf{n}_k$ zu den Referenznormalen $\mathbf{n}_{\text{ref},k}$ maximiert werden: $\max \left(\sum_{k=1}^N \langle \mathbf{n}_k, \mathbf{n}_{\text{ref},k} \rangle \right)$.
- Nach den bisherigen Schritten ist die Kopfposenänderung bekannt. Es folgt die eingangs motivierte Adaption des Trackingbereichs, der in der nächsten Iteration als Referenz dienen wird. Zu beachten ist, dass in den meisten Fällen nur geringe Bewegungen von Frame zu Frame vorliegen. Die oben formulierten Anforderungen an den Trackingbereich im Hinblick auf die Normaldichte werden deshalb in den meisten Fällen nicht verletzt. Es wird daher ein Schwellenwert festgelegt, ab dem eine Adaption erfolgt.

5 Diskussion erster Ergebnisse

Es sei vorab erwähnt, dass durch die Nutzung der photometrischen Stereometrie eine Verwendung von etablierten Datensätzen zur Kopfpopschätzung nicht möglich ist, da diese keine Aufnahmen unter definierten Beleuchtungsrichtungen beinhalten. Zum Zweck der ersten Tests werden Datenreihen verwendet, die durch Rotationsbewegungen des Kopfes unter Vorgabe von Referenzpunkten in der Umgebung des Probanden. Anhand dieser Referenzpunkte wird die Ground Truth festgelegt. Eine größer angelegte Datenaufnahme ist geplant.

5.1 Testergebnisse

Die Testergebnisse zeigen eine gute Erfassung starker Rotationen, wie Abbildung 4 zu entnehmen ist. Es wurden hierbei in erster Linie starke Gierbewegungen des Kopfes in Kombination mit leichten Nickbewegungen getestet, was einer Beobachtung des Umfelds nachempfunden ist. Sowohl der Wechsel zwischen den beiden Wangen als auch die Anpassung der Trackingregion bei starken Drehungen erzielen die gewünschte Wirkung, sodass das Tracking auch bei Rotationen bis $\pm 80^\circ$ aufrecht erhalten werden kann. Dennoch zeigt der Ansatz Probleme, die in den nächsten Schritten gelöst werden müssen. Hier sei zunächst eine systematische Abweichung bei translatorischen Bewegungen genannt, die im nächsten Abschnitt näher erläutert wird. Zudem wird trotz der vielversprechenden Ergebnisse bei starken Gierbewegungen eine Verbesserung der Genauigkeit angestrebt. Ein Teil der Ungenauigkeit tritt durch die Kombination verschiedener Rotationen auf, als Beispiel sei hier der Fall einer starken Gierbewegung genannt, nach der zusätzlich eine Nickbewegung durchgeführt wird. Aus Sicht der Kamera wirkt diese Bewegung wie eine Rollbewegung, was an der Bindung der Bewegungsdefinition an die vom Kopf vorgegebenen Koordinatenachsen liegt.

5.2 Wölbungsproblematik

Durch die Festlegung der Beleuchtungsrichtung für die Anwendung der photometrischen Stereometrie entsteht bei geringen Abständen zwischen Kopf und Lichtquellen ein systematischer Fehler. Während

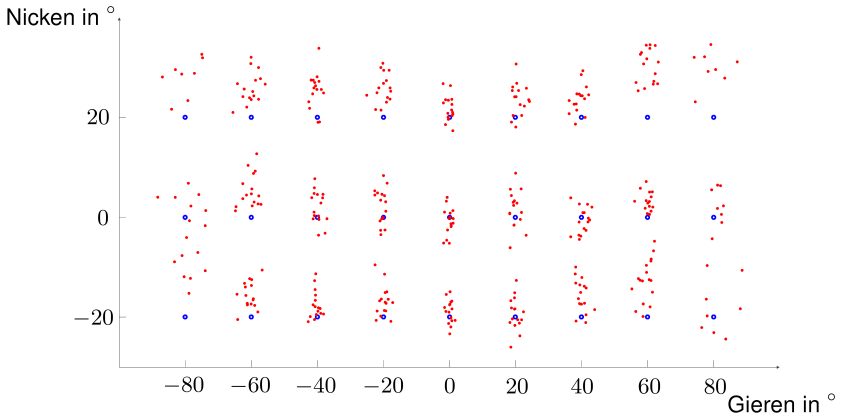


Abbildung 4: Testergebnisse (blau: Ground Truth, rot: Schätzungen).

in der Bildmitte die Beleuchtungsrichtungen korrekt sind, weichen sie mit zunehmendem Abstand zur Bildmitte mehr und mehr ab. Die Folge sind aufgrund des veränderten Einfallwinkels des Lichts geringere Intensitäten, die im Rahmen der Normalenberechnung fehlinterpretiert werden. Die sich ergebenden Normalen weisen einen Fehler auf, der sich in einem zu stark nach außen gerichteten Vektor zeigt. Die Folge hiervon wird deutlich, wenn zur Veranschaulichung eine Rekonstruktion der Oberfläche erfolgt: Sie weist eine Wölbung in der Bildmitte auf, die im Normalfall somit das Gesicht betrifft. Während mit dem in den vorherigen Abschnitten formulierten Trackingverfahren Rotationen des Kopfes von diesem Fehler nur geringfügig beeinflusst werden, verstärkt sich die Problematik bei translatorischen Bewegungen, die durch diesen Effekt von einer scheinbaren Rotation nach außen überlagert werden, selbst wenn ausschließlich translatorische Bewegungen vorliegen. Ein Verfahren zur Kompensation dieser Effekte ist derzeit in Entwicklung.

6 Zusammenfassung

Es wurde ein Ansatz zur Ableitung der Kopfpose mit Hilfe der photometrischen Stereometrie vorgestellt. Durch die Kombination mit der Auswahl der besser zum Tracking nutzbaren Wange und der Adapti-

on des Trackingbereichs können selbst starke Rotationen verfolgt werden. Die Vorteile gegenüber 2D-Verfahren sind somit deutlich erkennbar. In den nächsten Schritten wird an einer Anpassung der Normalenberechnung zur Kompensation der Lichtquellenrichtungen gearbeitet. Ferner ist die Aufnahme größerer Datensätze mit präziser Annotation der Kopfpose geplant, mit deren Hilfe auch eine vergleichende Einordnung zu anderen Verfahren erfolgen kann.

Literatur

1. E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 31, no. 4, pp. 607–626, 2009.
2. B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.
3. S. Vater, G. Mann, and F. Puente León, "A novel regularization method for optical flow based head pose estimation," in *Automated Visual Inspection and Machine Vision*, ser. Proceedings of SPIE, J. Beyerer and F. Puente, Eds., vol. 9530. Bellingham, WA: SPIE, 2015.
4. T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 681–685, 2001.
5. J. Beyerer, F. Puente León, and C. Frese, *Automatische Sichtprüfung: Grundlagen, Methoden und Praxis der Bildgewinnung und Bildauswertung*. Berlin Heidelberg: Springer, 2012.
6. M. F. Hansen, G. A. Atkinson, L. N. Smith, and M. L. Smith, "3d face reconstructions from photometric stereo using near infrared and visible light," *Computer Vision and Image Understanding*, vol. 114, no. 8, pp. 942–951, 2010.

Real-time stereo vision on FPGAs with SceneScan

Konstantin Schauwecker¹

Nerian Vision GmbH,
Gutenbergstr. 77a, 70197 Stuttgart, Germany
www.nerian.com

Abstract We present a flexible FPGA stereo vision implementation that is capable of processing up to 100 frames per second or image resolutions up to 3.4 megapixels, while consuming only 8 W of power. The implementation uses a variation of the Semi-Global Matching (SGM) algorithm, which provides superior results compared to many simpler approaches. The stereo matching results are improved significantly through a post-processing chain that operates on the computed cost cube and the disparity map. With this implementation we have created two stand-alone hardware systems for stereo vision, called SceneScan and SceneScan Pro. Both systems have been developed to market maturity and are available from Nerian Vision GmbH.

Keywords Stereo vision, depth sensing, FPGA, semi-global matching.

1 Introduction

Computer stereo vision is one of the best researched areas in the field of computer vision. Its origins date back to the 1970s, and it has since seen significant scientific advancement. Compared to other approaches for dense depth perception – such as time-of-flight cameras or structured light camera systems – stereo vision has the advantage of being a passive technology. This means that apart from the ambient light, no further active light source is required for performing measurements.

Active camera systems struggle in situations with bright ambient light such as outdoors during bright sunlight. In this case, the active

light source can no longer provide sufficient contrast from the ambient light, and hence measurements become impossible. Likewise, long-range measurements are challenging for active camera systems as this would require a particularly powerful light source. Stereo vision is consequently a promising technology for outdoor and long-range use cases, and can enable depth sensing for applications where other technologies are not applicable.

For autonomous mobile robots that need to operate outside of a controlled indoor environment, robustness towards the lighting situation is critical. Fast-moving robots, such as autonomous vehicles, must also be able to perform long-range measurements in order to operate safely. It is thus these applications that require a robust sensor solutions like real-time stereo vision.

Despite this need for robust depth sensing, there are currently only few applications that make use of stereo vision. The key problem that has prevented a wider spread adoption of this technology is the high computational demand. While many algorithms for stereo analysis have been proposed in literature within the past decades, good-performing methods are still very computationally demanding, even on modern hardware.

Traditional CPUs struggle with the vast number of computations that are required for stereo vision at typical camera frame rates. For simpler algorithms – such as block matching – real-time stereo vision can be achieved on a CPU, as has for example been demonstrated in [1]. Such simple algorithmic approaches significantly lack behind more modern methods in terms of accuracy and error robustness, as can be seen in the KITTI vision benchmark [2]. There has been some success creating optimized implementations of more sophisticated methods [3,4], but the overall computational burden remains high and the achieved processing rates and image resolutions are still comparatively low.

In order to speed-up the image processing, one requires hardware that is capable of massively parallel computations. One candidate hardware platform are graphics processors or GPUs. Numerous works in this field have demonstrated that fast real-time stereo vision is possible on GPUs with optimized algorithm implementations [5,6]. These works focused primarily on using high-end GPUs with outstanding computational capabilities. Unfortunately, such high-end GPUs also exhibit a very high-power consumption. The GeForce GTX TITAN X by NVIDIA



Figure 1: SceneScan Pro with connected Karmin2 stereo camera.

for example, requires up to 250 W of power [7]. Deploying such a power hungry processing hardware on a battery-powered mobile system is often impractical, which is why GPU-based stereo vision has not yet seen a wide spread use in mobile robotics.

An alternative hardware platform to GPUs are Field Programmable Gate Arrays (FPGAs). An FPGA is a generic integrated circuit that can be programmed to fulfill a particular application. Because programming is performed on the circuit level, an FPGA is not forced to follow the usual fetch-decode-execute cycle of CPUs or even modern GPUs. Rather, an application specific architecture can be found that divides the problem into many small sub-problems, which can each be solved in parallel and with minimal power consumption.

The disadvantage of FPGAs is that the programming effort vastly exceeds the effort of programming ordinary CPUs or GPUs. Thus, FPGA-based image processing is not yet commonplace in today's vision systems. In order to make this technology more accessible to researchers and application developers, Nerian Vision GmbH¹ developed the SceneScan and SceneScan Pro stereo vision sensor systems. The more capable SceneScan Pro system is shown in Figure 1. Using an FPGA, this small-scale processing device is able to perform stereo vision in real-time and at high processing rates. Its key features are:

- Processes input images with resolution up to 3.4 megapixels.
- Covers a disparity range of up to 256 pixels, with a sub-pixel resolution of $\frac{1}{16}$ pixel.
- Can process up to 100 frames per second.

¹ <http://nerian.com>

- Power consumption of 8 W.
- Computed disparity map is transmitted in real-time through gigabit ethernet.

SceneScan and SceneScan Pro are the successors to Nerian's earlier SP1 system [8]. Compared to the SP1, SceneScan Pro has tripled the processing performance and improved the measurement quality significantly. This article provides a review of how this has been possible and gives an insight into the achieved performance.

2 Architecture

SceneScan / SceneScan Pro connect to Nerian's own Karmin2 stereo camera or to two standard USB3 Vision cameras. It acquires a pair of synchronized left and right camera images and then processes these images on its built-in FPGA. The computed stereo disparity map, which is an inverse depth map, is then output through gigabit ethernet to a connected computer.

The stereo matching method that is implemented in SceneScan / SceneScan Pro is based on Semi Global Matching (SGM) [9]. Since its proposal, SGM has enjoyed much popularity, due to its high-quality results and its computational efficiency. SGM is at the heart of some of the best performing stereo matching methods, such as the one proposed by Žbontar et al. [10].

It has been shown that FPGA-based implementations of SGM are possible [11, 12]. However, only relying on SGM is not sufficient for receiving competitive stereo matching results. Rather, we require an entire image processing pipeline, which not only includes SGM, but also a range of different pre- and post-processing steps. We have implemented one such pipeline for SceneScan / SceneScan Pro, which is depicted in Figure 2.

2.1 Rectification

The first processing step that is applied to the input image data is image rectification. To perform image rectification, a pre-computed rectification map is required that is read from a dedicated memory. The rectifi-

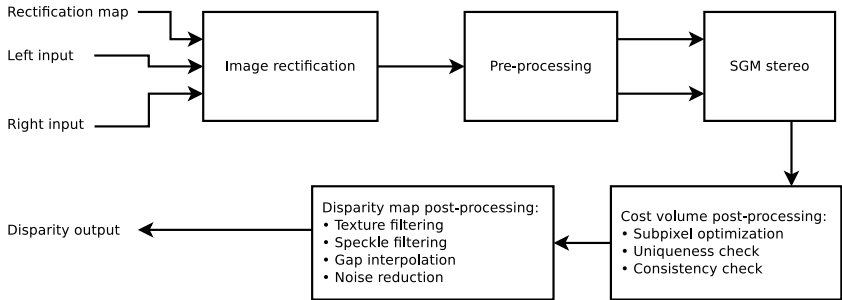


Figure 2: Block diagram of processing structure.

cation map contains a subpixel accurate x- and y-offset for each pixel of the left and right input images. Bi-linear interpolation is applied to map the subpixel offsets to image intensities.

The offsets are interleaved such that reading from a single data stream is sufficient for finding the displacement vector for each pixel in both images. To save bandwidth, the rectification map is stored in a compressed form. On average, one byte is required for encoding the displacement vector for a single pixel. Hence, the overall size of the rectification map is equal to the size of two input images.

Rectification is a window-based operation. Hence, the pixel offsets are limited by the employed window size. For SceneScan and SceneScan Pro a window size of 79×79 pixels is used. This allows for offsets in the range of -39 to $+39$ pixels.

2.2 Image pre-processing

An image pre-processing method is applied to both input images. This causes the subsequent processing steps to be more robust towards illumination variations and occlusions.

2.3 Stereo matching

Stereo matching is performed by applying a variation of the SGM algorithm [9]. The penalties P_1 and P_2 that are employed by SGM for small and large disparity variations can be configured at runtime. Several it-

erations are required for processing one pixel of the left input image. In each iteration, the left image pixel is compared to a group of pixels in the right image. For SceneScan Pro, the number of parallel pixel comparisons is $p = 32$. The number of iterations per left image pixel n_i can be configured at runtime by choosing the disparity range.

A *disparity offset* o_d can also be configured at runtime, which indicates the smallest disparity value that will be considered during stereo matching. If $o_d > 0$ then the observable depth range will have an upper limit, as disparities smaller than o_d will not be allowed. The disparity offset o_d , iteration count n_i and the parallelization p determine the maximum disparity d_{max} :

$$d_{max} = o_d + n_i p - 1 \quad (1)$$

2.4 Cost volume post-processing

The SGM stereo algorithm produces a *cost volume*, which encodes the matching costs for all valid combinations of left and right image pixels. Several of the applied post-processing techniques operate directly on this cost volume.

Subpixel optimization

Subpixel optimization is the first applied post-processing technique. This step increases the accuracy of depth measurements by evaluating the matching costs to the left and right of the detected minimum for each pixel. A curve is fitted to the matching costs and its minimum is determined with subpixel accuracy.

The improved disparity estimates are then encoded as fixed-point numbers. SceneScan / SceneScan Pro support 4 decimal bits for the subpixel optimized disparity. Hence, it is possible to measure disparities with a resolution of $1/16$ pixel.

Uniqueness check

Matches with a high matching uncertainty are discarded by imposing a uniqueness constraint. For a stereo match to be considered unique, the minimum matching cost c_{min} times a uniqueness factor $q \in [1, \infty)$ must

be smaller than the cost for the next best match. This relation can be expressed in the following formula, where C is the set of matching costs for all valid pixel pairs and $c^* = c_{min}$ is the cost for the best match:

$$c^* \cdot q < \min \{C \setminus \{c_{min}\}\}. \quad (2)$$

Stereo matches that are discarded through the uniqueness check are assigned an invalid disparity label.

Consistency check

A consistency check is employed for removing further matches with high matching uncertainties. The common approach to this post-processing technique is to repeat stereo matching in the opposite matching direction (in our case from the right image to the left image), and then only retaining matches for which

$$|d_l - d_r| \leq t_c, \quad (3)$$

where d_l is the disparity from left-to-right matching, d_r the disparity from right-to-left matching, and t_c is the consistency check threshold.

In order to save FPGA resources, we refrain from re-running stereo matching a second time in the opposite matching direction. Rather, the right camera disparity map is inferred from the matching costs that have been gathered during the initial left-to-right stereo matching. Pixels that do not pass the consistency check again receive an invalid disparity label.

2.5 Disparity map post-processing

Following the cost volume post-processing, the cost volume is reduced to a disparity map. Additional post-processing methods are then applied directly to the disparity values.

Texture filtering

Matching image regions with little to no texture is particularly challenging. Especially if such regions occur close to image borders, this might

lead to significant mismatches. In order to address this problem, a texture filter is applied. This filter computes a texture score s_t for each image pixel, which reflects the texture intensity within a local neighborhood. Pixels for which this score is below a configurable threshold t_t are again labeled as invalid in the computed disparity map.

Speckle filtering

The aforementioned methods are not always able to identify and label all erroneous matches. Fortunately, the erroneous matches that remain tend to appear as small clusters of similar disparity. These *speckles* are then removed with a speckle filter. The speckle filter identifies connected components that are below a specified minimum size. The minimum speckle size is controlled through the speckle filter window size w_s . The pixels that belong to identified speckles are again labeled as invalid.

Gap interpolation

The aforementioned post-processing techniques all remove pixels from the computed disparity map, which leaves gaps with no valid disparity data. If one such gap is small, it can be filled with valid disparities by interpolating the disparities from its edges. Interpolation is only performed for gaps whose vertical and horizontal extent l_h and l_v fulfill the condition

$$\min \{l_h, l_v\} \leq l_{max}, \quad (4)$$

where l_{max} is the maximum gap width. Interpolation is also omitted if the disparities from the edge of the identified gap do not have a similar magnitude.

Noise reduction

Finally, a noise reduction filter is applied to the generated disparity map. This filter performs a smoothing of the disparity map, while being aware of discontinuities and invalid disparities. The processing result of this filter is directly transmitted as output through SceneScan's ethernet port.

3 Results

Table 1 lists the achievable frame rates of SceneScan Pro on different image resolutions and disparity ranges. These numbers represent the frame rates at which a connected computer would receive the disparity data by SceneScan Pro, and hence already include all overheads for camera and network communication. At all configurations, SceneScan Pro requires only 8 W of power, without supplying power to the cameras.

The highest frame rate can be achieved at an image resolution of 640×480 with a 128 pixels disparity range. In this configuration SceneScan Pro can provide up to 100 fps. With increasing resolution and disparity range, the frame rate decreases. The maximum image resolution which SceneScan Pro can process is 1856×1856 , which is 3.4 megapixels.

When using the full 256 pixels disparity range, SceneScan Pro is able to perform 5.1 billion disparity evaluations per second, and it achieves an output of 20 million disparities per second. If the disparity range is reduced to 128 pixels, the processing performance increases to 30 million disparities per second. As each disparity value can be projected to a 3D coordinate, this means that SceneScan Pro provides 30 million 3D point measurements per second.

In Figure 3 we can see a collection of example input images and the corresponding disparity maps that were computed with SceneScan Pro. All images were recorded outdoors on a bright day. As stereo vision does not struggle with bright ambient light, we do receive very dense disparity maps for all examples.

Each example image contains sections with very close objects, as well as sections with distant objects. As can be seen, the distance of an object does not influence the measurement density or robustness. Even for very distant image regions, such as in the center of Figure 3(d) we do receive very dense disparity measurements.

Overall the disparity maps appear very smooth with only very few outliers. This can be credited to the extensive post-processing of the cost cube and the disparity maps. Occluded image regions are effectively filtered, as can be seen by the clear occlusion shadow to the left of all foreground objects in Figure 3.

Disparity Range	Image Resolution			
	640 × 480	800 × 592	1280 × 960	1600 × 1200
128 pixels	100 fps	65 fps	24 fps	15 fps
256 pixels	70 fps	45 fps	15 fps	10 fps

Table 1: SceneScan Pro processing performance.

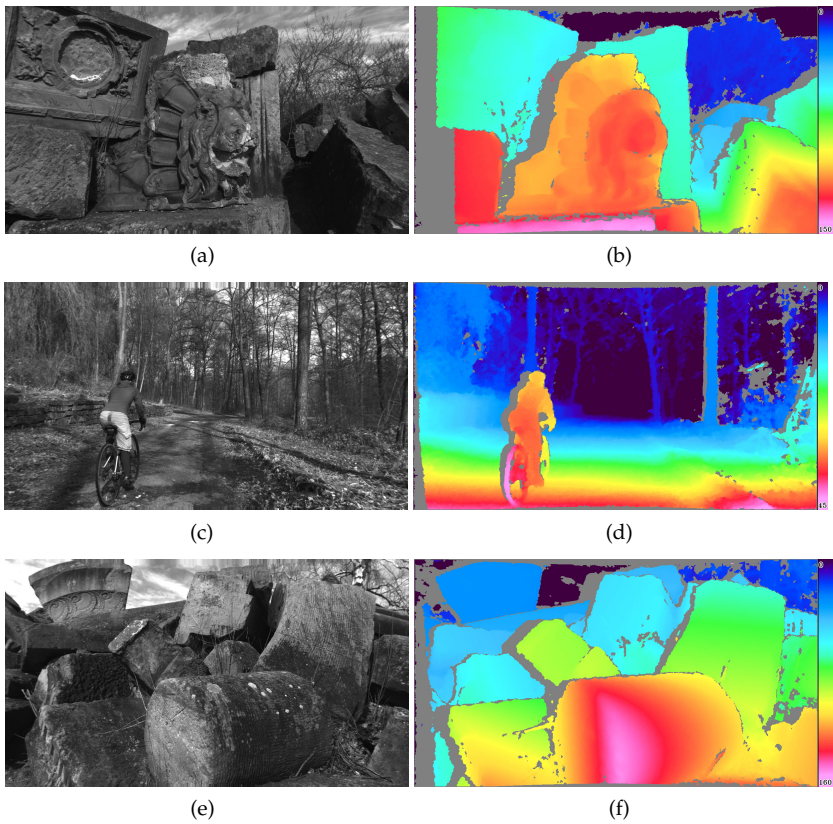


Figure 3: (a,c,e) Examples for left input images and (b,d,f) computed disparity maps.

4 Conclusion

With SceneScan and SceneScan Pro we have created two stand-alone systems for stereo vision that use FPGAs as the underlying computation hardware. By using FPGAs, our systems are able to achieve a very high-power efficiency. At the same time, they provide an exceptional computation performance, and are thus able to process camera images at frame rates up to 100 fps.

Particularly in the field of autonomous mobile robots, accurate high-speed depth estimates are a prerequisite for many applications. Stereo vision promises to provide such data even in bright environments and over long measurement distances.

Our SceneScan and SceneScan Pro systems can easily be integrated into battery-powered and energy constrained mobile platforms, where large power-hungry GPU-based systems are not an option. We hope that by introducing SceneScan and SceneScan Pro, we have made stereo vision more accessible to a wide range of researchers and developers. This will hopefully facilitate the development of more systems that make use of the principles of stereo vision.

References

1. M. Humenberger, C. Zinner, M. Weber, W. Kubinger, and M. Vincze, "A fast stereo matching algorithm suitable for embedded real-time systems," *Computer Vision and Image Understanding*, vol. 114, no. 11, pp. 1180–1202, 2010.
2. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "The kitti vision benchmark suite – stereo evaluation 2015," http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo, 2018, accessed: 17.09.2018.
3. S. K. Gehrig and C. Rabe, "Real-time semi-global matching on the cpu," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
4. R. Spangenberg, T. Langner, S. Adfeldt, and R. Rojas, "Large scale semi-global matching on the cpu," in *IEEE Intelligent Vehicle Symposium (IV)*, 2014, pp. 195–201.
5. I. Haller and S. Nedeveschi, "GPU Optimization of the SGM Stereo Algorithm," in *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2010, pp. 197–202.

6. J. Kowalczyk, E. T. Psota, and L. C. Perez, "Real-time stereo matching on cuda using an iterative refinement method for adaptive support-weight correspondences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 1, pp. 94–104, 2013.
7. NVIDIA, "GeForce GTX TITAN X specifications," <https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x/specifications>, 2018, accessed: 19.09.2018.
8. K. Schauwecker, "SP1: Stereo vision in real time." in *MuSRobS@ IROS*, 2015, pp. 40–41.
9. H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 807–814.
10. J. Žbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
11. S. K. Gehrig, F. Eberli, and T. Meyer, "A real-time low-power stereo vision engine using semi-global matching," *Computer Vision Systems*, pp. 134–143, 2009.
12. C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch, "Real-time stereo vision system using semi-global matching disparity estimation: architecture and FPGA-implementation," in *IEEE Int. Conf. on Embedded Comput. Syst. (SAMOS)*, 2010, pp. 93–101.

Camera projector system for cost-effective 3D surface detection

Ein Kamera-Projektor-System zur kostengünstigen 3D-Oberflächenmessung

Raja sekar Shantha kumar¹, Balamoni Yadaiah¹, and Michael Kaufmann¹

Ernst-Abbe-Hochschule Jena
Carl-Zeiss-Promenade 2, 07749 Jena

Abstract In this paper the development of a low-cost alternative to high-priced 3D measurement systems is presented. Expensive system components or cutting-edge hardware solutions increase the price of measurement systems for three-dimensional (3D) optical surface topology. The purpose of this project is to build affordable optical surface topology measurement devices using recent low cost hardware and high-level open source software. The camera and the projector are controlled by a single board computer. The calibration procedure is described. At this stage the system achieves useful image acquisition times of less than 15 seconds and sufficient accuracy for many applications. Subsequent studies has to be done on 3D reconstruction of objects.

Keywords Digital Fringe Projection (DFP), 3D Surface detection, camera calibration, projector calibration, Direct Linear Transformation (DLT), 3D reconstruction, computer vision.

Zusammenfassung In diesem Beitrag wird die Entwicklung einer kostengünstigen Alternative zu hochpreisigen 3D-Messsystemen vorgestellt. Teure Systemkomponenten oder hochmoderne Hardware-Lösungen erhöhen den Preis von Messsystemen für die dreidimensionale (3D) optische Oberflächentopologie. Der Zweck dieses Projekts besteht darin, erschwingliche optische Oberflächentopologie-Messgeräte unter Verwendung von neuer kostengünstiger Hardware und High-Level-Open-Source-Software zu bauen. Die Kamera und der Projektor werden von einem Einplatinen-Computer gesteuert. Das Kalibrierungsverfahren wird beschrieben. Im derzeitigen Entwicklungsstand erreicht das System brauchbare Bilderfassungszeiten von weniger als 15 Sekunden und ausreichende Genauigkeit für viele Anwendungen. Weitere Studien zur 3D-Rekonstruktion von Objekten müssen durchgeführt werden.

Schlagwörter Digital Fringe Projection (DFP), 3D Oberflächenmessung, Kamerakalibrierung, Projektorkalibrierung, Direct Linear Transformation (DLT), 3D Rekonstruktion, Bildverarbeitung.

1 Introduction

There are numerous methods and devices for optical three-dimensional scanning of objects or surfaces available. Examples are LIDAR, Fringe Projection or stereo camera systems. Most of these methods achieve amazing resolution and accuracy with high-quality and high-priced components. At the same time, a wide application is limited by high costs. With a low-cost technological solution, many new applications would be affordable. Examples are inline surface control in quality assurance, control of completeness in manufacturing or classification of components. In these areas, inexpensive 2D image processing systems have been used for several years. However, the detection of three-dimensional object data is not or only partially possible. This paper presents a technological solution for 3D capturing of surfaces and object geometries using cost-effective components. The low costs are associated with a lower accuracy and/or higher processing time of the measurement, which is acceptable for many applications.

2 Composition of the experimental system

The components of the presented system have been selected with regards to flexibility, programmability and costs. The main components of the system are a BeagleBone Black Single board computer, a DLP LightCrafter Display 2000 (projector) made by Texas Instruments and a Logitech HD Pro C920 camera module:

- The BeagleBone Black (BBB) single board computer is a compact and open-source Linux computing platform that can be used to build complex applications that interface high-level software and low-level electronic circuits.
- The DLP LightCrafter Display 2000 is a low-cost platform enabling the use of DLP technology with the embedded host processor (BBB). Inter-Integrated Circuit Communication and a 8/16/24-bit parallel RGB video interface supports virtually any low-cost host processor
- The Logitech HD Pro C920 is a USB2.0 web cam. It is capable of recording full-HD 1080p and encoding it in real-time using H264 compression format.

Total hardware costs are less than 200 Euro. Fig. 1 shows the presented experimental system in operation projecting a fringe pattern on a surface.

The DLP LightCrafter Display 2000 has been connected to the BBB single board computer using GPIOs. The Inter-Integrated Circuit (I2C) Protocol has been used to establish communication between them. Logitech HD Pro C920 web cam has been interfaced straightforwardly using USB communication. Scripting language Python and high-level library Open CV [1, 2] were used for camera-projector synchronization.

3 Scanning procedure

For 3D detection of surfaces Digital Fringe Pattern Profilometry (DFPP) has been adapted and optimized for the components used. The individual steps of DFPP are shown in Fig. 2.

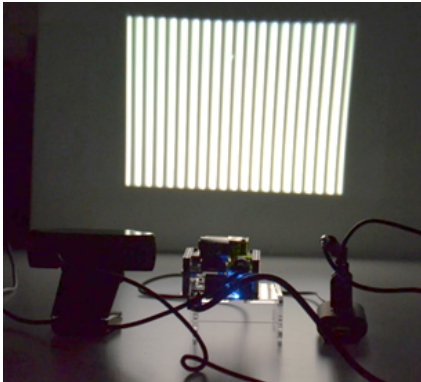


Figure 1: Overall set up including projector, camera, and single board computer in operation.

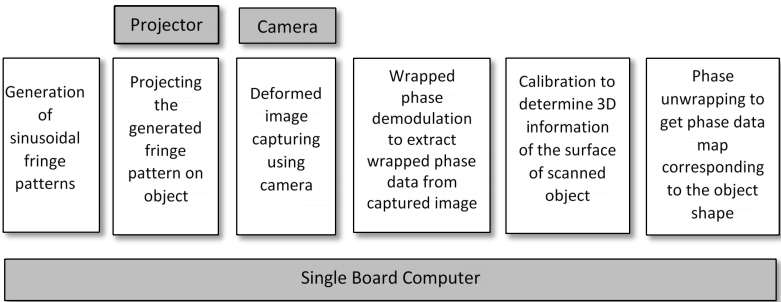


Figure 2: Adapted steps of Digital Fringe Pattern Profilometry (DFPP).

The phase shifting method of Digital Fringe Projection utilizes three sinusoidal fringe patterns which are out of phase by $2\pi/3$. These fringe patterns are computer generated using the phase shift algorithm in an open source computer vision library [1]. This algorithm is chosen because it can provide high-spatial resolution. Only three fringe patterns are projected which leads to less scanning time and less sensitivity to reflection variation [3]. The three fringe patterns are directly projected subsequently on the object surface. The fringe pattern is distorted with

respect to the shape of the object and the distorted fringe patterns are captured by the camera from a different angle [3]. Fig. 3 shows the projection of three phase shifted patterns on calibration plane using DLP LightCrafter Display 2000.

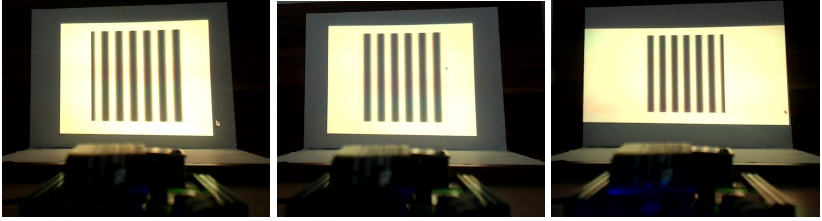


Figure 3: Sinusoidal fringe patterns out of phase by $2\pi/3$.

The generated fringe patterns are mathematically described by:

$$I(i, j) = \frac{255}{2} \left[1 + \cos \left(2\pi \cdot \frac{j}{P} + \delta \right) \right] \quad (1)$$

where P is the number of pixels per fringe period, δ is the phase shift, (i, j) is a pixel index and I is the magnitude of the pattern [3]. The three individual patterns are given by:

$$\begin{aligned} I_1(i, j) &= 255/2[1 + \cos(2\pi * j/P - 2\pi/3)] \\ I_2(i, j) &= 255/2[1 + \cos(2\pi * j/P)] \\ I_3(i, j) &= 255/2[1 + \cos(2\pi * j/P + 2\pi/3)]. \end{aligned} \quad (2)$$

Fig. 4 shows the projection of three patterns by DLP LightCrafter 2000 on an object. The image is captured by the Logitech pro C920 camera.

4 System calibration

System calibration is of great importance in 3D computer vision system to reduce the positional errors. DFP system calibration involves calibration of projector and camera. A plane based calibration method for calibrating camera-projector system has been used. The aim of the system calibration is to compute the intrinsic, extrinsic and non-linear lens distortion parameters of both the camera and the projector. In the following sections the calibration procedures are explained.

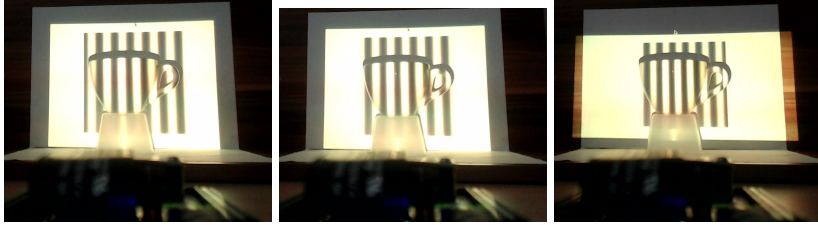


Figure 4: Sinusoidal fringe patterns out of phase by $2\pi/3$ projected on the object.

4.1 Camera calibration

To calibrate the camera, a set of correspondences between 3D world points and 2D image points has to be established. An algorithm proposed by ZHANG [4] is used to calibrate the camera because of its simplicity compared to Direct Linear Transformation (DLT) method. ZHANG's method requires a planar black and white checkerboard or circular grid which is to be placed at different orientation in front of the camera. The applied algorithm is able to find the feature points of the checkerboard (2D image points) in each image to compute a projective transformation between the image points of the n different images up to a scale factor. Intrinsic and extrinsic parameters are computed. The intrinsic camera parameters are the inner transformation of the camera including focal length, position of the principle point, skew, and the non-linear lens distortion. The extrinsic parameters are related to 3D rotation and translation of the camera. Fig. 5 shows the images of a checkerboard at different orientation. For good calibration result 10 to 20 images in different orientations are required. Fig. 6 shows the detected feature points calculated by algorithm:

$$sm' = A \begin{bmatrix} R & t \end{bmatrix} M' \quad (3)$$

or

$$s \begin{bmatrix} u \\ v \\ l \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4)$$

where: (X, Y, Z) are the coordinates of a 3D point in the world coordinate space, (u, v) are the coordinates of the projection point in pixels, A is a camera matrix or a matrix of intrinsic parameters, (c_x, c_y) , (f_x, f_y) are the focal lengths expressed in pixel units.

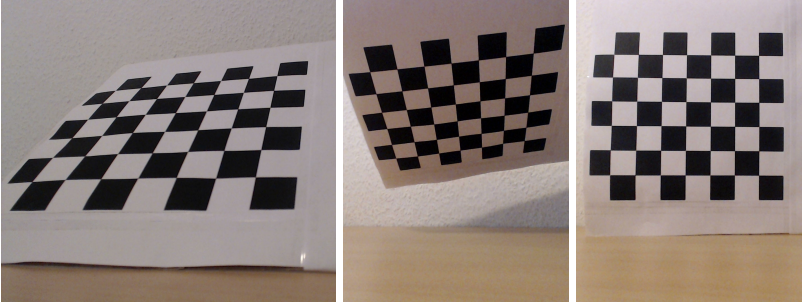


Figure 5: Black and white planar checkerboard in different orientation.

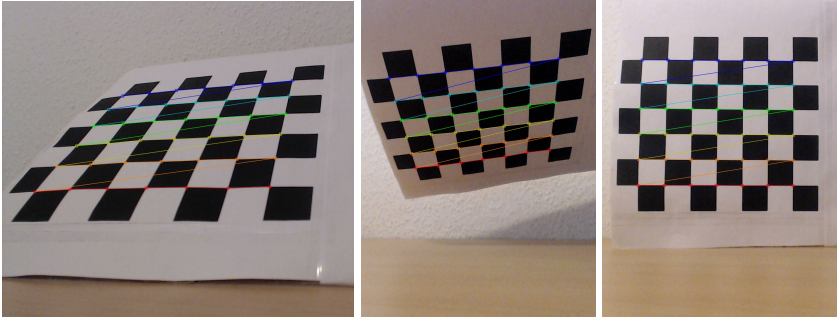


Figure 6: Detected feature points.

4.2 Projector calibration

In order to calibrate the projector, 2D points of the projected patterns from projector plane, 2D points of the projected patterns from the camera plane and 3D points of the projector are necessary [5]. To recover the 3D position of the projected pattern a ray plane intersection algorithm

is used [6]. This algorithm helps to compute the intersection between the rays and plane to obtain the 3D position of the projected corners. Plane normal and plane points are computed using external camera transformation. The result of the camera calibration gives rotation and translation vectors but no rotation matrix. To get the rotation matrix RODRIGUES' transformation is applied [7]. Fig. 7 shows the projected circular pattern for computing ray direction vector.

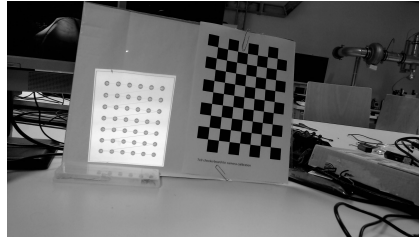


Figure 7: Projected circular grid pattern.

5 Results

The described calibration procedure for camera and projector has been executed in tests. The test results allow an assessment of the practical usability of the system. Further experiments have been performed to optimize the synchronization between the components.

5.1 Camera calibration

Using algorithm (4) the computed intrinsic parameters of the camera are:

$$\begin{pmatrix} 1.33820057 \cdot 10^3 & 0.0 & 1.02110829 \cdot 10^3 \\ 0.0 & 1.33275779 \cdot 10^3 & 5.26343178 \cdot 10^2 \\ 0.0 & 0.0 & 1.0 \end{pmatrix} \quad (5)$$

The parameters describe the internal transformation of the camera. The quality of the calibration can be evaluated by the reprojection error. The reprojection error is the distance between a pattern feature detected in

the calibration image and a corresponding world point projected into the same image. With the tested configuration the average reprojection error is $e_{rp} = 0.2801$ pixels, with a considerable dispersion. Fig. 8 shows the graphical representation of the reprojection error.

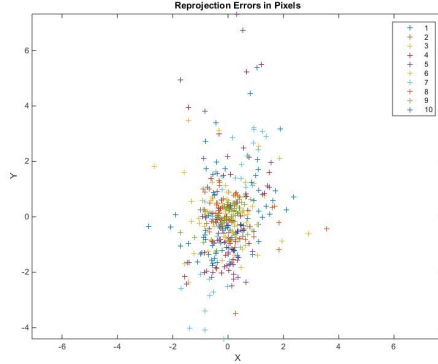


Figure 8: Reprojection error in pixels.

The radial and tangential distortion coefficients are:

$$(0.09608511 \ -0.36152505 \ 0.00087376 \ 0.00621724 \ 0.35577487) \quad (6)$$

Rotation and translation vectors are computed for each image used for calibration. The computed rotation vector for one image is

$$(0.00854413 \ -0.03395648 \ -0.01032706) \quad (7)$$

and the translation vector for one image is

$$(3.02166359 \ -4.99874661 \ 9.34334741) \quad (8)$$

The rotation and translation vectors are used to determine the positions of the checkerboard plane used for calibration. In Fig. 9 the plane positions have been visualized in a 3D space.

5.2 Projector calibration

The 3D positions of the projected patterns have been computed using ray plane intersection algorithm and camera transformation described

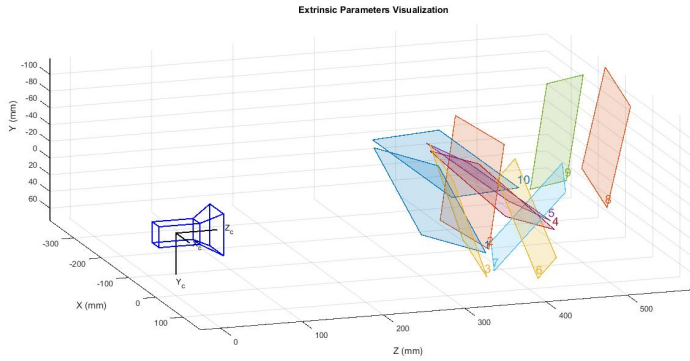


Figure 9: Determined positions of calibration plane.

in section 4.1. The computed rotation matrix from rotation vector after RODRIGUES' transformation is:

$$\begin{bmatrix} 0.99937022 & 0.01017972 & -0.03399305 \\ -0.01046981 & 0.99991018 & -0.00836692 \\ 0.03390483 & 0.00871755 & 0.99938705 \end{bmatrix} \quad (9)$$

The last row of the rotation matrix belongs to the normal plane. Using the normal plane and the translation vector the 3D position of a projected point is computed. In Fig. 10 the computed positions of plane points are visualized.

5.3 Single board computer performance

The dependency between CPU frequency, execution time and power consumption has been evaluated with regards to the development of mobile applications. The BeagleBone Black has different governors that can be used to change the performance/power usage ratio. Increasing CPU frequency will decrease the data processing time [8]. But running at higher frequency will consume more power.

The performance of BBB's CPU for all possible allowed frequencies is tested by running the synchronizing code. Tab. 1 shows the relation

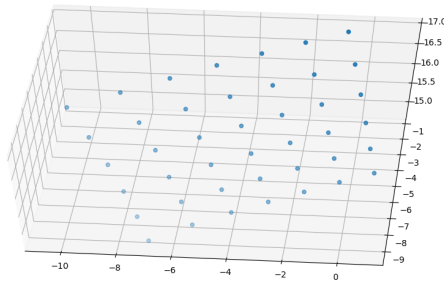


Figure 10: Computed 3D position of the projected pattern.

between the data processing time and CPU frequency. For higher frequencies the code executions time is less.

Table 1: CPU Performance of BBB.

CPU frequency	execution time
1000 MHz	12.49 s
800 MHz	15.65 s
720 MHz	18.52 s
600 MHz	23.43 s
300 MHz	36.80 s

Running the code at 300 MHz results in poor performance. A frequency between 1000 and 600 MHz has been chosen for tests.

6 Conclusion

The system composed of low-cost hardware components is working. Projection of fringe patterns on an object and capturing of images is done in less than 15 seconds. In fact, much of the processing time is consumed by the Python environment used in the single board com-

puter. It can be expected that using a compilable language like C++ will significantly reduce processing time.

Using the described algorithms, it is possible to calibrate the components and to calculate the 3D position of projected point. The intended function has been proved. Further development of the algorithms is necessary to reconstruct true 3D objects and determine their accuracy.

References

1. G. Bradski, "The OpenCV Library," *Dr. Dobbs's Journal of Software Tools*, 2000.
2. T. Petković, T. Pribanic, M. Donlić, and N. D'Apuzzo, "Software Synchronization of Projector and Camera for Structured Light 3D Body Scanning," in *7th International Conference on 3D Body Scanning Technologies*, Lugano, Switzerland, 11 2016, pp. 286–295.
3. S. Zhang, *High-Speed 3D Imaging with Digital Fringe projection Techniques (DFPP)*. 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487-2742: CRC Press Taylor & Francis Group, 2016.
4. W. Burger, "Zhang's Camera Calibration Algorithm: In-Depth Tutorial and Implementation," University of Applied Sciences Upper Austria, School of Informatics, Communications and Media, Dept. of Digital Media, Hagenberg, Austria, Tech. Rep. HGB16-05, May 2016. [Online]. Available: <http://staff.fh-hagenberg.at/burger/>
5. G. Falcao, N. Hurtos, and J. Massich, "Plane-based calibration of a projector-camera system," in *VIBOT Master*, vol. 9, 01 2008.
6. (2018) A Minimal Ray-Tracer: Rendering Simple Shapes (Sphere, Cube, Disk, Plane, etc.). [Online]. Available: <https://www.scratchapixel.com/lessons/3d-basic-rendering/minimal-ray-tracer-rendering-simple-shapes/ray-plane-and-ray-disk-intersection>
7. A. Kaehler and G. Bradski, *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*, 1st ed. O'Reilly Media, Inc., 2016.
8. D. Molloy, *Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux*. Wiley, 2014. [Online]. Available: <http://www.exploringbeaglebone.com/>

Bildverarbeitung spielt in vielen Bereichen der Technik zur schnellen und berührungslosen Datenerfassung eine Schlüsselrolle. Beispielsweise in der Qualitätssicherung industrieller Produktionsprozesse, in der Robotik und zur Fahrerassistenz haben sich Bildverarbeitungssysteme einen unverzichtbaren Platz erobert. Diese Entwicklung wird unterstützt durch die Verfügbarkeit qualitativ hochwertiger und günstiger Sensorsysteme sowie durch die Zunahme der Leistungsfähigkeit von Rechnersystemen.

Der vorliegende Tagungsband des „Forums Bildverarbeitung“, das am 29. und 30. November in Karlsruhe als gemeinsame Veranstaltung des Karlsruher Instituts für Technologie und des Fraunhofer-Instituts für Optronik, Systemtechnik und Bildauswertung stattfand, enthält die schriftlichen Aufsätze der eingegangenen Beiträge. Darin wird über aktuelle Trends und Lösungen der Bildverarbeitung in den methodischen Schwerpunkten Bildgewinnung, Computational Imaging, Bildsignalverarbeitung, Mathematische Modelle und Verfahren, Mustererkennung und Deep Learning sowie für die Anwendungsschwerpunkte Mess- und Automatisierungstechnik, Industrie 4.0, Umfeldwahrnehmung für Fahrzeuge und Robotik, Assistenzsysteme und Mensch-Maschine-Systeme, Sicherheitstechnik und Biometrie, Medizin und Biologie berichtet.

ISBN 978-3-7315-0833-5



9 783731 508335 >