Monokulare Blickrichtungsschätzung zur berührungslosen Mensch-Maschine-Interaktion

Zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS

von der KIT-Fakultät für

Elektrotechnik und Informationstechnik

des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Dipl.-Ing. Sebastian Vater

geb. in Minden

Tag der mündl. Prüfung: 27. 03. 2018
Hauptreferent: Prof. Dr.-Ing. Fernando Puente León, KIT
Korreferent: Prof. Dr. rer. nat. Olaf Dössel, KIT

Vorwort

Die vorliegende Arbeit entstand während meiner Forschungstätigkeit am Institut für Industrielle Informationstechnik (IIIT) am Karlsruher Institut für Technologie (KIT).

Danken möchte ich zuallererst Prof. Dr.-Ing. Fernando Puente León, der mir die spannenden und lehrreichen Jahre als wissenschaftlicher Mitarbeiter sowie die Erstellung dieser Arbeit mit dem nötigen Freiraum ermöglichte, während er mich unterstützend sowie motivierend mit zahlreichen wertvollen und kritischen Anregungen betreute. Ein besonderer Dank gilt auch Prof. Dr. rer. nat. Olaf Dössel für die Übernahme des Korreferats und für das hilfreiche und freundliche Verhältnis, welches bereits seit meinem Studienbeginn andauert.

Danken möchte ich allen festen Mitarbeitern des Instituts für die langjährige Zusammenarbeit. Allen Bachelor-, Master- und Diplomstudenten, die ich betreuen durfte, möchte ich für ihr Einbringen und ihre Begeisterung bei der Unterstützung von Teilen der Arbeit danken.

Einen bedeutsamen Einfluss haben die wissenschaftlichen Mitarbeiter auf diese Arbeit genommen, sowohl fachlich, durch Diskussionen und Korrekturvorschläge, durch das tägliche Zusammenarbeiten insgesamt, als auch während gemeinsamer Feierabende. Hierfür möchte ich allen Kollegen aufrichtig danken. Spezieller Dank für das Korrekturlesen und den fachspezifischen Austausch zu dieser Arbeit gilt Benjamin Jäschke, Sebastian Bauer, David Uhlig, Maximilian Schambach und Thomas Nürnberg. Meinen Freunden aus Institut und WG möchte ich für jeden einzelnen Kap-Abend danken. Ihr wisst, wer und was gemeint ist.

Ein letzter Dank richtet sich an meine Eltern Hartmut und Karin sowie meine Schwester Rebecca für eure Unterstützung, dass ihr mir stets zur Seite gestanden habt und mir auf der ganzen Welt hinterhergeflogen seid.

Karlsruhe, im März 2019 Sebastian Vater

Inhaltsverzeichnis

Symbolverzeichnis

Allgemeine Abkürzungen

Abkürzung	Bedeutung
AAM	*Active Appearance*-Modell(e)
Anz.	Anzahl
d. h.	das heißt
ED	*Edge Density*
engl.	englisch
EOH	*Edge Oriented Histograms*
FAR	Falschalarmrate (engl. *False Alarm Rate*)
FN	Menge der fälschlicherweise negativ detektierten Suchfenster (engl. *False Negatives*)
FP	Menge der fälschlicherweise positiv detektierten Suchfenster (engl. *False Positives*)
Gew.	Gewichte
Haar	*Haar*-Merkmal
HOG	*Histogram of Oriented Gradients*
HR	Detektionsrate (engl. *Hit Rate*)
IC	*Inverse Compositional*-Algorithmus
IImage	Integrales Bild (engl. *Integral Image*)
kum.	kumuliert(e)
LBP	*Local Binary Pattern*
LDA	Lineare Diskriminanzanalyse (engl. *Linear Discriminant Analysis)*
MBLBP	*Multi-block Local Binary Pattern*
MIP	*Multiple-Instance Learning*
MSLBP	*Modified Symmetric Local Binary Pattern*
NNEOH	*Neighbourhood Normalized Edge Oriented Histograms*

Abkürzung	Bedeutung
SD	*Steepest Descent Images*
SIFT	*Scale Invariant Feature Transform*
SVD	Singulärwertzerlegung (engl. *Singular Value Decomposition)*
SVM	*Support Vector Machine(s)*
SVR	*Support Vector Regression*
TN	Menge der richtig negativ detektierten Suchfenster (engl. *True Negatives*)
TP	Menge der richtig positiv detektierten Suchfenster (engl. *True Positives*)

Symbole

Lateinische Buchstaben

Symbol	Bedeutung
A	Kalibriermatrix
$\mathbf{d}(\mathbf{u})$	Verschiebungsvektor
$g_\iota(\mathbf{u})$	Trainingsbild mit Index ι
$g(\mathbf{u})$	Grauwertbild $g : \Omega_{\boldsymbol{g}} \rightarrow \mathbb{R}$
$\mathbf{H}$	Hessematrix
$\mathbf{H}_{\mathrm{Reg}}^{\mathrm{konv.}}$	Hessematrix bei konventioneller Regularisierung
$\mathbf{H}_{\mathrm{Reg}}^{\mathrm{bew.}}$	Hessematrix bei bewegungsadaptiver Regularisierung
$\mathbf{h}_{\mu^*,\iota}(\Theta_{\mu^*}, t)$	Hypothesen des ausgewählten Merkmals $\boldsymbol{m}_{\mu^*}$
$\mathbf{h}_{\mu^*,\iota}^{\mathcal{I}_{\mathrm{Neg}}}(\Theta_{\mu^*}, t)$	Hypothesen der Merkmalsantworten für Merkmal $\boldsymbol{m}_\mu$ der negativen Trainingsbilder
$\mathbf{h}_{\mu^*,\iota}^{\mathcal{I}_{\mathrm{Pos}}}(\Theta_{\mu^*}, t)$	Hypothesen der Merkmalsantworten für Merkmal $\boldsymbol{m}_\mu$ der positiven Trainingsbilder
$\mathcal{I}$	Menge aller Trainingsdaten
$\mathcal{I}_{\mathrm{Neg}}$	Menge der Klasse der negativen Trainingsdaten
$\mathcal{I}_{\mathrm{Pos}}$	Menge der Klasse der positiven Trainingsdaten
j	Index
$\mathbf{J}$	Jacobimatrix einer mehrdimensionalen Funktion
k	Index
$\mathcal{K}$	Trainierter Kaskadenklassifikator

Symbol	Bedeutung
$\mathcal{L}_{\iota}^{\lvert\mathcal{I}\rvert\times 1}(\boldsymbol{m}_{\mu})$	Liste der Merkmalsantworten für Merkmal $\boldsymbol{m}_{\mu}$ auf allen Trainingsdaten
$\mathcal{L}_{\mu,\iota}^{\lvert\mathcal{M}\rvert\times\lvert\mathcal{I}\rvert}$	Liste aller Merkmalsantworten, angeordnet nach positiven und negativen Samples
$\mathcal{L}_{\rho}^{1\times\lvert\mathbf{P}\rvert}(\mathbf{m}_{\mu})$	Liste der Merkmalsantworten für Merkmal $\boldsymbol{m}_{\mu}$ auf Graubild mit *Sliding-Window*-Ansatz
$\mathcal{L}_{\mu,\iota}^{\lvert\mathcal{M}\rvert\times\lvert\mathcal{I}_{\mathrm{Pos}}\rvert}$	Liste der Merkmalsantworten auf den positiven Trainingsbeispielen
$\mathcal{L}_{\mu,\iota}^{\lvert\mathcal{M}\rvert\times\lvert\mathcal{I}_{\mathrm{Neg}}\rvert}$	Liste der Merkmalsantworten auf den negativen Trainingsbeispielen
$\mathcal{M}$	Menge aller Merkmale
$\mathbf{m}_{\mu}$	Ein- oder mehrdimensionales Merkmal (schwacher Klassifikator)
$\mathbf{m}_{\mu,\iota}$	Ein- oder mehrdimensionale Merkmalsamtwort auf $g_{\iota}(\mathbf{u})$
$\mathbf{M}_{\mathrm{H}}$	Perspektivische Transformationsmatrix (Homographie)
$\mathcal{M}^{\mathrm{ED}}$	Menge aller *Edge Density*-Merkmale
$\mathcal{M}^{\mathrm{EOH}}$	Menge aller *Edge Oriented Histogram*-Merkmale
$\mathcal{M}^{\mathrm{Haar}}$	Menge aller Haar-Merkmale
$\mathcal{M}^{\mathrm{HOG}}$	Menge aller *Histogram of Oriented Gradients*-Merkmale
$\mathcal{M}^{\mathrm{LBP}}$	Menge aller *Local Binary Pattern* -Merkmale
$\mathcal{M}^{\mathrm{MBLBP}}$	Menge aller *Multi-block Local Binary Pattern* -Merkmale
$\mathcal{M}^{\mathrm{NNEOH}}$	Menge aller *Neighbourhood Normalized Edge Oriented Histogram*-Merkmale
$\mathbf{m}_{\mu^*}(s,t)$	Ausgewähltes, diskriminantestes Merkmal
$\mathbf{M}$	Starrkörpermodell
N	Gesamtzahl Pixel
n	Laufindex Pixel
$\mathbf{P}$	Positionen aller Detektionsfenster im *Sliding Window*-Ansatz
$\mathbf{P}_{\rho}^{\mathrm{FP}}(\mathbf{u})$	Falsch positive Detektionsfenster
$\mathbf{p}$	Kopfposenvektor $\mathbf{p} = [r_{\mathrm{x}}, r_{\mathrm{y}}, r_{\mathrm{z}}, t_{\mathrm{x}}, t_{\mathrm{y}}, t_{\mathrm{z}}]^{\mathrm{T}}$
$\mathcal{P}_{\mathrm{Neg}}$	Pool der Klasse negativer Trainingsdaten
$\mathcal{P}_{\mathrm{Pos}}$	Pool der Klasse positiver Trainingsdaten
$\mathbf{P}$	Projektion (Lochkameramodell)

Symbol	Bedeutung
$\mathbf{P}_{\rho}^{\mathrm{TP}}(\mathbf{u})$	Richtig positive Detektionsfenster
$\mathbf{q}_{\mu}$	Parametervektor eines Merkmals in Fenster $\varrho_{\rho}(\mathbf{u})$
$\mathbb{R}$	Menge der reellen Zahlen
$r(\mathbf{u})$	Radius an der Stelle $\mathbf{u}$
r_{x}	Rotation um x-Achse
r_{y}	Rotation um y-Achse
r_{z}	Rotation um z-Achse
s	Index einer Stufe des Kaskadenklassifizierers
S	Gesamtstufenanzahl des Kaskadenklassifizierers (starke Klassifikatoren)
t_{x}	Translation in x-Richtung
t_{y}	Translation in y-Richtung
t_{z}	Translation in z-Richtung
t	Aktueller Laufindex der Anzahl von Merkmalen der Klassifikatorstufe (schwache Klassifikatoren)
u	Pixelposition in horizontaler Richtung
$\mathbf{u} = (u, v)$	Pixelkoordinate
v	12D-Blickrichtungsvektor
v	Pixelposition in vertikaler Richtung
$\mathbf{w}_{\mu^*,\iota}(\mathbf{m}_{\mu^*}, t)$	Gewichtungen der Trainingsbilder für gewähltes Merkmal
$\mathbf{W}$	Warpingfunktion
$\mathbf{x}$	Dreidimensionale Modellkoordinate
$\mathcal{Z}$	Klassenzugehörigkeit der Trainingsdaten, $\mathcal{Z} = \{-1, 1\}$
$\mathrm{ZM}(\mathbf{u})$	Zentrumvotingmap

Griechische Buchstaben

Symbol	Bedeutung
$\alpha(\epsilon_{\mu^*,\iota}, t)$	Gewicht Klassifikationsgüte des ausgesuchten Merkmals
$\Delta\alpha$	Blickrichtungsfehler durch fehlerbehaftete Irislokalisation
β	Anzahl der richtungsunabhängigen Quantisierungen der Gradientenrichtung eines Pixels
$\beta(\epsilon_{\mu^*,\iota}, t)$	Verlustfunktion *Adaboost* eines Trainingsbeispiels
$\boldsymbol{\gamma}(\mathbf{u})$	Isophote an der Stelle $\mathbf{u}$

Symbol	Bedeutung
$\epsilon_{\mu^*,\iota}(t)$	Klassifikationsfehler durch ausgewähltes Merkmal
$\Theta_{\mu^*}(t)$	Schwellenwert des ausgewählten Merkmals
ι	Laufindex der Trainingsbilder
$\check{\iota}$	Absteigend sortierter Laufindex der Trainingsbilder
ι_{Neg}	Laufindex der Trainingsbilder im negativen Pool
ι_{Pos}	Laufindex der Trainingsbilder im positiven Pool
$\kappa\left(\mathbf{u}\right)$	Krümmung an der Stelle $\mathbf{u}$
μ	Laufindex der Merkmale
$\boldsymbol{\nu}_{\iota}(t)$	Kumulierter Merkmalsgütevektor in einer Stufe
$\boldsymbol{\nu}_{\iota}^{\downarrow}(t)$	Absteigend sortierter kumulierte Merkmalsgütevektor
$(\xi,\eta)^{\mathrm{T}}$	Intrinsiche Koordinaten
ρ	Laufindex der zu untersuchenden Detektionsfenster
$\varrho_{\rho}(\mathbf{u})$	Position des Detektionsfensters mit Index $\rho = \{1,\ldots,\lvert\mathbf{P}\rvert\}$, $\varrho_{\rho}(\mathbf{u}) = [\mathbf{u}_{\rho}, h_0, b_0]^{\mathrm{T}}$
τ_s	Schwellenwert der Stufe (des starken Klassifikators)
ψ	Winkel der Normalenrichtung eines Pixelgradienten
$\Omega_{\boldsymbol{g}}$	Definitionsmenge eines Grauwertbildes, $\Omega_{\boldsymbol{g}} \subseteq \mathbb{R}^2$
ω_k	Gewichtungen einzelner (Haar-)Merkmals Regionen

(hochgestellte) Indizes

Index	Bedeutung
$(\bullet)^{\downarrow}$	Absteigend sortiert
$(\bullet)^{\uparrow}$	Aufsteigend sortiert
$(\bullet)^{\mathrm{D}}$	Detektiertes Suchfenster
$(\bullet)^{\bar{\mathrm{D}}}$	Verworfenes Suchfenster
$(\bullet)^{\mathrm{T}}$	Transponiert

Indizes

Index	Bedeutung
$(\bullet)_{\mathrm{Ges}}$	Bezogen auf den gesamten Klassifizierer
$(\bullet)_{\mathrm{Neg}}$	Variable bezogen auf Anteil der negativen Trainingsbeispiele
$(\bullet)_{\mathrm{Pos}}$	Variable bezogen auf Anteil der positiven Trainingsbeispiele

Mathematische Operatoren

Operator	Bedeutung
$\lceil \cdot \rceil$	*Ceil*-Funkion
$\lfloor \cdot \rfloor$	*Floor*-Funkion
$\mathrm{grad}\,(\cdot)$	Gradientenvektor eines Pixels
H	Heaviside-Funktion
$\lvert \cdot \rvert$	Kardinalität einer Menge
δ	Kronecker-Produkt
$\lvert\lvert \cdot \rvert\rvert$	Norm eines Vektors
$\angle\,(\mathrm{grad}\,(\cdot))$	Richtung eines Pixelgradienten
$\mathbf{0}$	Nullvektor
$\mathbf{1}$	Einsvektor

1 Einleitung

Die Omnipräsenz einer immerzu fortschreitenden Technologisierung zeigt sich täglich in der Öffentlichkeit durch eine krasse Verwendung handgehaltener Multimediageräte [staa; stab] sowie einer zunehmenden Unterstützung [hei] oder gar Substitution von zuvor manuell ausgeführten Arbeiten durch Einsatz moderner Technologien [Haw]. In der Interaktion zwischen Mensch und Maschine in Form von Ein- und Ausgabe haben sich in den letzten 10 Jahren [Der] neue Bedienmöglichkeiten in Form von Touchscreens oder sogar durch die Bedienung über Gesten mittels Tiefeninformationen in modernsten Geräten etabliert [Appb]. Konventionelle Schnittstellen wie die Computermaus werden zusehends abgelöst. Gerade für Menschen mit Behinderungen sind neue Schnittstellen die einzige Möglichkeit nicht nur zur Interaktion mit einer Maschine, sondern auch zur zwischenmenschlichen Kommunikation [Fer11].

1.1 Motivation

Die berührungslose Mensch-Maschine-Interaktion wird in dieser Arbeit in Form einer Interaktion mittels Erkennen der Blickrichtung des Nutzers interpretiert. Auf Grundlage der Blickrichtung lassen sich für zahlreiche Anwendungen Lösungen und Hilfestellungen definieren, die eine Spannweite von Komfort und Unterhaltung über Anwendungen für Personen mit starken körperlichen Einschränkungen bis zu sicherheitsrelevanten Themen abdecken.

Zur Weiterentwicklung und Verbesserung des Komforts und der Unterhaltung kann bereits das Ersetzen der Computermaus, die zusammen mit der Tastatur seit Jahrzehnten die konventionelle Schnittstelle mit dem Computer darstellt, im täglichen Gebrauch eines Rechners beitragen. Darüber hinaus kann damit insbesondere das Hindernis einer körperlichen Einschränkung von Menschen, denen eine Bedienung der

Computermaus nicht möglich ist, überwunden werden [All; Haw]. Auch im Neuromarketing können Methoden zur Blickrichtungsschätzung genutzt werden, um etwa die Zielerreichung von geschalteten Werbungen oder anderen Punkten von Interessen auf Webseiten zu validieren [Bri]. Ein anderes Beispiel mit dem Ziel der Aufmerksamkeits- und Interessenerkennung sind *Digital Signages*, für welche es zahlreiche kommerzielle Anbieter gibt [htt; wwwa; wwwb]. Ein aktueller Forschungsaspekt und wichtiger Anwendungsfall der Blickrichtungsschätzung ist die Überwachung der Aufmerksamkeit sowie Müdigkeit von Führern von Personenkraft- sowie Lastkraftwagen. Die Blickrichtung wird für die Kontrolle der Wachsamkeit und bei der Bestimmung des Punktes, auf den der Fahrer gerade seine Aufmerksamkeit richtet, als wesentliches Merkmal herangezogen [JY02; LKS12; MBM16] und findet außerdem Interesse im Bereich der empirischen Bildungsforschung [Unia].

Der nächste Schritt der Mensch-Maschine-Interaktion ist die berührungslose Interaktion. Hier spielen die auditive und visuelle Wahrnehmung zentrale Rollen. Neben einer aktuellen Entwicklung der Sprachinteraktion [Ama; Appa; Goo] kann die Bedienung durch Erkennung der Blickrichtung in Kombination mit einer einfachen, eine Bestätigungsfunktion ausführenden, (auditiven) Kommunikation, physiologisch bedingt, schnell und nicht-invasiv erfolgen.

Die vorliegende Arbeit beschäftigt sich vor diesem Hintergrund mit der Erforschung der Schätzung der Blickrichtung in monokularen Bildsequenzen.

1.2 Vorüberlegungen und Randbedingungen

Der in dieser Arbeit verfolgte Ansatz arbeitet monokular, aus dem Griechischen für „ein“(monos) und dem Lateinischen für „Auge“(oculus) kommend, womit das Sehen mit einem Auge bezeichnet wird (im Kontrast zum binokularen, menschlichen Sehen). Technisch bedeutet dies, dass der verwendete Sensor nicht zur Aufnahme der Szene in Stereo geeignet ist, was impliziert, dass keine Tiefeninformation über die Szene zur Verfügung steht. Diese wesentliche Randbedingung birgt neben dem Nachteil, ohne Tiefeninformation auskommen zu müssen, den großen Vorteil, dass einfache monokulare Sensoren, wie Webcams oder in hand-

gehaltenen Geräten, wie Smartphones und Tablets, eingebaute Kameras in nahezu jedem modernen Multimedia-System vorhanden und damit überall verfügbar sind.

Für die Anwendung von Algorithmen und Lösungen zur monolularen Mensch-Maschine-Interaktion entfällt somit das Anschaffen zusätzlicher Hardware genauso wie das Umgewöhnen von bereits bekannter Hardware wie Mobiltelefone auf neue Systeme. Während bereits existierende kommerzielle Lösungen [tobb] teure Hardware voraussetzen oder obstruktiv sind und die Interaktion mit der realen Welt durch Verdeckung der Sicht einschränken, stellt der vorgeschlagene Ansatz eine kostengünstige und breit anwendbare Alternative zur berührungslosen Mensch-Maschine-Interaktion dar. Die Herausforderungen, die mit diesen Randbedingungen einhergehen und in dieser Arbeit ausführlich diskutiert werden, sind neben dem inhärenten Fehlen der Tiefeninformation, welche die Bestimmung der Blickrichtung in einer dreidimensionalen Welt erschwert, die teilweise schlechte Qualität der aufgenommenen Bilder, welche sich durch eine niedrige Bildauflösung, schlechten Kontrast oder schlechte Beleuchtungsbedinungen äußert.

Die Zielsetzung dieser Arbeit ist die Bewältigung dieser Herausforderungen im Kontext der Realisierbarkeit und unter der Motivation der Frage der zu erreichenden Genauigkeit.

Dabei liegen die Forschungsschwerpunkte insbesondere in der Extraktion der zur Bestimmung der Blickrichtung benötigten Informationen aus Bilddaten. Dies sind im Falle des hier verfolgten, monokularen, auf die 2D-Erscheinung des Nutzers basierten Ansatzes die

- Position der Augen und damit des Kopfes im Bild,
- die präzise Position der Iriden und
- die dreidimensionale Position des Kopfes,

mittels derer die Blickrichtung bestimmt werden soll.

1.3 Problemstellung

Als Ausgangspunkt dient bei der monokularen Blickrichtungserkennung ein ein- oder dreikanaliges Bild, welches mit handelsüblichen,

in vielen modernen Geräten verbauten Kameras aufgenommen werden kann. Hierbei ist durch das Zurückgreifen auf einfache Hardware sowohl eine breite als auch damit inhärent einhergehende kostengünstige Verbreitung der Technologie problemlos möglich, während invasive Ansätze zur Blickrichtungserkennung, wie etwa *Head Mounted Displays* [Man+15], oder solche, die auf aktiver Beleuchtung [WJ16] oder Stereobildern beruhen, aufwendigere und damit teurere und insbesondere nicht bereits überall verfügbare Hardware benötigen. Ein rein erscheinungsbasierter Ansatz, wie er in dieser Arbeit verfolgt wird, muss also sowohl die Herausforderung einer tendenziell niedrigen Bildauflösung sowie das Fehlen der Information von Tiefe bewältigen. Dies verlangt nach robusten und präzisen Methoden, welchen den Gegenstand dieser Forschungsarbeit bilden. Um ein intuitives Ansprechverhalten zu ermöglichen, sind darüber hinaus echtzeitfähige Methoden notwendig.

Eine schematische Skizze zur erscheinungsbasierten Blickrichtungserkennung zeigt Abbildung 1.1. Auf dem als Zylinder modellierten Kopf eines Benutzers, der sich vor einem Bildschirm befindet und dessen Blick auf diesen gerichtet ist, sind die detektierten Augen eingezeichnet. Der Kamerasensor, der hier als Webcam skizziert ist, befindet sich auf dem Monitor. Weiterhin illustriert die Kopfpose mit den Koordinatenachsen x, y, z und den Rotationsrichtungen Neigen (r_{x}), Gieren (r_{y}), Rollen (r_{z}) die für eine kopfposeninvariante Blickrichtungsschätzung notwendige 3D-Modellierung des Kopfes und damit die Herausforderung einer 3D-Kopfposenschätzung aus 2D-Bilddaten. Die Abbildung zeigt auf einen Blick die Problemstellung: Basierend auf einem mit einer monokularen Kamera aufgenommenen Bild gilt es, sowohl die Iriden zu lokalisieren als auch basierend auf dem unvollständigen Wissen über die Szene den dreidimensionalen Zustand des Kopfes zu schätzen, um so auf die Blickrichtung zu schließen. Die Herausforderungen der hierzu notwendigen Schritte werden im weiteren Verlauf als Kern dieser Arbeit diskutiert und Lösungsvorschläge erforscht.

Es folgt eine Übersicht zur Struktur der Arbeit, welche einen thematischen Abriss des Aufbaus der Arbeit skizziert. Anschliessend wird der eigene Beitrag bezüglich der partikulären Teilgebiete kurz geschildert, um die eigene wissenschaftliche Leistung dieser Arbeit herauszustellen.

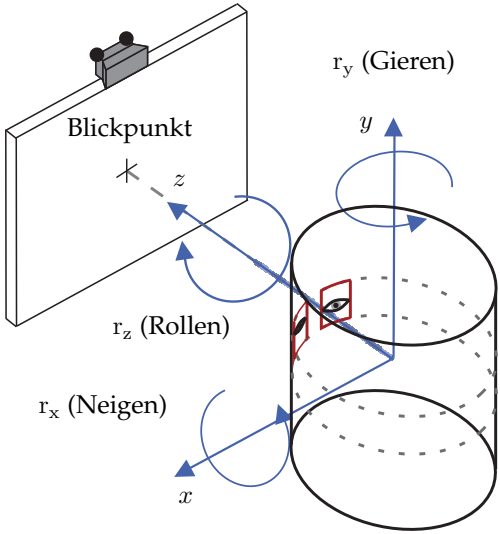


Abbildung 1.1 Skizze zur monokularen Blickrichtungsschätzung.

1.4 Struktur der Arbeit

Die Struktur der vorliegenden Arbeit folgt, analog der Chronologie des Denk- und Forschungsprozesses sowohl hinsichtlich der Ideenfindung der Ansätze als auch der Ausarbeitung jener Ideen, einem *Top-Down*-Aufbau. Die Visualisierung der Struktur der Arbeit in Abbildung 1.2 soll als Übersicht dienen und gleichzeitig die den subsequenten Kapiteln entsprechend benannte Abfolge der Behandlung der einzelnen Schwerpunkte aufzeigen.

Nach Aufnahme des Eingangsbildes ist zum Finden charakteristischer Bildregionen eine Detektion der Augen notwendig, welche in Kapitel 2 behandelt wird. Nach einer Erörterung zum Stand der Technik auf dem Gebiet der Objektklassifikation liegt der Schwerpunkt in diesem Kapitel auf der Diskussion der Klassifikationsaufgabe mit Hilfe eines Kaskadenklassifikators und den Erweiterungen durch die Verwendung mehrdimensionaler Merkmale und dem Bewerkstelligen der damit einhergehenden hohen Speicherplatzanforderungen. Basierend auf den so

gefundenden Augenregionen können mit Hilfe des in Kapitel 3 beschriebenen Ansatzes zur Irislokationsation präzise die zweidimensionalen Koordinaten der Iriden gefunden werden, welche zur Blickrichtungsschätzung benötigt werden. Um eine Blickrichtungsschätzung kopfposeninvariant zu gestalten, ist eine Schätzung der 3D-Kopfpose notwendig. Ein auf der Berechnung des optischen Flusses basierender monokularer Ansatz soll hierzu in Kapitel 4 beschrieben werden. Die gefundenen Iriden sowie die 3D-Kopfpose können dann mit Hilfe eines Regressionsansatzes zur Blickrichtungsschätzung genutzt werden. Die Beschreibung der Methode und die Diskussion der Ergebnisse erfolgt in Kapitel 5.

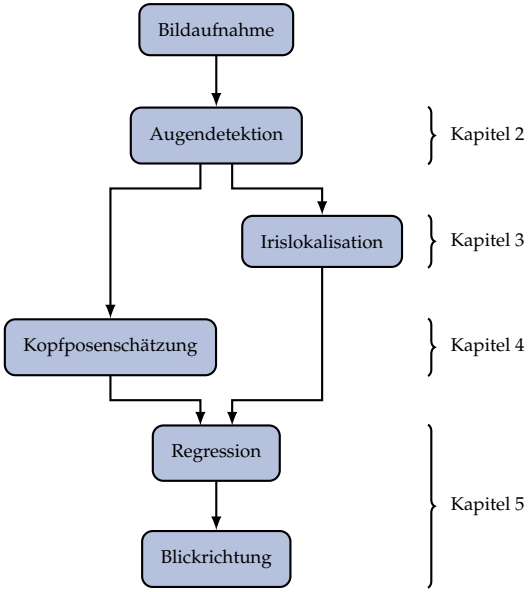


Abbildung 1.2 Schematische Gliederung der Arbeit und Aufteilung der Kapitel.

1.5 Eigener Beitrag

Neben einer übersichtlichen Strukturierung und klaren Formulierung der Arbeit sollen die eigenständig geleisteten wissenschaftlichen Beiträge eindeutig gekennzeichnet und die Trennung von Vorarbeiten anderer

Forscher hervorgehoben werden. Zusätzlich zur klaren Kennzeichnung der verwendeten und weiterführenden Literatur sollen in diesem Abschnitt die wesentlichen während dieser Arbeit enstandenen Beträge kurz erläutert werden.

1.5.1 Augendetektion

Zur Bewerkstelligung der Aufgabe der Augendetektion wurde in dieser Arbeit der Ansatz eines Kaskadenklassifikators gewählt. Hierzu wurde der recheneffiziente Ansatz des bekannten Viola-Jones-Kaskadenklassifikators [VJ01] durch folgende eigene Beiträge zum Zwecke der Erhöhung der Robustheit der Detektion gegenüber komplexen Beleuchtungsbedingungen sowie einer Verbessserung der Performanz [Par+11] erweitert:

- Erweiterung der alleinigen Verwendung eindimensionaler Merkmale in Kaskadenklassifizierern (Haar [LM02; Ore+97], *Edge Density* [PB07], Kombinationen [Zhi+06]) auf ein abgeschlossenes Rahmenwerk eines Multi-Merkmals-Kaskadenklassifikators.

- Verwendung von zuvor in *Support Vector Machines* [Vap98] eingesetzten mehrdimensionalen Merkmalen (*Histogram of Oriented Gradients* [DT05], *Local Binary Pattern* [OPM02]) in einen Multi-Merkmals-Kaskadenklassifikator durch *Boosting* der deskriptivsten Merkmalsdimensionen.

- Vorstellung des Konzepts des *Merkmals-Baggings*, um den erhöhten Speicherbedarf einer parallelen Merksmalsauswahl durch kaskadiertes *Boosting* vieler Merkmalstypen zu umgehen.

1.5.2 Irislokalisation

Aufbauend auf dem Vorschlag von Isophoten als Bildmerkmale von Lichtenauer et al. [LHR05] sowie Ansätzen zur Nutzung der Isophoten zur Irislokalisation von Valenti und Gevers [VG08] wird ein erscheinungsbasiertes Verfahren zur präzisen Lokalisierung von Iriden in Bildsequenzen mit folgenden Beiträgen vorgeschlagen:

- Ein Gewichtungsschema für die Schätzung der Lage der Iris, welche sowohl den Start- als auch den Endpunkt von Vektoren zur Bestimmung des Irismittelpunktes sinnvoll gewichtet und dabei die lokale Erscheinung des Irisbereiches unter Einbeziehung des lokalen Radiusses berücksichtigt, während zwischen hellen und dunklen Zentren innerhalb des Augenbereiches unterschieden wird.

- Die Fusion der so gewonnenen Information mit den Ergebnissen eines Kaskadenklassifikators, dessen konventionell binärer Ausgang auf einen quasi-kontinuierlichen Wertebereich erweitert wurde.

1.5.3 Erscheinungsbasierte Kopfposenschätzung

Der hier verfolge Ansatz zur Kopfposenschätzung basiert auf dem von Lucas und Kanade [LK81] vorgestellten Grundprinzip der Berechnung des optischen Flusses. Basierend auf Arbeiten von La Cascia et al. [LSA00], Xiao et al. [Xia+03], Jang und Kanade [JK08] beinhaltet die vorliegende Arbeit folgende Neuerungen:

- Um das Aperturproblem und das Problem der Invertierung der Hessematrix zu lösen, wird die in der Literatur nach Wissen des Autors ausschließlich skalarwertige Regularisierung des optischen Flusses auf eine vektorwertige Regularisierung erweitert. Es wird gezeigt, wie durch einen neuartigen Regularisierungsterm einzelne Bewegungen des optischen Flusses gezielt manipuliert werden können, um die Regularisierung an die aktuell vorherrschende Bewegung des zu verfolgenden Kopfes anzupassen.

- Basierend auf der Verfolgung stabiler Merkmale wird eine Methode zur adaptiven, online-Regularisierung vorgeschlagen, in der eine vom optischen Fluss unabhängige Schätzung der 3D-Kopfbewegung genutzt wird, um den vektorwertigen Regularisierungsparameter zu bestimmen.

1.5.4 Monokulare Blickrichtungsschätzung

Motiviert von der Arbeit in [VSG12] wurden abschliessend die erforschten Methoden in ein Rahmenwerk zur erscheinungsbasierten Blickrichtungsschätzung integriert. Während die Neuerung darin besteht, die erforschten Methoden und Erkenntnisse in einem Regressionsansatz zu fusionieren werden anhand eines *Proof-of-Concept*-Experimentes erzielte Ergebnisse mit einer Genauigkeit im Bereich des Standes der Technik präsentiert und das Potential als Ausblick und Grundlage für folgende Forschungsarbeiten diskutiert.

2 Augendetektion

Dieses Kapitel behandelt die Augendetektion in Bildern unter realen Bedingungen, die sich durch Beleuchtungsunterschiede, Verdeckungen und komplexe, heterogene Hintergründe auszeichnen.

Ziel ist es, die Position der Augen und damit auch implizit des Kopfes zu finden, um diese Information als Ausgangspunkt für die nachfolgenden Schritte der präzisen Irislokalisation und monokularen Kopfposenschätzung und schließlich der erscheinungsbasierten Blickrichtungsschätzung zu verwenden. Den Beginn des funktionalen Ablaufs in Abb. 1.1 markierend soll mit dem hier beschriebenen Algorithmus, ausgehend von einem Eingangsbild, welches in dieser Arbeit durch Bildaufnahme mittels einfacher Hardware geschieht, der erste Schritt der hier verfolgten Mensch-Maschine-Interaktion beschrieben werden.

2.1 Problemstellung

Das Erkennen und Lokalisieren von Objekten der Klasse Auge in einer unbekannten Umgebung stellt einen wichtigen Bestandteil und hier den ersten Schritt der erscheinungsbasierten Mensch-Maschine-Interaktion dar. Durch Anwendung von Algorithmen und Methoden aus der Bildverarbeitung und des maschinellen Lernens lassen sich aus der Bildregion der Augen wichtige Informationen, wie etwa über die Aufmerksamkeit oder die Blickrichtung, erschließen. Um eine breite Anwendbarkeit eines Systems zur berührungslosen Mensch-Maschine-Interaktion mit einfacher, handelsüblicher Hardware wie Webcams oder Smartphones und Tablets zu ermöglichen, müssen die angewandten Methoden robust, zuverlässig und echtzeitfähig gestaltet werden. Eine große Herausforderung stellt die große Variation innerhalb der Klasse Auge (*Intraklassenvarianz*) dar sowie, bedingt durch eine geringe Auflösung, der oftmals kleine Bildbereich, welcher das Auge beschreibt und somit nur eine

begrenzte Menge an Information bereitstellt, um zwischen Objekten anderer Klassen zu unterscheiden (*Interklassenvarianz*). Forschungsbedarf resultiert aus einer unzureichenden Performanz der Detektion insbesondere unter erschwerten Bedingungen, wie Beleuchtungsunterschiede und Skalenvariationen [Par+11] sowie auf Grund des Fehlens einer Validierung der Robustheit der Verfahren unter Berücksichtigung einer veränderlichen Blickrichtung.

2.2 Stand der Wissenschaft

Die Objektdetektion in Bildsequenzen stellt in den vergangenen Jahrzehnten einen Forschungsschwerpunkt in der Mensch-Maschine-Interaktion und in der Bildverarbeitung dar, was sich in der Anzahl der Publikationen zu diesem Thema widerspiegelt [YJS06]. Seine Relevanz wird durch zahlreiche Arbeiten zur Detektion von Fußgängern [EG09] und Gesichtern [HL01] unterstrichen; Fortschritte in den Methoden ermöglichen Detektionen in Alltagssituationen unter realen, komplexen Bedingungen [ZR12].

Während in den letzten Jahren zunehmend auch *Convolutional Neural Networks* Anwendung in der Objektdetektion finden, beispielsweise die Ansätze in [Gir+14; Li+15; Sch15a; STE13], soll sich hier auf eine detaillierte Diskussion relevanter Arbeiten zur Objektklassifikation mittels Kaskadenklassifikatoren, *Boosting* [FS95] sowie *Support Vector Machines* [Vap98] (SVM) konzentriert werden, da diese aufgrund ihrer Echtzeitfähigkeit in dieser Arbeit Anwendung finden und insbesondere der Ansatz der Kaskadenklassifikatoren erforscht und erweitert wurde und darüber hinaus eine wichtige Rolle für das hier vorgestellte Rahmenwerk zur Blickrichtungserkennung spielt.

Bestehende Verfahren zur erscheinungsbasierten Augen- und Objektdetektion lassen sich nach der verwendeten Klassifikationsmethode sowie der genutzten Merkmale einteilen. Zusammen mit einer SVM werden von Oren et al. [Ore+97] *Haar-Wavelet*-Koeffizienten als Merkmale zur Fussgängerdetektion eingesetzt.

Eindimensionale *Haar*-Merkmale fungieren bei Viola und Jones [VJ01] durch recheneffiziente Berechnung mit Hilfe *Integraler Bilder* (IImage) als schwache Klassifikatoren und werden durch *Boosting* zu starken Klas-

sifikatoren in einer Kaskade in Serie geschaltet, während in [LM02] ein erweitertes Set von Haar-Merkmalen präsentiert wird. Ein weiteres skalarwertiges Merkmal, welches lokal über die Beträge der Pixelgradienten intergiert, wird als *Edge Density*-Merkmal (ED) von Phung und Bouzerdoum [PB07] vorgestellt und in eine Kaskadenstruktur integriert. Zusammen mit einer SVM werden hochdimensionale *Histogram of Oriented Gradients*-Merkmale (HOG) von Dalal und Triggs [DT05] zur Fußgängerdetektion eingeführt. Auf dem von Ojala et al. [OPH96] beschriebenen *Local Binary Pattern* (LBP) basierend, werden LBP-Merkmale in [AHP06] zur Gesichtsdetektion eingesetzt. Während *Bootstrap Aggregating* (auch *Bagging*), ein statistisches Lernverfahren zur Kombination von Klassifikationen basierend auf verschiedenen, zufällig generierten Trainingsdaten für die negativen Trainingsbeispiele entsprechend [VJ01], den Stand der Technik darstellt, schlagen die Autoren in [Zhi+06] einen 2D-Kaskadenklassifikator mit Haar-Merkmalen vor, bei dem *Bootstrap Aggregating* auch auf die positiven Trainingsbilder angewandt wird. Hierzu wird eine Kaskade mit *Bootstrap Aggregating* der negativen Trainingsbilder, allerdings mit einer niedrigen Detektionsrate von 50 %, trainiert. Diese wird dann auf den Pool der positiven Trainingsbilder angewandt. Aus den so als fälschlicherweise negativ klassifizierten positiven Trainingsbeispielen wird nun das positive Trainingsset für eine weitere, mit der ersten in Serie geschalteten, erneut mit negativen *Bootstrap Aggregating* zu trainierende Kaskade, gebildet. In [WHY09] wird ein hochdimensionaler, zusammengesetzter HOG-LBP-Merkmalsvektor vorgestellt, mit dem eine SVM für die Personendetektion trainiert wird. Darauf aufbauend bilden Zeng et al. [ZMM10] eine Kaskade aus zwei mittels *Multiple-Instance Learning* (MIP) trainierten SVM, je eine für HOG- und eine für LBP-Merkmale. Ein Kaskadenklassifikatoransatz in Kombination mit HOG-Merkmalen wird in [Zhu+06] eingesetzt. Um die dort verwendeten 36-dimensionalen HOG-Merkmale in eine Kaskadenstruktur zu integrieren, werden mit SVM schwache Klassifikatoren aus einer Stichprobe der Gesamtheit der HOG-Merkmale [SS02] trainiert, welche mit *Adaboost* zu starken Klassifikatoren zusammengesetzt werden. Der Ansatz in [Zha+07] präsentiert *Multi-Block*-LBP-Merkmale (MBLBP) und erweitert damit den skalenvarianten LBP-Ansatz auf mehr als eine Skalierung. Um der Mehrdimensionalität der MBLBP-Merkmale zu begegnen und diese in einem Kaskadenklassifikator zu integrieren, werden binäre Ent-

scheidungsbäume als schwache Klassifikatoren genutzt, welche dann zu starken Klassifikatoren geboostet werden, wobei entsprechend der eingesetzten 8-Pixel-Nachbarschaft der LBP-Merkmale Regressionsbäume mit 256 Knoten zum Einsatz kommen. Xu et al. [Xu+12] berechnen *Modified Symmetric Local Binary Pattern*-Merkmale (MSLBP), welche die Vorteile von sowohl LBP als auch von gradientenbasierten Merkmalen ausnutzen. Sie kombinieren MSLBP mit Haar-Merkmalen in einem geboosteten Kaskadenklassifikator. Um eine skalarwertige Entscheidung mit Hilfe der mehrdimensionalen LBP oder HOG-Merkmale für die Klassifikation zu erhalten und sie in den *Boosting*-Algorithmus zu integrieren, verwenden sie sowohl SVM als auch *Lineare Diskriminanzanalyse* (LDA) zur Merkmalsprojektion in einen eindimensionalen Unterraum. Bei der Projektion durch SVM werden die eindimensionalen Konfidenzabstände der einzelnen Trainingsbeispiele als Eingang für den *Adaboost*-Algorithmus verwendet, während bei der LDA die lineare Projektion, die sich aus dem maximierten Quotienten von *Interklassenvarianz* und *Intraklassenvarianz* ergibt, als Kriterium für die schwachen Klassifikatoren verwendet wird. Die beiden Merkmalstypen werden sowohl gemischt durch Auswahl mittels *Adaboost* zu starken Klassifikatoren trainiert, als auch getrennt, wobei in den ersten 9 Stufen der Kaskade ausschließlich Haar-Merkmale und in den weiteren 20 Stufen MSLBP-Merkmale zum Einsatz kommen.

Um aus dem hochdimensionalen Grauwertvektor des kompletten Trainingsvektors Merkmale zu generieren, nutzen Wang und Ji [WJ07] die Fisher-Diskriminanzanalyse (FDA). Sie erweitern die FDA zu einer rekursiven nicht-parametrischen Diskriminanzanalyse, welche robust gegenüber Kopfdrehungen die Augen detektiert. Es ist zu bemerken, dass der Vektor aller Grauwerte des Trainingsbildes das Merkmal bildet anstatt auf lokale Bildausschnitte zurückzugreifen.

2.3 Lösungsansatz und eigener Beitrag

Die zuvor beschriebenen existierende Ansätze teilen das gemeinsame Problem der Trennbarkeit der Klassen bei hoher Varianz der Trainingsdaten.

Während durch eine Wahl deskriptiverer (deskriptiv hier: die relevante Information umfassend beschreibend) und somit diskriminativerer

(diskriminativ hier: fähig, zu unterscheiden) Merkmale die Einschränkung der Konvergenz des Trainings starker Klassifikatoren aufgehoben bzw. auf eine höhere Kaskadenstufe und damit Klassifikationsgüte verschoben werden kann, führen insbesondere hochdimensionale Merkmale, die ein Einbetten von SVM oder von Binärbäumen in eine Kaskadenstruktur benötigen, zu einem erhöhten Rechenaufwand. In dieser Arbeit soll aufgrund seiner Echtzeitfähigkeit sowie Vielseitigkeit hinsichtlich der Verwendung verschiedener Merkmalstypen und der damit verbundenen Bildinformation ein Kaskadenklassifikatoransatz zum Einsatz kommen.

Der hier vorgestellte Ansatz nimmt sich des Problems einer robusten Detektion unter der Bedingung der Echtzeitfähigkeit an, indem folgende Beiträge geliefert werden [VP14b], welche den Stand der Technik erweitern sollen:

1. Zum einen wird das Training eines Kaskadenklassifikators mit verschiedenen Merkmalstypen durch kaskadiertes Training einzelner Merkmalstypen vorgestellt, welches hier *Merkmals-Bagging* genannt wird. Dadurch kann dem sehr hohen Aufwand an Speicher für die Merkmalsauswahl begegnet werden und gleichzeitig aus einem maximal großen, verschiedene Merkmalstypen beinhaltenden Pool das geeigneteste durch *Boosting* gewählt werden. Hierdurch wird eine umfassende Kombination verschiedener Merkmale unter Ausnutzung der jeweiligen Vorteile eines Merkmals ermöglicht, durch Zusammenspiel beispielsweise schnell zu berechnender Haar-Merkmale in den ersten Kaskadenstufen und HOG-Merkmale hoher Deskriptivität und geringer Beleuchtungsvarianz in höheren Stufen.

2. Es wird eine effiziente Methode zur Einbindung mehrdimensionaler Merkmale in den Algorithmus präsentiert, bei der der implementierte *Boosting*-Algorithmus auf die hochdimensionalen Merkmale und einen separaten Trainingsdatensatz angewandt wird, um eine eindimensionale Projektion der Merkmale zu erhalten und auf rechenkostenintensive Zwischenschritte mittels SVM oder Binärbäumen zu verzichten.

3. Einen weiteren Beitrag stellt der eigens erstellte IIIT-Trainingsdatensatz dar, der ein bezüglich eines Iriskoordinatensystems örtlich stationäres Detektionsfenster liefert, dessen Mittelpunkt als Approximation der Irisposition dient. Es wurden dabei annotierte Bilder von linken und rechten Augen sowie Brillenträgern unter verschiedenen Blickrichtungen aufgenommen.

2.4 Kaskadenklassifikator

Das Rahmenwerk des in dieser Arbeit implementierten Kaskadenklassifizierers basiert auf der in [VJ01] beschriebenenen Arbeit. Abbildung 2.1 zeigt schematisch die grundlegende Struktur des Klassifizierers, welche die getrennt ablaufenden Schritte Training (offline) und Detektion (online/offline) beinhaltet.

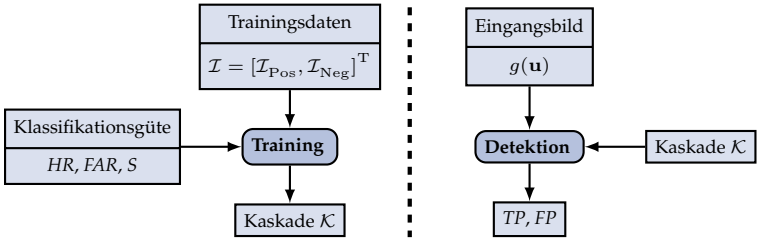


Abbildung 2.1 Zusammenhang Aufbau, Training und Detektion für den hier implementierten Kaskadenklassifikator.

Als überwachtes Lernverfahren muss der Klassifizierer zunächst anhand geeigneter Trainingsdaten trainiert werden, bevor er für eine Detektionsaufgabe angewandt werden kann. Der Teil links der gestrichelten Linie in Abb. 2.1 skizziert einen groben Überblick über das Training, auf welches in Abschnitt 2.5 näher eingegangen wird. Das Training einer Kaskade wird mit Hilfe von a priori zusammengesetzten Trainingsdaten $\mathcal{I}$ durchgeführt, deren Zusammenstellung sowie Auswirkungen auf die Güte des Klassifikators in Abschnitt 2.7.1 diskutiert wird.

Ein Kaskadenklassifikator zeichnet sich dadurch aus, dass beim Training eine minimale gewünsche Detektionsrate *HR* sowie eine maximal zulässige Falschalarmrate *FAR* vorgegeben wird, die jede Stufe s des seriell aufgebauten Klassifiziers mindestens erfüllen muss. Dabei wird eine ebenfalls vorgegebene Gesamtanzahl an Stufen S (sogenannte starke Klassifikatoren) nacheinander trainiert, welche dann bei der Detektion in Reihe geschaltet werden und sich somit eine Gesamtdetektionsrate

$$HR_{\text{Ges}} = (HR)^S \tag{2.1}$$

sowie eine Gesamtfalschalarmrate

$$FAR_{\text{Ges}} = (FAR)^S \tag{2.2}$$

für die Kaskade $\mathcal{K}$ ergibt. Die Falschalarmrate nimmt mit der Potenz S ab, was eine eher hohe *FAR* pro Stufe erlaubt.

Der Ablauf des Detektionsvorgangs eines Eingangsbildes mit Hilfe einer zuvor trainierten Kaskade $\mathcal{K}$ ist in Abbildung 2.2 skizziert.

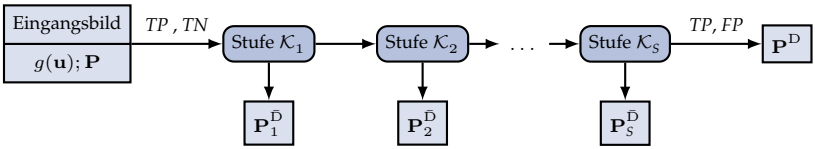


Abbildung 2.2 Schema Detektionsvorgang.

Alle Suchfenster, die sich in der mit Hilfe des *Sliding Window*-Ansatzes definierten Matrix $\mathbf{P} = \left[\varrho_1(\mathbf{u}), \ldots, \varrho_{|\mathbf{P}|}(\mathbf{u})\right]^{\mathrm{T}}$ befinden, wobei jedes Fenster $\varrho_\rho(\mathbf{u})$ als vierdimensionaler Vektor durch seine Position, Höhe und Breite definiert ist, werden separat klassifiziert, wobei die Teilmenge der pro Stufe korrekt als „*kein Auge*“detektierten Teilfenster,

$$1 - FAR = \frac{|TN_s|}{|TN_s| + |FP_s|}\,, \tag{2.3}$$

sowie die fälschlicherweise als negativ klassifizierte Teilmenge pro Stufe

$$1 - HR = \frac{|FN_s|}{|FN_s| + |TP_s|} \tag{2.4}$$

vor Eingang in die nächsten Kaskadenstufe ausgewiesen wird:

$$\left|\mathbf{P}_s^{\bar{D}}\right| := \{\varrho_\rho(\mathbf{u})\} : \rho \in \{TN_s \cup FN_s\}\,, \tag{2.5}$$

wobei mit $\left|\mathbf{P}_s^{\bar{D}}\right|$ die Menge der Positionsvektoren bezeichnet wird und *TN* die Anzahl der *True Negatives*, *FP* die Anzahl der *False Positives*, *FN* die Anzahl der *False Negatives* und *TP* die Anzahl der *True Positives* bezeichnen. Die hochgestellten Indizes $()^D$ bzw. $()^{\bar{D}}$ kennzeichnen hierbei Teilfenster, welche von der aktuellen Stufe als detektierte bzw. ausgewiesene Fenster klassifiziert werden.

Der in Glg. (2.4) beschriebene Anteil zurückgewiesener Suchfenster ist durch eine festgelegte $HR < 1$ bedingt. Insgesamt werden also pro Stufe

$$\left|\mathbf{P}_s^{\bar{D}}\right| = (1 - FAR + 1 - HR) \cdot |\mathbf{P}| = |TN_s| + |FN_s| \tag{2.6}$$

Suchfenster zurückgewiesen. Wie in Abschnitt 2.5 näher erläutert wird, kann durch Erlauben einer hohen *FAR* mit Hilfe weniger, mit geringem Rechenaufwand zu bestimmender, Entscheidungen schon eine Vielzahl von zu untersuchenden Suchfenstern zurückgewiesen werden, während durch die Kaskadierung eine niedrige Gesamtfalschalarmrate FAR_{Ges} sichergestellt wird. Bei gleichzeitiger hoher Trefferrate *HR*, typischerweise im Bereich über $99\,\%$, lässt sich trotz Kaskadierung eine hohe Gesamttrefferrate HR_{Ges} erreichen. Es gilt zu beachten, dass das Training so gestaltet wird, dass stets eine maximale *FAR* und eine minimale *HR pro Stufe* erzielt wird. Wenn, nach Auswahl eines oder mehrerer, für die gewählten Randbedingungen diskriminativster, Merkmale das Trainingsset bezüglich der *TP* perfekt getrennt wird (es werden alle *TP* korrekt klassifiziert), resultiert daraus bezüglich des Trainingsdatensatzes für diese Stufe $HR = 100\,\%$. Dabei kann die *FAR* genau eingehalten oder auch übertroffen werden ($< FAR$). Für den Detektionsvorgang bedeutet dies nicht, dass die trainierte *HR* oder *FAR* auch tatsächlich erreicht wird. Wie gut ein trainierter Klassifikator auf ungesehen Daten *generalisiert*, hängt insbesondere von der im Trainingsdatensatz abgedeckten *Intraklassenvarianz* ab. Da die tatsächliche *HR* und *FAR* von den vorgegebenen Raten abweichen können, sind die Raten der zurückgewiesenen Fenster, wie in Glg. (2.3) und Glg. (2.4) angegeben, als Näherungen zu verstehen.

Ein Suchfenster $\varrho_\rho(\mathbf{u})$ gilt als positiv klassifiziert, wenn es alle Stufen der Kaskade erfolgreich durchlaufen hat. Die Ergebnisfenster $\mathbf{P}^{\mathrm{D}}$ können anschließend unter Berücksichtigung der tatsächlichen Klassenzuordnung $\mathcal{Z}$ der Suchfenster als *TP* oder *FP* interpretiert werden.
Im folgenden Abschnitt soll nun zunächst genauer auf das Training einer Kaskade eingegangen werden, während in Kapitel 2.6 der verwendete Detektionsalgorithmus erläutert wird.

2.5 Training einer Kaskade

In diesem Abschnitt soll der während dieser Arbeit entstandene Trainingsalgorithmus für Kaskadenklassifikatoren detailliert beschrieben werden. Dem *Top-Down*-Ansatz folgend stellt Abb. 2.3 eine ausführlich Darstellung der linken Seite von Abb. 2.1 dar, während die Teilgebiete *Boosting* (Abschnitt 2.5.2) und *Bagging* (Abschnitt 2.5.4) im Weiteren detailliert diskutiert werden, um später auf die Erweiterungen des Standes der Technik durch *Merkmals-Boosting* (Abschnitt 2.5.3) und durch *Merkmals-Bagging* (Abschnitt 2.5.5) einzugehen.

Vor Beginn des Trainings müssen Klassifikationsgüte sowie die minimale zu detektierende Objekt- bzw. Fenstergröße h_0, b_0 festgelegt werden. Basierend auf h_0, b_0 sowie der Auswahl der Merkmalstypen wird dann die Menge $\mathcal{M}$ von Merkmalen berechnet, welche alle im Klassifikator verwendeten skalierten und verschobenen Instanzen der Basismerkmale der Merkmalstypen enthält. Gleichzeitig werden aus den Pools aller Trainingsdaten die Anzahl der Trainingsbilder, die im Training einer Stufe berücksichtigt werden, $\mathcal{I} = [\mathcal{I}_{\mathrm{Pos}}, \mathcal{I}_{\mathrm{Neg}}]^{\mathrm{T}}$, festgelegt, wobei deren Klassenzugehörigkeiten bekannt sind. Die Menge der Trainingsdaten, die pro Training einer Stufe verwendet werden können, ist dabei durch den Speicher des Rechners begrenzt; durch *Bagging* können sehr viel mehr Daten als die Anzahl der in einer Stufe eingesetzten zum Training beitragen, siehe hierzu Abschnitt 2.5.4.

Nach der Berechnung der Merkmalsantworten auf allen Trainingsdaten einer Stufe wird mit der so gewonnenen Liste $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ das *Boosting* der ersten Stufe (erster starker Klassifikator) begonnen, wobei $\mathcal{M}$ die Menge aller Merkmale mit dem Index μ, $\mathcal{I}$ die Menge der Trainingsbilder mit dem Index ι bezeichnet. Zu Beginn jeder Stufe wird jene

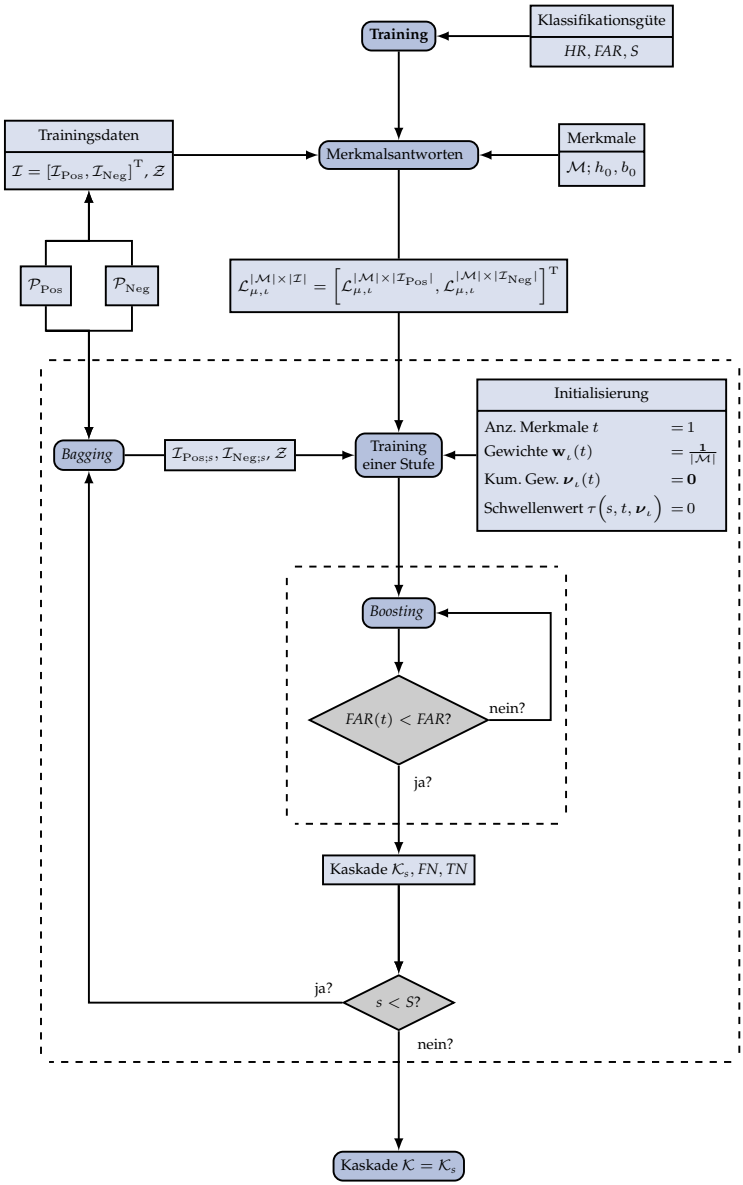


Abbildung 2.3 Schema des Trainings des implementierten Kaskadenklassifikators.

mit der Anzahl schwacher Klassifikatoren (Merkmale) $t = 0$ und den Gewichten jedes Merkmals für jedes Trainingsbild $\boldsymbol{w}_\iota(\boldsymbol{m}_\mu, t)$, gleichverteilt auf alle Trainingsbeispiele für jedes Merkmal, initialisiert. Im kumulierten Gewicht $\boldsymbol{\nu}_\iota(t)$ der Größe $|\mathcal{I}| \times 1$ einer Stufe werden die *Güte*gewichte der *ausgewählten* Merkmale $\mathbf{m}_{\mu^*}$ (schwache Klassifikatoren) für $t = 1, \ldots, t_{\mathrm{END}_s}$ (bis die *FAR* für den aktuellen starken Klassifikator erreicht ist) für jedes Trainingsbild gespeichert. Das Gütegewicht eines Merkmals hängt von seiner Fähigkeit, zwischen positiven und negativen Trainingsbeispielen zu unterscheiden, ab. Innerhalb von $\boldsymbol{\nu}_\iota(t)$ wird der Schwellenwert der Stufe, $\tau(s, t, \boldsymbol{\nu}_\iota)$, bestimmt, wobei das kumulierte Gütegewicht gewählt wird, das unter Erfüllung der *HR* die Trainingsdaten am besten trennt. Hierbei kann die Entscheidung *eines* Merkmals bei Einhaltung der *HR* die *FAR* deutlich überschreiten. Das Kumulieren vieler Entscheidungen sichert die Einhaltung der *FAR* .

Ist nach Auswahl mehrerer Merkmale die *FAR* für eine Stufe erreicht und die Gesamtstufenzahl S noch nicht erreicht, wird das Training der nächsten Stufe begonnen. Hierzu wird zunächst durch Anwendung des statistischen Lernverfahrens *Bootstrap Aggregating* die Menge der Trainingsdaten aktualisiert. Dabei werden alle nach Anwendung der aktuellen Kaskade falsch klassifizierten aktuellen positiven Trainingsbeispiele (*TP*) ersetzt, da der Klassifikator diese für Trainingsbeispiele mit der Klassenzugehörigkeit der negativen Beispiele erkennt. Weiterhin werden alle negativen Trainingsbeispiele, welche als *TN* klassifiziert wurden, ersetzt, um dem Gesamttrainingsprozess eine deutlich höhere Menge an Trainingsdaten zukommen zu lassen. Der Ablauf des Trainings soll nun im Detail diskutiert werden.

2.5.1 Berechnung der Merkmalsantworten

Die Berechnung der Antworten aller Merkmale auf allen Trainingsdaten $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ stellt den größten Rechen- sowie Speicheraufwand des Algorithmus dar. Dabei werden nach einer Auswahl der Trainingsdaten (siehe Abschnitt 2.7.1) basierend auf zuvor definierten Skalierungen und Translationen, welche für die jeweiligen Merkmalstypen lokal innerhalb des Trainingsfensters der Größe $b_0 \cdot h_0$ die Menge aller Merkmale $\mathcal{M}$ definieren, die Merkmalsantworten für alle Trainingsdaten $\mathcal{I}$ berechnet.

Es sollen zunächst die in dieser Arbeit verwendeten Merkmalstypen diskutiert werden, um dann näher auf den Trainingsablauf einzugehen.

2.5.1.1 Haar-Merkmale

Haar-Merkmale bestimmen lokale Grauwertunterschiede, wobei durch Darstellung der Bilder durch integrale Bilder die Berechnung einer Merkmalsantwort durch nur 4 Speicherzugriffe und Additionen effizient ermöglicht wird. Die skalarwertige Merkmalsantwort eines Haar-Merkmals mit Index μ lässt sich mathematisch durch

$$m_{\mu}^{\text{Haar}}\left(\varrho_{\rho}(\mathbf{u}), \mathbf{q}_{\mu}^{\text{Haar}}\right) = \sum_{k} \frac{1}{|\Omega_k|} \sum_{\mathbf{u}\in\Omega_k} \omega_k \, g(\mathbf{u}) \tag{2.7}$$

ausdrücken, wobei Ω_k eine Umgebung des Ortes $\mathbf{u}$ mit der Fläche $|\Omega_k|$ ist, ω_k der zugehörige Gewichtungsfaktor für Ω_k und $\mathbf{q}_{\mu}^{\text{Haar}}$ den Haar-Parametervektor des Merkmals μ darstellt. Da die Merkmalsantwort auf der Größe des Trainingsfensters berechnet wird, gilt stets: $|\Omega_k| \leq h_0 \cdot b_0 \, \forall \, k$.

Die Parametervektoren $\mathbf{q}_{\mu}^{\text{Haar}}$ definieren Position, Größe und Gewichtung sowie die im Training festgelegten Schwellenwerte der Merkmalsantworten für diese Flächen. Abbildung 2.4 zeigt die in Schwarz und Weiss dargestellten Flächen Ω_k, welche die Anordnung des Haar-Merkmals in der festen Trainingsfenstergröße beschreiben. Die graue Gesamtfläche entspricht hier schematisch einem Trainingsfenster der Größe $h_0 \times b_0$, in welchem die Merkmale definiert werden. In dieser Arbeit wurden, basierend auf verschiedenen Anordnungen der Umgebungen Ω_k, insgesamt 16 Basis-Haar-Merkmale implementiert. Abbildung 2.4 zeigt die 16 verwendeten Basis-Haar-Merkmale in einem Fenster der Größe $h_0 \times b_0 = 6 \times 6$ Pixel.

Durch das Detektionsfenster $\varrho_{\rho}(\mathbf{u})$, welches im Training gerade dem Trainingsfenster bei der Skalierung eins entspricht, wird durch Translation und Skalierung des Trainingsfensters die Position und Größe aller Auswertungsfenster im *Sliding-Window*-Ansatz definiert. Bei der Erstellung der Gesamtmerkmalsmenge $\mathcal{M}$ ist zu beachten, dass sich, zusammen mit einem Skalierungsfaktor 2 und einer Translation von einem Pixel, allein aus den 16 Basis-Haar-Merkmalen für ein Trainingsfenster

von 36×36 Pixeln (die Trainingsfenstergröße entspricht der minimalen Detektionsfenster- und damit Objektgröße, die detektiert werden kann) eine Gesamtzahl von 300510 möglichen Haar-Merkmalen ergibt. Werden die Merkmalsantworten mit einfacher Genauigkeit (*Single Precision*) gespeichert, so ergibt sich **pro Trainingsbild** ein Speicherbedarf von über 1,2 MB.

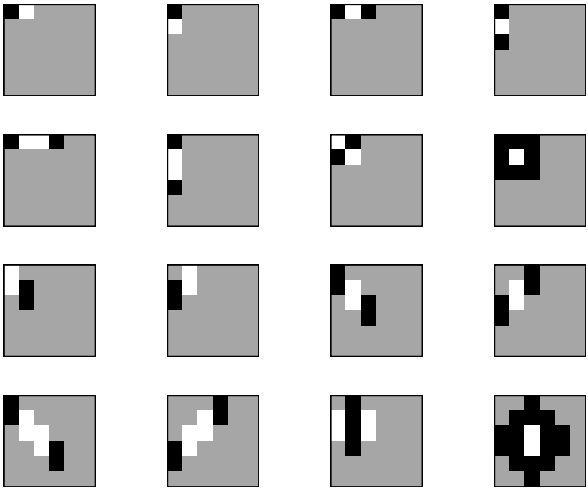


Abbildung 2.4 Die in dieser Arbeit verwendeten 16 Basis-Haar-Merkmale dargestellt in einem 6×6 Pixel großen Fenster. Es gilt zu beachten, dass in der Anwendung für die Augendetektion mit Breiten der Eingangsbilder von 480 bis 1920 Pixel eine Größe der Trainingssamples im Bereich 20×20 Pixel bis 36×36 Pixel sinnvoll ist.

2.5.1.2 Edge Density-Merkmal

Durch Berechnung von ED-Merkmalen lassen sich Gradienteninformationen erfassen:

$$m_{\mu}^{\mathrm{ED}}\left(\varrho_{\rho}(\mathbf{u}), \mathbf{q}_{\mu}^{\mathrm{ED}}\right) = \frac{1}{|\Omega|} \sum_{\mathbf{u}\in\Omega} ||\operatorname{grad}(\cdot)(g(\mathbf{u}))|| \,. \tag{2.8}$$

Hierbei ist durch $\operatorname{grad}(\cdot)(g(\mathbf{u}))$ der Gradient des Grauwertbildes an der Stelle $\mathbf{u}$ gegeben und mit $||\cdot||$ dessen Betrag. Eine Einschränkung der

ED-Merkmale ist, dass lediglich der Betrag des Gradienten genutzt wird und die Richtung vernachlässigt wird. Ebenso wie bei Haar-Merkmalen lässt sich eine Anzahl von Basis-ED-Merkmalen definieren, wobei in dieser Arbeit 8 Moden (Basismerkmale) verwendet werden.

2.5.1.3 Edge Oriented Histograms

EOH-Merkmale bieten eine effizient zu implementierende Möglichkeit, Richtungsinformation der Gradienten mit in die Merkmalsberechnung einzubeziehen. Die Merkmalsantwort ist als

$$\begin{aligned} & m_{\mu}^{\mathrm{EOH}}\left(\varrho_{\rho}(\mathbf{u}), \mathbf{q}_{\mu}^{\mathrm{EOH}}\right) \\ &= \sum_{\mathbf{u}\in\Omega} \frac{||\operatorname{grad}(\cdot)\left(g\left(\delta\left(\angle\left(\operatorname{grad}(\cdot)\left(g(\mathbf{u})\right), \gamma_j\right)\right)\right)\right)||}{||\operatorname{grad}(\cdot)(g(\mathbf{u}))||} \end{aligned} \tag{2.9}$$

definiert, mit der Anzahl der richtungsunabhängigen Quantisierungen der Gradienteninformation β, dem Kronecker-δ sowie

$$\gamma_j \in [\psi_0,\, \psi_0 + \Delta\psi)\,, \quad \Delta\psi = \frac{\pi}{\beta}, \quad \beta \in \mathbb{N}, \quad j = \{1, \ldots, \beta\}\,, \tag{2.10}$$

$$\psi_0 \in \{0 \cdot \Delta\psi, \ldots, (\beta - 1) \cdot \Delta\psi\}\,. \tag{2.11}$$

Durch $\angle(\operatorname{grad}(\cdot)(g(\mathbf{u})))$ wird die Richtung des Gradienten angegeben, ψ gibt die Intervallgrenzen der Gradientenrichtungen an und γ_j das entsprechende halboffene Intervall j der Breite $\Delta\psi$. Das Merkmal ist eindimensional, d. h., der Vektor $\mathbf{q}_{\mu}^{\mathrm{EOH}}$ enthält zusätzlich die Orientierung j, sodass für jede Skalierung und Translation innerhalb des Trainingsfensters β Merkmale bestimmt werden, welche dann analog zu den Haar-Merkmalen für alle Fenster des *Sliding Window*-Ansatzes wieder entsprechend skaliert und verschoben werden, um die Gesamtzahl der EOH-Merkmale zu erhalten. Genau wie bei den ED-Merkmalen wurden 8 Moden, also $8 \cdot \beta$ Basis-Merkmale implementiert.

2.5.1.4 Neighbourhood Normalized Edge Oriented Histograms

Eine im Verlauf dieser Arbeit entstandene Erweiterung des EOH-Merkmals stellen die *Neighbourhood Normalized Edge Oriented Histogram*-Merkmale (NNEOH) dar. Im Unterschied zu den gewöhnlichen EOH-Merk-

malen wird ein robusteres Verhalten gegenüber Beleuchtungsunterschieden durch eine großflächigere Normalisierung erreicht, wobei im Nenner in Glg. (2.9) ein Bereich $\Omega_{\text{Nenner}}^{\text{NNEOH}} \geq \Omega_{\text{Zähler}}^{\text{NNEOH}}$ berücksichtigt wird. Zur Berechnung der auf Gradienteninformation basierten Merkmale werden ebenfalls wieder integrale Bilder eingesetzt, wobei ein integrales Bild für jede der β richtungsunabhängigen Quantisierungen bestimmt wird. Während der Rechenaufwand gleich dem der EOH-Merkmale ist, stellt das Merkmal eine rechengünstige Alternative zwischen EOH-Merkmal und HOG-Merkmal dar. Berücksichtigt wurden 2 der 8 bei den EOH-Merkmalen verwendeten Moden.

2.5.1.5 Histrogram of Oriented Gradients

Beim konventionellen HOG-Merkmal wird der Bereich $b_0 \cdot h_0$ des Detektionsfensters in $\chi \times \chi$ Zellen aufgeteilt, wobei sich bei einer Blockbildung aus je 2×2 Zellen ein hochdimensionaler Merkmalsvektor der Dimension $4 \times \chi \times \chi \times \beta$ ergibt. Während durch Festlegung der Trainingsfenstergröße und Anzahl der Zellen die Zellengröße direkt definiert ist, ist die Blockgröße ein frei zu wählender Parameter.

Der HOG-Ansatz in dieser Arbeit hat das Ziel, zum einen die Dimensionalität so niedrig wie möglich zu halten und zum anderen gleichzeitig eine bezüglich des verwendeten Datensatzes und damit auch der ungesehenen Testdaten unspezifische Parametrierung der Merkmale bereitzustellen. Eine Neuerung der hier umgesetzten Implementierung dient dem Zwecke der (freien) Generalität der Anpassung an die Detektion verschiedener Objekte. Hierzu wurden folgende Ideen umgesetzt:

- Um die räumliche Struktur der Wahl der Anzahl der Blöcke nicht abhängig von der Größe des a priori festgelegten Detektionsfensters zu machen und damit die Wahl der Position einzelner HOG-Merkmalsblöcke nicht festzulegen, wie dies oft in der Literatur geschieht, wird die Wahl der Merkmale mit den geeignetesten Parametern $\mathbf{q}_{\mu}^{\text{HOG}}$ *Adaboost* überlassen, sodass, im Vergleich der Anwendung des konventionellen HOG-Merkmals, nicht notwendigerweise gleichmäßig innerhalb des Detektionsfensters Informationen extrahiert werden (kein globales Merkmal in Detektionsfenstergröße).

- Weiterhin wurde a priori eine niedrigere Merkmalsdimension erhalten, indem die Implementierung des HOG-Merkmals wie folgt durchgeführt wurde: Ein Merkmalsblock ist durch seine zentrale Zelle C, deren Position und Größe sowie deren nördlich (Ω_N), südlich (Ω_S), östlich (Ω_O) und westlich (Ω_W) angrenzenden, zur Hälfte überlappenden, Zellen gegeben, wodurch ein Vektor der Dimension 16 resultiert.

Der Merkmalsvektor ergibt sich dann durch Normalisierung der zentralen Zelle mit seinen angrenzenden Zellen für alle β Orientierungsquantisierungen zu

$$\mathbf{m}_\mu^{\mathrm{HOG}}\left(\varrho_\rho(\mathbf{u}), \mathbf{q}_\mu^{\mathrm{HOG}}\right) = \left[m_{1,\Omega_\mathrm{N}}^{\mathrm{HOG}}, m_{1,\Omega_\mathrm{O}}^{\mathrm{HOG}}, m_{1,\Omega_\mathrm{S}}^{\mathrm{HOG}}, \ldots, m_{\beta,\Omega_\mathrm{W}}^{\mathrm{HOG}}\right]^\mathrm{T} \tag{2.12}$$

als 16-dimensionaler Merkmalsvektor, wobei die einzelnen Einträge durch

$$m_{j,\Omega_k}^{\mathrm{HOG}} = \frac{\sum\limits_{\mathbf{u}\in\Omega_\mathrm{C}} || \operatorname{grad}(\cdot)\left(g\left(\delta\left(\angle\left(\operatorname{grad}(\cdot)\left(g(\mathbf{u})\right), \gamma_j\right)\right)\right)\right) ||}{\sum\limits_{\mathbf{u}\in\Omega_k,} || \operatorname{grad}(\cdot)\left(g\left(\delta\left(\angle\left(\operatorname{grad}(\cdot)\left(g(\mathbf{u})\right), \gamma_j\right)\right)\right)\right) ||}, \tag{2.13}$$

mit $k \in \{\mathrm{N}, \mathrm{O}, \mathrm{S}, \mathrm{W}\}$ und $j \in \{1, \ldots, \beta\}$ zu berechnen sind. Hierbei ist zu beachten, dass sich innerhalb des Trainingsfensters Blöcke für verschiedene $\mathbf{q}_\mu^{\mathrm{HOG}}$ selbst überlappen können, wobei alle möglichen Positionen der Merkmalsblöcke zu Beginn des Trainings aus den gewählten Skalierungen und Translationen, ausgehend vom kleinsten Block mit einer Zellengröße von 2×2 Pixeln, festgelegt werden. Der Parametervektor $\mathbf{q}_\mu^{\mathrm{HOG}}$ enthält neben den Positionen der Merkmalsblöcke die Größen der Zellen, die Überlappung der Zellen sowie β Schwellenwerte, Gewichte und Paritäten der einzelnen Dimensionen, siehe hierzu Abschnitt 2.5.3.

2.5.1.6 Local Binary Pattern und Multi-Block Local Binary Pattern

Durch die LBP sollen neben lokalen Grauwertunterschieden und Gradienteninformationen auch Information über die lokal vorherrschende Textur erfasst werden. Zur Berechnung von LBP-Merkmalen wird hierzu ein zentraler Grauwert $g(\mathbf{u}_\mathrm{c})$ mit seiner Umgebung Ω verglichen

und das Ergebnis dieses Vergleichs als Histogramm gespeichert. Das mehrdimensionale Merkmal ist durch die Anzahl seiner Abtastpunkte $n \in \{1, \ldots, N\}$ sowie seinen Radius R gegeben:

$$\mathbf{m}_{\mu}^{\mathrm{LBP}}\left(\varrho_{\rho}(\mathbf{u}), \mathbf{q}_{\mu}^{\mathrm{LBP}}\right) = \sum_{\mathbf{u}\in\Omega} \sum_{n=0}^{N-1} \mathrm{H}\big(g(\mathbf{u}_p) - g_{\mathrm{c}}(\mathbf{u})\big) \cdot 2^n \,. \tag{2.14}$$

Hier stellt $\mathrm{H}(\cdot)$ die Heaviside-Funktion (auch Sprungfunktion) dar, welche für Argumente größer gleich null den Wert 1 und ansonsten den Wert 0 annimmt. Man erhält somit für jedes Pixel einen Merkmalswert im Wertebereich $[0, 2^N - 1]$. Bei der Implementierung wurden die Varianten *Usual* (U), *Mean* (M) und *Center-Symmetric* (CS) berücksichtigt, wobei für das zentrale Pixel $g_{\mathrm{c}}(\mathbf{u})$ entweder direkt der Grauwert

$$g_{\mathrm{c}}^{\mathrm{U}}(\mathbf{u}) = g(\mathbf{u}) \,, \tag{2.15}$$

der Mittelwert aller beteiligten Grauwerte

$$g_{\mathrm{c}}^{\mathrm{M}}(\mathbf{u}) = \frac{g(\mathbf{u}) + \sum_{n=1}^{N-1} g(\mathbf{u}_n)}{N} \tag{2.16}$$

oder das gegenüberliegende Pixel

$$g_{\mathrm{c}}^{\mathrm{CS}}(\mathbf{u}) = g\Big(\mathbf{u}_{\frac{N}{2}+n}\Big) \tag{2.17}$$

gewählt wird. Der Parametervektor $\mathbf{q}_{\mu}^{\mathrm{LBP}}$ enthält sowohl die Zahl der Abtastpunkte N, den Radius R als auch die Information, ob es sich um ein U-, M- oder CS-LBP-Merkmal handelt. Die Anzahl der Texturmuster wird durch Erweiterung auf Rotationsinvarianz und durch Unterscheidung von *Uniform Pattern* und *Non-Uniform Pattern* auf insgesamt $N + 2$ Muster reduziert [OPM02]. Die *Uniform Pattern* zeichnen sich dadurch aus, dass aus der Gesamtzahl aus 2^N Mustern für $N = 8$ nicht mehr als zwei $0/1$- oder $1/0$-Übergänge auftreten dürfen, während alle anderen Muster zu einer einzelnen Klasse, der *Non-Uniform Pattern*, zusammengefasst werden. Durch Binärdarstellung der Muster und Verschieben der Bits der einheitlichen Muster, sodass die größtmögliche Folge an Nullen, ausgehend vom kleinsten Bit, entsteht, lassen sich dann aus den *Uniform*

Pattern rotationsinvariante Merkmale generieren, welche Kanten, Ecken und Punkte beschreiben. Auf diese Weise werden in dieser Arbeit LBP-Merkmale mit $N = 8$ Punkten durch ein Histogramm mit zehn Klassen repräsentiert, wobei die einheitlichen Muster durch neun elementare Bitfolgen beschrieben werden. Eine Übersicht der hier verwendeten Texturmuster ist in Abb. 2.5 für das Augenbild in Abb. 2.6(a) dargestellt. Der 10-dimensionale Stapel aller $N + 2$ Muster ist in Abb. 2.6 zu sehen.

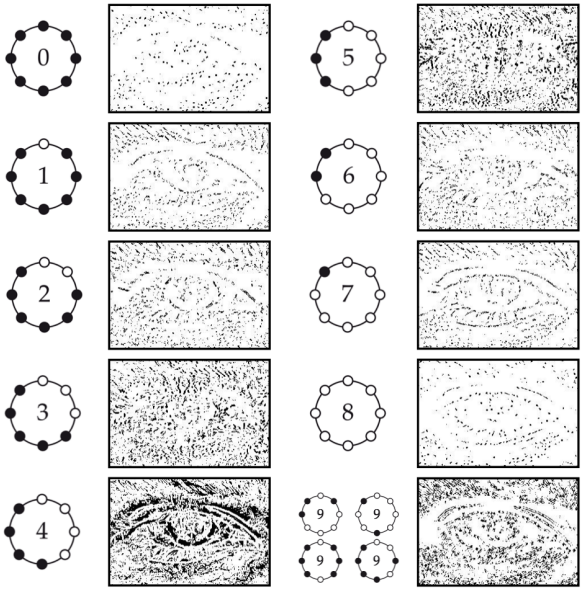
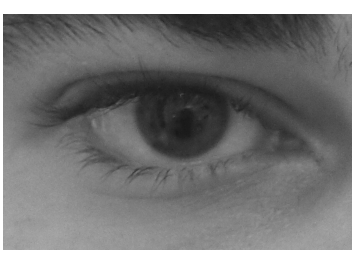


Abbildung 2.5 In dieser Arbeit einheitliche und nicht einheitliche verwendete LBP-Muster und deren Histogramm, visualisiert anhand des Bildes 2.6(a).

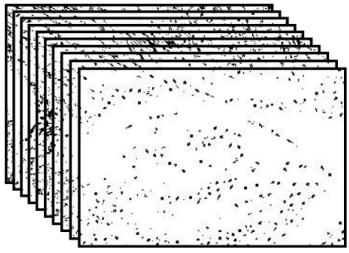
Gut zu erkennen sind die unterschiedlichen Texturen, die durch die einzelnen *Uniform Pattern* erfasst werden. Während das Muster 4 Kanten hervorhebt, lassen sich durch die Muster 0 und 8 helle und dunkle Punkte identifizieren.

Um der Skalenabhängigkeit des LBP-Merkmals zu begegnen, werden in dieser Arbeit darüber hinaus MBLBP-Merkmale eingesetzt. Diese werden gebildet, indem für die abgetasteten Grauwerte $g(\mathbf{u}_n)$ nicht die Pixel $\mathbf{u}_n$ direkt, sondern deren gemittelte Summen über eine Abtast-

region eingehen. Dadurch lassen sich durch entsprechende Wahl der Abtastregionen skalierte LBP-Merkmale bestimmen, welche Texturen unterschiedlicher Größenordnungen erfassen können.



(a) Beispielbild zur Darstellung der Histogrammklassen der in Abb. 2.5 dargestellten Texturmerkmale.



(b) Veranschaulichung des dreidimensionalen LBP-Histogramms für 10 Textureinheiten.

Abbildung 2.6 Beispielbild zur Visualisierung der LBP-Textureinheiten und dreidimensionales LBP-Histogramm.

2.5.1.7 Merkmalsantworten

Ist, basierend auf den Basis-Merkmalen der einzelnen Merkmalstypen, die Menge $\mathcal{M}$ aller möglichen Merkmale,

$$\mathcal{M} = \mathcal{M}\Big(h_0, b_0, \mathbf{q}_\mu^{\mathrm{Haar}}, \mathbf{q}_\mu^{\mathrm{ED}}, \mathbf{q}_\mu^{\mathrm{EOH}}, \mathbf{q}_\mu^{\mathrm{NNEOH}}, \mathbf{q}_\mu^{\mathrm{HOG}}, \mathbf{q}_\mu^{\mathrm{LBP}}, \mathbf{q}_\mu^{\mathrm{MBLBP}}\Big)\,, \tag{2.18}$$

innerhalb des Trainingsfensters bestimmt, so kann die Liste $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ unter Verwendung der initialen Trainingsdaten $\mathcal{I}$, welche a priori aus dem Pool $\mathcal{P}_{\mathrm{Pos}}$ der positiven sowie $\mathcal{P}_{\mathrm{Neg}}$ der negativen Trainingsdaten gewählt werden, berechnet werden. Es gilt zu beachten, dass die initiale Berechnung von $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ der Merkmalsantworten auf allen Trainingsdaten nur einmalig durchgeführt werden muss und anschließend lediglich Antworten auf durch *Boostrap Aggregating* substituierten Trainingsbeispiele berechnet werden (siehe Abschnitt 2.5.4).

Das iterative Training der Stufen (starke Klassifikatoren) bis zur Gesamtstufenzahl S wird nun begonnen (äußerer gestrichelter Kasten in Abb. 2.3), wobei jede neue Stufe s mit der Anzahl $t = 0$ der Merkmale

(schwache Klassifikatoren) initialisiert wird. Den einzelnen Trainingsbildern $g_\iota(\mathbf{u})$ wird in der ersten Iteration für $s = 1$ und $t = 1$ ein gleichverteiltes Gewicht $\mathbf{w}_{\mu^*,\iota}(t)$ zugeordnet, während die für eine Stufe über die gewählten Merkmale kumulierten Gewichte der Trainingsbeispiele $\boldsymbol{\nu}_\iota$ genauso wie der Schwellenwert der Stufe $\tau(s, \mu, \boldsymbol{\nu}_\iota)$ mit null initialisiert werden. Die kumulierten Gewichte $\boldsymbol{\nu}_\iota$ werden nach der Auswahl jedes schwachen Klassifikators für all die Trainingsbeispiele, auf denen der Merkmalswert den Merkmalsschwellenwert überschreitet, um einen Wert $\alpha(t)$ erhöht, welcher als Merkmalsgüte interpretiert werden kann (siehe unten). Mit $\boldsymbol{\nu}_\iota^\downarrow$ werden die sortierten kumulierten Gewichte der Trainingsbilder bezeichnet, wobei durch $\check{\iota}$ die sortierten Indizes gegeben sind, d. h. $\boldsymbol{\nu}_\iota^\downarrow(1) = \boldsymbol{\nu}_\iota(\iota = \check{\iota}(1))$. Während des *Boosting*-Vorgangs (äußerer gestrichelter Kasten in Abb. 2.3) werden dem aktuell trainierten starken Klassifikator dann solange schwache Merkmale hinzugefügt, bis die geforderte *FAR* erreicht wird (siehe Abschnitt 2.5.2). Es gilt zu beachten, dass im Falle von eindimensionalen Merkmalsantworten die Liste $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ zu einer Matrix wird, bei der die Spalten die einzelnen Merkmale und die Zeilen die Merkmalsantworten auf die einzelnen Trainingsbeispiele repräsentieren. Sind genug schwache Klassifikatoren zu einem starken Klassifikator geschaltet, so wird die aktuell trainierte Stufe als Glied $\mathcal{K}_s$ der existierenden Kaskade $\mathcal{K}_{1,\ldots,s-1}$ angehängt. Ist die a priori vorgegebene Gesamtstufenzahl S noch nicht erreicht, so werden die auf Basis des mit der aktuellen Kaskade $\mathcal{K}_s$ anhand der aktuellen Trainingsdaten $\mathcal{I}_{\mathrm{Pos};s}, \mathcal{I}_{\mathrm{Neg};s}$ ermittelten FN und TN durch *Bagging* ersetzt (Abschnitt 2.5.4) und eine weitere Stufe trainiert.

2.5.2 *Boosting*

Der *Boosting*-Algorithms wird beim Aufbau der Kaskade genutzt, um aus der Liste der Merkmalsantworten auf allen Trainingsdaten $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ das diskriminativste Merkmal $\mathbf{m}_{\mu^*}(s,t)$ (anhand dessen Antwort) mit Index $\mu = \mu^*$, $\mu = \{1, \ldots |\mathcal{M}|\}$, als schwachen Klassifikator auszusuchen, um dieses anschließend mit weiteren Merkmalen zu einem starken Klassifikator zu *boosten*. Die Idee geht auf die von Kearns in [Kea88; KV94] gestellte Frage zurück, welche durch Arbeiten von Freund und Schapire [FS95] sowie von Viola und Jones [VJ01] durch Prägung des

Standes der Technik beantwortet wurde: Als Fragestellung hierzu lässt sich formulieren, ob es mit einem effizienten Lernalgorithmus möglich ist, mit schwachen Lernern (schwachen Klassifikatoren), die eine binäre Klassifikationsaufgabe gerade besser als der Zufall ($> 50\,\%$) lösen, einen beliebig genauen starken Klassifikator zu trainieren. Der Vorteil, der sich aus der Randbedingung für die schwachen Klassifikatoren hieraus ergibt, lässt sich beim Entwurf der Merkmale ausnutzen und spiegelt sich in der Vielfalt und Einfachheit der oben beschriebenen Merkmale wider.

Der *Boosting*-Vorgang, wie er für die Merkmalsauswahl implementiert wurde, ist in Abb. 2.7 skizziert. Es soll in diesem Abschnitt von eindimensionalen Merkmalsantworten ausgegangen werden, sodass die Liste $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ als Matrix behandelt werden kann, in der mit den Indizes μ, ι (Spalten, Zeilen), mit $\iota = \{1, \ldots |\mathcal{I}|\}$, auf eine Merkmalsantwort auf ein Trainingsbeispiel zugegriffen werden kann:

$$\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}\Big|_{\mu,\iota} = \mathbf{m}_{\mu,\iota}. \tag{2.19}$$

Die Auswahl des diskriminativsten Merkmals (diskriminativ hier im Sinne der besten Trennung zwischen positiven und negativen Trainingsbeispielen) geschieht anhand der Gewichte $\mathbf{w}_{\mu,\iota}(t)$ der Trainingsdaten. Dabei wird ein Fehler $\boldsymbol{\epsilon}_{\mu,\iota}(t)$ der Klassifikation für alle Merkmale ausgewertet, wofür die entsprechend des Merkmalswertes sortierten Gewichte der einzelnen Trainingsbeispiele aufsummiert werden (siehe Abschnitt 2.5.2.1). Der Fehler ist dabei für jedes Merkmal abhängig vom Index ι, welcher das Trainingsbeispiel und den dazugehörigen Merkmalswert indiziert, welches zur Klassifikationsentscheidung mit seinem Schwellenwert herangezogen wird. Da die Gewichte zu Beginn des Trainings jeder Stufe ($s = 1, t = 1$) gleichverteilt initialisiert werden und sich weiterhin durch das gesamte Training entwickeln, wird der erste schwache Klassifikator jeder Stufe ausschließlich durch die Fähigkeit der direkten Trennung der positiven und negativen Trainingsbeispiele gewählt.

Nachdem das diskriminativste Merkmal $\mathbf{m}_{\mu^*}(s,t)$ mit seinem Merkmalsschwellenwert $\Theta_{\mu^*}(t)$ (gewählte Merkmalsantwort) anhand seines Fehlers $\epsilon_{\mu^*,\iota}(t)$ gewählt (siehe Abschnitt 2.5.2.1) und das auf dem Fehler $\epsilon_{\mu^*,\iota}(t)$ basierende Gütegewicht des gewählten Merkmals $\alpha\big(\epsilon_{\mu^*,\iota}, t\big)$

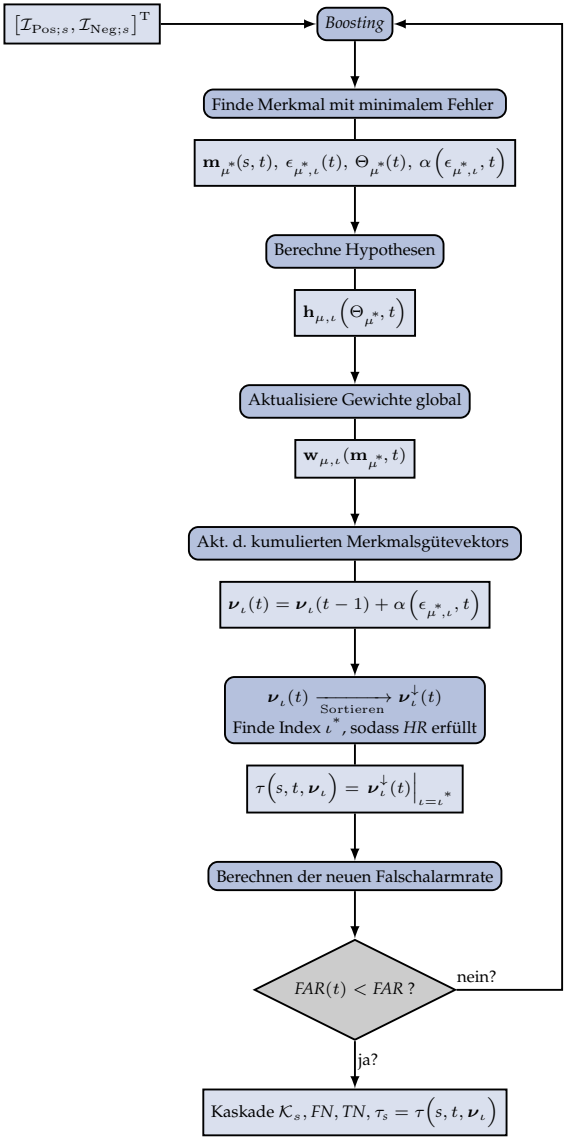


Abbildung 2.7 Schema des *Boosting*-Vorgangs, wie er durch den inneren gestrichelten Kasten in Abb. 2.3 skizziert ist.

bestimmt wurde (Abschnitt 2.5.2.3), können die Hypothesen $\mathbf{h}_{\mu^*,\iota}\left(\Theta_{\mu^*}, t\right)$ für alle Trainingsmerkmale bestimmt werden, deren Vergleich mit den Klassenzugehörigkeiten $\mathcal{Z}$ eine Aussage über die korrekte Klassifikation zulässt. Dabei bedeutet für ein einzelnes Trainingsbeispiel ein Hypothesenwert von 1, dass die Klassifikation mit der Kaskade bis zur aktuell trainierten Stufe korrekt, ein Wert von 0, dass die Klassifikation nicht korrekt war. Basierend auf den Hypothesen und dem gewählten Merkmal können dann die Gewichte $\mathbf{w}_{\mu^*,\iota}(t)$ der Trainingsdaten aktualisiert werden, wobei Gewichte korrekt klassifizierter Beispiele abgeschwächt werden. Für die Bestimmung der Verlustfunktionen zum Aktualisieren der kumulierten Trainingsgewichte wurden neben *Adaboost* auch *Gentle Adaboost* [FHT00] sowie *Real Adaboost* [SS98] implementiert. Während hier im Weiteren ausschließlich *Adaboost* eingesetzt wird und stets gemeint ist, wenn von *Boosting* gesprochen wird, sind Auswertungen der einzelnen Methoden, welche sich im Wesentlichen in der Berechnung der Verlustfunktion unterscheiden, in [Tia12] und [FHT00; Pra12] zu finden.

Anschließend wird im sortierten Vektor der kumulierten Gütegewichte der Beispiele $\boldsymbol{\nu}_\iota^\downarrow(t)$ der Index $\iota = \iota^*$ gewählt, der die vorgeschriebene *HR* erfüllt und dieser Wert als temporärer Stufenschwellenwert (für die aktuelle *Boosting*-Iteration) gewählt. In $\boldsymbol{\nu}_\iota(t)$ wird dann für all die Trainingsbeispiele mit Index ι der Wert um $\alpha\left(\epsilon_{\mu^*,\iota}, t\right)$ erhöht, deren Merkmalsmalsantworten den Schwellenwert $\Theta_{\mu^*}(t)$ des gewählten Merkmals überschreiten und anschließend wieder sortiert. Ist für den Wert

$$\tau\Big(s, t, \boldsymbol{\nu}_\iota\Big) = \boldsymbol{\nu}_\iota^\downarrow(t)\Big|_{\iota=\iota^*} \tag{2.20}$$

die *FAR* nicht erfüllt (während ι^* die *HR* stets in jeder Iteration erfüllt), so werden weitere schwache Klassifikatoren *geboostet*, bis die *FAR* erreicht wird oder ein Abbruch erfolgt, wenn der aktuelle Fehler $\boldsymbol{\epsilon}_{\mu,\iota}(t)$ für alle Merkmale größer als $0{,}5$ ist.

Die Idee bei der Auswahl durch Minimierung eines Fehlers ist es, unter Berücksichtigung des aktuell gewählten, geeignetesten Merkmals, basierend auf der Liste $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}_{s-1}|}$, die zuverlässig klassifizierten Trainingsbeispiele (der Wert der Merkmalsantwort dieses Merkmals ist weit entfernt vom Schwellenwert $\Theta_{\mu^*}(t)$ für das aktuell gewählte Merkmal $m_{\mu^*}(s,t)$) für die Wahl des nächsten schwachen Klassifizierers im

Gewicht abzuschwächen. Dadurch wird ermöglicht, dass sich künftige schwache Klassifikatoren mehr auf schwer zu klassifizierende Beispiele konzentrieren. Für diesen Ansatz der Merkmalsauswahl gilt Folgendes:

- Während $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ immer nur pro neuer Stufe s aktualisiert wird, werden die Gewichte $\mathbf{w}_{\mu^*,\iota}(t)$ nach der Auswahl jedes Merkmals $m_{\mu^*}(s,t)$ global, d. h. nach dieser Iteration festgelegt, aktualisiert.

Dieses Vorgehen ist gerade das die Frage von Kearns beantwortende Prinzip des *Boostings*: Es kann auf diese Weise durch Kombination schwacher Klassifikatoren ein beliebig genauer starker Klassifikator kreiert werden. Im Folgenden wird nun detaillierter auf die einzelnen Schritte des *Boostings*-Algorithmus eingegangen.

2.5.2.1 Finden des Merkmals mit minimalem Fehler

Für die Auswahl jedes schwachen Klassifikators wird die Liste aller Merkmalsantworten $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ für jedes Merkmal, das Vorzeichen beachtend, absteigend sortiert, Beispielbilder mit großen Merkmalsantworten unten anordnend:

$$\mathcal{L}_{\mu,\iota}^{\downarrow} := \mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|\downarrow}, \tag{2.21}$$

wobei bezüglich der Annotation analog zu oben gilt:

$$\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|\downarrow}\Big|_{\mu,\iota=1} = \mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}\Big|_{\mu,\iota=\check{\iota}(1)} = \min_{\iota} \mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}, \tag{2.22}$$

sowie

$$\check{\iota}(1) = \arg\min_{\iota} \mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}. \tag{2.23}$$

Dabei ist zu beachten, dass nach der Sortierung die Merkmalsantworten in jeder Spalte nicht mehr sukzessive, zunächst positive Trainingsbeispiele, dann negative, angeordnet sind. Basierend auf den entsprechend dem Index $\check{\iota}$ (zu unterscheiden von der Sortierung von $\boldsymbol{\nu}_{\iota}^{\downarrow}$) für jedes Merkmal absteigend sortierten Vektors der Gewichte der Trainingsbeispiele (hohe

Gewichte für die von $t = 0, \ldots, s-1$ klassifizierten Beispiele mit der aktuellen Kaskadenstufe)

$$\mathbf{w}^{\downarrow}_{\mu,\iota}(t) = \mathbf{w}_{\mu,\iota}(t)\big|_{\iota=\check{\iota}} \tag{2.24}$$

wird zur Wahl des geeignetesten schwachen Klassifikators mit dem Index $\mu = \mu^*$ aus allen Merkmalen das Minimum unter den mittels der Gewichte bestimmten Fehlern gesucht:

$$\mu^*(\iota) = \arg\min_{\mu} \boldsymbol{\epsilon}_{\mu,\iota}\left(\mathbf{w}^{\downarrow}_{\mu,\iota}(t)\right) , \tag{2.25}$$

$$\iota^* = \arg\min_{\iota} \boldsymbol{\epsilon}_{\mu^*,\iota}\left(\mathbf{w}^{\downarrow}_{\mu,\iota}(t)\right) , \tag{2.26}$$

wobei zu bemerken ist, dass $\boldsymbol{\epsilon}_{\mu,\iota}$ und $\mathbf{w}_{\mu,\iota}(t)$ matrixwertig und $\boldsymbol{\epsilon}_{\mu^*,\iota}$ sowie $\mathbf{w}_{\mu^*,\iota}(t)$ vektorwertig sind. Die Fehler geben ein Maß der Güte der Trennung von positiven und negativen Trainingsdaten, basierend auf den Gewichten der einzelnen Trainingsdaten, an. Hierbei ist zu bemerken, *dass ein minimaler Fehler nicht gleichzusetzen ist mit einer Trennung, welche die HR und FAR erfüllt*. Damit ergibt sich für das gewählte, diskriminanteste Merkmal $\mathbf{m}_{\mu^*}(s,t)$ ein zugehöriger Schwellenwert

$$\Theta_{\mu^*}(t) = \mathcal{L}^{|\mathcal{M}|\times|\mathcal{I}|\downarrow}_{\mu,\iota}\Big|_{\mu=\mu^*,\iota=\iota^*} , \tag{2.27}$$

mit welchem der minimale Fehler erreicht wird. Die Berechnung des Fehlers und das Suchen nach dem Minimum des Fehlers wurde nach Viola und Jones mit *Adaboost* implementiert [VJ01]. Um das Minimum des Klassifikationsfehlers über alle Merkmale zu finden, wird zunächst die Summe der Gewichte der positiven und negativen Beispielbilder gebildet:

$$\mathbf{w}^{\Sigma,\mathrm{Pos}}_{\mu} = \sum_{\iota\in\mathcal{I}_{\mathrm{Pos}}} \mathbf{w}_{\mu,\iota}(t) , \tag{2.28}$$

$$\mathbf{w}^{\Sigma,\mathrm{Neg}}_{\mu} = \sum_{\iota\in\mathcal{I}_{\mathrm{Neg}}} \mathbf{w}_{\mu,\iota}(t) . \tag{2.29}$$

Die Gewichte der positiven und negativen Trainingsbeispiele für jedes Merkmal mit Index μ sind jeweils Vektoren der Länge $|\mathcal{I}| = |\mathcal{I}_{\mathrm{Pos}}| + |\mathcal{I}_{\mathrm{Neg}}|$, wobei gilt:

$$\mathbf{w}_{\mu,\iota}^{\iota\in\mathcal{I}_{\mathrm{Pos}}} = \left[\mathbf{w}_{\mu,\iota_1}\ ,\ldots,\ \mathbf{w}_{\mu,\iota_{|\mathcal{I}_{\mathrm{Pos}}|}}\ ,\mathbf{0},\ldots,\mathbf{0}\right]^{\mathrm{T}}, \tag{2.30}$$

$$\mathbf{w}_{\mu,\iota}^{\iota\in\mathcal{I}_{\mathrm{Neg}}} = \left[\mathbf{0},\ldots,\mathbf{0},\ \mathbf{w}_{\mu,\iota_{|\mathcal{I}_{\mathrm{Pos}}|+1}}\ ,\ldots,\ \mathbf{w}_{\mu,\iota_{|\mathcal{I}|}}\right]^{\mathrm{T}}. \tag{2.31}$$

In Glg. (2.30), wie auch im Folgenden, sind die Indizes der einzelnen Gewichte für die Nummer des aktuellen schwachen Klassifikators t aus Gründen der Übersichtlichkeit weggelassen.

Es werden die Gewichte der einzelnen Trainingsbilder mit Index ι ihren zugehörigen Merkmalswerten entsprechend absteigend sortiert:

$$\mathbf{w}_{\mu,\iota}^{\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}} = \mathbf{w}_{\mu,\iota}^{\iota\in\mathcal{I}_{\mathrm{Pos}}}\Big|_{\iota=\check{\iota}(\iota)}, \tag{2.32}$$

$$\mathbf{w}_{\mu,\iota}^{\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Neg}}} = \mathbf{w}_{\mu,\iota}^{\iota\in\mathcal{I}_{\mathrm{Neg}}}\Big|_{\iota=\check{\iota}(\iota)}. \tag{2.33}$$

Die einzelnen Spaltenvektoren der sortierten Mermalsgewichte in den Matrizen (2.32) und (2.33) haben in *jenen* Zeilen ι einen Eintrag ungleich null, in denen $\check{\iota}(\iota) \in \mathcal{I}_{\mathrm{Pos}}$ bzw. $\check{\iota}(\iota) \in \mathcal{I}_{\mathrm{Neg}}$ gilt.

Damit taucht ein Eintrag in den sortierten Vektoren der Länge $|\mathcal{I}|$ in der Zeile, die dem Rang seines Merkmalswertes in $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|\downarrow}$ entspricht, auf:

$$\mathbf{w}_{\mu,\iota}^{\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}} = \left[\mathbf{w}_{\mu,\check{\iota}(1)}^{\iota\in\mathcal{I}_{\mathrm{Pos}}}\ ,\ldots,\ \mathbf{w}_{\mu,\check{\iota}(|\mathcal{I}|)}^{\iota\in\mathcal{I}_{\mathrm{Pos}}}\right]^{\mathrm{T}}, \tag{2.34}$$

$$\mathbf{w}_{\mu,\iota}^{\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Neg}}} = \left[\mathbf{w}_{\mu,\check{\iota}(1)}^{\iota\in\mathcal{I}_{\mathrm{Neg}}}\ ,\ldots,\ \mathbf{w}_{\mu,\check{\iota}(|\mathcal{I}|)}^{\iota\in\mathcal{I}_{\mathrm{Neg}}}\right]^{\mathrm{T}}. \tag{2.35}$$

Auf diese Weise sammeln sich in den ersten Zeilen von $\mathbf{w}_{\mu,\iota}^{\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}}$ und $\mathbf{w}_{\mu,\iota}^{\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Neg}}}$ die Gewichte der Trainingsbeispiele mit niedrigen Merkmalswerten (vorzeichenbehaftet) und entsprechend geringem Klassifikationsabstand zum Schwellenwert des Merkmals $\mathbf{m}_{\mu}$.

Es werden nun die kumulierten Summen

$$\mathbf{w}_{\mu,\iota}^{\Sigma\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}}, \quad \mathbf{w}_{\mu,\iota}^{\Sigma\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}} \tag{2.36}$$

gebildet, um einen Vektor zu erhalten, bei dem jeder Eintrag das summierte Gewicht aller Beispielbilder bis einschließlich des aktuellen Eintrages enthält, wobei gilt:

$$\mathbf{w}_{\mu,\iota}^{\Sigma\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}}\Big|_{\iota=|\mathcal{I}|} = \mathbf{w}_{\mu}^{\Sigma,\mathrm{Pos}}, \quad \mathbf{w}_{\mu,\iota}^{\Sigma\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Neg}}}\Big|_{\iota=|\mathcal{I}|} = \mathbf{w}_{\mu}^{\Sigma,\mathrm{Neg}}. \tag{2.37}$$

Um das Vorzeichen der Merkmalswerte zu berücksichtigen, je nachdem, ob die Merkmale der positiven Klasse oberhalb oder unterhalb eines Schwellenwertes liegen, werden zwei Fehler gebildet:

$$\boldsymbol{\epsilon}_{\mu,\iota}^{\mathrm{Pos}} = \mathbf{w}_{\mu,\iota}^{\Sigma\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}} - \mathbf{w}_{\mu,\iota}^{\Sigma\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Neg}}} + \mathbf{I}_{|\mathcal{M}|\times|\mathcal{I}|}\cdot\mathbf{w}_{\mu}^{\Sigma,\mathrm{Neg}}, \tag{2.38}$$

$$\boldsymbol{\epsilon}_{\mu,\iota}^{\mathrm{Neg}} = \mathbf{w}_{\mu,\iota}^{\Sigma\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Neg}}} - \mathbf{w}_{\mu,\iota}^{\Sigma\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}} + \mathbf{I}_{|\mathcal{M}|\times|\mathcal{I}|}\cdot\mathbf{w}_{\mu}^{\Sigma,\mathrm{Pos}}. \tag{2.39}$$

Das Addieren von $\mathbf{w}_{\mu}^{\Sigma,\mathrm{Neg}}$ bzw. $\mathbf{w}_{\mu}^{\Sigma,\mathrm{Pos}}$ hebt den Offset, der sich in den kumulierten Summen befindet, auf, und bewirkt, dass $\boldsymbol{\epsilon}_{\mu,\iota}^{\mathrm{Pos}}, \boldsymbol{\epsilon}_{\mu,\iota}^{\mathrm{Neg}} \geq 0$ gilt, damit eine Suche eines Minimums in einem Wertebereich ≥ 0 durchgeführt werden kann. Für Glg. (2.38) ergibt sich somit ein ausgeprägtes Minimum, wenn viele negative Beispielbilder nacheinander, also mit niedrigen Merkmalswerten, auftreten, bevor die Merkmalsantworten der positiven Trainingsbilder in der kumulierten Summe addiert werden. Im Falle von guter Trennung bei betragsmäßig niedrigen negativen Merkmalswerten findet sich ein Minimum in $\boldsymbol{\epsilon}_{\mu,\iota}^{\mathrm{Pos}}$, während sich für betragsmäßig niedrige Merkmalsanworten der positiven Beispielbilder ein Minimum in $\boldsymbol{\epsilon}_{\mu,\iota}^{\mathrm{Neg}}$ und entsprechend ein Schwellenwert, unterhalb dessen sich die Antworten der positiven Klasse befinden, finden lässt.

Es wird dazu die Matrix von Spaltenvektoren

$$\boldsymbol{\epsilon}_{\mu,\iota}\left(\mathbf{w}_{\mu,\iota}^{\downarrow}(t)\right) = \left[\boldsymbol{\epsilon}_{\mu,\iota}^{\mathrm{Pos}}, \boldsymbol{\epsilon}_{\mu,\iota}^{\mathrm{Neg}}\right]^{\mathrm{T}} \tag{2.40}$$

gebildet und durch Glg. (2.25) und (2.26) das diskriminanteste Merkmal gesucht.

Es gilt zu beachten, dass bei der Suche nach dem Minimum über ι noch berücksichtigt werden muss, ob $\iota^* > |\mathcal{I}|$ ist, in jenem Fall gilt $\iota^* = \iota^* - |\mathcal{I}|$ und die Merkmalswerte der positiven Klassen liegen *unterhalb* des gewählten Schwellenwertes.

Der Rechen- und Speicheraufwand der Methode ist anhand der vorangehenden Gleichungen gut ersichtlich. Bevor der minimale Fehler gesucht wird, muss jede Spalte $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ sortiert werden. Dies bringt, ausgehend von skalaren Merkmalswerten, bei einem typischen Training mit $|\mathcal{I}| = 5000$ sowie Trainingsbildern der Größe 36×36 Pixel, einer Verschiebung der Basis-Merkmale um 1 Pixel sowie einer Skalierung von 1,5 und ausschließlich Haar-Merkmalen, bei einer Anzahl von 300510 Merkmalen neben großem Rechenaufwand für das Sortieren einen Speicheraufwand von 6 GB bei einfacher Präzision der Daten mit sich.

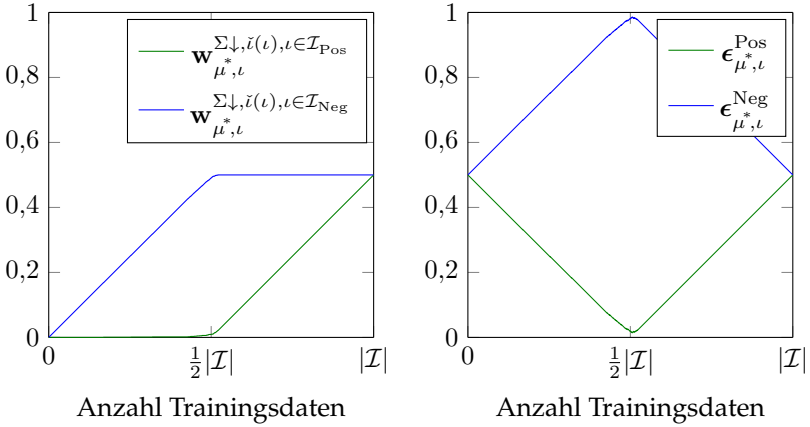
Anschaulich soll der Vorgang im Folgenden anhand der ausgewählten Merkmale für das Training eines Kaskadenklassifikators mit Haar-Merkmalen gezeigt werden, wobei hier $|\mathcal{I}_{\mathrm{Pos}}| = |\mathcal{I}_{\mathrm{Neg}}|$ gilt.

2.5.2.2 Beispielhafte Merkmalsauswahl

Zu Beginn des Trainings trennt das erste Merkmal der ersten Stufe die Trainingsdaten nahezu ideal, der Fehler für das gewählte Merkmal $\boldsymbol{\epsilon}_{\mu^*,\iota}^{\mathrm{Pos}}$ ist, wie in Abb. 2.8 dargestellt, fast null.

Die Gewichte sind für $t = 1$ gleichverteilt, was an den jeweils maximalen kumulierten Gewichten von 0,5 für eine gleiche Anzahl negativer und positiver Trainingsdaten zu erkennen ist. Die negativen Trainingsbeispiele zeigen niedrige Merkmalswerte, die Kurve der kumulierten Gewichte der negativen Trainingsbeispiele ist früh steigend, während die kumulierten Gewichte der positiven Trainingsbeispiele erst bei der Hälfte der Anzahl aller Trainingsbeispiele (hier sind nahezu alle Antworten der negativen Trainingsbeispiele kleiner als die der positiven) beginnen, von null wesentlich unterschiedliche Werte anzunehmen. Es ist möglich, einen Schwellenwert zu finden, oberhalb dessen nahezu alle Antworten der Instanzen der positiven Klasse liegen. Dies entspricht vielen Nulleinträgen in den Zeilen $\iota = \{1 \dots \mathcal{I}_{\mathrm{Pos}}\}$ in $\mathbf{w}_{\mu^*,\iota}^{\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}}$.
Obwohl das gewählte Merkmal alleine bereits die geforderte *HR* sowie eine sehr niedrige *FAR* erfüllt und die Klassen nahezu perfekt trennt, ist das Training lediglich auf die für diese Stufe gewählten Daten $\mathcal{I}_{\mathrm{Pos}}$ und $\mathcal{I}_{\mathrm{Neg}}$ angepasst. Das Sehen weiterer Trainingsdaten ist nötig, um eine bessere Generalisierungsfähigkeit auf ungesehene Daten und somit das Bewältigen einer hohen *Intraklassenvarianz* bei der späteren Detektion zu ermöglichen. Dies wird durch *Bootstrap Aggregating* (siehe Kap. 2.5.4)

und das Trainieren weiterer Stufen realisiert, bei dem die Anzahl gesehener Daten, insbesondere negativer Trainingsbeispiele, deutlich erhöht wird (am Ende des Trainings typischerweise im Bereich $100 \cdot 10^6$). An der Auswahl weiterer Merkmale höherer Stufen soll dies verdeutlicht werden.



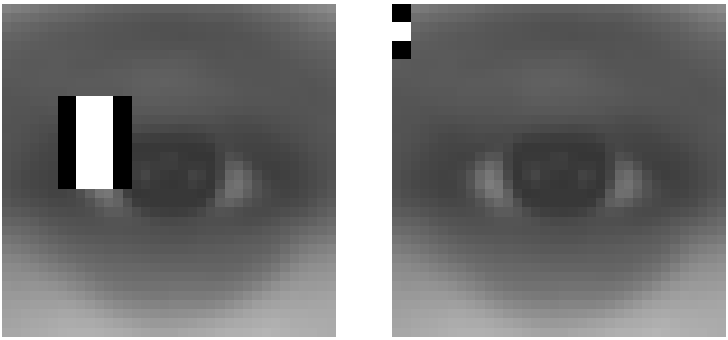
(a) Kumulierte Gewichte nach Sortierung anhand des Merkmalswertes.

(b) Fehler basierend auf den Gewichten der Trainingsdaten.

Abbildung 2.8 Erster schwacher Klassifikator ($t = 1$) in der ersten Stufe ($s = 1$) einer mit Haar-Merkmalen trainierten Kaskade. Nahezu ideale Trennung der Trainingsdaten anhand des ausgesuchten Merkmals.

Nach der Auswahl des ersten Merkmals werden die Gewichte aller Trainingsdaten angepasst, wobei Beispiele, deren Merkmalsantwort eine große Distanz zum gewählten Schwellenwert haben, abgeschwächt werden (Kap. 2.5.2.3).

Als weiteres Merkmal soll hier die Auswahl des dritten schwachen Klassifikators der sechsten Stufe derselben Kaskade gezeigt werden. Das gewählte sowie ein weiteres, nicht geeignetes Merkmal sind vor dem Hintergrund eines aus 3280 Augenbildern gemittelten Bildes in Abb. 2.9(a) dargestellt. Die Kurve der kumulierten Gewichte der negativen Trainingsbeispiele steigt für das durch *Adaboost* gewählte Merkmal stärker als das der positiven an (Abb. 2.10(a)).



(a) Haar-Merkmal entsprechend Abb. 2.10 und 2.11.

(b) Haar-Merkmal entsprechend Abb. 2.12 und 2.13.

Abbildung 2.9 Gewähltes und ein nicht gewähltes Merkmal als sechsten schwachen Klassifikator der Stufe 3. Die Merkmale sind in einem Trainingsfenster der Größe 36×36 Pixel dargestellt.

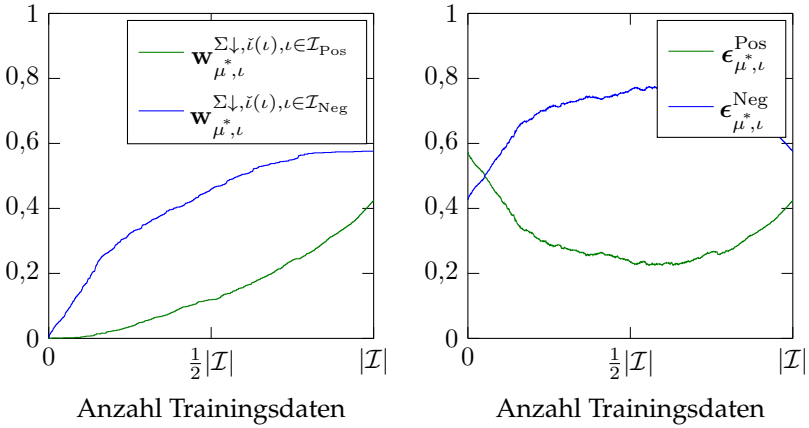
Dies spricht für viele Negativbeispiele mit niedrigen Merkmalsantworten sowie großen Gewichten und ermöglicht somit das Finden eines Mininums in $\epsilon_{\mu,\iota}^{\text{Pos}}$ sowie eines Schwellenwertes, oberhalb dessen die Antworten der positiven Trainingsbeispiele liegen. Die Abbildungen 2.10 und 2.11 zeigen die Gewichte, Fehler sowie Merkmalswerte.

Schlechter als das zuvor diskutierte, gewählte Merkmal trennt ein weiteres Merkmal (Abb. 2.9(b)), für welches in Abb. 2.12 die kumulierten Gewichte und Fehler sowie der zugehörige Schwellenwert in Abb 2.13 gezeigt sind. Für die Merkmalsauswahl nimmt der Fehler $\epsilon_{\mu,\iota}^{\text{Neg}}$ ein Minimum an.

Es soll nun das Update der Gewichte für den *Adaboost*-Algorithmus diskutiert werden.

2.5.2.3 Update der Gewichte der Trainingsbeispiele

Die Gewichte der Trainingsbeispiele werden anhand der Hypothesen $\mathbf{h}_{\mu^*,\iota}(\Theta_{\mu^*}, t) \in \mathcal{Z}$ des aktuell gewählten Merkmals aktualisiert, wobei die korrekt klassifizierten Trainingsbeispiele $\mathbf{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Pos}}}(\Theta_{\mu^*}, t) = 1$ für die positiven und $\mathbf{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Neg}}}(\Theta_{\mu^*}, t) = -1$ für die negativen Beispiele mit Hilfe



(a) Kumulierte Gewichte nach Sortiertung anhand des Merkmalswertes.

(b) Fehler basierend auf der Gewichten der Trainingsdaten.

Abbildung 2.10 Dritter schwacher Klassifikator ($t=3$) in der sechsten Stufe ($s=6$). Die Gewichte sind nicht mehr gleichverteilt.

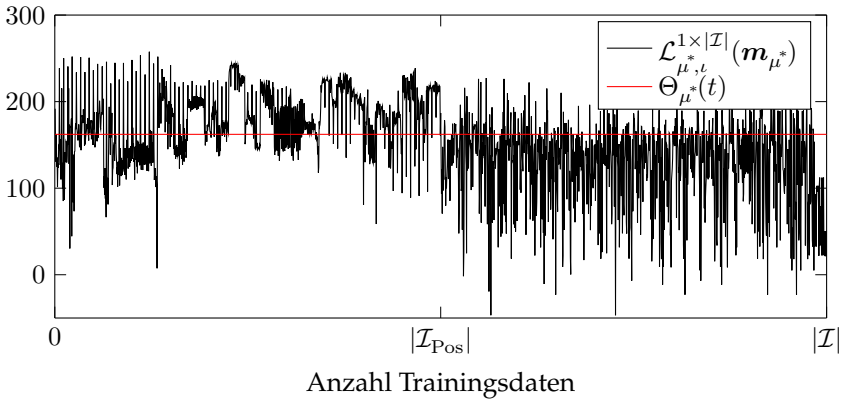
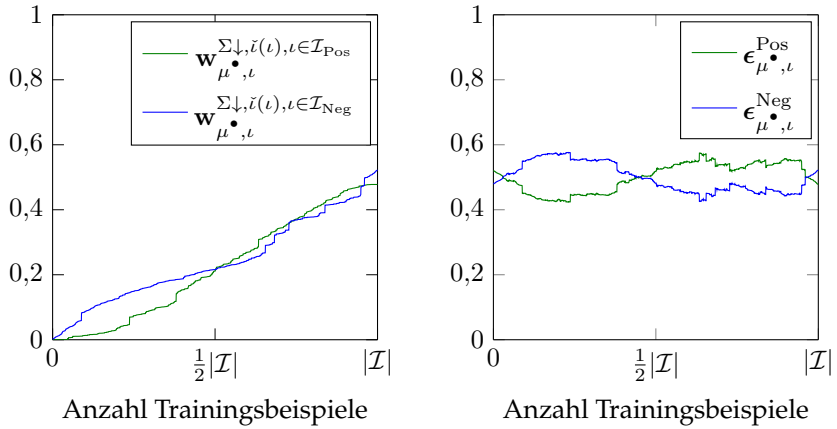


Abbildung 2.11 Merkmalswerte $\mathcal{L}_{\mu^*,\iota}^{1\times|\mathcal{I}|}(\boldsymbol{m}_{\mu^*})$ des dritten gewählten Merkmals ($t=3$) in der sechsten Kaskadenstufe ($s=6$) sowie Schwellenwert $\Theta_{\mu^*}(t)$ des Merkmals. Trainingsbeispiele, deren Merkmalswert oberhalb des Schwellenwertes liegen (Minimum des Fehlers wird durch $\boldsymbol{\epsilon}_{\mu,\iota}^{\text{Neg}}$ bestimmt), werden als positive Klasse erkannt.



(a) Kumulierte Gewichte nach Sortiertung anhand des Merkmalswertes.

(b) Fehler basierend auf der Gewichten der Trainingsdaten.

Abbildung 2.12 Die Trainingsdaten schlecht trennendes Merkmal, welches in der dritten Iteration der sechsten Stufe nicht als schwacher Klassifizierer gewählt wird. Die Gewichte sind nicht gleichverteilt.

einer Verlustfunktion

$$\beta\big(\epsilon_{\mu^*,\iota}, t\big) = \frac{\epsilon_{\mu^*,\iota}(t)}{1 - \epsilon_{\mu^*,\iota}(t)} \tag{2.41}$$

abgeschwächt werden, indem die Gewichte aktualisiert,

$$\mathbf{w}_{\mu^*,\iota}(t,s)\big|_{\boldsymbol{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Pos}}}=1} = \mathbf{w}_{\mu^*,\iota}(t-1,s)\big|_{\boldsymbol{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Pos}}}=1} \cdot \beta\big(\epsilon_{\mu^*,\iota}, t\big)\ , \tag{2.42}$$

$$\mathbf{w}_{\mu^*,\iota}(t,s)\big|_{\boldsymbol{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Neg}}}=-1} = \mathbf{w}_{\mu^*,\iota}(t-1,s)\big|_{\boldsymbol{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Neg}}}=-1} \cdot \beta\big(\epsilon_{\mu^*,\iota}, t\big)\ , \tag{2.43}$$

und anschließend normiert werden

$$\mathbf{w}_{\mu^*,\iota}(t,s) = \frac{\mathbf{w}_{\mu^*,\iota}(t,s)}{\sum_\iota \mathbf{w}_{\mu^*,\iota}(t,s)}\ . \tag{2.44}$$

Es stehen hierbei $\mathbf{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Pos}}}(\Theta_{\mu^*}, t)$ und $\mathbf{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Neg}}}(\Theta_{\mu^*}, t)$ für die Hypothesen der positiven bzw. negativen Trainingsdaten. Die Verlustfunktion $\beta\big(\epsilon_{\mu^*,\iota}, t\big)$ wird im *Adaboost*-Algorithmus in Abhängigkeit des Klassifizierungsfehlers des aktuellen Merkmals $\mathbf{m}_{\mu^*}(s,t)$ bestimmt, wobei der Abstand der

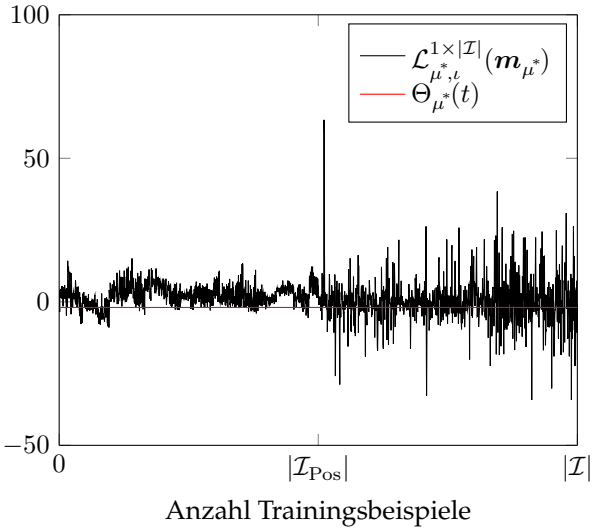


Abbildung 2.13 Merkmalswerte $\mathcal{L}^{1\times|\mathcal{I}|}_{\mu^*,\iota}(\boldsymbol{m}_{\mu^*})$ des schlecht trennenden Merkmals in der dritten Iteration in der sechsten Kaskadenstufe sowie Schwellenwert $\Theta_{\mu^*}(t)$ des Merkmals.

einzelnen Merkmalsantworten vom Schwellenwert nur indirekt über das Gewicht eingeht.
Der ebenfalls implementierte *Real Adaboost* besitzt eine quadratische Kostenfunktion, anhand deren das die Daten am besten trenndende Merkmal gesucht wird. Der Name leitet sich von der Berechnung der für die Hypothesen angenommenen Werte ab (reellwertig), beim *Real Adaboost* wird pro Beispiel ein links- sowie rechtsseitiger Wert für die Hypothese bestimmt:

$$\boldsymbol{h}_{\mu^*,\iota}(t) = \begin{cases} \boldsymbol{h}_\iota^{\text{links}}, & \mathcal{L}_{\mu^*,\iota}(\boldsymbol{m}_{\mu^*}) < \Theta_{\mu^*} \\ \boldsymbol{h}_\iota^{\text{rechts}}, & \text{sonst}, \end{cases} \tag{2.45}$$

wobei $\boldsymbol{h}_\iota^{\text{links}}$ und $\boldsymbol{h}_\iota^{\text{rechts}}$ entsprechend (2.45) reellwertige Werte annehmen im Vergleich zu *Adaboost*.

Der *Gentle Adaboost*-Algorithmus erweitert *Real Adaboost* dahingehend, dass die links- und rechtsseitigen Werte der Hypothesen im Bereich $[-1, 1]$ liegen. Für beide gilt für die Verlustfunktion

$$\mathbf{w}_{\mu^*,\iota}(t,s)\big|_{\boldsymbol{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Pos}}=1}} = \mathbf{w}_{\mu^*,\iota}(t-1,s)\big|_{\boldsymbol{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Pos}}=1}} \cdot \mathrm{e}^{\boldsymbol{h}_{\mu^*,\iota}(t)}\,, \tag{2.46}$$

$$\mathbf{w}_{\mu^*,\iota}(t,s)\big|_{\boldsymbol{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Neg}}=-1}} = \mathbf{w}_{\mu^*,\iota}(t-1,s)\big|_{\boldsymbol{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Neg}}=-1}} \cdot \mathrm{e}^{\boldsymbol{h}_{\mu^*,\iota}(t)} \tag{2.47}$$

mit jeweils unterschiedlich bestimmten $\boldsymbol{h}_{\iota}^{\text{links}}, \boldsymbol{h}_{\iota}^{\text{rechts}}$.

2.5.2.4 Aktualisierung der kumulierten Merkmalsgüte $\boldsymbol{\nu}_{\iota}(t)$ zur Bestimmung des Stufenschwellenwertes

Nachdem ein schwacher Klassifikator gewählt wurde, wird der Merkmalsgütevektor $\boldsymbol{\nu}_{\iota}(t)$ für die korrekt klassifizierten Trainingsbeispiele mit der Güte des gewählten Merkmals

$$\alpha\left(\epsilon_{\mu^*,\iota},t\right) = -\log\left(\beta\left(\epsilon_{\mu^*,\iota},t\right)\right) = -\log\left(\frac{\epsilon_{\mu^*,\iota}(t)}{1-\epsilon_{\mu^*,\iota}(t)}\right) \tag{2.48}$$

aktualisiert:

$$\boldsymbol{\nu}_{\iota}(t)\big|_{\boldsymbol{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Pos}}=1}} = \boldsymbol{\nu}_{\iota}(t-1)\big|_{\boldsymbol{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Pos}}=1}} + \alpha\left(\epsilon_{\mu^*,\iota},t\right)\,, \tag{2.49}$$

$$\boldsymbol{\nu}_{\iota}(t)\big|_{\boldsymbol{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Neg}}=-1}} = \boldsymbol{\nu}_{\iota}(t-1)\big|_{\boldsymbol{h}_{\mu^*,\iota}^{\mathcal{I}_{\text{Neg}}=-1}} + \alpha\left(\epsilon_{\mu^*,\iota},t\right)\,. \tag{2.50}$$

Hierbei kann $\alpha\left(\epsilon_{\mu^*,\iota},t\right)$ als Klassifikationsgüte des Merkmals verstanden werden, wobei große Werte einer gute Trennung des Trainingsdatensatzes mit dem Merkmal $\mathbf{m}_{\mu^*}$ entsprechen. Da beim Prinzip des *Boostings* die *HR* für jeden schwachen Klassikator erfüllt sein muss und anhand der korrekt klassifizierten Beispiele ermittelt wird, wird auch $\boldsymbol{\nu}_{\iota}(t)$ für ausschließlich korrekt klassifizierte Beispiele aktualisiert.

2.5.2.5 Finden des Stufenschwellenwertes $\tau\left(s,t,\boldsymbol{\nu}_{\iota}\right)$ in $\boldsymbol{\nu}_{\iota}(t)$

Nach jeder Auswahl eines schwachen Merkmals und dem Ausführen von Glg. (2.49) und (2.50) für die korrekt klassifizierten Trainingsdaten wird der Vektor $\boldsymbol{\nu}_{\iota}(t)$ für die Trainingsbeispiele sortiert:

$$\boldsymbol{\nu}_{\check{\iota}}^{\downarrow}(t) := \boldsymbol{\nu}_{\iota}^{1\times|\mathcal{I}|\downarrow}(t) = \boldsymbol{\nu}_{\iota}^{1\times|\mathcal{I}|}(t)\Big|_{\iota=\check{\iota}(\iota)}\,, \tag{2.51}$$

$$\boldsymbol{\nu}_{\iota}^{\downarrow}(t) = \left[\boldsymbol{\nu}_{\check{\iota}(1)}^{\downarrow} \leq \cdots \leq \boldsymbol{\nu}^{\downarrow}{}_{\check{\iota}(|\mathcal{I}_{\mathrm{Pos}}|)}\right]^{\mathrm{T}}\,. \tag{2.52}$$

Der Stufenschwellenwert wird nun in der sortierten Liste für die positiven Trainingsbeispiele $\iota \in \mathcal{I}_{\mathrm{Pos},s}$ in

$$\boldsymbol{\nu}_{\iota}^{\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}}(t) \tag{2.53}$$

gesucht, sodass die vorgegebene *HR* erreicht wird:

$$\tau\Big(s,t,\boldsymbol{\nu}_{\iota}\Big) = \boldsymbol{\nu}_{\iota}^{\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}}\Big|_{\iota=\check{\iota}(\iota^*)}\,, \tag{2.54}$$

$$\check{\iota}(\iota^*) = \lceil(1-HR)\cdot|\mathcal{I}_{\mathrm{Pos}}|\rceil \;\wedge\; \boldsymbol{\nu}_{\iota}^{\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}}(\check{\iota}(\iota^*)-1) < \boldsymbol{\nu}_{\iota}^{\downarrow,\check{\iota}(\iota),\iota\in\mathcal{I}_{\mathrm{Pos}}}(\check{\iota}(\iota^*))\,. \tag{2.55}$$

Der *Stufenschwellenwert* für die kumulierte Merkmalsgüte $\tau\Big(s,t,\boldsymbol{\nu}_{\iota}\Big)$ wird während des Trainings einer Stufe nach der Auswahl jedes schwachen Klassifikators entsprechend Glg. (2.54) und (2.55) aktualisiert, *sodass* stets die *HR erfüllt* ist. Die *FAR* hingegen wird dabei dann als *Kriterium* für das vollständige *Beenden* des Training *der Stufe* verwendet.

2.5.2.6 Berechnen der neuen Falschalarmrate

Die Falschalarmrate wird als Kriterium zum Abbruch bzw. zur Entscheidung des Trainings eines weiteren Merkmals innerhalb einer Stufe s herangezogen. Nach dem Hinzufügen jedes Merkmals zu einer Stufe wird die Falschalarmrate neu evaluiert, indem die als fälschlich als positiv klassifizierten negativen Trainingsbeispiele *FP* in Abhängigkeit aller negativen Trainingsbeispiele berechnet werden:

$$FAR(t) = \frac{\sum\limits_{\iota\in\mathcal{I}_{\mathrm{Neg}}} 1(\iota) : \boldsymbol{\nu}_{\iota}(t) > \tau\Big(s,t,\boldsymbol{\nu}_{\iota}\Big)}{|\mathcal{I}_{\mathrm{Neg}}|}\,. \tag{2.56}$$

Da, falls die kumulierten und sortierten Schwellenwerte $\boldsymbol{\nu}_{\iota}^{\downarrow}(t)$ um den Index aus Glg. (2.55) den gleichen Wert haben, der Index $\check{\iota}^*$ bezüglich

der absteigend sortierten Liste solange nach oben gesetzt wird, bis der nächstgelegene Wert echt größer ist, wird somit die *HR* erhöht. Es wird damit ein kleinerer Schwellenwert zugelassen und somit mehr Trainingsbilder als Instanzen der positiven Klasse klassifiziert. Die tatsächliche *HR* ergibt sich dann zu:

$$HR = \frac{|\mathcal{I}_{\text{Pos}}| - \iota^* + 1}{|\mathcal{I}_{\text{Pos}}|} \,. \tag{2.57}$$

Da der Schwellenwert für die Stufe herabgesetzt wird, kann dies somit auch Einfluss auf die *FAR* haben.

2.5.3 *Merkmals-Boosting*

Wie bereits in Abschnitt 2.5.1 diskutiert, wurden in dieser Arbeit sowohl mehr- als auch eindimensionale Merkmalstypen verwendet. Im Beispiel in Abschnitt 2.5.2.1 wurde dargelegt, wie das Verwenden eindimensionaler Mehrmalsantworten neben der einfacheren Handhabung durch skalarwertige Einträge in $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ auch Speicherplatzvorteile bietet, wenn pro Merkmal nur eine statt mehrerer Dimensionen im Merkmalsvektor gespeichert werden müssen. Aus diesem Grund soll in diesem Abschnitt eine in dieser Arbeit entworfene Methode zur Merkmalsreduktion vorgestellt werden, mit Hilfe derer mehrdimensionale Merkmalsantworten mit Hilfe von *Boosting* auf einen niederdimensionalen Raum projiziert werden. Weiterhin wurde in dieser Arbeit auch die LDA zur Merkmalsreduktion implementiert. Details zu den Ergebnissen mit der LDA sind in [Bis06; Sau13] zu finden.
Um bei der Merkmalsreduktion eine Überanpassung der Methode an die Trainingsdaten zu vermeiden, muss neben dem ursprünglichen Trainingsdatensatz zunächst ein zum eigentlichen Trainingsdatensatz disjunkter Datensatz zur Dimensionsreduktion herangezogen werden.

Gegeben sei ein mehrdimensionales Merkmal $\mathbf{m}_\mu$ mit den Dimensionen $k = 1, \ldots, K$:

$$\mathbf{m}_\mu^{|K|\times 1}\big(\varrho_\rho(\mathbf{u}), \mathbf{q}_\mu\big) = [m_1, m_2, m_3, \ldots, m_K]^{\mathrm{T}} \,. \tag{2.58}$$

Es werden nun die Merkmalsantworten jeder einzelnen Dimension auf den Trainingsdatensatz für die Dimensionssreduktion $\mathcal{I}_{\mathrm{Red}}$ berechnet:

$$\mathcal{L}_{\mu,\iota}^{K\times|\mathcal{I}_{\mathrm{Red}}|} = \left[\mathcal{L}_{\mu_k,\iota}^{1\times|\mathcal{I}_{\mathrm{Red}}|}(m_1), \ldots, \mathcal{L}_{\mu_k,\iota}^{1\times|\mathcal{I}_{\mathrm{Red}}|}(m_K)\right]^{\mathrm{T}}. \tag{2.59}$$

Anschließend werden zwei Vektoren der Dimension $K \times 1$ für die Gewichtung der einzelnen Dimensionen $\mathbf{a}_{\mu_k}^{K\times 1}$ und deren einzelne skalare Schwellenwerte erstellt sowie eine Hypothesenmatrix $\mathbf{H}^{|\mathcal{I}_{\mathrm{Red}}|\times|K|}$. Anhand von $\mathcal{I}_{\mathrm{Red}}$ wird nun eine Merkmalsauswahl, wie in Abschnitt 2.5.2.1 diskutiert, durchgeführt. Es wird hierzu für jede Dimension der minimale Fehler wie in Glg. (2.38), (2.39), (2.40) bestimmt. Der Schwellenwert jeder Dimension ergibt sich gerade wie in Glg. (2.27). Anschließend wird die Gewichtung jeder einzelnen Dimension des Merkmals durch

$$\alpha_{k_\mu}(\epsilon_{\mu_k}) = -\log\left(\frac{\epsilon_{\mu_k}}{1-\epsilon_{\mu_k}}\right) \tag{2.60}$$

bestimmt. Hierbei ist ϵ_{μ_k} der Klassifizierungsfehler der Dimension k des in der Dimension zu reduzierenden Merkmals $\mathbf{m}_{\mu}^{|K|\times 1}(\varrho_\rho(\mathbf{u}), \mathbf{q}_\mu)$.

Nach einer Normalisierung der Gewichte α_k wird dann die Liste der skalarwertigen Mermalsantworten

$$\mathcal{L}_{\mu,\iota}^{1\times|\mathcal{I}_{\mathrm{Red}}|}\left(\mathbf{m}_{\mu}^{|K|\times 1}\right) = \mathbf{H}^{|\mathcal{I}_{\mathrm{Red}}|\times|K|} \cdot \mathbf{a}_{\mu_k}^{|K|\times 1}\Big|_{\mu,\iota} \tag{2.61}$$

bestimmt. Im Parametervektor $\mathbf{q}_\mu$ des Merkmals wird dann der Vektor der Gewichte sowie der Vektor der Schwellenwerte aufgenommen. Eine einfache *Brute Force*-Implementierung sieht die Auswahl lediglich der k' signifikantesten Merkmalsdimensionen vor und ist in der Implementierung des hier besprochenen Rahmenwerks verfügbar.

Der große Vorteil des vorgestellten Verfahrens ist die Abkehr von einer Gleichverteilung aller Dimensionen eines Merkmals. Mittels Boosting werden dominante Richtungen, in Abhängigkeit der Position und Größe des Merkmals innerhalb des Trainingsfensters, identifiziert und mit entsprechenden Schwellenwerten und Gewichten versehen. Weiterhin zeichnet sich der Ansatz durch seine Fähigkeit zu generalisieren aus. Während bei der ursprünglichen Anwendung von mehrdimensionalen (HOG-) Merkmalen eine aufwändige Suche nach der idealen Parametrierung der Zellen- und Blockgröße der HOG-Merkmale in Abhängigkeit

der Trainingsdaten durchgeführt werden muss, werden hier die für den Anwendungsfall deskriptivsten Merkmale in Abhängigkeit von Position und Skalierung aus dem Merkmalspool durch *Boosting* bestimmt und damit automatisch mit Hilfe des Auswahlalgorithmus dem Trainingsdatensatz entsprechend ideal gewählt.

2.5.4 Bootstrap Aggregating (Bagging)

Unter dem Begriff des *Bootstrap Aggregating* (auch *Bagging*) wird in der Statistik generell das Berechnen einer Statistik basierend auf einer Vielzahl von Stichproben, welche mit Zurücklegen aus einer Gesamtzahl an Trainingsbeispielen gezogen werden, bezeichnet. In dieser sowie in Vorarbeiten wird unter *Bootstrap Aggregating* das Training des Kaskadenklassifikators basierend nicht nur auf den initialen Trainingsdaten $\mathcal{I}$, sondern basierend auf einem Pool $\mathcal{P}_{\mathrm{Pos}}, \mathcal{P}_{\mathrm{Neg}}$ von Trainingsdaten, der ein Vielfaches der initialen Trainingsdaten umfassen kann, verstanden.

Das grundlegende Vorgehen und die Einbindung in das Trainingsrahmenwerk ist in Abb. 2.14 zu sehen. Wie in Kap. 2.5.2 diskutiert, resultiert durch Wahl des Schwellenwertes für die Stufe $\tau(s, t, \boldsymbol{\nu}_\iota)$ eine Anzahl von *TN* und *FN* basierend auf den aktuellen Trainingsdaten. Während die *TN* korrekt klassifizierte Beispiele der negativen Klasse kennzeichnen und diese sich aus der aktuellen Liste der Güte der Merkmals $\boldsymbol{\nu}_\iota(t)$ ergeben, deren Wert für eine negative Klassenzugehörigkeit kleiner als der Stufenschwellenwert ist, lassen sich die fälschlicherweise als negative klassifizierten positiven Trainingsbeispiele wie folgt finden. Es wird hierzu der sortierte Vektor der Merkmalsgüte nur für die Indizes der positiven Trainingsbeispiele $\boldsymbol{\nu}_\iota^{\downarrow,\iota\in\mathcal{I}_{\mathrm{Pos}}}(t)$ betrachtet. Da ι^* in Glg. (2.54) und (2.55) innerhalb der Indizes der positiven Trainingsbeispiele so gewählt wird, dass die *HR* gerade erfüllt wird, stellen die Einträge $\iota = \{\check{\iota}(1), \ldots, \check{\iota}(\iota^*)\}$ die positiven Trainingsbeispiele, die fälschlicherweise als negativ klassifiziert werden, dar.

Der Kern des *Bagging* besteht nun darin, die Gesamtmenge der Trainingsdaten dadurch zu erhöhen, dass entsprechend der Klassifikation mit der aktuellen Kaskade und der Identifikation ihrer Indizes im Trainingsdatenvektor, die *TN* und *FN* aus dem Pool der negativen bzw. positiven Trainingsdaten ersetzt werden. Nach dem Ersetzen der *TN* und *FN* durch das *Bagging* wird die aktuelle Liste der Merkmalsantwor-

ten $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}_\mathrm{s}|}$ auf allen Trainiingsdaten aktualisiert (es müssen lediglich die Antworten auf den ersetzten Bilder berechnet werden) und das *Boosting* einer neuen Stufe kann beginnen.

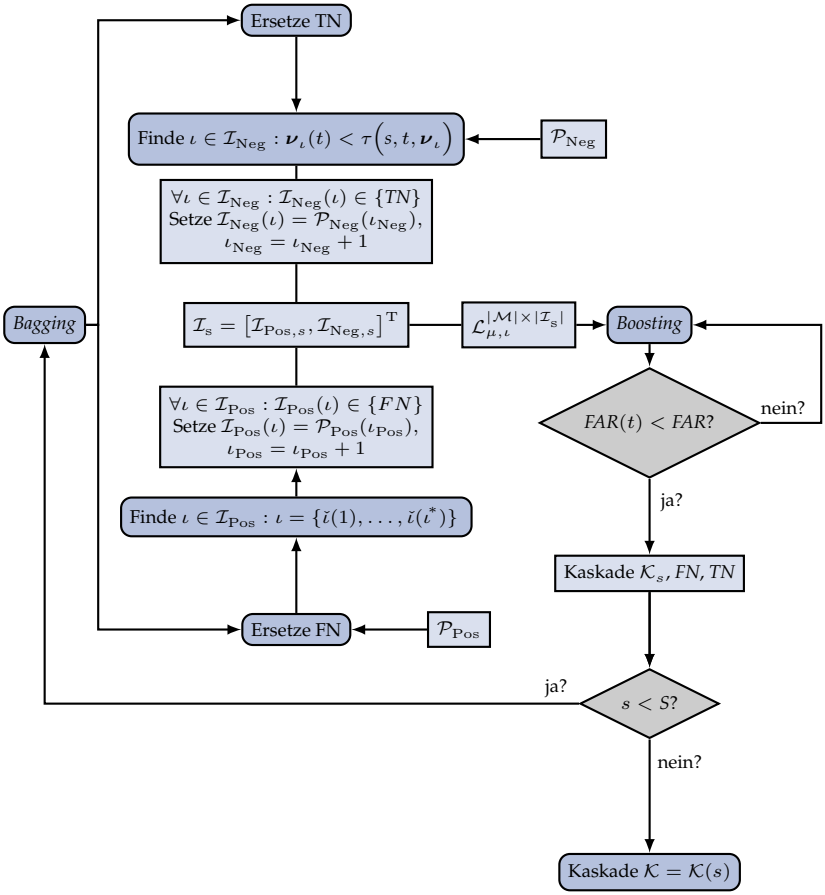


Abbildung 2.14 Übersicht *Boostrap Aggregating* (*Bagging*) und Schnittstelle zum Rahmenwerk.

2.5.5 *Merkmals-Bagging*

In Abschnitt 2.5.2.1 wurde bei der Diskussion der Kreierung der einzelnen Merkmale aus den Basis-Merkmalen der verschiedenen Merkmalstypen bereits der große Speicheraufwand erwähnt. Um den Ansatz des *Boosting* auszuschöpfen und gleichzeitig das Problem des Findens geeigneter Merkmale anzugehen, sollen so viele Merkmale wie möglich im Merkmalspool erzeugt werden, wobei gleichzeitig durch Berücksichtigen der verschiedenen Merkmalstypen die Stärken der einzelnen Merkmale ideal kombiniert werden sollen.

Das Problem, das sich beim Trainieren eines Multi-Merkmals-Klassifikators stellt, ist das des Speicherplatzes. Bei sinnvoller Parametrierung mit einer Translation von einem Pixel und einer Skalierung von 1,5 ergibt sich für ein Trainingsfenster von 36×36 Pixeln allein 6 GB Speicherbedarf nur für Haar-Merkmale, wie in Abschnitt 2.5.2.1 diskutiert. Der Bedarf lässt sich durch andere Parametrierungen sowie weniger Basis-Merkmale verringern, allerdings resultieren hierdurch folgende Einschränkungen, welche eine Reduktion der Güte des Klassifikators zur Folge haben:

- Durch Wahl größerer Skalierungen oder Translationsschritte können bei der Erzeugung der Merkmale wichtige, deskriptive Merkmale ausgelassen werden. Dabei kann z. B. ein mögliches erstes Basis-Haar-Merkmal (Abb. 2.4) der Breite 2 Pixel, welches den Übergang Iris-Sklera durch eine vertikale Kante beschreibt, durch zu große Translationsschritte nicht an der korrekten Stelle durch *Boosting* gewählt werden.

- Da theoretisch eine enorm große Anzahl von Basis-Merkmalen eines Typs möglich sind, und die Auswahl ohne a priori stattfindende Wertung durch *Boosting* geschieht, sollen möglicht viele Basis-Merkmale, die unterschiedliche Strukturen beschreiben, berücksichtigt werden, wie etwa um 45° gedrehte, rechteckige Merkmale (unten rechts in Abb. 2.4).

- Durch Weglassen einzelner Merkmalstypen können Stärken einzelner Typen (Robustheit gegenüber Beleuchtungsunterschieden, Erfassen von Bildtexturen) nicht genutzt werden.

So ergibt sich für einen Merkmalspool von

$$\mathcal{M} = \left\{ \mathcal{M}^{\mathrm{Haar}}, \mathcal{M}^{\mathrm{ED}}, \mathcal{M}^{\mathrm{EOH}}, \mathcal{M}^{\mathrm{NNEOH}}, \mathcal{M}^{\mathrm{HOG}}, \mathcal{M}^{\mathrm{LBP}}, \mathcal{M}^{\mathrm{MBLBP}} \right\} \tag{2.62}$$

für die oben gewählten Parametrierungen und $\mathcal{I} = 4000$ eine Gesamtzahl von 1 468 675 Merkmalen, was einem Speicherbedarf von 23,5 GB für $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ bei einem *Single Precision*-Datentyp entspricht. Es soll hier bemerkt werden, dass Matlab standardmäßig alle Variablen als *Double Precision* instanziiert, was im Arbeitsfluss mit Matlab und MEX-Dateien unbedingt beachtet werden muss.

Um das genannte Problem zu bewältigen, wurde im Verlauf dieser Arbeit *Merkmals-Bagging* eingeführt. Die Idee dabei ist es, das Prinzip des *Bootstrap Aggregating* auf die einzelnen Merkmalstypen anzuwenden und so aus den einzelnen Typen vor der Zusammenführung zu einer Multi-Merkmals-Kaskade bereits die für die trainierte Anwendung geeignetesten Merkmale zu wählen. Das Schema ist in Abb. 2.15 skizziert.

Die dargestellte, vereinfachte Struktur ist so umgesetzt, dass zunächst jeweils eine separate Kaskade mit dem vollen Umfang an Merkmalen eines Typs trainiert wird. Anschließend wird aus der Menge aller schwachen Klassifikatoren dieses einen Typs, welche die Kaskade aufbauen, der neue Pool erstellt, welcher dann in weiteren Schritten mit weiteren Pools zusammengelegt wird, um schließlich $\mathcal{M}^{Bagg}$ zu erhalten. Für eine Beispielkaskade, trainiert mit Haar-, HOG- und LBP-Merkmalen, lässt sich die Merkmalsanzahl von 286 781, 1245 und 90 304 auf 334, 126 und 788 reduzieren, während *HR* und *FAR* nahezu identisch sind [Sch15b]. Nach einmal trainierten Einzel-Merkmals-Kaskaden ist weiterhin das Training mit $\mathcal{M}^{Bagg}$ deutlich schneller, da erstens in $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ deutlich weniger Merkmale sortiert werden müssen und sich zum anderen zeigt, dass weniger Knoten trainiert werden müssen.

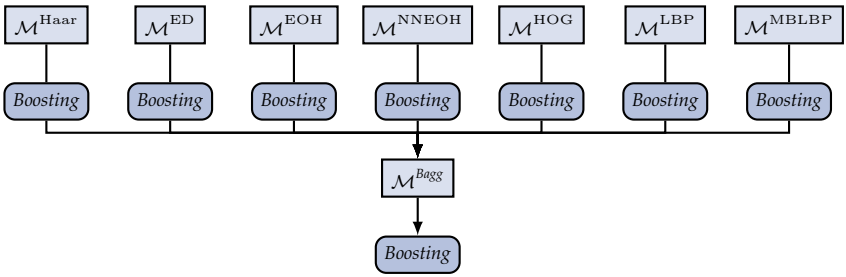


Abbildung 2.15 Übersicht zum *Merkmals-Bagging*. Vor dem finalen Training der Kaskade werden durch *Boosting* die geeignetesten Merkmale jedes Typs gewählt und so ein Training mit $\mathcal{M}^{Bagg}$, welches aus einem maximal großen Merkmalspool erstellt wurde, ermöglicht.

2.6 Detektion

In diesem Abschnitt soll der Detektionsvorgang erläutert werden. Bei der Implementierung kann für das Detektionsprogramm auf viele Teile des Trainings zurückgegriffen werden, sodass in diesen Fällen mit Verweis auf vorige Kapitel nicht näher eingangen wird. Eine Übersicht des Detektionsvorgangs ist in Abb. 2.16 gegeben.

Der hier erforschte Kaskadenklassifkator kann binäre Entscheidungen treffen, d. h. er kann zwischen „ist Element der Klasse 1“und „ist nicht Element der Klasse 1“unterscheiden. Für die Detektionsaufgabe wird hierzu das Problem so umformuliert, dass eine Vielzahl binärer Entscheidungen mit Hilfe des Kaskadenklassifikators getroffen wird. Dies geschieht durch einen *Sliding Window*-Ansatz.

2.6.1 Bestimmen der Detektionsfenster

Beim *Sliding Window*-Ansatz wird das Eingangsbild $g(\mathbf{u})$ in Teilfenster $\varrho_\rho(\mathbf{u})$, mit Index $\rho = \{1, \ldots, |\mathbf{P}|\}$,

$$\varrho_\rho(\mathbf{u}) = [\mathbf{u}_\rho, h, b]^{\mathrm{T}}\,, \tag{2.63}$$

aufgeteilt, wobei $\mathbf{P}$ die Vektoren aller Teilfenster in einer Matrix zusammenfasst und mit $|\mathbf{P}|$ die Anzahl der Positionsvektoren bezeichnet werden soll. Die einzelnen Teilfenster ergeben sich durch Verschiebung

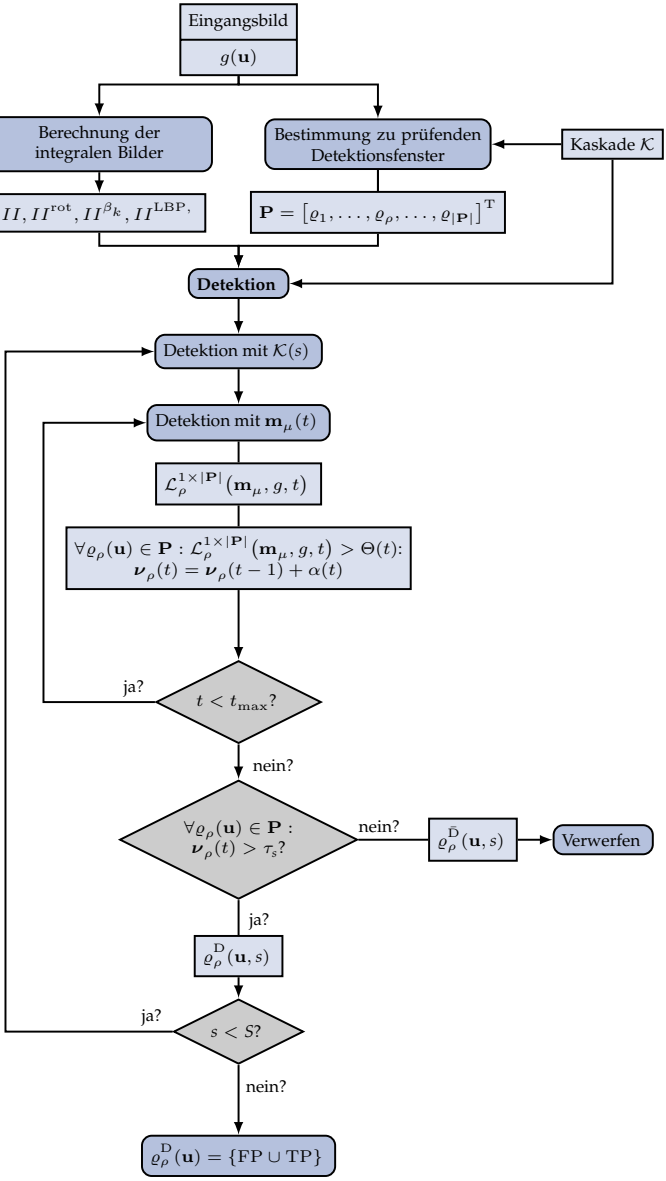


Abbildung 2.16 Ablauf der Detektion für den implementierten Kaskadenklassifikator.

und Skalierung des Trainingsfensters der Größe $h_0 \times b_0$ Pixel, welches gleichzeitig die minimal zu detektierende Objektgröße darstellt. Es gilt:

$$\varrho^{\mathrm{D}}(\mathbf{u}, s=1) \supseteq \varrho^{\mathrm{D}}(\mathbf{u}, s=2) \supseteq \cdots \supseteq \varrho^{\mathrm{D}}(\mathbf{u}) \supseteq \mathbf{P} \tag{2.64}$$

und weiterhin

$$\varrho^{\mathrm{D}}(\mathbf{u}) \cup \varrho^{\bar{\mathrm{D}}}(\mathbf{u}, s=1) \cup \varrho^{\bar{\mathrm{D}}}(\mathbf{u}, s=2) \cup \cdots \cup \varrho^{\bar{\mathrm{D}}}(\mathbf{u}, s=S) = \mathbf{P}(\mathbf{u}) \tag{2.65}$$

sowie

$$\varrho^{\mathrm{D}}(\mathbf{u}) \cap \varrho^{\bar{\mathrm{D}}}(\mathbf{u}, s=1) \cap \varrho^{\bar{\mathrm{D}}}(\mathbf{u}, s=2) \cap \cdots \cap \varrho^{\bar{\mathrm{D}}}(\mathbf{u}, s=S) = \emptyset\,, \tag{2.66}$$

wobei mit ϱ stets eine Menge von Teilfenstern bezeichnet ist und die Superskripte $(\bullet)^{\mathrm{D}}$ und $(\bullet)^{\bar{\mathrm{D}}}$ detektierte bzw. nicht detektierte Teilfenster kennzeichnen.

2.6.2 Berechnen der integralen Bilder

Um die Merkmalsantworten effizient bestimmen zu können, werden für die Berechnung zunächst die integralen Bilder II bestimmt:

$$II(u, v) = \sum_{u' \leq u, v' \leq v} g(u', v')\,. \tag{2.67}$$

Diese erlauben im Falle von Haar-Merkmalen eine Bestimmung des Merkmalswertes durch vier Gleitkommazahlabfragen aus dem II. Für die gradientenbasierten Merkmale können die *Integral Images* analog für die einzelnen Quantisierungen β der Gradientenrichtungen sowie des Betrages berechnet werden. Für die LBP wird das Histogramm mit insgesamt $N+2$ Klassen der Textureinheiten durch $N+2$ integrale Bilder beschrieben.

2.6.3 Detektionsschritt

Zusammen mit der zuvor trainierten Kaskade $\mathcal{K}$, den Teilfenstern $\mathbf{P}$ und den auf dem Eingangsbild $g(\mathbf{u})$ bestimmten II kann die seriellablaufende

Detektion beginnen (vgl. Abb. 2.2). Sukzessive werden hierbei die einzelnen Teilfenster $\varrho_\rho(\mathbf{u})$ für jede Stufe und jedes Merkmal geprüft.

Es wird hierzu die Liste der Antworten aller Teilfenster auf das aktuelle Merkmal, beginnend mit dem ersten schwachen Klassifikator der ersten Stufe der Kaskade, bestimmt, und der Wert für jedes einzelne Fenster mit dem im Training festgelegten Schwellenwert $\Theta(t)$ dieses Merkmals verglichen. Der Vektor $\boldsymbol{\nu}_\rho(t)$ mit der Dimension $1 \times |\mathbf{P}|$ summiert nun die beim Training bestimmten $\alpha(t)$ auf, sofern der Merkmalswert den Schwellenwert im aktuellen Teilfenster überschreitet. Dies wird nun sukzessive für alle schwachen Klassifikatoren der aktuellen Stufe durchgeführt und $\boldsymbol{\nu}_\rho(t)$ laufend aktualisiert. Sind für eine Stufe alle Merkmale ausgewertet, so wird für alle Teilfenster der Summeneintrag in $\nu_\rho(t)$ mit dem Stufenschwellenwert τ_s verglichen und entsprechend eine Klassifikationsentscheidung getroffen. Nach dem Durchlauf jeder Stufe werden die abgewiesenen Fenster aus der Gesamtmenge der Fenster entfernt und nur noch die verbliebene Menge an Fenstern weiterverarbeitet.

Es gilt hier zu beachten, dass durch Berücksichtigung der Güte der Merkmale ein Teilfenster nicht für alle Merkmale den Schwellenwert erfüllen muss, um als Detektion durchzugehen.

2.7 Auswertung

Es soll nun eine Auswertung der erforschten und zuvor diskutierten Methoden erfolgen. Hierzu sollen zunächt kurz die zum Training bzw. zum Testen verwendeten Datensätze besprochen werden, welche teilweise auch in den nachfolgenden Kapiteln verwendet werden.

2.7.1 Trainingsdaten

2.7.1.1 BioID

Die BioID-Datenbank [Bio01] besteht aus 1521 Bildern niedriger Auflösung und wurde von einer an einem Rechner platzierten Webcam unter Alltagsbürobedingungen aufgenommen. Sie zeigt unterschiedliche Individuen unter variierenden Beleuchtungsbedingungen, Posen, Abständen zum Sensor sowie mit und ohne Brillen, Okklusionen der Iris durch

überlappende Augenlider sowie vollständig geschlossene Augen. Der Datensatz der ausgeschnittenen und zentrierten Augen wird mit $\mathcal{I}_{\mathrm{BioID}}$ bezeichnet.

2.7.1.2 YaleB

Die YaleB Test-Datenbank besteht aus Bildern von 10 verschiedenen Individuen, welche unter 405 unterschiedlichen Bedingungen aufgenommen wurden. Dabei wurden insgesamt 9 Kopfposen und jeweils 45 Beleuchtungsbedingungen für die Aufnahmen berücksichtigt, womit sich die Datenbank für Untersuchungen von Methoden bezüglich Beleuchtungsinvarianzen eignet. Da in vorangehenden Tests eine Verbesserung der Performanz durch Verwendung höherdimensionaler Merkmale insbesondere ein besseres Abschneiden der Auswertungen auf Bildern mit Beleuchtungsunterschieden festgestellt wurde, soll die Eignung dieser Datenbank insbesondere zur Dimensionsreduktion von LBP- sowie HOG-Merkmalen untersucht werden. Der Datensatz wird mit $\mathcal{I}_{\mathrm{Yale}}$ bezeichnet.

Während weitere, von der Universitäten Utrecht sowie Essex zusammengestellte, Datensätze (Utrecht, Essex) hier ebenso verwendet werden, soll weiterhin der eigens erstellte IIIT-Datensatz (IIIT) erwähnt werden. Um eine Abhängigkeit der Performanz von der Qualität der Trainingsdaten zu untersuchen, wurde der Datensatz mit einer handelsüblichen Webcam von Personen, die sich vor einem Computer befinden, aufgenommen. Die Probanden variieren dabei die Blickrichtung und tragen Brillen oder Bart. Der zusammengesetze Datensatz der ausgeschnittenen und zentrierten Augen aus Utrecht, IIIT, YaleB und Essex wird mit $\mathcal{I}_{\overline{\mathrm{BioID}}}$ bezeichnet.

2.7.1.3 Größe der Trainingsdaten

Die Größe der gewählten Trainingsdaten begrenzt bei der späteren Detektion die minimale Objektgröße. Bei der Detektion werden im *Sliding Window*-Ansatz zahlreiche Fenster diverser Skalierungen definiert, auf denen die Merkmalsantworten berechnet werden. Während die Schwellenwerte für die Merkmale beim Training auf Basis der Größe der Trainingsbeispiele bestimmt werden, ist es aufgrund der Annahme skalen-

invarianter Merkmale möglich, entsprechend größer skalierte Objekte zu finden. Eine Skalierung kleiner 1 ist auch nach dem Training möglich, hat sich aber als unzweckmäßig ergeben, da hierbei stets ein Informationsverlust entsteht, welcher das Detektionsergebnis negativ beeinflusst. Dies wird in Abschnitt 2.7.7 diskutiert. Aus empirischer Analyse hat sich gezeigt, dass unter realistischen, in der Zielsetzung dieser Arbeit als Szenario beschriebenen, Bedingungen beim Sitzen vor einem Bildschirm (Bildgröße nicht kleiner als 640×480 Pixel) sowie nach Auswerten zahlreicher Datenbanken, sinnvoll eine minimale Augengröße von 36 Pixeln in der Breite angenommen werden kann. Für weitere Informationen zur Wahl der Trainingsdatengröße für überwachtes Lernen sei auf Fasel et al. [FFM05] verwiesen.

2.7.2 Testdaten

Die Auswertung der Methoden erfolgt auf dem Caltech-Datensatz [FFP06]. Der Datensatz beinhaltet Bilder von frontal aufgenommenen Gesichtern bei einer Bildgröße von 892×592 Pixeln. Die Gesichter zeichnen sich inbesondere durch unterschiedliche Skalierungen und Beleuchtungssituationen aus.

2.7.3 Quantitative Ergebnisse

Abbildung 2.17 zeigt eine Übersicht zur Detektionsgüte der unterschiedlichen in dieser Arbeit verwendeten Merkmalstypen. Die Trainings wurden alle unter vergleichbaren Bedingungen bezüglich Trainingsdatensatz, Implementierung und sämtlichen Parametereinstellungen des Kaskadenklassifikators im selben Rahmenwerk durchgeführt, um eine möglichst gute Vergleichbarkeit zu ermöglichen. Die Dimensionsreduktion der mehrdimensionalen Merkmale wurde mittels des Datensatzes $\mathcal{I}_{\text{Yale}}$ durchgeführt. Hierbei ist zu erwähnen, dass bei der Implemtierung der Datentyp beispielsweise für die Merkmalsschwellenwerte eine Rolle spielt, die nicht zu vernachlässigen ist, deren Diskussion allerdings über den Umfang dieser Arbeit hinausgeht.

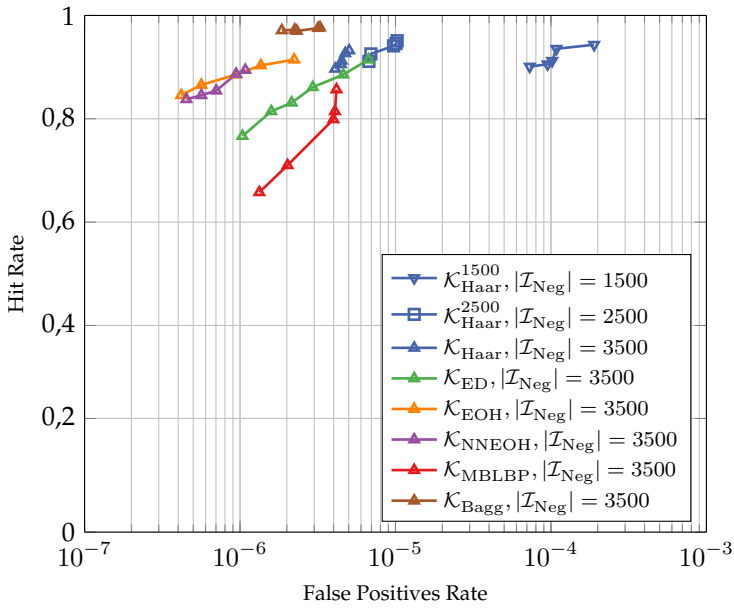


Abbildung 2.17 Detektionsgüte der in dieser Arbeit verwendeten Merkmalstypen.

2.7.3.1 Größe des Trainingsdatensatzes

Es soll zunächst die Auswirkung der Größe des Trainingsdatensatzes diskutiert werden. Für Haar-Merkmale wurden drei Kaskaden trainiert, mit 1500, 2500 und 3500 negativen und 1500 positiven Trainingsbeispielen pro Training einer Stufe. Die Erhöhung der Menge der negativen Trainingsdaten hat ausschließlich positiven Effekt auf die Detektionsgüte der Kaskade. Wie zu erwarten nimmt die *FAR* mit der Hinzunahme von negativen Trainingsbeispielen ab, da beim Training jeder Stufe die schwachen Klassifikatoren unter der Bedingung, deutlich mehr negative Beispiele von den positiven Trainingsbildern trennen zu müssen, gewählt werden. Es ist zu beachten, dass der Pool der negativen Trainingsbeispiele für alle Fälle gleich ist und durch *Bagging* alle negativen Beispiele in allen drei Trainings beitragen. Die Verminderung der *FAR* lässt sich dadurch begründen, dass *gleichzeitig* mehr Daten für die ein-

zelnen Entscheidungen im *Boosting* gesehen werden. Der Verwendung von noch mehr Daten sind dabei zum einen die Grenze der Verfügbarkeit geeigneter Daten zum anderen der Speicherplatz des Rechners beim Sortieren von $\mathcal{L}_{\mu,\iota}^{|\mathcal{M}|\times|\mathcal{I}|}$ gegeben. Da durch *Merkmals-Bagging* die Speicherauslastung stark verringert wird, kann die Anzahl der zum Training verwendeten Daten erhöht werden, was ein weiterer Vorteil des *Merkmals-Baggings* ist. In weiterführenden Forschungen sollte eine dadurch mögliche Steigerung der Detektionsgüte getestet werden.

2.7.4 Einfluss des *Merkmals-Baggings*

Neben den mit einem einzelnen Merkmal trainierten Kaskaden ist noch eine weitere, mittels *Merkmals-Bagging* erstellte Kaskade $\mathcal{K}_{\mathrm{Bagg}}$ ausgewertet worden. Sie wurde trainiert, indem die Merkmale, welche in den zuvor trainierten Kaskaden $\mathcal{K}_{\mathrm{ED}}$, $\mathcal{K}_{\mathrm{EOH}}$ und $\mathcal{K}_{\mathrm{Haar}}$ beim *Boosting* ausgewählt wurden, zusammen mit HOG-Merkmalen (ohne *Merkmals-Bagging*) zu einem neuen Merkmalspool zusammengefasst wurden. Tabelle 2.1 fasst den Prozess des *Merkmals-Baggings* zusammen.

Tabelle 2.1 Reduktion der zum Training verwendeten Anzahl an Merkmalen durch *Merkmals-Bagging*.

Merkmalstyp	ED	EOH	Haar	HOG
Kein *Merkmals-Bagging*	262176	257 548	300510	1245
$\mathcal{K}_{\mathrm{Bagg}}$	1225	430	316	1245

Da zu kleine Merkmalsgrößen für HOG-Merkmale zu zahlreichen unsinnvollen Nulleinträgen im Merkmalsvektor führen und zum anderen für die HOG-Merkmale nur eine Mode implementiert wurde, wurde hier kein *Merkmals-Bagging* angewendet. Die Tabelle zeigt die enorme Reduktion der Merkmalsanzahl und damit auch des Speicherbedarfs und erlaubt das Training eines Multi-Merkmals-Kaskadenklassifikators, ohne Gefahr zu laufen, deskriptive Merkmale durch empirische Vorauswahl (zum Beispiel durch eine grobe Abtastung der Verschiebung der Basis-Merkmale innerhalb des Trainingsfensters) vom Training auszuschließen.

2.7.5 Detektionsgüte der Merkmalstypen

Bei Betrachtung von Abb. 2.17 fallen breite Detektionsunterschiede der Merkmalstypen auf, wobei die *HR* um über $15\,\%$ und die *FAR* um einen Faktor 10 bei gleichen Trainingsdaten variiert. Speziell zeigt sich, dass die gradientenbasierten Merkmale EOH und NNEOH sowie die mit HOG-Merkmalen trainierte Kaskade $\mathcal{K}_{\mathrm{Bagg}}$ besser abschneiden als Haar-Merkmale, MBLBP-Merkmale oder ED-Merkmale. Die Kaskade $\mathcal{K}_{\mathrm{MBLBP}}$ wurde hier mit 88 264 LBP- und 316 Haar-Merkmalen trainiert. Während die MBLBP-Merkmale (*usual*) die niedrigste Detektionsrate aufweisen, zeigt die Auswertung, dass durch das *Boosting* mit dem einfach zu berechnenden Haar-Merkmal eine gute Performanz erreichbar ist.

Die höchste Detektionsrate wird durch die durch *Merkmals-Bagging* unter Berücksichtigung des maximal großen Merkmalspools erstellte Kaskade erreicht. Die *HR* liegt bei einer *FAR* von $1{,}8 \cdot 10^{-6}$ bei $97{,}11\,\%$ auf dem hier ausgewerteten Datensatz und damit bei gleicher *FAR* $7\,\%$ höher als die besten mit einem einzelnen Merkmal trainierten Kaskaden.

2.7.6 Einfluss der Merkmalstypen auf die Zusammensetzung einer Kaskade

Nach Untersuchung der Detetionsgüte der einzelnen Merkmalstypen sowie, durch *Merkmals-Bagging* ermöglicht, einer mit Haar-, ED-, EOH- und HOG-Merkmalen trainierten Kaskade, soll die Zusammensetzung letzterer aus den einzelnen Merkmalstypen untersucht werden. Abbildung 2.18 zeigt den Anteil jedes Merkmaltyps an der Gesamtzahl der schwachen Klassifikatoren pro Stufe für die Kaskade $\mathcal{K}_{\mathrm{Bagg}}$. Während ED-Merkmale nur eine untergeordnete Rolle spielen (ein ED-Merkmal und ein EOH-Merkmal bilden die erste Stufe) zeigt sich, dass auch zu höheren Stufen hin die einfach zu berechnenden Haar- und EOH-Merkmale einen Anteil an der Zusammensetzung der Stufe haben. Während zu höheren Stufen der Anteil des HOG-Merkmals am Aufbau jeder Stufe zunimmt, werden die eindimensionalen Merkmale auch in höheren Stufen noch als schwache Klassifikatoren durch *Boosting* ausgewählt.

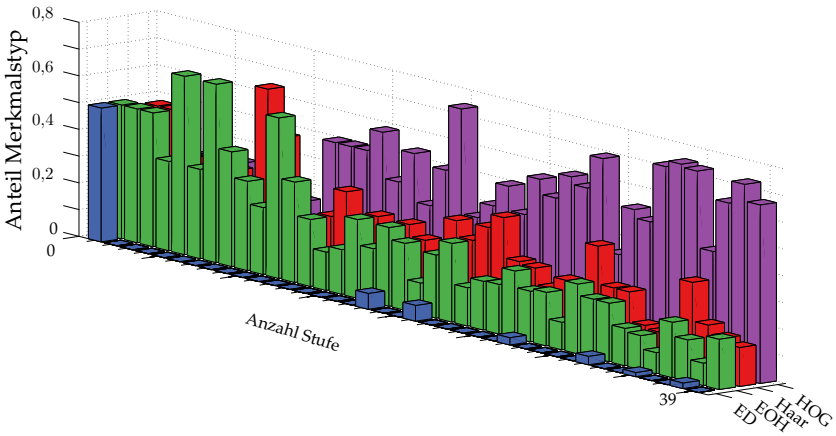


Abbildung 2.18 Anteil einzelner Merkmalstypen der Multi-Merkmals-Kaskade $\mathcal{K}_{\mathrm{Bagg}}$.

2.7.7 Skalenvarianz der Merkmalstypen

In einem weiteren Experiment wird die Skalenvarianz der Haar- und LBP-Merkmale untersucht. Hierzu wurden die ursprünglich 896×582 Pixel großen Bilder skaliert. Das Ergebnis ist in Abb. 2.19 zu sehen.

Der Graph zeigt die Detektionsergebnisse der Kaskaden $\mathcal{K}_{\mathrm{MBLBP}}$ und $\mathcal{K}_{\mathrm{Haar}}$, jeweils trainiert mit 3500 negativen Trainingsbeispielen für unterschiedlich skalierte Auswertebilder, ausgehend von einer Skalierung $f_{\mathrm{S}} = 1$ und einer Größe von 892×592 Pixeln, wobei Augen in der Größenordnung von 80×80 Pixel bis 40×40 Pixel vorkommen. Auf den ersten Blick scheint die Skalierung der Eingangsbilder die Güte der Klassifikation der Haar-Merkmale weniger zu beeinflussen als die MBLBP-Merkmale.

Beide Merkmale zeigen ein starkes Abfallen der *HR* ab einer Skalierung auf etwa $0{,}45$ der Originalgröße. Dies lässt sich damit erklären, dass dies der minimalen Größe von im Datensatz vorkommenden Augen entspricht. Ein Unterschied zeigt sich, dass bei Hochskalierung der Bilder um den Faktor 2 mit den Haar-Merkmalen eine deutlich höhrer Detektionsrate erreichbar ist, während eine Hochskalierung für die texturbasierten Merkmale keine Verbesserung bringt. Weiterhin lässt sich zusammenfassen, dass durch das MBLBP-Merkmal eine ähnlich gute Skaleninvarianz wie mit Haar-Merkmalen bewerkstelligen lässt.

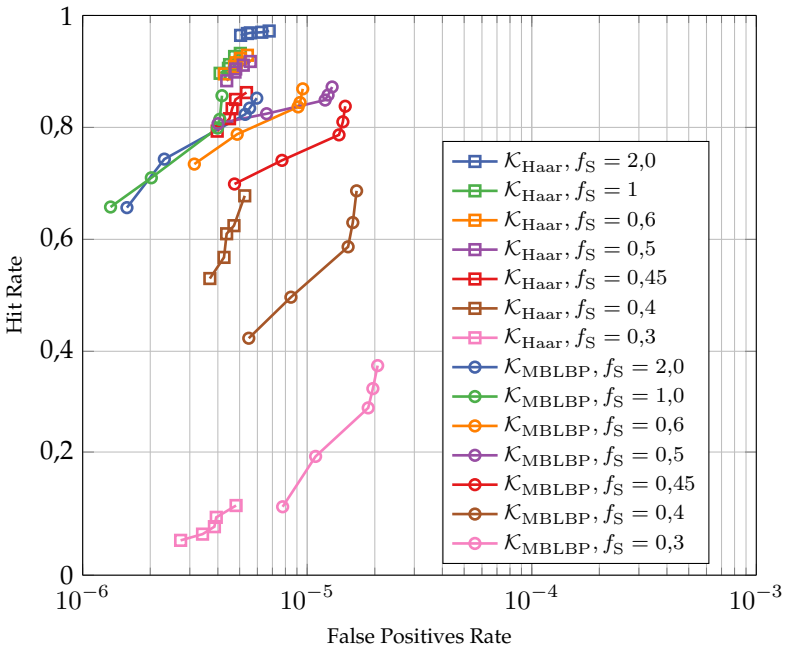


Abbildung 2.19 Einfluss einer Skalierung der zu untersuchenden Eingangsbilder.

2.8 Zusammenfassung

In diesem Kapitel wurde ein Rahmenwerk zum Trainieren eines Kaskadenklassifikators sowie zur Anwendung des Klassifikators für Detektionsaufgaben vorgestellt. Es wurde ein vollständiges System in Matlab/C++ implementiert, wobei folgende Beiträge zur Erweiterung des Standes der Technik beitragen sollen:

- Die Implementierung der Berechnung verschiedener Merkmalstypen sowie *Boosting*-Methoden innerhalb desselben Rahmenwerkes, wodurch die Erzielung vergleichbarer Ergebnisse ermöglicht wird, welche zur Evaluation einzelner Komponenten von Kaskadenklassifikatoren unter gleichen Randbedingungen herangezogen werden können.

- Gleichzeitiges Trainieren von Grauwert-, Gradienten- sowie Texturbasierten Merkmalen innerhalb einer Kaskade auf demselben Datensatz durch Lösen des Speicherplatzproblemes mittels des neu vorgestellten *Merkmal-Baggings*.
- Eine Möglichkeit zur Gewichtung dominanter Merkmalsdimensionen bei gleichzeitig automatischer Auswahl deskriptiver Merkmale mittels *Merkmals-Boosting*, welches empirisches Anpassen von Parametrierungen an einen Datensatz oder Problem obsolet macht sowie die Möglichkeit zur Dimensionsreduktion hochdimensionaler Merkmale durch Vernachlässigen weniger deskriptiver Merkmalsdimensionen.

Die Auswertungen der Detektionsgüte der verschiedenen Merkmalstypen sowie eines Multi-Merkmal-Multi-Dimensionen Kaskadenklassifikators in einem Rahmenwerk unter vergleichbaren Randbedingungen hat den positiven Einfluss der Verwendung verschiedener Merkmalstypen auf die *Hit Rate* gezeigt. Die durch die Reduktion des Speicherplatzes durch *Merkmals-Bagging* ermöglichte Auswertung der Multi-Merkmal-Kaskade hat gezeigt, dass die eindimensionalen Merkmale auch in höheren Stufen zum Aufbau des starken Klassifikators beitragen.

Da eine umfangreiche Implementierung durchgeführt wurde, bei der alle Programmteile, sofern nicht anders angegeben, innerhalb der vorgelegten Arbeit erstellt wurden, sollen die implementierten Module zusammengefasst werden:

- Berechnung der Merkmalswerte der oben aufgeführten Typen (eigene Implementierungen für HOG- und LBP-Merkmale) sowie jeweils eine dem *Sliding Window*-Ansatz folgende Detektion für eingehende Daten sowohl im Trainingsvorgang im Rahmen des *Baggings* als auch im eigentlichen Detektionsprozess.
- Implementierung verschiedener *Boosting*-Algorithmen in Form von *Adaboost*, *Real-Adaboost* sowie *Gentle-Adaboost*.
- Module zur Reduktion der Merkmalsdimension mittels *Merkmals-Boosting*, mittels eines *Brute Force*-Ansatzes sowie mittels LDA.

- Implementierung des Hauptprogramms inklusive einer Oberfläche zum Training (siehe Abb. 2.20) in Matlab, sowie Implementierung rechenintensiver Subroutinen zum Berechnen von Merkmalen sowie Detektionen im *Sliding Window*-Ansatz in C++ bzw. mittels MEX-Dateien.

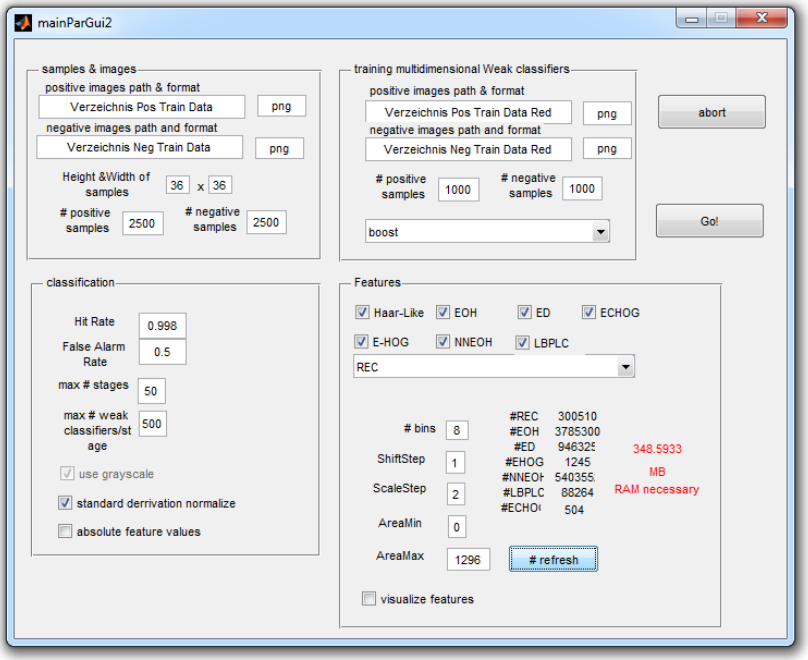


Abbildung 2.20 Die grafische Oberfläche zum Training einer Kaskade für das implementierte Rahmenwerk.

3 Präzise Irislokalisation

In diesem Kapitel werden Forschungsergebnisse zur präzisen Lokalisierung des Irismittelpunktes mittels eines kombinierten Ansatzes aus Isophoten und Kaskadenklassifikator vorgestellt. Die Methode nutzt dabei das Rahmenwerk des in Kap. 2 beschriebenen Multi-Merkmals-Kaskadenklassifikatoransatzes mit Anwendung von *Merkmals-Boosting* sowie *Merkmals-Bagging*, um zuvor mittels eines Ansatzes, der die Linien konstanter Intensität entlang des Pupillen- sowie Irisumfanges zusammen mit Symmetrie- und Grauwerteigenschaften des Auges ausnutzt, gefundene Kandidaten für das Iriszentrum zu klassifizieren. In dieser Arbeit wird sich insbesondere auf die Herausforderung, dies unter realen Bedingungen, bei denen Verdeckungen des Auges durch (halb-)geschlossene Lider, das Tragen von Brillen, welches Reflexionen hervorrufen kann, sowie auftretende Beleuchtungsunterschiede, konzentriert. Dies soll unter Verwendung niedrigaufgelöster Bilder geschehen, da solche durch handelsübliche Hardware, welche in täglich verwendeten Geräten nahezu überall verfügbar ist, eine maximal breite Anwendung der Methoden ermöglichen, während die Methode mit einhergehender steigender Genauigkeit auch auf hochaufgelöste Daten anwendbar sein soll. Die Zielsetzung bezüglich der Genauigkeit soll der Voraussetzung für eine erscheinungsbasierte Blickrichtungsschätzung genügen, welche dem Wort *präzise* hier eine Genauigkeit im Bereich weniger Pixel zuordnet.

3.1 Problemstellung

Die Irislokalisation im Vergleich zur Augendetektion zielt auf das präzise Auffinden des Mittelpunktes der Pupille in einem lokalen Bereich des Gesichtes, welcher als Eingang nach einem Vorverarbeitungsschritt dient, ab. Im Gegensatz zur Augendetektion, bei dem beispielsweise,

wie oben beschrieben, mit einem *Sliding Window*-Ansatz das komplette Eingangsbild nach möglichen Objekten der Klasse Auge durchsucht wird, soll hier ein solches gefundenes Objekt bereits als gegeben gelten und die exakte Position der Iris innerhalb dieses Suchfensters bestimmt werden. Obwohl Iristracking und -lokalisation in den letzten Dekaden ein aktives Forschungsgebiet darstellen, verdeutlichen die zahlreichen Publikationen auf diesem Gebiet, dass es sich weiterhin um ein offenes Problem, insbesondere unter den hier vorausgesetzten Randbedingungen, wie ungezwungene Konditionen bezüglich der technologischen Nutzung wie etwa bei *Head Mounted Devices*, Verdeckungen sowie Variationen von Kopfposition und Beleuchtung, handelt [Böh+06; Son+13]. Als essentieller Vorverarbeitungsschritt in vielen technischen sowie konsumorientierten Mensch-Maschine-Anwendungen, wie die Blickrichtungsschätzung oder die Gesichtserkennung, sind robuste, präzise und recheneffiziente Methoden notwendig. Daher soll hier der Ansatz einer akkuraten Irislokalisation verfolgt werden, bei dem Bilder, die aus einfacher, handelsüblicher Hardware, wie beispielsweise Webcams, gewonnen werden, verarbeitet werden. Das Ziel ist hierbei, in Hinblick auf eine erscheinungsbasierte Blickrichtungsschätzung, eine Methode mit dem Fokus einer Lokalisierung mit Fehlern im Pixelbereich zu erforschen, da diese Genauigkeit kritisch für eine Blickrichtungs- [VSG12] oder auch Gesichtserkennung ist [KHM08].

3.2 Stand der Wissenschaft

Das Finden der Pupille als Charakteristikum des Auges spielt eine wesentliche Rolle in der Bildverarbeitung sowie in dem Computersehen und kann genutzt werden, um eine Gesichtsdetektion oder -erkennung zu unterstützen und ist zwingend erforderlich für eine monokulare Blickrichtungsschätzung [KHM08; Son+13; VSG12]. Insbesondere für Letzteres ist eine präzise Lokalisierung der Iris notwendig, wenn Blickrichtungschätzung erscheinungsbasiert durchgeführt wird [MCP12; SSS14].

Es existiert eine Fülle von wissenschaftlichen Arbeiten, welche sich mit der Thematik Irislokalisation beschäftigen und dabei auf unterschiedliche Hardware-Randbedingungen, die von monokularen Bedingungen abweichen, zurückgreifen. Ansätze, die ein *Head-Mounted Device*

(HMD) verwenden, werden in [Fuh+16] gegenübergestellt, während Überblicke für sowohl elektrodenbasierte als auch auf Computersehen basierte Ansätze, welche Infrarotlicht und *Pan and Tilt*-Kameras [SS11] oder mehrere Lichtquellen und mehrere (Nicht-Stereo-)Kameras verwenden, in [Bat+05; HQ10; AF13] diskutiert werden. Bei letzteren lassen sich Methoden, die auf aktivem Licht (IR), 3D Daten (Stereo) sowie Kombinationen hieraus bestehen, sowie erscheinungsbasierte Methoden unterscheiden. Unter erscheinungsbasierten Methoden werden hier solche verstanden, welche als Eingangsdaten lediglich ein zweidimensionales ein- oder dreikanaliges Bild verarbeiten. Eine Einordnung dieser Arbeit in die beschriebende Klassifikation der Hardware der Ansätze zeigt Abb. 3.1.

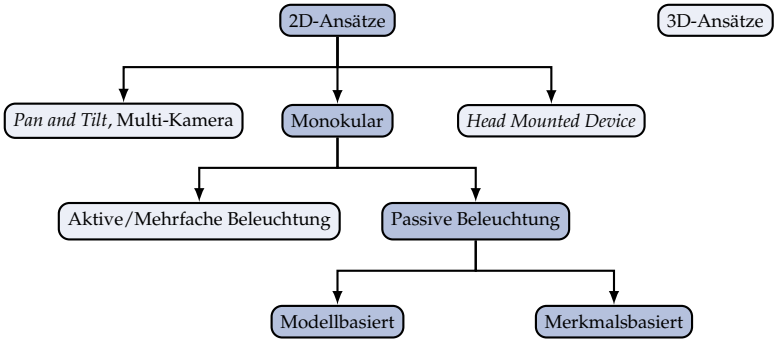


Abbildung 3.1 Einordnung der Anforderungen an die Hardware.

Hierbei sind unter 3D-Ansätze solche, welche dreidimensionale Rohdaten des Gesichtes, etwa durch Stereo, Laser-Triangulation oder mittels *Time-of-Flight*, verarbeiten, zusammengefasst und sollen hier nicht weiter betrachtet werden. Der hier gewählte erscheinungsbasierte Ansatz hat den Vorteil, auf über einen einfachen Sensor hinausgehende und häufig invasive Hardware, wie beispielsweise *Head-Mounted Displays*, bei denen oft die natürliche Sicht blockiert wird (vgl. hierzu neuere Ansätze: metavision.com [met]), zu verzichten. Auch Nachteile, die sich aus einer Verwendung von IR-Licht bei Tageslichtanwendungen ergeben, können ausgeschlossen werden. Aus diesem Grund soll sich nun auf

eine umfangreiche Schilderung des aktuellen Standes der Literatur für erscheinungsbasierte Methoden zur Lokalisation konzentriert werden.

3.2.1 Modellbasierte Verfahren

Modellbasierte Verfahren nutzen häufig den gesamten lokalen Bildausschnitt des Auges oder des Gesichts, um auf diesem komplexe Merkmale zu berechnen oder sie direkt weiterzuverarbeiten. Die Merkmalsantworten werden dann an Methoden des maschinellen Lernens übergeben und eine Lokalisierung mittels eines gelernten Modells durchgeführt. Die Verfahren zeichnen sich durch einen in der Regel erhöhten Rechnenbedarf und weiterhin dadurch, dass sie Trainingsdaten benötigen, aus, was stets eine Abhängigkeit der Generalisierungsfähigkeit der Methode vom Trainingsdatensatz bedingt.

Den Mittelwert, die Standardabweichung sowie die Entropie als Merkmale nutzend rastern die Autoren in [HW00] die Gesichtslandschaft und verwenden endliche Automaten und einen evolutionären Algorithmus, um eine Salienzenkarte für mögliche Augenregionen zu erstellen. In einer zweiten Stufen wenden sie den evolutionären Algorithmus und Entscheidungsbäume sowie 147 ausgesuchte Merkmale an, um in den zuvor bestimmten Regionen die Augen zu lokalisieren. Statistische Lernmethoden werden von Jesorsky et al. [JKF01] angewandt, um basierend auf Binärbildern ein Gesichts- und Augenmodell zu erstellen. Mittels der Hausdorff-Distanz werden die Augen anschließend durch einen zweistufigen Algorithmus lokalisiert. Dabei wird eine erste grobe Lokalisierung mittels eines zweiten, detaillierterem *Multilayer Perceptrons* (MLP), welches mit Iris-zentrierten Bildern trainiert wird, präzisiert. Cristinacce et al. [CCS04] nutzen Kaskadenklassifikatoren, um separate Klassifikatoren für 17 Gesichtsmerkmale zu trainieren. Sie gestalten die Suche mit Hilfe der Kaskaden robust, indem sie geometrische Randbedingungen für die Anordnung der einzelnen Merkmale festlegen und nennen das Vorgehen *Pairwise Reinforcement of Feature Responses*. Unter Verwendung von Histogrammen, welche mögliche Positionen von Charakteristika repräsentieren, werden die Prädiktionen der Klassifikatoren der einzelnen lokalen Gesichtsmerkmale zu einem Multi-Detektor fusioniert. Sie verfeinern anschließend ihr Ergebnis, indem sie ein *Active Appearance Model* (AAM) anwenden. Lineare und nicht-lineare *Support Vector Machines*

werden in einem überwachten Lernprozess für *Face Alignment* und Augenlokalisation in [Ham+05] verwendet. Die Autoren verwenden Gabor-Filter, um 10 charakteristische Gesichtsmerkmale zu detektieren und 2D-affin transformierte Gesichtsbilder in einem Referenzkoordinatensystem zu erhalten, um so Korrespondenzen zwischen Gesichtsmodell und Charakteristika herzustellen. Zhiheng et al. [Zhi+06] erweitern den konventionellen *Boostrap-Aggregating*-Ansatz (auch *Bagging*) auf einen zweidimensionalen Ansatz, indem sie die Methode nicht nur auf die negativen, sondern auch auf die positiven Trainingsbeispiele anwenden. Indem sie einige niedrige Detektionsraten für jede Stufe des Kaskadenklassifikators wählen, ist die 2D-Kaskade fähig, eine signifikant höhere Anzahl positiver Trainingsbeispiele zu handhaben. Dadurch liefert die Methode einen robusten Klassifikator, in dem eine sehr breite Intraklassenvarianz innerhalb des Trainingsdatensatzes abgedeckt werden kann. Campadelli et al. [CLL06] extrahieren Haar-Wavelet-Koeffizienten von Trainingsbildern und unterscheiden normalisierte Koeffizienten in zwei Klassen, welche systematische Einheitlichkeit und systematische Variation beschreiben. Sie definieren eine Zielfunktion, innerhalb der sie den Fehler zwischen einem mittleren Auge und einem Augenbild, welches sie aus einer Teilmenge der extrahierten Koeffizienten bilden, berechnen. Nach Finden der besten Teilmenge an Koeffizienten werden mit ihnen zwei SVM trainiert, mit Hilfe derer dann die Augenlokationen präzisiert werden. Zusätzlich zu einem SVM-basierten Augenlokalisierer präsentieren die Autoren in [CLL09] einen Mund-Detektor. Indem sie die geometrische Relation zwischen Augen und Mund modellieren, korrigieren sie die jeweiligen Lokalisationen. Sie verwenden die schnelle Wavelet-Transformation zur Merkmalsextraktion. Basierend auf einer Fisher-Diskriminanzanalye (FDA) nutzen die Autoren in [WJ07; Wan+05] direkt die Grauwertintensitäten als globale Merkmale, die sie aus Trainingsbildern lernen. Sie normalisieren die Bildgröße und verwenden statistische Lernmethoden, um Merkmale zu lernen. Um die Annahme einer Gaußverteilung für eine optimale Merkmalsfindung bei der FDA umgehen zu können, präsentieren die Autoren basierend auf der nicht-parametrischen Diskriminanzanalyse (NDA) eine rekursive Erweiterung (RNDA), welche globale Merkmale aus den kompletten Trainingsbeispielen lernt, statt Kombinationen lokaler Merkmale zu bilden. Mit Hilfe der so bestimmten Merkmale und *Adaboost* wird ein Klassifika-

tor für die Augenlokalisierung trainiert. Gabor-Filter sowie empirisch entworfene Filter zur Detektion der Augenwinkel werden in [SR01] kombiniert zur Augenlokalisation eingesetzt. Kim et al. [Kim+07] nutzen Multi-Skalen-Gabor-Merkmale und einen *Coarse-to-Fine*-Ansatz zur Irislokalisation, wodurch eine Verbesserung der Robustheit der Methode bezüglich der Initialisierung berichtet wird.

Multi-Skalen-LBP werden von Kroon et al. [Kro+09] verwendet, um lokale Histogramme von Merkmalsantworten zu bestimmen. Es werden a priori Augenpositionen aus Trainingsdaten gewonnen, indem, basierend auf der *Ground Truth*, mit Hilfe der *Bounding Boxes* eines Gesichtsdetektors Augenabstände und Positionen gelernt werden. Die so gewonnenen Abstände werden dann im Training auf zuvor definierte Augenabstände skaliert, für welche dann Merkmalsantworten bestimmt werden. Beim Testen werden die gelernten Daten dann genutzt, um aus den Eingangsbildern die Gesichter in ähnliche Positionen wie beim Training für die verschiedenen Skalierungen zu bringen. Der Suchbereich für die Testphase wird durch den Abstand der Augen definiert. Beim Training werden dann LBP-Merkmale an allen Positionen auf Größe der Suchfenster für die verschiedenen Skalen extrahiert und Histogramme gebildet, die die Häufigkeit des Auftretens einzelner Texturen in der jeweiligen Region repräsentieren. Es wird dann als Merkmal der Quotient zwischen Histogrammen von Regionen, in denen sich ein Auge befindet und solchen, in denen sich kein Auge befindet, bestimmt. Die Antworten der einzelnen Skalen werden dann superponiert und anschließend nach der Position mit dem größten Merkmalswert gesucht. Die Methode zeichnet sich durch eine hohe Parametrierung bezüglich der Anzahl der Skalen, der definierten Augenabstände und LBP-Fenstergrößen aus. Des Weiteren wird in der Arbeit der Einfluss der Irislokalisation auf die Güte der Gesichtserkennung untersucht.

Yang [Yan+11] extrahieren Bildausschnitte auf verschiedenen Skalen und an verschiedenen Bildpositionen in ausgerichteten Trainingsbildern um die Augenpositionen. Die hochdimensionalen Vektoren der Grauwertbildausschnitte werden dann durch *k-means* und Singulärwertzerlegung (K-SVD) in ihrer Dimension reduziert. Zur Lokalisierung wird, beginnend auf der größten Skale, sukzessive der *Orthogonal Matching Pursuit Algorithm* [Tro04] angewandt. Ergebnisse werden anhand des BioID-Datensatzes und zweifacher Kreuzvalidierung präsentiert.

Dong et al. [DZL11] formulieren den konventionellen Kaskadenklassifikatoransatz in einem probabilistischen Ansatz. Es wird jedem positivem Trainingsbild eine Wahrscheinlichkeit zugeordnet, auch für solche, die während des Trainings ab einer Stufe entsprechend als *FN* zurückgewiesen werden. Daraus ergibt sich für das behandelte Trainingsbild eine Wahrscheinlichkeit, zur positiven Klasse zu gehören in Abhängigkeit der Kaskadenstufe der Zurückweisung. Durch Auswertung aller in Frage kommenden Pixel im Trainingsbild kann so jedem dieser Pixel eine Wahrscheinlichkeit, zur Klasse *Auge* zu gehören, zugewiesen werden. Neben der Verwendung von LBP stellen sie weiterhin *extended*-LBP vor, welche die Geometrie von LBP auf Ellipsen erweitern. Sie verwenden einen *Coarse-to-Fine*-Ansatz, indem sie zwei Klassifikatoren hintereinanderschalten, wobei der erste robust mit Trainingsdaten, die ein großes Gebiet um die Augen herum abdecken, der zweite präzise mit Trainingsdaten, die sehr fein die Position des Auges wiedergeben, trainiert werden.

Die Autoren in [NL12] stellen ein kombiniertes Gesichts- und Augenlokalisierungsverfahren vor, in welchem sie SVM trainieren. Das Trainingsbild wird hierzu in vier Quadranten aufgeteilt, für welche zwei verschiedene Merkmale für Augen- und Nicht-Augenregionen trainiert werden. Zum einen werden *Local Ternary Pattern* (LTP) [TT10], die im Vergleich zu konventionellen LBP weniger empfindlich gegenüber Rauschen sind, da sie, anstatt eine auf einem festen Grauwert basierende Binärschwelle anzuwenden, ein Toleranzintervall einer festgelegten Breite um den Grauwert des zentralen Pixels zulassen und so einen tertiären Code erstellen. Zusätzlich werden *Local Phase Quantization*-Merkmale (LPQ) [OH08] eingesetzt. Die LPQ-Merkmale werden aus vier komplexen Koeffizienten aus einer 2D-Kurzzeit-Fourier-Transformation gewonnen, welche dann einer statistischen Analyse unterworfen weden, um schließlich quantisiert und auf einen Wertebereich $[0, \ldots, 255]$ abgebildet zu werden. Sie stellen robuste Merkmale dar, wobei sie insbesondere Invarianzen gegenüber Unschärfe im Bild aufzeigen.

Die Hough-Transformation wird zusammen mit einem *Least-Squares* (LS)-Ansatz in [Yan+14] eingesetzt, während in [CCD12] mehrere Schätzungen der Hough-Transformation, basierend auf unterschiedlichen Teilmengen der zuvor bestimmten Bildkanten, akkumuliert werden.

In [RG14] nutzen die Autoren Merkmale, die aus den Koeffizienten der zweidimensionalen diskreten Cosinus-Transformation gewonnen werden, um damit das erste von zwei kaskadierten *Multilayer-Perzeptren* (MLP) für eine nichtlineare Regression zu trainieren. Das zweite Perzeptron wird mit kleinen Bildausschnitten trainiert, welche die Prädiktionen der ersten Stufe bezüglich der Genauigkeit korrigieren sollen. Von Markuš et al. [Mar+14] werden grauwertige Augenbilder in normalisierten Koordinaten direkt als Merkmale verwendet, für die mit Hilfe von *Randomized Trees* Schwellenwerte gefunden werden, um eine Lokalisierung durchzuführen, wobei sie einer *Coarse-to-Fine*-Strategie nachgehen.

3.2.2 Merkmalsbasierte Verfahren

Merkmalsbasierte Ansätze nutzen keine Methoden, die auf maschinellem Lernen basieren und hängen somit auch nicht von Trainingsdaten und der dort abgebildeten Intraklassenvarianz ab. Es existieren zahlreiche merkmalsbasierte Ansätze zur Lösung des Irislokalisationsproblem, welche häufig Charakteristika wie die Symmetrie der Iris, Grauwert-Histogramme und Gradienten [KM96] oder morphologische Operationen und die Dunkelheit der Pupille [AS10] ausnutzen.

In [TPC07] und [ZYK13] werden Histogramme als vertikale und horizontale Projektionen der Grauwerte des Gesichtes bzw. des Bereiches der Augen erstellt, um hieraus auf die Position der Augen zu schließen. Indem Bildregionen von Interesse durch Grauwerthistogramme charakterisiert werden, bestimmen Asadifard und Shanbezadeh [AS10] eine kumulative Verteilungsfunktion der Grauwerte entlang der Augenregion und filtern dunkle Pixel durch eine Schwellenwertoperation. Nach Anwenden von morphologischen Filtern suchen sie nach Pixeln minimaler Intensität im Originalbild innerhalb der erhaltenen Suchregion. Reale et al. [Rea+11] modellieren den 3D-Augapfel und bilden den 2D-Bildausschnitt des Auges auf das Augenmodell ab. Anschließend rotieren sie das Augapfelmodell, sodass die Iris direkt in Richtung der monokularen Kamera zeigt. Mittels Kreis-*Fitting* wird der Rotationszustand zwischen gerendertem Bild und Modell gesucht. Sigut und Sidha [SS11] arbeiten im YCbCr-Farbraum und nutzen eine separate Lichtquelle, um das Eingangsbild mittels eines Schwellenwertes zu filtern und um die Reflexion der Lichtquelle an der Hornhaut des Auges zu nutzen (*Cor-*

neal Reflection). Der Irismittelpunkt wird dann durch Canny-Filterung und Anwendung von RANSAC sowie Ellipsen-*Fitting* bestimmt.
Eine radiale Symmetriegüte wird in [BSW06] ausgewertet und die Iris mithilfe eines Schwellenwertes bestimmt. Sie nutzen Schwellenwerte und Heuristiken, um unter lokalen Maxima für das Iriszentrum zu entscheiden. Timm und Barth [TB11] bestimmen das Vektorfeld der Gradienten in der Umgebung der Iris und bestimmen das Iriszentrum unter der Annahme, dass der Verschiebungsvektor zwischen Gradientenpixel und möglichem Zentrum der Iris kollinear zueinander sind. Lichtenauer et al. [LHR05] setzen Isophoten als Linien konstanter Intensität innerhalb eines Grauwertbildes als Merkmale zur Objektdetektion ein, indem der Radius eines jeden zu einer Isophotenkurve gehörenden Pixels berechnet wird. Durch Behandeln der Iris als ein konzentrisches Kreisobjekt und Berechnung von Verschiebungsvektoren basierend auf dem Isophotenradius werden Votings für das Iriszentrum, sogenannte Isocenter, von Valenti und Gevers [VG08] gewonnen. Die Autoren nutzen die Rundheit als Gewichtung für die Isophoten mit der Motivation, dass diese im Vergleich zu ihrer Nachbarschaft als gekrümmte Bögen mit hohen Rundheitswerten verstanden werden können. Eine weiterführende Klassifikation zuvor gefundener Isocenter geschieht durch Valenti und Gevers [VG12] in einem hybriden Ansatz, indem SIFT-basierte Deskriptoren trainiert werden und Kreuzvalidierung angewandt wird, um zwischen lokalen Maxima der Isocenter zu unterscheiden. Baek et al. [Bae+13] verwenden ein elliptisches Formmodell, welches sie entsprechend der Rotation des Augapfels variieren, um so durch *Template Matching* mit einer Datenbank auf die Position des Iriszentrums zu schließen. Ein anderer *Matching*-Ansatz wird in [WYM07] verfolgt. Sie behandeln grauwertige Augenbilder wie 3D-Topographieoberflächen, bei denen die Höhe jedes Ortes durch die Pixelintensität beschrieben wird. Indem sie Terrain-Merkmale berechnen, führen sie ein *Matching* jedes Pixels mit einer von 12 Terrain*templates* durch und labeln auf diese Weise die Oberfläche. Die Inferenz für die Augenlokalisation wird dann anhand eines gelernten probabilistischen Modells basierend auf SVM und der Bhattacharyya-Affinität durchgeführt. Von Pang et al. [Pan+15] werden die Methoden aus [VG12] übernommen und die Kandidaten für das Iriszentrum auf verschiedenen Skalen weiter durch Regression, basierend auf einer Angleichung von Gesichtsmerkmalen, inferiert.

Einen tabellarischen Überblick des Standes der Technik bietet Tabelle 3.1. Die Angabe der jeweils verwendeten Datenbanken (BioID [Bio01], XM2VTS [Mes+99], gi4e [Ari+16], UULM [Wei+07], YaleB [GBK01], FERET [Jon+00], BANCA [Bai+03], FRGC [Jon+05], IMM [IMM], JAFFE [JAF], CASPERL [Gao+08], AR [Mar98], PF01 [BKC01], LFPW [Bel+13], EX [Ess], Utrecht [Unib]) soll zur Evaluation der Ergebnisse berücksichtigt werden, um etwa Abweichungen der Disjunktheit zwischen Trainings- und Testdaten oder Einflüsse der Ergebnisse durch Kreuzvalidierung zu beachten. Eigene verwendete Datensätze der Autoren, wie etwa bei Cristinacce, wurden in der Tabelle nicht vermerkt.

3.3 Lösungsansatz und eigener Beitrag

Der in dieser Arbeit erforschte Ansatz zur präzisen Irislokalisation nutzt eine Isophotenrepräsentation für jedes Pixel, welcher durch Gewichtung sowohl der Start- als auch Endpunkte von Verschiebungsvektoren das Finden von Kandidaten für Iriszentren erlaubt. Durch Kombination mit erscheinungsbasierter Information aus einem quasi-kontinuierlichen Kaskadenklassifikator wird eine anschließende robuste und präzise Lokalisierung des Irismittelpunktes ermöglicht.
Einen Überblick über die Einordnung der erforschten und hier vorgestellten Methode zeigt Abb. 3.2, wobei der dunkel(blau) hervorgehobene Ellipsenbereich in der Mitte der Abbildung die Fusion der hier eingesetzten Informationen illustriert.

Basierend auf einem Isophotenansatz sollen die Informationen, die sich aus der punktsymmetrischen Eigenschaft der Iris und der Pupille ergeben, mit den niedrigen Grauwerten innerhalb der Pupille kombiniert werden. Dabei wird ausgenutzt, dass sich entlang der Übergänge Sclera (Lederhaut) – Iris sowie Iris – Pupille Linien konstanter Intensität auf einem unter frontaler Ansicht konzentrischen Kreis befinden, womit sich wiederum konzentrische Isophoten finden lassen können. Aufgrund der Symmetrie zeigen die vorzeichenunbehafteten Richtungen der Gradienten senkrecht zu dieser Isophoten alle in radialer Richtung ausgehend von der Pupille. Durch Gewichtung und Akkumulation von aus den Gradienten sowie der Isophotenkrümmung gewonnenen Verschiebungsvektoren werden unter Berücksichtigung der Rundheit der Isophoten

sowie der Dunkelheit der Pupille Kandidaten für das Pupillenzentrum bestimmt.

Um die Methode ohne Einschränkungen der Anforderungen an die verwendete Hardware zur Datenaufnahme breit anwendbar zu machen, muss sie eine hohe Robustheit aufweisen und auch mit niedrig aufgelösten Bildern entsprechend den Anforderungen funktionieren. Aus diesem Grund fallen Ansätze, die etwa auf der Hough-Transformation oder *Template Matching* beruhen, raus. Darüber ist bei auf Training basierenden Ansätzen stets das Problem der Generalisierungsfähigkeit gegeben.

Tabelle 3.1 Stand der Technik zur erscheinungsbasierten Irislokalisation. Die Superskripte geben weitere Informationen zur Auswertung an:
$(\,)^{x}$: Kreuzvalidierung;
$(\,)^{\bar{g}}$: Keine strikte Trennung von Trainings- und Testdaten, anfällig für Überanpassung;
$(\,)^{t}$: Satz aufgetrennt in Trainings- und Testdaten, siehe Tabelle 3.2.

Autor	Ansatz	Methoden	Trainingssatz
Jesorsky [JKF01]	Modell	Hausdorff, MLP	BioID, XM2VTS
Cristinacce [CCS04]	Modell	Multi-Kask., alignm.	
Hamouz [Ham+05]	Modell	Gabor, alignm.	BANCA
Niu [Zhi+06]	Modell	2D-Kaskade	FERET, BANCA, FRGC
Campadelli [CLL09]	Modell	Haar-Wavelets, SVM	FERET^{t}, BANCA^{t}
Kim [Kim+07]	Modell	Multi-Skalen Gabor	BioID^{x}, FERET^{x}, JAFFE^{x}, IMM^{x}
Yang [Yan+11]	Modell	Grauwerte, k-SVD	BioID^{x}
Dong [DZL11]	Modell	Probab. Kaskade	FRGC, CASPERL, AR, PF01
Rusek [RG14]	Modell	2D-DCT, MLP	BioID^{t}
Markuš [Mar+14]	Modell	Regressionsbäume	gi4e, UULM
Kothari [KM96]	Merkmal	Gradienten, Histogr.	
Asadifard [AS10]	Merkmal	Histogr. Morph. Op.	
Türkan [TPC07]	Merkmal	Histogr. Projektion	
Zhang [ZYK13]	Merkmal	Histogr. Projektion	
Bai [BSW06]	Merkmal	Symm., Schwellenwert	
Timm [TB11]	Merkmal	Vektorfeld Gradient	
Pang [Pan+15]	Hybrid	Isophoten, alignm.	$\text{BioID}^{\bar{g}}$, $\text{FERET}^{\bar{g}}$, LFPW
Valenti [VG12]$^{\bar{g}}$	Hybrid	Isophoten, SIFT	BioID^{x}
Kroon [Kro+09]	Modell	LBP, Bayes	YaleB
Vater [VP16a]$^{\bar{g}}$	Modell	Isophoten, Kaskade	BioID, YaleB, EX, Utrecht

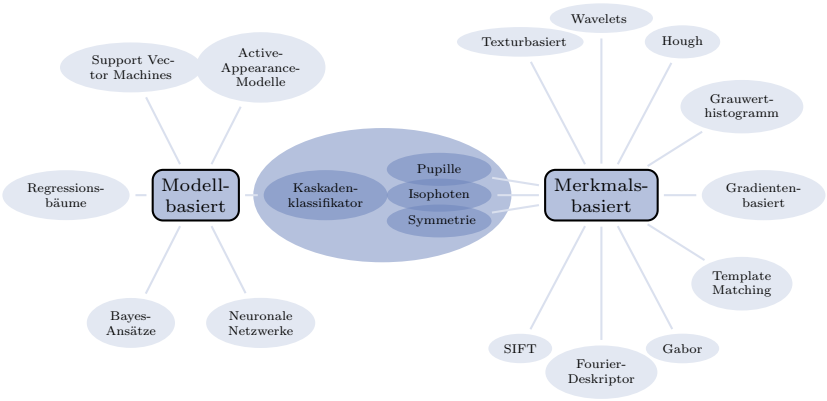


Abbildung 3.2 Einordnung des hier verfolgten Ansatzes zur Irislokalisation in einen Überblick über den Stand der Technik. Bei der Grafik ist zu beachten, dass die modellbasierten Ansätze mit merkmalsbasierten Ansätzen kombiniert werden können, diese dann jedoch im Gegensatz zu rein merkmalsbasierten Ansätzen von einem Trainingsprozess und damit von Trainingsdaten abhängen.

Durch Fusion dreier die Iris bzw. Pupille beschreibender Charakteristika soll hier eine robuste, gut generalisierende Lösung vorgestellt werden.

Insbesondere werden auf dem Gebiet der präzisen Irislokalisation die folgenden, den Stand der Technik erweiternden, Beiträge geliefert [VP16a]:

1. Der erste Beitrag besteht in einem neuen Gewichtungsschema für die Verschiebungsvektoren der Isophoten, welches das Ausschließen von mit Fehlern und Ungenauigkeiten behafteter Information, die sich aus der Radiusermittlung aus dem Isophotenansatz ergibt, unterstützt. Es wird gezeigt, wie durch Hinzunahme heller Zentren, wie sie beispielsweise durch Reflexionen in der Pupille entstehen können, die Genauigkeit begünstigend in den Zentrumsvotingalgorithmus eingebaut werden kann.

2. Zweitens wird gezeigt, wie die in Kap. 2 beschriebene Implementierung eines konventionellen Kaskadenklassifikators modifiziert und erweitert werden kann, um einen quasi-kontinuierlichen Ausgang statt eines binären zu produzieren. Durch Hinzunahme der

Information aus dem Kaskadenklassifikator werden neben den drei auf der rechten Seite in Abb. 3.2 dunkelblau hinterlegten merkmalsbasierten Ansätzen durch die in Kap. 2.5.1 beschriebenen, im Kaskadenklassifikator verwendeten Merkmale, weiterhin komplexe Gradientenmerkmale, komplexe Texturmerkmale sowie Kanteninformation (Haar-Merkmal) zur Entscheidungsfindung hinzugezogen.

3. Die Bestimmung des Pupillenzentrums wird schließlich durch Fusion der beiden Schritte durchgeführt. Es wird gezeigt, wie der merkmalsbasierte Isophotenansatz effizient mit dem modifizierten Kaskadenklassifikator kombiniert werden kann, um die Ambiguität der aus dem Isophotenansatz stammenden Lokalisierung des Irismittelpunktes in Form zahlreicher lokaler Maxima zu bewältigen. Dabei werden Kandidaten aus dem Voting der Iriszentren mittels des quasi-kontinuierlichen Ausgangs des modifizierten Kaskadenklassifikators weiter evaluiert. Dadurch, dass nur wenige Punkte durch den Klassifikator ausgewertet werden müssen (im Vergleich zu einer Vielzahl in einem *Sliding-Window*-Ansatz), kann die Klassifikation effizient durchgeführt werden.

Die zur präzisen Irislokalisation verwendete Fülle verdeutlich hierbei die Dichte an Informationen, die im erforschten Ansatz zur Anwendung kommt, um die angestrebte Robustheit zu gewährleisten

3.4 Kandidaten für die Suche des Iriszentrums

Der präsentierte Ansatz nimmt an, dass jedes Pixel $\mathbf{u} = (u, v)^{\mathrm{T}}$ Teil einer Isophoten $\boldsymbol{\gamma}(\mathbf{u})$ ist, welche als Kontur konstanter Intensität im Grauwertbild verstanden werden kann,

$$g\left(\boldsymbol{\gamma}(\mathbf{u})\right) = \text{const.}\,, \tag{3.1}$$

wobei $g(\mathbf{u})$ den Grauwert eines Bildes an der Stelle $\mathbf{u}$ beschreibt. Es kann gezeigt werden, dass die Richtung sowie die Krümmung $\kappa\left(\mathbf{u}\right)$ jedes Pixels unter Annahme eines intrinsichen Koordinatensystems $(\xi, \eta)^{\mathrm{T}}$, wobei ξ in Richtung der örtlichen Tangente und η orthogonal dazu orientiert ist, berechnet werden kann, indem der Quotient der zweiten

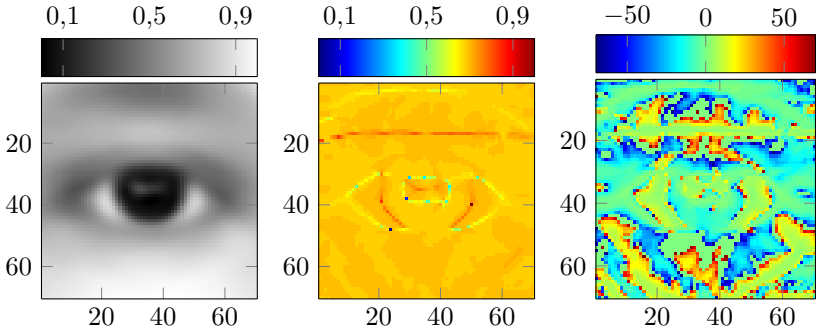
Ableitung orthogonal zur Gradientenrichtung des Pixel $g_{\eta\eta}(\mathbf{u})$ ins Verhältnis zur Ableitung in Richtung des Gradienten $g_\xi(\mathbf{u})$ gesetzt wird, (Verbeek [Ver85]):

$$\kappa\left(\mathbf{u}\right) = \frac{g_{\eta\eta}(\mathbf{u})}{g_\xi(\mathbf{u})} = -\frac{\left(g_{uu}\, g_v^2 - 2\, g_u\, g_v\, g_{uv} + g_{vv}\, g_u^2\right)}{\left(g_u^2 + g_v^2\right)^{\frac{3}{2}}}\,. \tag{3.2}$$

Auf der rechten Seite von Glg. (3.2) wurde die unabhängige Variable $\mathbf{u}$ zur besseren Lesbarkeit weggelassen. Das Reziproke der Isophotenkrümmung führt für jedes Pixel direkt zur Schätzung des Radiusses:

$$r(\mathbf{u}) = \frac{1}{\kappa\left(\mathbf{u}\right)}\,. \tag{3.3}$$

Abbildung 3.3 zeigt das Bild eines mittleren Auges 3.3(a) gebildetet aus von Hand ausgeschnittenen und zentrierten Augenbildern aus dem BioID-Datensatz [Bio01] sowie dessen Krümmung 3.3(b) und örtlichen Radius 3.3(c), welcher sich als pixelweise Inverse der Krümmung ergibt (siehe Glg. (3.3)).
In Abb. 3.3(a) wurden Augen auf die Iris zentriert ausgeschnitten und auf 70 Pixel Kantenlänge quadratisch skaliert. Die Originalgröße in Pixeln der Augenregion im Datensatz liegt bei $b_{\max} = 60, h_{\max} = 48$, der mittlere Augenabstand bei $54{,}9$ Pixel [Kro+09]. Die nebenstehende Abb. 3.3(b) zeigt die zugehörige Krümmung nach Glg. (3.2). Die Isophoten sind gut an den monochromatischen Linien entlang der Sklera außen sowie entlang der Iris und der Reflexion innerhalb der Iris zu erkennen. Der korrespondierende örtliche Radius ist in Abb. 3.3(c) gezeigt, wobei Rot einem positiven, Blau einem negativen Radius entspricht. Die runden Strukturen konstanten Radiusses sind trotz Unvollkommenheit der theoretisch geschlossenen konzentrischen Kreise entlang der Iris zu erkennen. Unter der Annahme einer mathematisch positiven Rotationsrichtung entlang der Isophotenkurve $\boldsymbol{\gamma}(\mathbf{u})$ hat der Radius ein positives Vorzeichen für eine hellere rechte Seite (Gradient zeigt nach rechts) [LHR05]. Unter Vernachlässigung von sichtbaren numerischen Artefakten lässt sich dies gut an den positiven Werten für den Radius entlang des Übergangs Iris–Sklera erkennen, welcher einen Bildbereich mit wichtigen Informationen für die vorgestellte Methode darstellt.



(a) Gemitteltes Auge (BioID-mean) aus 1782 Bildern aus der BioID-Datenbank.

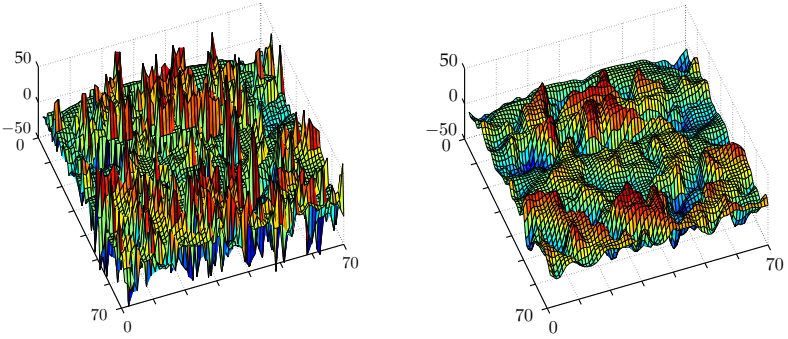
(b) Zu Abb. 3.3(a) gehörende Krümmung (normiert).

(c) Zu Abb. 3.3(a) gehörende Radiuskarte als örtliche, pixelweise Inverse der Krümmung.

Abbildung 3.3 Veranschaulichung der berechneten Isophoten sowie des daraus resultierenden Radius anhand eines gemittelten Augenbildes.

Da die Ermittlung der Radien eine zentrale Rolle für die vorgestellte Methode spielt, sollen in den Abbildungen 3.4 und 3.5 die Radienkarten detailliert dargestellt werden.
Anhand Abb. 3.4(a) erkennt man gut die niedrige Genauigkeit der Methode für eine direkte Bestimmung des Radiusses, was eine Bestimmung des Iriszentrums exklusive der Nutzung des Radiusses nicht zum Ziel führen lässt (siehe auch Kap. 3.6.3). Um dennoch die im Radius beinhaltete, wertvolle Information besser darzustellen, wurden die Rohdaten tiefpassgefiltert und in Abb. 3.4(b) dargestellt.
Ein topographischer Ring, welcher einen konstanten Radius repräsentiert, ist entlang des Überganges Sklera–Iris gut zu erkennen. Da hier ein – bei mathematisch positiver Verfolgung der Grenzkurve – Verlauf von dunkel nach hell stattfindet (Bildgradient zeigt in Richtung der Sklera), zeigt der tiefpassgefilterte Radius sehr gut die positiven Radien nahezu konstanten Betrages. Während dieser Ring positiven Radiusses in Abb. 3.5(b) zu erkennen ist, kann man die innerhalb der Iris liegenden topographischen Strukturen in Abb. 3.5(a), die zur Reflexion im Auge in Abb. 3.3(a) gehören, beobachten. Die Strukturen resultieren aus dem hellen Zentrum innerhalb der dunklen Iris und können durch die auftretenden negativen Radien charakterisiert werden. Es soll angemerkt

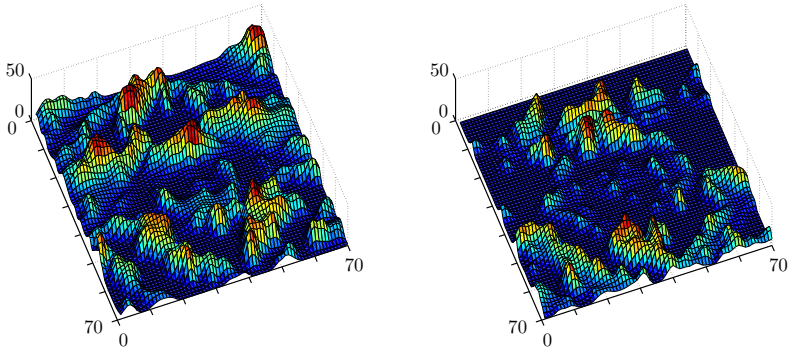
werden, dass die hier dargestellten Informationen auf einem gemittelten Augenbild, welches weder Verdeckungen, noch starke Reflektionen oder eine sehr geringe Auflösung aufweist (es kann angenommen werden, dass geringaufgelöste Bilder, welche zum Bildaufbau beigetragen haben, herausgemittelt wurden) und somit schlechtere Daten bei realen Bildern zu erwarten sind. Beispielhaft sind zwei Bilder aus der BioID-Datenbank sowie deren Krümmungen und Radien in Abb. 3.6 gezeigt.



(a) Ungefilterte Radien für das mittlere Auge in Abb. 3.3(a).

(b) Mit einem Gaußfilter mit $\sigma = 1$ und einer Breite von $4 \cdot \sigma + 1$ gefilterte Karte des Radiusses.

Abbildung 3.4 Vorzeichenbehaftete Radien des Bildes 3.3(a).



(a) Nur negative Radien, Rohdaten mit gleichem Filter wie in Abb. 3.4(b) gefaltet.

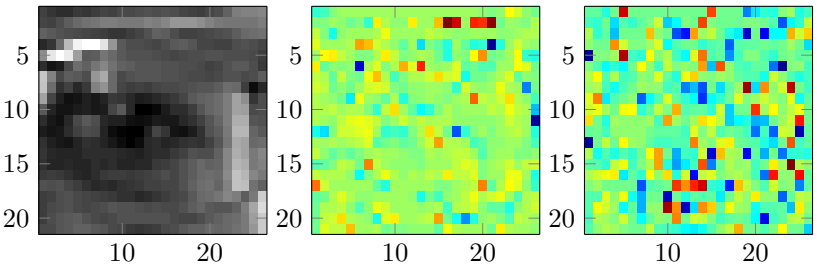
(b) Betrag der positiven Radien, Rohdaten mit gleichem Filter wie in Abb. 3.4(b) gefaltet.

Abbildung 3.5 Negative und positive Radien des Auges aus 3.3(a) separat dargestellt.

Die Bilder wurden nicht vorverarbeitet. Indem man die Orientierung jedes Pixels ausnutzt, welche durch die Ableitung seiner Grauwerte gegeben ist, kann man, basierend auf den bestimmten Radien, einen Verschiebungsvektor

$$\mathbf{d}(\mathbf{u}) = r(\mathbf{u}) \cdot (g_u(\mathbf{u}), g_v(\mathbf{u}))^{\mathrm{T}} \tag{3.4}$$

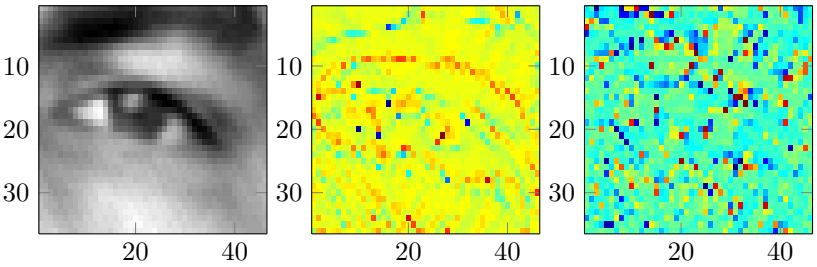
für jedes einer Isophoten zugehörige Pixel innerhalb eines Bildausschnittes berechnen.



(a) Linkes Auge der Datei BioID0767 (BioID0767L), unbearbeitet.

(b) Zu Abb. 3.6(a) gehörende Krümmung.

(c) Zu Abb. 3.6(a) gehörende Radiuskarte.



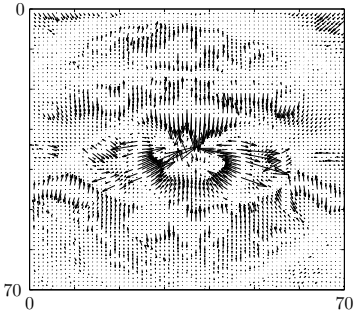
(d) Linkes Auge der Datei BioID0005 (BioID0005L), unbearbeitet.

(e) Zu Abb. 3.6(d) gehörende Krümmung.

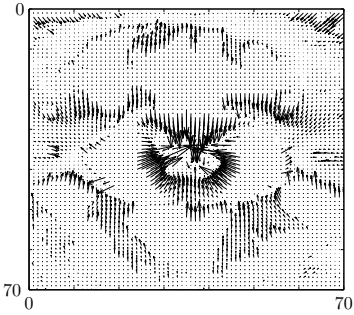
(f) Zu Abb. 3.6(d) gehörende Radiuskarte.

Abbildung 3.6 Veranschaulichung der berechneten Isophoten sowie des daraus resultierenden Radiusses anhand typischer Beispielbilder aus dem BioID-Datensatz. Die Bilder zeigen ein durch eine Brille und Reflexionen verdecktes Auge niedriger Auflösung sowie ein für den Datensatz qualitativ gutes Augenbild.

Abbildung 3.7 zeigt Verschiebungsvektoren für das mittlere Auge resultierend aus der BioID-Datenbank, während Abb. 3.8 die Verschiebungsvektoren der Beispielbilder BioID0767L 3.6(a) und BioID0005L 3.6(d) zeigt.
Die dominanten Richtungen der Verschiebungsvektoren des mittleren Auges sowie des BioID0005L entspringen der Grenze Iris–Sklera und zeigen deutlich in Richtung des Pupillenzentrums. Es sind zudem unbeabsichtigte Ansammlungen im Bereich der Augenwinkel zu erkennen, welche in fehlerhafte lokale Maxima für die Schätzung des Pupillenzentrums resultieren. Aufgrund der sehr niedrigen Auflösung des Bildes BioID0767L sind die Ziele der einzelnen Verschiebungsvektoren nicht so deutlich ausgeprägt. Wie in späteren Kapiteln gezeigt wird, wird auch hiermit durch Skalierung und entsprechende Gewichtung umgegangen, um präzise Ergebnisse zu erzielen.

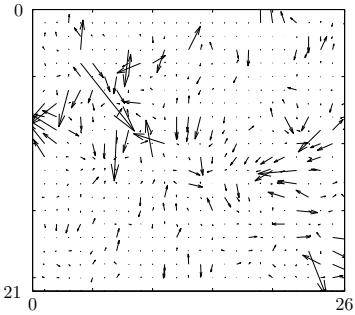


(a) Verschiebungsvektoren resultierend aus positiven sowie negativen Radien.

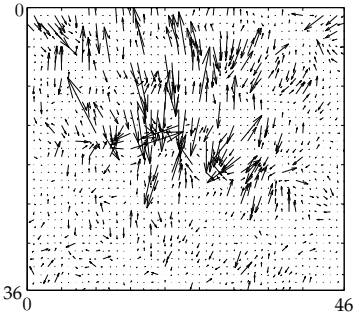


(b) Verschiebungsvektoren resultierend aus ausschließlich positiven Radien.

Abbildung 3.7 Verschiebungsvektoren basierend auf Glg. (3.4) für BioIDmean (Abb. 3.3(a)).



(a) Verschiebungsvektoren resultierend aus positiven sowie negativen Radien.



(b) Verschiebungsvektoren resultierend aus positiven sowie negativen Radien.

Abbildung 3.8 Verschiebungsvektoren basierend auf Glg. (3.4) für BioID0767L (Abb. 3.6(a)) (links) und BioID0005L (Abb. 3.6(d)) (rechts, skaliert).

3.4.1 Gewichtungen der Verschiebungsvektoren

Anstatt die Zentrumsvotes für die einzelnen Pixel, zu denen die Verschiebungsvektoren zeigen, einfach aufzuaddieren, werden diese mit der Rundheit

$$\mathrm{w}_{\mathrm{Rund}}(\mathbf{u}) = \sqrt{g_{uu}^2(\mathbf{u}) + 2g_{uv}^2(\mathbf{u}) + g_{vv}^2(\mathbf{u})} \tag{3.5}$$

der entsprechenden Isophoten am Startpunkt des Verschiebungsvektors $\mathbf{d}(\mathbf{u})$ [VG08] gewichtet. Das Gewichtungsschema wird erweitert, indem nicht nur der Startpunkt, sondern auch der Endpunkt $\mathbf{u}' = (u', v')^{\mathrm{T}}$ der Vektoren (also im Idealfall das Pupillenzentrum) mit in die Gewichtung einbezogen wird. In Abwesenheit sehr starker Reflexionen kann angenommen werden, dass das Pupillenzentrum sehr niedrige Grauwerte hat. Um dies zu berücksichtigen, wird eine Zielgewichtungskarte

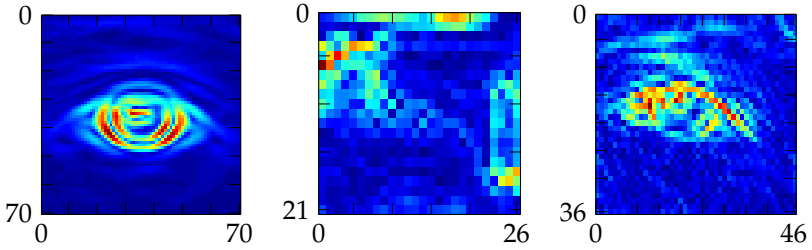
$$\mathrm{w}_{\mathrm{Pup}}(\mathbf{u}) = \max(g(\mathbf{u})) - g(\mathbf{u}) \tag{3.6}$$

erstellt und das totale Gewicht für einen Verschiebungsvektor, der einer Isophoten am Pixel $\mathbf{u}$ entspringt, mit

$$\mathrm{w}(\mathbf{u}) = \mathrm{w}_{\mathrm{Rund}}(\mathbf{u}) \cdot \mathrm{w}_{\mathrm{Pup}}(\mathbf{u}') \tag{3.7}$$

berechnet. Für $\mathrm{w}_{\mathrm{Pup}}(\mathbf{u})$ wird eine *Nearest-Neighbor*-Interpolation angewandt, um vereinzelte Pixel schwarz zu setzen und vor dem Erstellen der Karte eine Histogrammstauchung durchgeführt. Weiterhin wurde die Schätzung des örtlichen Radiusses verwendet, um eine Schätzung der Größe des Zielgebietes durchzuführen, welche bei der Bildung der Zielgewichtskarte als Wertebereich herangezogen wird. Durch Verwenden der Gewichtung in Glg. (3.7) können Verschiebungsvektoren, die von Stellen starker Rundheit, wie beispielsweise den Augenlidern, entspringen und zu fehlerhaften lokalen Maxima für das Iriszentrum beitragen, unterdrückt werden. Diese Pixel tragen zu lokalen Maxima unterhalb der Iris bei, da der Bogen der Augenlider einen Radius liefert, welcher zu einem Zentrum zeigt, das von der Iris abweicht. Die Rundheit der drei betrachteten Beispielbilder ist in Abb. 3.9 gezeigt, während Abb. 3.10 das entsprechende Gewicht $\mathrm{w}_{\mathrm{Pup}}(\mathbf{u})$ zeigt.
Es ist gut zu erkennen, dass insbesondere für Bilder sehr niedriger Auflösung fehlerhafte Verschiebungsvektoren, die aufgrund der Rundheit ein starkes Gewicht erfahren, allerdings keinen sinnvollen Beitrag

zum Zentrumsvoting liefern, ausgeschlossen werden können. Durch Anwenden von Glg. (3.7) tragen so ausschließlich Verschiebungsvektoren, die von Isophoten mit einer starken Rundheit starten *und* in Regionen von niedrigen Grauwerten enden, einen merklichen Beitrag zum Zentrumsvoting bei. Um ein finales Pupillenzentrum zu determinieren, wird an die Idee aus [VG08] angeknüpft und die Zielpunkte von Verschiebungsvektoren aufaddiert und anschließend mit einem Gaußfilter tiefpassgefiltert, um Votes zusammenzufassen und den Einfluss ungewollter Votes durch Bildrauschen und andere Artefakte, wie Reflexionen, zu unterdrücken.

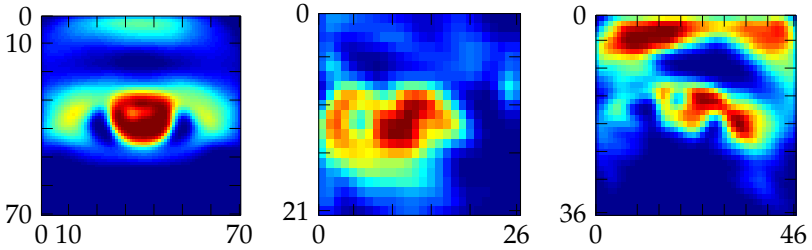


(a) Zu Abb. 3.3(a) gehörende Rundheit w_{Rund}.

(b) Zu Abb. 3.6(a) gehörende Rundheit w_{Rund}.

(c) Zu Abb. 3.6(d) gehörende Rundheit w_{Rund}.

Abbildung 3.9 Rundheit der Bilder BioIDmean (3.3(a)), BioID0767L (3.6(a)) und BioID0005L (3.6(d)) basierend auf Glg. (3.5).



(a) Zu Abb. 3.3(a) gehörendes Gewicht w_{Pup}.

(b) Zu Abb. 3.6(a) gehörendes Gewicht w_{Pup}.

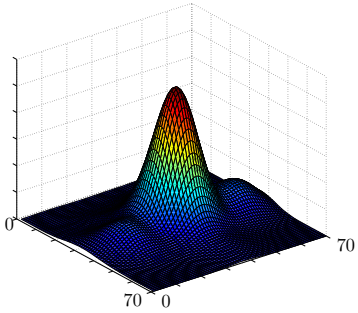
(c) Zu Abb. 3.6(d) gehörendes Gewicht w_{Pup}.

Abbildung 3.10 Gewichtung durch Grauwerte für die Bilder BioIDmean (3.3(a)), BioID0767L (3.6(a)) und BioID0005L (3.6(d)) basierend auf Glg. (3.6).

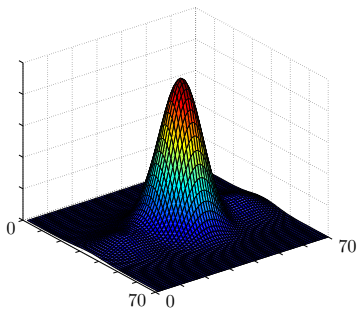
Mit $\mathbf{u}' = \mathbf{u} + \mathbf{d}(\mathbf{u})$ wird die Zentrumsvotingmap $\mathrm{ZM}(\mathbf{u})$ dann durch Akkumulation der Gewichte an den Endpunkten der Verschiebungsvektoren durch

$$\mathrm{ZM}(\mathbf{u}') = \sum_{\mathbf{u} \in \Omega_g} \mathrm{w}_{\mathrm{Rund}}(\mathbf{u}) \cdot \mathrm{w}_{\mathrm{Pup}}(\mathbf{u}') \tag{3.8}$$

erstellt. Die Abbildungen 3.11, 3.12 und 3.13 zeigen die $\mathrm{ZM}(\mathbf{u})$ sowohl mit der Rundheit allein als auch mit der kombinierten Gewichtung für die drei Augenbilder, wobei empirisch ein für die Bildgröße entsprechend sinnvolles σ gewählt wurde.



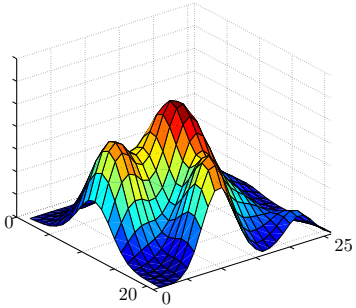
(a) Zentrumvotingmap zu Abb. 3.3(a) für eine Gewichtung durch Rundheit, tiefpassgefiltert mit $\sigma = 5{,}5$.



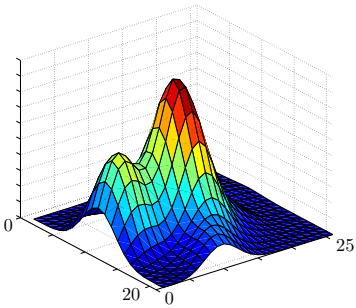
(b) Zentrumvotingmap zu Abb. 3.3(a) mit kombinierter Gewichtung, tiefpassgefiltert mit $\sigma = 5{,}5$.

Abbildung 3.11 Zentrumvotingmaps für BioIDmean (Abb. 3.3(a)) basierend auf den Gewichten in Glg. (3.5) und (3.7).

Insbesondere in Abb. 3.12(b) lässt sich gut erkennen, wie das Voting für das Zentrum deutlich robuster, zu erkennen am größerem Verhältnis zwischen den stärksten beiden lokalen Maxima, wird. Die neue Gewichtung generiert weniger fehlerhafte Information und trägt zu stärker ausgeprägten Votes im Pupillenzentrum bei. Präzise ausgedrückt nimmt die Relation zwischen den ersten beiden Maxima für das Bild BioID0767L von $77{,}4\,\%$ auf $60{,}1\,\%$ ab.

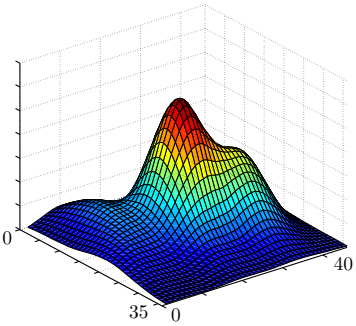


(a) Zentrumvotingmap zu Abb. 3.6(a) für eine Gewichtung durch Rundheit, tiefpassgefiltert mit $\sigma = 2{,}0$.

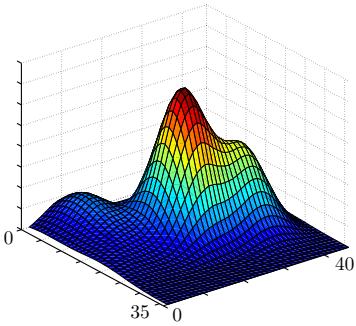


(b) Zentrumvotingmap zu Abb. 3.6(a) mit kombinierter Gewichtung, tiefpassgefiltert mit $\sigma = 2{,}0$.

Abbildung 3.12 Zentrumvotingmaps für BioID0767L (Abb. 3.6(a)) basierend auf den Gewichten in Glg. (3.5) und (3.7).



(a) Zentrumvotingmap zu Abb. 3.6(d) für eine Gewichtung durch Rundheit, tiefpassgefiltert mit $\sigma = 4{,}0$.

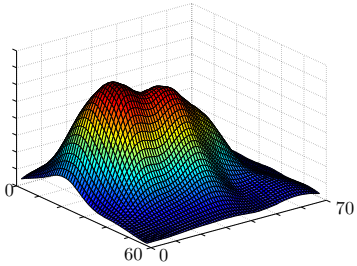


(b) Zentrumvotingmap zu Abb. 3.6(d) mit kombinierter Gewichtung, tiefpassgefiltert mit $\sigma = 4{,}0$.

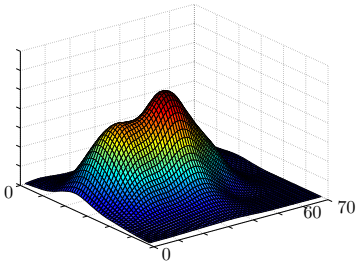
Abbildung 3.13 Zentrumvotingmaps für BioID0005L (Abb. 3.6(d)) basierend auf den Gewichten in Glg. (3.5) und (3.7).

Das Clustern der Votes und damit die Wahl der Breite des Tiefpassfilters spielt eine zentrale Rolle. Da die Größe der Eingangsbilder unbekannt ist, soll, um eine sehr gute Generalisierbarkeit der Methode zu gewährleisten, durchgängig eine feste Parametrierung gewählt werden. In dieser Arbeit wurden hierzu die Bilder auf eine Breite von 70 Pixeln skaliert und ein $\sigma = 5{,}5$ des Tiefpassfilters gewählt. Diese Werte wurden empi-

risch anhand des Datensatzes BioID gefunden, siehe hierzu Kap. 3.6.2. Beispielhaft soll der Einfluss der Breite der Tiefpassfilter anhand der folgenden Abbildungen gezeigt werden. Die Abb. 3.14 und 3.15 zeigen die Zentrumvotingmaps für die auf 70 Pixel Breite skalierten Bilder BioID0767L und BioID0005L.

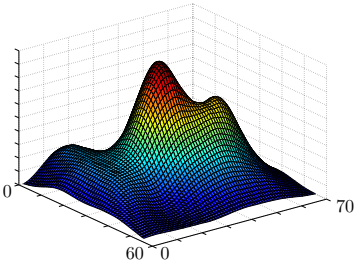


(a) Zentrumvotingmap zu für das skalierte Bild 3.6(a) für eine Gewichtung durch Rundheit, tiefpassgefiltert mit $\sigma = 5{,}5$.

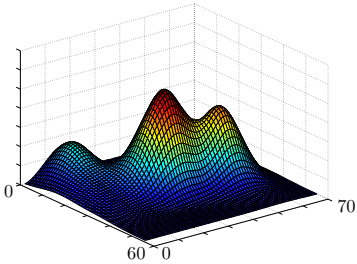


(b) Zentrumvotingmap für das skalierte Bild 3.6(a) mit kombinierter Gewichtung, tiefpassgefiltert mit $\sigma = 5{,}5$.

Abbildung 3.14 Zentrumvotingmaps für das auf 70 Pixel Breite skalierte Bild BioID0767L (3.6(a)).



(a) Zentrumvotingmap für das skalierte Bild 3.6(d) für eine Gewichtung durch Rundheit, tiefpassgefiltert mit $\sigma = 5{,}5$.



(b) Zentrumvotingmap für das skalierte Bild 3.6(d) mit kombinierter Gewichtung, tiefpassgefiltert mit $\sigma = 5{,}5$.

Abbildung 3.15 Zentrumvotingmaps für das auf 70 Pixel Breite skalierte Bild BioID0005L (3.6(d)).

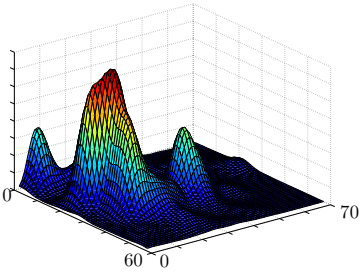
Der Unterschied zu den Ergebnissen der unskalierten Bilder mit von Hand angepassten Filterbreiten ist deutlich erkennbar, insbesondere der positive Einfluss der neuartigen Gewichtung. Während sich nach Ska-

lierung für BioID0005L mit der neuen Gewichtung weiterhin eine klare Abgrenzung verschiedener lokaler Maxima zeigt, entspricht das globale Maximum für BioID0767L ohne neuartige Gewichtung nicht mehr der tatsächlichen Position der Pupille. Durch Einbringen der neuartigen Gewichtung ist auch nach Skalierung eine Schätzung des Pupillenzentrums möglich. Im Weiteren soll untersucht werden, wie die Schätzung des Pupillenmittelpunktes, insbesondere bei Reflexionen, wie sie bei BioID0005L auftauchen, robuster und markanter gestaltet werden kann.

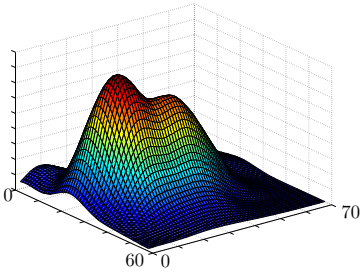
3.4.2 Helle und dunkle Zentren

Im Gegensatz zu anderen Ansätzen in der Literatur soll das Potential untersucht werden, nicht nur die Votes für dunkle Zentren, welche aus positiven Werten für den lokalen Radius resultieren, sondern auch solche, die aus hellen Zentren resultieren, auszunutzen. Der Gedanke hierbei ist der folgende: Der Bereich der Sklera macht einen großen, hellen, tendentiell nicht konzentrischen Bereich innerhalb der Augenregion aus; hier sind niedrige Werte der Gewichte, insbesondere von w_{Pup}, zu erwarten. Helle Spots innerhalb der Iris korrespondieren, aufgrund der sphärischen Krümmung des Augapfels sowie zumeist kreisrunder Lichtquellen, mit rundlichen Reflexionen eben dieser Lichtquellen. Aufgrund der *Nearest-Neighbor*-Interpolation im Anschluss an die Anwendung von Glg. (3.6) lassen sich die im Vergleich zum Irisdurchmesser meist räumlich wenig ausgeprägten hellen Zentren mit hohen Grauwerten (kleines Gewicht w_{Pup}) mit einem Gewicht ausstatten, welches einen sinnvollen Beitrag zum Voting liefern kann; Isophoten und eine entsprechende Rundheit sind ohnehin vorhanden, siehe Abb. 3.9(c).

Das zusätzlich Einsetzen von hellen Zentren kann das eigentliche Voting der dunklen Zentren weiter verstärken und helfen, fälschlich angenommene Zentren (z. B. Augenwinkel) zu unterscheiden. Die Votingmaps berechnet aus den hellen sowie kombiniert aus den hellen und dunklen Zentren für die Bilder BioID0767L und BioID005L sind in den Abb. 3.16 und 3.17 zu sehen. Während die Hinzunahme des Votings durch helle Zentren für BioID0767L die ZM in der Region der Iris verstärkt, erkennt man, dass die Werte der beiden stärksten Votes für BioID0005L nun weiter auseinanderliegen.

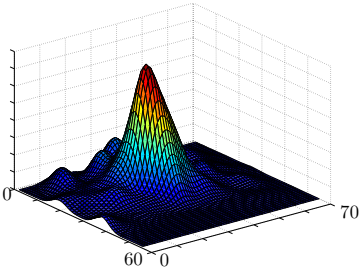


(a) ZM der hellen Zentren mit kombinierter Gewichtung, tiefpassgefiltert mit $\sigma = 5{,}5$.

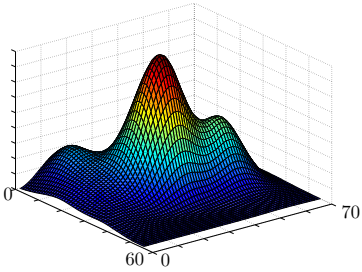


(b) ZM mit hellen und dunklen Zentren mit kombinierter Gewichtung, tiefpassgefiltert mit $\sigma = 5{,}5$.

Abbildung 3.16 Zentrumvotingmaps für BioID0767L (Abb. 3.6(a)), vgl. ZM in Abb. 3.14.



(a) ZM der hellen Zentren mit kombinierter Gewichtung, tiefpassgefiltert mit $\sigma = 5{,}5$.



(b) ZM mit hellen und dunklen Zentren mit kombinierter Gewichtung, tiefpassgefiltert mit $\sigma = 5{,}5$.

Abbildung 3.17 Zentrumvotingmaps für BioID0005L (Abb. 3.6(d)), vgl. ZM in Abb. 3.15.

Ein Vergleich zeigt eine Erhöhung der relativen Ausprägung der Schätzung für das Pupillenzentrum zum zweitstärksten lokalen Maximum, welches sich im Bereich des Augenwinkels befindet, wobei eine Änderung von $75{,}6\,\%$ (Abb. 3.15(b), dunkle Zentren) auf $48{,}6\,\%$ (Abb. 3.17(b), helle und dunkle Zentren) zu beobachten ist. Diese Erhöhung unterstützt eine Unterscheidung der beiden lokalen Maxima positiv. Es lässt sich beobachten, dass Reflexionen innerhalb der Sklera weniger prominent sind als solche in der Iris, was sich in höheren Werten in Glg. (3.6) für helle Zentren im Bereich der Iris im Vergleich zur Sklera ausdrückt. Diese Beobachtung ist wichtig, da insbesondere der Bereich der Augenwinkel große Werte für die Rundheit aufweist (siehe Abb. 3.9(c)) und kleine

Werte für das Gesamtgewicht in dieser Region erzielt werden sollen. Wie in Kap. 3.6.3 näher erläutert, zeigt sich, dass die hellen Zentren in den aggregierten Votes für helle und dunkle Zentren nach einer Tiefpassfilterung mit einem Gaußkern positiv zur finalen Schätzung des Iriszentrums beitragen können.

3.4.3 Skalenraum

Um auf die unbekannte Auflösung und damit Größe der Augenbilder, welche sich aus der Verwendung verschiedener Aufnahmeauflösungen beispielsweise einer Webcam sowie veränderlicher Entfernungen eines Nutzers vom Sensor ergeben, einzugehen und somit die Schätzung des Pupillenmittelpunktes unanfällig gegen Skalierungen des Eingangs zu gestalten, wurde die Idee der skaleninvarianten Berechnung von Merkmalen [Lin94; Lin98] auf die Erstellung der ZM angewandt. Hierzu wird das Eingangsbild, ausgehend von der Originalgröße, durch verschiedene Oktaven mit dem Faktor 2 unterabgetastet. Innerhalb der Oktaven werden die Bilder anschließend mit einem Gaußfilter gefaltet, wobei die Breite $\sqrt{\sigma}$ des Filters so iteriert wird, dass σ sich gerade in N Schritten innerhalb einer Oktave verdoppelt. Durch Berechnung auf verschiedenen Bildoktaven und Verdoppelung des σ innerhalb einer Oktave kann somit gewährleistet werden, dass das Verhältnis der Breiten Bild-zu-Filter für das breiteste Filter der höheren Oktave zum schmalsten Filter der niedrigeren Oktave gleich ist, während man durch das Unterabtasten einen recheneffizienten Ansatz verfolgen kann. Abbildung 3.18 zeigt die so erstellte Gaußpyramide für BioIDmean. Durch Subtraktion im Ortsbereich der tiefpassgefilterten Bilder werden so bandpassgefilterte Bilder erstellt, welche als Repräsentation des Bildes auf unterschiedlichen Skalen interpretiert werden können (Abb. 3.19). Die ZM wird anschließend auf jeder Skale berechnet und die Ergebnisse aufaddiert. Die Abb. 3.20 und 3.21 zeigen die zugehörigen Rohdaten sowie die mit $\sigma = 5{,}5$ gefilterten Zentrumvotingmaps. Die vorgeschlagene Methode liefert im Idealfall ein einziges dominantes Maximum. Wie die Beispiele in den Abb. 3.16 und 3.17 zeigen, treten bei realen Bildern mehrere lokale Maxima auf, bei denen das globale Maximum, resultierend aus dem vorgestellten Isophotenansatz, nicht notwendigerweise die beste Wahl für das Iriszentrum darstellt. Aus diesem Grund soll eine Möglichkeit zur weiteren Klassifikation lokaler Maxima vorgestellt werden.

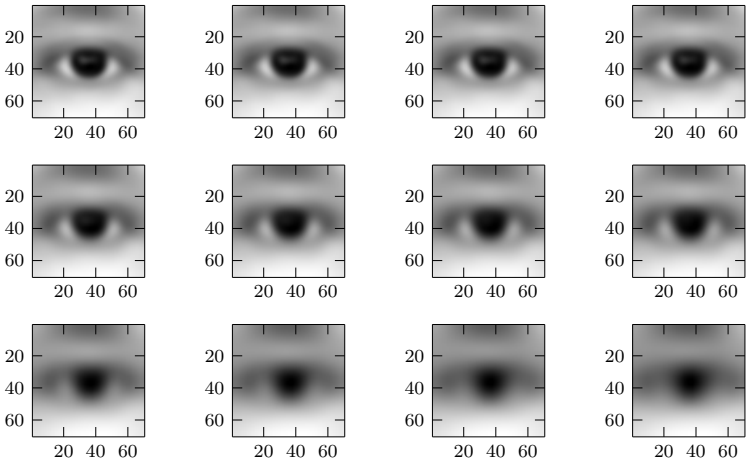


Abbildung 3.18 Gaußpyramide für BioIDmean mit 3 Oktaven (Zeilen) und über $N = 4$ Skalen (Spalten) verdoppelte Filterbreite. Die Abbildungen in der zweiten und dritten Zeile resultieren durch Skalierung der gefilterten Bilder.

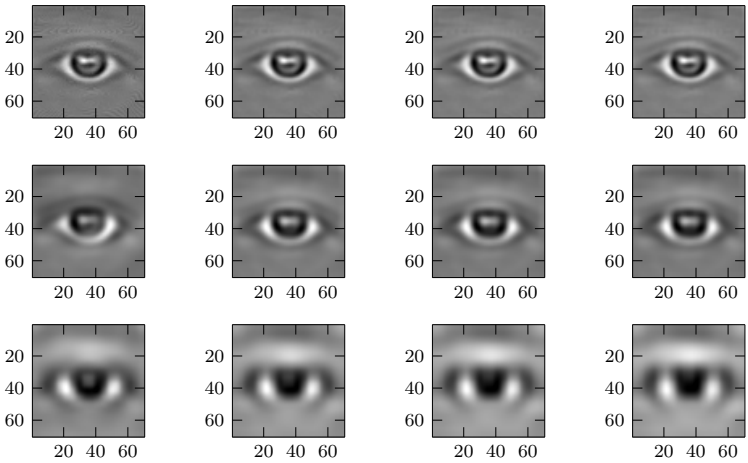


Abbildung 3.19 Bandpassgefiltere Darstellungen für BioIDmean mit 3 Oktaven und 4 Skalen.

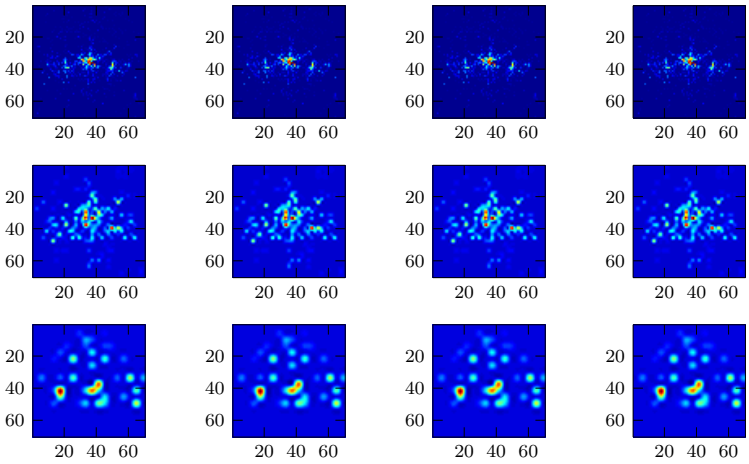


Abbildung 3.20 Zentrumsvotingmaps für BioIDmean mit 3 Oktaven und 4 Skalen.

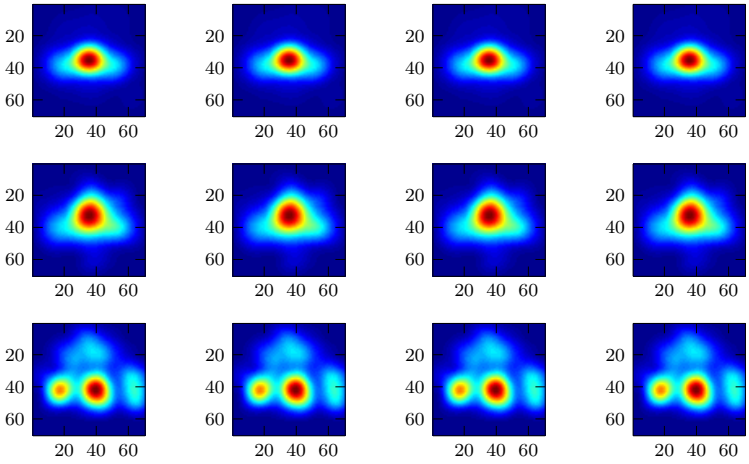


Abbildung 3.21 Zentrumsvotingmaps für BioIDmean mit 3 Oktaven und 4 Skalen, tiefpassgefiltert mit $\sigma = 5{,}5$.

3.5 Quasi-kontinuierlicher Kaskadenklassifikator

Um eine weitere Klassifikation der zuvor bestimmen Maxima der ZM durchzuführen, wird als Ausgangspunkt der in Kap. 2 vorgestellte Klassifikator als Rahmenwerk verwendet. Um die Information aus der ZM mit Information aus dem Klassifikator fusionieren zu können, werden am Ausgang des Kaskadenklassifikators quasi-kontinuierliche Werte im Bereich $[0, \ldots, 1]$ anstatt eines binären Ausgangs benötigt.

3.5.1 Konventionelle Implementierung

Der Klassifikator wird aufgebaut, indem einzelne Stufen (starke Klassifikatoren) $\mathcal{K}(g(\mathbf{u}), \tau)$ in Reihe geschaltet werden. Jede einzelne Stufe besteht aus vielen schwachen Klassifikatoren und deren Hypothesen $h(s, t, \Theta)$ der Merkmale $\mathbf{m}_\mu$, deren Merkmalsgewichte $\alpha\left(\epsilon_{\mu^*,\iota}, t\right)$ mit deren Schwellenwerten $\Theta(t)$ und dem Stufenschwellenwert τ.

Wie bereits in Kap. 2 vermerkt, umfassen die schwachen Klassifikatoren örtlich lokale, fixierte Merkmale innerhalb der Trainingsbildfenstergröße, für deren Merkmalswerte während des Trainings durch *Boosting* (siehe Abb. 2.7) abhängig vom Klassifizierungsfehler das Gewicht $\alpha\left(\epsilon_{\mu^*,\iota}, t\right)$ festgelegt wird. Das Gewicht jedes schwachen Klassifikators wird zwar in Abhängigkeit der Fähigkeit des Merkmals, zwischen positiven und negativen Beispielbildern differenzieren zu können, bestimmt; bei der endgültigen Entscheidung, ob der Stufenschwellenwert τ erreicht wird (Glg. (2.48), 2.49, 2.50, 2.54) wird allerdings nur binär berücksichtigt, ob das einzelne Merkmal seinen Merkmalsschwellenwert $\Theta(\mu)$ erreicht oder nicht. Die Entscheidung einer einzelnen konventionellen Klassifikatorstufe wird dann durch

$$\mathcal{K}(s, t, g(\mathbf{x}), \tau) = \begin{cases} 1\,, & \sum\limits_{t} \alpha(t) h\,(s, t, \Theta) > \tau(s) \\ 0\,, & \text{sonst} \end{cases}\,, \tag{3.9}$$

bestimmt. Die Anzahl t der schwachen Klassifikatoren der Stufen wurden dabei beim Training so gewählt, dass die vorgegebene *HR* und *FAR* erreicht wurde. Hierbei ist nun wichtig zu beachten, dass die Gesamt-

summe aller $\alpha(t)$ deutlich größer sein kann als $\tau(s)$, da die Stufenentscheidung als Superposition der Antworten der schwachen Klassifikatoren bestimmt wird und ein Eingangsbild nicht notwendigerweise von allen schwachen Klassifikatoren korrekt klassifiziert werden muss. Diese Information soll im nachfolgenden Ansatz ausgenutzt werden. Die konventionelle Klassifikation für ein Eingangsbild $g(\mathbf{u})$ liefert dann

$$\mathcal{K}_{\mathrm{S}} = \begin{cases} 1\,, & \mathcal{K}(s) = 1, \forall s \in \{1, \ldots, \mathrm{S}\} \\ 0\,, & \text{sonst} \end{cases} \quad , \tag{3.10}$$

mit der Gesamtanzahl an Stufen S, welche die finale Falschalarmrate durch $FAR = FAR(s)^{\mathrm{S}}$ festlegt.

Es wird nun ein Ansatz vorgestellt, welcher den binären Ausgang aus Glg. (3.10) unter Ausnutzung jeder einzelnen auf einem Eingang berechneten Merkmalsantwort zu einem quasi-kontinuierlichen Klassifikatorausgang erweitert werden kann, um mit dem Isophotenansatz fusioniert zu werden.

3.5.2 Quasi-kontinuierliche Klassifikatorwerte

Unter quasi-kontinuierlichen Klassifikatorwerten werden hier diskrete Werte beliebig genauer Quantisierung im Bereich $[0, \ldots, 1]$ verstanden. Um einen kalibrierten Wertebereich zu erhalten, der eine sinnvolle Fusion mit dem in Kap. 3.4 beschriebenen Ansatz zur Kandidatensuche für Iriszentren zulässt, wird hierzu jeder einzelne schwache Klassifikator jeder Stufe unabhängig von der Gesamtentscheidung der Stufe behandelt. Dazu wird die Bedingung in Glg. (3.10) aufgehoben und ungeachtet des Erreichens des Stufenschwellenwertes über alle Merkmalsgewichte, für deren Merkmalsantwort der Schwellenwert erreicht wird, aufsummiert, sodass jedes einzelne Merkmal einer positiven Klassifikation beiträgt:

$$\hat{\mathcal{K}}_{\mathrm{S}} = \sum_{s} \sum_{t} \alpha(s,t) h\left(s,t,\Theta\right) \; . \tag{3.11}$$

Bei Vergleich von Glg. (3.9) und (3.10) mit Glg. (3.11) fallen zwei Unterschiede auf: Der erste Unterschied ist offensichtlich, da in Glg. (3.11) über alle Stufen über alle schwachen Klassifikatoren summiert wird. Weiterhin werden alle Gewichte $\alpha(s,t)$ der schwachen Klassifikatoren, für die $\mathbf{m}_{\mu} > \Theta(t)$ gilt, summiert, wobei missachtet wird, ob der Stufenschwel-

lenwert $\tau(s)$ der Stufe $\mathcal{K}(s,t)$, zu der das Merkmal gehört, erreicht wird. Der Ansatz impliziert, dass jeder einzelne schwache Klassifikator als unabhängiges, alleinstehendes Merkmal zur Klassifikation verstanden werden kann, dessen Antwort nützliche Information beinhaltet, auch im Falle einer nicht erfüllten Kaskadenstufe.

Es soll dennoch die Information, inwieweit ein Bild von den einzelnen starken Klassifikatoren korrekt klassifiziert wurde, berücksichtig werden. Dies wird erzielt, indem die Antworten nach Glg. (3.9) summiert und mit S normiert werden und damit

$$\mathcal{K}^{\Sigma} = \frac{1}{\mathrm{S}} \sum_{s} \mathcal{K}(s,t,g(\mathbf{u}),\tau) \tag{3.12}$$

gebildet wird. Nach Normalisieren von Glg. (3.11) liefert die Gewichtung mit $\mathcal{K}^{\Sigma}$ somit:

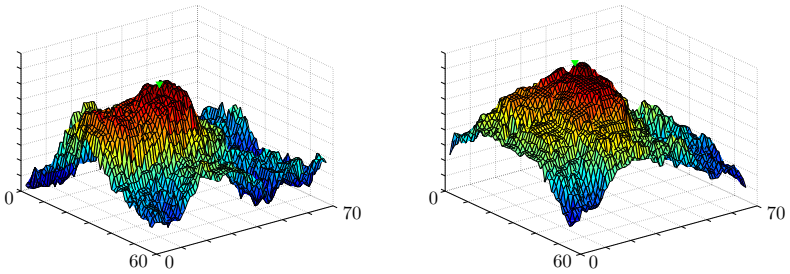
$$\mathcal{K}_{\mathrm{S}} = \frac{\hat{\mathcal{K}}_{\mathrm{S}} - \sum_{s} \tau_s}{\sum_{s} \sum_{t} \alpha(s,t) - \sum_{s} \tau_s} \cdot \mathcal{K}^{\Sigma}. \tag{3.13}$$

Der Wertebereich von $\mathcal{K}_{\mathrm{S}}$ ist nun quasi-kontinuierlich und durch die Normierung auf einen Wertebereich von null bis eins beschränkt. Es muss allerdings berücksichtigt werden, dass der funktionale Zusammenhang zwischen Eingang und normiertem Ausgang weiterhin unbekannt und durch die durch das *Boosting* festgelegten, beliebig großen Werte $\alpha(s,t)$ beschrieben wird und somit von allen Parametern, welche das Training einer Kaskade beschreiben, abhängig ist.

3.5.3 Fusion

Im beschriebenen Ansatz soll der Isophotenansatz genutzt werden, um Kandidaten für die Pupillenzentren zu finden. Die Werte der ZM sollen dann weiter mit Information, die aus dem erweiterten und angepassten Kaskadenklassifizierer gewonnen wird, kombiniert werden. Dabei sollen lokale Maxima, die näher am tatsächlichen Iriszentrum liegen gegenüber peripheren Maxima, stärker gewichtet werden. Für eine geeignete Informationsfusion, bei welcher fälschlicherweise als Pupillenzentren angenommene Votings aufgrund von ungewollten lokalen Maxima der ZM korrigiert werden, wird ein monotones Verhalten von Glg. (3.13)

bezüglich des tatsächlichen Irismittelpunktes bevorzugt. Abbildung 3.22 zeigt das Ergebnis der Anwendung von Glg. (3.13) für die Bilder BioID0767L (3.6(a)) und BioID0005L (3.6(d)).



(a) Nach Glg. (3.13) bestimmte Werte $\mathcal{K}_{\mathrm{S}}$ für jedes Pixel des Bildes BioID0767L.

(b) Nach Glg. (3.13) bestimmte Werte $\mathcal{K}_{\mathrm{S}}$ für jedes Pixel des Bildes BioID0005L.

Abbildung 3.22 Quasi-kontinuierliche Kaskadenwerte für BioID0005L (3.6(d)) und BioID0767L (3.6(a)).

Die Größe des Auswertefensters für die Kaskade wurde hier so gewählt, dass sie annähernd der Bildbreite entspricht; das bedeutet, dass bei einer Bildgröße von 56×70 Pixeln eine Fentergröße von 64×64 Pixeln gewählt wurde.

Ein monoton steigender Wert in Richtung des Pupillenzentrums ist beobachtbar in Abb. 3.22(b), während auch in Abb. 3.22(a) eine Tendenz höherer Werte in Richtung der Pupille zu erkennen ist. Bei der Berechnung der Rückgabe durch die Kaskade muss berücksichtigt werden, dass der gesamte Gesichtsauschnitt verwendet werden muss, da das Auswertefenster der Kaskade über den Augenausschnitt selbst hinaus geht. Es lässt sich daher für das mittlere Auge keine Auswertung mit Fusion der Kaskadeninformation ohne semantische Angleichung aller vollständigen Gesichter (z. B. durch ein *Active Appearance*-Modell [VIP16; VIP17]) durchführen.

Die finale Schätzung geschieht dann durch Multiplikation des Schätzwertes der ZM an den Stellen der lokalen Maxima mit den korrespondierenden Klassifikatorwerten. Dadurch, dass nur die signifikantesten Kandidaten berücksichtigt werden, wird der Klassifikator nur an wenigen Stellen ausgewertet, was den Ansatz recheneffizient sein lässt.

3.6 Auswertung

Um die vorgestellten Methoden zu validieren, werden verschiedene Datenbanken, welche eine große auftretende Intraklassenvarianz von Gesichtern und Augen abdecken sollen, um eine möglichst umfassende Auswertung zu gewährleisten, bei der Auswertung berücksichtigt.

Als Vorverarbeitungsschritt werden die OpenCV-Implementierung eines Gesichtsdetektors [Ope] sowie anthropometrische Werte [VG08] eingesetzt, um grob die Regionen der Augen zu finden. Dieser Schritt wurde gewählt, um die vorgestellten Ergebnisse vergleichbar mit denen anderer Forscher zumachen. Abweichungen bezüglich Datensatz, Auswertung, verwendete Methoden werden stets nach bestem Wissen über die Randbedingungen der Ergebnisse anderer Autoren angegeben.

3.6.1 Datenbanken und Gütemaß

Als Gütemaß wird, entsprechend zahlreichen Veröffentlichungen in der Literatur, das auf den Abstand der Augen normierte Fehlermaß nach Jesorsky et al. [JKF01] verwendet. Hierbei wird der minimale, mittlere und maximale Fehler der beiden Augen bestimmt. Der maximale Fehler ϵ_{MAX}, welcher, falls nicht anders bemerkt, im Weiteren stets als Vergleichsgröße für die Auswertungen herangezogen wird, bestimmt sich durch

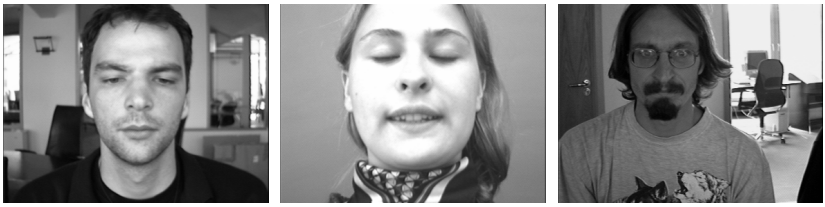
$$\epsilon_{\mathrm{MAX}} = \frac{\max(\mathbf{c}_l^{\mathrm{est}} - \mathbf{c}_l^{\mathrm{GT}}, \mathbf{c}_r^{\mathrm{est}} - \mathbf{c}_r^{\mathrm{GT}})}{\|\mathbf{c}_l^{\mathrm{GT}} - \mathbf{c}_r^{\mathrm{GT}}\|}, \tag{3.14}$$

wobei $\mathbf{c}_l^{\mathrm{est}}$ und $\mathbf{c}_r^{\mathrm{est}}$ die geschätzten linken und rechten Pupillenzentren sind und $\mathbf{c}_l^{\mathrm{GT}}, \mathbf{c}_r^{\mathrm{GT}}$ die *Ground Truth* angeben. Das vorgeschlagene Gütemaß lässt sich für die Größenordnungen $0{,}25$, $0{,}1$ und $0{,}05$ als die Distanzen zwischen den beiden Augenwinkeln bzw. die Durchmesser von Iris und Pupille interpretieren. Ein maximaler Fehler kleiner als $0{,}05$ entspricht also dem Ergebnis, dass im Schnitt aller ausgewerteten Augen das schlechtere Ergebnis der beiden Augen innerhalt der Pupille liegt, während das bessere deutlich genauer sein kann.

3.6.1.1 BioID

Die BioID-Datenbank [Bio01] besteht aus 1521 Bildern niedriger Auflösung und wurde von einer an einem Rechner platzierten Webcam unter alltäglichen Bürobedingungen aufgenommen. Sie zeigt unterschiedliche Individuen unter variierenden Beleuchtungsbedingungen und Posen, Abständen zum Sensor, mit Brillen sowie Okklusionen der Iris durch überlappende Augenlider sowie vollständig geschlossene Augen.

In dieser Auswertung wurden die Bilder vor der eigentlichen Irisdetektion vorverarbeitet, indem sie nach einer Gesichtsdetektion und dem Finden der groben Augenregion mittels antropometrischer Werte für jedes Auge auf eine Mindestbreite von 70 Pixel skaliert wurden, wobei der Wert empirisch so gefunden wurde, dass kein Beispielbild unter Informationsverlust mit einem Faktor kleiner eins skaliert wird. Damit wurde auch auf die stark variierende Bildgröße einzelner Augen, bedingt durch variierende Abstände zum Sensor, eingegangen. Drei Bilder aus der BioID-Datenbank sind beispielhaft in Abb. 3.23 gezeigt.



(a) BioID0000. **(b)** BioID0055. **(c)** BioID0355.

Abbildung 3.23 Die Bilder BioID0000, BioID0055 und BioID0355.

3.6.1.2 ColorFERET

Die ColorFERET-Datenbank [Jon+00] besteht aus insgesamt 11338 Bildern von 994 Personen, die in unterschiedlichen Gesichtsposen und einer Auflösung von 512×768 Pixeln aufgenommen wurde. Da für die Auswertung *Ground Truth*-Daten bekannt sein müssen und diese nur für die frontalen Bilder ($\ldots\_fa\_a.ppm$) vorhanden sind, geschieht die Auswertung auf 2662 Bildern. Die Bilder werden in dieser Arbeit abweichend von den Dateinamen mit ‚Feret' und einer angehängten, 4 Ziffern umfassenden, Zahl beschrieben. Um Ergebnisse vergleichbar zu halten

und die Generalisierbarkeit der Methoden zu unterstreichen, sind die Vorverarbeitungsschritte genau wie bei der BioID-Datenbank ausgeführt und auch für alle weiteren Auswertungen aus gleicher Motivation so beibehalten.

Drei mittels Detektor ausgeschnittene Gesichter aus der Datenbank sind in Abb. 3.24 gezeigt.

(a) Feret0001. **(b)** Feret0199. **(c)** Feret0239.

Abbildung 3.24 Ausschnitte der Bilder Feret0001, Feret0199 und Feret0239.

3.6.1.3 PUT

Die *Put Face Database* (PUT) [KFS08]) ist eine Gesichtsdatenbank mit hochaufgelösten Bildern. Sie besteht aus 9971 farbigen Bildern mit einer Bildgröße von 2048×1536 Pixeln. Im Gegensatz zur BioID- und ColorFERET Datenbank wurden dabei die 100 Testpersonen aus verschiedenen Kamerawinkeln fotografiert. Die Bilder wurden unter gleichbleibenden Beleuchtungsbedingungen vor einem einfarbigen Hintergrund aufgenommen. Die Augen der Personen sind dabei nicht immer geöffnet und die Koordinaten der Augen annotiert.

Von den 9971 Bildern werden mit dem *frontalface_alt* Gesichtsklassifikator 271 Gesichter nicht detektiert. Des Weiteren wurden Bilder, bei denen eine Annotation der Augen nicht möglich ist, aussortiert. Die Ursache liegt darin, dass die Testpersonen in einer zu starken Kopfneigung fotografiert wurden und so eines oder beide Augen im Bild nicht zu erkennen sind. Für den Fall, dass nur ein Auge eines Bildes annotierte Koordinaten besitzt, wurde das Bild komplett aus der Auswertung ausgeschlossen. Daraus ergeben sich dann 9094 auszuwertende Bilder.

Neben den Auswertungen der gesamten Datenbank wurde eine Auswahl an Bildern untersucht, die als frontale Gesichter annotiert sind. Diese Auswahl besteht aus 2188 Bildern. Dabei wurde in 65 Fällen kein Gesicht detektiert. Drei mittels Detektor ausgeschnittene Gesichter aus der Datenbank sind in Abb. 3.25 gezeigt.

(a) PUT 00011030. **(b)** PUT 00023018. **(c)** PUT 00041009.

Abbildung 3.25 Ausgeschnittene Gesichter aus der PUT-Datenbank.

3.6.2 Wahl des Tiefpassfilters

Zunächst soll der Einfluss der Breite der Gaußfilters zum Clustern der Votes und Unterdrücken von Rauschen und numerisch bedingten Artefakten in der ZM untersucht werden. Hierzu wurden die Eingangsbilder auf eine Referenzbreite von 70 Pixeln skaliert. Der Wert wurde anhand der am größten aufgelösten auftretenden Bildausschnitten von Augen in der BioID-Datenbank gewählt, sodass es ausschließlich zu Skalierungen größer 1 kommt, um keinen direkten Informationsverlust im Vorverarbeitungsschritt zu erhalten. Für alle weiteren Auswertungen wurden eine Breite von 70 Pixeln unter Inkaufnahme von Skalierungen kleiner 1 bei hochaufgelösten Daten übernommen. Die Einheitsgröße ist notwendig, um zum Erfüllen einer Generalisierbarkeit eine einzige Parametrierung für die vorgeschlagene Methode zu finden.

Abbildung 3.26 zeigt den Einfluss der Gaußfilterbreiten auf das Schätzergebnis der Pupillenzentren. Angegeben ist der maximale Fehler ϵ_{MAX}, ausgewertet für eine Bestimmung durch den Isophotenansatz ohne Kombination mit dem quasi-kontinuierlichen Kaskadenklassifizierer.

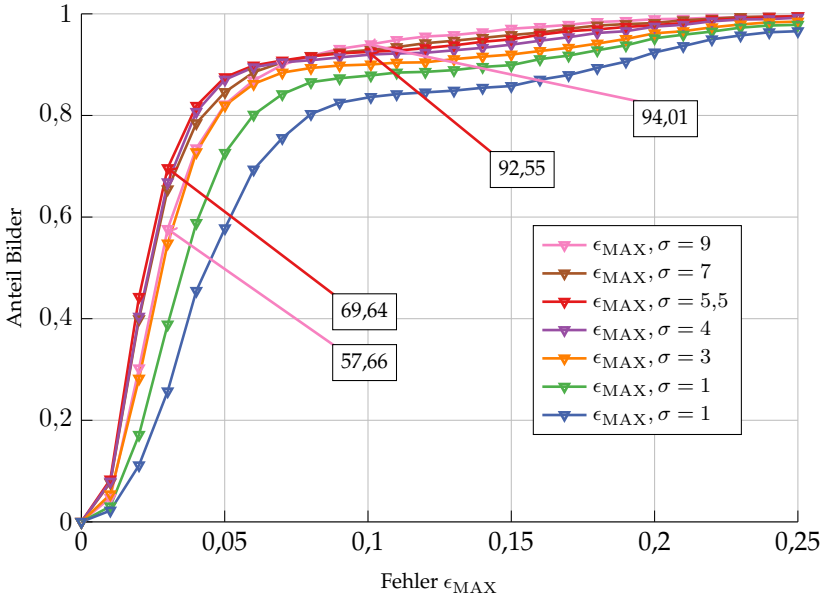


Abbildung 3.26 Fehler ϵ_{MAX}, indiziert durch die nach unten gerichteten, dreieckigen Marker, im Vergleich der verschiedener Breiten des Gaußfilters zum Clustern der ZM. Voting aus Isophoten allein, kombinierte Gewichtung aus Glg. (3.7). Skalenraum mit 4 Skalen pro Oktave erstellt.

Man erkennt deutlich den starken Einfluss des Clusterns auf das Ergebnis. Schmale Tiefpassfilter können das Rauschen nicht genügend unterdrücken. Das beste Ergebnis, inbesondere für eine präzise Irislokalisation, wird mit $\sigma = 5{,}5$ erreicht. Weiterhin ist zu erkennen, dass breite Filter in Bereichen eines größeren zugelassenen Fehlers bessere Ergebnisse liefern. Dies lässt sich damit erklären, dass immer mehr Daten zu einem Voting zusammengefasst werden, welches global näher an der *Ground Truth* liegt, während hochaufgelöste Information verloren geht.

3.6.3 Einfluss heller Zentren und Gewichtungen

Der Einfluss der Hinzunahme von hellen Zentren sowie die Abhängigkeit von der jeweiligen Gewichtung sind in Abb. 3.27 illustriert.

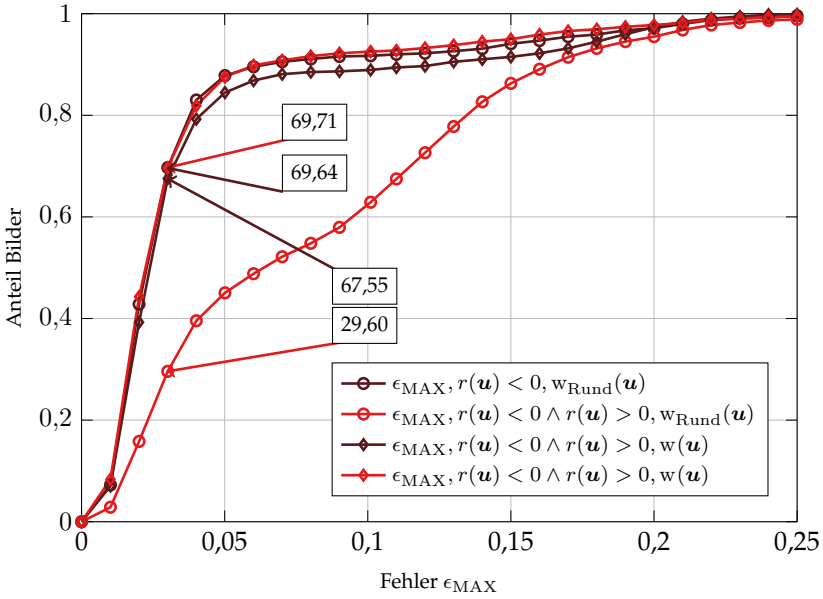


Abbildung 3.27 Fehler ϵ_{MAX} im Vergleich der ZM bei Hinzunahme der Votings aus hellen Zentren; Voting aus Isophoten allein. Die kreisförmigen Marker indizieren Ergebnisse mit einer Gewichtung durch die Rundheit allein, während die rautenförmigen Marker eine kombinierte Gewichtung anzeigen. Skalenraum mit 4 Skalen pro Oktave erstellt, $\sigma = 5{,}5$.

In roter Farbe sind die Ergebnisse für eine Bildung der ZM mit hellen und dunklen Zentren (positiver und negativer Betrag des Radiusses), und in dunkelroter Farbe die Ergebnisse nur aus dunklen Zentren resultierend skizziert. Kreise zeigen Ergbnisse mit nur der Rundheit als Gewichtung, Rauten Ergebnisse mit der kombinierten Gewichtung.

3.6.3.1 Helle Zentren

Die Hinzunahme der Information aus hellen Zentren zeigt bei einer Gewichtung ausschließlich durch die Rundheit ein starke Verschlech-

terung der Ergebnisse. Dies lässt sich durch hohe Werte der Rundheit im Bereich der Sklera in der Nähe der Augenwinkel erklären. Bei dieser Gewichtung ist eine Verwendung von Verschiebungsvektoren mit ausschließlich negativem Radius sinnvoll [VG12]. Bei Erweiterung des Gewichtungsschemas auf Start- und Endpunkt des Verschiebungsvektors und entsprechender Akkumulation lässt sich hingegen eine Verbesserung von über $2\,\%$ bei einem $\epsilon_{\mathrm{MAX}} = 0{,}03$ und fast $4\,\%$ bei einem $\epsilon_{\mathrm{MAX}} = 0{,}10$ erreichen.

3.6.3.2 Kombinierte Gewichtung

Die Kompensation des großen Verlustes an Genauigkeit durch Mehrinformation durch helle Zentren für die kombinierte Gewichtung lässt sich damit erklären, dass gerade innerhalb der Sklera $\mathrm{w}_{\mathrm{Pup}}(\mathbf{u})$ sehr kleine Werte hat und so Verschiebungsvektoren, die aufgrund von $\mathrm{w}_{\mathrm{Rund}}(\mathbf{u})$ einen großen Beitrag liefern, abgeschwächt werden. Durch die kombinierte Gewichtung und die Berücksichtigung von $r(\mathbf{u}) < 0 \wedge r(\mathbf{u}) > 0$ wird somit das beste Ergebnis erzielt.

Es soll nun untersucht werden, wie die Fusion mit Informationen aus dem Kaskadenklassifikator das Ergebnis beeinflusst.

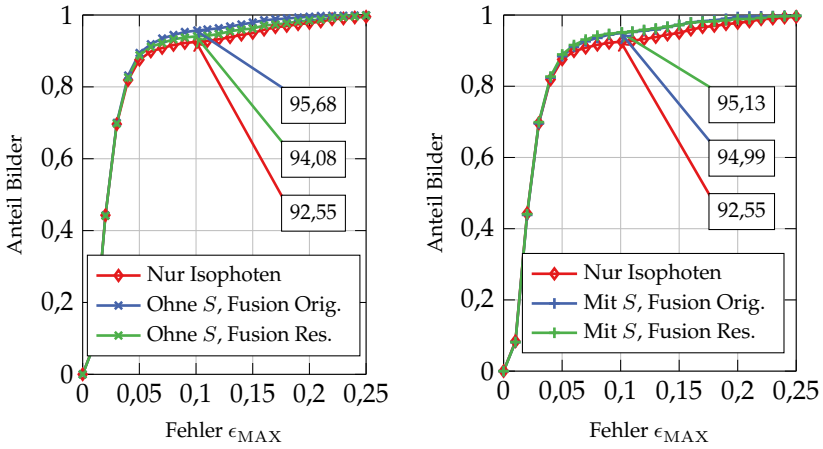
3.6.4 Fusion mit dem Kaskadenklassifikator

Bei der Fusion der Informationen muss der Wertebereich der einzelnen Methoden beachtet werden. Da die ZM trotz Skalierung auf unterschiedlich großen Bildern (Breite konstant, Höhe variiert) berechnet wird und weiterhin noch von den lokalen Werten der Rundheit sowie den Grauwerten abhängt, zeigt sich eine Normierung auf einen festen Wertebereich schwierig.
Der quasi-kontinuierlich Kaskadenklassifikator liefert nach Implementierung nach Glg. (3.13) einen begrenzten Ausgang, wobei der Wertebereich über $[0, \ldots, 1]$ hinausgeht, da in Glg. (3.13)

$$\hat{\mathcal{K}}_{\mathrm{S}} > \sum_{s} \tau_s \tag{3.15}$$

gelten kann. Der Zusammenhang mit dem Eingang ist weiterhin unbekannt (beispielsweise nicht notwendigerweise linear). In einem ersten

Test wurde zunächst das Ergebnis des Isophotenansatzes mit dem Ergebnis des quasi-kontinuierlichen Kaskadenklassifizieres naiv (unkalibrierte Kaskade) kombiniert. Dabei wurde die Information für die Fusion einmal im skalierten Bild sowie einmal im Originalbild gewonnen, wobei die Kaskadenbreite mit 70 Pixeln bzw. mit der Breite des Auges im Originalbild gewählt wurde. Die Ergebnisse mit dem unveränderten quasi-kontinuierlichen Kaskadenausgang sind in Abb. 3.28(a) zu sehen.



(a) Fehler ϵ_{MAX} für Ergebnisse der Irislokalisation durch Fusion mit einer unkalibrierten Kaskade (Kreuze).

(b) Fehler ϵ_{MAX} für Ergebnisse der Irislokalisation durch Fusion mit einer kalibrierten Kaskade (Striche).

Abbildung 3.28 Einfluss auf das Ergebnis der Irislokalisation durch Fusion mit Information aus dem Kaskadenklassifizierer mit $\mathcal{K}_{\mathrm{Bagg}}$ und $S\,(a, b, c, \mathcal{K}_{\mathrm{S}})$ bestimmt auf BioID. Die Abkürzungen Ori. und Skal. unterscheiden die Ermittlung der Kaskadenklassifikatorwerte im Originalbild und im skalierten Bild.

Es wurde die Kaskade $\mathcal{K}_{\mathrm{Bagg}}$ verwendet und der Skalenraum mit 4 Skalen pro Oktave erstellt.

Zu erkennen ist, dass die Fusion keinen Einfluss auf den Bereich des präzisen Schätzergebnisses hat. Ab einem Fehler von etwa $\epsilon_{\mathrm{MAX}} = 0{,}04$ divergieren die unterschiedlich erzielten Ergebnisse und die Fusion zeigt sich synergetisch sowohl für die Bestimmung der Kaskadenklassifikatorwerte im Originalbild sowie im skalierten Bild. Das bessere Abschneiden der Ergebnisse im unskalierten Bild lässt sich mit der nicht perfekten

Invarianz gegenüber Skalierung des Klassifikators erklären, welcher auf einer Bildgröße von 36×36 Pixeln trainiert wurde und damit näher an der mittleren, originalen Augenbreite als an der fest skalierten Breite liegt.

3.6.5 Kalibrierung der Kaskade

Es soll nun der Einfluss einer vor der Fusion durchgeführten Abbildung des Kaskadenausgangs auf einen Wertebereich $[0, \ldots, 1]$ mit einer geeigneten Funktion überprüft werden. Dabei wird der Ansatz aus Platt [Pla99] adaptiert, in dem für SVM gezeigt wird, wie der Wertebereich des Klassifikators zu einem probabilistischen Ausgang manipuliert werden kann. Der Autor findet hierfür eine Sigmoid-Funktion als beste Lösung, deren Parameter anhand eines Trainingsdatensatzes sowie der Klassifikatorausgänge mittels kleinster Fehlerquadrate gelernt werden, sodass der Ausgang der Sigmoid-Funktion als die Wahrscheinlichkeit, dass es sich (bezogen auf den Trainingsdatensatz) um ein Objekt der positiven Trainingsklasse handelt, interpretiert werden kann.

Der Ansatz wird hier neuartig auf Kaskadenklassifikatoren angewandt, wobei die zuvor beschriebene Erweiterung des Kaskadenklassifikators auf einen quasi-kontinuerlichen Ausgang hierzu gerade genutzt werden kann. Es werden alle Antworten des Kaskadenklassifikators $\mathcal{K}_\mathrm{S}$ für eine Menge $\mathcal{I}$ Trainingsdaten gespeichert und anschließend die Parameter a, b, c einer Funktion

$$S\left(a, b, c, \mathcal{K}_\mathrm{S}\right) = \frac{1}{\mathrm{e}^{a\mathcal{K}_\mathrm{S}^2 + b\mathcal{K}_\mathrm{S} + c}} \tag{3.16}$$

mit Hilfe von auf einem Trainingsdatensatz bestimmten Kaskadenwerten $\mathcal{K}_\mathrm{S}$ gelernt, die dann später bei der eigentlichen Irislokalisation auf unbekannte Daten angewandt wird. Die Daten $\mathcal{K}_\mathrm{S}$ werden hierbei so bestimmt, dass die Kaskade an genau dem annotierten Iriszentrum der Trainingsdaten ausgewertet wird. Für die am IIIT von Hand zentrierten Augenbilder der verwendeten Datensätze entspricht dies gerade dem zentralen Pixel

$$\mathbf{u}_\mathrm{c} = \left(\left\lfloor \frac{h_0}{2} \right\rfloor + 1, \left\lfloor \frac{b_0}{2} \right\rfloor + 1 \right)^\mathrm{T}. \tag{3.17}$$

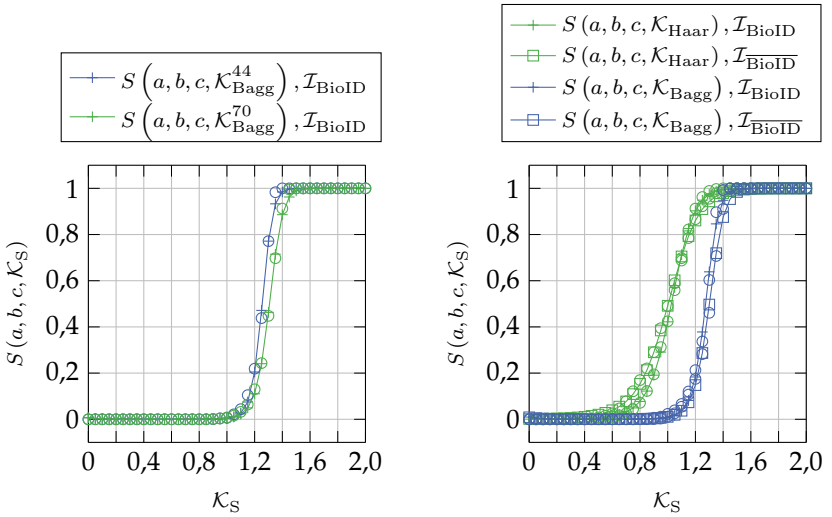
Folgende Abhängigkeiten sind hierbei zu beachten:

- Die Funktion $S\left(a,b,c,\mathcal{K}_{\mathrm{S}}\right)$ ist abhängig vom Trainingsdatensatz. Eine Überanpassung an den Testdatensatz sollte zur Gewährleistung der Generalisierungsfähigkeit durch Disjunktheit der Testdatensätze vom Trainingsdatensatz geprüft bzw. vermieden werden.
- Aufgrund der nicht perfekten Skalierungsinvarianz der Kaskaden (u. a. durch LBP -Merkmale) ist $S\left(a,b,c,\mathcal{K}_{\mathrm{S}}\right)$ abhängig von der gewählten Detektionsfenstergröße $h^{\mathrm{Det}}, b^{\mathrm{Det}}$ des Kaskadenklassifikators. Hierbei sind weiterhin Effekte, die sich aus der damit zusammenhängenden Skalierung der Trainingsdaten ergeben, zu beachten.
- Die bestimmte Funktion kann nur für die Kaskade, für die sie gelernt wurde, sinnvoll angewandt werden.

Abbildung 3.28(b) zeigt die mit der Kaskade $\mathcal{K}_{\mathrm{Bagg}}$ und dem Datensatz $\mathcal{I}_{\mathrm{BioID}}$ kalibrierten Ergebnisse bei ansonsten unveränderten Randbedingungen, wobei die Abbildung des Kaskadenausgangs auf den Wertebereich $[0,\ldots,1]$ anhand der in Abb. 3.29(a) dargestellten Funktionen geschieht. Für diesen Fall ist die Disjunktheit zwischen Trainingsdatensatz der Sigmoidfunktion und Testdaten nicht berücksichtigt, um das Potential der Methode auszuloten.

Die Ergebnisse mit Hilfe des Abbildens des Klassifikatorausgangs auf einen definierten Wertebereich in Abb. 3.28(b) zeigen Folgendes: Während trotz zuvor gesehener Daten (S mit $\mathcal{I}_{\mathrm{BioID}}$ trainiert) das Ergebnis in Abb. 3.28(a) ohne Anwendung von Glg. (3.16) für den skalierten Fall kaum verbessert wird, erhöht die Fusion nach Berücksichtigen des Zusammenhangs zwischen Klassifikatorausgang und Wahrscheinlichkeit der Detektion das Ergebnis der Lokalisierung um mehr als $1\,\%$ bei einem Fehler $\epsilon_{\mathrm{MAX}} = 0{,}1$. Für den unskalierten Fall wurde die Detektion wieder mit einem Fenster, welches in etwa der Augenbreite entspricht, durchgeführt und durch verschiedene Funktionen S, welche für $h^{\mathrm{Det}} = \{36,40,44,48,53,64,70\}$ Pixel bestimmt wurden, ermittelt. Das schlechtere Abschneiden insgesamt für die Auswertung im skalierten Bild wurde bereits oben diskutiert. Hierbei ist zu beachten, dass insbesondere bei Anwendung auf Daten, deren mittlere Augengröße

von dem des Trainingsdatensatzes abweicht, der Fall der Auswertung mit S generalisierender ist und insbesondere hierfür wie gewünscht eine Erhöhung der Genauigkeit erreicht werden kann.



(a) $S\left(a,b,c,\mathcal{K}_{\mathrm{S}}\right)$ für $\mathcal{K}_{\mathrm{Haar}}$ sowie $\mathcal{I}_{\mathrm{BioID}}$ für die Breiten $b^{\mathrm{Det}} = 44$ Pixel und $b^{\mathrm{Det}} = 70$ Pixel.

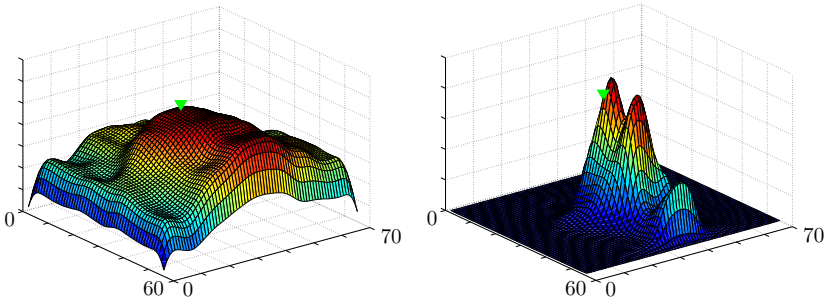
(b) $S\left(a,b,c,\mathcal{K}_{\mathrm{S}}\right)$ für $\mathcal{K}_{\mathrm{Haar}}$ und $\mathcal{K}_{\mathrm{Bagg}}$ sowie $\mathcal{I}_{\mathrm{BioID}}$ und $\mathcal{I}_{\overline{\mathrm{BioID}}}$. Die Breite beträgt jeweils 70 Pixel.

Abbildung 3.29 Funktionen $S\left(a,b,c,\mathcal{K}_{\mathrm{S}}\right)$ für das Abbilden der Ausgänge des quasi-kontinuierlichen Kaskadenklassifizierers. Durch die Kreise sind einige Datenpunkte der Kaskadenausgänge $\mathcal{K}_{\mathrm{S}}$ skizziert.

3.6.5.1 Einfluss der Kaskade und Trainingsdaten auf S

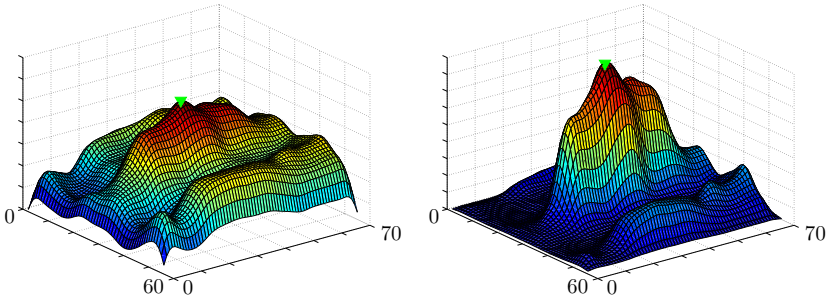
Die Parameter a, b, c hängen zum einen von $\mathcal{K}_{\mathrm{S}}$ und damit von der Kaskade selbst, zum anderen von dem Datensatz, für den die Werte des Kaskadenklassifikators bestimmt werden, ab. Der Einfluss auf die quasi-kontinuierlichen Kaskadenwerte durch Abbildung des Ausgangs auf eine Sigmoidfunktion in Abhängigkeit der verwendeten Kaskade sowie des Trainingsdatensatzes zur Bestimmung von S ist in Abb. 3.29 dargestellt, während die Abbildungen 3.30 und 3.31 den Zusammenhang

anhand des Beispielbildes BioID0005R illustrieren. Das grüne Dreieck zeigt die tatsächliche Irisposition.



(a) Werte $\mathcal{K}_{\mathrm{Bagg}}$ ohne Kalibrierung. **(b)** Werte $\mathcal{K}_{\mathrm{Bagg}}$ nach Kalibrierung.

Abbildung 3.30 Einfluss der Kalibrierung $S\,(a, b, c, \mathcal{K}_{\mathrm{S}})$ auf den Kaskadenausgang für das Bild BioID0005R, ausgewertet mit der Kaskade $\mathcal{K}_{\mathrm{Bagg}}$ und $\mathcal{I}_{\mathrm{BioID}}$.



(a) Werte $\mathcal{K}_{\mathrm{Haar}}$ ohne Kalibrierung. **(b)** Werte $\mathcal{K}_{\mathrm{Haar}}$ nach Kalibrierung.

Abbildung 3.31 Einfluss der Kalibrierung $S\,(a, b, c, \mathcal{K}_{\mathrm{S}})$ auf den Kaskadenausgang für das Bild BioID0005R, ausgewertet mit der Kaskade $\mathcal{K}_{Haar}$ und $\mathcal{I}_{\mathrm{BioID}}$.

In Abb. 3.29(a) erkennt man die Abhängigkeit von der gewählten Größe für das Kaskadenfenster. Aufgrund der hohen Steigung der Kurven im Bereich $\mathcal{K}_{\mathrm{Bagg}} = 1{,}2$ betragen die Unterschiede im Ausgang mehr als $0{,}5$. Das bedeutet, dass bei der Verwendung derselben Kaskaden in der Auswertung für denselben Punkt im Bild sich die Rückgabewerte um einen Faktor 2 unterscheiden können wie im hier gezeigten Beispiel. Wenn alle Werte $\mathcal{K}_{\mathrm{S}}$ in den linearen Bereich abgebildet werden,

so spielt dies bei rein multiplikativer Verknüpfung keine Rolle, da alle ausgewerteten Punkte um denselben Faktor manipuliert werden. Da dies allerdings nicht angenommen werden kann, ist eine Abhängigkeit der Fenstergröße zu berücksichtigen.

Der Abb. 3.29(b) kann man entnehmen, dass bei gegebener Kaskade der Trainingdatensatz für S ebenfalls eine Rolle spielt. Dies zeigt sich für beide ausgewerteten Kaskaden und ist insofern interessant, dass beispielsweise für $\mathcal{K}_{\mathrm{Haar}}$ auf einem unbekannten Datensatz ($\mathcal{I}_{\mathrm{BioID}}$) eine ähnliche Funktion S resultiert im Vergleich zur Funktion S, welche auf dem der Kaskade bekannten Datensatz bestimmt wird. Für $\mathcal{K}_{\mathrm{Bagg}}$ und ab etwa dem Wendepunkt der Sigmoidfunktion für $\mathcal{K}_{\mathrm{Haar}}$ erkennt man für die Anwendung auf dem BioID-Datensatz, dass niedrige Kaskadenwerte bereits zu einer höheren Detektionswahrscheinlichkeit führen; der Ausgang der Kaskaden hat für BioID niedrigere Werte. Dies spricht für die hohe Intraklassenvarianz der Datenbank und zeigt, wie anspruchvoll der Datensatz ist. Weiterhin fällt auf, dass die Kurve für $\mathcal{K}_{\mathrm{Haar}}$ weniger steil ist. Dies lässt auf eine weniger scharfe Klassikationsschwelle und damit schlechtere Performanz schließen, was die Ergebnisse weiter unten (Abb. 3.32) bestätigen. Für die Fusion ist eine scharfe Klassikationsschwelle zu wünschen, sodass sinnvoll auch unter eng beeinanderliegenden Werten für Kandidaten des Irismittelpunktes differenziert werden kann.

Abschließend lassen sich für die Fusion folgende Gedanken zusammenfassen: Die gleiche Kaskade liefert auf einem Bild stets die gleichen Rückgabewerte. Abhängig von S wird dieser Wert auf einen Ausgang abgebildet, der zwar vom Training von S abhängt, aber für alle in die Auswertung einbezogenen Bilder gleich ist. Somit sollte eine Verschiebung nach links oder rechts von S keinen Einfluss auf das Endergebnis für die Werte, welche in den linearen Bereich von S abgebildet werden, haben. Es lässt sich weiterhin schlussfolgern, dass der Trainingsdatensatz für S dann insbesondere eine Rolle spielt, wenn er die Form von S verändert. Die Form (Steigung im linearen Bereich) ist auschlaggebend für die Performanz des Ergebnisses.

Die Abbildungen 3.30 und 3.31 veranschaulichen die diskutierten Abhängigkeiten. Sehr gut ist die Abhängigkeit der Werte des Ausgangs des quasi-kontinuierlichen Kaskadenklassifikators durch Abbilden mit

der vorgestellten Methode zu erkennen. Besonders auffällig ist die stark unterschiedliche Performanz der Kaskaden, deren Ungenauigkeiten Streuungen in vertikaler bzw. orthogonal dazu zeigen, wobei deren Maximum stets im tatsächlichen Irismittelpunkt liegt. Dieser Zusammenhang wurde während des Erarbeitens der Ergebnisse mehrfach überprüft und bestätigt.

3.6.5.2 Einfluss unterschiedlicher Kaskaden und Kalibrierdaten

Der Zusammenhang soll nun global an Hand von Auswertungen auf Bildern der BioID-Datenbank geschehen, wobei zur Kalibrierung und zum Training der Kaskaden die Daten $\mathcal{I}_{\mathrm{BioID}}$ und $\mathcal{I}_{\overline{\mathrm{BioID}}}$ herangezogen wurden, die einmal ausschließlich aus den BioID-Bildern und einmal aus Bildern exklusive jener zusammengesetzt sind. Abbildung 3.32 zeigt Ergebnisse der Kaskaden $\mathcal{K}_{\mathrm{Bagg}}$ (trainiert mit einer Teilmenge aus $\mathcal{I}_{\mathrm{BioID}}$) und $\mathcal{K}_{\mathrm{Haar}}$ (trainiert mit $\mathcal{I}_{\overline{\mathrm{BioID}}}$), wobei jeweils mit den Trainingsdaten $\mathcal{I}_{\mathrm{BioID}}$ und $\mathcal{I}_{\overline{\mathrm{BioID}}}$ kalibriert und das Schätzergebnis ermittelt wurde. Die Abbildung soll den Einfluss von der gewählten Kaskade und dem gewählten Trainingsdaten zur Bestimmung von S in Abhängigkeit der Performanz der Kaskade allein illustrieren. Die Kurven „Nur Kaskade“ geben die Ergebnisse der Irislokalisierung an, wenn zwar als Kandidaten weiterhin die Schätzung der lokalen Maxima der ZM dient, unter diesen jedoch die Kaskade allein die endgültige Entscheidung trifft. Da die Ergebnisse „Nur Kaskade“ keine Kombination mit den Werten der ZM vorsehen, sind sie unabhängig von der Kalibrierung, was die Anzahl von lediglich zwei dieser Ergebniskurven erklärt. Zur weiteren Performanz der Kaskaden $\mathcal{K}_{\mathrm{Bagg}}$ und $\mathcal{K}_{\mathrm{Haar}}$ sei auf Abb. 2.17 in Kap. 2 verwiesen.

Folgendes ist der Abbildung zu entnehmen:

- Der Kalibrierdatensatz für S spielt für das Ergebnis der Irislokalisation eine untergeordnete Rolle; sogar das Verwenden des BioID-Datensatzes als Trainingsdatensatz für S ändert für beide ausgewertete Kaskaden wenig an den Ergebnissen, wobei festzuhalten ist, dass bessere Ergebnisse erzielt werden, wenn eine verwendete Kaskade auf seinem eigenen Trainingsdatensatz trainiert wird.

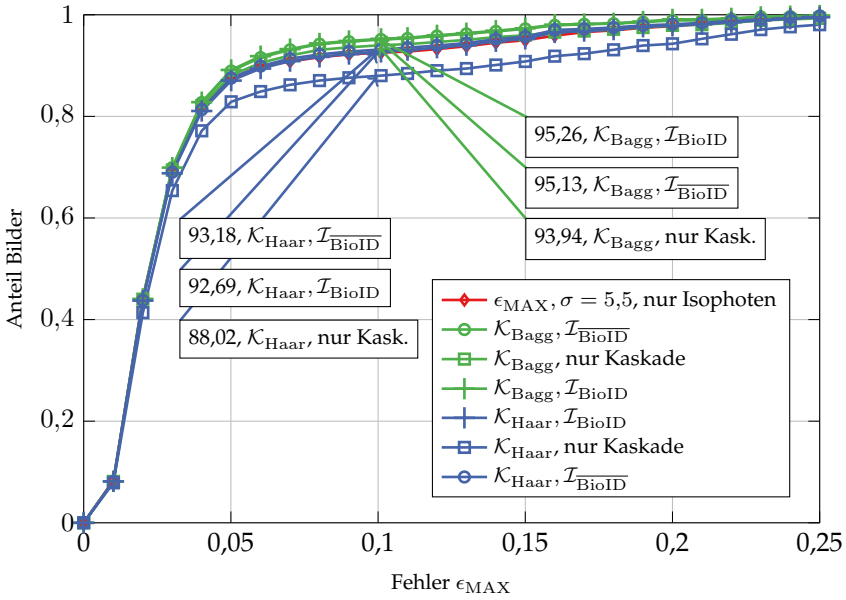


Abbildung 3.32 Fehler ϵ_{MAX} bei Verwendung unterschiedlicher Kaskaden sowie unterschiedlicher Trainingsdaten zur Bestimmung von S.

- Die Fusion mit einer Kaskade, die selbst schlechtere Ergebnisse liefert als die Methode durch Isophoten allein, verschlechtert das Ergebnis nicht (siehe $\mathcal{K}_{\mathrm{Haar}}$).
- Eine Kaskade hingegen, deren Ergebnis allein bereits über dem Isophotenergebnis liegt ($\epsilon_{\mathrm{MAX}} = 92{,}55$ für die rote Kurve bei $0{,}1$), kann durch Fusion das Ergebnis über ihre eigene Genauigkeit hinaus verbessern. Hierbei muss berücksichtigt werden, dass die Ergebnisse „Nur Kaskade“ auf den lokalen Maxima der ZM beruhen; sie geben nicht die Präzision einer globalen Augen*detektion* mittels Kaskade allein wieder.

Daraus lässt sich schlussfolgern, dass die vorgeschlagene Fusion

- robust ist, da sie bei Fusion mit einem schwachen Klassifikator kein verschlechtertes Ergebnis liefert und darüber hinaus

- die Güte der Lokalisierung weiter über die Performanz der Kaskade selbst verbessert.

3.6.6 Quantitative Auswertung und Vergleich mit dem Stand der Technik

Abschließend soll eine globale Übersicht und Diskussion der Ergebnisse im Kontext des Standes der Technik gegeben werden. Tabelle 3.2 vergleicht das hier erzielte Ergebnis mit anderen Arbeiten aus der Literatur quantitativ, wobei eine Aufteilung entsprechend Tabelle 3.1 nach modellbasierten (oben) und merkmalsbasierten (unten) Ansätzen fortgeführt wird.

Die vorgeschlagene Methode zeigt insbesondere für den sehr präzisen Bereich $\epsilon_{\mathrm{MAX}} \leq 0{,}02$, welcher einem Fehlerradius innerhalb der Pupille entspricht, eine höhere Genauigkeit auf dem BioID-Datensatz als die verglichenen Methoden. Für die merkmalsbasierten Methoden zeigen sich im Fehlerbereich bis einschließlich $\epsilon_{\mathrm{MAX}} \leq 0{,}05$ präzisere Ergebnisse als in der hier gegenübergestellten Literatur. Für den Bereich bis $\epsilon_{\mathrm{MAX}} \leq 0{,}1$ zeigt der Ansatz von Pang et al. [Pan+15] eine höhere Genauigkeit. Sie verwenden einen hybriden Ansatz, bei dem sie Isophoten mit einem Regressionsansatz kombinieren, wobei sie die Bilder des BioID-Datensatzes zum Training benutzen, wodurch eine strikte Trennung von Trainings- und Testdatensatz nicht mehr gegeben ist. Bei den trainingsbasierten Ansätzen, welche auf Modellen beruhen und damit immer der Herausforderung der Generalisierungsfähigkeit gegenüberstehen, zeigen die Ansätze von Kroon et al. [Kro+09] und Markuš et al. [Mar+14] im hohen erlaubten Fehlerbereich ab $\epsilon_{\mathrm{MAX}} \geq 0{,}05$ bzw. $\epsilon_{\mathrm{MAX}} > 0{,}05$ genauere Ergebnisse auf dem BioID Datensatz als die hier vorgestellte Methode. Dabei ist bei Kroon et al. [Kro+09] zu beachten, dass ihr Modell sehr parameterabhängig ist und auf den BioID-Datensatz angepasst wurde; beispielsweise verwenden sie a priori ermittelte mittlere Augenabstände des BioID-Datensatzes für die Parametrierung, welche sie für weitere Auswertungen beibehalten, womit der Ansatz nicht fähig für den Online-Einsatz ist. Sie geben weiterhin Ergebnisse für die FERET-Datenbank an, welche mit den hier erzielten Ergebnissen in Tabelle 3.4 verglichen werden. Markuš et al. [Mar+14] führen für die Schätzung des

Irismittelpunktes Perturbationen bezüglich des Ortes und der Größe des gewählten Augenausschnittes, welcher zur Lokalisierung herangezogen wird, durch. Sie finden dann das finale Ergebnis, indem sie den Median aller Ergebnisse der Schätzungen mittels Regressionsbäumen als Endergebnis wählen, um mit dem stark verrauschten Ausgang der Regressionsbäume umzugehen. Das beste veröffentlichte Ergebnis erhalten sie mit $p = 31$ Pertubationen, während die Genauigkeit ihres Ansatzes für $p = 7$ Pertubationen für $\epsilon_{\mathrm{MAX}} \leq 0{,}05$ bei $85{,}7$ liegt.

Abbildung 3.33 veranschaulicht bildlich die Verteilung von ϵ_{MAX} der in dieser Arbeit erzielten Ergebnisse, dargestellt über dem gemittelten Auge BioIDmean. Bei der Darstellung wurde angenommen, dass der Irisdurchmesser einem Fehlerradius $\epsilon_{\mathrm{MAX}} \leq 0{,}1$ entspricht. Es ist gut zu erkennen, dass die Genauigkeit der Methode, entsprechend dem Maß ϵ_{MAX}, nahezu aller Schätzungen auf dem BioID-Datensatz innerhalb der Iris liegen.

3.6.7 Performanz auf hochaufgelösten Bildern

Um den Einfluss der Qualität der auszuwertenden Bilder zu evaluieren, wurden weiterhin Experimente auf dem PUT- sowie dem FERET-Datensatz durchgeführt. Tabelle 3.3 zeigt die Ergebnisse auf der hochaufgelösten PUT Datenbank. Gut zu erkennen ist, dass das Ergebnis der Methode auf qualitativ besseren Bildern steigt. Beispielsweise liegt der Anteil ausgewerteter Bilder des PUT-Datensatzes, für die ein maximaler Fehler $\epsilon_{\mathrm{MAX}} \leq 0{,}05$ gilt, mit $98{,}54\,\%$ fast $10\,\%$ über dem auf dem BioID-Datensatz erzielten Ergebnis. Ohne näher auf die genaue Zusammensetzung der Datenbank einzugehen (siehe Abb. 3.25), lässt sich weiterhin erkennen, dass die Methode auch unter durch Gieren und Neigen verdrehten Kopfpositionen zuverlässige Ergebnisse liefert.
Ein weiterer ausgewerteter Datensatz ist FERET. Die erzielten Ergebnisse der vorgestellten Methode sowie einige Ergebnisse aus der Literatur sind Tabelle 3.4 zu entnehmen. Man erkennt gut die Fähigkeit der vorgestellten sowie der ebenfalls merkmalsbasierten Methode von Valenti und Gevers [VG12] zu generalisieren, wohingegen die trainingsbasierte Methode von Kroon et al. [Kro+09] schlechter, vor allem relativ im Vergleich zu den Ergebnisse auf der BioID-Datenbank, abschneidet.

Tabelle 3.2 BioID. Maximaler normalisierter Fehler ϵ_{MAX} nach Glg. (3.14) in Prozent. Die Superskripte bedeuten:
( )*: dargelegte Werte sind aus einem Graphen abgelesen;
( )^{a}: 1430 Bilder verwendet, 451 Test, 979 Training (*frontalface_alt*);
( )^{v1}: 1462 Bilder evaluiert (*frontalface_alt*);
( )^{v2}: 1457 Bilder evaluiert (*frontalface_default*);
( )^{k}: Kombiniert mit Kaskade ($\mathcal{K}_{\mathrm{Bagg}}$);
( )^{g}: keine Angabe (*frontalface_default*);
( )^{m}: Mapping des Kaskadenausgangs [Pla99], $\mathcal{I}_{\mathrm{BioID}}$, siehe Kap. 3.6.5.

Autor	BioID 0,02	BioID 0,03	BioID 0,05	BioID 0,10
Jesorsky [JKF01]			40,00*	79,00*
Cristinacce [CCS04]			56,00*	96,00
Hamouz [Ham+05]	20,00*		59,00*	77,00*
Niu [Zhi+06]			78,00*	93,00
Campadelli [CLL09]	30,00*		80,70	93,20
Kim [Kim+07]				96,40
Yang [Yan+11]			89,60	95,50
Dong [DZL11]	28,00*	60,00*	89,00*	99,01
Kroon [Kro+09]^{v2}	39,00		92,30	97,90
Rusek [RG14]^{a}	10,00*		60,00*	98,00
Markuš [Mar+14]	43,00*		89,90	97,10
Asadifard [AS10]	15,00*		47,00	86,00
Türkan [TPC07]			22,00*	73,68
Bai [BSW06]	10,00*		37,00*	64,00*
Timm [TB11]	22,00*		82,50	94,40
Pang [Pan+15]	40,00*		83,65	96,12
Valenti [VG12]^{g}	44,00*	66,00*	86,09	91,67
Vater Isophoten^{v1}	45,33	71,45	87,95	91,17
Vater Kaskade$^{v1\ k}$	45,47	71,66	89,35	94,92
Vater Kombiniert$^{v1\ km}$	**45,13**	**72,14**	**89,97**	**95,68**

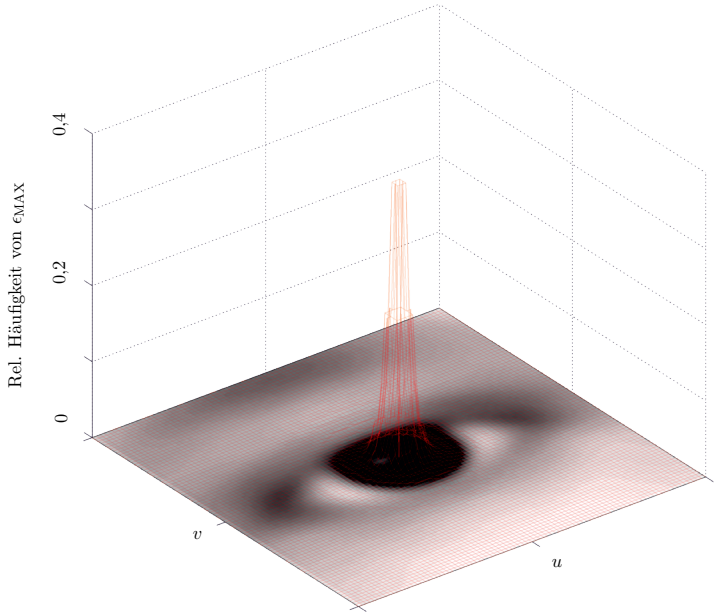


Abbildung 3.33 Visualisierung der relativen Häufigkeit des Fehlers ϵ_{MAX} entsprechend der anthropometrischen Analogie der Fehlergrenze, dargestellt über dem gemittelten Auge BioIDmean 3.3(a).

3.7 Zusammenfassung

In diesem Kapitel wurde eine Methode zur präzisen Irislokalisation vorgestellt. Aufbauend auf Arbeiten von Lichtenauer [LHR05] und Valenti [VG08] wurde dabei der Isophotenansatz adaptiert und um folgende Ideen ergänzt und erweitert:

- Es wurde ein neuartiges Gewichtungsschema vorgestellt, welches die auf den Isophoten liegenden intrinsischen Radien sowohl am Startpunkt als auch am Endpunkt gewichtet. Dabei wurde gezeigt, wie durch Berücksichtigung von betragsmäßig sowohl positiver als auch negativer Radien auch helle Zentren, welche bei Reflexionen innerhalb der Iris entstehen, positiv zum Lokalisierungsergebnis beitragen können.

Tabelle 3.3 PUT. Maximaler normalisierter Fehler ϵ_{MAX} nach Glg. (3.14) in %.
$(\,)^{v}$: 9063 von 9940 Bildern evaluiert (*frontalface_alt*);
$(\,)^{f}$: 2188 frontale Gesichter, 2123 evaluiert (*frontalface_alt*);
$(\,)^{k}$: Kombiniert mit Kaskade ($\mathcal{K}_{\mathrm{Bagg}}$);
$(\,)^{m}$: Mapping des Kaskadenausgangs [Pla99], $\mathcal{I}_{\mathrm{BioID}}$, siehe Kap. 3.6.5.
$(\,)^{sk}$: Augen auf 70 Pixel Breite herunterskaliert.

Autor	0,02	0,03	0,05	0,10
	PUT alle Bilder			
Vater Isophoten$^{v\ sk}$	42,01	78,83	97,12	99,24
Vater Kaskade$^{v\ ksk}$	34,87	76,29	96,02	98,18
Vater Kombiniert$^{v\ kmsk}$	35,13	77,04	97,06	99,24
	PUT frontale Bilder			
Vater Isophoten$^{f\ sk}$	44,98	85,21	98,63	99,76
Vater Kaskade$^{f\ ksk}$	36,60	81,49	98,07	99,25
Vater Kombiniert$^{f\ kmsk}$	36,69	81,77	98,54	99,72

Tabelle 3.4 FERET. Maximaler normalisierter Fehler ϵ_{MAX} nach Glg. (3.14) in %.
$(\,)^{f}$: 2662 Bilder evaluiert (*frontalface_alt*);
$(\,)^{k}$: Kombiniert mit Kaskade ($\mathcal{K}_{\mathrm{Bagg}}$);
$(\,)^{m}$: Mapping des Kaskadenausgangs [Pla99], $\mathcal{I}_{\mathrm{BioID}}$, siehe Kap. 3.6.5.
$(\,)^{sk}$: Augen auf 70 Pixel Breite herunterskaliert.

Autor	0,02	0,03	0,05	0,10
Kroon[Kro+09]			65,70	97,60
Valenti[VG12]	16,00*	35,00*	73,47	94,44
Vater Kombiniert$^{f\ kmsk}$	18,25	40,42	77,52	97,68

- Der in Kap. 2 beschriebene, selbst implementierte, Kaskadenklassifikatoransatz wurde zum Zwecke der Fusion mit dem Isophotenansatz modifiziert:
  - Es wurde die binäre Klassifikationsstruktur aufgebrochen, um einen quasi-kontinuierlichen Klassifikatorausgang zu erhalten, wobei jedes einzelne Merkmal *Stand-Alone* behan-

delt wurde, d. h., jedes Merkmal gleichberechtigt am Ergebnis beitragen kann.

  - Um eine sinnvolle Fusion zu ermöglichen, wurde mittels eines geeigneten Ansatzes der Kaskadenklassifikatorausgang auf den Wertebereich $[0, \ldots, 1]$ abgebildet. Dabei wurde der funktionale Zusammenhang der Abbildung sowie deren Einfluss auf das Lokalisierungsergebnis in Abhängigkeit vom Klassifikator, dessen Trainingsdaten und der eigentlichen Abbildungsfunktion untersucht.

- Die Methode wurde zur besseren Invarianz gegenüber der Größe der auszuwertenden Bilder – und damit der Augen – im Skalenraum implementiert und eine globale Parametrierung gewählt, welche eine gute Generalisierung der Methode ermöglicht.
- Es wurde eine vollständige eigene Implementierung sowohl in MATLAB als auch in C++ vorgenommen.

Die Methode wurde auf unterschiedlichen Datensätzen unter verschiedenen Gesichtspunkten evaluiert:

- Einfluss des Clusterns der Zentrumsmaps (Abb. 3.26),
- Einfluss der unterschiedlichen Gewichtungen und
- Möglichkeiten der Verbesserung durch Hinzunahme der Information aus hellen Zentren (Abb. 3.27),
- Einfluss der Fusion und hierbei insbesondere:
  - Einfluss der Kaskade (und Trainingsdatensatz) (Abb. 3.32 und Abb. 3.29(b)),
  - Einfluss der Abbildung S (Abb. 3.32),
  - Einfluss der Skalierung des Eingangsbildes (Abb. 3.29(a), Abb. 3.28(a) und Abb. 3.28(b)),
- Auswirkungen der Auflösung der Eingangsbilder auf die Kalibrierung S (Abb. 3.29(a)).

Insbesondere wurde in Tabelle 3.2 eine umfassende und nach bestem Wissen des Authors bezüglich der genauesten Ansätze aus der Literatur vollständige Übersicht von Methoden zur Irislokalisation sowie eine Einordnung der vorgestellten Methode gegeben.
Schlussfolgernd lässt sich in diesem Zusammenhang auf dem BioID-Datensatz die Überlegenheit der vorgestellten Methode hinsichtlich der Genauigkeit insbesondere im hochpräzisen Bereich, in dem der Fehler innerhalb des Bereiches der Pupille (auf Augenabstand normalisierter Fehler von $\leq 0{,}02$) liegt, gegenüber anderen Ansätzen innerhalb des Standes der Technik feststellen. Während auf ausgewählten Datenbanken andere Ansätze (Kroon et al. [Kro+09], BioID) im Bereich über $0{,}02$ ein besseres Ergebnis publiziert haben, zeigt sich bei Auswertungen auf anderen Datensätzen, dass sich dies nicht bestätigt, was beispielsweise im genannten Fall durch eine starke Parameteranpassung an den BioID-Datensatz zu erklären ist. Der Ansatz von Dong et al. [DZL11] zeigt bessere Ergebnisse für den nicht pixelgenauen Fall ($\epsilon_{\mathrm{MAX}} \geq 0{,}10$). Sie verwenden einen gestuften Klassifikatoransatz, der eher einer Detektionsaufgabe denn einer Lokalisierungsaufgabe zuzuordnen ist. Hierbei stellen sie herausragende Ergebnisse zur Detektion vor, welche allerdings mit geringerer Genauigkeit bei der präzisen Lokalisierung einhergehen (bereits weniger genau als der vorgeschlagene Ansatz für $\epsilon_{\mathrm{MAX}} \leq 0{,}05$). Ebenfalls sehr gute Ergebnisse für einen erlaubten Fehler $> 0{,}05$ zeigt der Ansatz von Markuš et al. [Mar+14].

Die Kombination der Informationen aus Isophoten und quasi-kontinuierlichem Kaskadenklassifikator führt zu unterstützenden Ergebnissen, was das Vermögen der vorgestellte Methode aufzeigt.

Weiterhin lässt sich für die Anwendung einer präzisen Irislokalisation als Vorverarbeitungsschritt zusammenfassen, dass ein kleiner Fehler, dessen Genauigkeit innerhalb des Durchmessers der Pupille liegt, neben einer direkten Verbesserung einer erscheinungsbasierten Blickrichtungsschätzung auch positiven Einfluss auf die Aufgabe des *Face Alignment* hat. Es wird berichtet, dass der Erfolg des *Face Alignment* bei Verwendung von LDA bei einem zugelassenen Fehler von $\epsilon_{\mathrm{MAX}} \leq 0{,}05$ um $25\,\%$ sinkt [Kro+09]. Aufgrund der schnellen Berechnung der Iriszentren und deren Genauigkeit eignet sich die Methode zur Implementierung in ein Rahmenwerk zur Blickrichtungsschätzung.

4 Monokulare 3D-Kopfposenschätzung

Die monokulare 3D-Kopfposenschätzung beschäftigt sich mit der Bestimmung des sechsdimensionalen Kopfposenvektors mit Hilfe erscheinungsbasierter Methoden des Computersehens. Dabei soll die aus drei Translationen sowie drei Rotationen bestehende Pose mit Hilfe einfacher Hardware, wie etwa Webcams, welche lediglich 2D-RGB-Daten liefern, bestimmt werden, um nicht auf aufwendige oder kostspielige Randbedingungen angewiesen zu sein und eine möglichst breite Anwendung der Methoden zu ermöglichen. Während erste, modernste handgehaltene Geräte bereits mit Tiefenkameras ausgestattet werden (iPhone X [Appb]), um über erscheinungsbasierte Methoden hinaus Möglichkeiten zur Kopfposen- und Blickrichtungsschätzung bereitzustellen, ist anzunehmen, dass monokulare oder monochromatische Kameras auch weiterhin eine weite Verbreitung finden, da sie auch in Zukunft eine robust zu verwendende und kostengünstige Alternative bleiben werden, die beispielsweise keine Interferenzen mit Tageslicht bei Verwendung aktiven Lichts zu befürchten hat und mit einem einzigen Sensor auskommt, bei dem die komplexe Kalibrierung bei Systemen mit mehreren Sensoren entfällt.

Auf der Anwendungsseite stellt die monokulare 3D-Kopfposenschätzung durch ihre vielseitigen Einsatzmöglichkeiten einen wichtigen Aspekt der Mensch-Maschine-Interaktion dar und wird genutzt in der Gesichtsdetektion sowie -erkennung, in der Emotionserkennung, in der Bildregistrierung und ist insbesondere notwendig für eine erscheinungsbasierte, kopfposeninvariante Blickrichtungsschätzung.

Um aus 2D-Bilddaten die Kopfpose zu schätzen, ist die Berechnung des optischen Flusses [LK81] ein häufig gewählter Ansatz [HB98; Xia+03], welcher auch in dieser Arbeit verfolgt wird. Ziel der monokularen 3D-

Kopfposenschätzung ist es, Verfahren zu erforschen, die es ermöglichen, die vollständige Bewegung des Kopfes aus einer 2D-Bildsequenz in Echtzeit robust zu bestimmen.

4.1 Problemstellung

Der Bewegungszustand des Kopfes wird durch die sechsdimensionale Kopfpose beschrieben, welche bereits zu Anfang der Arbeit in Abb. 1.1 illustriert wird.

Ein Problem bei der Nutzung einfacher Hardware für die erscheinungsbasierte Kopfposenschätzung ist das inhärente Fehlen von Tiefeninformation sowie eine niedrige Bildauflösung. Insbesondere Ersteres führt zum Aperturproblem bei der Berechnung des optischen Flusses, welcher als Vektorfeld der Bewegung der im Bild dargestellten Punkte verstanden werden kann. Das Aperturproblem beschreibt allgemein die Schwierigkeit, dass durch Bestimmung des lokalen Gradienten, beispielsweise in einem kleinen Ausschnitt eines Bildes, nur die parallel zum Bildgradienten verlaufende Bewegung des optischen Flusses bestimmt werden kann. Das Aperturproblem drückt sich im hier behandelten Szenario durch Ambiguitäten in der Schätzung unterschiedlicher Bewegungsrichtungen aus [VMP15]. Bei großen Kopfdrehungen in die Bildebene hinein (Neigen, Gieren) leidet der Ansatz aufgrund dessen unter Stabilitätsproblemen, wenn der lokale Gradient nicht ausreicht, um beispielsweise zwischen Gieren und x-Translation zu unterscheiden.

Ein kritischer Schritt bei der Berechnung des optischen Flusses ist die Bestimmung der Inversen der Hessematrix, welche die Ableitungen des Bewegungsmodells nach seinen Parametern konstituiert, wobei die Parameter für die 3D-Kopfposenschätzung den Elementen des Vektors der Pose $\mathbf{p}$ entsprechen. Mehrdeutigkeiten in der Hessematrix aufgrund des Aperturproblemes sowie kleine Bildgradienten, welche in homogenen Bildregionen auftreten, führen zu Singularitäten bei der Invertierung der Hessematrix. Mathematisch drückt sich dieser Zusammenhang durch eine hohe Konditionszahl der Hessematrix aus, welche durch den Quotienten aus größtem und kleinstem Eigenwert der Matrix nach einer Singulärwertzerlegung beschrieben werden kann. Die Konditionszahl beschreibt anschaulich, wie stark kleine Änderungen (Störungen) am

„Eingang“ einer Matrix durch Multiplikation den „Ausgang “ beeinflussen. Bei Konditionszahlen deutlich größer eins reichen also kleine Variationen im Eingang, um starke Veränderungen im Ausgang hervorzurufen.

Im Ansatz des optischen Flusses ist die Hessematrix linear mit dem Fehlerbild und dem optischen Fluss verbunden. Da die Hessematrix im Falle der Kopfposenschätzung die Ableitungen des Bewegungsmodells enthält – und somit Terme, die als Sensitivitäten des optischen Flusses bezüglich der einzelnen Bewegungsrichtungen verstanden werden können – bedeutet eine hohe Konditionszahl, dass kleine, durch Bildrauschen, Verdeckungen, oder vorausgehende Fehlschätzungen des Modells große (negative) Auswirkungen auf die Posenschätzung haben können.

In der Literatur wird zur Kompensation einer hohen Konditionszahl von La Cascia et al. [LSA00] und Xiao et al. [Xia+03] ein Regularisierungsterm zweiter Ordnung eingesetzt, welcher den optischen Fluss dämpfen soll. Dies geschieht allerdings unter Vorgabe einer skalarwertigen Gewichtungsfunktion, welche den aktuellen Zustandsvektor (hier: Kopfpose) nicht berücksichtigt. Weitere Herausforderungen bei der Bestimmung der Kopfpose ergeben sich durch veränderliche Beleuchtungsbedingungen, Verdeckungen sowie komplexe Bildhintergründe und -rauschen.

Obwohl zahlreiche Ansätze und Lösungen zur Kopfposenschätzung in der Literatur vorgeschlagen wurden [MT09; ZG14], stellt das robuste Handhaben der Probleme der niedrigen Bildqualität und Auflösung weiterhin eine Herausforderung dar. Der in dieser Arbeit vorgeschlagene Ansatz beschäftigt sich unter diesen Randbedingungen mit der Lösung des Problems.

4.2 Stand der Wissenschaft

Als wichtigen Bestandteil der Mensch-Maschine-Interaktion erfährt die Kopfposenschätzung große Aufmerksamkeit innerhalb des maschinellen Sehens, Computersehens und maschinellen Lernens, wobei Übersichten von Murphy-Chutorian und Trivedi [MT09] sowie Zhang und Gomes [ZG14] zu finden sind. Während es zahlreiche Ansätze gibt, die auf dreidimensionale (Tiefen-)daten zurückgreifen, soll hier unter Berück-

sichtigung der Zielsetzung der Arbeit eine umfangreiche Übersicht über relevante erscheinungsbasierte Ansätze zur 3D-Kopfposenschätzung gegeben werden. Die Sensitivität gegenüber Beleuchtungsänderungen sowie das Auftreten des Aperturproblemes werden früh als Probleme, die bei Modellen basierend auf der Berechnung des optisches Flusses auftreten, in der Literatur beschrieben [Hor86]. Parametrische Modelle zur Beschreibung der Geometrie des Kopfes und veränderlicher Beleuchtungsbedingungen wurden von Hager und Belhumeur [HB98] sowie La Cascia et al. [LSA00] vorgestellt. Hager und Belhumeur [HB98] verwenden neben parametrischen Bewegungsmodellen, in denen sie unter anderem affine Bewegungen untersuchen, ein lineares Modell zur Beschreibung von Beleuchtungsänderungen. Unter der Annahme der Erfüllung lambertscher Strahlung bestimmen sie Basis-Beleuchtungsvektoren, um ein Modell zu erstellen, welches Beleuchtungsvariationen kompensiert. Einen ähnlichen Ansatz verwenden La Cascia et al. [LSA00]. Während keine lambertsche Strahlung angenommen wird, wird auch die in [HB98] getroffene Annahme, dass alle Bilder der gleichen Oberfläche (selbigen Objektes) unter verschiedenen Beleuchtungsbedinungen in einem dreidimensionalen Unterraum des Raumes aller möglichen Bilder des Objekts liegen [Sha92], nicht getroffen. Sie verwenden dann einen linearen Ansatz zur Kompensation der Beleuchtungsunterschiede und erhalten Beleuchtungs*templates* durch SVD von Trainingsbildern, welche unter unterschiedlichen Beleuchtungsbedingungen aufgenommen werden. Sie stellen außerdem einen Ansatz zur Regularisierung des schlecht gestellten Problems der Invertierung der Hessematrix vor. Sie definieren das Problem als Energieminimierung und fügen dem optischen Fluss Terme jeweils für die Beleuchtungskompensation und für die Bewegung hinzu, die das Modell davor bewahren, dass Lösungen für die Beleuchtungs- und Bewegungsparameter zu hohe Werte annehmen. Unter der Annahme unabhängiger Beleuchtungs- und Bewegungsparameter, welche in der Realität nicht notwendigerweise gegeben sein müssen, lösen sie das Problem zunächst im Unterraum der Bewegungsparameter und anschließend für die Beleuchtung. Weiterhin wird auch die Sensitivität bezüglich der Initialisierung des Modells diskutiert, wobei sie schlussfolgern, dass, aufgrund der höheren Robustheit gegenüber der Initialisierung, einfache parameterische Kopfmodelle, wie etwa die

Modellierung durch einen Zylinder, komplexen Modellen vorzuziehen sind.

Baker und Matthews [BM01] präsentieren den *Inverse Compositional Algorithm*, welcher auf dem *Forward Additive* [LK81] und dem *Forward Compositional Algorithm* [HB98; SS01] beruht, und zeigen, dass dieser bei vergleichbar genauen Ergebnissen deutlich effizienter zu berechnen ist als die vorwärts-Algorithmen. In ihrem Ansatz ist die Hessematrix nicht abhängig von den Bewegungsparametern und somit unabhängig von den Recheniterationen der Methode des steilsten Abstiegs innerhalb eines Registrierungsschritts und muss so nur einmal am Anfang jeder Registrierung berechnet werden. Xiao et al. [Xia+03] ziehen ebenfalls ein perspektivisches Starrkörpermodell und eine Repräsentation des Kopfes durch einen Zylinder einem komplexeren Modell vor. Um das Tracking zu stabilisieren, nutzen sie zum einen *Iteratively Re-Weighted Least Squares* [Bla92], um Pixel, die durch Okklusion, Rauschen oder Nicht-Starrkörperbewegungen einen negativen Einfluss auf die Berechnung des optischen Flusses haben, niedriger zu gewichten, und berücksichtigen zum anderen die Dichte der Pixel, die aufgrund der 3D-Modellierung des Kopfes durch die Tiefe gegeben ist, um ein pixelweises Gewicht einzuführen. Sie greifen die Idee der Regularisierung des optischen Flusses auf und fügen der Zielfunktion additiv einen Tikhonov-Regularisierungsterm hinzu, welcher das parametrische Bewegungsmodell enthält. Dabei wählen sie den skalarwertigen Regularisierungsparameter monoton fallend mit steigender Iterationszahl bei der Bestimmung des optischen Flusses, unabhängig vom aktuellen Bewegungszustand des Kopfes. Um eine Langzeitrobustheit des Systems zu gewährleisten, wird eine Stabilisierung der Kopfposenberechnung durch Bildregistrierung dynamischer Objektmodelle basierend auf vorigen Schätzergebnissen durchgegeführt, bei denen Ausreißer – auf Basis der lokalen (Grauwert-)Varianz definiert – aktuelle *Templates* zusammen mit der Pose abgespeichert und bei Bedarf als Referenz*template* verwendet werden. Xiao et al. [Xia+04] kombinieren 2D- und 3D-*Active Appearance*-Modelle (AAM), um die Kopfpose zu schätzen. Statt auf den optischen Fluss zurückzugreifen, definieren sie Bewegungsrandbedingungen, die den großen Lösungsraum der 3D-AAM einschränken, um so zu sinnvollen Ergebnissen mit Hilfe des *Inverse Compositional-*

Algorithmus (IC) für die Kopfpose zu gelangen. Sung et al. [SKK08] kombinieren den Optischen-Fluss-Ansatz und ein Zylindermodell mit einem 2D+3D-AAM, wobei sie die jeweiligen Gültigkeitsbereiche der einzelnen Methoden bezüglich des Neigens und Gierens berücksichtigen. Während der optische Fluss eingesetzt wird, um die Pose grob zu schätzen, korrigiert das AAM die Schätzung der Bewegungsparameter für den optischen Fluss zur Initialisierung des nächsten Frames. Jang und Kanade [JK08] verzichten auf eine Berechnung des optischen Flusses und nutzen eine Merkmalsregistrierung und Kalman-Filter zur Bestimmung der Kopfpose. Sie berechnen *Scale Invariant Feature Transform*-Deskriptoren (SIFT) und bestimmen mit Hilfe eines Starrkörpermodells die Bewegung eines Zylinders zwischen zwei Frames, während sie durch Auswerten eines Gütekriteriums zur Abschätzung der Eignung eines neuen *Templates* eine Kopfposendatenbank aufbauen. Durch Fusion in einem Kalman-Filter der besten Pose aus der Datenbank mit der Schätzung aus der Starrkörperbewegung aus der Merkmalsregistrierung wird so die neue Pose geschätzt. An und Chung [AC09] nutzen ein Ellipsoidenmodell und schlagen eine nutzerabhängige Beleuchtungskompensation vor. Sie argumentieren, dass bei einer Extraktion von Basis-Beleuchtungsvektoren aus einem allgemeinen Trainingsdatensatz, in welchem Beleuchtungsvariationen von Gesichtern unterschiedlicher Personen enthalten sind, nicht zwischen intrinsischen (antropometrisch) und extrinsischen (Beleuchtungs-) Varianzen unterschieden werden kann. Chen et al. [CCH10] nutzen den optischen Fluss und ein probabilistisches Kopfmodell, welches jedes Pixel durch eine kontinuierlich aktualisierte Gauß-Verteilung repräsentiert, um die Schätzung robuster zu gestalten. Prasad und Aravind [PA10] verfolgen SIFT-Deskriptoren, berechnen daraus 2D-3D-Korrespondenzen des Objektes im Bild und bestimmen mit Hilfe des POSIT-Algorithmus [DD95] die Kopfpose. Die Autoren Orozco et al. [Oro+13] setzen einen *On-line Appearance-Based Tracker* (OABT) ein, um die 3D-Kopfpose aus der Gesichtsregion, exklusive der Augenregion, zu schätzen, wobei sie gleichzeitig Gesichtsmerkmale wie Augenbrauen und -lider lokalisieren. Für ihren Ansatz projizieren sie alle gewonnenen Bilder in einen Texturraum, wo sie als multivariate Gauß-Verteilungen modelliert werden. Aus diesem Raum wird während eines Beobachtungsprozesses eine Wahrscheinlichkeitsfunk-

tion für die aktuelle Erscheinung erstellt, während der Raum ständig aktualisiert wird, um Veränderungen des Objektes zu kompensieren. Im anschließenden Transitionsprozess wird anhand eines Bewegungsmodells die Erscheinungsvariation zwischen aufeinanderfolgenden Frames bestimmt. Für ihren Algorithmus verwenden sie das *3D Face Candide Model* [Ryd87], welches von ihnen manuell initialisiert werden muss. Chen et al. [Che+16a] vollführen eine Kopfposenschätzung in ultra-niedrig aufgelösten Bildern, mit Kopfregionen von 10×10 Pixeln. Sie verwenden HOG-Merkmale und *Support Vector Regression*, um auf die Kopfpose zu schließen. In [Che+16b] werden SIFT-Deskriptoren zwischen sukzessiven Frames registriert und, wie beim optischen Fluss, ein *Template Warping* durchgeführt, während die Tiefenänderung separat über die SIFT-Korrespondenzen bestimmt wird. Nach Fusion der Informationen und der Schätzung der Pose wird mittels Autoregression die aktuelle Pose geschätzt und im Falle einer sicheren Schätzung das aktuelle *Template* einer *Template*-Sammlung hinzugefügt.

4.3 Lösungsansatz und eigener Beitrag

Im hier vorgestellten Ansatz wird eine Implementierung des *Forward Compositional Image Alignment Algorithm* (FC) zur Berechung des optischen Flusses unter Verwendung des Verfahrens des steilsten Abstiegs mittels Gauß-Newton-Verfahrens zur Optimierung einer Gütefunktion bei der Bestimmung des sechsdimensionalen Zustandsvektors der 3D-Kopfpose eingesetzt. Es wird ein neuartiger Regularisierungsterm für Bilderregistrierungsmethoden mit der Anwendung der erscheinungsbasierten Kopfposenschätzung vorgestellt, in welchem unabhängige, durch projektive Transformation verfolgter robuster Deskriptoren berechnete Bewegungsinformation fusioniert wird, um die Robustheit und Genauigkeit des Systems zu verbessern.

Es werden dabei im Einzelnen folgende Beiträge zum Stand der Technik geliefert [VMP15; VPP15; VP16b; VP17; Vat+16]:

1. Um die hohe Konditionszahl der Hessematrix zu verringern, wird in dieser Arbeit ein bewegungsabhängiger Regularisierungsterm in die Berechnung der Zielfunktion integriert. Durch Einsetzen

eines vektorwertigen Regularisierungsparameter können reine sowie kombinierte Bewegungsrichtungen unabhängig voneinander manipuliert werden. Es wird gezeigt, wie eine sinnvolle Wahl des Regularisierungsparameters positiven Einfluss auf das Schätzergebnis ausübt.

2. Es wird, durch Registrieren von robusten Deskriptoren in sukzessiven Frames und der Berechnung einer projektiven Transformation, eine vom optischen Fluss unabhängige Bewegungsschätzung durchgeführt. Es werden Kriterien zur geeigneten Auswahl von Punktepaaren erörtert und deren Performanz getestet.

3. Es wird gezeigt, wie die vom optischen Fluss unabhängige Bewegungsschätzung genutzt werden kann, um eine bewegungsadaptive Regularisierung durch Integration der genannten Methoden zu implementieren und die Kopfposenschätzung ohne empirische Vorgaben durch eine direkte Kopplung von Starrkörperbewegung und Regularisierung online und robust zu gestalten. Es wird insbesondere die Konditionierung der Hessematrix diskutiert.

4.4 Bildregistrierung und optischer Fluss

Der ursprünglich von Lucas und Kanade [LK81] vorgestellte und nach ihnen benannte Algorithmus zum Angleichen eindimensioner Funktionen wird in dieser Arbeit in seiner erweiterten zweidimensionalen und an die erscheinungsbasierte Kopfposenschätzung angepassten Form zur Berechung des optischen Flusses eingesetzt. Die Berechnung des optischen Flusses dient im einfachsten, zweidimensionalen Fall dem Ziel, einen Registrierungsschritt zwischen zugeordneten Objekten in unterschiedlichen Bildern durchzuführen. Dies geschieht, indem unter Annahme eines Bewegungsmodells und der Bestimmung der Sensitivitäten dieses Modells bezüglich seiner abhängigen Parameter der Fehler zwischen den Erscheinungen der zuzuordnenden Bildobjekte minimiert wird.

Für die Berechnung des optischen Flusses wird in dieser Arbeit angenommen, dass, für den monochromatischen Fall (einkanaliges Bild), die grauwertige Erscheinung eines Bildausschnittes bzw. eines Objektes

$$g_{\mathrm{Obj}}(\mathbf{u}), \quad \mathbf{u} \in \Omega_{\mathrm{Obj}}\,, \tag{4.1}$$

wobei $\mathbf{u} = [u, v]^{\mathrm{T}}$ der Vektor der Pixelkoordinaten des Objektes im Eingangsbildes beschreibt, sich nicht erheblich zwischen zwei aufeinanderfolgenden Frames ändert. Für ein zweidimensionales Tracking des Ortes eines Objekts werden also neue Koordinaten $\mathbf{u}'$ gesucht, an denen die Grauwerte $g_{\mathrm{Obj}}(\mathbf{u})$ aller Pixel $\mathbf{u} \in \Omega_{\mathrm{Obj}}$ wiedergefunden werden. Aufgrund nicht vollkommenem Abhandenseins von Änderungen in der Erscheinung des Objekts zwischen sukzessiven Bildern geschieht das Finden der neuen Koordinaten in der Regel nicht fehlerlos, weshalb eine Zielfunktion formuliert wird, unter derer Minimierung man den die Bewegung beschreibenden Zustand sucht. Unter der Annahme, dass die Änderung des Zustands von $g_{\mathrm{Obj}}(\mathbf{u})$ zwischen den Frames N und $N+1$ durch eine Transformation $T(\mathbf{u}, \mathbf{p})$ mit einem Parametervektor $\mathbf{p}$ beschrieben werden kann, besteht der Registrierungsschritt darin, $\mathbf{p}$ so zu bestimmen, dass

$$g_{\mathrm{Obj}}^{N+1}\left(T(\mathbf{u}, \mathbf{p})\right) = g_{\mathrm{Obj}}^{N+1}(\mathbf{u}') = g_{\mathrm{Obj}}^{N}(\mathbf{u}) \tag{4.2}$$

gilt, wobei $\mathbf{u}' = T(\mathbf{u}, \mathbf{p})$ den Koordinaten des Objekts $g_{\mathrm{Obj}}^{N}(\mathbf{u})$ im neuen Frame $N+1$ entsprechen.

4.4.1 Kopfposenschätzung durch Berechnung des optischen Flusses

Für die dreidimensionale erscheinungsbasierte Kopfposenschätzung ist der Parametervektor durch die Pose

$$\mathbf{p} = \left[r_{\mathrm{x}}, r_{\mathrm{y}}, r_{\mathrm{z}}, t_{\mathrm{x}}, t_{\mathrm{y}}, t_{\mathrm{z}}\right]^{\mathrm{T}} \tag{4.3}$$

definiert, wobei r für die Rotationen um die x-, y-, und z-Achse (Rollen, Gieren, Neigen) und t für die entsprechenden Translationen steht.

Um den Kopf zu modellieren, wird dieser hier durch einen Zylinder angenähert, dessen Bewegung durch die in dieser Arbeit vorgeschlagene Methode geschätzt wird. Durch Beschreibung des Zylinders durch ein Starrkörpermodell, welches durch die 3D-Punkte $\mathbf{x} = [x, y, z]$ – in homogenen Koordinaten durch $\tilde{\mathbf{x}} = [x, y, z, 1]^{\mathrm{T}}$ – repräsentiert wird, sowie durch Anwenden einer Starrkörperbewegung $\mathbf{M}$ [BM98; MLS94]

$$\mathbf{M}(\mathbf{p}) = \begin{pmatrix} 1 & -r_{\mathrm{z}} & r_{\mathrm{y}} & t_{\mathrm{x}} \\ r_{\mathrm{z}} & 1 & -\omega_x & t_{\mathrm{y}} \\ -r_{\mathrm{y}} & r_{\mathrm{x}} & 1 & t_{\mathrm{z}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.4}$$

werden die 3D-Punkte $\tilde{\mathbf{x}}$ der Starrkörperoberfläche mittels des Warps $\widetilde{\mathbf{W}}$ an die neuen Koordinaten

$$\tilde{\mathbf{x}}' = \mathbf{M}(\mathbf{p})\,\tilde{\mathbf{x}} = \widetilde{\mathbf{W}}(\tilde{\mathbf{x}}, \mathbf{p}) \tag{4.5}$$

transformiert. Mit einer geeigneten 3D-2D-Abbildung $\widetilde{\mathbf{P}}$ lässt sich dann der Zusammenhang zwischen Pixelkoordinaten und Pose schreiben als:

$$\tilde{\mathbf{u}}' = \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \widetilde{\mathbf{P}}\left(\widetilde{\mathbf{W}}(\tilde{\mathbf{x}}, \mathbf{p})\right) . \tag{4.6}$$

Für die Abbildung $\widetilde{\mathbf{P}}$ wird in dieser Arbeit ein Lochkameramodell gewählt, welches mittels Zentralprojektion Weltpunkte auf Bildpunkte abbildet. Das Lochkameramodell ist durch seine intrinsiche Kameramatrix $\mathbf{K}$,

$$\mathbf{K} = \begin{pmatrix} fS_{\mathrm{x}} & 0 & O_{\mathrm{x}} \\ 0 & fS_{\mathrm{y}} & O_{\mathrm{y}} \\ 0 & 0 & 1 \end{pmatrix} , \tag{4.7}$$

beschrieben, wobei f die Brennweite, S_{x} und S_{y} Skalierungsfaktoren zur Korrektur nicht quadratischer Pixel auf dem Sensor, und O_{x} sowie O_{y} Offsets zwischen der optischen Achse des Sensors und dem Bildkoordinatensysten darstellen. Unter idealen Bedingungen geht die optische Achse des Sensors durch den Mittelpunkt der Bildebene, weshalb O_{x} und O_{y} Werte in der Nähe der halben Bildgröße der Sensordaten annehmen sollten. Die Zentralprojektion $\widetilde{\mathbf{P}}$ kann dann in homogenenen Koordinaten mit der intrinsischen Kameramatrix geschrieben werden als:

$$\tilde{\mathbf{u}} = \mathbf{K}\left[\mathbf{I}_{3\times 3}\middle|\mathbf{0}\right]\tilde{\mathbf{x}} = \widetilde{\mathbf{P}}(\tilde{\mathbf{x}}) , \tag{4.8}$$

wobei $\mathbf{I}$ die Einheitsmatrix und $\mathbf{0}$ ein Spaltenvektor von Nullen beschreibt. Durch Kombination von Glg. (4.5) und (4.8) lässt sich die

Transformation $T(\cdot)$ aus Glg. (4.2) für eine Starrkörperbewegung unter Verwendung eines Lochkameramodells in homogenen Koordinaten schreiben als:

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \begin{pmatrix} fS_{\mathrm{x}} & 0 & O_{\mathrm{x}} \\ 0 & fS_{\mathrm{y}} & O_{\mathrm{y}} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & -r_{\mathrm{z}} & r_{\mathrm{y}} & t_{\mathrm{x}} \\ r_{\mathrm{z}} & 1 & -r_{\mathrm{x}} & t_{\mathrm{y}} \\ -r_{\mathrm{y}} & r_{\mathrm{x}} & 1 & t_{\mathrm{z}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} . \quad (4.9)$$

Vereinfachen von Glg. (4.9) und Auflösen der homogenen Koordinaten liefert für die Bildkoordinten $\mathbf{u}'$ als Projektion der mit $\mathbf{p}$ gewarpten Modellkoordinaten $\mathbf{x}$:

$$\mathbf{P}\Big(\mathbf{W}\left(\mathbf{x},\mathbf{p}\right)\Big) = \mathbf{u}' = \begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} \dfrac{fS_{\mathrm{x}}(x - r_{\mathrm{z}}y + r_{\mathrm{y}}z + t_{\mathrm{x}})}{r_{\mathrm{x}}y - r_{\mathrm{y}}x + z + t_{\mathrm{z}}} \\ \dfrac{fS_{\mathrm{y}}(r_{\mathrm{z}}x + y - r_{\mathrm{x}}z + t_{\mathrm{y}})}{r_{\mathrm{x}}y - r_{\mathrm{y}}x + z + t_{\mathrm{z}}} \end{pmatrix} + \begin{pmatrix} O_{\mathrm{x}} \\ O_{\mathrm{y}} \end{pmatrix} . \quad (4.10)$$

Mit Hilfe der Projektion $\mathbf{P}$ und dem Warp $\mathbf{W}$ kann nun die gesuchte Pose $\mathbf{p}$ in Abhängigkeit des Fehlers zwischen einem Objektmodell $g^0_{\mathrm{Obj}}(\mathbf{u})$ und den Grauwerten $g^N(\mathbf{P}(\mathbf{W}(\mathbf{p},\mathbf{x})))$ der durch Projektion an den Koordinaten $\mathbf{W}(\mathbf{p},\mathbf{x})$ liegenden 3D-Modellpunkte bestimmt werden. Das Objektmodell entspricht einem $m \times n$ Pixel großen Bildausschnitt $g(\mathbf{u})$, $\mathbf{u} \in \Omega_{\mathrm{Obj}}$ und repräsentiert das zu verfolgende Objekt. Das Objekt muss zu Beginn der Objektverfolgung durch den optischen Fluss entweder manuell oder durch eine automatische Detektion – mit oder ohne Verwendung von A priori-Wissen wie anthropometrischen Werten – festgelegt werden.

Das Minimierungsproblem zum Finden der neuen Pose im nächsten Bild $g^{N+1}(\mathbf{u})$ lässt sich dann mit dem FC-Algorithmus [SS01] für inkrementelle Änderungen $\Delta\mathbf{p}$ der Pose wie folgt definieren:

$$\Delta\mathbf{p}^* = \arg\min_{\Delta\mathbf{p}} \mathrm{E}(\Delta\mathbf{p}) , \quad (4.11)$$

mit der Definition der Fehlerfunktion für den Registrierungsschritt

$$\mathrm{E}(\Delta\mathbf{p}) = \sum_{\mathbf{u}\in\Omega_{\mathrm{Obj}}} \left[g^N \left(\mathbf{P}\left(\mathbf{W}\left(\mathbf{x},\mathbf{p}\right) \circ \mathbf{W}\left(\mathbf{x},\Delta\mathbf{p}\right) \right) \right) - g^0(\mathbf{u}) \right]^2 \quad (4.12)$$

$$= \sum_{\mathbf{u}\in\Omega_{\mathrm{Obj}}} \left[g^N \left(\mathbf{P}\left(\mathbf{W}\left(\mathbf{W}(\mathbf{x},\Delta\mathbf{p}),\mathbf{p} \right) \right) \right) - g^0(\mathbf{u}) \right]^2, \quad (4.13)$$

wobei $\circ$ die Komposition von Funktionen kennzeichnet. Das Update des Warps geschieht dann durch

$$\mathbf{W}(\mathbf{x},\mathbf{p}) \leftarrow \mathbf{W}\left(\mathbf{x},\mathbf{p}\right) \circ \mathbf{W}\left(\mathbf{x},\Delta\mathbf{p}\right) . \quad (4.14)$$

In Glg. (4.13) wird ausgenutzt, dass bis zur ersten Ordnung in $\Delta\mathbf{p}$ gilt (siehe [BM01] sowie Kap. 4.4.3):

$$\mathbf{W}\left(\mathbf{x},\mathbf{p}\right) \circ \mathbf{W}\left(\mathbf{x},\Delta\mathbf{p}\right) = \mathbf{W}\left(\mathbf{W}\left(\mathbf{x},\Delta\mathbf{p}\right),\mathbf{p}\right) . \quad (4.15)$$

Der Ansatz unterscheidet sich vom *Forward Additive Image Alignment Algorithm* (Lucas-Kanade-Algorithmus, FA-Algorithmus) dadurch, dass die Posenänderung über eine Verkettung des Warps der aktuellen Pose mit dem Warp einer inkrementellen Änderung der Pose in das Modell integriert wird, statt den neuen Grauwert an einer Position, die durch additive Überlagerung erzeugt wird, zu bestimmen:

$$\mathrm{E}(\Delta\mathbf{p}) = \sum_{\mathbf{u}\in\Omega_{\mathrm{Obj}}} \left[g^N \left(\mathbf{P}\left(\mathbf{W}\left(\mathbf{x},\mathbf{p}+\Delta\mathbf{p}\right) \right) \right) - g^0(\mathbf{u}) \right]^2 . \quad (4.16)$$

Der Vorteil des Verwendens der verketteten Formulierung zeigt sich bei Lösen von Glg. (4.13) für $\Delta\mathbf{p}$.

Aufgrund unzureichender analytischer Herleitungen in der Literatur beschäftigt sich der folgende Abschnitt mathematisch mit dem Lösen von Glg. (4.13).

4.4.2 Herleitung der Taylor-Approximation des *Forward Compositional*-Algorithmus für den dreidimensionalen optischen Fluss unter einer Lochkameraabbildung

Der folgende Abschnitt baut auf der Arbeit von Baker und Matthews [BM01] auf und soll eine umfassende Herleitung der Gleichungen des

optischen Flusses für die erscheinungsbasierte Kopfposenschätzung mittels 3D-Starrkörpermodell und Lochkameramodell bereitstellen, die den 2D-Fall aus [BM01] erweitert und ausführlich für den hier verwendeten Ansatz mathematisch beschreibt. Dem Autor ist trotz Anwendungen der Gleichungen in anderen Arbeiten keine entsprechende mathematisch geschlossene Herleitung in der Literatur bekannt.

Um die Zielfunktion in Glg. (4.13) zu lösen, wird zunächst eine Linearisierung des Termes vorgenommen und anschließend nach der interessierten Größe $\Delta\mathbf{p}$ aufgelöst. Für die Taylorerweiterung für den *Forward Compositional*-Algorithmus (FC) bis zur ersten Ordnung in $\Delta\mathbf{p}$ um das Tripel der Punkte $\Delta\mathbf{p} = \mathbf{0}, \mathbf{p} = \mathbf{p}_0, \mathbf{x} = \mathbf{x}_0$ werden die Verkettungen im ersten Term in Glg. (4.13) zunächst als Funktion von $\Delta\mathbf{p}$ geschrieben:

$$f(\Delta\mathbf{p}) := g^N\Big(\mathbf{P}\big(\mathbf{W}\big(\mathbf{W}(\mathbf{x}=\mathbf{x}_0, \Delta\mathbf{p}), \mathbf{p}=\mathbf{p}_0\big)\big)\Big) \tag{4.17}$$

$$= \Big(g \circ \mathbf{P} \circ \mathbf{W}_{\mathbf{p}_0} \circ \mathbf{W}_{\mathbf{x}_0}\Big)(\Delta\mathbf{p})\,. \tag{4.18}$$

Er wurden dabei folgende Funktionen eingeführt:

$$g : \mathbb{R}^2 \to \mathbb{R}, \qquad \mathbf{u} \mapsto g(\mathbf{u})\,, \tag{4.19}$$

$$\mathbf{P} : \mathbb{R}^3 \to \mathbb{R}^2, \qquad \mathbf{x} \mapsto \mathbf{P}(\mathbf{x}) = \mathbf{u}\,, \tag{4.20}$$

$$\mathbf{W}_{\mathbf{p}} : \mathbb{R}^3 \to \mathbb{R}^3, \qquad \mathbf{x} \mapsto \mathbf{W}(\mathbf{x}, \mathbf{p})\,, \tag{4.21}$$

$$\mathbf{W}_{\mathbf{x}} : \mathbb{R}^6 \to \mathbb{R}^3, \qquad \mathbf{p} \mapsto \mathbf{W}(\mathbf{x}, \mathbf{p})\,. \tag{4.22}$$

Die Taylorapproximation erster Ordnung $Tf^{(1)}(\Delta\mathbf{p})$ von f ausgewertet an der Stelle $\Delta\mathbf{p} = \mathbf{0}$ ist dann zu schreiben als:

$$Tf^{(1)}(\Delta\mathbf{p})|_{\Delta\mathbf{p}=\mathbf{0}} = Tf^{(1)}(\mathbf{0}) = f(\mathbf{0}) + \frac{\partial f}{\partial \Delta\mathbf{p}}(\mathbf{0}) \cdot \Delta\mathbf{p}\,. \tag{4.23}$$

Der erste Term $f(\mathbf{0})$ ergibt sich direkt durch Auswerten von Glg. (4.17) zu

$$f(\mathbf{0}) = g^N\Big(\mathbf{P}\big(\mathbf{W}\big(\mathbf{W}(\mathbf{x}, \mathbf{0}), \mathbf{p}\big)\big)\Big)\,, \tag{4.24}$$

was sich mit Hilfe des Identitätswarps [BM01] $\mathbf{W}(\mathbf{x}, \mathbf{0}) = \mathbf{x}$ schreiben lässt als:

$$f(\mathbf{0}) = g^N\Big(\mathbf{P}\big(\mathbf{W}\left(\mathbf{x}, \mathbf{p}\right)\big)\Big)\,. \tag{4.25}$$

Der zweite Term muss durch das Produkt der Jacobimatrizen als Ableitungen von Funktionen mehrdimensionaler Veränderlicher definiert werden, wobei die Reihenfolgen der Verkettung beachtet werden muss. (Verkettung ist in der Regel nicht kommutativ.) Da die Zielfunktion trotz dreifacher Verkettung nur von einer Variablen abhängt, erfolgt die Taylorapproximation nur eindimensional:

$$\frac{\partial f}{\partial \Delta \mathbf{p}}(\mathbf{0}) = \mathbf{J}_g\left(\mathbf{P}\left(\mathbf{W}\left(\mathbf{W}(\mathbf{x}, \Delta\mathbf{p}), \mathbf{p}\right)\right)\right) \tag{4.26}$$

$$\cdot \mathbf{J}_{\mathbf{P}}\Big(\mathbf{W}\left(\mathbf{W}(\mathbf{x}, \Delta\mathbf{p}), \mathbf{p}\right)\Big) \tag{4.27}$$

$$\cdot \mathbf{J}_{\mathbf{W}_{\mathbf{P}}}\left(\mathbf{W}(\mathbf{x}, \Delta\mathbf{p})\right) \tag{4.28}$$

$$\cdot \mathbf{J}_{\mathbf{W}_{\mathbf{x}}}(\Delta\mathbf{p}), \tag{4.29}$$

wobei $\frac{\partial f}{\partial \Delta \mathbf{p}}(\mathbf{0})$ am Tripel $\Delta\mathbf{p} = \mathbf{0}, \mathbf{p} = \mathbf{p}_0, \mathbf{x} = \mathbf{x}_0$ ausgewertet wird. Einsetzen für $\mathbf{p} = \mathbf{p}_0$ und $\mathbf{x} = \mathbf{x}_0$ ergibt:

$$\frac{\partial f}{\partial \Delta \mathbf{p}}(\mathbf{0}) = \mathbf{J}_g\left(\mathbf{P}\left(\mathbf{W}\left(\mathbf{W}(\mathbf{x}_0, \Delta\mathbf{p}), \mathbf{p}_0\right)\right)\right) \tag{4.30}$$

$$\cdot \mathbf{J}_{\mathbf{P}}\Big(\mathbf{W}\left(\mathbf{W}(\mathbf{x}_0, \Delta\mathbf{p}), \mathbf{p}_0\right)\Big) \tag{4.31}$$

$$\cdot \mathbf{J}_{\mathbf{W}_{\mathbf{P}_0}}\left(\mathbf{W}(\mathbf{x}_0, \Delta\mathbf{p})\right) \tag{4.32}$$

$$\cdot \mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}}(\Delta\mathbf{p}). \tag{4.33}$$

Hier bezeichnet allgemein $\mathbf{J}_h$ die Jacobimatrix einer Funktion h. Mit

$$\mathbf{W}(\mathbf{x}, \mathbf{p}) = \begin{pmatrix} 1 & -r_{\mathrm{z}} & r_{\mathrm{y}} & t_{\mathrm{x}} \\ r_{\mathrm{z}} & 1 & -r_{\mathrm{x}} & t_{\mathrm{y}} \\ -r_{\mathrm{y}} & r_{\mathrm{x}} & 1 & t_{\mathrm{z}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{4.34}$$

$$= \begin{pmatrix} x - r_{\mathrm{z}} y + r_{\mathrm{y}} z + t_{\mathrm{x}} \\ r_{\mathrm{z}} x + y - r_{\mathrm{x}} z + t_{\mathrm{y}} \\ -r_{\mathrm{y}} x + r_{\mathrm{x}} y + z + t_{\mathrm{z}} \\ 1 \end{pmatrix} \tag{4.35}$$

ergibt sich für die einzelnen Jacobimatrizen:

$$\mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}}\left(\Delta\mathbf{p}\right)\Big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.36}$$

$$= \begin{pmatrix} \frac{\partial \mathbf{W}_{\mathrm{x}}(\mathbf{x}_0,\Delta\mathbf{p})}{\partial r_{\mathrm{x}}} & \frac{\partial \mathbf{W}_{\mathrm{x}}(\mathbf{x}_0,\Delta\mathbf{p})}{\partial r_{\mathrm{y}}} & \cdot \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{W}_{\mathrm{z}}(\mathbf{x}_0,\Delta\mathbf{p})}{\partial r_{\mathrm{x}}} & & \frac{\partial \mathbf{W}_{\mathrm{z}}(\mathbf{x}_0,\Delta\mathbf{p})}{\partial t_{\mathrm{z}}} \end{pmatrix}\Bigg|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.37}$$

$$= \begin{pmatrix} 0 & \mathrm{z}_0 & -\mathrm{y}_0 & 1 & 0 & 0 \\ -\mathrm{z}_0 & 0 & \mathrm{x}_0 & 0 & 1 & 0 \\ \mathrm{y}_0 & -\mathrm{x}_0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.38}$$

und

$$\mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}}\left(\mathbf{x}=\mathbf{W}(\mathbf{x}_0,\Delta\mathbf{p})\right)\Big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.39}$$

$$= \begin{pmatrix} \frac{\partial \mathbf{W}_{\mathrm{x}}(\mathbf{x},\mathbf{p}_0)}{\partial x} & \frac{\partial \mathbf{W}_{\mathrm{x}}(\mathbf{x},\mathbf{p}_0)}{\partial y} & \frac{\partial \mathbf{W}_{\mathrm{x}}(\mathbf{x},\mathbf{p}_0)}{\partial z} \\ \frac{\partial \mathbf{W}_{\mathrm{y}}(\mathbf{x},\mathbf{p}_0)}{\partial x} & \frac{\partial \mathbf{W}_{\mathrm{y}}(\mathbf{x},\mathbf{p}_0)}{\partial y} & \frac{\partial \mathbf{W}_{\mathrm{y}}(\mathbf{x},\mathbf{p}_0)}{\partial z} \\ \frac{\partial \mathbf{W}_{\mathrm{z}}(\mathbf{x},\mathbf{p}_0)}{\partial x} & \frac{\partial \mathbf{W}_{\mathrm{z}}(\mathbf{x},\mathbf{p}_0)}{\partial y} & \frac{\partial \mathbf{W}_{\mathrm{z}}(\mathbf{x},\mathbf{p}_0)}{\partial z} \end{pmatrix} \left(\mathbf{W}(\mathbf{x}_0,\Delta\mathbf{p})\right)\Big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.40}$$

$$= \begin{pmatrix} 1 & -r_{\mathrm{z}} & r_{\mathrm{y}} \\ r_{\mathrm{z}} & 1 & r_{\mathrm{x}} \\ -r_{\mathrm{y}} & r_{\mathrm{x}} & 1 \end{pmatrix} \left(\mathbf{x}=\mathbf{W}(\mathbf{x}_0,\Delta\mathbf{p})\right)\Big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.41}$$

$$= \begin{pmatrix} 1 & -r_{\mathrm{z}_0} & r_{\mathrm{y}_0} \\ r_{\mathrm{z}_0} & 1 & r_{\mathrm{x}_0} \\ -r_{\mathrm{y}_0} & r_{\mathrm{x}_0} & 1 \end{pmatrix}. \tag{4.42}$$

Man erkennt, dass die Jacobimatrizen nicht von $\mathbf{p}$ bzw. $\mathbf{x}$ abhängen, also konstant in diesen Variablen sind, und somit die Auswertung für $\mathbf{p}$ und $\mathbf{x}$ an den Stellen $\Delta\mathbf{p}$ und $\mathbf{W}(\mathbf{x}_0,\Delta\mathbf{p})$ entfällt.

Die dritte Jacobimatrix berechnet sich aus

$$\mathbf{J_P}\Big(\mathbf{W}\big(\mathbf{W}(\mathbf{x}_0, \Delta\mathbf{p}), \mathbf{p}_0\big)\Big)\Big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.43}$$

$$= \begin{pmatrix} \frac{\partial \mathbf{P}_{\mathrm{u}}(\mathbf{x})}{\partial x} & \frac{\partial \mathbf{P}_{\mathrm{u}}(\mathbf{x})}{\partial y} & \frac{\partial \mathbf{P}_{\mathrm{u}}(\mathbf{x})}{\partial z} \\ \frac{\partial \mathbf{P}_{\mathrm{v}}(\mathbf{x})}{\partial x} & \frac{\partial \mathbf{P}_{\mathrm{v}}(\mathbf{x})}{\partial y} & \frac{\partial \mathbf{P}_{\mathrm{v}}(\mathbf{x})}{\partial z} \end{pmatrix} \Big(\mathbf{W}\big(\mathbf{W}(\mathbf{x}_0, \Delta\mathbf{p}), \mathbf{p}_0\big)\Big)\Big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.44}$$

$$= \begin{pmatrix} \frac{Sf_{\mathrm{x}}}{z} & 0 & -\frac{Sf_{\mathrm{x}}x}{z^2} \\ 0 & \frac{Sf_{\mathrm{y}}}{z} & -\frac{Sf_{\mathrm{y}}y}{z^2} \end{pmatrix} \Big(\mathbf{W}\big(\mathbf{W}(\mathbf{x}_0, \Delta\mathbf{p}), \mathbf{p}_0\big)\Big)\Big|_{\Delta\mathbf{p}=\mathbf{0}} . \tag{4.45}$$

Es gilt für den Punkt der Auswertung

$$\mathbf{x}_{\mathbf{J_P}} = \Big(\mathbf{W}\big(\mathbf{W}(\mathbf{x}_0, \Delta\mathbf{p}), \mathbf{p}_0\big)\Big)\Big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.46}$$

mit

$$\mathbf{W}(\mathbf{x}_0, \Delta\mathbf{p})|_{\Delta\mathbf{p}=\mathbf{0}} = \begin{pmatrix} x_0 - \Delta r_{\mathrm{z}} y_0 + \Delta r_{\mathrm{y}} z_0 + \Delta t_{\mathrm{x}} \\ \Delta r_{\mathrm{z}} x_0 + y_0 - \Delta r_{\mathrm{x}} z_0 + \Delta t_{\mathrm{y}} \\ -\Delta r_{\mathrm{y}} x_0 + \Delta r_{\mathrm{x}} y_0 + z_0 + \Delta t_{\mathrm{z}} \\ 1 \end{pmatrix}\Bigg|_{\Delta\mathbf{p}=0} \tag{4.47}$$

$$= \mathbf{x}_0 , \tag{4.48}$$

was dem Identitätswarp entspricht, und somit

$$\mathbf{x}_{\mathbf{J_P}} = \begin{pmatrix} x_0 - r_{\mathrm{z}_0} y_0 + r_{\mathrm{y}_0} z_0 + t_{\mathrm{x}_0} \\ r_{\mathrm{z}_0} x_0 + y_0 - r_{\mathrm{x}_0} z_0 + t_{\mathrm{y}_0} \\ -r_{\mathrm{y}_0} x_0 + r_{\mathrm{x}_0} y_0 + z_0 + t_{\mathrm{z}_0} \\ 1 \end{pmatrix} . \tag{4.49}$$

Damit lässt sich $\mathbf{J_P}$ schreiben als

$$\mathbf{J_P}\Big(\mathbf{W}\big(\mathbf{W}(\mathbf{x}_0, \Delta\mathbf{p}), \mathbf{p}_0\big)\Big)\Big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.50}$$

$$= \begin{pmatrix} \frac{Sf_{\mathrm{x}}}{z} & 0 & -\frac{Sf_{\mathrm{x}}x}{z^2} \\ 0 & \frac{Sf_{\mathrm{y}}}{z} & -\frac{Sf_{\mathrm{y}}y}{z^2} \end{pmatrix} \big(\mathbf{x}_{\mathbf{J_P}}\big) . \tag{4.51}$$

Einsetzen führt auf einen sehr unhandlichen Term für $\mathbf{J_P}$. Wertet man hingegen

$$\mathbf{J_P}\Big(\mathbf{W}\big(\mathbf{W}(\mathbf{x}_0, \Delta\mathbf{p}), \mathbf{p}_0\big)\Big)\Big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.52}$$

für $\mathbf{p}_0 = \mathbf{0}$ aus (dabei ist es egal, ob man die Schritte 4.46–4.50 zuerst ausführt oder direkt $\mathbf{p}_0 = \mathbf{0}$ einsetzt), so erhält man für das Produkt der drei Jacobimatrizen

$$\mathbf{J_P} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}} \tag{4.53}$$

$$= \begin{pmatrix} \frac{fS_\mathrm{x}}{z_0} & 0 & -\frac{fS_\mathrm{x}x_0}{z_0^2} \\ 0 & \frac{fS_\mathrm{y}}{z_0} & -\frac{fS_\mathrm{y}y_0}{z_0^2} \end{pmatrix} \begin{pmatrix} 1 & -r_{\mathrm{z}_0} & r_{\mathrm{y}_0} \\ r_{\mathrm{z}_0} & 1 & r_{\mathrm{x}_0} \\ -r_{\mathrm{y}_0} & r_{\mathrm{x}_0} & 1 \end{pmatrix} \begin{pmatrix} 0 & z_0 & -y_0 & 1 & 0 & 0 \\ -z_0 & 0 & x_0 & 0 & 1 & 0 \\ y_0 & -x_0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.54}$$

Der Term ist erneut sehr unhandlich. Auswerten von $\mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}}$ an $\mathbf{p}_0 = \mathbf{0}$ liefert die Einheitsmatrix $\mathbf{I}_{3\times3}$. Damit ergibt sich

$$\mathbf{J_P} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}} \tag{4.55}$$

$$= \begin{pmatrix} \frac{Sf_\mathrm{x}}{z_0} & 0 & -\frac{Sf_\mathrm{x}x_0}{z_0^2} \\ 0 & \frac{Sf_\mathrm{y}}{z_0} & -\frac{Sf_\mathrm{y}y_0}{z_0^2} \end{pmatrix} \begin{pmatrix} 0 & z_0 & -y_0 & 1 & 0 & 0 \\ -z_0 & 0 & x_0 & 0 & 1 & 0 \\ y_0 & -x_0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.56}$$

$$= \begin{pmatrix} -Sf_\mathrm{x}\frac{x_0y_0}{z_0^2} & Sf_\mathrm{x}(1+\frac{x_0^2}{z_0^2}) & -Sf_\mathrm{x}\frac{y_0}{z_0} & \frac{Sf_\mathrm{x}}{z_0} & 0 & -Sf_\mathrm{x}\frac{x_0}{z_0^2} \\ -Sf_\mathrm{y}(1+\frac{y_0^2}{z_0^2}) & Sf_\mathrm{y}(\frac{x_0y_0}{z_0^2}) & -Sf_\mathrm{y}\frac{x_0}{z_0} & 0 & \frac{Sf_\mathrm{y}}{z_0} & -Sf_\mathrm{y}\frac{y_0}{z_0^2} \end{pmatrix} \tag{4.57}$$

$$= \begin{pmatrix} Sf_\mathrm{x} & 0 \\ 0 & Sf_\mathrm{y} \end{pmatrix} \begin{pmatrix} -\frac{x_0y_0}{z_0^2} & 1+\frac{x_0^2}{z_0^2} & -\frac{y_0}{z_0} & \frac{1}{z_0} & 0 & -\frac{x_0}{z_0^2} \\ -1-\frac{y_0^2}{z_0^2} & \frac{x_0y_0}{z_0^2} & -\frac{x_0}{z_0} & 0 & \frac{1}{z_0} & -\frac{y_0}{z_0^2} \end{pmatrix}. \tag{4.58}$$

Gleichung (4.58) ist gerade die Gleichung, die in den Arbeiten [Xia+03], [SKK08] und [VSG12] (ohne Herleitung) verwendet wurde.

Der erste Term in Glg. (4.30) beinhaltet die Jacobimatrix des Eingangsbildes. Der Term ergibt sich zu:

$$\mathbf{J}_g\left(\mathbf{P}\left(\mathbf{W}\left(\mathbf{W}(\mathbf{x}_0,\Delta\mathbf{p}),\mathbf{p}_0\right)\right)\right)\big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.59}$$

$$= \left(\frac{\partial g(\mathbf{u})}{\partial u}\frac{\partial g(\mathbf{u})}{\partial v}\right)\left(\mathbf{P}\left(\mathbf{W}\left(\mathbf{W}(\mathbf{x}_0,\Delta\mathbf{p}),\mathbf{p}_0\right)\right)\right)\big|_{\Delta\mathbf{p}=\mathbf{0}}\,. \tag{4.60}$$

Der Term lässt sich so interpretieren, dass das Gradientenbild des aktuellen Grauwertbildes g^N berechnet wird und dieses dann an den Koordinaten $\mathbf{u}$ an der Stelle

$$\mathbf{u} = \mathbf{P}\left(\mathbf{W}\left(\mathbf{W}(\mathbf{x}_0,\Delta\mathbf{p}),\mathbf{p}_0\right)\right) \tag{4.61}$$

am Punkt $\Delta\mathbf{p} = \mathbf{0}$ ausgewertet wird (an diesen Koordinaten wird das Gradientenbild abgetastet):

$$\mathbf{J}_g\left(\mathbf{P}\left(\mathbf{W}\left(\mathbf{x}_0,\mathbf{p}_0\right)\right)\right)\big|_{\Delta\mathbf{p}=\mathbf{0}} = \nabla g\left(\mathbf{P}\left(\mathbf{W}(\mathbf{x}_0,\mathbf{p}_0)\right)\right)\,. \tag{4.62}$$

4.4.3 Herleitung der Warpkomposition (Glg. (4.15))

In diesem Abschnitt sollen die Schritte der Herleitung für Glg. (4.15) ausführlich und mit vollständiger Notation zum besseren Verständnis des oben beschriebenen Algorithmus dargelegt werden (vgl. [BM01]), wobei sich möglichst nah ab der ursprünglichen Notation gehalten werden soll. Hintergrund ist das Zeigen der Äquivalenz des Updateschrittes für den FA- sowie den FC-Algorithmus.

Der Updateschritt des Warps für den FC-Algorithmus

$$\mathbf{W}\left(\mathbf{x},\mathbf{p}\right)\circ\mathbf{W}\left(\mathbf{x},\Delta\mathbf{p}\right) = \mathbf{W}\left(\mathbf{W}\left(\mathbf{x},\Delta\mathbf{p}\right),\mathbf{p}\right) \tag{4.63}$$

wird unter der Voraussetzung durchgeführt, dass obige Gleichung bis zur ersten Ordnung in $\Delta\mathbf{p}$ gilt.

Taylorapproximation erster Ordnung in $\Delta\mathbf{p}$ an der Stelle $\Delta\mathbf{p} = \mathbf{0}$ für den inneren Term in Glg. (4.63) ergibt:

$$\mathbf{W}\left(\mathbf{x},\Delta\mathbf{p}\right) \approx \mathbf{W}(\mathbf{x},\mathbf{0}) + \left.\frac{\partial\mathbf{W}(\mathbf{x},\Delta\mathbf{p})}{\partial\Delta\mathbf{p}}\right|_{\Delta\mathbf{p}=\mathbf{0}}\cdot\Delta\mathbf{p} \tag{4.64}$$

$$= \mathbf{W}(\mathbf{x},\mathbf{0}) + \mathbf{J}_{\mathbf{W}_\mathbf{x}}\big|_{\Delta\mathbf{p}=\mathbf{0}}\cdot\Delta\mathbf{p}\,. \tag{4.65}$$

Einsetzen von Glg. (4.65) in Glg. (4.63) ergibt:

$$\mathbf{W}(\mathbf{W}(\mathbf{x},\Delta\mathbf{p}),\mathbf{p}) = \mathbf{W}\left(\mathbf{x} + \mathbf{J}_{\mathbf{W}_\mathbf{x}}\big|_{\Delta\mathbf{p}=\mathbf{0}} \cdot \Delta\mathbf{p}, \mathbf{p}\right)\Big|_{\Delta\mathbf{p}=\mathbf{0}} . \tag{4.66}$$

Eine erneute Taylorapproximation in $\Delta\mathbf{p}$ an der Stelle $\Delta\mathbf{p} = \mathbf{0}$ ergibt:

$$\begin{aligned}\mathbf{W}(\mathbf{W}(\mathbf{x},\Delta\mathbf{p}),\mathbf{p}) &= \mathbf{W}\left(\mathbf{x} + \mathbf{J}_{\mathbf{W}_\mathbf{x}}\big|_{\Delta\mathbf{p}=\mathbf{0}} \cdot \Delta\mathbf{p}, \mathbf{p}\right)\Big|_{\Delta\mathbf{p}=\mathbf{0}} \\ &\quad + \frac{\partial}{\partial\Delta\mathbf{p}}\left(\mathbf{W}\left(\mathbf{x} + \mathbf{J}_{\mathbf{W}_\mathbf{x}}\big|_{\Delta\mathbf{p}=\mathbf{0}} \cdot \Delta\mathbf{p}, \mathbf{p}\right)\Big|_{\Delta\mathbf{p}=\mathbf{0}}\right) \\ &= \mathbf{W}(\mathbf{x},\mathbf{p}) \\ &\quad + \frac{\partial}{\partial\Delta\mathbf{p}}\left(\mathbf{W}\left(\mathbf{x} + \frac{\partial\mathbf{W}(\mathbf{x},\Delta\mathbf{p})}{\partial\Delta\mathbf{p}} \cdot \Delta\mathbf{p}, \mathbf{p}\right)\right) \\ &= \mathbf{W}(\mathbf{x},\mathbf{p}) \\ &\quad + \frac{\partial\mathbf{W}(\mathbf{x},\mathbf{p})}{\partial\mathbf{x}} \cdot \frac{\partial}{\partial\Delta\mathbf{p}}\left(\mathbf{x} + \frac{\partial\mathbf{W}(\mathbf{x},\Delta\mathbf{p})}{\partial\Delta\mathbf{p}} \cdot \Delta\mathbf{p}, \mathbf{p}\right).\end{aligned} \tag{4.67}$$

Nach der ersten Zeile wurde aus Gründen der Übersicht die Notation für die Auswertung an der Stelle $\Delta\mathbf{p} = \mathbf{0}$ weggelassen. In der letzten Zeile ist für die äußere Ableitung nach $\mathbf{x}$ der Term mit der Klammer mit der partiellen Ableitung nach $\Delta\mathbf{p}$ ausgewertet an $\Delta\mathbf{p} = \mathbf{0}$ nicht von $\Delta\mathbf{p}$ abhängig. Der Ausdruck in der Klammer (innere Funktion) hat die Form: $(a + b \cdot \Delta\mathbf{p}, \mathbf{p})$. Somit ergibt sich:

$$\mathbf{W}(\mathbf{W}(\mathbf{x},\Delta\mathbf{p}),\mathbf{p}) = \mathbf{W}(\mathbf{x},\mathbf{p}) + \frac{\partial\mathbf{W}(\mathbf{x},\mathbf{p})}{\partial\mathbf{x}}\frac{\partial\mathbf{W}(\mathbf{x},\Delta\mathbf{p})}{\partial\Delta\mathbf{p}} , \tag{4.68}$$

ausgewertet an $\Delta\mathbf{p} = \mathbf{0}$ (siehe Glg. (14) in [BM01]).

Das Warp-Update des Terms für den Lucas-Kanade-Algorithmus

$$\mathbf{W}(\mathbf{x},\mathbf{p} + \Delta\mathbf{p}) \tag{4.69}$$

ergibt sich nach Taylorapproximation erster Ordnung zu

$$\mathbf{W}(\mathbf{x},\mathbf{p} + \Delta\mathbf{p}) = \mathbf{W}(\mathbf{x},\mathbf{p}) + \frac{\partial\mathbf{W}(\mathbf{x},\mathbf{p})}{\partial\Delta\mathbf{p}} . \tag{4.70}$$

Die partielle Ableitung nach $\Delta\mathbf{p}$ wird beim Lucas-Kanade-Algorithmus an der Stelle $\mathbf{p}$ ausgewertet, womit die hier vereinfacht als partielle Ableitung geschriebene Jacobimatrix abhängig vom Parametervektor ist. In [BM01] wird gezeigt, dass Glg. (4.68) und Glg. (4.70) das gleiche Optimum besitzen, womit das Warp-Update des FC-Algorithmus dem FA-Algorithmus bis zur ersten Ordnung in $\Delta\mathbf{p}$ äquivalent gesetzt wird.

4.4.4 Zweidimensionaler Fall ohne Projektion mit affinem Warp – *Forward Compositional*-Algorithmus

Es soll beispielhaft der zweidimensionale Fall mit einem affinen Warp

$$\mathbf{W}_{\mathrm{A}}\left(\mathbf{u}, \mathbf{p}_{\mathrm{A}}\right) = \begin{pmatrix} (1+p_1)u & +p_3 v & +p_5 \\ p_2 u & +(1+p_4)v & +p_6 \end{pmatrix} \tag{4.71}$$

mit den sechs affinen Parametern $p_1, p_2, p_3, p_4, p_5, p_6$ betrachtet werden. Für die Taylorapproximation in Glg. (4.23) gilt dann:

$$\frac{\partial f}{\partial \Delta\mathbf{p}}(\mathbf{0}) = \mathbf{J}_g\left(\mathbf{W}\big(\mathbf{W}(\mathbf{u}, \Delta\mathbf{p}), \mathbf{p}\big)\right) \tag{4.72}$$

$$\cdot\, \mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}}\left(\mathbf{W}(\mathbf{u}, \Delta\mathbf{p})\right) \tag{4.73}$$

$$\cdot\, \mathbf{J}_{\mathbf{W}_{\mathbf{u}_0}}\left(\Delta\mathbf{p}\right), \tag{4.74}$$

ausgewertet am Tripel $\Delta\mathbf{p} = \mathbf{0}, \mathbf{p} = \mathbf{p}_0, \mathbf{u} = \mathbf{u}_0$, wobei der Einfachheit halber der Index A, welcher den affinen Warp andeutet, weggelassen wird: $\mathbf{p}_{\mathrm{A}} = \mathbf{p}$. Die Jacobimatrizen ergeben sich zu:

$$\mathbf{J}_{\mathbf{W}_{\mathbf{u}_0}}(\Delta\mathbf{p}) \tag{4.75}$$

$$= \begin{pmatrix} \frac{\partial \mathbf{W}_{\mathrm{u}}(\mathbf{u}_0, \Delta\mathbf{p})}{\partial p1} & \frac{\partial \mathbf{W}_{\mathrm{u}}(\mathbf{u}_0, \Delta\mathbf{p})}{\partial p2} & \cdots \frac{\partial \mathbf{W}_{\mathrm{u}}(\mathbf{u}_0, \Delta\mathbf{p})}{\partial p6} \\ \frac{\partial \mathbf{W}_{\mathrm{v}}(\mathbf{u}_0, \Delta\mathbf{p})}{\partial p1} & \frac{\partial \mathbf{W}_{\mathrm{v}}(\mathbf{u}_0, \Delta\mathbf{p})}{\partial p2} & \cdots \frac{\partial \mathbf{W}_{\mathrm{v}}(\mathbf{u}_0, \Delta\mathbf{p})}{\partial p6} \end{pmatrix}\Bigg|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.76}$$

$$= \begin{pmatrix} u & 0 & v & 0 & 1 & 0 \\ 0 & u & 0 & v & 0 & 1 \end{pmatrix}. \tag{4.77}$$

Für das Gradientenbild ergibt sich

$$\mathbf{J}_g\left(\mathbf{W}\big(\mathbf{W}(\mathbf{u}, \mathbf{0}), \mathbf{p}\big)\right) = \nabla g\left(\mathbf{W}\left(\mathbf{u}_0, \mathbf{p}_0\right)\right), \tag{4.78}$$

wobei wieder der Identitätswarp ausgenutzt wurde. Man erkennt die fehlende Abhängigkeit von der aktuellen inkrementellen Posenänderung und damit den Vorteil des FC-Algorithmus. Als Vergleich dient die Herleitung des Lukas-Kanade-Algorithmus.

4.4.5 Herleitung der Taylor-Approximation des Lukas-Kanade-Algorithmus für den dreidimensionalen optischen Fluss unter einer Lochkameraabbildung

Für die Taylorerweiterung bis zur ersten Ordnung in $\Delta\mathbf{p}$ um den Punkt $\Delta\mathbf{p} = \mathbf{0}$ ergibt sich für den Term $g^N\left(\mathbf{P}\left(\mathbf{W}\left(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p}\right)\right)\right)$ aus Glg. (4.16) für die Ableitung der Taylorapproximation an $\Delta\mathbf{p} = \mathbf{0}$:

$$\frac{\partial f}{\partial \Delta\mathbf{p}}(\mathbf{0}) = \mathbf{J}_g\left(\mathbf{P}\left(\mathbf{W}(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p})\right)\right)\big|_{\Delta\mathbf{p}=\mathbf{0},\mathbf{p}=\mathbf{p}_0,\mathbf{x}=\mathbf{x}_0} \tag{4.79}$$

$$\cdot\, \mathbf{J}_\mathbf{P}\left(\mathbf{W}(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p})\right)\big|_{\Delta\mathbf{p}=\mathbf{0},\mathbf{p}_0,\mathbf{x}_0} \tag{4.80}$$

$$\cdot\, \mathbf{J}_\mathbf{W}\left(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p}\right)\big|_{\Delta\mathbf{p}=\mathbf{0},\mathbf{p}_0,\mathbf{x}_0} \tag{4.81}$$

$$\frac{\partial f}{\partial \Delta\mathbf{p}}(\mathbf{0}) = \mathbf{J}_g\left(\mathbf{P}\left(\mathbf{W}(\mathbf{x}_0, \mathbf{p}_0 + \Delta\mathbf{p})\right)\right)\big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.82}$$

$$\cdot\, \mathbf{J}_\mathbf{P}\left(\mathbf{W}(\mathbf{x}_0, \mathbf{p}_0 + \Delta\mathbf{p})\right)\big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.83}$$

$$\cdot\, \mathbf{J}_\mathbf{W}\left(\mathbf{x}_0, \mathbf{p}_0 + \Delta\mathbf{p}\right)\big|_{\Delta\mathbf{p}=\mathbf{0}}\,. \tag{4.84}$$

Es ergibt sich dann:

$$\mathbf{J}_{\mathbf{W}}\big(\mathbf{x}_0, \mathbf{p}+\Delta\mathbf{p}\big)\big|_{\Delta\mathbf{p}=\mathbf{0}} = \mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}}\big(\mathbf{p}=\mathbf{p}_0+\Delta\mathbf{p}\big)\Big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.85}$$

$$= \begin{pmatrix} \frac{\partial \mathbf{W}_{\mathrm{x}}(\mathbf{x}_0,\mathbf{p})}{\partial r_{\mathrm{x}}} & \frac{\partial \mathbf{W}_{\mathrm{x}}(\mathbf{x}_0,\mathbf{p})}{\partial r_{\mathrm{y}}} & \cdots \\ \vdots & \ddots & \\ \frac{\partial \mathbf{W}_{\mathrm{z}}(\mathbf{x}_0,\mathbf{p})}{\partial r_{\mathrm{x}}} & & \frac{\partial \mathbf{W}_{\mathrm{z}}(\mathbf{x}_0,\mathbf{p})}{\partial t_{\mathrm{z}}} \end{pmatrix} \big(\mathbf{p}_0+\Delta\mathbf{p}\big)\big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.86}$$

$$= \begin{pmatrix} 0 & z_0 & -y_0 & 1 & 0 & 0 \\ -z_0 & 0 & x_0 & 0 & 1 & 0 \\ y_0 & -x_0 & 0 & 0 & 0 & 1 \end{pmatrix} \big(\mathbf{p}_0+\Delta\mathbf{p}\big)\big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.87}$$

$$= \begin{pmatrix} 0 & z_0 & -y_0 & 1 & 0 & 0 \\ -z_0 & 0 & x_0 & 0 & 1 & 0 \\ y_0 & -x_0 & 0 & 0 & 0 & 1 \end{pmatrix} (\mathbf{p}_0)\ . \tag{4.88}$$

Bei Vergleich der Jacobimatrix der innersten Funktion beim Lukas-Kanade-Algorithmus zeigt sich, dass der Term jetzt tatsächlich nicht an $\mathbf{p} = \mathbf{0}$, sondern an $\mathbf{p}_0$ ausgewertet wird und damit grundsätzlich von der aktuellen Pose abhängt. Im Fall der hier gewählten Warpingfunktion fallen durch die Ableitungen der Jacobimatrix allerdings alle von $\mathbf{p}$ abhängigen Terme weg, insbesondere unter der in Glg. (4.54) bis Glg. (4.58) getroffenen Vereinfachung. Dies gilt **nicht allgemein** und der Term kann für andere Warps abhängig von $\mathbf{p}$ sein (beispielsweise bei einer Homographie, siehe Anhang in [BM04]).

Für die Jacobimatrix der Projektion

$$\mathbf{J}_{\mathbf{P}}\left(\mathbf{W}(\mathbf{x}, \mathbf{p}+\Delta\mathbf{p})\right)|_{\Delta\mathbf{p}=\mathbf{0},\mathbf{p}_0,\mathbf{x}_0} \tag{4.89}$$

ergibt sich:

$$\mathbf{J}_{\mathbf{P}}\big(\mathbf{W}(\mathbf{x}_0, \mathbf{p}_0+\Delta\mathbf{p})\big)\big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.90}$$

$$= \begin{pmatrix} \frac{\partial \mathbf{P}_{\mathrm{u}}(\mathbf{x})}{\partial x} & \frac{\partial \mathbf{P}_{\mathrm{u}}(\mathbf{x})}{\partial y} & \frac{\partial \mathbf{P}_{\mathrm{u}}(\mathbf{x})}{\partial z} \\ \frac{\partial \mathbf{P}_{\mathrm{v}}(\mathbf{x})}{\partial x} & \frac{\partial \mathbf{P}_{\mathrm{v}}(\mathbf{x})}{\partial y} & \frac{\partial \mathbf{P}_{\mathrm{v}}(\mathbf{x})}{\partial z} \end{pmatrix} \big(\mathbf{W}(\mathbf{x}_0, \mathbf{p}_0+\Delta\mathbf{p})\big)\big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.91}$$

$$= \begin{pmatrix} \frac{Sf_{\mathrm{x}}}{z} & 0 & -\frac{Sf_{\mathrm{x}}x}{z^2} \\ 0 & \frac{Sf_{\mathrm{y}}}{z} & -\frac{Sf_{\mathrm{y}}y}{z^2} \end{pmatrix} \big(\mathbf{W}(\mathbf{x}_0, \mathbf{p}_0+\Delta\mathbf{p})\big)\big|_{\Delta\mathbf{p}=\mathbf{0}}\ . \tag{4.92}$$

Mit den Punkten

$$\mathbf{x}_{\mathbf{J_P}} = \mathbf{W}(\mathbf{x}_0, \mathbf{p}_0 + \Delta\mathbf{p})|_{\Delta\mathbf{p}=\mathbf{0}} = \begin{pmatrix} x_0 - r_{z_0} y_0 + r_{y_0} z_0 + t_{x_0} \\ r_{z_0} x_0 + y_0 - r_{x_0} z_0 + t_{y_0} \\ -r_{y_0} x_0 + r_{x_0} y_0 + z_0 + t_{z_0} \\ 1 \end{pmatrix} \tag{4.93}$$

ergibt sich der gleiche Term für $\mathbf{J_P}$ wie beim *Forward Compositional*-Algorithmus. Das Produkt $\mathbf{J_P} \cdot \mathbf{J_W}$ wird dann mit $\mathbf{p}_0 = \mathbf{0}$ in $\mathbf{J_P}$ zu

$$\mathbf{J_P} \cdot \mathbf{J_W} \tag{4.94}$$

$$= \begin{pmatrix} Sf_x & 0 \\ 0 & Sf_y \end{pmatrix} \begin{pmatrix} -\frac{x_0 y_0}{z_0^2} & 1 + \frac{x_0^2}{z_0^2} & -\frac{y_0}{z_0} & \frac{1}{z_0} & 0 & -\frac{x_0}{z_0^2} \\ -1 + \frac{y_0^2}{z_0^2} & \frac{x_0 y_0}{z_0^2} & -\frac{x_0}{z_0} & 0 & \frac{1}{z_0} & -\frac{y_0}{z_0^2} \end{pmatrix} . \tag{4.95}$$

Der erste Term ergibt sich weiterhin zu:

$$\mathbf{J}_g \left(\mathbf{P}\big(\mathbf{W}(\mathbf{x}_0, \mathbf{p}_0 + \Delta\mathbf{p})\big)\right)\big|_{\Delta\mathbf{p}=\mathbf{0}} = \nabla g \left(\mathbf{P}\big(\mathbf{W}(\mathbf{x}_0, \mathbf{p}_0)\big)\right) . \tag{4.96}$$

Analog zur Vorgehensweise der Herleitung als Verkettung von Funktionen wie in Glg. (4.18) soll weiterhin eine alternative Herleitung aufgezeigt werden. Für die Taylorerweiterung bis zur ersten Ordnung in $\Delta\mathbf{p}$ um den Punkt $\Delta\mathbf{p} = \mathbf{0}$ lässt sich der Term $g^N\Big(\mathbf{P}\big(\mathbf{W}\big(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p}\big)\big)\Big)$ aus Glg. (4.16) mit den Definitionen (4.19) bis (4.22) schreiben als

$$g^N\Big(\mathbf{P}\big(\mathbf{W}\big(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p}\big)\big)\Big) = \Big(g^N \circ \mathbf{P} \circ \mathbf{W}_\mathbf{p} \circ \xi_\mathbf{x}\Big)\,(\Delta\mathbf{p}) \tag{4.97}$$

mit

$$\xi_\mathbf{x} : \Delta\mathbf{p} \mapsto \mathbf{p} + \Delta\mathbf{p} . \tag{4.98}$$

Taylorapproximation liefert dann

$$\Big(g^N \circ \mathbf{P} \circ \mathbf{W}_\mathbf{p} \circ \xi_\mathbf{x}\Big)\,(\Delta\mathbf{p}) \approx \Big(g^N \circ \mathbf{P} \circ \mathbf{W}_\mathbf{p} \circ \xi_\mathbf{x}\Big)\,(\mathbf{0}) \tag{4.99}$$

$$+ \mathbf{J}_g \cdot \mathbf{J_P} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{P}_0}} \cdot \mathbf{J}_{\xi_\mathbf{x}}\,(\mathbf{0}) . \tag{4.100}$$

Da gilt $\mathbf{J}_{\xi_\mathbf{x}} = 1$ ergibt sich

$$g^N\Big(\mathbf{P}\big(\mathbf{W}\big(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p}\big)\big)\Big) \approx g^N\Big(\mathbf{P}\big(\mathbf{W}\left(\mathbf{x}, \mathbf{p}\right)\big)\Big) \tag{4.101}$$

$$+ \mathbf{J}_g \cdot \mathbf{J}_\mathbf{P} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}}(\mathbf{p} + \Delta\mathbf{p})\Big|_{\Delta\mathbf{p}=\mathbf{0}} . \tag{4.102}$$

Der Unterschied zwischen den Methoden zeigt sich in der Implementierung. Zu Beginn der iterativen Berechnung der Pose in einem Frame, ausgehend von der aktuellen Pose $\mathbf{p}$ und den Modellpunkten $\mathbf{x}$, wird sowohl die kumulierte, aktuelle Pose $\mathbf{p}_0 = \mathbf{0}$ als auch $\Delta\mathbf{p} = \mathbf{0}$ initialisiert. Während die Matrix $\mathbf{J}_\mathbf{P} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}}$ für den FA- sowie den FC-Algorithmus in den oben gemachten Herleitungen direkt nur von $\mathbf{x}_0$ abhängen, wird wegen Glg. (4.88) die Matrix aufgrund der indirekten Abhängigkeit von $\mathbf{x}_0$ von $\mathbf{p}_0$ im FA-Algorithmus in jeder Iteration aktualisiert, um die Änderung durch $\Delta\mathbf{p} \rightarrow \Delta\mathbf{p} + \mathbf{p}_0$ in $\mathbf{x}_0$ zu berücksichtigen. Auf diesen Schritt wird im FC-Algorithmus verzichtet, womit die Matrix $\mathbf{J}_\mathbf{P} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}}$ nur einmal pro Frame berechnet werden muss.

4.4.6 Zweidimensionaler Fall ohne Projektion mit affinem Warp – Lucas-Kanade-Algorithmus

Für die Taylorerweiterung bis zur ersten Ordnung in $\Delta\mathbf{p}$ um den Punkt $\Delta\mathbf{p} = \mathbf{0}$ ergibt sich für den Term $g^N\Big(\mathbf{W}\big(\mathbf{u}, \mathbf{p} + \Delta\mathbf{p}\big)\Big)$ im zweidimensionalen Fall mit einem affinen Warp

$$\mathbf{W}_\mathrm{A}\left(\mathbf{u}, \mathbf{p}_\mathrm{A}\right) = \begin{pmatrix} (1+p_1)u & +p_3 v & +p_5 \\ p_2 u & +(1+p_4)v & +p_6 \end{pmatrix} \tag{4.103}$$

der Term

$$\frac{\partial f}{\partial \Delta\mathbf{p}}(\mathbf{0}) = \mathbf{J}_g\left(\mathbf{W}\left(\mathbf{u}_0, \mathbf{p}_0 + \Delta\mathbf{p}\right)\right)\big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.104}$$

$$\cdot \mathbf{J}_{\mathbf{W}_{\mathbf{u}_0}}\left(\mathbf{u}_0, \mathbf{p}_0 + \Delta\mathbf{p}\right)\Big|_{\Delta\mathbf{p}=\mathbf{0}} , \tag{4.105}$$

wobei der Einfachheit halber $\mathbf{p}_\mathrm{A} = \mathbf{p}$ und $\mathbf{W}_\mathrm{A} = \mathbf{W}$ notiert wird.

Die Jacobimatrizen ergeben sich zu:

$$\mathbf{J}_{\mathbf{W}_{\mathbf{u}_0}}(\mathbf{p}+\Delta\mathbf{p}) \tag{4.106}$$

$$= \begin{pmatrix} \frac{\partial \mathbf{W}_{\mathrm{u}}(\mathbf{u}_0,\mathbf{p})}{\partial p1} & \frac{\partial \mathbf{W}_{\mathrm{u}}(\mathbf{u}_0,\mathbf{p})}{\partial p2} & \cdots & \frac{\partial \mathbf{W}_{\mathrm{u}}(\mathbf{u}_0,\mathbf{p})}{\partial p6} \\ \frac{\partial \mathbf{W}_{\mathrm{v}}(\mathbf{u}_0,\mathbf{p})}{\partial p1} & \frac{\partial \mathbf{W}_{\mathrm{v}}(\mathbf{u}_0,\mathbf{p})}{\partial p2} & \cdots & \frac{\partial \mathbf{W}_{\mathrm{v}}(\mathbf{u}_0,\mathbf{p})}{\partial p6} \end{pmatrix} (\mathbf{p}+\Delta\mathbf{p})\big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.107}$$

$$= \begin{pmatrix} u & 0 & v & 0 & 1 & 0 \\ 0 & u & 0 & v & 0 & 1 \end{pmatrix} \tag{4.108}$$

$$\mathbf{J}_{\mathbf{W}_{\mathbf{p}_0+\Delta\mathbf{p}}}(\mathbf{u}_0) \tag{4.109}$$

$$= \begin{pmatrix} \frac{\partial \mathbf{W}_{\mathrm{u}}(\mathbf{u},\mathbf{p}_0)}{\partial u} & \frac{\partial \mathbf{W}_{\mathrm{u}}(\mathbf{u},\mathbf{p}_0)}{\partial v} \\ \frac{\partial \mathbf{W}_{\mathrm{v}}(\mathbf{u},\mathbf{p}_0)}{\partial u} & \frac{\partial \mathbf{W}_{\mathrm{v}}(\mathbf{u},\mathbf{p}_0)}{\partial v} \end{pmatrix} (\mathbf{u}_0)\big|_{\Delta\mathbf{p}=\mathbf{0}} \tag{4.110}$$

$$= \begin{pmatrix} 1+p1 & p3 \\ p2 & 1+p4 \end{pmatrix} (\Delta\mathbf{p})\big|_{\Delta\mathbf{p}=\mathbf{0}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{4.111}$$

Für das Gradientenbild ergibts sich

$$\nabla g\left(\mathbf{W}\left(\mathbf{u}_0,\mathbf{p}_0\right)\right), \tag{4.112}$$

wobei wieder der Identitätswarp ausgenutzt wurde.

4.5 Herleitung des optischen Flusses für den *Inverse Compositional*-Algorithmus mit einer 3D-Starrköperbewegung unter einer Lochkameraabbildung

Schließlich kann unter Kenntnis der oben gemachten Herleitungen der Algorithmus zum Lösen des in dieser Arbeit betrachteten Schätzproblems aufgestellt werden. Nach Linearisierung des Warps im Term

$$\sum_{\mathbf{u}\in\Omega_{\mathrm{Obj}}} \left[g^N\left(\mathbf{P}\left(\mathbf{W}\left(\mathbf{W}(\mathbf{x},\Delta\mathbf{p}),\mathbf{p}\right)\right)\right) - g^0(\mathbf{u})\right]^2$$

in $\Delta\mathbf{p}$ ergibt sich für die Zielfunktion

$$\mathrm{E}(\Delta\mathbf{p}) = \sum_{\mathbf{u}\in\Omega_{\mathrm{Obj}}} \left[g^N\Big(\mathbf{P}\big(\mathbf{W}\big(\mathbf{W}(\mathbf{x},\Delta\mathbf{p}),\mathbf{p}\big)\big)\Big) - g^0(\mathbf{u}) \right]^2 \tag{4.113}$$

$$= \sum_{\mathbf{u}\in\Omega_{\mathrm{Obj}}} \left[g^N\Big(\mathbf{P}\big(\mathbf{W}\left(\mathbf{x},\mathbf{p}\right)\big)\Big) + \nabla g \cdot \mathbf{J}_{\mathbf{P}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}} \cdot \Delta\mathbf{p} - g^0(\mathbf{u}) \right]^2. \tag{4.114}$$

Ableiten nach $\Delta\mathbf{p}$ liefert für $\mathrm{E}(\Delta\mathbf{p})$:

$$\frac{\partial}{\partial\Delta\mathbf{p}}\mathrm{E}(\Delta\mathbf{p}) = 2 \cdot \sum_{\mathbf{u}\in\Omega_{\mathrm{Obj}}} \left(\nabla g \cdot \mathbf{J}_{\mathbf{P}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}} \right)^{\mathrm{T}}$$

$$\cdot \left(\nabla g \cdot \mathbf{J}_{\mathbf{P}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}} \cdot \Delta\mathbf{p} + g^N\Big(\mathbf{P}\big(\mathbf{W}\left(\mathbf{x},\mathbf{p}\right)\big)\Big) - g^0(\mathbf{u}) \right). \tag{4.115}$$

Hier ist zu beachten, dass die Jacobimatrizen alle an der Stelle $\Delta\mathbf{p} = \mathbf{0}$ ausgewertet sind (Taylorapproximation um diesen Punkt) und somit nicht von der gesuchten Variablen abhängen.
Die Terme

$$\begin{aligned}
&\nabla g \cdot \mathbf{J}_{\mathbf{P}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}} \qquad (4.116)\\
&= \begin{pmatrix} \frac{\partial g(\mathbf{u})}{\partial u} & \frac{\partial g(\mathbf{u})}{\partial v} \end{pmatrix} \big(\mathbf{P}\big(\mathbf{W}(\mathbf{x}_0,\mathbf{p}_0)\big)\big)\\
&\cdot \begin{pmatrix} \frac{\partial \mathbf{P}_{\mathrm{u}}(\mathbf{x})}{\partial x} & \frac{\partial \mathbf{P}_{\mathrm{u}}(\mathbf{x})}{\partial y} & \frac{\partial \mathbf{P}_{\mathrm{u}}(\mathbf{x})}{\partial z} \\ \frac{\partial \mathbf{P}_{\mathrm{v}}(\mathbf{x})}{\partial x} & \frac{\partial \mathbf{P}_{\mathrm{v}}(\mathbf{x})}{\partial y} & \frac{\partial \mathbf{P}_{\mathrm{v}}(\mathbf{x})}{\partial z} \end{pmatrix} \big(\mathbf{W}(\mathbf{x}_0,\mathbf{p}_0)\big)\\
&\cdot \begin{pmatrix} \frac{\partial \mathbf{W}_{\mathrm{x}}(\mathbf{x},\mathbf{p}_0)}{\partial x} & \frac{\partial \mathbf{W}_{\mathrm{x}}(\mathbf{x},\mathbf{p}_0)}{\partial y} & \frac{\partial \mathbf{W}_{\mathrm{x}}(\mathbf{x},\mathbf{p}_0)}{\partial z} \\ \frac{\partial \mathbf{W}_{\mathrm{y}}(\mathbf{x},\mathbf{p}_0)}{\partial x} & \frac{\partial \mathbf{W}_{\mathrm{y}}(\mathbf{x},\mathbf{p}_0)}{\partial y} & \frac{\partial \mathbf{W}_{\mathrm{y}}(\mathbf{x},\mathbf{p}_0)}{\partial z} \\ \frac{\partial \mathbf{W}_{\mathrm{z}}(\mathbf{x},\mathbf{p}_0)}{\partial x} & \frac{\partial \mathbf{W}_{\mathrm{z}}(\mathbf{x},\mathbf{p}_0)}{\partial y} & \frac{\partial \mathbf{W}_{\mathrm{z}}(\mathbf{x},\mathbf{p}_0)}{\partial z} \end{pmatrix} \big(\mathbf{x}_0\big)\\
&\cdot \begin{pmatrix} \frac{\partial \mathbf{W}_{\mathrm{x}}(\mathbf{x}_0,\mathbf{p})}{\partial r_{\mathrm{x}}} & \frac{\partial \mathbf{W}_{\mathrm{x}}(\mathbf{x}_0,\mathbf{p})}{\partial r_{\mathrm{y}}} & \cdots \\ \vdots & \ddots & \\ \frac{\partial \mathbf{W}_{\mathrm{z}}(\mathbf{x}_0,\mathbf{p})}{\partial r_{\mathrm{x}}} & & \frac{\partial \mathbf{W}_{\mathrm{z}}(\mathbf{x}_0,\mathbf{p})}{\partial t_{\mathrm{z}}} \end{pmatrix} \big(\mathbf{p}_0\big)\\
&= \begin{pmatrix} \frac{Sf_{\mathrm{x}}}{z} & 0 & -\frac{Sf_{\mathrm{x}}x}{z^2} \\ 0 & \frac{Sf_{\mathrm{y}}}{z} & -\frac{Sf_{\mathrm{y}}y}{z^2} \end{pmatrix} \big(\mathbf{x}_{\mathbf{J}_{\mathbf{P}}}\big) \begin{pmatrix} 1 & -r_{\mathrm{z}} & r_{\mathrm{y}} \\ r_{\mathrm{z}} & 1 & r_{\mathrm{x}} \\ -r_{\mathrm{y}} & r_{\mathrm{x}} & 1 \end{pmatrix} (\mathbf{x}_0) \begin{pmatrix} 0 & z_0 & -y_0 & 1 & 0 & 0 \\ -z_0 & 0 & x_0 & 0 & 1 & 0 \\ y_0 & -x_0 & 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}$$

werden in [BM04] auch *Steepest Descent Images* (Bilder des steilsten Abstiegs) genannt, deren Diskussion in Kap. 4.6 erfolgt und die mit

$$\mathbf{SD}(\mathbf{x}_{\mathbf{J_P}}, \mathbf{p}_0) = \left(\frac{\partial}{\partial \mathbf{p}} \nabla g \, \mathbf{J_P} \, \mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}} \mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}} \right) = \left(\frac{\partial}{\partial \mathbf{p}} \mathrm{Warp} \right) \tag{4.117}$$

$$= \left(\frac{\partial \mathrm{Warp}}{\partial r_\mathrm{x}}, \frac{\partial \mathrm{Warp}}{\partial r_\mathrm{y}}, \frac{\partial \mathrm{Warp}}{\partial r_\mathrm{z}}, \frac{\partial \mathrm{Warp}}{\partial t_\mathrm{x}}, \frac{\partial \mathrm{Warp}}{\partial t_\mathrm{y}}, \frac{\partial \mathrm{Warp}}{\partial t_\mathrm{z}} \right) \tag{4.118}$$

bezeichnet werden sollen. Mit den Bildern des steilsten Abstiegs sowie der Fehlerfunktion

$$g_{\mathrm{Err}}(\mathbf{u}) = -g^N\Big(\mathbf{P}\big(\mathbf{W}\left(\mathbf{x}, \mathbf{p}\right)\big)\Big) + g^0(\mathbf{u}) \tag{4.119}$$

ergibt sich für die Zielfunktion:

$$\sum_{\mathbf{u} \in \Omega_{\mathrm{Obj}}} \big(\mathbf{SD}(\mathbf{x}_{\mathbf{J_P}}, \mathbf{p}_0)\big)^{\mathrm{T}} \cdot \big(\mathbf{SD}(\mathbf{x}_{\mathbf{J_P}}, \mathbf{p}_0) \cdot \Delta\mathbf{p} - g_{\mathrm{Err}}(\mathbf{u})\big) \stackrel{!}{=} \mathbf{0}\,.$$

Auflösen nach $\Delta\mathbf{p}$ und Weglassen des Arguments der Bilder des steilsten Abstiegs führt auf:

$$\sum_{\mathbf{u} \in \Omega_{\mathrm{Obj}}} \Big(\mathbf{SD}^{\mathrm{T}} \cdot \mathbf{SD}\Big) \cdot \Delta\mathbf{p} = \sum_{\mathbf{u} \in g_{\mathrm{Obj}}(\mathbf{u})} \mathbf{SD}^{\mathrm{T}} \cdot g_{\mathrm{Err}}(\mathbf{u})\,. \tag{4.120}$$

Mit der Hessematrix $\mathbf{H}$

$$\mathbf{H} = \sum_{\mathbf{u} \in \Omega_{\mathrm{Obj}}} \mathbf{SD}^{\mathrm{T}} \cdot \mathbf{SD}\,, \tag{4.121}$$

ergibt sich für die inkrementelle Posenänderung

$$\Delta\mathbf{p} = \mathbf{H}^{-1} \cdot \sum_{\mathbf{u} \in \Omega_{\mathrm{Obj}}} \mathbf{SD}^{\mathrm{T}} \cdot g_{\mathrm{Err}}(\mathbf{u})\,. \tag{4.122}$$

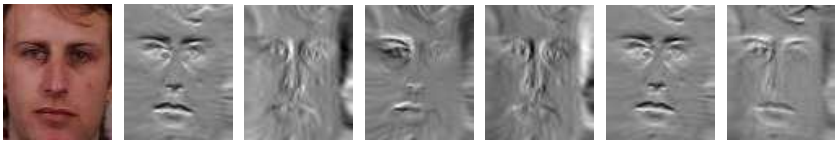
Der $m \times n \times 6$-Tensor $\mathbf{SD}$ beinhaltet hierbei insbesondere die Ableitungen der Warpingfunktion und damit des Bewegungsmodells. Als für jedes Pixel $\mathbf{u} \in \Omega_{\mathrm{Obj}}$ definertes $1 \times 2 \cdot 2 \times 3 \cdot 3 \times 3\dot{3} \times 6$ Produkt $\nabla g \cdot \mathbf{J_P} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{p}_0}} \cdot \mathbf{J}_{\mathbf{W}_{\mathbf{x}_0}}$ stellen sie durch Verknüpfung der Ableitungen

von Projektion und Bewegungsmodell die Sensitivitäten gegenüber der einzelnen Bewegungsrichtungen dar. Die 6×6-Hessematrix kann als empirische Korrelationsmatrix der Bilder des steilsten Abstiegs verstanden werden und spielt somit eine entscheidende Rolle bei der Bestimmung von $\Delta\mathbf{p}$. Die Bilder des steilsten Abstiegs können als zur Lösung des Minimierungsproblems verfolgte Gradienten im 6D-Lösungsraum in Richtung des lokalen Minimums interpretiert werden.

Durch iteratives Lösen der Zielfunktion für inkrementelle Posenänderungen kann dann für jedes Frame das 3D-Objektmodell aktualisiert und somit die Kopfpose durch den sechsdimensionalen Zustandsvektor des Objektes geschätzt werden.

4.6 Bewegungsabhängige Regularisierung

Die durch das Aperturproblem entstehenden Ambiguitäten bei der Berechnung des optischen Flusses treten insbesondere bei starken, kombinierten Dreh- und Translationsbewegungen des Kopfes auf. Das Problem drückt sich mathematisch in einer schlecht konditionierten Hessematrix $\mathbf{H}$ aus, was bei der Invertierung der Hessematrix aufgrund von Singularitäten zu Unstabilitäten im optischen Fluss führen kann. Die dadurch verursachten Ungenauigkeiten bei der Bestimmung der Bewegung in Glg. (4.121), führen schließlich zu einer fehlerhaften Schätzung der Kopfpose [VMP15]. Diese Ambiguitäten zeigen sich anschaulich, wenn man die in der Berechnung der inkrementellen Posenänderungen auftretenden Bilder des steilsten Abstiegs visualisiert. Dies ist in Abbildung 4.1 für den Probanden *jam* aus dem Datensatz in [LSA00] zu sehen.



(a) $g(\mathbf{u})$. (b) $\frac{\partial \mathrm{Warp}}{\partial r_\mathrm{x}}$. (c) $\frac{\partial \mathrm{Warp}}{\partial r_\mathrm{y}}$. (d) $\frac{\partial \mathrm{Warp}}{\partial r_\mathrm{z}}$. (e) $\frac{\partial \mathrm{Warp}}{\partial t_\mathrm{x}}$. (f) $\frac{\partial \mathrm{Warp}}{\partial t_\mathrm{y}}$. (g) $\frac{\partial \mathrm{Warp}}{\partial t_\mathrm{z}}$.

Abbildung 4.1 Originalbild und die Bilder steilsten Abstiegs.

Interpretiert man die Bilder des steilsten Abstiegs als Sensitivitäten der projizierten verketteten Warpingfunktionen bezüglich der 3D-Pose, so sind die als Aperturproblem bekannten Abhängigkeiten zwischen Gieren (r_y) und x-Translation (t_x) sowie Nicken (r_x) und y-Translation (t_y) visuell gut erkennbar.

Innerhalb der Gleichungen des optischen Flusses erkennt man durch das Aperturproblem, dass es zu hohen Quotienten der Einträge der Hessematrix und damit zu einer schlechten Konditionierung kommt (siehe Kap. 4.8.5). Weiterhin spiegeln hohe Werte auf den Nebendiagonalen der Hessematrix Abhängigkeiten der verschiedenen Bewegungsformen wider, welche bereits optisch oben dargelegt wurden.

Ziel ist es daher, durch geeignete Operationen die Konditionszahl zu senken, um bei der Invertierung der Hessematrix insbesondere Instabilitäten, die durch Singularitäten entstehen, zu vermeiden. Dies soll hier durch einen Regularisierungsansatz geschehen, der im folgenden Abschnitt besprochen wird, während der Einfluss der Regularisierung auf die Hessematrix im Anschluss in Kap. 4.8.5 diskutiert wird.

4.6.1 Konventioneller Regularisierungsansatz

In der Literatur wird zur Vermeidung von Singularitäten, welche zu Unstabilitäten bei der Invertierung der Hesse-Matrix führen, das Hinzufügen eines konstanten Regularisierungsparameters λ zur Hessematrix

$$\mathbf{H}_\mathrm{Reg}^\mathrm{konv.} = \mathbf{H} + \Gamma^2 \sum_{\mathbf{u}\in\Omega_\mathrm{Obj}} \left(\frac{\partial \mathrm{Warp}}{\partial \mathbf{p}}\right)^\mathrm{T} \left(\frac{\partial \mathrm{Warp}}{\partial \mathbf{p}}\right), \quad \Gamma = \sqrt{\lambda}\mathbf{I},$$

wobei $\mathbf{I}$ eine 6×6-Einheitsmatrix ist, vorgeschlagen. Hierdurch kann typischerweise eine Senkung der Konditionszahl um bis zu 10^2 erreicht werden [Xia+03]. Durch das Senken der Konditionszahl kann zwar die Stabilität erhöht werden, die Kopfposenschätzung ist allerdings weiterhin unzureichend und führt insbesondere bei Drehbewegungen in die Bildebene zu einem Abbruch des Trackings [VMP15; VP16b]. Da der Fehler der Genauigkeit mit der oben genannten Methode auf bekannten Datenbanken [LSA00] teilweise über 5° für Drehungen aus der Bildebene hinaus beträgt, wird im Folgenden ein Ansatz zu einer alternativen Regularisierung der Hessematrix vorgeschlagen.

4.6.2 Bewegungsabhängiger Regularisierungsansatz

Um den optischen Fluss adaptiv an die vorherrschende Bewegung anzupassen, wird in [VMP15] eine Integration der aktuellen Bewegung in die Regularisierung vorgeschlagen. Durch Definition eines Regularisierungsparametervektors

$$\boldsymbol{\lambda} = \left(\lambda_{r_\mathrm{x}},\, \lambda_{r_\mathrm{y}},\, \lambda_{r_\mathrm{z}},\, \lambda_{t_\mathrm{x}},\, \lambda_{t_\mathrm{y}},\, \lambda_{t_\mathrm{z}}\right)^\mathrm{T} \tag{4.123}$$

kann die regularisierte Hessematrix wie folgt definiert werden:

$$\mathbf{H}_\mathrm{Reg} = \mathbf{H} + \sum_{\mathbf{u}\in\Omega_\mathrm{Obj}} \boldsymbol{\lambda} \left(\frac{\partial \mathrm{Warp}}{\partial \mathbf{p}}\right) \left(\frac{\partial \mathrm{Warp}}{\partial \mathbf{p}}\right)^\mathrm{T} \boldsymbol{\lambda}^\mathrm{T}. \tag{4.124}$$

Mit der Notationen

$$\frac{\partial \mathrm{Warp}}{\partial \mathbf{p}_i} \lambda_{\mathbf{p}_j} \equiv \lambda_{\mathbf{p}_j}^{\mathbf{p}_i}, \quad i,j \in \{r_\mathrm{x}, r_\mathrm{y}, r_\mathrm{z}, t_\mathrm{x}, t_\mathrm{y}, t_\mathrm{z}\}, \tag{4.125}$$

erhält man für den Term in der Summe in Glg. (4.124)

$$\sum_{\mathbf{u}\in g_\mathrm{Obj}(\mathbf{u})} = \begin{pmatrix}
\lambda_{r_\mathrm{x}}^{r_\mathrm{x}}\lambda_{r_\mathrm{x}}^{r_\mathrm{x}} & \lambda_{r_\mathrm{x}}^{r_\mathrm{y}}\lambda_{r_\mathrm{y}}^{r_\mathrm{x}} & \lambda_{r_\mathrm{x}}^{r_\mathrm{z}}\lambda_{r_\mathrm{z}}^{r_\mathrm{x}} & \lambda_{r_\mathrm{x}}^{t_\mathrm{x}}\lambda_{t_\mathrm{x}}^{r_\mathrm{x}} & \lambda_{r_\mathrm{x}}^{t_\mathrm{y}}\lambda_{t_\mathrm{y}}^{r_\mathrm{x}} & \lambda_{r_\mathrm{x}}^{t_\mathrm{z}}\lambda_{t_\mathrm{z}}^{r_\mathrm{x}} \\
\lambda_{r_\mathrm{y}}^{r_\mathrm{x}}\lambda_{r_\mathrm{x}}^{r_\mathrm{y}} & \lambda_{r_\mathrm{y}}^{r_\mathrm{y}}\lambda_{r_\mathrm{y}}^{r_\mathrm{y}} & \lambda_{r_\mathrm{y}}^{r_\mathrm{z}}\lambda_{r_\mathrm{z}}^{r_\mathrm{y}} & \lambda_{r_\mathrm{y}}^{t_\mathrm{x}}\lambda_{t_\mathrm{x}}^{r_\mathrm{y}} & \lambda_{r_\mathrm{y}}^{t_\mathrm{y}}\lambda_{t_\mathrm{y}}^{r_\mathrm{y}} & \lambda_{r_\mathrm{y}}^{t_\mathrm{z}}\lambda_{t_\mathrm{z}}^{r_\mathrm{y}} \\
\lambda_{r_\mathrm{z}}^{r_\mathrm{x}}\lambda_{r_\mathrm{x}}^{r_\mathrm{z}} & \lambda_{r_\mathrm{z}}^{r_\mathrm{y}}\lambda_{r_\mathrm{y}}^{r_\mathrm{z}} & \lambda_{r_\mathrm{z}}^{r_\mathrm{z}}\lambda_{r_\mathrm{z}}^{r_\mathrm{z}} & \lambda_{r_\mathrm{z}}^{t_\mathrm{x}}\lambda_{t_\mathrm{x}}^{r_\mathrm{z}} & \lambda_{r_\mathrm{z}}^{t_\mathrm{y}}\lambda_{t_\mathrm{y}}^{r_\mathrm{z}} & \lambda_{r_\mathrm{z}}^{t_\mathrm{z}}\lambda_{t_\mathrm{z}}^{r_\mathrm{z}} \\
\lambda_{t_\mathrm{x}}^{r_\mathrm{x}}\lambda_{r_\mathrm{x}}^{t_\mathrm{x}} & \lambda_{t_\mathrm{x}}^{r_\mathrm{y}}\lambda_{r_\mathrm{y}}^{t_\mathrm{x}} & \lambda_{t_\mathrm{x}}^{r_\mathrm{z}}\lambda_{r_\mathrm{z}}^{t_\mathrm{x}} & \lambda_{t_\mathrm{x}}^{t_\mathrm{x}}\lambda_{t_\mathrm{x}}^{t_\mathrm{x}} & \lambda_{t_\mathrm{x}}^{t_\mathrm{y}}\lambda_{t_\mathrm{y}}^{t_\mathrm{x}} & \lambda_{t_\mathrm{x}}^{t_\mathrm{z}}\lambda_{t_\mathrm{z}}^{t_\mathrm{x}} \\
\lambda_{t_\mathrm{y}}^{r_\mathrm{x}}\lambda_{r_\mathrm{x}}^{t_\mathrm{y}} & \lambda_{t_\mathrm{y}}^{r_\mathrm{y}}\lambda_{r_\mathrm{y}}^{t_\mathrm{y}} & \lambda_{t_\mathrm{y}}^{r_\mathrm{z}}\lambda_{r_\mathrm{z}}^{t_\mathrm{y}} & \lambda_{t_\mathrm{y}}^{t_\mathrm{x}}\lambda_{t_\mathrm{x}}^{t_\mathrm{y}} & \lambda_{t_\mathrm{y}}^{t_\mathrm{y}}\lambda_{t_\mathrm{y}}^{t_\mathrm{y}} & \lambda_{t_\mathrm{y}}^{t_\mathrm{z}}\lambda_{t_\mathrm{z}}^{t_\mathrm{y}} \\
\lambda_{t_\mathrm{z}}^{r_\mathrm{x}}\lambda_{r_\mathrm{x}}^{t_\mathrm{z}} & \lambda_{t_\mathrm{z}}^{r_\mathrm{y}}\lambda_{r_\mathrm{y}}^{t_\mathrm{z}} & \lambda_{t_\mathrm{z}}^{r_\mathrm{z}}\lambda_{r_\mathrm{z}}^{t_\mathrm{z}} & \lambda_{t_\mathrm{z}}^{t_\mathrm{x}}\lambda_{t_\mathrm{x}}^{t_\mathrm{z}} & \lambda_{t_\mathrm{z}}^{t_\mathrm{y}}\lambda_{t_\mathrm{y}}^{t_\mathrm{z}} & \lambda_{t_\mathrm{z}}^{t_\mathrm{z}}\lambda_{t_\mathrm{z}}^{t_\mathrm{z}}
\end{pmatrix}. \tag{4.126}$$

Durch geschickte Wahl von $\boldsymbol{\lambda}$ ist es somit möglich, die aktuelle Bewegung direkt in die Regularisierung zu integrieren. Es ist wichtig zu bemerken, dass sich durch die 6×6-Matrix $\boldsymbol{\lambda}\boldsymbol{\lambda}^\mathrm{T}$ auf der Hauptdiagonalen direkt die isolierten Bewegungen beeinflussen lassen, während auf den Nebendiagonalen Kreuzterme auftreten, mit denen sich kombinierte Bewegungen manipulieren lassen.

Der folgende Abschnitt beschäftigt sich mit der Frage, wie die Regularisierungsparameter sinnvoll bestimmt werden können.

4.7 Bewegungsadaptive Regularisierungsparameter

Um eine unabhängige Schätzung der Bewegung zu erhalten, wird im verfolgten Ansatz eine perspektivische Transformation (Homographie) zwischen zwei aufeinanderfolgenden Frames für den Bildausschnitt, der den Kopf repräsentiert, bestimmt.

Dazu wird im jedem neuen Frame N basierend auf der letzten Schätzung der Kopfpose und der daraus resultierenden Position des Kopfmodells ein Suchbereich durch Projektion $\mathbf{P}$ des Objektmodells auf den Bildbereich bestimmt. In diesem Bereich wird nun nach stabilen Punkten $\mathbf{u}^* = [u, v, w] = [u, v, 1]$ mit dem SURF-Verfahren [BTG06] gesucht, welche hierzu mit ihrem 128-dimensionalen Merkmalsvektor und ihrer Position sowie dem aktuellen Frame abgespeichert werden. Die Punkte $\mathbf{u}^*_{i^{N-1}}$ und $\mathbf{u}^*_{i^N}$ können nun genutzt werden, um daraus die Homographie zwischen den Frames $N-1$ und N zu bestimmen.

4.7.1 Homographieberechnung

Eine Homographie lässt sich als 3×3-Matrix $\mathbf{M}_\mathrm{H}$ ausdrücken, wobei $\mathbf{M}_\mathrm{H}$ acht Freiheitsgrade besitzt und das neunte Element ein Skalierungsfaktor ist, der aus der Berechnung von $\mathbf{M}_\mathrm{H}$ aus homogenen Koordinaten resultiert. Während eine affine Transformation endliche Punkte auf endliche Punkte (Punkte, die sich im Endlichen schneiden) und unendliche Punkte auf unendliche Punkte abbildet (parallele Geraden), beschreibt eine Homographie eine lineare Abbildung, die eine Transformation vormals paralleler Geraden auf einen endlichen Schnittpunkt erlaubt. Damit lassen sich unter anderem auch perspektivische Transformationen, die bei Drehungen von 2D-Körpern in die Bildebene hinein entstehen, beschreiben. Aufgrund der acht Freiheitsgrade sind für die Bestimmung der Homographie $n \geq 4$ 2D-Punkte notwendig. Für die korrespondierenden Punktepaare $\mathbf{u}^*_{i^N}$ und $\mathbf{u}^*_{i^{N-1}}, i = \{1, \ldots, l\}$, kann dann jeweils zur Berechnung der Homographiematrix $\mathbf{M}_\mathrm{H}$ mit den Zeilenvektoren $\mathbf{h_1}^\mathrm{T}, \mathbf{h_2}^\mathrm{T}, \mathbf{h_3}^\mathrm{T}$ der Zusammenhang

$$\mathbf{u}_{i^N}^* \sim \mathbf{M}_{\mathrm{H}} \mathbf{u}_{i^{N-1}}^* = \begin{pmatrix} \mathbf{h_1}^{\mathrm{T}} \\ \mathbf{h_2}^{\mathrm{T}} \\ \mathbf{h_3}^{\mathrm{T}} \end{pmatrix} \mathbf{u}_{i^{N-1}}^* \tag{4.127}$$

bzw.

$$\frac{u_{i^N}^*}{w_{i^N}^*} = \frac{\mathbf{h_1}^{\mathrm{T}} \mathbf{u}_{i^{N-1}}^*}{\mathbf{h_3}^{\mathrm{T}} \mathbf{u}_{i^{N-1}}^*} \quad \wedge \quad \frac{v_{i^N}^*}{w_{i^N}^*} = \frac{\mathbf{h_2}^{\mathrm{T}} \mathbf{u}_{i^{N-1}}^*}{\mathbf{h_3}^{\mathrm{T}} \mathbf{u}_{i^{N-1}}^*} \tag{4.128}$$

aufgestellt werden. Damit erhält man

$$w_{i^N}^* \mathbf{h_1}^{\mathrm{T}} \mathbf{u}_{i^{N-1}}^* + 0\, \mathbf{h_2}^{\mathrm{T}} - u_{i^N}^* \mathbf{h_3}^{\mathrm{T}} \mathbf{u}_{i^{N-1}}^* = 0 \tag{4.129}$$

$$0\, \mathbf{h_1}^{\mathrm{T}} + w_{i^N}^* \mathbf{h_2}^{\mathrm{T}} \mathbf{x} - v_{i^N}^* \mathbf{h_3}^{\mathrm{T}} \mathbf{u}_{i^{N-1}}^* = 0 \tag{4.130}$$

$$\begin{pmatrix} w_{i^N}^* \mathbf{u}_{i^{N-1}}^{*\,\mathrm{T}} & \mathbf{0} & -u_{i^N}^* \mathbf{u}_{i^{N-1}}^{*\,\mathrm{T}} \\ \mathbf{0} & w_{i^N}^* \mathbf{u}_{i^{N-1}}^{*\,\mathrm{T}} & -v_{i^N}^* \mathbf{u}_{i^{N-1}}^{*\,\mathrm{T}} \end{pmatrix} \begin{pmatrix} \mathbf{h_1} \\ \mathbf{h_2} \\ \mathbf{h_3} \end{pmatrix} = \mathbf{0}\,, \tag{4.131}$$

womit sich dann schließlich für vier Punkte ein 8×9-Gleichungssystem ergibt, das mit Hilfe eines *Least Squares*-Schätzers gelöst werden kann.

4.7.2 Auswahl und Zuordnung der Merkmale

Das Verfolgen der Punkte beginnt ab dem ersten Frame eines Videos und wird kontinuierlich weiter geführt. Für aufeinanderfolgende Frames können dann auf Basis des 128-dimensionalen Merkmalsvektors unter Berücksichtigung des mittleren quadratischen Fehlers zwischen Merkmalsvektoren zwei Punkte $\mathbf{u}_{i^{N-1}}^*$ und $\mathbf{u}_{j^N}^*$ in unterschiedlichen Frames einander zugeordnet werden, wobei $j = i$ einer erfolgreichen Zuordnung entspricht. Abbildung 4.2(a) illustriert anhand des Videos *jim9* aus dem *Boston University Head Pose Dataset* [LSA00] die Zuordnung von Merkmalen.

4.7.2.1 Alter

Bei der Featureauswahl nach dem Alter steigt das Gewicht mit zunehmender Anzahl an Frames, an denen ein erfolgreiches Zuordnen von Merkmalen stattfindet. Hierbei geschieht die Zuordnung nur im binären

Sinne der Erfüllung eines Schwellenwertes, während die Güte der Zuordnung ungeachtet bleibt. Dies ist gut erkennbar in Abb. 4.2(b), wo die Länge der Trajektorien (Alter) mit der Größe der Marker korreliert. Das Charakteristikum Alter gibt somit eine Information über die Stabilität des Merkmals.

(a) Zugeordnete Merkmale. **(b)** Merkmalsauswahl Alter. **(c)** Merkmalsauswahl Metrik. **(d)** Merkmalsauswahl Distanz.

Abbildung 4.2 Frame 120 des Videos *jim9*. Zu sehen sind zugeordnete Merkmale von Frame 119 auf 120 (a), das Alter der Merkmale (b), die Güte der Metrik (c) sowie die größte Distanz der Merkmale (d). Bei dem Alter und der Metrik entsprechen große Rauten einem hohen Gewicht.

4.7.2.2 Metrik

Die Metrik (Abb. 4.2(c)) ergibt sich direkt aus dem mittleren quadratischen Fehler zweier zugeordneter Merkmalsvektoren. Dabei wird das Gewicht stets nur von der Zuordnung von Frame $N-1$ auf N beeinflusst, da eine erfolgreiche Zuordnung in der Vergangenheit keine Auswirkungen auf die Güte der Zuordnung zum aktuellen Frame und damit auch keine sinnvolle Auswirkung auf die Bestimmung der Homographie hat.

4.7.2.3 Distanz

Bei der Gewichtung nach der Distanz werden vier Merkmale aus einer Aufteilung in vier Quadranten des Suchbereichs möglichst so gewählt, dass ihre Abstände zueinander maximal sind. Insbesondere bei Rotationen in die Ebene (perspektivischen Transformationen) spielt die Breite der Verteilung der gewählten Punkte eine wichtige Rolle, da beispielsweise die Wahl von Punkten nahe einer Drehachse in die Bildebene eine starke Unterschätzung dieser Bewegung bei der Homographieberech-

nung zur Folge hat. Abbildung 4.2(d) zeigt beispielhaft die Verteilung der gewählten Merkmale für Frame 120 des Videos *jim9*.

4.7.2.4 Diskussion

Ein Unterschied zwischen der Auswahl nach Alter und Metrik lässt sich gut anhand des hellgrün dargestellten Markers links oberhalb der Lippe des Probanden erklären. Während das Merkmal nach Alter das größte Gewicht besitzt, zeigt sich, dass sich seine Metrik für die Zuordnung von Frame 119 auf das hier dargestellte Frame 120 nicht abhebt und auch nicht als eines der vier stärksten Merkmale gewählt wurde. Bei der Auswahl nach Distanz zeigt Abb. 4.2(d), dass bei ausreichender Anzahl gefundener Merkmale für eine erfolgreiche Homographieberechnung der Suchbereich komplett entsprechend seiner Ausdehnung abgedeckt werden kann.

4.7.2.5 RANSAC

Als weitere Auswahlmöglichkeit der Merkmale für die Berechnung der Homographie wurde der *Random Sample Consensus*-Algorithmus (RANSAC) getestet, welcher als Parameterschätzverfahren für Stichproben mit vielen Ausreißern verwendet werden kann. Mit RANSAC werden hier iterativ vier Merkmalspunkte zufällig ausgewählt und mit ihnen eine Schätzung der Homographie durchgeführt. Dieser Prozess wird entsprechend einer vorgegebenen Anzahl an maximalen Iterationen wiederholt. Basierend auf der ersten Schätzung werden mit jeder Iteration die neue Schätzung mit der vorigen auf Übereinstimmung anhand eines Schwellenwerts überprüft. Damit werden *inlier* und *outlier* bezüglich der Parameterschätzung bestimmt, wobei *inlier* dem *Consensus Set* zugeordnet werden. Schließlich wird die Homographie mit Hilfe eines *Least-Squares*-Schätzers aus dem größten *consensus set* bestimmt. Nachteil des Verfahrens ist ein erhöhter Rechenaufwand, der sich aus der iterativen Suche des *consensus sets* ergibt.

4.7.3 Bestimmung der Regularisierungsparameter

Die aus den ausgewählten Merkmalen und der Schätzung der Homographie resultierende unabhängige Posenänderung

$$\Delta\mathbf{p}^{\mathrm{H}} = \left(\Delta r_{\mathrm{x}}^{\mathrm{H}}, \Delta r_{\mathrm{y}}^{\mathrm{H}}, \Delta r_{\mathrm{z}}^{\mathrm{H}}, \Delta t_{\mathrm{x}}^{\mathrm{H}}, \Delta t_{\mathrm{y}}^{\mathrm{H}}, \Delta t_{\mathrm{z}}^{\mathrm{H}}\right)^{\mathrm{T}} \tag{4.132}$$

soll nun indirekt verwendet werden, um Parameter zur Regularisierung des optischen Flusses zu bestimmen.

Hierzu werden Funktionen $\boldsymbol{\lambda} = f_{\lambda_i}(\Delta\mathbf{p}^{\mathrm{H}})$ für die einzelnen Bewegungen $i = \{r_{\mathrm{x}}, r_{\mathrm{y}}, r_{\mathrm{z}}, t_{\mathrm{x}}, t_{\mathrm{y}}, t_{\mathrm{z}}\}$ bestimmt. Zur Bestimmung von $f(\cdot)$ wurden für alle Frames in allen Videos die auftretenden Bewegungen nach deren Auftrittshäufigkeit und maximalen Bewegungen untersucht. Aus den minimal zulässigen und maximal auftretenden Bewegungen ergeben sich dann zwei Randbedingungen zur Bestimmung von f. Es wurde dann empirisch ein exponentielles Modell gewählt, welches die unabhängig geschätzte Bewegungsänderung durch Verfolgung der stabilen Punkte auf die Regularisierungsparameter abbildet.

4.8 Ergebnisse

Um die Methoden zur Auswahl der Merkmale zu evaluieren, wurden insgesamt 45 Videos aus dem *Boston University Head Pose Dataset* (BU) sowie 60 Videos mit gleichmäßiger Beleuchtung des *IIIT Motion Capture Head Pose Datasets* (IIIT) [Vat+16] (www.iiit.kit.edu/datasets) ausgewertet.

4.8.0.1 BU

Die Videos zeigen jeweils 200 Frames einer Einzelperson unter verschiedenen Kopfbewegungen bei konstanten Beleuchtungsbedingungen, siehe Abb. 4.2, 4.9 und 4.4. Insgesamt handelt es sich um fünf verschiedene Probanden und jeweils neun Videos. Die *Ground Truth* wurden mit einem *Flock of Birds*-Tracker aufgenommen, der eine Genauigkeit von $0{,}5^{\circ}$ aufweist.

4.8.0.2 IIIT

Die Videos zeigen jeweils 300 Frames einer Einzelperson unter verschiedenen, teilweise sehr starken Kopfbewegungen von bis zu 51° Gieren (siehe Video auf www.iiit.kit.edu/datasets). Die *Ground Truth* wurden mit einem Motion-Capture-System aufgenommen, wofür eine Genauigkeit von besser als $0{,}11^\circ$ angegeben wird [Vat+16].

4.8.1 Qualitative Auswertung der bewegungsabhängigen Regularisierung

Um eine Evaluierung des Potentials der vorgestellten Regularisierungsmethode durchzuführen und den Einfluss auf die Kopfposenschätzung zu untersuchen, soll zunächst am Beispiel reiner Gier-Bewegungen ein Experiment durch manuelles Wählen der Regularisierungsparameter durchgeführt werden. Hierzu wurden für drei Probanden des IIIT-Datensatzes Videos mit ausgeprägter Gier-Bewegung für verschiedene Wahlen von λ_{r_y} ausgewertet. Das Ergebnis ist in Tabelle 4.1 zu sehen. Dabei wurde der optische Fluss bezüglich der Gier-Rotation um ein Vielfaches gegenüber den anderen Bewegungsrichtungen bevorzugt. Der positive Effekt auf die Posenschätzung durch eine Anpassung der Regularisierung ist deutlich zu erkennen, sodass selbst bei enormen Kopfdrehungen eine Schätzung mit Hilfe des optischen Flusses durchführbar ist. Im nächsten Abschnitt sollen die Ergebnisse, welche mit einer Online-Berechung der Regularisierungsparameter mit der vorgestellten Methode erreicht wurden, diskutiert werden.

4.8.2 Quantitative Auswertung bezüglich der Merkmalsauswahl

4.8.2.1 Konditionszahl

Zunächst wurde der direkte Einfluss der bewegungsabhängigen Regularisierung auf die Konditionszahl (Verhältnis von größtem zu kleinstem Eigenwert) der Hessematrix untersucht. In den Experimenten hat sich gezeigt, dass mit Hilfe der konventionellen Regularisierung (siehe Abschnitt 4.6.1) eine Senkung der Konditionszahl um $3 \cdot 10^1$ bis $8 \cdot 10^1$

möglich ist, während in der Literatur eine Senkung von $1 \cdot 10^2$ beschrieben wird [Xia+03]. Durch Regularisieren unter Berücksichtigung der aktuellen Bewegung konnte eine deutlich verbesserte Senkung der Konditionszahl zwischen $1 \cdot 10^2$ und $3{,}5 \cdot 10^2$ (*vam5*) erreicht werden. Der Einfluss hiervon auf die Genauigkeit soll im folgenden Abschnitt aufgezeigt werden.

Tabelle 4.1 *Mean Absolute Error* der Gier-Bewegungen ausgesuchter Videos des *IIIT*-Datensatzes in $^\circ$ für manuell gewählte Regularisierungsparameter.

Proband	Lennart	Thomas	Spiro	Gemittelt
$\text{Gieren}_{\max}$ in $^\circ$	51°	32°	34°	39°
λ_{r_y} konv.	$24{,}94^\circ$	$13{,}3^\circ$	$15{,}39^\circ$	$17{,}51^\circ$
$4 \cdot \lambda_{r_y}$	$19{,}99^\circ$	$10{,}01^\circ$	$10{,}80^\circ$	$13{,}59^\circ$
$8 \cdot \lambda_{r_y}$	$16{,}14^\circ$	$7{,}81^\circ$	$8{,}08^\circ$	$10{,}67^\circ$
$16 \cdot \lambda_{r_y}$	$11{,}95^\circ$	$5{,}95^\circ$	$6{,}51^\circ$	$7{,}95^\circ$

4.8.2.2 *BU*-Datensatz

Die Ergebnisse des *Mean Absolute Error* (MAE) der Roll-, Gier- und Nick-Bewegungen sind in der Abb. 4.3 zu sehen. Bei der Auswertung wurden nur die Videos, bei denen ein erfolgreiches Tracking durchgeführt werden konnte, berücksichtigt. Dabei gilt ein Tracking als nicht erfolgreich, wenn ein mittlerer Fehler von mehr als 10° bezüglich einer Rotation erreicht wurde. Die Ergebnisse wurde mit einem Zylinder als Kopfmodell und einer automatischen Initialisierung basierend auf den detektierten Iriden durchgeführt, wobei die Größe des das Gesicht repräsentierenden Rechtecks mittels anthroprometischer Werte bestimmt wurde.
Man erkennt, dass die Metrik ähnlich wie die Distanz ein aussagekräftiges Charakteristikum zur Auswahl der Merkmale ist. Das Alter alleine gibt zwar Aufschluss über die Stabilität eines einzelnen Merkmalspunktes, ist allerdings ungeeignet als Kriterium bei der Auswahl der Merkmalspunkte zur Bestimmung der Homographie. Mit RANSAC wird das beste Ergebnis bezüglich der Kopfposenschätzung erzielt, wobei eine Verbesserung der Genauigkeit um über $19{,}2\,\%$ im Vergleich zur kon-

ventionellen Regularisierung, wobei für alle sechs Parameter der Wert $0{,}5$ gewählt wird, erreicht wird. Es gilt weiterhin zu beachten, dass bei Wahl von $\boldsymbol{\lambda} = \mathbf{0}$ in Glg. (4.124) die Kopfposenschätzung für nahezu alle Videos fehlschlägt, was die Wichtigkeit einer sinnvollen Regularisierung unterstreicht.

MAE in $^\circ$	Rollen	Gieren	Neigen	Gesamt
$\mathbf{H}_{\mathrm{Reg}}^{\mathrm{konv.}}$	2,21	6,65	4,19	13,05
Alter	2,38	6,00	3,89	12,26
Metrik	2,18	6,05	3,70	11,84
Distanz	2,11	5,88	3,90	11,89
RANSAC	2,12	4,79	3,62	10,54
$\Delta\mathbf{p}_{\mathbf{dir}}^{\mathrm{H}}$	2,85	4,59	4,80	12,25

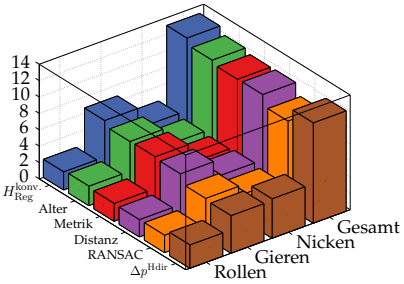


Abbildung 4.3 *Mean Absolute Error* gemittelt über 38 Videos des *BU*-Datensatzes für die konventionelle Regularisierung, unterschiedliche Wahlen der Merkmalspunkte zur Homographieberechnung sowie der direkten Anwendung der berechneten Homographie. Fehler in $^\circ$ in der rechten Abbildung.

4.8.2.3 Direkte Anwendung der Homographie

Des Weiteren wurde untersucht, inwiefern sich die aus der Auflösung der Homographie resultierenden direkten Bewegungsänderungen $\Delta\mathbf{p}_{\mathrm{dir}}^{\mathrm{H}}$ für die Kopfposenschätzung nutzen lassen. Erste Untersuchungen zeigten, dass eine direkte Verwendung der Posenschätzung aus der Homographieberechnung zur Aktualisierung der Kopfpose nicht zielführend ist. Probleme resultierten zum einen aus einer zu geringen Anzahl gefundener Merkmale, die eine ausbleibende Schätzung zur Folge haben, und zum anderen aus einer Schätzung von $\Delta\mathbf{p}^{\mathrm{H}}$, die zu unrealistischen Bewegungsänderungen führte. Zur Lösung wurde zum einen der Schwellenwert zum Finden möglicher Interessenpunkte herabgesetzt (Determinanten der lokalen 2×2-Hesse-Matrix der tiefpassgefilterten Bildregion). Weiterhin wurden Restriktionen für maximale Bewegungsänderungen gesetzt, sodass bei einer unrealistischen Bewegungschätzung diese verworfen und stattdessen die vorhergehende Schätzung verwendet wird, wobei eine langsame relative Änderung der Bewegung angenommen

wurde. Wie in der Tabelle in Abb. 4.3 zu sehen ist, lassen sich auf diese Weise insbesondere deutlich bessere Gieren-Ergebnisse erzielen. Dies lässt sich damit begründen, dass bei direkter Anwendung von $\Delta\mathbf{p}_{\mathrm{dir}}^{\mathrm{H}}$ das Aperturproblem, welches insbesondere Schwierigkeiten bei starken Gier-Bewegungen verursacht, keine Rolle spielt.

4.8.2.4 Rechenaufwand

Bei der Betrachtung des Rechenaufwandes in Tabelle 4.2 stellt man fest, dass der Gewinn an Genauigkeit durch Verlust an Performanz in der Rechenzeit erkauft wird. Hierbei unterscheiden sich die Methoden einer deterministischen Merkmalsauswahl kaum von RANSAC. Die Erklärung hierfür sowie für die insgesamt nicht ausreichende Echtzeitfähigkeit ist, dass weite Programmteile prototypisch in MATLAB implementiert sind, insbesondere die Merkmalsauswahl, während für RANSAC die MATLAB-Implementierung verwendet wurde. Die Defizite bei der Performanz lassen sich durch Migration des Codes in *C++* beheben und so ein echtzeitfähiges System gestalten.

Tabelle 4.2 Mittlere Bildrate (fps) über 38 Videos des *BU*-Datensatzes.

Methode	$\mathbf{H}_{\mathrm{Reg}}^{\mathrm{konv.}}$	RANSAC	Metrik	$\Delta\mathbf{p}_{\mathrm{dir}}^{\mathrm{H}}$
fps in $1\,\mathrm{s}^{-1}$	9,5	5,2	4,9	10,7

4.8.3 Quantitative Auswertung bezüglich Initialisierung und Kopfmodell

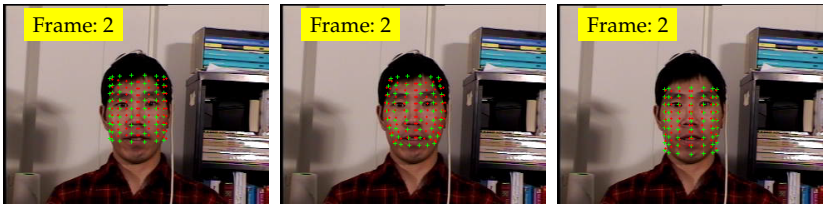
Basierend auf den Ergebnissen aus Kap. 4.8.2 sollen hier nun die Einflüsse des gewählten Kopfmodells sowie seiner Initialisierung diskutiert werden. Dabei wurde stets die Regularisierung durch die Verfolgung stabiler Merkmale unterstützt und die Homographie mittels RANSAC bestimmt.

In Abb. 4.5 sind die *Area Under Curve* (AUC)-Ergebnisse für drei Szenerien gezeigt

1. Zylinder als Kopfmodell, Initialisierung durch Irislokalisation und antropometrische Werte (voll automatisch);
2. Ellipsoid als Kopfmodell, Initialisierung durch Irislokalisation und antropometrische Werte (voll automatisch);
3. Zylinder als Kopfmodell, Initialisierung aus Datei (manuell) mit Vorgabe der Gesichtsbox sowie Iridenzentren, initiale Tiefe aus Augenabstand und anthropometrischen Werten bestimmt.

Bei der Diskussion ist zu beachten, dass die voll automatische Initialisierung auf einen für alle Auswertungen konstanten Parametersatz für die Bestimmung der Größe des Zylinders zurückgreift. Da die Probanden unterschiedliche Anatomien aufweisen, sind die Initialisierungen somit nicht immer ideal. Für den Fall der Initialisierung wurde versucht aus zuvor von Hand bestimmten Werten diesen Einfluss mittels Einlesens aus Datei zu unterdrücken.

Die Abb. 4.4 zeigt die Initialisierung der einzelnen Varianten für das Video *llm4* des *BU*-Datensatzes. Es wurde jeweils der FC-Algorithmus sowie RANSAC bei der Regularisierung genutzt.



(a) Zyl., autom., t_z^0 autom. **(b)** Ell., manuell, t_z^0 autom. **(c)** Zyl., manuell, t_z^0 autom.

Abbildung 4.4 Initialisierungen der Kopfmodelle für die Sequenz *llm4*, rot gekennzeichnete Punkte zeigen im aktuellen Frame nicht sichtare 3D-Körperpunkte.

Man erkennt insbesondere zwischen Abb. 4.4(a) und 4.4(c) die stark unterschiedlichen initialen Posen des Zylinders, wobei das Template im einem Fall die Stirn und im anderen das Kinn mit einbezieht. Die automatische Initialisierung über die Augen ist empirisch so implementiert, dass inbesondere bei dem komplexeren Ellipsoidenkopfmodell das Modell bei visueller Inspektion das Gesicht möglichst ideal abdeckt (4.4(b)).

Die Abb. 4.5 zeigt für die Bewegungen Rollen, Gieren und Neigen einen Vergleich der Schätzergebnisse der in Abb. 4.4 gezeigten Initialisierungen.

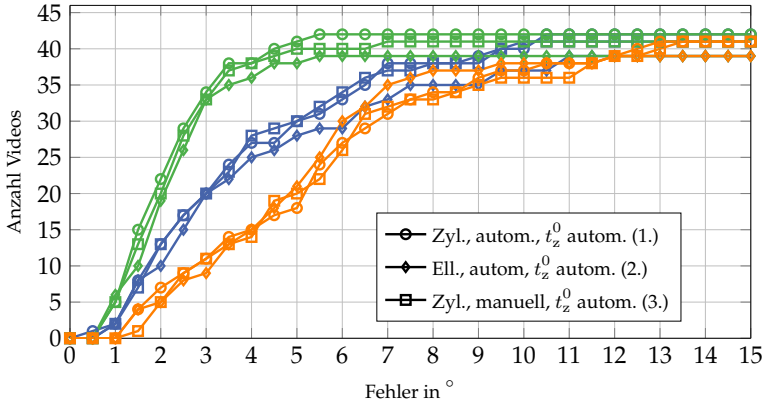


Abbildung 4.5 AUC für den *BU*-Datensatz mit den in Abb. 4.4 illustrierten Ansätzen zur Initialisierung. In Grün sind die Ergebnisse für Rollen, in Blau für Nicken und in Orange für Gieren gezeigt. Die AUC gibt den kumulierten Anteil aller Videos mit einem maximalen Trackingfehler entsprechend der Abszissenachse an.

Zunächst erkennt man, dass die Methode die einzelnen Bewegungen Rollen (grün), Gieren (orange) und Nicken (blau) unterschiedlich genau schätzt. Während für das Rollen und Neigen das Ellipsoid schlechtere Ergebnisse als der Zylinder liefert, verbessert das Verwenden eines Ellipsoiden die Gieren-Schätzung. Beim Vergleich der automatischen und manuellen Initialisierung zeigt sich, dass inbesondere beim Rollen der automatisch initialisierte Zylinder genauere Ergebnisse liefert, während für die Bewegungen in die Ebene hinein bessere Ergebnisse für eine manuelle Initialisierung zu beobachten sind. Eine Begründung könnte sein, dass die bessere Anpassung durch manuelle Initialisierung die Schätzung tatsächlich verbessert, während der Stirnbereich im automatischen Fall oft größer ist und dieser beim Rollen mehr Information fürs Template bereitstellt, diese bei Bewegungen in die Ebene hinein allerdings nicht von Nutzen sind.
Tabelle. 4.3 liefert einen quantitativen Vergleich der Ergebnisse um folgende Initialisierungen ergänzt:

4. Zylinder als Kopfmodell, Initialisierung aus Datei (manuell) mit Vorgabe der Gesichtsbox sowie Iridenzentren, initiale Tiefe aus der *Ground Truth* (Zyl., manuell, t_z^0 GT.), Regularisierungsparameter mit linearer statt exponentieller Abbildung;

5. Ellipsoid als Kopfmodell, Initialisierung aus Datei (manuell) mit Vorgabe der Gesichtsbox sowie Iridenzentren, initiale Tiefe aus Augenabstand und anthropometrischen Werten bestimmt (Ell., manuell, t_z^0 autom.).

Tabelle 4.3 *Area Under Curve* (AUC), angegeben in der Anzahl vollständig verfolgter Videos. Farbliche Hervorhebungen der besten Ergebnisse sind entsprechend den Abb. 4.6, 4.7 und 4.8 gewählt.

Ansatz	Rollen			Gieren			Nicken		
	$\leq 1^\circ$	$\leq 5^\circ$	$\leq 10^\circ$	$\leq 1^\circ$	$\leq 5^\circ$	$\leq 10^\circ$	$\leq 1^\circ$	$\leq 5^\circ$	$\leq 10^\circ$
1. Zyl., autom., t_z^0 autom.	5	41	42	0	18	37	2	30	40
2. Ell., autom., t_z^0 autom.	6	38	39	0	21	38	2	28	37
3. Zyl., manuell, t_z^0 autom.	5	40	41	0	20	36	2	30	41
4. Ell., manuell, t_z^0 autom.	4	40	41	0	21	39	2	29	39
5. Zyl., manuell, t_z^0 GT	7	43	45	0	21	41	2	31	43

Beim Ansatz „Zyl., manuell, t_z^0 GT“ wurde eine andere Abbildungsfunktion zur Bestimmung der Regularisierungsparameter genutzt, deren Einfluss auf das Gesamtergebnis gering ist und daher hier vernachlässigt werden soll. Der Vergleich fünf unterschiedlicher Initialisierungen zeigt zunächst den starken Einfluss der Kopfmodelle und der Genauigkeit der Initialisierung auf das Trackingergebnis. Gut zu erkennen ist, dass eine genaue, manuelle Initialisierung das Ergebnis positiv beeinflusst. Weiterhin lässt sich erkennen, dass, obwohl das Ergebnis insgesamt schlechter ist, sich ein Ellipsoid besser zur Schätzung der Gierenbewegung eignet. Die Abbildungen 4.6, 4.7 und 4.8 zeigen AUC-Ergebnisse separat für Rollen, Gieren und Nicken. Die Darstellung als *Area Under Curve* bietet den Vorteil, dass die Anzahl der Videos mit einem Fehler oberhalb eines Schwellenwertes direkt als vertikaler Abstand zur 100 %-Linie (hier 45 Videos) abgelesen werden kann, was aus einem MAE nicht direkt ersichtlich ist [WSA17]. Bei Betrachtung des Verlaufs aller Kurven werden die

Herausforderungen bei der Gieren-Schätzung deutlich. Insbesondere im Bereich geringer Fehler unterscheiden sich die Initialisierungen kaum. Dies wird dadurch erklärt, dass es einige Videos gibt, in denen nur geringe r_{y}-Bewegungen stattfinden, welche dann von allen Ansätzen gut getrackt werden können. Erst für das Erlauben größerer Fehler bis hin zum Abreißen des Trackings macht sich die bessere Initialisierung bemerkbar (siehe Ansatz 5 für alle drei Abbildungen).

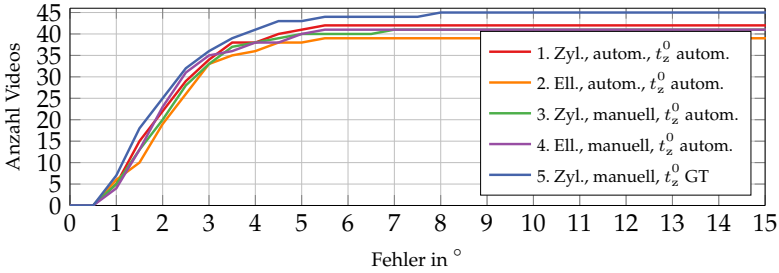


Abbildung 4.6 AUC für Rollen für den *BU*-Datensatz. Anzahl der Videos aufgetragen über dem Fehler in °.

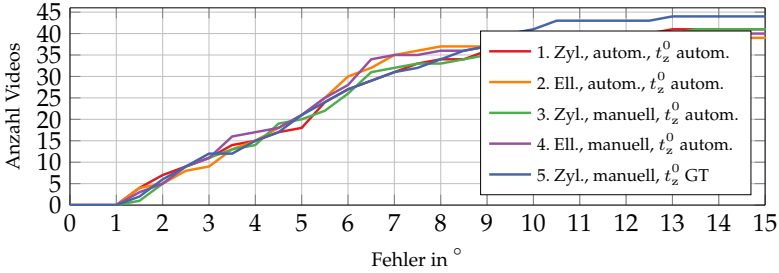


Abbildung 4.7 AUC für Gieren für den *BU*-Datensatz. Anzahl der Videos aufgetragen über dem Fehler in °.

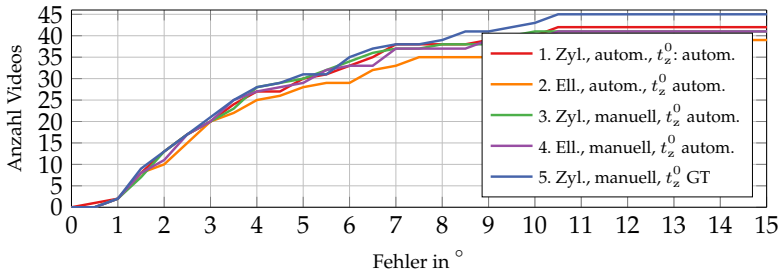


Abbildung 4.8 AUC für Neigen für den *BU*-Datensatz. Anzahl der Videos aufgetragen über dem Fehler in °.

4.8.4 Beispielhafte Auswertung zur bewegungsabhängigen Regularisierung

Zur Illustration der Unterschiede der Kopfposenschätzung in Abhängigkeit der Auswahl der zur Bestimmung der Regularisierungsparameter gewählten Merkmale sind in den Abb. 4.9 und 4.10 jeweils vier Frames des Videos *jim9* dargestellt.

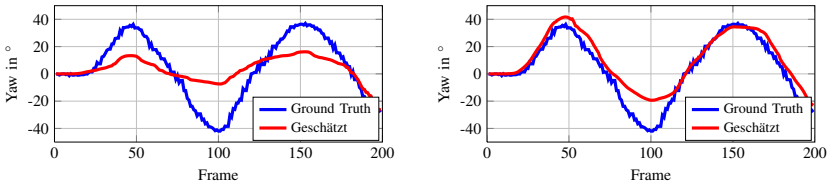
(a) Frame 20. **(b)** Frame 30. **(c)** Frame 40. **(d)** Frame 50.

Abbildung 4.9 Kopfposenschätzung für *jim9* mit Alter als Merksmalsauswahl.

(a) Frame 20. **(b)** Frame 30. **(c)** Frame 40. **(d)** Frame 50.

Abbildung 4.10 Kopfposenschätzung für *jim9* mit RANSAC als Merksmalsauswahl.

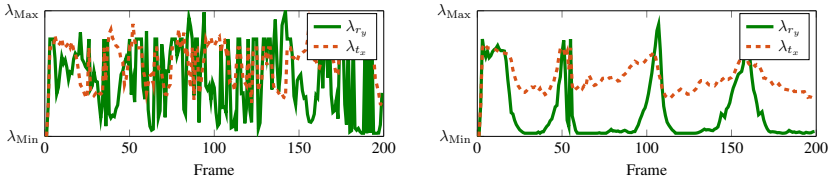
Der rote Pfeil zeigt den aktuellen Normalenvektor der Frontalen des den Kopf modellierenden Zylinders an und entspricht der allein aus der Kopfposition resultierenden Blickrichtung. Man erkennt, dass in Abb. 4.9 sich der Zylinder zum einen zu weit rechts vom Kopf befindet, was aus einer Überschätzung von t_x resultiert. Zum anderen erkennt man bei Vergleich von Abb. 4.9 und Abb. 4.10, dass die Gier-Rotation r_y stark unterschätzt wird, wenn man das Alter zur Merkmalsauswahl heranzieht. Die Missdeutung des optischen Flusses zwischen r_y und t_x in Abb. 4.9 verdeutlicht sehr anschaulich das Aperturproblem, während in Abb. 4.10 das verbesserte Ergebnis mit der vorgestellten Methode demonstriert wird. Abbildung 4.11 zeigt diesen Zusammenhang anhand der *Ground Truth* und Schätzdaten für die beiden Experimente.



(a) Merkmalsauswahl nach Alter. **(b)** Merkmalsauswahl mit RANSAC.

Abbildung 4.11 Schätzung und *Ground Truth* der Gier-Bewegung für das Video *jim9*.

Die Verbesserung der Genauigkeit der Kopfposenschätzung lässt sich anhand von Abb. 4.12 am Verlauf der Regularisierungparameter für r_y und t_x nachvollziehen. In Abb. 4.12(a) lässt sich kein klar unterschiedliches Verhalten von λ_{r_y} und λ_{t_x} erkennen. Bei Verwendung von RANSAC reagieren die Regularisierungsparameter wie gewünscht: Man beobachtet eine geringe Regularisierung (kleines λ_{r_y}) des optischen Flusses in Bereichen starker Gier-Bewegung. Eine starke Bewegungsänderung kann man insbesondere an den Null-Durchgängen in Abb. 4.11 erkennen (Frame 70, 120 und 180). Insbesondere zeigt Abb. 4.12(b) ein dominantes Verhalten von λ_{r_y}, welches im Vergleich zu λ_{t_x} durch niedrige Werte viel optischen Fluss zulässt. Ingesamt verbessert sich der MAE für das Gieren von $14{,}3^\circ$ auf $7{,}7^\circ$.



(a) Merkmalsauswahl nach Alter. **(b)** Merkmalsauswahl mit RANSAC.

Abbildung 4.12 Regularisierungsparameter λ_{r_y} (Gieren, in Grün) und λ_{t_x} (x-Translation, in Braun) für *jim9*.

4.8.5 Betrachtung des Hessematrix

Der Einfluss der bewegungsabhängigen Regularisierung soll nun anhand der regularisierten Hessematrix und derer Einträge näher untersucht werden (vgl. hierzu Glg. (4.126)). Neben dem Beispiel aus Abb. 4.10 soll hierzu die Sequenz *llm3* herangezogen werden. Die Ergebnisse für t_x, r_z (Rollen) und r_y (Gieren) sind in den Abb. 4.13(a), 4.13(b) und 4.13(c) gezeigt. Die Ergebnisse der Kopfposenschätzung wurden analog zu *jim7* mit dem Initialisierungsansatz 3, dem FC-Algorithmus und RANSAC produziert.

Zugehörig zu den beiden Sequenzen zeigt Abb. 4.14 die Einträge der Hessematrizen für Frame 30 bei *jim7* und Frame 130 bei *llm3* jeweils für die konventionelle Regularisierung $\mathbf{H}_\text{Reg}^\text{konv.}$ und für die bewegungsabhängige Regularisierung $\mathbf{H}_\text{Reg}^\text{bew.}$, wobei ein Kasten gerade einem Eintrag der 6×6-Matrix entspricht.

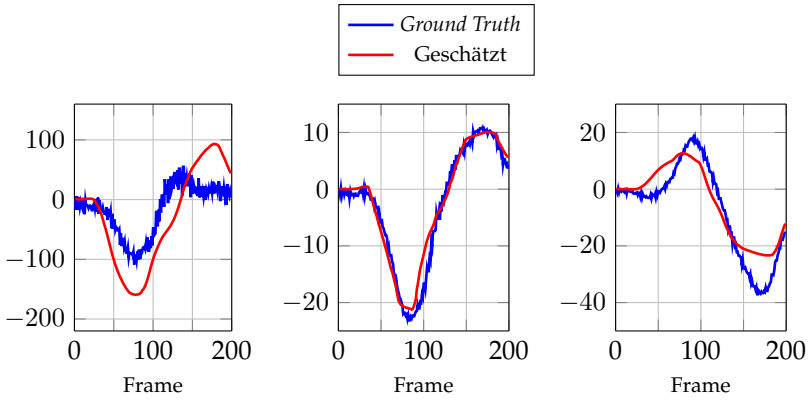
Die Matrix zeigt auf der Hauptdiagonalen die Quadrate der regularisierten Jacobimatrix der Warpingfunktionenen der einzelnen Bewegungsrichtungen (siehe Glg. (4.117) und vorangehende). Auf den Nebendiagonalen sind Kreuzterme zu finden, welche die intrinsischen Ambiguitäten zwischen beispielsweise kleinen x-Translationen und y-Rotationen wiedergeben. Bei einer perfekten Entkopplung der Warpingfunktionen würden die Nebendiagonalen den Wert null annehmen.

4.8.5.1 Interpretation und Regularisierung

Mittels der Hessematrix lässt sich durch die Kopplung mit dem optischen Fluss in Glg. (4.121) die bevorzugte Richtung des optischen

Flusses und damit das Schätzergebnis manipulieren. Idealerweise sollte die Manipulation in Abhängigkeit des aktuellen Fehlerbildes und der vorherrschenden Bewegungsänderung geschehen. Dabei führen große Werte in der Hessematrix zu einer Dämpfung des optischen Flusses, während kleine Werte viel optischen Fluss in diese Richtung zulassen.

Für den ersten Fall ist eine starke x-Translation gekoppelt mit einer Gierenbewegung zu beobachten. Beim Vergleich der regularisierten Hessematrizen (untere Zeile) für *jim7* fällt auf, dass nach Regularisierung inbesondere die gekoppelte t_x-r_y-Bewegung bevorzugt wird (niedrige Werte im Kasten (2,4) bzw. (4,2)), während reine Bewegungen (Hauptdiagonale) blockiert werden. Das Ergebnis kann hierdurch von einer konventionellen Regularisierung von einem Fehler $\left[3{,}5^\circ, 14{,}3^\circ, 1{,}1^\circ\right]$ auf $\left[3{,}8^\circ, 7{,}7^\circ, 1{,}8^\circ\right]$ gesenkt werden.



(a) Translation in x-Richtung (t_x).

(b) Rollen (r_z).

(c) Gieren (r_y).

Abbildung 4.13 Schätzergebnise für *llm3*. Die rote Linie kennzeichnet jeweils das Schätzergebnis, die blaue die GT.

Beim Video *llm3* ist Ähnliches zu beobachten. Die Bewegung im Video zeichnet sich um Frame 130 herum durch starke t_x- und r_y-Bewegungen und gleichzeitig starkes Rollen aus. Die konventionelle Hessematrix blockiert stark die reine Gierenbewegung. Während die bewegungsadaptive Regularisierungsmatrix die reine Gierenbewegung weniger

dämpft, lässt sie insbesondere kombinierte t_x-r_y- (2,4) bzw. (4,2) sowie r_y-r_z-Bewegungen zu (2,3) bzw. (3,2). Im Ergebnis zeichnet sich dies durch eine Verbesserung von $[2{,}3^\circ, 10{,}6^\circ, 1{,}7^\circ]$ auf $[1{,}4^\circ, 5{,}4^\circ, 2{,}2^\circ]$ aus.

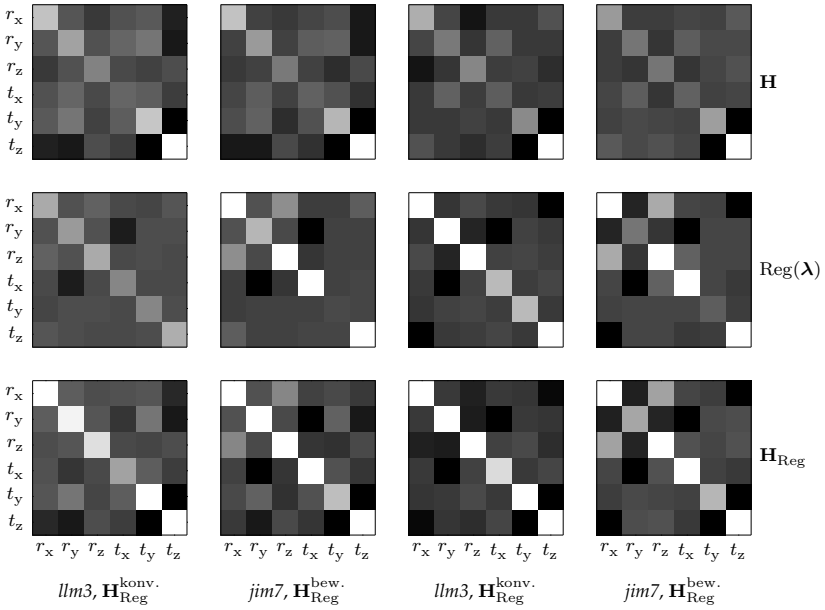


Abbildung 4.14 Hessematrizen der Sequenzen *jim7* (Frame 30) und *llm3* (Frame 130) für die konventionelle und die bewegungsadaptive Regularisierung. Helle Kästen entsprechen großen Werten.

4.9 Zusammenfassung

In diesem Kapitel wurde eine Methode zur erscheinungsbasierten Kopfposenschätzung mittels optischem Fluss vorgestellt. Es wurde ein bewegungsadaptiver Ansatz zur Regularisierung des optischen Flusses vorgestellt, wobei folgende, den Stand der Technik erweiternde, Neuerungen erforscht wurden:

- Mit Hilfe der Definition eines vektorwertigen Regularisierungsparameters konnte gezeigt werden, wie dieser sinnvoll eingesetzt

werden kann, um die Konditionszahl der in den Gleichungen des optischen Flusses auftretenden Hessematrix zu senken und so den optischen Fluss angepasst an die auftretende Bewegegung zu manipulieren, um sowohl die Genauigkeit als auch die Stabilität der Schätzung zu erhöhen.

- Es wurde eine Methode implementiert und vorgestellt, mit Hilfe derer durch eine vom optischen Fluss unabhängige Schätzung der euklidischen Bewegung des Kopfes durch Berechnung der Homographie sukzessiv verfolgter stabiler Punkte durchgeführt werden kann und somit die Regularisierungsparameter adaptiv in Echtzeit bestimmt werden können.

- Insbesondere wurde anschaulich die positive Manipulation der Hessematrix durch Visualisierung der einzelnen Matrixeinträge durch Gegenüberstellen der regularisierten und nicht regularisierten Hessematrizen verdeutlicht.

Mit Hilfe der vorgestellten Methoden konnte der über die drei Rotationsbewegungen gemittelte mittlere absolute Fehler der erscheinungsbasierten Kopfposenschätzung für den weit verbreiteten Boston University-Datensatz um fast $20\,\%$ gesenkt werden.

Darüber hinaus wurde eine nach Wissen des Autors über die Literatur hinaus vollständige Herleitung der verwendeten Gleichungen bereitgestellt, um die Beschreibung der erforschten Methoden mathematisch ausführlich zu erfassen.

Der modulare Aufbau und eine vollständige eigene Implementierung erlauben die Integration in ein Rahmenwerk, welches durch Kombination mit einer Irislokalisation den Aufbau eines Algorithmus zur erscheinungsbasierten Schätzung der Blickrichtung ermöglicht.

In diesem Zusammenhang wurden im Verlaufe der Arbeit zur hier vorgelegten Dissertation bestehende Ansätze um folgende Erweiterungen ergänzt:

- Algorithmen zur Berechnung des optischen Flusses basierend auf dem *Additive Forward-*, *Compositional Forward-* und *Inverse Compositional*-Algorithmus unter Verwendung einer Starrkörperbewegung und eines Lochkameramodells;

- Zahlreiche Ansätze, welche die Robustheit des System erhöhen und die Rechnenzeit verringern, hierzu gehören:
  - Berechnung des optischen Flusses auf einer Pyramide verschiedener Bildoktaven;
  - ein Ansatz zur Bewältigung von Beleuchtungsinhomogenitäten durch Kompensation von örtlichem Mittelwert und Varianz durch Berechnungen im Spektralbereich;
  - den Update-*Template*-Algorithmus aus [MIB04];
  - Zylinder- und Ellipsoidenkopfmodelle sowie deren Projektionsfunktionen.

5 Monokulare Blickrichtungsschätzung

Wie bereits in den Einleitungen zu den vorangegangenen Kapiteln dargelegt, durchlebt die Mensch-Maschine-Interaktion durch ein zunehmendes Ersetzen konventioneller Ein- und Ausgabesysteme, wie die Computermaus und -tastatur, einen Wandel, welcher sich bereits in handgehaltenen Geräten in Form von Touchscreens und seit Herbst des Jahres 2017 [Appb], für die Aufgaben der Erkennung der Identität sowie der Aufmerksamkeit des Nutzers, auch in Form von berührungslosen Interaktionen abzeichnet. Im abschließenden Kapitel dieser Arbeit soll auf die starke Rolle, die die Blickrichtungsschätzung als Form der berührungslosen Mensch-Maschine-Interaktion in Zukunft einnehmen kann, eingegangen werden.

Hierzu werden die Erkenntnisse und Algorithmen, die in den Kapiteln 2 bis 4 erforscht und dargelegt wurden, entsprechend der Abb. 1.2 zusammengeführt. Der hier verfolgte Ansatz nutzt dabei im Rahmen eines Kalibrierungsprozesses die geschätzten Informationen über die Kopfpose sowie die Positionen der Augen, um mit Hilfe dieser eine Regressionsfunktion zu bestimmen. Die Regressionsfunktion kann dann online zur Abbildung der geschätzten Daten auf die Blickrichtung genutzt werden, um damit auf den Punkt des Interesses des Nutzers zu schließen.

5.1 Problemstellung

Beim erscheinungsbasierten Ansatz zur Bestimmung der Blickrichtung werden ausschließlich zweidimensionale Bilddaten für die Schätzung verwendet. Während somit minimale Randbedingungen an die Sensorik (es wird – im Vergleich zu anderen Ansätzen – lediglich ein (monochro-

matisches) Bild benötigt) gestellt werden, steht man dem Problem des inhärenten Fehlens von Tiefeninformation gegenüber. Weiterhin kann die Aufnahmehardware in ihrer Qualität stark variieren, was wiederum direkten Einfluss auf die Qualität der zur Blickrichtungsschätzung zur Verfügung stehenden Information hat.

Der Einfluss der Bildqualität auf die hier verfolgte Methodik wurde insbesondere in Kap. 3 bereits diskutiert. Dieser Einfluss soll nun im Kontext der Zielsetzung der Arbeit unter der Motivation der monokularen Blickrichtungsschätzung, hinsichtlich der Güte der Blickrichtungsschätzung, näher betrachtet werden. Anhand eines schematischen Beispiels zur Abschätzung der Genauigkeit der Blickrichtungssschätzung mittels der erscheinungsbasierten Methoden des optischen Flusses sowie der pixelgenauen Irislokalisation wird diese Güte unter fehlerhafter Irislokalisation in Abb. 5.1 untersucht. Dargestellt ist der Einfluss einer um Δu vom Irismittelpunkt abweichenden Lokalisation auf die aus der Iris und der 3D-Kopfpose resultierenden Blickrichtung. Der Einfachheit halber ist ein Szenario skizziert, welches den Winkelfehler in die horizontale Ebene projiziert betrachtet.

Mit der Brennweite f des Sensors, der das monokulare Bild aufnimmt, und dem Faktor zur Umrechnung von Pixel in Weltkoordinaten in mm, S_x, welcher der intrinsischen Kameramatrix zu entnehmen ist, kann für einen Abstand Benutzer–Bildschirm von $800\,\mathrm{mm}$ der aus einem Fehler $\Delta u = 1$ Pixel resultierende Winkelfehler durch

$$\Delta\alpha = \arctan\left(\frac{\mathrm{z}_\mathrm{Auge}\cdot\Delta u}{\mathrm{f}\cdot S_\mathrm{x}\cdot\mathrm{r}}\right) \approx 1^\circ \tag{5.1}$$

bestimmt werden, wobei in Glg. (5.1) ein Lochkameramodell angenommen wird. Die Beispielrechnung für nur *ein* Pixel Fehler zeigt die starke Abhängigkeit der Genauigkeit der Blickrichtungsschätzung von der Irislokalisation für ein realistisch angenommenes Szenario, wie es auch im hier durchgeführten Experiment erfüllt wird.

In Kap. 4 wurde eine Möglichkeit zur Schätzung der Kopfpose vorgestellt, auf welche hier zugegriffen werden soll, um der Herausforderung der fehlenden Tiefeninformation der Rohdaten zu begegnen und die Blickrichtungsschätzung kopfposeninvariant zu gestalten.

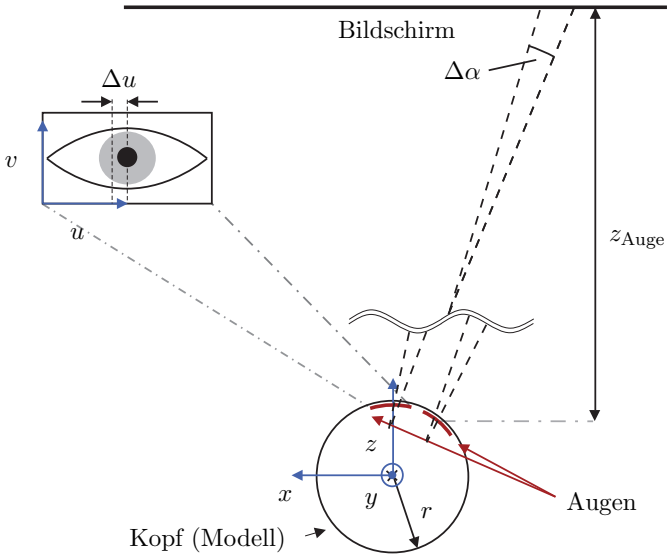


Abbildung 5.1 Skizze des aus ungenauer Irislokalisation resultierenden horizontalen Winkelfehlers $\Delta\alpha$ der Blickrichtungsschätzung als Projektion in die (x, z)-Ebene. (Siehe auch 3D-Skizze in Abb. 1.1.)

5.2 Stand der Technik

In dieser Arbeit wird die Blickrichtungsschätzung auf Grundlage der erscheinungsbasierten Methoden der vorangegangenen Kapitel unter Verwendung eines Regressionsansatzes durchgeführt. Während andere Möglichkeiten zur Inferenz der Blickrichtung aus 2D-Bilddaten hier kurz genannt werden sollen, soll der Stand der Technik auf algorithmischer Seite Ansätze zum Rahmenwerk solcher Systeme beleuchten (Methoden, Abschnitt 5.2.1). Die Algorithmik zur Bestimmung der zur Regression genutzten Eingangsdaten wurde in den vorangegangenen Kapiteln untersucht und es wurden zum jeweiligen Stand der Technik in den einzelnen Kapiteln ausführliche Übersichten gegeben (Abschnitte 2.2, 3.2, 4.2). Neben Methoden zum Stand der Technik soll weiterhin ein Überblick über kommerzielle Produkte (Abschnitt 5.2.2) gegeben werden.

5.2.1 Methoden

Eine Übersicht zu erscheinungsbasierten sowie nicht erscheinungsbasierten Methoden zur Blickrichtungsschätzung ist von Hansen und Qiang [HQ10] gegeben, während von Gawande und Nathaney [GN17] in einer Übersicht zur Blickrichtungsschätzung der Fokus auf der Anwendung solcher Algorithmen auf Smartphones liegt.

Die Ansätze, die eine Regressionsanalyse zur Bestimmung der Blickrichtung nutzen, lassen sich zum einen in die unterschiedlichen erscheinungsbasierten Methoden, die zur Gewinnung der unabhängigen Variablen angewandt werden, zum anderen in die Regressionsmethoden, die angewandt werden, um von den unabhängigen auf die abhängigen Variablen zu schließen (im Fall hier die Pixelkoordinaten auf dem Monitor), unterscheiden.

Ein erscheinungsbasierter Ansatz, der auf Basis der lokalisierten Augenwinkel Bildauschnitte ausrichtet und so auf die Blickrichtung schließt, wird von Schneider et al. [SSS14] vorgeschlagen. Sie nutzen, ohne Information über die Kopfpose zu verwenden, die Pixelintensitäten in einem Bereich um das Auge, welche sie zunächst vorverarbeiten, indem sie den Bereich mittels *Manifold Alignment* an ein Referenzmodell anpassen, um dann einen 2D-Blickrichtungsvektor zu bestimmen. Sie werten neben einem Polynom dritter Ordnung und *Support Vector Regression* auch eine *Relevance Vector Regression* und eine Nächster-Nachbar-Klassifikation aus. Eine Polynomregression wird früh von Ramanauskas [Ram06] genutzt, um mittels durch ein *Head-Mounted-Device* aufgenommener Daten auf die Blickrichtung zu schließen. Die Autoren in [Hil+10] fusionieren nach Bestimmung der groben Blickrichtung mittels eines künstlichen neuronalen Netzwerkes unter Festhalten der Kopfposition mit einem *Chin-Rest* die so gewonnenene Blickrichtung mit einer Aufmerksamkeitskarte, die sie auf Basis von Salienzen im Bild bestimmen.

Mittels Modellierung des Augapfels als 3D-Körper wird in [CJ11] ein probabilistischer Ansatz zur Kalibrierung mittels Salienzenkarten angewandt.

In [VSG12] wird die Kalibrierung mittels 2D-Augenpositionen für eine initiale, auf Basis des optischen Flusses berechnete, Kopfpose für eine Referenz-Kalibrierebene (Monitor) durchgeführt. Die auf Basis der 2D-Augenpositionen kalibrierte Ebene zur Blickrichtungsbestimmung wird

dann für Änderungen der Kopfpose anhand der Anpassung an bekannte Blickrichtungsdaten neu kalibriert, indem die Referenzebene mit der Transformationsmatrix der Kopfpose manipuliert wird. Auf diese Weise umgehen die Autoren die Prozedur des Durchfahrens der Kalibrierebene. Die Autoren in [Lai+14] erlauben freie Kopfbewegungen für die Blickrichtungsschätzung, indem sie die Information der Augenerscheinung mit der 3D-Kopfpose in einem Entscheidungsbaum fusionieren. Sugano et al. [SMS13] präsentieren einen Ansatz, bei dem für den Benutzer durch Anschauen eines Videos, für welches im Vorfeld Salienzen berechnet wurden, eine automatische Kalibrierung durchgeführt wird. Sie nutzen nur die Grauwertbilder des Augenbereichs als Information. Die Autoren erweiteren in [Lu+15] ihren Ansatz auf eine kopfposeninvariante Methode. Sie nehmen hierzu neben Augenbildern in neutraler Pose vier weitere Bilder unter verschiedenen Kopfposen auf. Anschließend synthetisieren sie neue Trainingsbilder der Augenregionen für ungesehene Kopfpositionen und lernen direkte Zusammenhänge zwischen Augenerscheinungsbild und Blickrichtung. Sugano et al. [SMS14] präsentieren den synthetisch generierten *UT*-Datensatz und lernen ein Modell zur Bestimmung der Blickrichtung, indem sie eine 3D-Rekonstruktion der Augenregionen aus dem zahlreiche Blickrichtungen beinhaltenden Datensatz durchführen. Als Regressionsfunktion nutzen sie Entscheidungsbäume. Mittels eines PUPIL-*Head-Mounted-Device* [pup] nehmen die Autoren in [Man+15] präzise Daten zur Blickrichtungsbestimmung auf und untersuchen den Einfluss der Dimensionalitäten zwischen 2D und 3D der Eingangsdaten und Blickpunkte. Wood et al. [Woo+16] nutzen eine Million Bilder, die ebenfalls synthetisch erzeugt wurden, um dann eine Nächster-Nachbar-Regression zwischen unter unterschiedlichen Kopfposen und Beleuchtungen aufgenommenen Trainingsbildern und Testbildern durchzuführen. Zhang et al. [Zha+15] nutzen verfügbare *State of the Art*-Methoden zur Kopfposenschätzung und zur Bestimmung von Gesichtsmerkmalen und trainieren mit diesen ein *Convolutional Neural Network*. Hierbei verwenden sie Bildausschnitte von Augen, die unter verschiedenen Kopfposen aufgenommen wurden. Ihre berichteten Ergebnisse auf dem herausfordernden gesamten *UT*-Datensatz betragen einen Fehler von 6°, mit einem *Leave-One-Person-Out*-Training, wobei Trainings- und Testdatensatz jeweils der *UT*-Datensatz ist, während ihre

Genauigkeit auf dem *UT*-Multiview-Datensatz mit einem speziell für eine Person trainierten Modell etwa $1{,}2^\circ$ beträgt, womit sie im Vergleich mit anderen *State of the Art*-Methoden laut der Veröffentlichung um einen Faktor zwei genauer sind.

5.2.2 Produkte

Obwohl kommerzielle Systeme existieren, mit Hilfe derer die Kopfpose bzw. die Blickrichtung bestimmt werden kann, werden die benötigten Informationen bei diesen Systemen unter Randbedingungen gewonnen, die eine breite Anwendung nur schwer oder gar nicht ermöglichen und somit bezüglich der Zielsetzung dieser Arbeit als Nachteile klassifiziert werden. Zu den Nachteilen gehören:

- Obstruktive Verfahren: Durch Blockieren der Sicht zur realen Welt, beispielsweise bei Verwendung von *Head Mounted Devices*, welche ein opakes Visier haben [Viv].
- Obstruierende Verfahren: Das zusätzliche Tragen einer Brille oder *Head Mounted Device* mit durchsichtigem Visier, welches zwar weiter Blickfreiheit zur realen Welt gewährleistet, aber dennoch als störend empfunden werden kann. Solche Verfahren entsprechen einer weiteren Hardwarevoraussetzung zur Blickrichtungssschätzung. Hierzu gehören die prominenten Beispiele *tobii Glasses* [toba] sowie die *Virtual Reality* Brille *Meta 2* [met].
- Präzise, aber teure (siehe Übersicht in [imo]) und auf aktiver Beleuchtung beruhende Systeme, wie etwa *tobii Spectrum* [tobb].

Es soll nun beschrieben werden, wie eine berührungslose Mensch-Maschine-Interaktion allein auf Hardwarebasis einer Webcam geschehen kann.

5.3 Bestimmung der Blickrichtung

Um die Blickrichtungsschätzung erscheinungsbasiert zu gestalten, werden die in den vorangegangenen Kapiteln beschriebenen Algorithmen eingesetzt, um die Informationen zu gewinnen, die Rückschlüsse auf

die Blickrichtung zulassen. Wie in den Abb. 1.1 bzw. 5.1 skizziert, entsprechen diese Informationen den Positionen der Iriden (Kap. 3) sowie der Kopfpose (Kap. 4), welche unter Verwendung der vorgestellten Methoden bestimmt werden.

Die Gliederung (siehe Abb. 1.2) der Arbeit ist die Grundlage des Arbeitsablaufes beim globalen Vorgehen zur Blickrichtungsschätzung. Die Blickrichtungsschätzung selbst gliedert sich in 2 Schritte:

1. Bestimmung einer Abbildung der Information aus Kopfpose und Irisposition (Kalibrierung).

2. Anwenden der gelernten Abbildung und Schätzen der Blickrichtung.

5.3.1 Schritt 1: Kalibrierung

Beim hier verfolgten Ansatz werden zu Beginn des Kalibriervorgangs zunächst Daten gesammelt, mit Hilfe derer eine Kalibriermatrix erstellt wird. Hierzu wird online über eine am Rechner angeschlossene Webcam [Log] ein Videostream geöffnet, welcher den Nutzer, der berührungslos interagieren will, aufzeichnet.

5.3.1.1 Aufnahme der Daten

Der Benutzer verfolgt nun mit seinen Augen unter natürlichen Bewegungen des Kopfes die Position des Mauszeigers. Gleichzeitig wird – als hier angenommene *Ground Truth* – die Kopfpose und die Iridenpositionen synchron mit der Mauszeigerposition aufgenommen und für den Videostream gespeichert. Im Detail wird für jeden zur Kalibrierung zu verwendenden Videostream das Folgende durchgeführt:

- Basierend auf einer Detektion der Augen werden innerhalb der *Bounding Box* der Augen, welche sich aus der Detektion ergibt, mittels des Verfahrens zur Irislokalisation die Positionen der Iriden bestimmt. Mit Hilfe anthropometrischer Werte wird dann das Kopfmodell (Zylinder) automatisch initialisiert:
  - Abstand Augen (Autor: $64\,\mathrm{mm}$).

  - Höhe und Radius des Zylinders als Vielfache des Augenabstandes.

- Für den Videostream wird nun am Ende jedes Frames die sechsdimensionale Kopfpose $\mathbf{p}$ abgespeichert. Die Positionen der Iriden werden als 3D-Punkte durch Projektion der 2D-Lokalisierungen auf die Zylinderoberfläche mit Hilfe einer inversen Lochkameraabbildung gespeichert.
- Gleichzeitig werden *Grount Truth*-Daten der Blickrichtung aufgenommen, indem mittels einer Windows-API [Sim], welche Zugriff auf die Kontrolle der Ein- und Ausgabegeräte erlaubt, die Position des Mauszeigers fortwährend gespeichert wird.
- Der Vorgang kann nun für mehrere Kalibriervideos wiederholt werden. Das Verwenden mehrerer Kalibriervideos kann herangezogen werden, um den Kalibrierraum, der, im Falle des Beispiels der Blickrichtungsschätzung vor einem Monitor, als der dreidimensionale Raum vor dem Monitor verstanden werden kann, in dem sich der Nutzer bewegt, abzudecken.

Gebildet mit der sechsdimensionalen Kopfpose $\mathbf{p}$ sowie den beiden 3D-Irispunkten steht nun ein 12D-Vektor $\mathbf{v}$ für jedes Frame zusammen mit seinen 2D-Mauszeigerdaten zur Verfügung.

5.3.1.2 Bestimmen der Kalibriermatrix

Die Herausforderung der erscheinungsbasierten Blickrichtungsschätzung wird in dieser Arbeit als Regressionsproblem aufgefasst. Hierbei soll ein Zusammenhang

$$\mathbf{u}^{\mathrm{M}} = \mathbf{A}\mathbf{v} \tag{5.2}$$

gelernt werden, wobei $\mathbf{A}$ eine 2×12-Matrix mit den Zeilenvektoren $\boldsymbol{a}$ und $\boldsymbol{b}$ ist.

Aus den Trainingsdaten werden nun entsprechend

$$u_n^{\mathrm{M}} = \sum_j a_j v_{j,n}\,, \quad v_n^{\mathrm{M}} = \sum_j b_j v_{j,n} \tag{5.3}$$

die Parameter a_j, b_j mittels multipler linearer Regression auf Basis von $n = 1, \ldots, N$ aufgenommenen Frames, bestimmt, wobei auf einen Gleichanteil im Modell verzichtet wurde.

Nach einem Kalibriervorgang steht die Matrix $\mathbf{A}$ zur Verfügung und kann zur Durchführung der monokularen Blichrichtungsschätzung zu einem beliebigen Zeitpunkt verwendet werden und muss nicht neu bestimmt werden. Die Mensch-Maschine-Interaktion kann nach Abschluss der Kalibrierung für die bei der Kalibrierung abdeckten Randbedingungen (Ausfüllung des Raumes) durchgeführt werden.

5.3.2 Schritt 2: Blickrichtungsschätzung

In der Anwendungen kann nun das ursprüngliche Programm zur Aufnahme der Kalibrierdaten umfunktioniert werden, sodass nicht die Mauspositionen aufgenommen werden, sondern durch die Windows-API der Mauszeiger bewegt wird, indem online der Vektor $\mathbf{v}$ geschätzt wird und durch Anwendung von Glg. (5.2) in jedem Frame die Position des Mauszeigers *berührungslos* gesteuert wird.

5.4 Ergebnisse

Um die in dieser Arbeit erforschten Methoden zur Blickrichtungsschätzung zu evaluieren, wurde zunächst ein *Proof of Concept*-Experiment durchgeführt, dessen Versuchsaufbau und Ergebnisse im Folgenden detailliert diskutiert werden, während in Abschnitt 5.5.1 der Einfluss der Irisschätzung auf die Blickrichtungsschätzung untersucht wird.

In Abschnitt 5.5.2 werden weitere Experimente diskutiert, wobei insbesondere der Einfluss von Kalibrierdaten, die in unterschiedlichen Sitzungen zeitlich versetzt aufgenommen wurden, auf deren Generalisierungsfähigkeit bezüglich unterschiedlicher Sitzungen sowie deren Auswirkung auf die Genauigkeit der Blickrichtungsschätzung untersucht wird.

5.4.1 Diskussion der Kalibrierdaten

Die Abdeckung des Kalibrierraumes spielt für das Ergebnis eine wichtige Rolle, da die Regression nach Glg. (5.2) nicht geeignet ist, um aus den Trainingsdaten zu extrapolieren. Daher sollte sich der online auf die Blickrichtung abzubildende 12D-Vektor $\mathbf{v}$ innerhalb des 12D-Kalibrierraumes zwischen Punkten, die während der Kalibrierung gelernt wurden, befinden. Abbildung 5.2 zeigt diesen Zusammenhang für eine Beispielkalibrierung anhand der Projektion der Kalibrierdaten in den 3D-Unterraum der Kopfposition (Translationsanteil der Pose). Der Ursprung des x-, y-, z-Koordinatensystems liegt in der Webcam, welche auf dem Monitor platziert ist. Die Kalibrierdaten einzelner Videos sind durch Ellipsoiden repräsentiert, deren Position im Raum durch den Mittelwert der Kopfposition während der Aufnahme dargestellt ist und deren Halbachsen die entsprechenden Standardabweichungen der Positionsänderung zeigen. Während die in Grautönen gezeigten Ellipsoiden zum Erstellen der Kalibriermatrix genutzt wurden, stellt der rote Ellipsoid (a posteriori) die Position des Testvideos im 3D-Unterraum dar. Für die Trainingsvideos ist durch den Grauton die Tiefe (Abstand zum Sensor) kodiert, wobei der Grauton zum Sensor hin dunkler wird.

5.5 Quantitative Auswertung

Abbildung 5.3 zeigt das Ergebnis der erscheinungsbasierten Blickrichtungsschätzung. In Grün ist die mit dem Mauszeiger aufgenommene *Ground Truth* für das Testvideo zu sehen, in Rot das Ergebnis der Schätzung. Der mittlere Fehler der Blickrichtungsschätzung beträgt $2{,}2^\circ$ in u- bzw. x-Richtung und $2{,}5^\circ$ in v- bzw. y-Richtung. Man erkennt, dass trotz nicht vollständiger räumlicher Umschließung der Testdaten (siehe Abb. 5.2) ein Fehler im Bereich von etwa $\frac{1}{20}$ der Bildschirmgröße erreicht wird. Der mittlere absolute Fehler beträgt in horizontaler Richtung 104, in vertikaler Richtung 117 Pixel, bei einer Standardabweichung von 86 bzw. 71 Pixel.

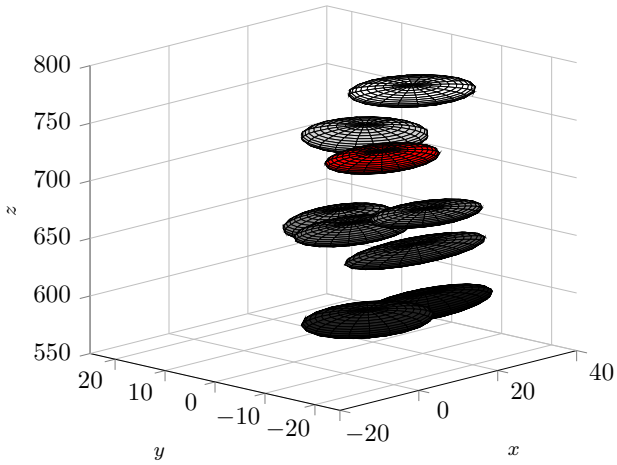


Abbildung 5.2 3D-Unterraum der Kopfpositionen des 12D-Kalibrierraumes für die verwendeten Videodaten zum Training und zum Testen (*Proof of Concept*-Experiment, Sitzung 1). Die Achsen geben die Entfernungen vom Mittelpunkt des Sensors in mm an, wobei die z-Richtung den Tiefenabstand in mm kennzeichnet.

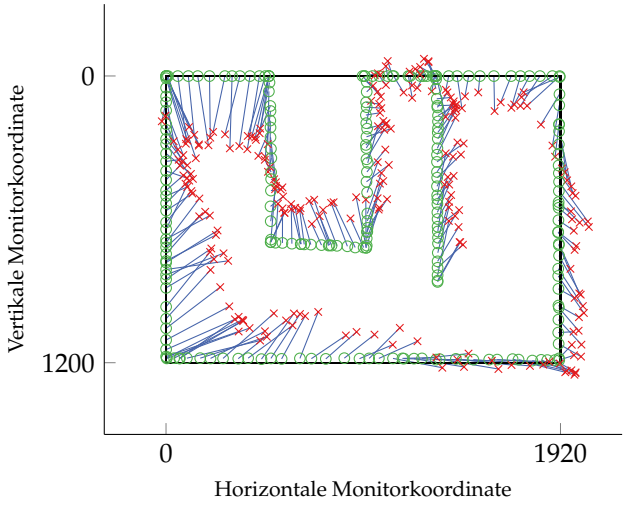


Abbildung 5.3 Visualisierung der Auswertung der Blickrichtungsschätzung anhand eines beispielhaften Testvideos mit den Kalibrierdaten illustriert in Abb. 5.2.

Auffällig ist der lokal stark variierende Betrag des Fehlers, wobei im Bereich der Bildschirmkoordinaten der rechten Hälfte eine visuelle Inspektion einen kleineren Fehler erkennen lässt. Tatsächlich beträgt der Fehler in diesem Bereich $1{,}9^\circ$ (horizontal) bzw. $1{,}9^\circ$ (vertikal). Dies lässt vermuten, dass die Abbildung $\mathbf{A}$ auf den linken Monitorbereich unzureichend bei der Kalibrierung abgedeckt wurde und die eigentliche Genauigkeit durch weitere Kalibrierung deutlich erhöht werden kann, da angenommen werden kann, dass die Algorithmen zur Kopfposenschätzung sowie zur Irislokalisation innerhalb des gleichen Kalibrierraumes unabhängig vom betrachteten Monitorsegment ähnliche Genauigkeiten liefern. Eine weitere Fehlerquelle kann darin liegen, dass bei Aufnahme des Testvideos die *Ground Truth* nicht gleichermaßen genau aufgenommen wurde, indem bei der Aufnahme in einigen Bereichen beispielsweise der Mauszeiger nicht exakt verfolgt wurde.

Abbildung 5.8 zeigt einige Frames des Testvideos zusammen mit der, durch einen Zylinder modellierten, geschätzten Kopfpose sowie den auf die Zylinderoberfläche projizierten Irispositionen. Der grüne Bereich des Zylinders entspricht dem durch die Kamera sichtbaren, der rote dem unsichtbaren Bereich. Die Iriden sind durch gelbe Kreuze dargestellt.

5.5.1 Einfluss der präzisen Irislokalisation

Um den bereits schematisch in Abb. 5.1 skizzierten Einfluss der Irislokalisation auf das Blickrichtungsschätzungsergebnis zu untersuchen, wurde dem Testvideo nun für die Positionen der Augen weißes Rauschen mit einer Standardabweichung von einem Pixel additiv hinzugefügt. Das Ergebnis verschlechterte sich auf $2{,}8^\circ$ (vorher $2{,}2^\circ$) in horizontaler Richtung und auf $2{,}8^\circ$ (vorher $2{,}5^\circ$) in vertikaler Richtung, für Rauschen mit zwei Pixeln Standardabweichung auf $3{,}9^\circ$ und $3{,}6^\circ$, siehe Tabelle in Abb. 5.4.

Auf der rechten Seite in Abb. 5.4 sind der MAE (ausgefüllte Kästen) sowie die Standardabweichung (nicht ausgefüllte Kästen) für den Einfluss von Rauschen auf die Iris- sowie Kopfposenschätzung dargestellt, wobei die Höhe und Breite mit dem doppelten des jeweiligen MAE bzw. der Standardabweichung gewählt wurde. Es werden die Fälle unterschieden:

- kein Rauschen (innen, hell),
- 2 Pixel Rauschen bei der Irislokalisation (mittlere Grauwerte),
- 2 Pixel Rauschen bei der Irislokalisation und 1° Rauschen auf die Kopfposenschätzung (außen, schwarz).

σ Rauschen Iris	0	1	2
MAE Horizontal	$2{,}2^\circ$	$2{,}8^\circ$	$3{,}9^\circ$
MAE Vertikal	$2{,}5^\circ$	$2{,}8^\circ$	$3{,}6^\circ$

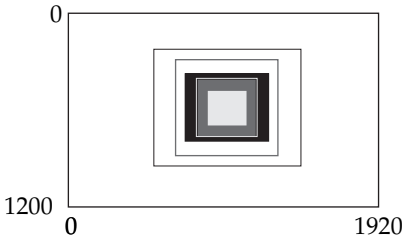


Abbildung 5.4 Einfluss eines der Irislokalisation (Tabelle) bzw. der Irislokalisation und der Kopfposenschätzung (Grafik) additiv hinzugefügten normalverteilten Fehlers auf die Blickrichtungsschätzung.

Es ist dabei zu bemerken, dass beim durchschnittlichen Abstand Sensor–Benutzer des hier betrachteten Szenarios von etwa $718\,\mathrm{mm}$ bei einer Bildgröße von 800×600 Pixeln der Durchmesser der Iris etwa 15 Pixel beträgt, der Radius der Iris entsprechend 7,5 Pixel und der Radius der Pupille etwa 3,3 Pixel, was einem Fehlerradius nach Glg. (3.14) in Kap. 3 von $\epsilon_{\mathrm{MAX}} \leq 0{,}05$ entspricht. Der mittlere absolute Fehler für den Fall von 2 Pixeln Rauschen steigt dabei auf 174 und 158, die Standardabweichung auf 135 und 115 Pixel.

5.5.2 Weitere Experimente

Es wurde zunächst zusätzlich zum in Abb. 5.2 gezeigten Kalibrierprozess ein weiterer Kalibrierprozess durchgeführt. Ziel hiervon ist es, die Robustheit und Übertragbarkeit der Kalibrierdaten sowie deren Auswirkungen auf die Genauigkeit der vorgestellten Methode abhängig vom Zeitpunkt der Aufnahme oder Anwendung der Daten zu untersuchen.

Abbildung 5.5 zeigt den 3D-Unterraum des 12D-Kalibrierraumes einer zweiten Sitzung, welche einige Wochen nach der Aufnahme der Sitzung

1 erfolgte. Die neuen Kalibrierdaten sind in Blau gezeigt, während zusätzlich einige Kalibrierdaten aus dem ersten Experiment in Grautönen dargestellt sind.

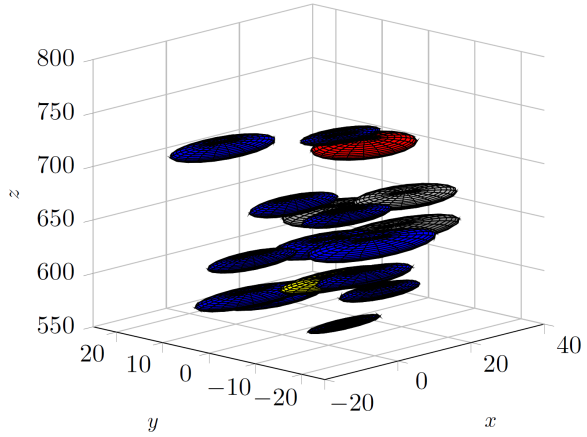


Abbildung 5.5 3D-Unterraum der Kalibrierdaten der zweiten Sitzung (blaue Ellipsoiden) und deren Testdaten (gelbes Ellipsoid). Weiterhin sind in Grautönen und Rot Daten aus der ersten Sitzung eingezeichnet.

5.5.2.1 Quantitatives Ergebnis zweite Sitzung

Die zweite Sitzung wurde mit mehr Daten, insgesamt 15581 Frames, kalibriert (vgl. Sitzung 1: 3872 Frames). Anschließend wurde zunächst ein Video ausgewertet, welches ebenfalls innerhalb dieser zweiten Sitzung aufgenommen wurde und in Abb. 5.5 in Gelb skizziert ist. Tabelle 5.1 zeigt in der dritten Spalte quantitativ das Ergebnis der Blickrichtungsschätzung, während dieses Ergebnis in Abb. 5.6 auf Basis der Bildschirmgröße visualisiert ist.

Man erkennt, dass die Genauigkeit der Blickrichtungsschätzung in der zweiten Sitzung mit $1{,}2^\circ$ insbesondere in horizontaler Richtung höher ist und mit einem MAE von 46 Pixeln einem Fehler kleiner als $\frac{1}{40}$ der Bildschirmauflösung entspricht.

Tabelle 5.1 Übersicht zur quantitativen Auswertung unterschiedlicher Kalibrierdaten und Testvideos.

Kalibr. (Sitzung)	1	Abb. 5.7	2	Abb. 5.5	1 und 2
Test (Sitzung)	1 (Rot)	1 (Rot)	2 (Gelb)	2 (Gelb)	2 (Gelb)
MAE Horizontal	2,2°	2,5°	1,2°	1,2°	1,2°
MAE Vertikal	2,5°	2,6°	2,7°	2,7°	2,9°

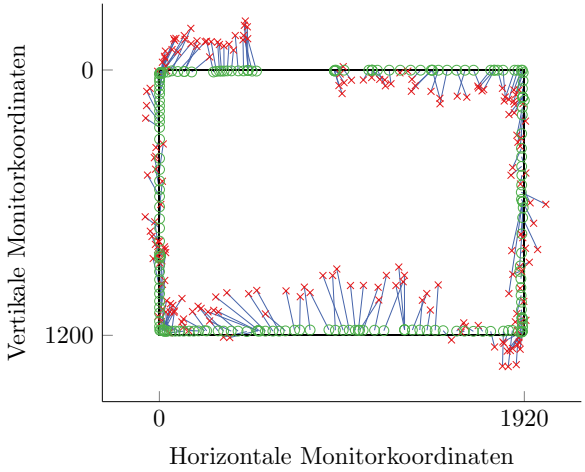


Abbildung 5.6 Visualisierung der quantitativen Auswertung der Blickrichtungsschätzung anhand des Testvideos aus Sitzung 2 (gelbes Ellipsoid) mit den Kalibrierdaten aus Sitzung 2 (blaue Ellipsoiden in Abb. 5.5).

5.5.2.2 Quantitative Ergebnisse mit unterschiedlichen Sitzungen

Um die Robustheit der Kalibrierdaten zu testen, wurden Experimente mit fusionierten Daten aus beiden Sitzungen gemacht. Abbildung 5.7 zeigt ergänzend zu Abb. 5.5 einige Kalibrierdaten der zweiten Sitzung im Kalibrierraum der ersten Sitzung.

Die Untersuchung der Blickrichtungsschätzung – mit Daten bei denen Kalibrier- und Testvideos aus unterschiedlichen Sitzungen zusammengeführt sind – ist in den Spalten 2, 4 und 5 in Tabelle 5.1 zusammengefasst.

Die Ergebnisse zeigen eine gute Übertragbarkeit von Kalibrierdaten zwischen Sitzungen. Das Anwenden der Kalibrierdaten auf die Testvideos der jeweils anderen Sitzung zeigt mit Veränderungen im Bereich von $0{,}1^\circ$ bis $0{,}3^\circ$ vergleichbare Ergebnisse. Die fünfte Spalte zeigt das Ergebnis der Schätzung bei Berücksichtigung aller Kalibrierdaten auf das Testvideo aus Sitzung 2, wobei unter gleichzeitiger Erweiterung des Kalibrierraumes ein geringfügig höherer Fehler von $0{,}2^\circ$ in vertikaler Richtung zu beobachten ist. Die Ergebnisse lassen sich insgesamt damit erklären, dass der Kalibrierraum auch durch Zusammenführen der Daten nicht ideal abgedeckt ist, während die Regression über einen größeren Kalibrierbereich durchgeführt werden muss. Weiterhin ist keine Beurteilung der Güte der einzelnen Kalibrierdaten eingegangen, sodass durch Hinzunahme weiterer, eventuell ungeeigneter, Kalibrierdaten nicht notwendigerweise eine Verbesserung erwartet werden kann.

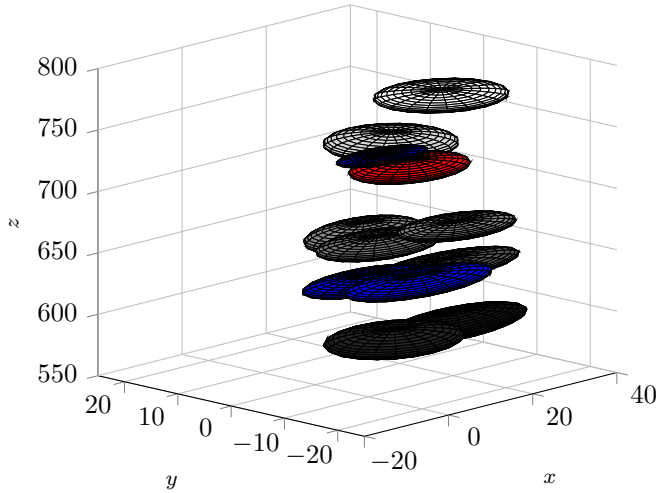


Abbildung 5.7 Der 3D-Unterraum der Kalibrierdaten der ersten Sitzung erweitert um Daten der zweiten Sitzung (blaue Ellipsoiden).

Abbildung 5.8 Acht Frames aus dem Testvideo (Rot) aus Abb. 5.2 sowie die Kopfposen- und Irisschätzung. In Frame 202 ist darüber hinaus ein *Template Update*-Prozess an den durch am linken Rand des Zylinders dargestellten kleinen gelben Kreuzen zu erkennen. Sie stellen neu in das *Template* aufgenommene Pixelregionen dar.

5.6 Zusammenfassung

Es wurde gezeigt, wie die in dieser Arbeit vorgestellten Methoden zur Augendetektion, zur Irislokalisation und zur Kopfposenschätzung zusammen mit einem Regressionsansatz in einem Programm zur monokularen Blickrichtungsschätzung kombiniert werden können.

Der Zusammenhang zwischen den Schätzergebnissen von Iris und Kopfpose und der Blickrichtung wurde mittels multipler linearer Regression gelöst. Es wurden folgende Beiträge geliefert:

- Es wurde ein Rahmenwerk implementiert, das die Methoden der Kapitel 2, 3 und 4 fusioniert und um ein Modul, welches das gleichzeitige Aufnehmen von Schätzdaten sowie der Mauszeigerposition erlaubt, erweitert.
- Ein Ansatz zur Abbildung der Schätzdaten auf den Blickpunkt auf den Monitor wurde basierend auf einer Regressionsanalyse implementiert.
- Entstanden ist ein Ansatz, der nach Abschluss eines Kalibrierungsprozesses fähig ist, online, durch monokulare Schätzung der Blickrichtung, den Mauszeiger entsprechend des Punktes von Interesse zu bewegen.

Die Auswertungen von Experimenten zur Bestimmung der Blickrichtungsschätzung ergaben Folgendes:

- Die Genauigkeit der Ergebnisse liegt für die betrachteten Experimente im Bereich des Standes der Technik.
- Es wird ein starker Einfluss der Abdeckung des Kalibrierraumes in Abhängigkeit des Ortes der durchgeführten Blickrichtung im 12D-Kalibrierraum auf das Schätzergebnis vermutet. Die Darstellung der Kalibrierdaten in 3D-Unterräumen in Form von Ellipsoiden wurde hierzu als hilfreiches Mittel zur Evaluierung identifiziert.
- Eine Untersuchung der Präzision der Irislokalisation auf die Genauigkeit des Schätzergebnisses wurde durchgeführt mit dem Ergebnis, dass kleine Abweichungen von nur einem Pixel starke Auswirkungen auf das Ergebnis haben. Dies unterstützt die

Annahme, dass für den hier verfolgten Ansatz zur Blickrichtungsschätzung eine hochpräzise Irislokalisation eine wichtige Rolle spielt.

- Es wurde die Übertragbarkeit von Kalibrierdaten zwischen unterschiedlichen Sitzungen, welche zeitlich auseinander liegen, gezeigt. Dies beschreibt die Möglichkeit des Anwendens einer zuvor bestimmten Kalibriermatrix zur Blickrichtungsschätzung zu einem späteren Zeitpunkt ohne erneute Kalibrierung.

6 Schluss

Während die einzelnen Kapitel mit ihren Kernaussagen lokal an ihrem jeweiligen Ende zusammengefasst wurden, sollen die erzielten Forschungsergebnisse abschließend, den Gesamtkontext der Arbeit betrachtend, global herausgestellt werden.

6.1 Zusammenfassung des wissenschaftlichen Beitrags

Wie bereits zu Anfang in Abb. 1.2 skizziert, besteht der Fokus der vorgelegten Arbeit in der Erforschung von Methoden, die eine monokulare Blickrichtungsschätzung, als Kernaufgabe der berührungslosen Mensch-Maschine-Interaktion, ermöglichen. Mit der Zielsetzung eines erscheinungsbasierten Ansatzes wurden in diesem Zusammenhang Verfahren erforscht, welche Informationen über den Kopf und die Augen bereitstellen, mit Hilfe derer sich durch Fusion in einem Regressionsansatz eine Blickrichtungsschätzung basierend auf der Verwendung einfacher Hardware durchführen lässt.

Es wurde hierzu in dieser Arbeit ein Blickrichtungsvektor aus den dreidimensionalen Positionen der Iriden sowie der sechsdimensionalen Kopfpose bestimmt. Da das endgültige System zur Blickrichtungsschätzung als Zusammenspiel einzelner Module aufgefasst werden kann, wurde die vorliegende Arbeit entsprechend modular aufgebaut, und es wurden sowohl innerhalb der Module als auch modulübergreifend den Stand der Technik erweiternde Beiträge geliefert und gekennzeichnet.

Als Ausgangspunkt zur Interaktion mit einem Nutzer wurde ein Rahmenwerk zum Trainieren eines Kaskadenklassifikators sowie zur Anwendung eines solchen Klassifikators zur Detektion von Augen implementiert. Es wurde mit dem Verfahren des *Merkmal-Baggings* eine

neuartige Methode vorgestellt, die das Training verschiedener Merkmalstypen ohne Verzicht auf die Auswahl aus der vollständigen, durch Skalierungen und Translationen innerhalb des Trainingsfensters kreierbaren, Merkmalsmenge erlauben. In diesem Zusammenhang wurde eine effiziente, durch *Boosting* einzelne dominante Merkmalsdimensionen bevorzugende, und dadurch mehrdimensionale Merkmale automatisch an die Klassifikationsaufgabe anpassende, Methode präsentiert, welche die sonst notwendige empirische Parametrierung mittels *Merkmals-Boosting* automatisiert und die manuelle Auswahl durch ein statistisches Lernverfahren ersetzt.

Als wesentlicher Bestandteil der erscheinungsbasierten Blickrichtungsschätzung wurde ein pixelgenaues Verfahren zur Lokalisierung der Irisposition vorgeschlagen. Unter der Annahme der Präsenz von Linien konstanter Intensität wurden auf Basis der örtlichen Radien dieser Linien Verschiebungsvektoren bestimmt, welche zum Zentrum der Pupille zeigen. Durch Berücksichtigung lokaler Bildmerkmale wie der Rundheit sowie des innerhalb eines durch Schätzung des lokalen Radiusses gewonnenen Zielgebietes bestimmten lokalen Grauwertes konnte eine diskriminative Gewichtung der stark mit Fehlinformationen behafteten Verschiebungsvektoren durchgeführt werden. Es wurde gezeigt, wie durch Modifikation der Kaskadenklassifikatorimplementierung dessen konventioneller, binärer Ausgang in quasi-kontinuierliche Werte gewandelt und diese nach Abbildung durch eine geeignete Funktion auf einen normierten Wertebereich mit dem Schätzergebnis fusioniert werden kann. Das Ergebnis ist ein robuster, präziser Algorithmus zur Lokalisierung der Iriden, dessen Genauigkeit bei Auswertung auf dem repräsentativen BioID-Datensatz nach bester Kenntnis des Autors andere Verfahren der Literatur übertrifft.

Um die 3D-Kopfpose in Echtzeit kontinuierlich zu schätzen, wurde der Ansatz der Berechnung des optischen Flusses aufgegriffen. Zur Stabilisierung der Kopfposenschätzung bei starken Gier- und Nickbewegungen sowie zur Erhöhung der Genauigkeit wurde der in der Literatur verbreitete, skalarwertige Regularisierungsansatz zur Konditionierung der Hessematrix auf einen vektorwertigen Ansatz erweitert. Unter Zuhilfenahme einer auf der Verfolgung stabiler Merkmale basierenden unabhängigen Bewegungsschätzung des Kopfes konnte gezeigt werden,

wie der neuartige Regularisierungsterm online bewegungsadaptiv den optischen Fluss regularisiert und somit diesen in dominanter Richtung bevorzugt, wodurch sich eine Verbesserung der Genauigkeit auf dem Boston University-Datensatz von 20 % ergeben hat.

Abschließend wurden die den Stand der Technik erweiternden Implementierungen der präsentierten Methoden in einem gemeinsamen Programm zur Schätzung der Blickrichtung zusammengefasst. Es wurde ein ergänzender Ansatz vorgestellt, mit Hilfe dessen Daten zur Bestimmung einer Kalibriermatrix, welche die aus den erscheinungsbasierten Methoden gewonnenen Information auf den Punkt des Interesses auf den Monitor abbildet, aufgenommen und weiterverarbeitet werden können. Als Resultat liegt ein Programm vor, mit welchem der Mauszeiger mit einer während eines *Proof of Concept*-Experimentes entstandenen Kalibriermatrix mit einer Genauigkeit im Bereich des Standes der Technik ausschließlich durch Bewegung des Kopfes und der Augen bewegt werden kann, ohne auf aufwendigere Hardware als eine Webcam angewiesen zu sein.

6.2 Weitere Forschungsrichtungen

Es sollen kurz Ideen und Ansätze, die während der Ausarbeitung der Beiträge und während des Zusammenschreibens entstanden sind, genannt werden:

- Durch den Gewinn an freiem Speicher durch das *Merkmals-Bagging* kann eine Kaskade mit einer größeren Anzahl an Trainingsbeispielen trainiert werden, was die Detektionsgüte erhöhen sollte.
- Es sind weitere Auswertungen möglich, hierzu gehören:
  - Training mit nur linken Augen / rechten Augen / Brille;
  - Auswertung der Genauigkeit der Irislokalisierung mittels Kaskadenklassifikator und Vergleich mit Ergebnissen aus der Literatur, die einen ähnlichen, trainingsbasierten Ansatz verfolgen;
  - Auswerten des LBP-Merkmals für *mean* und *CS* in Kombination mit anderen Merkmalen;

- Nutzen anderer Kaskaden zur Kombination mit der Irislokalisierung;
- Umfangreiche Auswertung der Irislokalisierung für unterschiedliche Blickrichtungen;
- Stabilisierung der Kopfposenschätzung mittels Augendetektion (Re-Initialisierung) und Irislokalisation (Korrektur der Pose);
- Umfangreiche Experimente zur Kalibrierung der Kopfpose und Auswerten der Genauigkeit der Blickrichtungsschätzung;
- Ausarbeiten einer Idee zum berührungslosen Bedienen der Maustaste.

Anhang

A Anhang

A.1 Implementierungen

Abschließend sollen kurz weitere Ergebnisse und Implementierungen, die im Laufe der Arbeit entstanden sind und keine direkten neuen Beiträge zum Stand der Wissenschaft liefern, allerdings als Rahmenwerke und Basis für Implementierungen sowie Vergleiche mit bestehenden Algorithmen genutzt wurden und zur Verfügung stehen, beschrieben werden. Einige der im Folgenden erwähnten Implementierungen sind zudem auch für die Erstellung der Ergebnisse dieser Arbeit zum Einsatz gekommen.

Es wurden nicht weniger als drei vollständig funktionierende Rahmenwerke, welche im Bereich des Computersehens sowie des maschinellen Lernens in der letzten Dekade den Stand der Technik dargestellt haben, von Grund auf implementiert.

A.1.1 Implementierungen des Standes der Technik

Bei allen Implementierungen wurde auf keine Vorarbeiten zurückgegriffen und externe Module und Routinen, die verwendet wurden, sollen gekennzeichnet werden.

Zu den externen Modulen gehören:

- OpenCV-Routinen zur Berechnung von Gradienten und weitere, vorverarbeitende Methoden, Anwenden von *Support Vector Machines*.
- mexOpenCV, Grundstruktur des *Piece-Wise* affinen Warps (nutzbar für ein isoliertes Polygon – erweitert auf ein *Delaunay-Mesh* im Rahmen dieser Arbeit).

- eine MATLAB-Datei zur Berechnung von LBP-Merkmalen (eigene C++ Implementierung).

Inbesondere wurde der Stand der Technik für folgende Algorithmen implementiert:

- Kaskadenklassifikator, basierend auf der Arbeit von Viola und Jones [VJ01], inklusive:
  - Integrale Bilder (sowohl für Grauwerte (Haar) als auch Gradienten, LBP) in C++,
  - Verschiedene *Boosting*-Algorithmen (*AdaBoost*, *Real-*, *Gentle-*, *Forward Feature Selection*),
  - Implementierung diverser Mermale (LPB-Grundmerkmalsberechnung mit zwei MATLAB-Dateien von Ahonen [AHP06]) in C++,
  - *Bootstrap-Aggregating*.
- Kopfposenschätzung mittels optischem Fluss basierend auf Xiao et al. [Xia+03]:
  - Bildhomogenisierung ersten und zweiten Grades in C++,
  - Vollständige Berechnung des optischen Flusses als *Forward Additive-*, *Forward Compositional-* und *Inverse Compositional-*Algorithmus,
  - Regularisierung der Hessematrix,
  - Schätzung der Kopfbewegung basierend auf der Verfolgung stabiler Merkmale (MATLAB-Funktionen für: *SURF*, RANSAC),
  - Bestimmen der Bewegungsmatrix aus der Homographiematrix (Verwenden der MATLAB `step`-Funktion).
- Modellbildung und Bildsuche für AAM ([BM04], [CET01]) (nicht Bestandteil dieser Arbeit):
  - Tools zum Landmarking und Daten sammeln,
  - Prokrustes-Analyse, Umsetzen der *PCA* in MATLAB zum Extrahieren der Eigenvektoren aus der Kovarianzmatrix,

  - Modellbildung zum einen durch Regression sowie auch mit Hilfe des *Inverse Compositional Algorithmus*,
  - Fitting mit beiden Modellen,
  - *Piece-Wise* affiner Warp mittels *Delaunay*-Triangulation (C++ Erweiterung der OpenCV-Routine),
  - Erweiterungen zum Stand der Technik für AAM durch Fusion mit präziser Irislokalisation [VIP17].

- Berechnungen aller Methoden wurden auf Bildpyramiden durchgeführt.

Literaturverzeichnis

[Abr+03] **Abrahama, D. P., Liua, J., Chena, C. H., Hyunga, Y. E., Stolla, M., Elsena, N., MacLarenb, S., Twestenb, R., Haaschb, R., Sammannb, E., Petrovb, I., Aminea, K. und Henriksen, G.** *Diagnosis of power fade mechanisms in high-power lithium-ion cells*. In: *Journal of Power Sources* 119-121 (2003), S. 511–516.

[Ada+10] **Adams, Andrew, Talvala, Eino-Ville, Park, Sung Hee, Jacobs, David E., Ajdin, Boris, Gelfand, Natasha, Dolson, Jennifer, Vaquero, Daniel, Baek, Jongmin, Tico, Marius, Lensch, Hendrik P. A., Matusik, Wojciech, Pulli, Kari, Horowitz, Mark und Levoy, Marc**. *The Frankencamera: An Experimental Platform for Computational Photography*. In: *ACM Transactions on Graphics* 29.4 (2010), 29:1–29:12.

[Ado+09] **Adornato, B., Patil, R., Filipi, Z., Baraket, Z. und Gordon, T.** *Characterizing naturalistic driving patterns for Plug-in Hybrid Electric Vehicle analysis*. In: *Vehicle Power and Propulsion Conference, 2009. VPPC'09. IEEE*. IEEE. 2009, S. 655–660.

[AHP06] **Ahonen, Timo, Hadid, Abdenour und Pietikainen, Matti**. *Face description with local binary patterns: Application to face recognition*. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28.12 (2006), S. 2037–2041.

[All] **Allied Vision**. *Eye-Tracking hilft schwerbehinderten Menschen*. "https://www.alliedvision.com/de/news/detail/news/an-insight-into-eye-tracking-applications.html. Zuletzt zugegriffen am 31.10.17.

[Ama] **Amazon Developer**. *Amazon Alexa*. "https://developer.amazon.com/alexa. Zuletzt zugegriffen am 09.10.17.

[AC09] **An, Kwang Ho und Chung, Myung Jin**. *Robust real-time 3D head tracking based on online illumination modeling and its application to face recognition*. In: *International Conference on Intelligent Robots and Systems*. IEEE, 2009, S. 1466–1471.

[Appa] **Apple Inc. (US)**. *Apple Siri*. https://www.apple.com/ios/siri/. Zuletzt zugegriffen am 09.10.17.

[Appb] **Apple Inc. (US)**. *The iPhone X*. "https://www.apple.com/iphone-x/specs/. Zuletzt zugegriffen am 09.10.17.

[Ari+16] **Ariz, Mikel, Bengoechea, José J., Villanueva, Arantxa und Cabeza, Rafael**. *A novel 2D/3D database with automatic face annotation for head tracking and pose estimation*. In: *Computer Vision and Image Understanding* 148 (2016), S. 201–210.

[AS10] **Asadifard, Mansour und Shanbezadeh, Jamshid**. *Automatic adaptive center of pupil detection using face detection and CDF analysis*. In: *Proceedings of the International Multi Conference of Engineers and Computer Scientists*. Bd. 1. 2010, S. 130–133.

[Bae+13] **Baek, S. J., Choi, K. A., Ma, C., Kim, Y. H. und Ko, S. J.** *Eyeball model-based iris center localization for visible image-based eye-gaze tracking systems*. In: *IEEE Transactions on Consumer Electronics* 59.2 (2013), 415–421.

[BSW06] **Bai, L., Shen, L. und Wang, Y.** *A novel eye location algorithm based on radial symmetry transform*. In: *18th IEEE Int. Conf. Pattern Recognition*. Bd. 3. 2006, S. 511–514.

[Bai+03] **Bailly-Bailliére, Enrique, Bengio, Samy, Bimbot, Frédéric, Hamouz, Miroslav, Kittler, Josef, Mariéthoz, Johnny, Matas, Jiri, Messer, Kieron, Popovici, Vlad und Porée, Fabienne**. *The BANCA database and evaluation protocol*. In: *Audio- and Video-Based Biometric Person Authentication Conference*. Springer, 2003, S. 625–638.

[BM01] **Baker, Simon und Matthews, Iain**. *Equivalence and efficiency of image alignment algorithms*. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Bd. 1. IEEE, 2001, S. 1090–1097.

[BM04] **Baker, Simon und Matthews, Iain**. *Lucas-Kanade 20 years on: A unifying framework*. In: *International Journal of Computer Vision* 56.3 (2004), S. 221–255.

[BKC01] **Bang, S., Kim, D. und Choi, S.** *Asian Face Image Database PF01*. In: *Intelligent Multimedia Lab, Pohang University of Science and Technology* (2001). http://nova.postech.ac.kr.

[Bat+05] **Bates, R., Istance, H., Oosthuizen, L. und Majaranta, P.** *Survey of De-Facto Standards in Eye Tracking*. In: *Information Society Technology* (2005).

[BTG06] **Bay, Herbert, Tuytelaars, Tinne und Gool, Luc Van**. *Surf: Speeded up robust features*. In: *European Conference on Computer Vision*. Springer, 2006, S. 404–417.

[Bel+13] **Belhumeur, Peter N., Jacobs, David W., Kriegman, David J. und Kumar, Neeraj**. *Localizing parts of faces using a consensus of exemplars*. In: *IEEE transactions on pattern analysis and machine intelligence* 35.12 (2013), S. 2930–2940.

[Bio01] **BioID Technology Research**. *The BioID Face Database*. https://www.bioid.com/About/BioID-Face-Database. 2001.

[Bis06] **Bishop, Christopher M.** *Pattern Recognition and Maschine Learning*. 2006.

[Bla92] **Black, Michael Julian**. *Robust incremental optical flow*. Diss. 1992.

[Böh+06] **Böhme, Martin, Meyer, André, Martinetz, Thomas und Barth, Erhardt**. *Remote eye tracking: State of the art and directions for future development*. In: *Proceedings of the 2006 Conference on Communication by Gaze Interaction (COGAIN)*. 2006, S. 12–17.

[BM98] **Bregler, Christoph und Malik, Jitendra**. *Tracking people with twists and exponential maps*. In: *Proceedings of th eIEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1998, S. 8–15.

[Bri] **Briesemeister, Benny B.** *Wo der Blick hinfällt - Einsatz von Eyetracking im Marketing*. "https://de.ryte.com/magazine/wo-der-blick-hinfaellt-eyetracking-und-dessen-einsatz-fuers-marketing. Zuletzt zugegriffen am 31.10.17.

[CLL06] **Campadelli, Paola, Lanzarotti, Raffaella und Lipori, Giuseppe**. *Precise eye localization through a general-to-specific model definition*. In: *Proceedings on the British Machine Vision Conference*. Citeseer, 2006, 187–196.

[CLL09] **Campadelli, Paola, Lanzarotti, Raffaella und Lipori, Giuseppe**. *Precise eye and mouth localization*. In: *International Journal of Pattern Recognition and Artificial Intelligence* 23.03 (2009), 359–377.

[Che+16a] **Chen, Jiawei, Wu, Jonathan, Richter, Kristi, Konrad, Janusz und Ishwar, Prakash**. *Estimating head pose orientation using extremely low resolution images*. In: *2016 IEEE Southwest Symposium on Image Analysis and Interpretation*. IEEE, 2016, S. 65–68.

[CJ11] **Chen, Jixu und Ji, Qiang**. *Probabilistic gaze estimation without active personal calibration*. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, 609–616.

[Che+16b] **Chen, S., Liang, L., Liang, W. und Foroosh, H.** *3D pose tracking with multitemplate warping and SIFT correspondences*. In: *IEEE Transactions on Circuits and Systems for Video Technology* 26.11 (2016), S. 2043–2055.

[CCH10] **Chen, Zhih-Wei, Chiang, Cheng-Chin und Hsieh, Zi-Tian**. *Extending 3D Lucas–Kanade tracking with adaptive templates for head pose estimation*. In: *Machine Vision and Applications* 21.6 (2010), S. 889–903.

[CCD12] **Cherabit, Noureddine, Chelali, Fatma Zohra und Djeradi, Amar**. *Circular Hough transform for iris localization*. In: *Science and Technology* 2.5 (2012), S. 114–121.

[CET01] **Cootes, Timothy F., Edwards, Gareth J. und Taylor, Christopher J.** *Active appearance models*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.6 (2001), S. 681–685.

[CCS04] **Cristinacce, David, Cootes, Tim und Scott, Ian**. *A multi-stage approach to facial feature detection*. In: *Proceedings on the British Machine Vision Conference*. 2004, S. 1–10.

[DT05] **Dalal, N. und Triggs, B.** *Histograms of Oriented Gradients for Human Detection*. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Bd. 1. 2005, S. 886–893.

[DD95] **Dementhon, Daniel F. und Davis, Larry S.** *Model-based object pose in 25 lines of code*. In: *International Journal of Computer Vision* 15.1 (1995), S. 123–141.

[Der] **Dernbach, Christoph**. *Die Geschichte des iPod*. http://www.mac-history.de/apple-products/ipod/2008-06-14/die-geschichte-des-ipod. Zuletzt zugegriffen am 09.10.17.

[DZL11] **Dong, Yi, Zhen, Lei und Li, S. Z.** *A robust eye localization method for low quality face images*. In: *2011 International Joint Conference on Biometrics (IJCB)*. 2011, S. 1–6.

[DJ99] **Dowski Jr., Edward R. und Johnson, Gregory E.** *Wavefront Coding: A modern method of achieving high-performance and/or low-cost imaging systems*. In: *Current Developments in Optical Design and Optical Engineering VIII*. Hrsg. von **Fischer, Robert E. und Smith, Warren J.** SPIE - International Society for Optical Engineering, 1999, S. 137–145.

[EG09] **Enzweiler, M. und Gavrila, D. M.** *Monocular Pedestrian Detection: Survey and Experiments*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.12 (2009), S. 2179–2195.

[Ess] **Essex, University of**. *Essex face database*. http://cswww.essex.ac.uk/mv/allfaces/.

[FFM05] **Fasel, Ian, Fortenberry, Bret und Movellan, Javier**. *A generative framework for real time object detection and classification*. In: *Computer Vision and Image Understanding* 98.1 (2005), S. 182–210.

[FFP06] **Fei-Fei, Li, Fergus, Robert und Perona, Pietro**. *One-shot learning of object categories*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.4 (2006), S. 594–611.

[Fer11] **Ferguson, Kitty**. *Stephen Hawking: His Life and Work*. Random House, 2011.

[FS95] **Freund, Yoav und Schapire, Robert E.** *A desicion-theoretic generalization of on-line learning and an application to boosting*. In: *Computational learning theory*. Springer, 1995, S. 23–37.

[FHT00] **Friedman, Jerome, Hastie, Trevor und Tibshirani, Robert**. *Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)*. In: *The annals of statistics* 28.2 (2000), 337–407.

[Fuh+16] **Fuhl, Wolfgang, Tonsen, Marc, Bulling, Andreas und Kasneci, Enkelejda**. *Pupil detection for head-mounted eye tracking in the wild, An evaluation of the state of the art*. In: *Machine Vision and Applications* 27.8 (2016), S. 1275–1288.

[Gao+08] **Gao, Wen, Cao, Bo, Shan, Shiguang, Chen, Xilin, Zhou, Delong, Zhang, Xiaohua und Zhao, Debin**. *The CAS-PEAL large-scale Chinese face database and baseline evaluations*. In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 38.1 (2008), 149–161.

[GN17] **Gawande, Akshay A. und Nathaney, Gangotri**. *A Survey on Gaze Estimation Techniques in Smartphone*. In: (2017).

[GBK01] **Georghiades, Athinodoros S., Belhumeur, Peter N. und Kriegman, David J.** *From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.6 (2001), S. 643–660.

[Gir+14] **Girshick, Ross, Donahue, Jeff, Darrell, Trevor und Malik, Jitendra**. *Rich feature hierarchies for accurate object detection and semantic segmentation*. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, S. 580–587.

[Goo] **Google**. *Google Assistant*. https://assistant.google.com/. Zuletzt zugegriffen am 09.10.17.

[Gro+10] **Gross, Ralph, Matthews, Iain, Cohn, Jeffrey, Kanade, Takeo und Baker, Simon**. *Multi-PIE*. In: *Image and Vision Computing* 28.5 (2010), S. 807–813.

[HB98] **Hager, Gregory D. und Belhumeur, Peter N.** *Efficient region tracking with parametric models of geometry and illumination*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.10 (1998), S. 1025–1039.

[Ham+05] **Hamouz, Miroslav, Kittler, Josef, Kamarainen, J.-K., Paalanen, Pekka, Kalviainen, Heikki und Matas, Jiri**. *Feature-Based Affine-Invariant Localization of Faces*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.9 (2005), S. 1490–1495.

[HM82] **Hanley, James A. und McNeil, Barbara J.** *The meaning and use of the area under a receiver operating characteristic (ROC) curve*. In: *Radiology* 143.1 (1982), S. 29–36.

[HQ10] **Hansen, D. W. und Qiang, Ji**. *In the Eye of the Beholder: A Survey of Models for Eyes and Gaze*. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.3 (2010), S. 478–500.

[Haw] **Hawking, Stephen**. *Essay zur Entwicklung von Künstlicher Intelligenz*. "https://www.unlimited.world/unlimited/this-is-the-most-dangerous-time-for-our-planet. Zuletzt zugegriffen am 09.10.17.

[hei] **heise.de**. *Beitrag zum Einsatz von Industrierobotern in Deutschland*. "https://www.heise.de/newsticker/meldung/Deutschland-fuehrend-beim-Einsatz-von-Industrie-Robotern-3337763.html. Zuletzt zugegriffen am 09.10.17.

[Hil+10] **Hillaire, S., Breton, G., Ouarti, N., Cozot, R. und Lécuyer, A.** *Using a Visual Attention Model to Improve Gaze Tracking Systems in Interactive 3D Applications*. In: *Computer Graphics Forum* 29.6 (2010), S. 1830–1841.

[HL01] **Hjelmås, Erik und Low, Boon Kee**. *Face detection: A survey*. In: *Computer vision and image understanding* 83.3 (2001), S. 236–274.

[Hor86] **Horn, Berthold**. *Robot vision*. MIT press, 1986.

[htt] **https://scala.com**. "https://scala.com/. Zuletzt zugegriffen am 31.10.17.

[HW00] **Huang, Jeffrey und Wechsler, Harry**. *Visual Routines for Eye Location Using Learning and Evolution*. In: *IEEE Transactions on Evolutionary Computation* 4.1 (2000), S. 73–82.

[IMM] **IMM**. *IMM face database*. http://www2.imm.dtu.dk/~aam/.

[imo] **imotions.com**. *Übersicht zu kommerziellen Eye-Trackern und Preise*. "https://imotions.com/blog/eye-tracker-prices/. Zuletzt zugegriffen am 25.10.17.

[JAF] **JAFFE**. *The JAFFE database*. http://www.mis.atr.co.jp/ mlyons/jaffe.htm.

[JK08] **Jang, Jun-Su und Kanade, Takeo**. *Robust 3D head tracking by online feature registration*. In: *8th IEEE Internation Conference on Automatic Face and Gesture Recognition*. Citeseer, 2008.

[JKF01] **Jesorsky, O., Kirchberg, K. J. und Frischholz, R. W.** *Robust Face Detection Using the Hausdorff Distance*. In: *Proc. 3rd Int. Conf. Audio- and Video-Based Biometric Person Authentication*. Bd. 2091. Springer, 2001, S. 90–95.

[JY02] **Ji, Qiang und Yang, Xiaojie**. *Real-time eye, gaze, and face pose tracking for monitoring driver vigilance*. In: *Real-Time Imaging* 8.5 (2002), 357–377.

[Jon+00] **Jonathon Phillips, P., Moon, Hyeonjoon, Rizvi, Syed und Rauss, Patrick J.** *The FERET Evaluation Methodology for Face-Recognition Algorithms*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.10 (2000), S. 1090–1104.

[Jon+05] **Jonathon Phillips, P., Flynn, Patrick J., Scruggs, Todd, Bowyer, Kevin W., Chang, Jin, Hoffman, Kevin, Marques, Joe, Min, Jaesik und Worek, William**. *Overview of the Face Recognition Grand Challenge*. In: *IEEE computer society conference on Computer vision and pattern recognition*. Bd. 1. IEEE, 2005, S. 947–954.

[KFS08] **Kasinski, Andrzej, Florek, Andrzej und Schmidt, Adam**. *The PUT face database*. In: *Image Processing and Communications* 13.3-4 (2008), S. 59–64.

[Kea88] **Kearns, Michael**. *Thoughts on hypothesis boosting*. In: *Unpublished manuscript* 45 (1988), S. 105.

[KV94] **Kearns, Michael und Valiant, Leslie**. *Cryptographic limitations on learning Boolean formulae and finite automata*. In: *Journal of the ACM (JACM)* 41.1 (1994), S. 67–95.

[KRL90] **Keeler, J. D., Rumelhart, D. E. und Leow, W.-K.** *Integrated Segmentation and Recognition of Hand-Printed Numerals*. In: *Proc. Conf. Advances in neural information processing systems*. Morgan Kaufmann Publishers Inc., 1990.

[Kim+07] **Kim, Sanghoon, Chung, Sun-Tae, Jung, Souhwan, Oh, Dusik, Kim, Jaemin und Cho, Seongwon**. *Multi-Scale Gabor Feature Based Eye Localization*. In: *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 1.9 (2007), S. 2646–2650.

[KM96] **Kothari, Ravi und Mitchell, Jason L.** *Detection of Eye Locations in Unconstrained Visual Images*. In: *Proceedings on the International Conference on Image Processing*. Bd. 3. IEEE, 1996, S. 519–522.

[KHM08] **Kroon, B., Hanjalic, A. und Maas, S.** *Eye Localization for Face Matching: Is It Always Useful and Under What Conditions?* In: *Proc. 2008 Int. Conf. Content-based Image and Video Retrieval*. ACM, 2008, S. 379–388.

[Kro+09] **Kroon, B., Maas, S., Boughorbel, S. und Hanjalic, A.** *Eye Localization in Low and Standard Definition Content with Application to Face Matching*. In: *J. Computer Vision and Image Understanding* 113.8 (2009), S. 921–933.

[LSA00] **La Cascia, Marco, Sclaroff, Stan und Athitsos, Vassilis**. *Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.4 (2000), S. 322–336.

[Lai+14] **Lai, Chih-Chuan, Chen, Yu-Ting, Chen, Kuan-Wen, Chen, Shen-Chi, Shih, Sheng-Wen und Hung, Yi-Ping**. *Appearance-based gaze tracking with free head movement*. In: *22nd International Conference on Pattern Recognition (ICPR)*. IEEE, 2014, S. 1869–1873.

[Li+15] **Li, Haoxiang, Lin, Zhe, Shen, Xiaohui, Brandt, Jonathan und Hua, Gang**. *A convolutional neural network cascade for face detection*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, S. 5325–5334.

[LHR05] **Lichtenauer, J., Hendriks, E. und Reinders, M.** *Isophote Properties as Features for Object Detection*. In: *Proc. IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition*. Bd. 2. 2005, S. 649–654.

[LKS12] **Liebner, Martin, Klanner, Felix und Stiller, Christoph**. *Der Fahrer im Mittelpunkt — Eye-Tracking als Schlüssel zum mitdenkenden Fahrzeug?* In: *8. Workshop Fahrerassistenzsysteme (FAS2012)*. 2012, S. 87–96.

[LM02] **Lienhart, R. und Maydt, J.** *An extended set of Haar-like features for rapid object detection*. In: *Proceedings of the 2002 International Conference on Image Processing*. Bd. 1. 2002, S. 900–903.

[Lin94] **Lindeberg, Tony**. *Scale-space theory: a basic tool for analyzing structures at different scales*. In: *Journal of Applied Statistics* 21.1-2 (1994), 225–270.

[Lin98] **Lindeberg, Tony**. *Feature detection with automatic scale selection*. In: *International journal of computer vision* 30.2 (1998), S. 79–116.

[Log] **Logitech**. *Logitech Webcam c920*. "https://www.logitech.com/en-us/product/hd-pro-webcam-c920. Zuletzt zugegriffen am 30.10.17.

[Lu+15] **Lu, F., Sugano, Y., Okabe, T. und Sato, Y.** *Gaze Estimation From Eye Appearance, A Head Pose-Free Method via Eye Image Synthesis*. In: *IEEE Transactions on Image Processing* 24.11 (2015), S. 3680–3693.

[LK81] **Lucas, Bruce D. und Kanade, Takeo**. *An iterative image registration technique with an application to stereo vision*. In: *Proceedings of the DARPA image Understanding Workshop*. Bd. 81. 1981, S. 674–679.

[Man+15] **Mansouryar, Mohsen, Steil, Julian, Sugano, Yusuke und Bulling, Andreas**. *3D gaze estimation from 2D pupil positions on monocular head-mounted eye trackers*. In: *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research and Applications*. ACM, 2015, 197–200.

[MBM16] **Maralappanavar, Shweta, Behera, ReenaKumari und Mudenagudi, Uma**. *Driver's distraction detection based on gaze estimation*. In: *International Conference on Advances in Computing, Communications and Informatic*. IEEE, 2016, S. 2489–2494.

[Mar+14] **Markuš, Nenad, Frljak, Miroslav, Pandžić, Igor S, Ahlberg, Jörgen und Forchheimer, Robert**. *Eye pupil localization with an ensemble of randomized trees*. In: *Pattern recognition* 47.2 (2014), S. 578–587.

[Mar98] **Martinez, Aleix M.** *The AR face database*. In: *CVC technical report* (1998).

[MCP12] **Martinez, F., Carbone, A. und Pissaloux, E.** *Gaze Estimation Using Local Features and Non-linear Regression*. In: *19th IEEE Int. Conf. on Image Processing*. 2012, S. 1961–1964.

[MIB04] **Matthews, Iain, Ishikawa, Takahiro und Baker, Simon**. *The template update problem*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.6 (2004), S. 810–815.

[Mes+99] **Messer, Kieron, Matas, Jiri, Kittler, Josef, Luettin, Juergen und Maitre, Gilbert**. *XM2VTSDB: The extended M2VTS database*. In: *Second international conference on audio and video-based biometric person authentication*. Bd. 964. 1999, S. 965–966.

[met] **metavision.com**. *Meta 2*. "https://buy.metavision.com/products/meta2. Zuletzt zugegriffen am 26.10.17.

[MT09] **Murphy-Chutorian, Erik und Trivedi, Mohan M.** *Head pose estimation in computer vision: A survey*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.4 (2009), S. 607–626.

[MLS94] **Murray, Richard M., Li, Zexiang und Sastry, S. Shankara**. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[NL12] **Nanni, Loris und Lumini, Alessandra**. *Combining face and eye detectors in a high-performance face-detection system*. In: *IEEE MultiMedia* 4 (2012), S. 20–27.

[OPH96] **Ojala, Timo, Pietikäinen, Matti und Harwood, David**. *A comparative study of texture measures with classification based on featured distributions*. In: *Pattern recognition* 29.1 (1996), S. 51–59.

[OPM02] **Ojala, Timo, Pietikainen, Matti und Maenpaa, Topi**. *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*. In: *IEEE Transactions on pattern analysis and machine intelligence* 24.7 (2002), S. 971–987.

[OH08] **Ojansivu, Ville und Heikkilä, Janne**. *Blur insensitive texture classification using local phase quantization*. In: *International conference on image and signal processing*. Springer, 2008, S. 236–243.

[Ope] **OpenCV**. *Open Source Computer Vision Library*. http://sourceforge.net/projects/opencvlibrary/.

[Ore+97] **Oren, M., Papageorgiou, C., Sinha, P., Osuna, E. und Poggio, T.** *Pedestrian detection using wavelet templates*. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1997, S. 193–199.

[Oro+13] **Orozco, Javier, Rudovic, Ognjen, Gonzàlez, Jordi und Pantic, Maja**. *Hierarchical on-line appearance-based tracking for 3D head pose, eyebrows, lips, eyelids and irises*. In: *Image and Vision Computing* 31.4 (2013), 322–340.

[Pan+15] **Pang, Zhiyong, Wei, Chuansheng, Teng, Dongdong, Chen, Dihu und Tan, Hongzhou**. *Robust Eye Center Localization through Face Alignment and Invariant Isocentric Patterns*. In: *PloS one* 10.10 (2015), 1–19.

[Par+11] **Parris, Jonathan, Wilber, Michael, Heflin, Brian, Rara, Ham, El-Barkouky, Ahmed, Farag, Aly, Movellan, Javier, Castrilon-Santana, Modesto, Lorenzo-Navarro, Javier und Teli, Mohammad Nayeem**. *Face and eye detection on hard datasets*. In: *International Joint Conference on Biometrics*. IEEE, 2011, S. 1–10.

[PB07] **Phung, Son Lam und Bouzerdoum, Abdesselam**. *A new image feature for fast detection of people in images*. In: *International Journal of Information and Systems Sciences* 3.3 (2007), S. 383–391.

[Pla99] **Platt, J.** *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*. In: *Advances in Large Margin Classifiers* 10.3 (1999), S. 61–74.

[PA10] **Prasad, B. H. und Aravind, R.** *A robust head pose estimation system for uncalibrated monocular videos*. In: *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*. ACM, 2010, S. 162–169.

[Pra12] **Prasad, Dilip K**. *Survey of the problem of object detection in real images*. In: *International Journal of Image Processing* 6.6 (2012), S. 441–466.

[pup] **pupil-labs.com**. *Pupil-Labs Pupil Headsets*. "https://pupil-labs.com/pupil/. Zuletzt zugegriffen am 25.10.17.

[AF13] **Al-Rahayfeh, A. und Faezipour, M.** *Eye Tracking and Head Movement Detection: A State-of-Art Survey*. In: *IEEE Journal of Translational Engineering in Health and Medicine* 1 (2013).

[Ram06] **Ramanauskas, N**. *Calibration of video-oculographical eye-tracking system*. In: *Electronics and Electrical Engineering* 8.72 (2006), S. 65–68.

[Rea+11] **Reale, Michael J., Canavan, Shaun, Yin, Lijun, Hu, Kaoning und Hung, Terry**. *A Multi-Gesture Interaction System Using a 3-D Iris Disk Model for Gaze Estimation and an Active Appearance Model for 3-D Hand Pointing*. In: *IEEE Transactions on Multimedia* 13.3 (2011), S. 474–486.

[Ren+06] **Rentzeperis, Elias, Stergiou, Andreas, Pnevmatikakis, Aristodemos und Polymenakos, Lazaros**. *Impact of face registration errors on recognition*. In: *Artificial Intelligence Applications and Innovations*. Springer, 2006, S. 187–194.

[RG14] **Rusek, Krzysztof und Guzik, Piotr**. *Two-stage neural network regression of eye location in face images*. In: *Multimedia Tools and Applications* (2014), S. 1–14.

[Ryd87] **Rydfalk, Mikael**. *CANDIDE – a parameterized face*. 1987.

[SS98] **Schapire, Robert E und Singer, Yoram**. *Improved boosting algorithms using confidence-rated predictions*. In: *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, S. 80–91.

[Sch15a] **Schmidhuber, Jürgen**. *Deep learning in neural networks: An overview*. In: *Neural networks* 61 (2015), S. 85–117.

[SSS14] **Schneider, T., Schauerte, B. und Stiefelhagen, R.** *Manifold Alignment for Person Independent Appearance-based Gaze Estimation*. In: *Proc. 21st Int. Conf. Pattern Recognition*. IEEE, 2014.

[SS02] **Schölkopf, Bernhard und Smola, Alexander J.** *Learning with kernels: support vector machines, regularization, optimization, and beyond*. In: MIT press, 2002.

[Sha92] **Shashua, Amnon**. *Geometry and photometry in 3D visual recognition*. In: *Ph.D. thesis* (1992).

[SS01] **Shum, Heung-Yeung und Szeliski, Richard**. *Construction of panoramic image mosaics with global and local alignment*. In: *Panoramic vision*. Springer, 2001, S. 227–268.

[SS11] **Sigut, J. und Sidha, S. A.** *Iris Center Corneal Reflection Method for Gaze Tracking Using Visible Light*. In: *IEEE Transactions on Biomedical Engineering* 58.2 (2011), S. 411–419.

[SBB03] **Sim, Terence, Baker, Simon und Bsat, Maan**. *The CMU Pose, Illumination, and Expression Database*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.12 (2003), S. 1615–1618.

[Sim] **Simon, Jan**. *Windows API*. "http : / / de . mathworks . com / matlabcentral/fileexchange/31437-windowapi. Zuletzt zugegriffen am 30.10.17.

[SR01] **Sirohey, Saad A. und Rosenfeld, Azriel**. *Eye detection in a face image using linear and nonlinear filters*. In: *Pattern recognition* 34.7 (2001), S. 1367–1391.

[Son+13] **Song, Fengyi, Tan, Xiaoyang, Chen, Songcan und Zhou, Zhi-Hua**. *A Literature Survey on Robust and Efficient Eye Localization in Real-life Scenarios*. In: *Pattern Recognition* 46.12 (2013), S. 3157–3173.

[staa] **statista.com**. *Prognose zur Anzahl der Smartphone-Nutzer weltweit von 2012 bis 2020*. "https://de.statista.com/statistik/daten/studie/309656/umfrage/prognose-zur-anzahl-der-smartphone-nutzer-weltweit/.

[stab] **statista.com**. *Studie zur Anzahl der Smartphonenutzer in Deutschland*. "https://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenutzer-in-deutschland-seit-2010/. Zuletzt zugegriffen am 09.10.17.

[SMS13] **Sugano, Yusuke, Matsushita, Yasuyuki und Sato, Yoichi**. *Appearance-based gaze estimation using visual saliency*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.2 (2013), S. 329–341.

[SMS14] **Sugano, Yusuke, Matsushita, Yasuyuki und Sato, Yoichi**. *Learning-by-synthesis for appearance-based 3D gaze estimation*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, S. 1821–1828.

[SKK08] **Sung, Jaewon, Kanade, Takeo und Kim, Daijin**. *Pose robust face tracking by combining active appearance models and cylinder head models*. In: *International Journal of Computer Vision* 80.2 (2008), 260–274.

[STE13] **Szegedy, Christian, Toshev, Alexander und Erhan, Dumitru**. *Deep neural networks for object detection*. In: *Advances in Neural Information Processing Systems*. 2013, S. 2553–2561.

[TT10] **Tan, Xiaoyang und Triggs, Bill**. *Enhanced local texture feature sets for face recognition under difficult lighting conditions*. In: *IEEE transactions on image processing* 19.6 (2010), S. 1635–1650.

[TB11] **Timm, F. und Barth, E.** *Accurate Eye Center Localization by Means of Gradient*. In: *Proc. 6th Int. Conf. Computer Vision Theory and Applications*. Bd. 1. 2011, S. 125–130.

[toba] **tobii**. *Tobii Glasses*. "https://www.tobiipro.com/product-listing/tobii-pro-glasses-2/. Zuletzt zugegriffen am 26.10.17.

[tobb] **tobii**. *Tobii Spectrum*. "https://www.tobiipro.com/product-listing/tobii-pro-spectrum/. Zuletzt zugegriffen am 26.10.17.

[Tro04] **Tropp, Joel A.** *Greed is good: Algorithmic results for sparse approximation*. In: *IEEE Transactions on Information theory* 50.10 (2004), S. 2231–2242.

[TPC07] **Türkan, Mehmet, Pardas, Montse und Cetin, A. Enis**. *Human Eye Localization Using Edge Projections*. In: *Proceedings of the Second International Conference on Computer Vision Theory and Applications*. 2007, S. 410–415.

[Unia] **Universität des Saarlandes**. *Eye Tracking zur Untersuchung der Gefahrenwahrnehmung im Verkehr*. "http://www.uni-saarland.de/lehrstuhl/bruenken/forschung/aktuell/eye-tracking-zur-untersuchung-der-gefahrenwahrnehmung-im-verkehr.html. Zuletzt zugegriffen am 31.10.17.

[Unib] **University, Utrecht**. *Utrecht face database*. http://pics.psych.stir.ac.uk/2Dfacesets.htm.

[VG08] **Valenti, R. und Gevers, T.** *Accurate Eye Center Location and Tracking Using Isophote Curvature*. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 2008, S. 1–8.

[VG12] **Valenti, R. und Gevers, T.** *Accurate Eye Center Location through Invariant Isocentric Patterns*. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 34.9 (2012), S. 1785–1798.

[VSG12] **Valenti, R., Sebe, N. und Gevers, T.** *Combining Head Pose and Eye Location Information for Gaze Estimation*. In: *IEEE Trans. Image Process.* 21.2 (2012), S. 802–815.

[Val+09] **Valenti, R., Staiano, J., Sebe, N. und Gevers, T.** *Webcam-Based Visual Gaze Estimation*. In: *Proc. 15th Int. Conf. Image Analysis and Processing*. Springer, 2009, S. 662–671.

[Van+99] **Van Ginkel, M., Van de Weijer, J., Van Vliet, L.J. und Verbeek, P.W.** *Curvature estimation from orientation fields*. In: *Proc. Scandinavian Conf. Image Analysis*. Bd. 2. 1999, 545–552.

[Vap98] **Vapnik, Vladimir N.** *Statistical learning theory*. In: Bd. 1. 1998.

[VP14a] **Vater, S. und Puente León, F.** *Mehrdimensionale Merkmale zur Augendetektion*. In: *Forum Bildverarbeitung*. KIT Scientific Publishing, Karlsruhe, 2014, S. 119–128.

[Ver85] **Verbeek, P.W.** *A Class of Ssampling-error Free Measures in Oversampled Band-limited Images*. In: *Pattern Recognition Letters* 3.4 (1985), 287–292.

[VJ01] **Viola, P. und Jones, M.** *Rapid Object Detection using a Boosted Cascade of Simple Features*. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Bd. 1. 2001, S. 511–518.

[Viv] **Vive, HTC**. *HTC Vive*. "https://www.vive.com/ca/enterprise/. Zuletzt zugegriffen am 26.10.17.

[WYM07] **Wang, Jun, Yin, Lijun und Moore, Jason**. *Using Geometric Properties of Topographic Manifold to Detect and Track Eyes for Human-Computer Interaction*. In: *ACM Transactions on Multimedia Computing, Communications, and Applications* 3.4 (2007), Article No. 3.

[WJ16] **Wang, Kang und Ji, Qiang**. *Real time eye gaze tracking with Kinect*. In: *23rd International Conference on Pattern Recognition*. IEEE, 2016, 2752–2757.

[WJ07] **Wang, Peng und Ji, Qiang**. *Multi-view face and eye detection using discriminant features*. In: *Computer Vision and Image Understanding* 105.2 (2007), S. 99–111.

[Wan+05] **Wang, Peng, Green, Matthew B., Ji, Qiang und Wayman, James**. *Automatic Eye Detection and Its Validation*. In: *Proceedings of the IEEE Computer Society Conference onComputer Vision and Pattern Recognition-Workshops*. IEEE, 2005, S. 164–171.

[WHY09] **Wang, Xiaoyu, Han, Tony X. und Yan, Shuicheng**. *An HOG-LBP human detector with partial occlusion handling*. In: *12th International Conference on Computer Vision*, IEEE, 2009, S. 32–39.

[Wei+07] **Weidenbacher, U., Layher, G., Strauss, P.-M. und Neumann, H.** *A comprehensive head pose and gaze database*. In: *3rd IET International Conference on Intelligent Environments*. 2007, S. 455–458.

[WSA17] **Werner, Philipp, Saxen, Frerk und Al-Hamadi, Ayoub**. *Landmark based head pose estimation benchmark and method*. In: *IEEE International Conference on Image Processing*. 2017.

[WS08] **Wojek, Christian und Schiele, Bernt**. *A performance evaluation of single and multi-feature people detection*. In: *Pattern Recognition* (2008), 82–91.

[Woo+16] **Wood, Erroll, Baltrušaitis, Tadas, Morency, Louis-Philippe, Robinson, Peter und Bulling, Andreas**. *Learning an appearance-based gaze estimator from one million synthesised images*. In: *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research and Applications*. ACM, 2016, S. 131–138.

[wwwa] **www.digitalsignage.com**. "http://www.digitalsignage.com/. Zuletzt zugegriffen am 31.10.17.

[wwwb] **www.digitalsignagetoday.com**. "https://www.digitalsignagetoday.com/. Zuletzt zugegriffen am 31.10.17.

[Xia+03] **Xiao, Jing, Moriyama, Tsuyoshi, Kanade, Takeo und Cohn, Jeffrey F**. *Robust full-motion recovery of head by dynamic templates and re-registration techniques*. In: *International Journal of Imaging Systems and Technology* 13.1 (2003), S. 85–94.

[Xia+04] **Xiao, Jing, Baker, Simon, Matthews, Iain und Kanade, Takeo**. *Real-time combined 2D+3D active appearance models*. In: *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*. 2004, S. 535–542.

[Xu+12] **Xu, J., Wu, Q., Zhang, J. und Tang, Z.** *Fast and Accurate Human Detection Using a Cascade of Boosted MS-LBP Features*. In: *IEEE Signal Processing Letters* 19.10 (2012), S. 676–679.

[Yan+14] **Yang, Dingli, Bai, Qiuchan, Zhang, Yulin, Ji, Rendong und Zhao, Huanyu**. *Eye Location Based on Hough Transform and Direct Least Square Ellipse Fitting*. In: *Journal of Software* 9.2 (2014), S. 319–323.

[Yan+11] **Yang, Fei, Huang, Junzhou, Yang, Peng und Metaxas, Dimitris**. *Eye Localization through Multiscale Sparse Dictionaries*. In: *International Conference on Automatic Face and Gesture Recognition and Workshops*. IEEE, 2011, S. 514–518.

[YJS06] **Yilmaz, Alper, Javed, Omar und Shah, Mubarak**. *Object tracking: A survey*. In: *Acm computing surveys (CSUR)* 38.4 (2006), S. 13.

[ZMM10] **Zeng, C., Ma, H. und Ming, A.** *Fast human detection using MI-SVM and a cascade of HOG-LBP features*. In: *International Conference on Image Processing*. IEEE, 2010, S. 3845–3848.

[ZPV05] **Zhang, C., Platt, J. und Viola, P.** *Multiple Instance Boosting for Object Detection*. In: *Advances in neural information processing systems*. 2005, S. 1417–1424.

[ZYK13] **Zhang, Hua, Yang, Fan und Kong, Zhe**. *An Eye Location Algorithm Based on the Image Marking and Curve Blending*. In: *Journal of Computational Information Systems* 9.14 (2013), S. 5827–5835.

[Zha+07] **Zhang, Lun, Chu, Rufeng, Xiang, Shiming, Liao, Shengcai und Li, Stan Z.** *Face Detection Based on Multi-Block LBP Representation*. In: *Proceedings of the International Conference on Advances in Biometrics, Seoul, Korea.* Hrsg. von **Lee, Seong-Whan und Li, Stan Z.** Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, S. 11–18.

[ZG14] **Zhang, Tong und Gomes, Herman Martins**. *Technology survey on video face tracking*. In: *SPIE Electronic Imaging*. International Society for Optics und Photonics, 2014.

[Zha+15] **Zhang, Xucong, Sugano, Yusuke, Fritz, Mario und Bulling, Andreas**. *Appearance-based gaze estimation in the wild*. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, 4511–4520.

[Zhi+06] **Zhiheng, Niu, Shiguang, Shan, Shengye, Yan, Xilin, Chen und Wen, Gao**. *2D Cascaded AdaBoost for Eye Localization*. In: *18th International Conference on Pattern Recognition*. Bd. 2. IEEE, 2006, S. 1216–1219.

[Zhu+06] **Zhu, Qiang, Yeh, M-C, Cheng, Kwang-Ting und Avidan, Shai**. *Fast human detection using a cascade of histograms of oriented gradients*. In: *Computer Society Conference on Computer Vision and Pattern Recognition*. Bd. 2. IEEE, 2006, S. 1491–1498.

[ZR12] **Zhu, X. und Ramanan, D.** *Face detection, pose estimation, and landmark localization in the wild*. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2012, S. 2879–2886.

Eigene Veröffentlichungen

[VIP16] **Vater, Sebastian, Ivancevic, Ralph und Puente León, Fernando**. *Robustes Gesichtstracking durch Fusion von Active-Appearance-Modellen und präziser Irislokalisation*. In: *Forum Bildverarbeitung*. KIT Scientific Publishing, Karlsruhe, 2016, S. 257–268.

[VIP17] **Vater, Sebastian, Ivancevic, Ralph und Puente León, Fernando**. *Integration of precise iris localization into active appearance models for automatic initialization and robust deformable face tracking*. In: *IEEE International Conference on Image Processing*. 2017.

[VMP15] **Vater, Sebastian, Mann, Guillermo und Puente León, Fernando**. *A novel regularization method for optical flow based head pose estimation*. In: *Automated Visual Inspection and Machine Vision*. Hrsg. von **Puente León, Fernando und Beyerer, Jürgen**. Bd. Vol. 9530 of Proceedings of SPIE. Bellingham, 2015.

[VPP15] **Vater, Sebastian, Pallauf, Johannes und Puente León, Fernando**. *Referenzdatenbestimmung für die 3D-Kopfposenschätzung unter Verwendung eines Motion-Capture-Systems*. In: *XXIX. Messtechnisches Symposium*. De Gruyter Oldenbourg, Berlin, 2015, S. 115–122.

[VP14b] **Vater, Sebastian und Puente León, Fernando**. *Mehrdimensionale Merkmale zur Augendetektion*. In: *Forum Bildverarbeitung*. KIT Scientific Publishing, Karlsruhe, 2014, S. 119–128.

[VP16a] **Vater, Sebastian und Puente León, Fernando**. *Combining isophote and cascade classifier information for precise pupil localization*. In: *IEEE International Conference on Image Processing*. 2016, S. 589–593.

[VP16b] **Vater, Sebastian und Puente León, Fernando**. *Registrierung stabiler Merkmale zur Regularisierung des optischen Flusses bei der erscheinungsbasierten Schätzung der 3D-Kopfpose*. In: *Forum Bildverarbeitung*. KIT Scientific Publishing, Karlsruhe, 2016, S. 233–244.

[VP17] **Vater, Sebastian und Puente León, Fernando**. *Monokulare Kopfposenschätzung basierend auf dem optischen Fluss und der Verfolgung stabiler Merkmale*. In: *tm–Technisches Messen* (2017).

[Vat+16] **Vater, Sebastian, Pallauf, Johannes, Hoffmann, Marian, Stein, Thorsten und Puente León, Fernando**. *Erzeugung präziser Referenzdaten für die 3D-Kopfposenschätzung*. In: *tm–Technisches Messen* 83.9 (2016), S. 521–530.

Betreute studentische Arbeiten

[Dai16] **Daitsch, Alexander**. *Methoden zur Regularisierung des optischen Flusses bei der Kopfposenschätzung*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2016.

[Emm14] **Emmel, Fabian**. *Optimierung und Kalibrierung eines Modells zur Schätzung der Kopfpose*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2014.

[Fan15] **Fandos Duce, Inés**. *Application of Active Appearance Models for Gaze Vector Estimation*. Bachelor thesis. Karlsruher Institut für Technologie (KIT), 2015.

[Gri15] **Grimm, Daniel**. *Langzeit-3D-Kopfposentracking durch Verwenden eines dynamischen Objektmodells*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2015.

[Gru15] **Gruseck, Raphael**. *Robuste Irisdetektion mit Hilfe der Isophotendarstellung*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2015.

[Güh16] **Gühna, Sebastian**. *Skaleninvariante Irisdetektion in Echtzeit*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2016.

[Ham15] **Hampel, Lukas**. *Komplexe Modelle zur 3D-Kopfposenschätzung*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2015.

[Iva16] **Ivancevic, Ralph**. *Kopfposenschätzung mit dreidimensionalen Active Appearance Models*. Masterarbeit. Karlsruher Institut für Technologie (KIT), 2016.

[Man14] **Mann, Guillermo**. *Schätzung der 3D-Kopfposition aus 2D-Bilddaten*. Masterarbeit. Karlsruher Institut für Technologie (KIT), 2014.

[Sau13] **Saur, Gregor**. *Multifeature Kaskadenklassifikator zur Augendetektion*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2013.

[Sch15b] **Schnekenbühl, Marius**. *Nutzung von Texturinformationen in Kaskadenklassifikatoren zur Augendetektion*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2015.

[Sei15] **Seitz, Patrick**. *Erscheinungsbasierte Prädiktion der Kopfpose zur Regularisierung der Hessematrix*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2015.

[Sem14] **Semmler, Michaela**. *Bildinterpretation mit Active Shape und Active Appearance-Modellen*. Masterarbeit. Karlsruher Institut für Technologie (KIT), 2014.

[Sta16] **Stadler, Daniel**. *Regularisierung des optischen Flusses bei der Kopfposenschätzung*. Bachelorarbeit. Karlsruher Institut für Technologie (KIT), 2016.

[Tia12] **Tian, Wei**. *Echtzeit-Detektion und Tracking von Fußgängern mittels Haar-like Merkmalen in Videosequenzen*. Diplomarbeit. Karlsruher Institut für Technologie (KIT), 2012.

[Zhe14] **Zheng, Qiyuan**. *Blickrichtungserkennung mit Hilfe von IR-Licht*. Masterarbeit. Karlsruher Institut für Technologie (KIT), 2014.