Tracking von Menschen und menschlichen Zuständen

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Dipl.-Inform.

Patrick Dunau

aus Hermeskeil

Tag der mündlichen Prüfung:	03.12.2020
Erster Gutachter:	Prof. Dr.-Ing. habil. Jürgen Beyerer
Zweiter Gutachter:	Prof. Dr.-Ing. Marco Huber

Kurzfassung

Im Bereich der Kameraüberwachung von Menschen werden unterschiedliche Aspekte wichtig. Dazu gehört das Tracking von Menschen, wobei nicht nur die aktuelle Position wichtig ist. Das Trackingergebnis muss weiterverarbeitet werden, um Rückschlüsse auf den Zustand des Beobachteten zu ziehen, wie zum Beispiel die derzeitige Leistungsfähigkeit oder die Emotion. Zur Beurteilung der Leistungsfähigkeit von Probanden, muss ein Basiswert für diesen vorliegen. Für die Schätzung des emotionalen Zustands muss der Gesichtsausdruck beobachtbar sein.

Zur Auswertung von Bilddaten durch Menschen und Maschinen muss eine Registrierung der Bilddaten erfolgen. Am Beispiel von Beobachterversuchen zur Beurteilung von emissionshemmenden Materialien in Infrarotaufnahmen, wurde durch die direkte Projektion von GPS-Punkten in Bilddaten die Schätzung von Bild-zu-Bild Homographien verbessert. Das Tracking von Objekten im Video wurde zunächst am Beispiel eines Flugzeugs evaluiert. Feste Messstationen am Boden empfangen nicht informative Signale von einem Flugzeug. Der Sendezeitpunkt war unbekannt, nur der Empfangszeitpunkt lag vor. Durch die paarweise Subtraktion der Empfangszeiten ergeben sich sogenannte Time Differences of Arrival. Setzt man diese Zeitdifferenzen als Messungen in ein, um die aktuelle Position zu ermitteln, ergibt sich hierduch ein hyperbolischer Schnitt. Mit der direkten Verwendung der Empfangszeiten vereinfacht sich die Positionsbestimmung zu einem Kegelschnitt. In einem stochastischen Filter wurde der unbekannte Sendezeitpunkt simultan mit der Position geschätzt, wodurch eine robuste Ermittlung der Flugzeugposition erreicht werden konnte.

Für die Schätzung des emotionalen Zustands des Menschen muss das Trackingergebnis mehr enthalten als die Position. Am Beispiel des Auges wurde von

der Iris mit dem Tracking von ausgedehnten Objekten sowohl die Position, als auch Ausmaß und Form verfolgt. Hier wurde zunächst mit einem einfachen parametrischen Formmodell gearbeitet. Das Tracking ausgedehnter Objekte wurde im Anschluss für die Verfolgung von Gesichtern angewendet. Da hier kein einfaches parametrisches Formmodell verwendet werden konnte, wurde auf ein 68 Punkte umfassendes Landmarkenmodell zurückgegriffen. Um einem Auseinanderdriften der Landmarken entgegenzuwirken, wurde eine nichtlineare Nebenbedingung eingeführt. Mit dieser Nebenbedingung wird garantiert, dass die Landmarken die Form des Gesichts beibehalten. Dazu wird die Schätzung des Modells mit der vorher trainierten Hauptkomponentenanalyse transformiert und rücktransformiert, so dass Fehler durch eine Drift eliminiert werden. Hierdurch wird garantiert, dass das Gesicht in der erwarteten Form verbleibt und eine weitere Analyse des Gesichtsausdrucks vorgenommen werden kann.

Anschließend geht es um die Leistungsfähigkeit von Menschen. Zunächst wird darauf eingegangen, die Beobachtungsleistung zu evaluieren. In einer ersten Studie wird untersucht, ob ein Trainingseffekt für Beobachter nachgewiesen werden kann. In Videos einer simulierten Menschenmenge, in der sich Avatare mit und ohne Rucksack über einen Platz bewegen, mussten die Probanden die Avatare mit Rucksack finden und markieren. Mit Einzelbildern dieser Videos, in denen sich ausschließlich Avatare ohne und mit Rucksack befanden, wurden die Probanden trainiert. Durch Auswertung des Zustands vor dem Training und nach dem Training wurde versucht, einen Trainingseffekt nachzuweisen. Aufgrund einer sehr geringen Teilnehmerzahl konnte kein eindeutiger Trainingseffekt nachgewiesen werden. In einer zweiten Studie wurden die Probanden durch automatische Trackingsysteme unterstützt. Hier ging es darum, herauszufinden, ob ein solches System als unterstützend oder störend wahrgenommen wird. Unter der Ausführung einer Nebentätigkeit, in der ein zufällig erklingender akustischer Stimulus quittiert werden sollte, wurde die Arbeitslast der Probanden evaluiert. Bei einer moderaten Anzahl an Markierungen zeigt sich ein tendenziell positiver Effekt, der durch eine Erhöhung der Markierungsanzahl wieder aufgehoben wird.

Im Anschluss liegt der Fokus der Arbeit auf der Schätzung des emotionalen Zustands aus dem Gesichtsausdruck des beobachteten Menschen. Das Problem der Ermittlung des emotionalen Gesichtsausdrucks wurde bereits vielfach mit dem Einsatz von tiefen, neuronalen Netzen gelöst. Aus diesem Grund konzentriert sich diese Arbeit auf den Einsatz von analytischen Merkmalen. Mit einem neuartigen Modell, das auf dem 68 Punkte umfassenden Landmarkenmodell basiert, wird anhand von Winkel- und Größenmerkmalen ein Merkmalsvektor generiert. Die Winkelmerkmale enthalten zum Beispiel den Öffnungswinkel der Augenlider. Als Größenmerkmale werden die Achsenverhältnisse von Ellipsen verwendet, die anhand der Landmarken der Augen oder des Mundes geschätzt werden. Daraus ergibt sich ein 29 Einzelmerkmale beinhaltender Merkmalsvektor, der als *Angle-and-Size-Feature Set* (ASF) bezeichnet wird. In Experimenten ergaben sich vergleichbare Ergebnisse zu aktuellen tiefen, neuronalen Netzarchitekturen.

Abschließend befasst sich diese Arbeit mit der dynamischen Erweiterung der emotionalen Gesichtsausdruckschätzung. In einem neuartigen Ansatz wird zunächst mit einem Gaußprozess eine Abbildung des ASF-Merkmals in den Valenz-Erregungs-Raum definiert. Diese zweidimensionale Repräsentation des aktuellen emotionalen Zustands wird dann als Systemzustand für ein stochastisches Filter genutzt. Es wird eine Nebenbedingung definiert, die verhindert, dass der Systemzustand den Einheitskreis des Valenz-Erregungs-Raums verlässt. Dadurch wird eine unkontrollierte Drift des Zustands verhindert. Die dynamische Verfolgung des emotionalen Zustands konnte nicht mit dem Stand der Technik verglichen werden, da hier keine entsprechende Arbeit vorhanden war.

Acknowledgements

An dieser Stelle möchte ich allen beteiligten Personen meinen großen Dank aussprechen, die mich bei der Anfertigung meiner Doktorarbeit unterstützt haben.

Mein besonderer Dank gilt Prof. Dr.-Ing. Jürgen Beyerer und Prof. Dr.-Ing. Marco Huber für die hervorragende Betreuung bei der Durchführung der gesamten Arbeit.

Bei dieser Gelegenheit muss ich zudem Dr. Max Winkelmann, Dr. Alexander Schwarz, Dr. Karla Schiller und Ulrike und Norbert Ding meinen Dank äußern, die meine Arbeit durch ihre Unterstützung geprägt haben.

Malsch, Juni 2020

Patrick Dunau

Contents

Notation

Allgemeine Notation

Skalare	kursive lateinische und griechische kleingeschriebene Buchstaben	x, α
Mengen	Griechische großgeschriebene Buchstaben	Θ
Vektoren	unterstrichene lateinische, kleingeschriebene Buchstaben	$\underline{t}$
Matrizen	fette großgeschriebene lateinische Buchstaben	$\mathbf{R}$
Zufallsvariablen	kursive lateinische großgeschriebene Buchstaben	E
Mehrdimensionale Zufallsvariablen	fette kursive lateinische Buchstaben	$\boldsymbol{E}$

In mehrdimensionalen Mengen die im Zusammenhang zu Zeitserien stehen beschreibt der erste Index die Zeit.

Verteilungen

$\mathcal{N}$	Gauß'sche Normalverteilung

Zahlen und Indizes

k, t	Diskrete Zeitpunkte
i, j, ℓ, q	Indizes für Objekte, Messungen und Punkte
m, n	Anzahl Messungen, Anzahl getrackter Objekte

Geometrie (Koordinaten und Kamera Modellierung)

x, y, z	Weltkoordinaten
f	Brennweite
l, w, h	Länge, Breite, Höhe
r	Radius
α	Winkel
$\mathbf{p}$	Punkt im 2D und 3D Raum
$\mathbf{R}$	Rotationsmatrix
$\mathbf{t}$	Translationsvektor

Objektzustandsmodellierung und Wahrscheinlichkeiten

$\mathbf{A}$	Systemmatrix für das Kalman Filter
$\mathbf{H}$	Messmatrix für das Kalman Filter
$\mathbf{K}_k$	Kalman-Gain zum Zeitpunkt k
$\mathbf{C}_w$	Kovarianzmatrix Systemrauschen
$\mathbf{C}_v$	Kovarianzmatrix Messrauschen

1 Einleitung

Durch zunehmende Technisierung in verschiedenen Bereichen entstehen immer weitere Anwendungsmöglichkeiten zur Unterstützung der Sicherheit mit Kameras. Kameras werden zum Beispiel in Autos eingesetzt, um Müdigkeit beim Fahrer frühzeitig zu erkennen. Die kamerabasierte Überwachung öffentlicher Plätze zur Vermeidung von Straftaten oder Identifizierung einzelner Personen generiert eine riesige Datenmenge. Daher wird die Erforschung automatischer Methoden zur Auswertung der Datenfülle notwendig, um menschliche Beobachter gezielt zu unterstützen. Daraus leiten sich unterschiedliche Problemstellungen ab. Zum Einen müssen Beobachter für die Beobachtung geschult werden, um sie für die Erkennung von Menschen mit spezifischen Eigenschaften zu trainieren. Zum Anderen müssen Algorithmen entwickelt werden, die zur Unterstützung der menschbasierten Überwachung eingesetzt werden können. Zur Entwicklung von automatischen Algorithmen muss zunächst das automatische Überwachungsproblem formuliert werden. Das automatische System soll Menschen erfassen und tracken. Das Trackingergebnis muss weiter ausgewertet werden können, um zum Beispiel die aktuelle Situation zu beurteilen. Wird der Mensch als ausgedehntes Objekt getrackt, können nachträglich Merkmale extrahiert werden zur Erkennung der aktuellen Emotion des Menschen. Der emotionale Zustand kann auf zwei Arten bewertet werden: Zum Einen durch die isolierte Betrachtung zum jeweils aktuellen Zeitpunkt. Zum Anderen durch die Betrachtung des zeitlichen Ablaufs des emotionalen Zustands, durch Tracking. Von diesen Problemstellungen werden in dieser Arbeit folgende Teilaspekte betrachtet:

- Automatische Verfolgung von Menschen, wobei in dieser Arbeit das *Tracking* von ausgedehnten Objekten behandelt wird.

- Training von menschlichen Beobachtern, um die Aufgabe der Beobachtung effizient unter Verwendung von Verfahren zur automatischen Objektdetektion durchführen zu können.
- Automatische *Klassifikation* des emotionalen Gesichtsausdrucks.
- Zur Erfassung des emotionalen Gesichtsausdrucks wird in dieser Arbeit die Fragestellung behandelt, ob die *dynamische Entwicklung* des Gesichtsausdrucks verfolgt werden kann.

Zur Auswertung der entstehenden Bilddaten durch menschliche Beobachter ist eine spezifische Schulung wichtig. Hierzu werden verschiedene Ansätze betrachtet: Zunächst sollen Interaktionen von Menschen mit dem Video genau erfasst werden. Die Auswertung muss auf einem statistisch relevanten Fundament ausgeführt werden, um die Beobachtung korrekt beurteilen zu können. Unter Verwendung dieser Grundlagen wird in dieser Arbeit untersucht, ob ein Trainingseffekt für solche Beobachtungsversuche ermittelt werden kann. Zusätzlich wird der Einfluss von automatischen Zielerfassungssystemen auf die effektive Beobachtungsleistung von menschlichen Beobachtern untersucht. Der beschriebene Einfluss auf die Leistungsfähigkeit ist die Konzentration, die ein Mensch aufwenden muss, um einen effektiven Nutzen aus dem automatischen Zielerfassungssystems zu ziehen. Ein hohes Maß an Konzentration führt zu Ermüdung und einem Nachlassen der Leistungsfähigkeit.

Neben der Müdigkeit gibt es weitere Einflussfaktoren, die die Leistungsfähigkeit von Menschen beeinflussen. Ein wichtiger Aspekt ist die Emotion eines Menschen. Eine Emotion kann aus einem beobachteten Gesichtsausdruck erkannt werden. Das Gesicht muss über einen längeren Zeitraum beobachtet werden, um den Gesichtsausdruck analysieren zu können. Mit einem effizienten Trackingalgorithmus kann das Gesicht automatisch und dynamisch verfolgt werden. Trackingalgorithmen werden zur Verfolgung von Objekten eingesetzt, wie zum Beispiel von Flugzeugen.

Flugzeuge stellen ein klassisches Trackingobjekt dar. Ein Flugzeug führt koordinierte Bewegungen aus, die leicht modelliert werden können. Für ein

Flugzeug genügt die Betrachtung der aktuellen Position. Ein Gesicht hingegen kann sich auf verschiedene Arten bewegen. Es kann seine Position verändern und es kann sich durch Anspannung der mimischen Muskulatur verändern, zum Beispiel durch Veränderung des Gesichtsausdrucks. Um eine Änderung des Gesichts zu erfassen, reicht eine punktförmige Verfolgung nicht aus. Das Gesicht muss als ausgedehntes Objekt betrachtet werden. Um das Tracking ausgedehnter Objekte einzuführen, wird diese Form des Trackings am Beispiel der menschlichen Iris gezeigt. Dieser Ansatz wird im Anschluss auf Gesichter erweitert und mit Hilfe von Nebenbedingungen definiert. Aus der ausgedehnten Betrachtung des Gesichts können der Gesichtsausdruck und weitere Informationen analysiert werden. Hierdurch wird eine Grundlage für die dynamische Beobachtung von Gesichtern geschaffen. Das Tracking des Gesichts als ausgedehntes Objekt ist eine Grundlage für die statische und dynamische Betrachtung der aktuellen Emotion.

Ausgehend von der Beobachtung des ausgedehnten Objekts, dem Gesicht, werden weitergehende Problemstellungen betrachtet. Durch Gesichtsausdrücke werden Informationen transportiert, die Hinweise auf die aktuelle Gemütslage beinhalten. Es wird die Fragestellung bearbeitet, ob es Merkmale gibt, die eine Klassifikation des emotionalen Gesichtsausdrucks zulassen. Unter Merkmalen versteht man eindeutig beschreibende Eigenschaften von Teilen des Gesichts. In dieser Arbeit wird eine Menge von Merkmalen entwickelt, mit der sich der emotionale Gesichtsausdruck klassifizieren lässt.

Die Klassifikation basiert auf Erkennung statischer emotionaler Gesichtsausdrücke. Ein Gesichtsausdruck entwickelt sich dynamisch. Der neutrale Gesichtsausdruck kann als Ausgangspunkt definiert werden. Durch willkürliche Anspannung der mimischen Muskulatur wird ein Gesichtsausdruck eingenommen. Ausgehend von dieser Annahme wird die Fragestellung untersucht, ob die dynamische Entstehung dieses Gesichtsausdrucks unter Kombination von Trackingverfahren und des im vorherigen Schritt entwickelten Merkmalssatzes ausgewertet werden kann.

Die Arbeit ist wie folgt aufgebaut: Kapitel 2 geht auf den Stand der Technik ein. Zusätzlich werden in diesem Kapitel eigene Publikationen erläutert. Kapitel 3 stellt die in dieser Arbeit verwendeten Methoden vor und führt diese

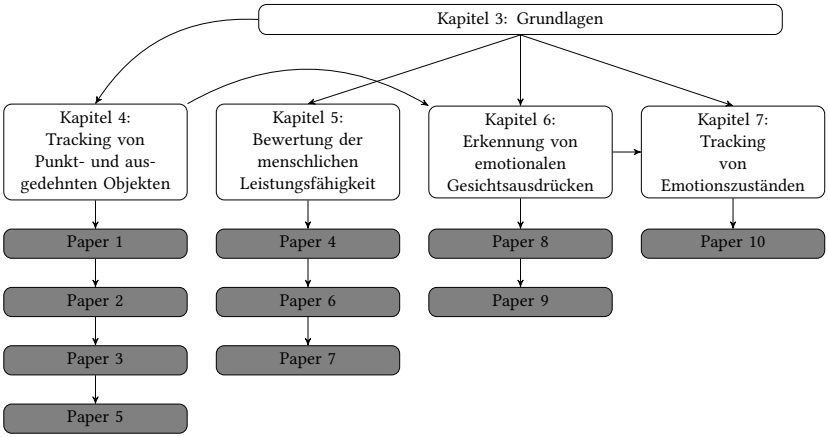


Figure 1.1: Relation der einzelnen Kapitel zueinander und Abhängigkeit der Kapitel von den Veröffentlichungen zu dieser Arbeit.

technisch ein. In Kapitel 4 wird das Tracking beschrieben mit dem Anwendungsfall des Gesichtslandmarkentrackings. Kapitel 5 befasst sich mit dem Training von menschlichen Beobachtern und der Bewertung von Beobachtungsleistung. Kapitel 6 behandelt die Klassifikation von Emotionen unter der Konstruktion neuer Merkmale für die Erkennung von emotionalen Zuständen. Abschließend behandelt Kapitel 7 die dynamische Schätzung und Verfolgung des emotionalen Zustands unter Verwendung der Merkmale zur Klassifikation von Emotionen aus dem vorangehenden Kapitel. Das letzte Kapitel schließt mit einer bewertenden und zusammenfassenden Diskussion der Arbeit ab. Abbildung 1.1 verdeutlicht den Aufbau der Arbeit.

2 Stand der Technik

2.1 Tracking von Menschen/Gesichtern

Diese Arbeit behandelt das Tracking von Menschen. Es werden emotionale und leistungsbezogene Zustände des Menschen betrachtet. Die Emotionen werden durch Beobachtung des Gesichts erkannt. Als Sensorik kommen dabei hauptsächlich Kameras im visuellen bis nahen Infrarotspektrum zu Einsatz. Es wird davon ausgegangen, dass sich das Gesicht zu jeder Zeit im Blickfeld der Kamera befindet. Das Tracking von Gesichtern kann in zwei Klassen unterteilt werden: Gesichtstracking basierend auf lokalen Bildmerkmalen und das Tracking basierend auf Formmodellen.

Bradski hat in [Bra98] bereits 1998 einen Gesichtstracker veröffentlicht, der lokale Bildmerkmale verwendet und ein Echtzeittracking von Gesichtern und beliebigen Objekten ermöglicht. Grundlage für die Verfolgung von Objekten sind approximierte Wahrscheinlichkeitsverteilungen basierend auf Farbhistogrammen. Bradski nutzt diese Farbhistogramme aufgrund der Invarianzeigenschaften gegenüber Translation und Rotation des Bildes. Bradski hat den Meanshift-Algorithmus erweitert, um ein verbessertes Tracking der Moden der Wahrscheinlichkeitsverteilungen zu implementieren. Bradski detektiert das Massezentrum der Verteilung und kann hierdurch neben der Position des Gesichts auch die Orientierung des Gesichts ermitteln.

Kim et al. [Kim08] und Kalal et al. [Kal10] verwenden Formmodelle zum Tracking des menschlichen Gesichts, was in die zweite Klasse von Gesichtstrackern gehört. Kim et al. verwenden ein adaptives Zielmodell für das Tracking

mit zusätzlichen visuellen Nebenbedingungen. Als grundlegendes Formmodell wird ein Active Appearance Model (AAM) eingesetzt. Die visuelle Nebenbedingung soll die Anpassung des Formmodells an Nichtziele verhindern. Die Nebenbedingung wird verwendet, um verschiedene Hypothesen zu verwerfen. Durch dieses Vorgehen konnten Kim et al. in [Kim08] einen stabilen Tracker präsentieren, der in der Lage ist, Gesichter in längeren, verrauschten Bildsequenzen zu tracken. Zur Einhaltung der Nebenbedingung verwendeten die Autoren eine Support Vektor Maschine (SVM). Kalal et al. verwendeten in [Kal10] einen Tracking Learning Detection (TLD) Ansatz. Kalal et al. konnten in diesem Framework einen beliebigen Detektor verwenden, um dynamisch auf neue Hypothesen reagieren zu können. Zusätzlich zum Detektor wurde ein Validierer verwendet. Die Autoren nutzten einen Random-Forest-Klassifikator für die Validierung von Hypothesen. Beide Ansätze nutzen Modelle zur Abbildung der Gesichter. Es wurden zusätzliche Stabilitätskriterien definiert, um das aktuell getrackte Objekt nicht zu verlieren und der Verwendung von Fehlmessungen vorzubeugen.

Die vorliegende Arbeit entwickelt einen Tracker, der auf Formmodellen basiert. Als Formmodell wird ein Landmarkenmodell verwendet, das durch einen generischen Detektor an ein Gesicht angepasst werden kann. Für diese Detektoren gibt es verschiedene Ansätze, die auf Active Appearence Modellen oder Constraint Local Methods basieren. Beide Modellansätze werden durch ein 68 Punkte umfassendes Landmarkenmodell repräsentiert. Die vorhandenen Verfahren nutzen kaskadierende Regression.

Ein Repräsentant für die merkmalsbasierten Ansätze ist der Algorithmus von King. In [Kin09] stellte King die Bibliothek Dlib vor. Darin ist ein Detektor enthalten, der Histogram of Oriented Gradients (HoG)-Deskriptoren verwendet. Der Algorithmus verwendet eine SVM, um die Kandidatenpunkte als dem Modell zugehörig oder nicht zugehörig zu klassifizieren. Durch dieses Vorgehen wird das Landmarkenmodell einem mittleren Modell angepasst. Das Vorgehen des Detektors basiert grundsätzlich auf dem in [Kaz14] von Kazemi et al. vorgestellten robusten, kaskadierenden Regressionsverfahren.

Qu hat in seiner Dissertation [Qu18] eine Lösung für die dreidimensionale Registrierung von Gesichtsmodellen präsentiert. Zur Erhöhung der Robustheit wurde die Gesichtssuperresolution eingesetzt, um niedrigaufgelöste Bilder nutzen zu können. In Qu et al. [Qu14] wurde gezeigt, dass dreidimensionale Gesichtsmodelle robust und schnell aus Videos extrahiert werden können. Für die Erkennung von Gesichtern wurde in Qu et al. [Qu15c] das Verfahren um die Extraktion von Texturen erweitert. In Qu et al. [Qu15b] wurde die zweidimensionale zur dreidimensionalen Verwendung von Gesichtsmodellen verglichen. Für die Anpassung von dreidimensionalen Gesichtsmodellen bilden Landmarken eine Basis. In Qu et al. [Qu15a] wurde ein leistungsfähiger, regressionsbasierter Algorithmus zur Anpassung von Landmarken an Gesichter in unterschiedlichen Posen vorgestellt. Dieser Algorithmus wird in dieser Arbeit als Grundlage für die Erkennung von emotionalen Gesichtsausdrücken verwendet. Im Vergleich zu Kazemi et al. [Kaz14] erreichen Qu et al. eine Verbesserung der Robustheit bei der Landmarkenanpassung. In Qu et al. [Qu15d] wurde die Robustheit der dreidimensionalen Gesichtsmodellschätzung durch Rotationsupdates gesteigert. Eine weitere Verbesserung der Texturnutzung wurde durch die Nutzung von Patches in Qu et al. [Qu17] erreicht.

2.2 Leistungstracking und -klassifikation

Neben der Erkennung und dem Tracking von Gesichtern können menschliche Zustände auch durch Leistung charakterisiert werden. Die Leistungsfähigkeit eines Menschen wird durch Müdigkeit beeinflusst. Somit ist die Müdigkeit ein starker Indikator für die Leistungsfähigkeit. Müdigkeit kann direkt und indirekt gemessen werden.

Für die direkte Erkennung von Müdigkeit werden visuelle Verfahren eingesetzt. Vural et al. präsentierten in [Vur07] einen Klassifikationsalgorithmus der auf Facial Action Unit (FAU)-Merkmalen basiert. Vural et al. verwendeten Methoden aus dem maschinellen Lernen, um ein Training mit Bildern aus verschiedenen Datenbanken zu ermöglichen. Neben der visuellen Betrachtung des Gesichts verwendeten Vural et al. sekundäre Sensorik wie Eye-Tracker

und Beschleunigungssensoren. Die Nutzung von multimodalen Eingangsgrößen wird für diese Arbeit ausgeschlossen. Für eine Messung der Müdigkeit bieten die Beobachtungen aus einem Eye-Tracker und von Beschleunigungssensoren wertvolle Informationen. Es ist denkbar, dass sich die Augenbewegungen unter zunehmender Müdigkeit signifikant verändern.

Ebenfalls auf FAU-Merkmalen basiert die Arbeit von Valstar und Pantik [Val12]. Valstar und Pantik verwendeten ein dynamisches Modell mit SVM und einem Hidden-Markov-Modell (HMM). Die Autoren nutzten die Kombination aus SVM und HMM zur Erkennung unterschiedlicher temporaler Phasen bei der Entstehung von Gesichtsausdrücken. Die Entstehung eines Gesichtsausdruck geschieht in einer hohen Geschwindigkeit, sodass eine Analyse der temporalen Phasen bei der Entstehung von Gesichtsausdrücken einen hohen Rechenaufwand benötigt. Dieser Aufwand wird in dieser Arbeit durch die Verwendung einer Support-Vektor-Maschine verringert.

Die Leistungsfähigkeit kann auch über die Beobachtung der Leistung bei dedizierten Aufgaben betrachtet werden. In dieser Arbeit wird die Leistungsfähigkeit von Menschen anhand der zu bewältigenden Aufgabe betrachtet. Hierzu wurden in der Literatur keine vergleichbaren Studien gefunden. Somit wird auf eigene Arbeiten in Abschnitt 2.5 verwiesen.

2.3 Emotionserkennung anhand von Einzelbilderkennung

Der emotionale Zustand eines Menschen zeigt sich auf unterschiedliche Arten. Ein emotionaler Gesichtsausdruck bietet eine visuelle Repräsentation für einen emotionalen Zustand [Ekm99]. Neben dem Gesichtsausdruck können multimodale Informationen verwendet werden, zum Beispiel audio-visuelle Reize wie die Körperhaltung und die Stimme von Beobachteten. In der vorliegenden Arbeit konzentriert sich die Emotionserkennung auf die Klassifikation von Gesichtsbildern. Die Klassifikation von Gesichtsbildern kann in merkmalsbasierte Verfahren und tiefe neuronale Netze unterteilt werden.

Die merkmalsbasierten Verfahren unterscheiden sich durch die verwendeten Merkmale. Huang verwendete in seiner Disseration [Hua14] zwei unterschiedliche Merkmale: Local Binary Patterns (LBP) und Local Quantised Patterns (LQP). Huang unterteilte dazu das Gesicht in Blöcke und bestimmte Histogramme basierend auf den Merkmalen als Deskriptor für das gesamte Gesicht. Die so entstehenden Merkmalsvektoren werden mit Klassifikationsverfahren verarbeitet. Die Extraktion der LBP und LQP ist sehr aufwändig und benötigt daher viel Zeit. Für eine Echtzeitverarbeitung müssen effizientere Verfahren gefunden werden.

LQP sind eine Verallgemeinerung von LBP und Local Ternary Patterns (LTP). LQP werden über großen Nachbarschaften und tiefer Quantisierung berechnet. Dazu wird domänenadaptive Vektorquantisierung eingesetzt. Hussain et al. benutzen in [Hus12] LQP, um Gesichter zu erkennen. Die resultierenden Merkmalsvektoren, die nach der Quantisierung entstehen, sind hochdimensional. Mittels Hauptkomponentenanalyse wurde die Dimensionalität des Merkmalsvektors reduziert. Dies geschah zusätzlich zur Reduktion redundanter Informationen im Merkmalsvektor, sowie zur Verhinderung von Overfitting. Die hohe Dimension des Merkmalsvektors setzt voraus, dass eine sehr große Anzahl an Eingabedaten vorliegt. Um diesem Umstand entgegenzuwirken, wurde eine Merkmalsreduktion durchgeführt. Diese beiden Punkte sprechen gegen eine direkte Verwendung dieses Ansatzes, da auch hier sehr hoher Aufwand betrieben werden muss.

Neuronales Netz (NN) bilden eine Alternative zu den merkmalsbasierten Ansätzen. In der aktuellen Literatur werden verschiedene Netzarchitekturen verwendet, um hoch genaue Klassifikatoren für emotionale Gesichtsausdrücke zu entwickeln. Liu et al. verwendeten in [Liu15] ein dreidimensionales Faltungsnetz (3D-CNN) mit deformierbaren Nebenbedingungen für Gesichtsteile. Hierzu importieren sie ein deformierbaren Action-Parts-Lerner in das 3D-CNN. Das 3D-CNN verfolgt zwei Ziele: Die Lokalisierung von Bereichen mit Gesichtsaktionen und das Lernen bereichsbasierter Repräsentationen. Das Netzwerk von Liu setzte sich aus sieben einzelnen Schichten zusammen: Einer Eingabeschicht, einer räumlich, temporalen Faltungsfilterschicht, einer weiteren Faltungsfilterschicht zur Erkennung von Einzelemotionen. Die weiteren

Schichten summieren die Ergebnisse und Prüfen die Einhaltung von Nebenbedingungen.

Mollahosseini et al. [Mol16] präsentierten ein dünn besetztes (engl. sparse) CNN zur Erkennung von emotionalen Gesichtsausdrücken. Die Verwendung eines sparse CNN soll die Komplexität und die Neigung zu Overfitting reduzieren. Die Autoren nutzten eine Inception-Schicht, um ein Netzwerk im Netzwerk zu realisieren. Hierdurch wurden lokale Merkmale verstärkt und die Pooling Leistung erhöht. Um vorab eine Registrierung der Bilddaten zu ermöglichen, verwenden Mollahosseini et al. ein Active Appearance Model mit Supervised Descent Methodik.

Lopes et al. entwickelten in [Lop17] ein CNN für die Verwendung von geringen Datenmengen für das Training. Sie verwendeten eine künstliche Vergrößerung der Datenbasis durch die Generierung von synthetischen Samples. Die Autoren beschrieben einen sehr aufwändigen Vorverarbeitungsprozess, der für alle präsentierten Faltungsnetze (CNNs) vorhanden ist. Die vorgestellten Architekturen unterscheiden sich geringfügig in der Zusammensetzung ihrer Schichten und liefern grundsätzlich sehr gute Ergebnisse bei der Verwendung der bekannten Gesichtsausdruckdatenbanken.

Den NN ist gemeinsam, dass hier ein hohes Maß an Vorverarbeitung durchgeführt werden muss. Diese Vorverarbeitung ist nötig, um den Fokus der Merkmalsextraktion der NN auf für die Aufgabenstellung relevante Bildausschnitte zu fokussieren. Wenn der Bildausschnitt zu viel Hintegrund beinhaltet kann es passieren, dass statt des gewünschten Gesichtsausdrucks, Gemeinsamkeiten im Hintergrund gesucht werden, die dann nicht zwingend mit den korrekten emotionalen Gesichtsausdrücken zusammenfallen. Der Aufwand ist nahezu genauso groß, wie bei einem Ansatz, der auf Merkmalen basiert. Daher wird in dieser Arbeit auf einen merkmalsbasierten Ansatz zurückgegriffen.

2.4 Tracking von Emotionen

Der Einzelbilderkennung steht das Tracking von Emotionen gegenüber, das Emotionen in einem dynamischen Kontext betrachtet. Im Verlauf einer Kommunikation oder während der Betrachtung eines Films kann sich die Emotion eines Menschen verändern. Durch das Tracking von Emotionen kann diese Veränderung beobachtet werden. Der Stand der Technik kann in verschiedene Strömungen unterteilt werden. Häufig werden Emotionen durch zweidimensionale Größen repräsentiert. Dies wird durch den Valenz-Erregungs-Raum realisiert. Das Tracking von Emotionen verwendet häufig multimodale Ansätze, die neben der visuellen Beobachtung des Gesichts auch Audio oder Haltungsinformationen verwenden.

Malandrakis et al. verwenden in [Mal11] ein HMM, um in Filmen dargestellte Emotionen basierend auf visuellen und Audio-Merkmalen zu tracken. Die Merkmale wurden von Malandrakis et al. durch Gaußmischdichtenmodell (GMM) repräsentiert und somit dem HMM übergeben. Die Grundwahrheit der verwendeten Experimente wurde anhand von Experten und Laien annotierten Filmsequenzen erstellt. Die in den Sequenzen wiedergegebenen Emotionen wurden von Schauspielern dargestellt. Es gibt keine natürlichen, spontanen Emotionen. Es kann keine Aussage darüber getroffen werden, wie das präsentierte Modell mit Sequenzen der gängigen Datenbanken funktioniert. Daher wird das Modell von Malandrakis et al. nicht zum Vergleich herangezogen.

Metallinou et al. verwenden in [Met11] Merkmale basierend auf der Beobachtung der Körpersprache und von Audio-Informationen. Für das Training wurde die CreativeIT-Datenbank mit Theaterszenen mit sehr ausgeprägten Bewegungen verwendet. Mit der annotierten Datenbank wurde ein Gaußmischdichten (GMM) Modell trainiert. Die Einzelschritt Klassifikation erfolgte unter Verwendung von Maximum Likelihood Schätzungen (MLE). Diese

Schätzungen wurden als Tracking-Ergebnis verwendet. Metallinou et al. nutzten eine validierte Datenbank von durch Schauspielern dargestellten Emotionen. Es wurde auch auf Audio-Informationen zurückgegriffen. Durch die Darstellung von Schauspielern könnte insbesondere die Audio-Information sehr stark überbetont sein.

2013 präsentierten Metallinou et al. in [Met13] einen erweiterten Ansatz für das Tracking von Emotionen basierend auf Körpersprache und Spracherkennung. Das verwendete Modell ist grundsätzlich das gleiche wie in [Met11]. Der Unterschied besteht in der Verwendung weiterer Sprachmerkmale und einer genaueren Annotation der Szenen. Dies konnte durch die Verwendung von Motion Capturing Daten erreicht werden. In [Met13] wurde der MLE Ansatz basierend auf GMM durch die Verwendung von einem Long short-term Memory (LSTM)-Regressionsansatz erweitert. Metallinou et al. konnten die Ergebnisse ihrer vorherigen Studie leicht verbessern. In dieser Studie wurde insbesondere die Annotation verbessert, da hier gänzlich auf Laien verzichtet wurde. Die Erweiterung um Motion Capturing Daten liefert genauere Messungen der Körpersprache. Außerdem wurde das Modell um ein LSTM erweitert, wodurch die Verbesserung der Ergebnisse leichter zu erklären sind.

Ein auf Gesichtsmerkmalen und physiologischen Reaktionen basierendes dynamisches Tracken von Emotionen ist in [Bai08] von Bailenson et al. veröffentlicht worden. Die Zielsetzung des Modells ist die Unterscheidung der Emotionen Freude und Traurigkeit. Die Autoren verwendeten multimodale Merkmale aus dem Gesicht und physiologische Messungen. Die Gesichtsmerkmale basieren auf der Extraktion von Punkten aus Gesichtsvideos. Die physiologischen Informationen werden durch Messungen der kardiovaskulären, somatischen Aktivitäten und der elektrischen Aktivität der Haut erfasst. Mit ihrem Modell erreichten Bailenson et al. eine gute Trennung der Emotionsklassen Freude und Traurigkeit. Das allgemeine Modell wurde zusätzlich mit personen- und geschlechtsspezifischen Modellen verglichen. Die Verwendung von Messungen des kardiovaskulären Systems liefert nur Informationen mit einer Latenz, da das kardiovaskuläre System verspätet auf emotionale Änderungen reagiert. Anders ist das insbesondere bei Angst und Stress, da der Körper hier schnell auf Flucht umgestellt wird. Die somatische Aktivität hängt

stark von der Situation ab, in der die Emotion beobachtet wird. Nicht in jeder Situation wird eine Emotion durch große somatische Aktivitäten begleitet. Ebenso verhält es sich mit der elektrischen Aktivität der Haut. Auch hier ist nicht mit sehr starken Ausprägungen zu rechnen. Die Emotionsklassen Traurigkeit und Freude besitzen sehr unterschiedliche Gesichtsausdrücke, woraus hier bereits durch die Betrachtung der Ausdrücke eine starke Trennung vorliegt.

In der Recherche haben sich beim Tracking von Emotionen sehr viele Ansätze finden lassen, die auf multimodale Informationen zurückgreifen. Hier werden häufig Kombinationen aus Sprache und Gesten verwendet. Auf Gesichtsmerkmalen basiertes Tracking ist in der Literatur noch unterrepräsentiert.

2.5 Eigene Veröffentlichungen

Im Verlauf der Promotion wurden insgesamt zehn Artikel veröffentlicht, wovon neun Artikel in Konferenzbänden erschienen sind und ein Artikel in einer Zeitschrift.

2.5.1 Efficient multilateration tracking with concurrent offset estimation using stochastic filtering techniques

Der Artikel [Dun10] wurde auf der International Conference on Information Fusion (FUSION 2010) in Edinburgh vorgestellt. In dem Artikel geht es um das Tracking von Flugzeugen basierend auf kooperativen Signalen. Die grundlegende Idee bezieht sich auf die Verwendung von ausreichend weit voneinander verteilten Bodenstationen, die kooperative Signale ohne Positionsinformation von Flugzeugen erhalten. Aus den ankommenden Signalen wurden

Zeitdifferenzen aus den Signalankunftszeiten ermittelt. Die Position des Flugzeuges kann unter Verwendung des hyperbolischen Schnitts bestimmt werden. Die Bestimmung des hyperbolischen Schnitts ist vergleichsweise komplex. In dem Artikel wird auf die Verwendung der Zeitdifferenzen verzichtet und auf die rohen Ankunftszeiten zurückgegriffen. Implizit ist in diesen Informationen ein Offset enthalten, basierend auf der Entfernung des Flugzeuges zu den spezifischen Basisstationen. Daraus ergibt sich die unbekannte Laufzeit des Signals. Um den unbekannten Offset zu bestimmen, erfolgt eine simultane Schätzung der Position und des Offsets. Damit wird ein Systemzustand erstellt, der sich aus der Objektposition, der Objektgeschwindigkeit und dem unbekannten Offset zusammensetzt.

Die Schätzung erfolgt unter Verwendung eines stochastischen Schätzers. Zur Formulierung des Schätzers wird das Messmodell unter Verwendung der entsprechenden Ankunftszeitformel mit einem additiven Rauschterm modelliert. Das Rauschen wird als gaußverteilt angenommen. Für das Systemmodell wird ein Interacting Multiple Model (IMM) unter Verwendung von Constant Position, Constant Velocity und Constant Acceleration Models eingesetzt. Je nach erwarteter Bewegung wird ein passendes Bewegungsmodell eingesetzt. Das Messmodell approximiert die Laufzeit des Signals und berechnet die Ankunftszeiten des Signals durch nichtlineare Gleichungen.

Aufgrund dieser Modellierung wird ein nichtlineares Filter eingesetzt, das sogenannte Gaußfilter. Das Gaußfilter verwendet, wie auch das Unscented Kalman Filter ein deterministisches Sampling auf Basis der ersten beiden Momente der A-Priori-Verteilung des Systemzustands. Da eine Gaußverteilung vollständig durch die ersten beiden Momente charakterisiert werden kann, gilt im Gaußfilter die Annahme, dass alle Zustände durch Gaußdichten repräsentiert werden. Die Schätzung der Position und des Offsets erfolgt simultan. Für die Experimente wurden Flugzeug und Sensoren simuliert. In drei unterschiedlichen Experimenten wurden zunächst die Rauschstärken variiert, die Anzahl der Sensoren und anschließend der Sensorausfall. Das vorgestellte Verfahren wurde mit Methoden aus dem Stand der Technik verglichen. Zusätzlich wurden das Gaußfilter und das Unscented Kalman Filter gegenübergestellt, wobei das Gaußfilter die besten Ergebnisse erzielte. In allen Fällen wurde der Stand

der Technik in Bezug auf die Leistung übertroffen. Die Ergebnisse des Artikels wurden im Rahmen einer Patentanmeldung verwendet.

Tracking ist ein elementarer Bestandteil für die dynamische Erfassung und Klassifizierung des aktuellen Emotionszustands eines Menschen. Tracking wird am Beispiel des Flugzeugs sehr häufig angewendet. Somit stellt hier das Tracking auf Basis nichtlinearer Messfunktionen eine Grundlage für das spätere Tracking von Emotionen dar.

2.5.2 Exploitation of GPS control points in low-contrast IR imagery for homography estimation

Der Artikel [Dun14] wurde beim Forum Bildverarbeitung 2014 in Regensburg vorgestellt. Das Papier beschreibt ein Verfahren zur Bestimmung von Bild-zu-Bild Transformationen, sogenannten Homographien. Im Artikel werden Bildsequenzen verwendet, die mit einer Infrarot-Kamera an Bord eines Helikopters aufgenommen wurden. Das Kamerasystem des Helikopters verfügt zusätzlich zu der Kamera noch über einen GPS Empfänger. Die Bilddaten werden aus erhöhter Position aufgenommen. Eine Grundvoraussetzung für die Berechnung von Homographien ist eine planare Szene. Aus dem Helikopter wurde eine Landschaft aufgenommen, in der nicht von einer planaren Szene ausgegangen werden kann. Aufgrund des Fehlens einer planaren Szene entsteht ein Modellierungsfehler. Wegen dieser Problematik funktioniert ein Bildregistrierungsverfahren, das auf Punktkorrespondenzen von Merkmalspunkten basiert, nicht immer optimal. Es kann zu Bewegungsfehlern kommen, die zu Drift führen. Um diese Probleme zu umgehen, wird in dem Artikel die Kameramatrix verwendet, um eine Abbildung zu generieren, die GPS Positionen in Bildpositionen projiziert.

Hierzu werden im Zielgebiet feste Positionen mit GPS Geräten vermessen. Die Punkte werden in jedes Bild der Bildsequenz abgebildet. Daraus ergeben sich nahezu perfekte Korrespondenzen aus denen sich die Homographien bestimmen lassen. In der Experimentsektion des Papiers wurde ein merkmalsbasierter Ansatz mit Korrespondenzanalyse mit dem GPS-Punkt basierten Ansatz

verglichen. Mit Hilfe des GPS-basierten Ansatzes konnte das Driftverhalten deutlich reduziert werden.

Die Vorverarbeitung von Videomaterial, das bei Beobachterversuchen eingesetzt wird, ist ein wichtiger Schritt. Über die verbesserte Berechnung der Bild-zu-Bild Transformationen, können Zielobjekte, die sich im Bild befinden, zu jeder Zeit korrekt zugeordnet werden. Dies erleichtert die Auswertung der von den Beobachtern gesetzten Markierungen. Dies ist eine Grundlage für die Auswertung der Leistung der Beobachter.

2.5.3 Homography estimation for low-contrast IR image sequences utilizing GPS control points

Artikel [Dun15a] erschien in einem Special Issue zum Forum Bildverarbeitung 2014 in der Zeitschrift Technisches Messen. Der Artikel beinhaltet eine Erweiterung von [Dun14]. Hier werden weitere Informationen neben der Kameramatrix und den GPS Punkten verwendet. Zusätzlich werden Informationen aus der inertialen Messeinheit (IMU), die in dem Helikopter verwendet wird, in das System eingearbeitet. Ferner setzt sich die Projektionsmatrix aus mehreren Rotationsmatrizen und der Kameramatrix zusammen.

Die erste Rotationsmatrix bezieht sich auf das Weltkoordinatensystem. Hierzu wurde ein Koordinatensystem aus der Geodäsie verwendet, das sogenannte WGS-84-Koordinatensystem. WGS ist ein Akronym und setzt sich zusammen aus den Worten World Geodetic System. Es handelt sich dabei um ein 1989 festgelegtes Standardsystem, das zum Beispiel in Navigationssystemen verwendet wird. Bei diesem Koordinatensystem handelt es sich um ein euklidisches Rechtssystem, in dem ein Referenzellipsoid, ein Geoid und die Lage der Fundamentalstation definiert sind. Der Referenzellipsoid ist näherungsweise an das Erdellipsoid angepasst. Die erste Rotationsmatrix wird dazu verwendet, um eine Transformation aus dem Kamerakoordinatensystem in das Weltkoordinatensystem ausführen zu können. Eine weitere Rotationsmatrix ergibt sich aus den Roll-, Pitch- und Yaw-Messungen der IMU, die im Kamerakoordinatensystem definiert sind. Außerdem wird eine Viewport-Transformation verwendet, in der die Kameraorientierung und IMU in die Achsen einsortiert sind.

Durch Hintereinanderschalten der Rotationen und der Kombination mit der Kameramatrix wird eine Transformation von GPS-Koordinaten in das Bildkoordinatensystem definiert.

Durch die verbesserte Formulierung und die Nutzung weiterer Informationen, die aus der IMU erhalten werden können, kann das GPS-basierte System mit dem Stand der Technik mithalten. Verglichen wurde mit Bildregistrierungsverfahren, die Scale-Invariant Feature Transform (SIFT) und Speeded Up Robust Features (SURF) Merkmale verwenden. Die Genauigkeit, die mit SIFT und SURF erreicht wurde, konnte nicht erreicht werden, aber im Mittel wurde eine starke Verbesserung im Vergleich zum vorherigen Artikel erreicht.

Mit diesem Artikel wurden die Ergebnisse des vorangehenden Artikels verbessert. Die Bild-zu-Bild Transformationen (Homographien) konnten erneut verbessert geschätzt werden. Die Auswertung der Beobachterversuche und die damit erbrachte Leistung der Beobachter kann verbessert ausgewertet werden. Der Fehler durch Ungenauigkeiten konnte signifikant reduziert werden.

2.5.4 Evaluation of statistical methods for the evaluation of observer trials for the assessment of the effectiveness of signature measures

Der Artikel [Dun15b] ist eine zusammenfassende Aufzählung und Bewertung von statistischen Methoden für die Auswertung von Beobachterversuchen. In diesem Artikel wurden Verfahren aufgezählt, mit denen sich interaktive Beobachterversuche evaluieren lassen. Die Problemformulierung führt zunächst das Problem eines interaktiven Beobachterversuchs ein. Es werden Annäherungsflüge auf Zielobjekte per Infrarotkamera aufgenommen. Im Bild gibt es zwei Zielobjekte, wovon eines in ungetarnter Form vorliegt und das andere mit einer zu bewertenden Tarnvorrichtung bedeckt ist. Ziel der Versuche war es herauszufinden, welches der beiden Objekte vorteilhafter versteckt war, d.h. welches der Objekte später im Anflug entdeckt würde.

Die Beobachter setzen mit der Computermaus Markierungen in Videos. Diese Markierungen können verschiedene Zustände annehmen. Diese Zustände repräsentieren die Erkennungsmodi Detektion, Klassifikation und Identifikation. In der Aufgabenbeschreibung stand bereits der Fahrzeugtyp fest. Aus dieser Interaktion ergeben sich mehrere Probleme. Zunächst besteht das Zuordnungsproblem, zu welchem Zielobjekt eine Markierung gehört. Aufgrund der Dynamik der Kameraführung liegen die Markierungen nicht immer genau auf den Zielobjekten. Ein weiteres Problem ist das erneute Anklicken einer Markierung, um den Zustand zu verändern. Hieraus ergibt sich ein weiteres Zuordnungsproblem. Anschließend müssen die Erkennungszeiten statistisch betrachtet werden und die Verteilungen der Markierungen korrekt behandelt werden, damit die korrekten Schlüsse für einen Abschlussbericht getroffen werden können.

Dieser Artikel liefert einen Überblick über statistische Methoden, die bei der Auswertung von Beobachterversuchen zum Einsatz kommen. Die Auswertung der Beobachterleistung muss auf validen, statistischen Verfahren basieren. Die beschriebenen Verfahren adressieren die Problemstellung bei der Zuordnung von Markierungen zu einem in den Bilddaten befindlichen Zielobjekt. Die Zuordnung von Kandidatenpunkten ist ein häufiges Problem und wird beim Tracking für ausgedehnte Objekte erneut aufgegriffen.

2.5.5 Iris Tracking using extended object tracking

Der Artikel [Dun16a] beschreibt einen Ansatz zum Verfolgen von größeren Objekten. Größere Objekte sind Objekte, die nicht durch einen Punkt repräsentiert werden, sondern durch eine geometrische Form beschrieben werden. In [Dun15a] wurde die Iris des menschlichen Auges als erweitertes Objekt betrachtet. Modelliert wurde die Form der Iris als ein Kreisobjekt. Die Parameter des Zustands sind auf den Mittelpunkt und den Radius des Kreises begrenzt. Die menschliche Iris wurde als zu trackendes Objekt ausgewählt, da hierdurch eine zuverlässige Verfolgung des Auges möglich ist. Hieraus können eine Vielzahl von Anwendungen abgeleitet werden, wie zum Beispiel das Eye Tracking, zur Feststellung der Blickrichtung.

Das Trackingproblem wird in dem Papier in vier Teile aufgeteilt: Zunächst wird die Augenregion detektiert, die Iris wird segmentiert, die Parameter werden geschätzt und die Iris wird getrackt. Die Detektion der Augenregion wird unter Verwendung des Kaskadenklassifikators von Viola und Jones [Vio04] aus dem Gesichtsrechteck extrahiert. Die Segmentierung der Iris erfolgt zunächst über Erstellung eines Histograms über den Grauwerten und mittels adaptiven Einstellens eines Grenzwertes. Hierdurch kann die Iris in ein Binärbild überführt werden. Per Kantendetektion und der Bilder der konvexen Hülle kann die Iris erfasst werden. Die Punkte, die aus der Kantendetektion resultieren, werden als verrauschte Messungen eines Kreises angenommen und per generativer Formfunktion mit dem zu trackenden Modell in Verbindung gebracht. Somit kann festgestellt werden, welche Winkel betrachtet werden müssen, um die reale Messung mit dem Model in Verbindung zu bringen. Das Modell wird mit Hilfe eines Unscented Kalman Filter (UKF) geschätzt und verfolgt.

In simulierten Experimenten erfolgt eine Überprüfung der Methodik. Es wird ein Augenbild per Random-Walk Modell positionell verändert. Der Tracking Algorithmus ist in der Lage das Auge verlässlich zu verfolgen. Prinzipiell bietet die Verfolgung der Iris eine robuste Methodik, das Auge zu beobachten und verschiedene Aufgaben zu bewältigen. Die Iris ist leichter zu segmentieren als die Pupille, da die Sklera zur Iris einen sehr starken Kontrast besitzt.

Trackingmethoden verwenden häufig punktförmige Zielobjekte. Das Ziel dieser Arbeit ist die Verfolgung des menschlichen Gesichts. In jedem Trackingschritt muss die Emotion aus dem Gesichtsausdruck abgeleitet werden können. Die reine Erfassung der Position ist nicht ausreichend. Es müssen weitere Informationen beim Tracking erfasst werden. Am Beispiel der Iris wird das Tracking von ausgedehnten Objekten vorgestellt. Die Iris wurde gewählt, weil die Form der Iris durch einen Kreis repräsentiert werden kann. Ein Gesicht ist nicht durch eine simple Form zu beschreiben.

2.5.6 Asynchronous threat awareness by observer trials using crowd simulation

Mit dem Artikel [Dun16b] wurde eine Studie dokumentiert, bei der es um die Feststellung des Einflusses eines Trainingseffekts auf Überwachungsaufgaben ging. Hierzu wurde in Kooperation mit den Schweizer Unternehmen Forventis und Armasuisse eine simulierte Überwachungssituation geschaffen. Die Schweizer Kollegen stellten für die Studie einen Crowdsimulation zur Verfügung, in der ein öffentlicher Platz dargestellt wurde. Auf diesem Platz liefen 100, 150 oder 200 Avatare durcheinander. Sie bewegten sich von einer Seite des Platzes zur anderen, nicht immer auf geraden Linien. Unter den Avataren gab es eine kleinere Anzahl an Avataren, die mit einem Rucksack ausgestattet waren. Es waren immer 10 Prozent der Avatare mit einem Rucksack ausgestattet. Die Probanden hatten die Aufgabe diese Avatare zu detektieren und zu markieren.

In der Studie wurde zunächst die Ist-Situation erfasst. Die Probanden haben mehrere Videos betrachtet und diese Aufgabe ausgeführt. Dann folgten drei Trainingsläufe, bei denen ausschließlich Einzelbilder betrachtet wurden, in denen sich sowohl Avatare mit Rucksack befinden konnten als auch ausschließlich Avatare ohne Rucksack. Die Reihenfolge der Trainingsbilder wurde zufällig verändert. In den Experimenten wurden das Alter und die Erfahrung der Probanden erfasst und die jeweilige Erkennungsleistung der Probanden. Es stellte sich heraus, dass es keine signifikante Verbesserung der Erkennungsleistung durch Training gab. Es gab auch keine signifikante Verschlechterung durch das Training.

Neben der Emotion eines Menschen ist die Leistungsfähigkeit eine weitere Beschreibungseigenschaft für einen Menschen. Mit der Leistungsfähigkeit lassen sich Informationen über Müdigkeit und Konzentration ableiten. In diesem Artikel wurde untersucht, ob die Leistungsfähigkeit der Probanden durch ein Training verbessert werden kann.

2.5.7 Dependency of human target detection performance on clutter and quality of supporting image analysis algorithms in a video surveillance task

Der Artikel [Hub17] greift die Ergebnisse der Studie aus [Dun16b] erneut auf. Das Ziel dieser Studie war die Untersuchung, ob ein automatischer Detektionsalgorithmus die Leistung der Probanden unterstützt, oder eine Verschlechterung der Leistung hervorruft. Es wurden fünfzehn Sequenzen verwendet und Detektionsrahmen manuell auf Avatare gesetzt.

In den Sequenzen waren 100, 150 oder 200 Avatare vorhanden. Die Markierungen wurden variiert zwischen 0, 5, 10, 20 und 40 markierten Subjekten. Die Anzahl der mit Rucksack ausgestatteten Avatare liegt bei 10 Prozent. Die Probanden mussten während der Versuche auf einen akustischen Stimulus reagieren. Die Reaktionszeit und Trefferquote wurden gemessen, um die Arbeitslast abzuschätzen.

Die Ergebnisse der Studie zeigten, dass die Detektionszeit mit der Anzahl der Avatare steigt. Ebenso sinkt die Detektionsrate mit der Anzahl der Avatare. Für die Workload Messung mit dem akustischen Stimulus wurde der gleiche Effekt für die Reaktionszeit festgestellt. Mit steigender Avataranzahl stieg auch die Reaktionszeit. Für die Trefferquote konnte kein Effekt festgestellt werden. Es wurde eine U-förmige Korrelation zwischen der Anzahl der Markierungen und der Reaktionszeit für den Workload-Task gefunden. Die Ergebnisse lassen Implikationen für das Design von Beobachtertrainings zu.

In diesem Artikel wurde ein zusätzlicher Task eingefügt, mit dem die aktuelle Belastung der Probanden gemessen werden konnte. Die aktuelle Belastung ist ein direkter Indikator für eine erhöhte Konzentration, die zu einer beschleunigten Ermüdung führen kann. Durch diese Betrachtung wurde eine weiterer Indikator festgestellt, mit dem sich der aktuelle Zustand des Menschen beschreiben lässt. Allerdings ist das mit zusätzlichem Aufwand verbunden und kann nicht durch reine Beobachtung des Gesichts festgestellt werden.

2.5.8 Reduced Feature Set for Emotion Recognition based on Angle and Size Information

Mit dem Artikel [Dun18a] wurde die Schätzung der menschlichen Emotion weiter optimiert. Das Gesicht wird erfasst und es wird ein 68 Punkte umfassendes Landmarkenmodell an das Gesicht angepasst. Dazu wird der Landmarkendetektor von Qu et al. [Qu15a] eingesetzt. Basierend auf diesen Landmarken werden Merkmale abgeleitet. Diese Merkmale werden aus primitiven Merkmalen zusammengesetzt: aus 26 Winkeln und 3 Größencharakterisierungen, woraus sich der Name Angle-and-Size Features (ASF) ableitet. Zur Bestimmung der Winkel werden Geraden aus Punktpaaren konstruiert; die Winkel entsprechen den Schnittwinkeln je zwei dieser Geraden. Die Größenmerkmale werden aus dem Verhältnis der Halbachsen von Ellipsen bestimmt. Die Ellipsen werden durch eine Least-Squares-Schätzung an die Randpunkte der Augen und des Mundes bestimmt.

Zur Verdeutlichung der Leistungsfähigkeit des Merkmalssatzes erfolgen in dem Artikel Experimente auf den zwei Datenbanken Cohn-Kanade+ und Oulu-Casia. Ermittelt wurden die Erkennungsraten, Präzision und F_1-Score, sowie die Genauigkeit für den Datensatz. Als Vergleichsmerkmalsatz wurde das Landmarkenmodell verwendet, das als Grundlage für den neuen Merkmalssatz dient. Zur Klassifikation für beide Modelle, wurde ein Multilayer Perzeptron Klassifikator eingesetzt. Es zeigte sich, dass die entwickelten Merkmale zu einer signifikant besseren Klassifikation führen, als ein Klassifikator, der die reinen Landmarken als Merkmale verwendet. Die generierten Merkmale lassen eine einfache Berechnung zu, wodurch eine starke Reduktion des Merkmalsraums stattfindet.

Die Merkmale aus diesem Artikel sind die Basis für die Auswertung der Emotion. Die vermutliche Emotion des Menschen wird aus dem Gesichtsausdruck abgeleitet. Die Merkmale werden aus dem 68 Punkte umfassenden Landmarkenmodell extrahiert. Es kommen nur einfache Primitive zum Einsatz, sodass eine schnelle Berechnung möglich wird.

2.5.9 Comparison of Angle and Size Features with Deep Learning for Emotion Recognition

Der Artikel [Dun18b] beschließt die statische Emotionserkennung mit einem Vergleich des Merkmalssatzes aus [Dun18a] mit Deep Learning Methoden. Zu diesem Zweck wurde das Deep Learning Modell durch ein VGG-16 Netzwerk vertreten. Um eine gute Vergleichsbasis zu erhalten, wurde das Netzwerk mit geringer Vorverarbeitung betrieben. Beide Methoden erhielten genau den gleichen Anteil an Vorverarbeitung der Bilddaten. Während das Deep Learning Modell direkt das Bild als Eingang bekam, wurde ausgehend von der Vorverarbeitung die Extraktion der Merkmale angestoßen.

Die Experimente wurden wiederum auf der Cohn-Kanade+ und der Oulu-Casia Datenbank durchgeführt. Für die Klassifikation wurde in diesem Artikel der XGBoost Klassifikator verwendet, einem Vertreter der Gradient Boosting Verfahren. Es wurden drei Experimente durchgeführt. Eines mit den Bildern der Datenbank Cohn-Kanade+, ein Zweites mit den Bildern der Datenbank Oulu-Casia und ein Drittes mit den Bildern beider Datenbanken. Beim Test mit den Bildern der Cohn-Kanade+ Datenbank wurden zwischen beiden Methoden nur geringe Unterschiede festgestellt, wobei der Merkmalssatz leicht bessere Ergebnisse lieferte. Bei der Oulu-Casia Datenbank erzielte der Merkmalssatz signifikant bessere Ergebnisse als das VGG-16 Modell. Dieser Umstand setzte sich im gemeinsamen Test fort. Im Vergleich der Genauigkeiten mit State of the Art Deep Learning Modellen liefert der generierte Merkmalssatz analoge Ergebnisse.

Deep Learning wird in vielen aktuellen Artikeln zur visuellen Emotionserkennung eingesetzt. Der eingebrachte Merkmalssatz basiert auf Feature Engineering und stellt die Merkmalerzeugung in den Vordergrund. Bei Deep Learning wird häufig der Eindruck erweckt, dass ein geringerer Aufwand im Vergleich zum Feature Engineering. Der Artikel zeigt, dass die meisten vorgestellten Deep Learning Artikel sehr viel Aufwand bei der Vorverarbeitung der Bilddaten aufwenden, um sehr gute Erkennungsraten zu erreichen. Ohne diesen erhöhten Aufwand bei der Vorverarbeitung kann die Leistung eines tiefen

neuronalen Netzes nur schwer mit der Leistung des vorgestellten Merkmalsatzes konkurrieren.

2.5.10 Gaussian Process based Dynamic Facial Emotion Recognition

Der Artikel [Dun19] verwendet die ASF-Merkmale als Messungen für das Tracking des Emotionszustands. Dieser wird durch einen zweidimensionalen Vektor repräsentiert. Der Vektor ist im Valenz-Erregungsraum (engl. Valence-Arousal-Space) definiert. Es wird ein Gaußprozess trainiert, um die ASF-Merkmale in den Valenz-Erregungsraum abzubilden. Die Messungen können direkt im Tracker verarbeitet werden. Der Trackingalgorithmus wird unter Verwendung eines Unscented-Kalman-Filters implementiert. Da keine direkte Annahme über die Systemdynamik für den Emotionszustand getroffen werden kann, wird hier ein Constant-Position-Model (CP-Modell) als Systemmodell verwendet. Die direkte Verwendung der Messungen mit diesem einfachen Systemmodell im Unscented-Kalman-Filter führt zu Drift des Systemzustands. Zur Verhinderung dieser Drift wird in diesem Paper eine Ungleichheitsnebenbedingung in das Unscented-Kalman-Filter integriert.

Die Funktionsweise des Trackers wird mittels Sequenzen aus den Bilddatenbanken Oulu-Casia und Cohn-Kanade+ getestet. Da zum Zeitpunkt der Veröffentlichung kein vergleichbares Verfahren existierte, konnte kein direkter Vergleich zum Stand der Technik durchgeführt werden. Dennoch wurden gründliche Experimente durchgeführt, die die Qualität des präsentierten Verfahrens verdeutlichen.

In diesem Artikel werden die Untersuchungen der vorangehenden Artikel zusammengeführt. Der Merkmalssatz wird mittels eines Gaußprozesses in eine Valenz-Erregungsraum Darstellung überführt und mit einem stochastischen Schätzer getrackt.

3 Grundlagen

Das Grundlagenkapitel führt Verfahren aus den Bereichen *Computer Vision*, *stochastische Zustandsschätzung*, *Klassifikation* und *Regression* ein. Die hier präsentierten Verfahren und Methoden werden in den folgenden Kapiteln vorausgesetzt. Aus dem Bereich *Computer Vision* wird ein Modell zur Repräsentation von Gesichtsausdrücken eingeführt und Algorithmen, mit denen ein solches Modell an ein Gesicht angepasst werden kann. Mehrere Kapitel dieser Arbeit handeln vom Tracking von punktförmigen oder ausgedehnten Objekten. Aus diesem Grund werden stochastische Schätzer und die zugehörigen Grundlagen aus dem Gebiet der *stochastischen Zustandsschätzung* vorgestellt. Weitere Kapitel behandeln die automatische Erkennung von emotionalen Gesichtsausdrücken anhand von frontalen Gesichtsabbildungen, daher werden verschiedene Methoden des maschinellen Lernens benötigt. Diese kommen aus den Bereichen *Klassifikation* und *Regression*.

Im ersten Abschnitt werden Methoden aus der maschinellen Bildverarbeitung eingeführt.

3.1 Computer Vision

Ein zentraler Bestandteil dieser Arbeit ist eine für Maschinen verwertbare Darstellung von Gesichtsausdrücken. Die Extraktion von strukturellen Informationen eines Gesichtsausdrucks aus Bildern ist für die Klassifikation von Emotionen wichtig. Zur Vereinfachung der Struktur eines Gesichtsausdrucks wird eine geometrische Beschreibung eingeführt. Grundlage dieser Beschreibung ist ein auf 68 Punkten basierendes Landmarkenmodell. In Abbildung 3.1

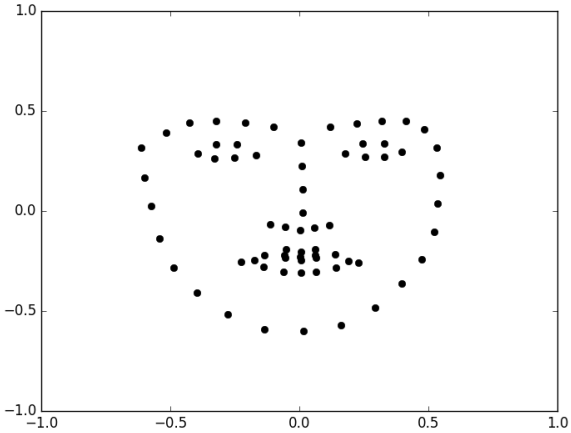


Abbildung 3.1: 68 Punkte umfassendes Landmarkenmodell zur geometrischen Repräsentation eines Gesichtsausdruck. Die Ausprägung des Gesichtsausdrucks resultiert als Mittelwert einer großen Menge von traurigen Gesichtsausdrücken.

ist eine Mittelung des Landmarkenmodells aus einer großen Menge trauriger Gesichtsausdrücke zu sehen.

Die Punkte des Landmarkenmodells sind nummeriert mit Indizes aus der Menge $\{0, \cdots, 67\}$. Nicht alle Punkte des Landmarkenmodells werden als Landmarken bezeichnet. Nur jene Punkte, die sich an gut differenzierbaren Stellen des Gesichts befinden, werden als echte Landmarken benannt. Beispiele für diese gut differenzierbaren Stellen sind Mund- und Augenwinkel. Zwischen den Landmarken werden Punkte äquidistant verteilt. Der generelle Aufbau des Landmarkenmodells wird in Cootes und Taylor [Coo00] beschrieben.

Für die Anpassung des Landmarkenmodells an beliebige Gesichter existieren in der Literatur verschiedene Ansätze. Zwei Beispiele sind hier die Verfahren von Qu et al. [Qu15a] sowie von Kazemi et al. [Kaz14]. Diese beiden Verfahren werden in den nächsten beiden Unterabschnitten eingeführt.

3.1.1 Landmarkendetektion von Qu et al.

Die Anpassung des Landmarkenmodells wird in Qu et al. [Qu15a] als Regressionsproblem

$$\sum_{i=1}^{N} \|\mathbf{r}(I_i, \mathbf{x}_i^{(0)}) - \mathbf{x}_i^*\|_2^2 \tag{3.1}$$

definiert, wobei die $\mathbf{r}(\cdot,\cdot)$ die Regressionsfunktion ist mit dem Bild I_i und der anzupassenden mittleren Form $\mathbf{x}_i^{(0)}$. Durch $\mathbf{x}_i^*$ wird die wahre Form für das Gesicht definiert. Nach Qu et al. ist eine direkte Lösung dieser Funktion mit einer einschrittigen Regression nur unzureichend möglich. Aus diesem Grund wählen Qu et al. in [Qu15a] ein kaskadiertes Regressionsverfahren, um der Komplexität der Formanpassung zu genügen. Als Regressionsmethode verwenden die Autoren das *Iteratively Reweighted Least-Squares* (IRLS) Verfahren. In jedem Einzelschritt wird ein gewichtetes Least-Squares Problem gelöst, wodurch eine robuste Regression erreicht wird [Gre84].

Die Landmarken werden durch Deskriptoren repräsentiert. Die Autoren setzen dazu Root-SIFT Merkmale ein. Diese basieren auf Histogrammen. SIFT steht für Scale-Invariant Feature Transform und ist ein Detektor und Merkmalsdeskriptor für Bilddaten. Der resultierende Merkmalsvektor des SIFT-Detektors besteht aus 128 Elementen, die weitere Informationen über den detektierten Pixel enthalten. Die Deskriptoren sind skalen- und rotationsinvariant und unempfindlich gegenüber Beleuchtungsänderungen. Das SIFT-Verfahren wurde von Lowe in [Low99] publiziert. Die Umwandlung des Merkmalsvektors in Root-SIFT Merkmale geschieht über die Transformation der Deskriptoren mit dem Hellinger-Kernel

$$H(x, y) = \sum \sqrt{x_i y_i} \tag{3.2}$$

in den Hellinger Raum. Hierdurch wird garantiert, dass die Distanzberechnung in der Regression anhand der Hellingermetrik geschieht. Nach Qu et al. [Qu15a] hat dies den Vorteil, dass schwach ausgeprägte Histogrammwerte verstärkt werden, die bei der Verwendung der euklidischen Distanz zu schwach repräsentiert werden. Die Autoren folgern, dass hierdurch eine höhere Präzision bei der Lokalisierung erreicht wird.

Ein weiterer Optimierungsschritt erfolgt durch eine verbesserte Anpassungsstrategie in den Einzelschritten. Die Autoren beschreiben, dass leichte Rotationen des Gesichts zu Schwierigkeiten bei der Anpassung führen können. Aus diesem Grund schlagen die Autoren vor, eine zweischrittige Anpassungsstrategie einzusetzen. Im ersten Schritt wird durch die trainierten Regressoren eine approximierende Form angepasst. Auf Basis dieser approximierenden Form wird die Ähnlichkeitstransformation zur mittleren Form bestimmt. Das Eingangsbild wird anhand dieser Transformation an die Form angepasst. Die weitere Anpassung erfolgt durch Anwendung der trainierten Regressoren. Im Anschluss wird die berechnete Transformation auf das Bild angewendet. Das Bild rotiert und skaliert, um das Gesicht in eine aufrechte Position zu bringen. Basierend auf den extrahierten Root-SIFT Merkmalen der mittleren Form wird diese erneut mit den trainierten Regressoren angepasst. Die Autoren können mit Experimenten die durch die Optimierung entstandene Robustheit belegen.

3.1.2 Landmarkendetektion von Kazemi et al.

Kazemi et al. in [Kaz14] wenden zur Lösung des Regressionsproblems (3.1), wie Qu et al., einen kaskadierten Regressionsalgorithmus an. Die Ansätze von Kazemi et al. und Qu et al. unterscheiden sich in zwei Punkten: Kazemi et al. verwenden Regressionsbäume im Gegensatz zum IRLS Verfahren und Pixelintensitäten.

Die Regressionsbäume für die Kaskanden werden mit dem Gradient-Boosting-Algorithmus von Friedman [Fri02] trainiert. Als Fehlerfunktion wird die Summe der quadratischen Distanzen verwendet. Das Ziel jeder Kaskade ist die

Verbesserung der Genauigkeit. Die Anzahl der Kaskaden wird durch die gewünschte Genauigkeit limitiert. Auf Basis von Pixelintensitäten werden in jedem Regressionsbaum relevante Merkmale ausgewählt. Hierzu werden lokale Pixeldifferenzen berechnet. Der Grenzwert für die Auswahl der Pixeldifferenzen wird adaptiv verändert, um lokale Beleuchtungsunterschiede auszugleichen. Zur Einschränkung des Suchraums für optimale Grenzen definieren die Autoren eine A-Priori-Exponentialverteilung. Zusätzlich erhalten die Autoren eine hohe Genauigkeit. Zur Verifizierung der Methode führen die Autoren in [Kaz14] Tests auf einer Vielzahl von Bilddatenbanken mit vielversprechenden Ergebnissen durch.

3.2 Stochastische Filterung

Die stochastische Filterung präsentiert eine Lösung für das Filterproblem aus der Theorie der stochastischen Prozesse. Das Filterproblem soll die *beste Schätzung des Zustands* für ein System auf der Basis von verrauschten Beobachtungen finden. Das von Rudolf Emil Kálmán in [Kál60] entwickelte Kalman Filter ist das beste lineare Filter für lineare Systeme. Für die Verwendung in nichtlinearen Systemen existieren nichtlineare Erweiterungen für das Kalman Filter. Ein Beispiel ist das Unscented Kalman Filter (UKF)von Julier und Uhlmann [Jul99].

Im Folgenden werden Verfahren zur stochastischen Filterung dargestellt. Einen guten Überblick über weitere stochastische Filterverfahren erhält man in Simon [Sim06]. Zunächst werden stochastische Prozesse und das stochastische Filterproblem erläutert. Abschließend werden konkrete Lösungen für das Filterproblem mit dem Kalman Filter und Unscented Kalman Filter vorgestellt.

3.2.1 Stochastische Prozesse

Ein stochastischer Prozess besteht aus einer Menge von Zufallsvariablen. Als Voraussetzung muss zunächst ein Wahrscheinlichkeitsraum (Ω, Σ, P) gegeben sein, wobei Ω die Grundgesamtheit ist, Σ eine σ-Algebra und P ein Wahrscheinlichkeitsmaß über Σ. Die Menge Σ wird auch als Ereignisraum bezeichnet, da hier alle möglichen Ereignisse aus der Grundgesamtheit enthalten sind. Für den Fall einer disrekten Ereignismenge Ω gilt die Potenzmenge $\Sigma = 2^{|\Omega|}$ als triviale σ-Algebra. Das Wahrscheinlichkeitsmaß P weist jedem Ereignis in Σ eine Wahrscheinlichkeit

$$P : \Sigma \mapsto [0,1] \tag{3.3}$$

zu. Für einen stochastischen Prozess wird zusätzlich zum Wahrscheinlichkeitsraum eine Indexmenge T benötigt. Die Indexmenge enthält die Beobachtungszeitpunkte eines stochastischen Prozesses. Übliche Mengen für T sind $\mathbb{N}_0$ und $\mathbb{R}_+$. Im Fall $T = \mathbb{N}_0$ spricht man von einem zeitdiskreten stochastischen Prozess. Wenn $T = \mathbb{R}_+$ gilt, dann ist der stochastische Prozess zeitkontinuierlich. Für einen stochastischen Prozess muss es für jedes $t \in T$ eine Zufallsvariable

$$X_t : (\Omega, \Sigma) \mapsto (\mathcal{X}_t, \mathcal{B}_t) \tag{3.4}$$

geben, wobei $(\mathcal{X}_t, \mathcal{B}_t)$ der Bildraum der Zufallsvariablen ist, mit Zustandsmenge $\mathcal{X}_t$ und Bildmenge $\mathcal{B}_t$. Ein allgemeiner stochastischer Prozess wird durch

$$X_T = \left(\Omega, \Sigma, P, (X_t)_{t \in T}\right) \tag{3.5}$$

definiert. Mit einem stochastischen Prozess lassen sich Modelle entwickeln, die mit Unsicherheiten behaftet sind. Beispielsweise kann bereits durch die

Modellierung eine Unsicherheit eingeführt werden. Ebenso Unsicherheiten, die durch die Umwelt induziert werden. Diese können ebenfalls als stochastische Prozesse definiert werden. Um das Filterproblem herzuleiten, wird die Zustandsraummodellierung für Prozesse eingeführt.

3.2.2 Zustandsraummodelle

Ein Zustandsraummodell kann über einen speziellen stochastischen Prozess motiviert werden. Ein solcher spezieller stochastischer Prozess ist durch den Markov-Prozess gegeben. Ein Markov-Prozess ist ein diskreter stochastischer Prozess mit einem endlichen oder abzählbar unendlichen Zustandsraum und der zusätzlichen Eigenschaft, dass der aktuelle Zustand von vorhergehenden Zuständen abhängt. In einer Markov-Kette können Übergangswahrscheinlichkeiten definiert werden. Die Wahrscheinlichkeit, dass der stochastische Prozess zum Zeitpunkt $(t+1)$ in Zustand (x_{t+1}) übergeht ist durch

$$P\left(X_{t+1} = x_{t+1} | X_t = x_t, \dots, X_0 = x_0\right) \tag{3.6}$$

definiert, wobei die Wahrscheinlichkeit von allen vorherigen Zuständen abhängt. Das heißt, in diesem Prozess muss die gesamte Historie des stochastischen Prozesses beachtet werden. Durch die Ordnung einer Markov-Kette kann dieser Umstand limitiert werden. Eine Markov-Kette erster Ordnung hängt nur vom vorhergehenden Zustand ab, wohingegen eine Markov-Kette n-ter Ordnung von den n-letzten Zuständen abhängt. In dieser Arbeit werden nur Markov-Prozesse erster Ordnung verwendet. Die Übergangswahrscheinlichkeit aus (3.6) vereinfacht sich zu

$$P\left(X_{t+1} = x_{t+1} | X_t = x_t, \dots, X_0 = x_0\right) = P\left(X_{t+1} = x_{t+1} | X_t = x_t\right) \ . \tag{3.7}$$

Betrachtet man den Fall, dass es zusätzlich einen Bildraum gibt, in dem ein stochastischer Prozess Beobachtungen emittiert, so kann die Wahrscheinlichkeit

einer Beobachtung $y_t \in \mathcal{B}_t$ in Abhängigkeit des aktuellen Zustands $x_t \in \mathcal{X}_t$ durch

$$P(y_t|x_t) \tag{3.8}$$

bestimmt werden. Um das Filterproblem zu definieren, wird das Hidden-Markov-Modell (HMM) eingeführt. Bei dem HMM geht man davon aus, dass der aktuelle Zustand unbekannt ist und nur die Wahrscheinlichkeiten der Zustandsübergänge bekannt sind. Ferner sind nur die Beobachtungen sichtbar. Im HMM heißen Beobachtungen Emissionen. Mit (3.8) und der Anwendung der Formel von Bayes kann das Filterproblem für das HMM durch

$$P(X_t = x_t|y_t) = \frac{P(y_t|X_t = x_t) \cdot P(X_t = x_t)}{\sum_{i=1}^{N} P(y_t|X_t = x_i) \cdot P(X_t = x_i)} \tag{3.9}$$

angegeben werden, wobei $P(y_t|X_t = x_t)$ als Likelihood bezeichnet wird, die Wahrscheinlichkeit $P(X_t = x_t)$ wird für die A-Priori-Wahrscheinlichkeit eingesetzt und der Nenner ist die Normierungskonstante $P(y_t) = \sum_{i=1}^{N} P(y_t|X_t = x_i) \cdot P(X_t = x_i)$. Die Zielgröße ist die auf die aktuelle Beobachtung bedingte A-Posteriori-Wahrscheinlichkeit $P(X_t = x_t|y_t)$ des Zustands. Hier sind alle Zufallsvariablen diskret.

Die Lösung des Filterproblems wird basierend auf kontinuierlichen Zufallsvektoren $\underline{x}_k$ und $\underline{y}_k$ diskutiert. Durch die Verwendung kontinuierlicher Größen verändert sich der Nenner in der Filtergleichung in (3.9) zu einem Integral. Dieses Integral ist nicht immer geschlossen lösbar. Aus diesem Grund werden für das Kalman Filter unterschiedliche Annahmen getroffen. Eine Annahme besteht darin, dass alle verwendeten Verteilungen Gaußverteilungen sind. Diese haben den besonderen Vorteil, dass Produkte und Faltungen zweier Gaußverteilungen wieder Gaußverteilungen sind. Zusätzlich wird davon ausgegangen, dass das zugrundeliegende System linear ist. Sind diese Voraussetzungen erfüllt, kann das Kalman Filter zur Lösung des Filterproblems

eingesetzt werden. Zusätzlich ist dann garantiert, dass das Filter eine optimale Lösung ist. Durch die Annahme der Gaußverteilung kann der Systemzustand durch einen Mittelwertvektor und die zugehörige Kovarianzmatrix repräsentiert werden. In einem zeitdiskreten System ergibt sich die Systemkovarianzmatrix zu

$$\mathbf{C}_k = \begin{bmatrix} \sigma_1^2 & \sigma_{1,2} & \cdots & \sigma_{1,n} \\ \sigma_{2,1} & \sigma_2^2 & \cdots & \sigma_{2,n} \\ \vdots & & \ddots & \vdots \\ \sigma_{n,1} & \cdots & \sigma_{n,n-1} & \sigma_n^2 \end{bmatrix} \tag{3.10}$$

mit Zeitindex k, wobei die σ_i^2 die Varianzen der Einzelkomponenten x_i des Systemzustands sind und es gilt $\sigma_{i,j} = \operatorname{cov}(x_i, x_j)$. Der zugehörige Mittelwertvektor wird auch Systemzustand genannt und ist durch $\underline{x}_k = [x_1, \cdots, x_n]^T$ gegeben.

Ein konkretes Beispiel für die Lösung des Filterproblems stellt das Verfolgen von bewegten Objekten in Videos dar. Ein verfolgtes Objekt wird in diesem Fall durch seine Position repräsentiert. Im Fall eines Videos genügt die Angabe von x_1,x_2-Koordinaten. Daher ist durch

$$\underline{x}_k = \begin{bmatrix} x_1^k \\ x_2^k \end{bmatrix} \tag{3.11}$$

ein Systemzustand definiert, wobei die Position in diesem Fall durch die Werte x_1^k und x_2^k repräsentiert wird. Die Systemkovarianzmatrix wird durch

$$\mathbf{C}_k = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} \\ \sigma_{x_1 x_2} & \sigma_{x_2}^2 \end{bmatrix} \tag{3.12}$$

beschrieben. Ist der Systemzustand unkorreliert, so gilt $\sigma_{x_1 x_2} = 0$. Ein mögliches Systemmodell ist das Constant Position Modell (CP). Das CP geht davon

aus, dass die Position des beobachteten Objekts, bis auf Rauschen, konstant ist. Ein Vorteil dieses Systemmodells stellt sich durch die Angabe einer linearen Systemgleichung dar. In diesem Fall kann die Systemgleichung durch eine Vektor-Matrix-Operation

$$\underline{x}_{k+1} = \mathbf{A} \cdot \underline{x}_k + \underline{w}_k \tag{3.13}$$

angegeben werden, wobei durch $\mathbf{A}$ die Systemmatrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.14}$$

gegeben ist und mit $\underline{w}_k \sim \mathcal{N}\left(\underline{0}, \mathbf{C}^w\right)$ ein normalverteiltes, mittelwertfreies Rauschen mit definierter Kovarianzmatrix $\mathbf{C}^w$ beschrieben wird. Die Kovarianzmatrix des Rauschterms beschreibt die Rauschanteile, die auf die x_1- und x_2-Koordinaten Einfluss nehmen. Bei der Modellierung wird meist von unkorreliertem Rauschen ausgegangen, woraus sich die Kovarianzmatrix

$$\mathbf{C}^w = \begin{bmatrix} \sigma_{x_1}^2 & 0 \\ 0 & \sigma_{x_2}^2 \end{bmatrix} \tag{3.15}$$

ableiten lässt. Das Systemmodell beschreibt die angenommene, eigenständige Bewegung des beobachteten Objekts. Das Kalman Filter verwendet Messungen aus der realen Welt, um eine verbesserte Schätzung des Systemzustands zu bestimmen. Hierzu wird ein Modell benötigt, mit dem der interne Zustand des Systems von außen beobachtet werden kann. Ein solches Modell wird als Abbildung aus dem Zustandsraum im den Beobachtungsraum definiert und

als Messmodell bezeichnet. Für das Kalman Filter wird ein lineares Messmodell

$$\underline{y}_k = \mathbf{H} \cdot \underline{x}_k + \underline{v}_k \tag{3.16}$$

benötigt, wobei die Matrix $\mathbf{H}$ eine lineare Abbildung des Systemzustands $\underline{x}_k$ ist und durch $\underline{v}_k$ wird ein additiver Rauschterm definiert. Es gilt $\underline{v}_k \sim \mathbb{N}\left(\underline{0}, C_k^v\right)$. Durch

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.17}$$

ist eine direkte Abbildung des gesamten Systemzustands definiert. Man spricht hier von einem direkt beobachtbaren Systemzustand, wohingegen ein indirekt beobachtbarer Zustand durch eine komplexere Abbildung beschrieben wird. Damit sind alle Komponenten eines linearen, dynamischen Systems eingeführt. Das Schätzproblem umfasst die bestmögliche Schätzung des *Zustands $\underline{x}_k$ unter Minimierung der Systemkovarianzmatrix $\mathbf{C}_k$. Bezüglich des quadratischen Fehlers garantiert das Kalman Filter die bestmögliche Schätzung des Zustands für lineare Systeme.

3.2.3 Kalman Filter

Das Kalman Filter ist ein lineares Schätzverfahren. Das Kalman Filter wird in zwei Schritte aufgeteilt: einen Prädiktionsschritt und einen Filterschritt, wobei der Filterschritt davon abhängt, ob eine wahre Beobachtung $\hat{\underline{y}}_k$ vorliegt. Im Prädiktionsschritt werden der Systemzustand und die Systemkovarianzmatrix für den aktuellen Zeitschritt prädiziert, unter Verwendung der Systemmatrix A und der Kovarianzatrix des Systemrauschens C_w. Die Prädiktionsgleichungen sind durch

$$\underline{x}_k^p = \mathbf{A} \cdot \underline{x}_{k-1}^e \tag{3.18}$$

$$\mathbf{C}_k^p = \mathbf{A} \cdot \mathbf{C}_{k-1}^e \cdot \mathbf{A}^T + \mathbf{C}_w \tag{3.19}$$

gegeben, wobei $\underline{x}_{k-1}^e$ und C_{k-1}^e der geschätzte Systemzustand und die geschätzte Systemkovarianzmatrix des vorherigen Zeitschrittes gegeben sind. Aus der Prädiktion der Systemkovarianzmatrix (3.19) wird die Unsicherheit erhöht. Im Fall des CP erhöht sich die Kovarianzmatrix um das Systemrauschen. Eine Verbesserung der Schätzung kann im Filterschritt gewonnen werden. Hierzu müssen mehrere Gleichungen gelöst werden. Zunächst muss die wahre Messung $\underline{\hat{y}}_k$ mit der prädizierten Messung aus (3.16) in Beziehung gebracht werden. Dies geschieht mit Hilfe des Innovationsvektors. Der Innovationsvektor ist durch

$$\underline{s}_k = \underline{\hat{y}}_k - \mathbf{H} \cdot \underline{x}_k^p \tag{3.20}$$

definiert und beschreibt die Abweichung der prädizierten Messung von der wahren Beobachtung. Ausgehend vom Messmodell kann die Residualkovarianzmatrix durch

$$\mathbf{S}_k = \mathbf{H} \cdot \mathbf{C}_k^p \cdot \mathbf{H}^T + \mathbf{C}_v \tag{3.21}$$

berechnet werden. Mithilfe der Residualkovarianzmatrix kann das Kalman-Gain

$$\mathbf{K}_k = \mathbf{C}_k^p \cdot \mathbf{H}^T \cdot (\mathbf{S}_k)^{-1} \tag{3.22}$$

berechnet werden. Das Kalman-Gain führt zu einer verbesserten Schätzung des Zustands im Filterschritt. Die Optimalität des Kalman-Gain kann bewiesen und zum Beispiel in [Kál60] nachgelesen werden. Das Kalman-Gain wird in den Filtergleichungen verwendet. Anhand der Filtergleichungen werden der Systemzustand und die Systemkovarianzmatrix geschätzt. Die Gleichungen sind durch

$$\underline{x}_k^e = \underline{x}_k^p + \mathbf{K}_k \cdot \underline{s}_k \tag{3.23}$$

$$\mathbf{C}_k^e = (\mathbf{I} - \mathbf{K}_k \cdot \mathbf{H}) \cdot \mathbf{C}_k^p \tag{3.24}$$

gegeben. Das Resultat der Filtergleichungen wird im Bayes'schen Sinn A-posteriori Schätzung genannt. Das Ergebnis der Prädiktionsgleichungen wird A-Priori Schätzung genannt. Im Filterschritt wird die aktuelle Information, die durch eine konkrete Beobachtung eingeführt wird, in den Zustand aufgenommen. Hierdurch kann eine Verbesserung der Schätzung erlangt werden. Für lineare Systeme mit normalverteiltem Rauschen ist das Kalman Filter optimal und liefert akkurate Schätzungen des Systemzustands. Ist das Rauschen nicht normalverteilt, so ist das Kalman Filter ein einfaches lineares Filter. Liegt jedoch ein nichtlineares System- oder Messmodell vor, so geht man von einem nichtlinearen System aus. In einem solchen Fall ist das Kalman Filter nicht optimal und es gibt für solche Systeme nichtlineare Erweiterungen des Kalman Filters. Eine dieser Erweiterungen ist das Unscented Kalman Filter.

3.2.4 Unscented Kalman Filter

Simon Julier und Jeffrey Uhlmann haben in [Jul99] eine nichtlineare Erweiterung des Kalman Filters publiziert. Durch diese Erweiterung konnte das Kalman Filter auf Systeme mit nichtlinearen System- und Messmodellen angewendet werden. Durch

$$\underline{x}_k = \underline{a}(\underline{x}_{k-1}) + \underline{w}_k \tag{3.25}$$

$$\underline{y}_k = \underline{h}(\underline{x}_k) + \underline{v}_k \tag{3.26}$$

sind vektorwertige, nichtlineare Funktionen definiert, wobei durch $\underline{w}_k$ ein additives, mittelwertfreies Gauß'sches Rauschen gegeben mit Kovarianzmatrix $\mathbf{C}_w$ und durch $\underline{v}_k$ ebenso ein additiver, mittelwertfreier Gauß'scher Rauschterm mit Kovarianzmatrix $\mathbf{C}_v$. Um den Systemzustand prädizieren und filtern zu können, schlagen Julier und Uhlmann vor, ein Sampling der Zustandsverteilung vorzunehmen. Dieses Sampling wird als Unscented Transformation bezeichnet, siehe [Jul02]. Für die Unscented Transformation wird von einer Normalverteilung des Zustands ausgegangen. Der Mittelwert der Normalverteilung ist die aktuelle Zustandsschätzung $\underline{x}_k^e$. Als Kovarianzmatrix wird die geschätzte Kovarianzmatrix des aktuellen Zustands $\mathbf{C}_k^e$ verwendet. Aus dieser Verteilung werden skalierte, symmetrische Sigmapunkte entnommen. Für einen n-dimensionalen Zustandsvektor ergeben sich $2 \cdot n + 1$ Sigmapunkte. Zu jedem Sigmapunkt gehört ein Gewicht. Die Sigmapunkte mit den zugehörigen Gewichten werden durch

$$\underline{\mathcal{X}}_0 = \underline{x}_k^e \tag{3.27}$$

$$\underline{\mathcal{X}}_i = \underline{x}_k^e + \left(\sqrt{(n+\kappa)\cdot \mathbf{C}_k^e}\right)_i \tag{3.28}$$

$$\underline{\mathcal{X}}_{i+n} = \underline{x}_k^e - \left(\sqrt{(n+\kappa)\cdot \mathbf{C}_k^e}\right)_i \tag{3.29}$$

$$\mathcal{W}_0 = \kappa/(n+\kappa) \tag{3.30}$$

$$\mathcal{W}_i = 1/2 \cdot (n+\kappa) \tag{3.31}$$

$$\mathcal{W}_{i+n} = 1/2 \cdot (n+\kappa) \tag{3.32}$$

bestimmt, wobei durch $\left(\sqrt{(n+\kappa)\cdot \mathbf{C}_k^e}\right)_i$ die i-te Zeile der resultierenden Kovarianzmatrix gemeint ist und es gilt $\kappa \in \mathbb{R}$. Die Unscented Transformation

sieht vor, die Sigmapunkte mit der nichtlinearen Systemfunktion zu transformieren, indem die Systemfunktion auf jeden einzelnen Sigmapunkt angewendet wird. Die resultierenden Sigmapunkte repräsentieren die Sigmapunkte der prädizierten Verteilung. Die Sigmapunkte können jetzt dazu verwendet werden, den prädizierten Systemzustand, sowie die prädizierte Kovarianzmatrix durch

$$\underline{x}_k^p = \sum_{i=0}^{2\cdot n} \mathcal{W}_i \cdot \underline{\mathcal{X}}_i^p \tag{3.33}$$

$$\mathbf{C}_k^p = \sum_{i=0}^{2\cdot n} \mathcal{W}_i \cdot \left(\underline{\mathcal{X}}_i^p - \underline{x}_k^p\right) \cdot \left(\underline{\mathcal{X}}_i^p - \underline{x}_k^p\right)^T + \mathbf{C}_w \tag{3.34}$$

zu berechnen, wobei die $\underline{\mathcal{X}}_i^p$ die durch das Systemmodell prädizierten Sigmapunkte sind. Für die Filtergleichung müssen die transformierten Sigmapunkte erneut durch die Messfunktion in Sigmapunkte der Messungen $\underline{\mathcal{Y}}_i^p = \underline{h}(\underline{\mathcal{X}}_i^p)$ transformiert werden. Die prädizierte Messung wird durch

$$\underline{y}_k^p = \sum_{i=0}^{2\cdot n} \mathcal{W}_i \cdot \underline{\mathcal{Y}}_i^p \tag{3.35}$$

bestimmt, wobei hier die Gewichte, sowie die transformierten Sigmapunkte Verwendung finden. Unter Verwendung der prädizierten Messung und der transformierten Sigmapunkte wird die Messkovarianzmatrix

$$\mathbf{C}_{yy}^p = \sum_{i=0}^{2\cdot n} \mathcal{W}_i \cdot \left(\underline{\mathcal{Y}}_i^p - \underline{y}_k^p\right) \cdot \left(\underline{\mathcal{Y}}_i^p - \underline{y}_k^p\right)^T + \mathbf{C}_v \tag{3.36}$$

berechnet. Um den Zusammenhang zwischen dem Systemzustand und der Messung zu beschreiben, wird die Kreuzkovarianzmatrix erzeugt. Hierzu werden die prädizierten Sigmapunkte $\underline{\mathcal{X}}_i$, sowie der prädizierte Systemzustand

$\underline{x}_k^p$, als auch die transformierten Sigmapunkte $\underline{\mathcal{Y}}_i$ und die zugehörige prädizierte Messung $\underline{y}_k^p$ benötigt. Die Kreuzkovarianzmatrix ergibt sich dann durch

$$\mathbf{C}_{xy}^p = \sum_{i=0}^{2\cdot n} \mathcal{W}_i \cdot \left(\underline{\mathcal{X}}_i^p - \underline{x}_k^p\right) \cdot \left(\underline{\mathcal{Y}}_i^p - \underline{y}_k^p\right)^T . \tag{3.37}$$

Das Kalman-Gain kann jetzt durch Matrixmultiplikation der Kreuzkovarianzmatrix mit der Inversen Messkovarianzmatrix berechnet werden. Durch

$$\mathbf{K}_k = \mathbf{C}_{xy}^p \cdot \left(\mathbf{C}_{yy}^p\right)^{-1} \tag{3.38}$$

leitet sich das Kalman-Gain her. Der Filterschritt ergibt sich äquivalent zum Filterschritt des linearen Kalman Filters. Somit wird die a-posteriori Schätzung durch die Gleichungen

$$\underline{x}_k^e = \underline{x}_k^p + \mathbf{K}_k \cdot \left(\underline{\hat{y}}_k - \underline{y}_k^p\right) \tag{3.39}$$

$$\mathbf{C}_k^e = \mathbf{C}_k^p - \mathbf{K}_k \cdot \mathbf{C}_{yy}^p \cdot \mathbf{K}_k^T \tag{3.40}$$

angegeben. Die Approximation des Systemzustands durch die Sigmapunkte weicht die Normalverteilungsannahme des Standard Kalman Filters auf. Hierdurch kann eine beliebige Transformation auf den Systemzustand modelliert werden und trotzdem kann der Systemzustand durch die ersten beiden zentralen Momente repräsentiert werden. Das Unscented Kalman Filter präsentiert eine elegante Lösung für das nichtlineare Filterproblem.

Das UKF gehört zur Klasse der Lineare Regressions-Kalman-Filter (LRKF). Ein weiteres Beispiel für ein nichtlineares, stochastisches Filter ist das Partikelfilter. Das Partikelfilter unterscheidet sich vom Unscented Kalman Filter durch die Verwendung von zufälligen Samples der zugrundeliegenden Verteilung. Darüber hinaus benötigt das Partikelfilter eine weitaus größere Menge an Samplepunkten, um eine ausreichend gute Repräsentation des Systems zu

erhalten. Aufgrund seiner Speichereffizienz wird für nichtlineare Filterprobleme in dieser Arbeit auf das Unscented Kalman Filter zurückgegriffen.

3.3 Maschinelles Lernen

Zu den Problemstellungen des maschinellen Lernens gehören Klassifikations-, Regressions- und Clusteringprobleme. Die für die Problemstellungen verwendeten Lernverfahren können in zwei Klassen unterteilt werden: überwachte und unüberwachte Lernverfahren. Zu den überwachten Lernverfahren gehören Klassifikations- und Regressionsverfahren. Clusteringverfahren zählen zu den unüberwachten Lernverfahren. Überwachte Lernverfahren werden eingesetzt, wenn fest definierte Label für die Trainingsdaten vorliegen. Wenn keine Label vorliegen können unüberwachte Lernverfahren eingesetzt werden, um die Trainingsdaten auf Muster zu untersuchen. In der Emotionserkennung können den Emotionen feste Labels zugeordnet werden. Daher werden in dieser Arbeit ausschließlich überwachte Lernverfahren angewandt. In diesem Abschnitt werden zwei Klassifikationsverfahren eingeführt, das *eXtreme Gradient Boost*-Verfahren und *künstliche neuronale Netze*. Im darauffolgenden Abschnitt werden Gaußprozesse als Vertreter der Regressionsverfahren definiert.

3.3.1 Klassifikationsprobleme

Das Klassifikationsproblem setzt voraus, dass ein Lernverfahren eine unbekannte Beobachtung der korrekten Kategorie zuordnen kann. Eine Kategorie kann als Klasse bezeichnet werden. Das Verfahren, das für die Zuordnung einer unbekannten Beobachtung zur korrekten Klasse trainiert wird, heißt Klassifikator. Für das Training eines Klassifikators wird eine Trainingsmenge benötigt, die Beobachtungen beinhaltet, deren Klassenzuordnung bekannt ist. Aus dem Grund, dass die Klassenzuordnungen bekannt sind, spricht man vom überwachten Lernen. Beim Training wird die sogenannte Diskriminierungsfunktion erlernt, sodass spätere Beobachtungen der korrekten Klasse zugeordnet werden können.

Die Beobachtungen enthalten erklärende Variablen, die eine Zuordnung zur korrekten Klasse begünstigen. Daher wird im maschinellen Lernen vor der Aufstellung eines Klassifikationsproblems eine Vorverarbeitung der Beobachtungen durchgeführt. Im Schritt dieser Vorverarbeitung wird die Beobachtung auf möglichst unabhängige erklärende Variablen reduziert, um somit einen Beobachtungsvektor oder Merkmalsvektor zu erzeugen. Die erklärenden Variablen oder Merkmale können nominal, ordinal oder numerisch sein. Unter nominalen Merkmalen versteht man kategorische Variablen die keiner Ordnung unterliegen, wie zum Beispiel Emotionen (Wut, Furcht, Freude, etc.). Ordinale Merkmale sind kategorische Merkmale, die eine Ordnung induzieren, wie zum Beispiel: lang, mittellang und kurz. Numerische Merkmale hingegen können ganzzahlige oder reelle Werte sein, d.h. zum Beispiel $x \in \mathbb{N}$ oder $x \in \mathbb{R}$.

Klassifikationsverfahren sind mathematische Verfahren, die durch Algorithmen implementiert werden können, die Merkmalsvektoren auf Klassen abbilden. Ein klassisches statistisches Klassifikationsverfahren ist die logistische Regression, die unabhängige erklärende Variablen oder Regressoren auf Klassen abbildet. Für eine ausführliche Einführung in die logistische Regression wird auf Cramer [Cra02] verwiesen. Im maschinellen Lernen gibt es eine Vielzahl von Methoden zur Klassifikation angefangen mit Support-Vektor-Maschine (SVM) Kivinen et al. [Kiv04], Decision Trees Sonquist et al. [A S64], Random Forests Breiman [Bre01], Gradient Boosting Machines von Friedman [Fri02] oder künstliche neuronale Netze in Haykin [Hay98].

Im Folgenden wird das eXtreme Gradient Boosting (XGBoost) Verfahren als Erweiterung der Gradient Boosting Machine von Friedman eingeführt und künstliches neuronales Netz (KNN) werden als aktuelle Vertreter für leistungsfähige Klassifikationsverfahren vorgestellt.

3.3.2 eXtreme Gradient Boosting - XGBoost

In 2016 veröffentlichten Chen und Guestrin mit [Che16] das eXtreme Gradient Boosting Verfahren (XGBoost). Durch XGBoost wurde die Gradient Boosting Machine von Friedman [Fri02] mit einem Random Forest von Breiman [Bre01]

verbunden. Grundlegend für XGBoost ist die Kombination einzelner Gradient Boosted Decision Trees, mit dem Ziel effizient und schnell zu sein.

Das XGBoost-Verfahren nutzt eine regularisierte Trainingszielfunktion

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \tag{3.41}$$

wobei $l(\hat{y}_i, y_i)$ eine konvexe Verlustfunktion ist, die den Unterschied zwischen der Prädiktion $\hat{y}_i$ und dem Ziel y_i misst und $\Omega(f_k)$ ist ein Strafterm, der die Komplexität des Baumes f_k bestraft. Der Strafterm wird nach Chen et al. [Che16] dazu verwendet, um die final gelernten Gewichte zu glätten und somit Overfitting zu vermeiden. Der Regularisierungsterm ist durch

$$\Omega(f_k) = \gamma \cdot T + \frac{1}{2}\dot{\lambda} \cdot \|\omega\|^2 \tag{3.42}$$

definiert, wobei γ und λ Regularisierungsparameter sind, T ein diskreter Zeitschritt und durch $\dot{\lambda}$ wird die erste Ableitung nach der Zeit des Regularisierungsparameters λ bezeichnet. Da die regularisierte Zielfunktion (3.41) Funktionen enthält, wenden Chen et al. eine Approximation an. Sie nutzen einen additiven Ansatz, um den aktuellen Baum hinzufügen zu können. Dieser Baum wird durch eine Taylorapproximation zweiter Ordnung linearisiert und somit in einen dynamischen und konstanten Teil separiert, wodurch sich die regularisierte Zielfunktion zu

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} \left[g_i \cdot f_t(\underline{x}_i) + \frac{1}{2} \cdot h_i \cdot f_t^2(\underline{x}_i) \right] + \Omega(f_t) \tag{3.43}$$

reduziert [Che16]. Mit t wird der aktuelle Baum f_t indiziert und g_i ist die erste partielle Ableitung der Verlustfunktion nach der Prädiktion des Baumes $(t-1)$. Durch h_i wird die zweite partielle Ableitung nach der Prädiktion des

vorherigen Baumes $(t-1)$ bezeichnet. Mit $\underline{x}_i$ wird die Beobachtung zur Zielvariablen y_i bezeichnet.

Chen und Guestrin verwenden (3.43), um einen Score für die aktuelle Baumstruktur zu entwickeln. Somit haben sie eine Optimierungsgröße geschaffen, die die aktuelle Baumstruktur bewertet. Es entsteht die Möglichkeit, optimale Splits für Blätter abzuleiten und somit eine bessere Baumstruktur zu generieren. Außerdem verwenden Chen und Guestrin weitere Techniken, um das Overfitting weitergehend zu verhindern. Dazu gehört Shrinkage, das auf Friedman [Fri02] zurückgeht. Durch Shrinkage wird ein Parameter η eingeführt, der die Wichtigkeit der Baumstruktur nach jedem Gradientenschritt verringert. Somit entsteht die Möglichkeit, dass weitere Bäume den Klassifikator verbessern können. Chen und Guestrin verwenden ein Subsampling von Merkmalen (Spalten). Diese Technik wird auch in Random Forests [Bre01] zur Vermeidung von Overfitting eingesetzt. Beim Finden des optimalen Splits wird im XGBoost Verfahren davon ausgegangen, dass nicht immer alle Merkmale vorhanden sind. Es wird mit dünnbesetzten Merkmalsvektoren gerechnet, dieses Vorgehen ermöglicht eine parallele Verarbeitung des Trainingsaufwands.

Der XGBoost Klassifikator vereint viele gute Eigenschaften. Durch eine geschickte Definition der Optimierungsfunktion mit einem Strafterm für komplexe Baumstrukturen, wird Overfitting effektiv verhindert. Außerdem werden gleichzeitig Shrinkage und das Subsampling eingesetzt, um eine weitere Maßnahme gegen Overfitting einzusetzen. Durch diese effektive Vermeidung von Overfitting kann das Verfahren auch für Datensätze eingesetzt werden, die nur über wenige Vertreter aller Klassen verfügen. Diese Eigenschaft ermöglicht eine grundlegende Generalisierung für allgemeine Probleme. Zusätzlich gehört das XGBoost-Verfahren zu den effizientesten Verfahren. Das Training der Bäume kann durch Parallelisierungen erheblich beschleunigt werden. Der Aufwand für einzelne Prädiktionen ist gering. Das Verfahren hat seine Effizienz und Güte in vielen Kaggle-Wettbewerben[1] unter Beweis

[1] https://www.kaggle.com/dansbecker/xgboost

gestellt. Es konnten häufig vergleichbare und bessere Ergebnisse als bei der Verwendung von künstlichen neuronalen Netzen (KNN) erreicht werden.

3.3.3 Künstliche neuronale Netzte (KNN)

Unter KNN versteht man Netzwerke aus künstlichen Neuronen. Mit neuronalen Netzen wird versucht, das menschliche Gehirn zu modellieren. So sollen Denk- und Lernprozesse im menschlichen Gehirn besser verstanden werden. Es ist bis heute nicht gelungen, ein neuronales Netz zu erschaffen dessen Neuronenanzahl dem menschlichen Gehirn entspricht.

Auf Rosenblatt geht die Entwicklung des Perzeptrons [Ros58] zurück. Das Perzeptron stellt eine Verknüpfung von parallelen Eingabeneuronen mit Ausgabeneuronen dar. Frank Rosenblatt zeigte, dass er durch den Einsatz von zwei Eingabeneuronen und einem Ausgabeneuron die Funktionalität von logischen Operatoren wie dem Und sowie dem Oder nachbilden konnte. Weiter entwickelte Rosenblatt erste Lernprozeduren für neuronale Netze. Auf das Beispiel des Perzeptrons gehen die Architekturen von aktuellen neuronalen Netzen zurück.

Im Allgemeinen ist ein KNN aus vernetzten Neuronen konstruiert. In Abbildung 3.2 ist ein einfaches Multilayer Perceptron dargestellt.

Die Neuronen eines neuronalen Netzes sind miteinander verknüpft und gewichtet. Durch eine Lernprozedur werden diese Verknüpfungsgewichte an das zu lösende Problem angepasst. Damit ein Neuron entscheiden kann, ob es *feuert*, wird für jedes Neuron eine Aktivierungsfunktion definiert. Mit dieser Aktivierungsfunktion wird gesteuert, ab wann das Neuron eine Ausgabe liefert und wie diese Ausgabe aussieht. Eine typische Aktivierungsfunktion ist die Schwellwertfunktion, die nur dann eine Ausgabe abgibt, wenn der Eingabewert einen Schwellwert übertrifft. Durch

$$a(t) = \begin{cases} 1 & \text{wenn } t > s \\ 0 & \text{sonst} \end{cases} \tag{3.44}$$

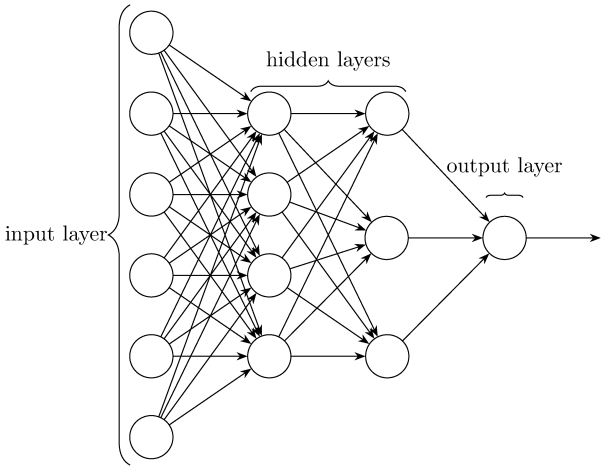


Abbildung 3.2: Exemplarischer Aufbau eines Multilayer Perzeptrons mit einer Eingabeschicht, zwei versteckten (hidden) Schichten und einer Ausgabeschicht. (Bild freundlicherweise zur Verfügung gestellt durch https://github.com/ledell/sldm4-h2o/blob/master/sldm4-deeplearning-h2o.Rmd)

wird ein Beispiel für eine Schwellwertaktivierungsfunktion angegeben, wobei s der Schwellwert ist. Durch

$$a(t) = c \cdot t \tag{3.45}$$

wird die lineare Aktivierungsfunktion definiert, die proportional zum Eingang t der Aktivierungsfunktion ist. Mit

$$a(t) = \frac{1}{1 + e^{-t}} \tag{3.46}$$

wird die Sigmoid-Aktivierungsfunktion eingeführt. Die Sigmoid-Funktion ist eine glatte, nichtlineare Schrittfunktion. Sie wird in der Logistischen Regression als Klassifikationsfunktion verwendet. Um den Koordinatenursprung weist die Sigmoid-Funktion eine starke Steilheit auf, sodass in diesem Bereich des

Definitionsbereichs bereits kleine Änderungen zu signifikanten Änderungen in der Ausgabe der Aktivierungsfunktion führen. Für $t \to -\infty$ konvergiert die Sigmoid-Funktion sehr schnell gegen 0 und für $t \to \infty$ konvergiert die Funktion sehr schnell gegen 1. Somit reagiert die Aktivierungsfunktion weiter entfernt vom Mittelpunkt nur geringfügig bis gar nicht in der Ausgabe. Durch diese Eigenschaft ist die Sigmoid Funktion hervorragend für die Klassifikation geeignet. Die Ausgabe ist zwischen 0 und 1 begrenzt und diese Funktion ist gut für die Kombination mehrerer Schichten von Neuronen geeignet. Die Konvergenzeigenschaften an den Rändern des Definitionsbereichs führen dazu, dass Gradienten verschwinden können. Die Eigenschaft der verschwindenden Gradienten ist unerwünscht im Bereich der künstlichen neuronalen Netze. Daher wurden weitere Aktivierungsfunktionen entwickelt, um diese Eigenschaft zu umgehen. Durch

$$a(t) = \tanh(t) = \frac{2}{1 + e^{-2t}} \tag{3.47}$$

wird die $\tanh(t)$ Aktivierungsfunktion definiert. Sie ist der Sigmoid-Funktion ähnlich, da $\tanh(x) = 2 \cdot \text{sigmoid}(2x) - 1$ gilt. Im Gegensatz zur Sigmoid-Funktion liefert die Tanh-Funktion Werte im Intervall $[-1,1]$. Die Funktion ist zudem steiler als die Sigmoid Funktion. Allerdings geht auch für diese Funktion der Gradient für $t \to \infty$ gegen 0. Die Ableitung der tanh-Funktion ist durch

$$\frac{d}{dt}\tanh t = 1 - \tanh^2 t \tag{3.48}$$

gegeben, wobei diese Funktion für $t \to \infty$ den Grenzwert 0 besitzt, dennoch weißt die Aktivierungsfunktion stabilere Gradienten an ihren Rändern auf. Hierdurch wird die tanh-Aktivierungsfunktion häufig in aktuellen neuronalen Netzen verwendet.

Eine weitere aktuelle Aktivierungsfunktion ist durch

$$a(t) = \max(0, t) \tag{3.49}$$

gegeben. Diese Funktion heißt Rectified Linear Unit Function (ReLU). Auf den ersten Blick ist die ReLU Funktion einer linearen Funktion sehr ähnlich, jedoch feuert diese Funktion nur, wenn der Eingabewert größer 0 ist. Diese Funktion eignet sich insbesondere, wenn es viele Eingaben gibt die 0 oder kleiner sind. Funktionen wie die Sigmoid- oder die Tanh-Funktion feuern auch für diese Werte. Die ReLU Funktion ist gerade dann interessant, wenn ein dünn besetzter Eingang zu erwarten ist. Die ReLU Funktion ist jedoch nicht ohne Nachteil: wegen des Phänomens der sterbenden Aktivierungsfunktion. Für Eingaben kleiner gleich 0 wird der Gradient 0 und wird im Gradientenabstieg abgeschaltet. Dieses Phänomen heißt sterbende Aktivierungsfunktion. Die Funktion wird häufig in neuronalen Netzen verwendet.

Ein Neuron wird durch einen Eingabevektor $\underline{x}$ und eine Konstante 1 angeregt. Jedem Element des Eingabevektors wird ein Gewicht aus dem Vektor $\underline{\omega}$ zugeordnet. Der Konstanten 1 wird das Gewicht b, genannt Bias, zugeordnet. Durch Anwendung der Aktivierungsfunktion $\underline{a}(\cdot)$ wird der Ausgabevektor $\underline{y}$ generiert. Der funktionale Zusammenhang ergibt sich durch

$$\underline{a}\left(\omega^T \cdot \underline{x} + 1 \cdot b\right) = \underline{y}\,. \tag{3.50}$$

Jedes KNN besitzt eine feste Anzahl von Schichten, die miteinander verknüpft sind. Die Anzahl dieser Schichten kann sich jedoch zwischen verschiedenen KNNs unterscheiden. Die einfachste Schichtenarchitektur ist durch das Multilayer Perceptron (MLP) gegeben, siehe Abbildung 3.2. Im MLP enthält jede Schicht eine Vielzahl von parallelen Neuronen. In modernen künstlichen neuronalen Netzen werden komplexere Schichtenarchitekturen verwendet. Zu den beliebtesten Architekturen gehören Convolutional Neural Networks

(CNN), Long short-term Memory (LSTM), oder Rekurrente Neuronale Netze (RNN). Diese Aufzählung ist ein kleiner Ausschnitt der vorhandenen Varianten und nicht vollständig. Aufgrund der Verwendung von CNNs in der Analyse von Bilddaten befasst sich dieses Kapitel mit CNNs. Für weitere Architekturen wird auf die einschlägige Literatur verwiesen, zum Beispiel auf Haykin [Hay99]. Eine beispielhafte CNN-Architektur ist schematisch in Abbildung 3.3 dargestellt.

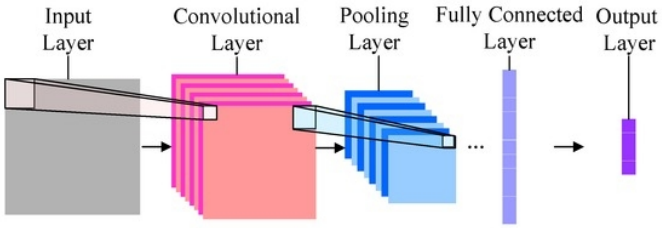


Abbildung 3.3: Beispiel einer CNN-Architektur mit unterschiedlichen Schichten aus Peng et al. [Pen16].

In Abbildung 3.3 ist eine beispielhafte CNN-Architektur dargestellt. Das Bild stammt aus der Arbeit von Peng et al. [Pen16], in der es darum geht mit Hilfe eines CNN Gesichtserkennung in Bildern im nahen Infrarotspektrum zu erkennen. Im Bild erkennt man, dass ein CNN typischerweise aus einer Eingabeschicht gefolgt von einer Faltungsschicht (engl. Convolutional Layer) und einer Poolingschicht (engl. Pooling-Layer), die sich wie in der Grafik angedeutet wiederholen können. Im Anschluss daran folgt eine Dichteschicht (engl. Fully-Connected-Layer), die die Ergebnisse der vorherigen Schichten in einem Vektor zusammenfasst, sodass in der Ausgabeschicht das Ergebnis ausgegeben werden kann.

Mit einer Faltungsschicht extrahiert ein CNN Merkmale aus den Bilddaten. Die angeschlossene Poolingschicht modelliert örtliche Zusammenhänge zwischen den Merkmalen. Mit der Dichteschicht werden die extrahierten Merkmale in einem Vektor zusammengefasst. Mit der Ausgabeschicht wird das Ergebnis des Netzes formatiert und in der gewünschten Form ausgegeben. Mit der Ausgabeschicht wird zum Beispiel garantiert, dass ein CNN für ein sechs Klassenproblem auch nur sechs Ausgabeneuronen besitzt.

Im Folgenden wird auf den generellen Aufbau eines CNN eingegangen. Ein minimales CNN besteht aus drei Schichten: einer Eingabeschicht, einer Merkmalsschicht bestehend aus einer Faltungs- und Poolingschicht und einer Dichteschicht mit Ausgabeschicht.

Die Faltungsschicht ist die Eingabeschicht für ein CNN. Diese Eingabeschicht ist speziell für zwei- bis dreidimensionale Eingabedaten entwickelt worden, wie zum Beispiel Bilddaten oder Bildsequenzen. Die Faltungsschicht ist einem biologischen rezeptivem Feld nachempfunden. Jedes Neuron wird anhand der Faltung der lokalen Umgebung eines Pixels angesprochen. Dazu wird ein Faltungsfilter über die einzelnen Pixel der Eingabe geschoben. Als Aktivierungsfunktion können beliebige Funktionen, wie oben beschrieben, eingesetzt werden. Die Faltungsschicht dient der Extraktion von Merkmalen aus den Eingabedaten. Mit der Poolingschicht werden Informationen zusammengefasst, da nicht jedes Detail für die Erkennung von Objekten wichtig ist. Zum Beispiel werden 2x2 Umgebungen zu einem Punkt zusammengefasst. Weit verbreitet ist das Max-Pooling. Hier wird das Maximum der 2x2 Umgebung weitergegeben. Dieses Vorgehen verringert den Speicherbedarf des CNN und beschleunigt die weitere Verarbeitung der Merkmale. Durch das Pooling wird Overfitting verhindert. Eine weitere Möglichkeit Overfitting in CNNs zu verhindern ist die Verwendung von sogenannten Dropout Schichten. In Dropout Schichten kann ein Anteil einer vorherigen Schicht an der Weitergabe an die nächste Schicht verhindert werden. Damit gibt es mehrere Möglichkeiten Overfitting in Neuronalen Netzen zu verhindern oder zu vermindern. Faltungs- und Poolingschichten treten in einem CNN immer gemeinsam auf. Spricht man in einem CNN von mehreren Schichten, so ist meist eine Wiederholung von mehreren Faltungs- und Poolingschichten gemeint. Auf die Faltungs- und Poolingschichten folgt im CNN als letztes eine Dichteschicht oder voll vernetzte Schicht (engl. Fully-Connected-Layer). Mit dieser voll vernetzten Schicht werden die extrahierten Informationen aus den vorherigen Schichten in einem Vektor zusammengefasst, bevor diese in die Ausgabeschicht weitergleitet werden.

Für das Training eines KNN wird eine Kostenfunktion eingesetzt. Eine häufig verwendete Kostenfunktion ist der mittlere, quadratische Fehler

$$e = \frac{1}{n} \sum_{i=1}^{n} (y_i, o_i)^2 \ , \tag{3.51}$$

wobei n der Anzahl der Dimensionen der Ausgabeschicht entspricht, die y_i sind die Einzelelemente des Ausgabevektors $\underline{y}$ für das überwachte Lernproblem und die o_i sind die Einzelausgaben der Ausgabeschicht. Im ersten Schritt des Trainings wird mit den initialen Gewichten des Netzes ein Bild durch das Netz propagiert. Das entspricht einer Prädiktion durch das Netz. Man spricht bei der Prädiktion durch das Netz auch von einer Forwardpropagation. Durch die initialen Gewichte ist zu erwarten, dass der initiale Fehler erhöht ist. Ziel der Backpropagation ist die Rückführung dieses Fehlers zur nachträglichen Anpassung der Gewichte zur Minimierung des Fehlers. Um den Fehler durch die *falschen* Gewichte zu minimieren, erfolgt die Anpassung der Gewichte durch einen Gradientenabstieg über mehrere Schritte. Die globale Optimierung dieses Fehlers ist schwierig, da eine Vielzahl von Gewichten zu optimieren ist. Aus diesem Grund kann das Training eines komplexen CNN sehr viel Zeit in Anspruch nehmen. In der Backrpopagation Prozedur wird der entstandene Fehler zu den entsprechenden Neuronen zurückgeführt und die Gewichte werden entgegen des Gradientenanstiegs der Fehlerfunktion angepasst. Hierdurch wird der Fehler schrittweise reduziert und das CNN wird an seinen dezidierten Zweck angepasst.

In dieser Arbeit kommt das VGG-16 Netz von Simonyan und Zisserman [Sim14] zum Einsatz. Die verwendete Implementierung des VGG-16 Netzes entstammt der Keras Deep Learning Bibliothek. Für dieses Netz müssen Eingabebilder auf eine quadratische Größe von 224 x 224 Pixeln verwendet werden. Die Bilder müssen drei Farbkanäle besitzen mit RGB-Werten. Das Netz verwendet 13 Faltungsschichten mit 3x3 Faltungsfiltern, von denen die meisten Faltungsschichten zum Erhalt der örtlichen Zusammenhänge von Max-Poolingschichten mit 2x2-Filtern gefolgt werden. Im Anschluss schließen sich drei Dichteschichten an. Die Ausgabe erfolgt über eine

Softmax-Ausgabeschicht. Aus der Anzahl der Schichten ergibt sich eine Anzahl von 138 Millionen Gewichten.

Um den Aufwand des Trainings so gering wie möglich zu halten wurde in dieser Arbeit ein Vorinitialisiertes Netz verwendet. Für die Vorinitialisierung wurden die Gewichte aus dem Imagenet Datensatz von Deng et al. [Den09] verwendet. Die genauen Vorverarbeitungsschritte für dieses Netz finden sich in Kapitel 6.

3.4 Gaußprozess

Gaußprozesse gehören zur Klasse der stochastischen Prozesse. Aufgrund des Aufbaus können Gaußprozesse als Verteilung über dem Funktionenraum betrachtet werden. Grundsätzlich besteht ein Gaußprozess aus normalverteilten Zufallsvariablen. Jede finite Untermenge dieser Zufallsvariablen in einem Gaußprozess besitzt eine normalverteilte Verbunddichte [Ras06]. Daraus ergibt sich die nützliche Eigenschaft, dass jede marginale Verteilung eines Gaußprozesses ebenfalls wieder normalverteilt ist. Diese nützliche Eigenschaft macht den Gaußprozess zu einem mächtigen Werkzeug im maschinellen Lernen. Insbesondere werden Gaußprozesse für Regressionsprobleme eingesetzt. Um einen Gaußprozess (GP) vollständig zu definieren, genügt die Angabe einer Mittelwert- und Kovarianzfunktion. Durch

$$m(x) = \mathbb{E}\left[f(x)\right] \tag{3.52}$$

$$k(x,x') = \mathbb{E}\left[(f(x) - m(x)) \cdot (f(x') - m(x'))\right] \ , \tag{3.53}$$

werden die Mittelwertfunktion $m(\cdot)$ als Erwartungswert eines reellen Prozesses $f(x)$ und die Kovarianzfunktion $k(\cdot,\cdot)$ als Kovarianzmatrix über dem

reellen Prozess $f(x)$ definiert. Unter Verwendung dieser Definition kann die Verteilung der Funktionen $f(x)$ als Gaußprozess durch

$$f(x) \sim GP\left(m(x), k(x,x')\right) \tag{3.54}$$

angegeben werden. Diese Schreibweise wird als Funktionsraumschreibweise [Ras06] bezeichnet. Mit den Kovarianz- und Mittelwertfunktionen können Eigenschaften der zu schätzenden Funktion beeinflusst werden, wie zum Beispiel Glattheit und Stationarität.

Sei eine beliebige Funktion

$$y = g(x) + \epsilon \tag{3.55}$$

gegeben, wobei $g(x)$ eine nichtlineare Funktion und x typischerweise ein Vektor ist. Hier wird ein Skalar für x verwendet und ϵ ist ein mittelwertfreier, normalverteilter Rauschterm mit $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Unter der Verwendung von Trainingsdaten $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ kann ein Gaußprozess verwendet werden, um die unbekannte Funktion $g(x)$ zu schätzen. Die Trainingsdaten beinhalten Wertepaare, die als Messung von der zu approximierenden Funktion bestimmt werden. Die Trainingsdaten erlauben die Verwendung eines überwachten Lernverfahrens. Bei einem Gaußprozess sei darauf verwiesen, dass es sich um einen sogenannten Lazylearner handelt. Die Trainingsdaten werden erst bei der Prädiktion verwendet. Der trainierte GP für $g(x)$ bestimmt eine posteriore Verteilung

$$y \sim \mathcal{N}\left(\mu_y, \sigma_y^2\right) \tag{3.56}$$

für die resultierenden Funktionswerte. Die Funktionswerte können basierend auf einer Menge von x-Werten abgeleitet werden. In Abbildung 3.4 sind sechs Trainingspunkte dargestellt. Die Trainingspunkte wurden zufällig

aus einer Standardnormalverteilung ausgewählt. Für die Prädiktion wurden 100 x-Werte äquidistant im Intervall $\left[\min(x^{\text{train}}), \max(x^{\text{train}})\right]$ gesammelt. Der GP wurde verwendet, um den Mittelwert und die Standardabweichung an jeder Stelle x des Gaußprozesses zu ermitteln. In der Abbildung sind die Mittelwerte für jedes x des Gaußprozesses, sowie die 2σ-Grenze der Schätzung abgebildet.

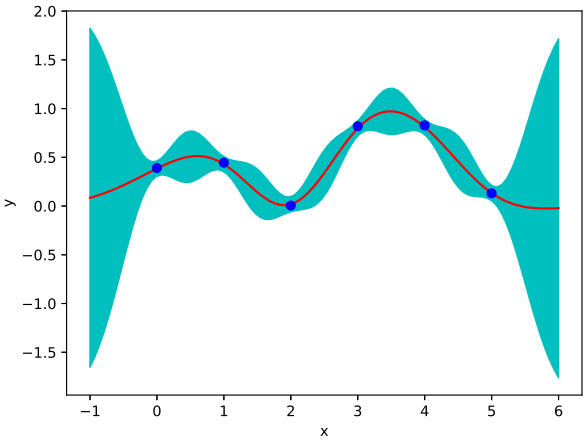


Abbildung 3.4: Gaußprozess basierend auf sechs Trainingspunkten. Prädiktion auf 100 äquidistanten x-Werten unter Angabe des Mittelwertes und der $2 \cdot \sigma$-Grenze.

In der Abbildung zeigt sich, dass die Unsicherheit nicht für jeden Testwert konstant ist. Die Unsicherheit ist kleiner innerhalb des Bereichs, der durch die Trainingsdaten gegeben ist. An den Rändern steigt die Unsicherheit an, je weiter sich die Eingabedaten von den Trainingsdaten entfernen. Durch eine geschickte Wahl von Trainingsdaten kann die Qualität des Gaußprozesses beeinflusst werden und die Unsicherheit der Schätzung verringert werden. Die durch den Mittelwert gegebene Funktionskurve scheint glatt zu sein. Für dieses Beispiel wurde als Kovarianzfunktion eine Radiale-Basis-Funktion (RBF) verwendet. Es handelt sich um eine stationäre Kovarianzfunktion. Unter Stationarität versteht man die Annahme, dass ein stochastischer Prozess zeitinvariant ist. Man geht also davon aus, dass zu jedem Zeitpunkt der Mittelwert

und die Varianz unabhängig von der zeitlichen Entwicklung sind. Diese Eigenschaften wie Stationarität und Glattheit können durch die Wahl der Kovarianzfunktion des GP beeinflusst werden. Aus diesem Grund werden im Folgenden unterschiedliche Kovarianzfunktionen eingeführt.

3.4.1 Kovarianzfunktionen

Betrachtet man eine zu schätzende Funktion, werden Punkte, die nahe beieinander liegen, ähnliche Funktionswerte aufweisen. Es ist wichtig, die Nähe oder Ähnlichkeit von Eingabewerten beurteilen zu können. Die Kovarianzfunktion eines Gaußprozesses hat genau diese Aufgabe. Die Funktion beinhaltet eine Beurteilung der Ähnlichkeit zwischen Eingabewerten [Ras06]. Rasmussen [Ras06] schränkt ein, dass nicht jede Funktion, die den Abstand zwischen zwei Eingabewerten x und x' enthält, eine Kovarianzfunktion ist.

Um eine gültige Kovarianzfunktion zu erhalten, muss das Ergebnis positivsemidefinit sein. Weiter kann eine Kovarianzfunktion als stationär bezeichnet werden, wenn die Funktion nur vom Abstand $x-x'$ abhängt. Hängt die Funktion strenger von $|x-x'|$ ab, so spricht man von einer isotropen, stationären Kovarianzfunktion. Hängt die Kovarianzfunktion vom Skalarprodukt $x \cdot x'$ ab, so spricht man von einer Skalarprodukt Kovarianzfunktion. Rasmussen behandelt in [Ras06] eine weitere Eigenschaft für die Kovarianzfunktion. Er betont, dass die Funktion Mean-Squared-Continuous (MS-kontinuierlich) ist. Das bedeutet für eine Kovarianzfunktion $f(x)$ an einer Stelle x_* muss

$$\mathbb{E}\left[|f(x_k) - f(x_*)|^2\right] \to 0 \tag{3.57}$$

gelten, für $|x_k - x_*| \to 0$ mit $k \to \infty$. Für eine genauere Behandlung der MS-Kontinuitätseigenschaft sei auf Rasmussen [Ras06]verwiesen.

3.4.1.1 Squared-Exponential-Kernel

Die Squared-Exponential (SE) Kovarianzfunktion ist eine stationäre Kovarianzfunktion. Sie hängt nur von $r = |x - x'|$ ab. Die SE-Kovarianzfunktion ist durch

$$k_{SE}(r) = \exp\left(-\frac{r^2}{2 \cdot l^2}\right) \tag{3.58}$$

definiert, wobei der Parameter l als charakteristische Längenskalierung bezeichnet wird. Der SE-Kernel gehört zu den Radial-Basis-Funktion-Kovarianzfunktionen. Eine praktische Eigenschaft dieser Kovarianzfunktion ist die unendliche Differenzierbarkeit, wodurch dieser Kernel als sehr glatt gilt [Ras06]. Nach Rasmussen gibt es Argumente dafür, dass diese übermäßige Glattheit sich für manche physikalischen Prozesse nicht eignet, weshalb häufig auf die Kovarianzfunktionen der Matérn-Klasse zurückgegriffen wird.

3.4.1.2 Matérn-Klasse

Die Matérn-Klasse der Kovarianzfunktionen ist durch

$$k_{Matern}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{l}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu} r}{l}\right) \tag{3.59}$$

gegeben, mit den Parametern ν, l und der modifizierten Besselfunktion K_{ν}. Lässt man ν gegen unendlich gehen, so ergibt sich k_{Matern} zur SE-Kovarianzfunktion. Für die Differenzierbarkeit der Matérn-Funktion muss $\nu < k$ gelten. Dann ist die Funktion k-fach differenzierbar. Die Matérn-Funktion vereinfacht sich, wenn $\nu = p + \frac{1}{2}$ gilt und p eine nichtnegative Ganzzahl ist. In diesem Fall vereinfacht sich k_{Matern} zu dem Produkt einer Exponentialfunktion und einem Polynom der Ordnung p. Für eine genaue Beschreibung sei auf [Ras06] verwiesen. Häufig verwendete Werte für ν sind

$\frac{3}{2}$ und $\frac{5}{2}$. Für $\nu = \frac{1}{2}$ wird die Matérn-Funktion sehr rau und für $\nu > \frac{7}{2}$ nicht mehr gut zu parametrisieren [Ras06].

Durch den SE-Kernel und die Matérn-Funktion wurden stationäre Kovarianzfunktionen beschrieben. Die Liste an Kovarianzfunktionen ist nicht vollständig, in Rasmussen [Ras06] werden weitere Kernel-Methoden betrachtet. Zur Vervollständigung der Behandlung wird im Folgenden eine nicht-stationäre Kovarianzfunktion betrachtet.

3.4.1.3 Dot-Produkt-Kovarianzfunktion

Zu den Dot-Produkt-Kovarianzfunktionen zählen die Kernel-Funktionen, die durch

$$k(x,x') = \sigma_0^2 + x \cdot x' \tag{3.60}$$

definiert sind, wobei σ_0^2 ein Parameter ist. Die Gleichung ist ein Resultat der linearen Regression. Für den Fall, dass $\sigma_0^2 = 0$ gilt, spricht man von einem homogenen linearen Kernel. Für $\sigma_0^2 \neq 0$ ist von inhomogenen linearen Kernelfunktionen die Rede. Unter Verwendung einer allgemeinen Kovarianzfunktion K_p kann die lineare Kernel-Funktion zu

$$k(x,x') = \sigma_0^2 + x \cdot K_p \cdot x' \tag{3.61}$$

verallgemeinert werden.

Der lineare Kernel ist eine einfache, nichtstationäre Kovarianzfunktion. Eine weitaus komplexere, nichtstationäre Kovarianzfunktion ist die neuronale Netz-Kovarianzfunktion, die in Rasmussen [Ras06] nachgelesen werden kann.

Neben den Kovarianzfunktionen kommen weiter Mittelwertfunktionen zum Einsatz. Eine häufig Verwendete Mittelwertfunktion, ist die Null-Mittelwertfunkion. Diese Mittelwertfunktion gibt für jede Eingabe den Wert

Null zurück. Weitere Mittelwertfunktionen sind die konstanten Mittelwertfunktionen, die immer einen konstanten Wert zurückliefern, wovon die Nullmittelwertfunktion ein Spezialfall ist. In Abhängigkeit von dem betrachteten Problem, das mit einem Gaußprozess gelöst werden soll, kommt häufig die Nullmittelwertfunktion zum Einsatz. In dieser Arbeit wird ein Gaußprozess dazu verwendet, um eine Abbildung von einem Emotionsdeskriptor basierend auf Gesichtsbildern in den Valenz-Erregungs-Raum zu schätzen. Hierbei kommt die Nullmittelwertfunktion zusammen mit der Radial-Basis-Kovarianzfunktion zum Einsatz.

4 Tracking von Punkt- und ausgedehnten Objekten

Tracking bezeichnet eine Methode, um Objekte automatisch zu verfolgen. Unter Tracking kann man die Verfolgung von bewegten Objekten verstehen, als auch von internen Zuständen des Objekts. Vertreter von bewegten Objekten sind beispielsweise Fahrzeuge oder Flugzeuge. Vertreter interner Zustände können Maschinenparameter oder die menschliche Emotion sein. Um ein bewegtes Objekt wie ein Flugzeug zu tracken, werden Informationen über den aktuellen Aufenthaltsort des Flugzeugs benötigt. Liegen solche Positionsinformationen vor, spricht man von einem direkt beobachtbaren Zustand. Im Fall von internen Zuständen wie der menschlichen Emotion, kann diese nur indirekt beobachtet werden, zum Beispiel durch das Vorliegen eines spezifischen Gesichtsausdrucks.

Damit diese unterschiedlichen Objekte getrackt werden können, muss das zu verfolgende Objekt durch einen Systemzustand repräsentiert werden. Dieser Systemzustand wird durch die Systemfunktion in einen konsekutiven Zustand überführt. Das Zusammenspiel des Systemzustands und einer Systemfunktion wird als Systemmodell bezeichnet. Im allgemeinen wird davon ausgegangen, dass der Systemzustand sich mit fortschreitender Zeit verändert. Zur Vereinfachung der zeitlichen Veränderung wird im Tracking von einem zeitdiskreten Modell ausgegangen. Da jede Modellierung einem stochastischen Fehler unterliegt, wird dieser Umstand durch einen additiven,

normalverteilten Rauschterm modelliert. Das zeitdiskrete Systemmodell wird durch

$$\underline{x}_{k+1} = f\left(\underline{x}_k\right) + \underline{\omega}_k \tag{4.1}$$

repräsentiert, wobei $f(\cdot)$ eine beliebige Funktion, $\underline{\omega}_k$ eine gaußverteilte Rauschvariable und $\underline{x}_k$ der Systemzustand zum Zeitpunkt k sind. Die Modellierung lässt sowohl lineare als auch nichtlineare Systemfunktionen zu. Die Auswahl der Funktion hängt maßgeblich von dem zu modellierenden Modell ab. Wird eine lineare Veränderung des Systemzustands erwartet, so kann eine lineare Funktion gewählt werden. Geht man von nichtlinearen Änderungen des Systemzustands aus, ist eine nichtlineare Systemfunktion zu wählen. Ein Beispiel für ein lineares Systemmodell ist in Abschnitt 3.2 beschrieben. Ein einfaches Beispiel für ein nichtlineares System ist die Wurfparabel. Die Wurfparabel ergibt sich beim schiefen Wurf eines Balls oder beliebigen Festkörpers. Die vektorielle Systemfunktion für die Wurfparabel ist durch

$$\underline{r}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} v_0 \cdot t \cdot \cos(\beta) \\ v_0 \cdot t \cdot \sin(\beta) - \frac{g}{2} \cdot t^2 \end{pmatrix} \tag{4.2}$$

gegeben, wobei v_0 die Geschwindigkeit während des Abwurfs, β der Abwurfwinkel und g die Erdbeschleunigung sind.

Durch das Systemmodell (4.1) wird der Systemzustand $\underline{x}_k$ zum Zeitpunkt k in den nächsten Zeitpunkt $\underline{x}_{k+1}$ überführt. Das Systemmodell modelliert das autonome Verhalten des zu beschreibenden Objekts. In einem Trackingmodell entspricht das der Beschreibung der angenommenen Bewegung des zu beobachtenden Objekts. Der Systemzustand ist die minimale vollständige Repräsentation des Systems. Man unterscheidet in Trackingmodellen zwischen direkt und indirekt beobachtbaren Zuständen. Ein direkt beobachtbarer Zustand enthält nur Systemkomponenten, die direkt durch ein Messsystem beobachtet werden können. Indirekt beobachtbare Zustände können nur durch einen funktionalen Zusammenhang auf den Systemzustand zurückgeführt werden.

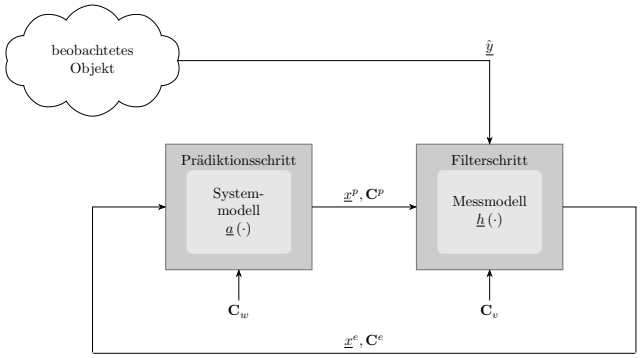


Abbildung 4.1: Allgemeines Trackingmodell zur Verdeutlichung des Zusammenhangs zwischen dem beobachteten Objekt und dem Tracker.

Dieser funktionale Zusammenhang wird durch die Messabbildung definiert. Das Messmodell wird durch

$$\underline{y}_k = h\left(\underline{x}_k\right) + \underline{v}_k \tag{4.3}$$

definiert. Die Messung $\underline{y}_k$ ergibt sich durch Anwendung des Messmodells $h(\cdot)$ auf den Systemzustand und einem additiven, gaußverteilten Rauschterm $\underline{v}_k$. Durch den Rauschterm werden nichtsystematische Messunsicherheiten modelliert. Mit der Definition der Messabbildung (4.3) sind alle Komponenten des Trackingmodells beschrieben. Das Zusammenspiel der Einzelkomponenten, sowie aller möglichen Einflüsse auf das Trackingsystem sind in Abbildung 4.1 dargestellt.

Abbildung 4.1 zeigt ein Trackingmodell, dass in einen Prädiktions- und Filterschritt aufgeteilt ist. Im Prädiktionsschritt wird das Systemmodell verwendet, um den aktuellen Zustand $\underline{x}^p$, $\mathbf{C}^p$ des beobachteten Systems zu prädizieren. Im Filterschritt wird die Messfunktion verwendet, um den modellierten Zustand auf eine Messung des beobachteten Objekts abzubilden. Im Filterschritt wird ebenso die aktuelle Messung $\hat{\underline{y}}$ verwendet. Durch die Verwendung der

Messung des Originalsystems wird im Tracker basierend auf dem prädizierten Zustand eine verbesserte Schätzung $\underline{x}^e$, $\mathbf{C}^e$ des Systemzustands bestimmt. Die Kovarianzmatrizen $\mathbf{C}_w$ und $\mathbf{C}_v$ charakterisieren das System- und Messrauschen für das System- und Messmodell des Trackers.

Tracking kann auf verschiedene dynamische Systeme angwendet werden. Ein klassischer Anwendungsfall ist das Tracking von Flugzeugen, wobei hier ein dynamisches Objekt beobachtet wird.

Ein weiterer Anwendungsfall ist das statische Tracking. Beim statischen Tracking werden zum Beispiel Parameter einer Kamera getrackt, um so Aussagen über die Aufnahmen zu treffen. Aus diesem Grund wird in Abschnit 4.1 ein Tracker vorgestellt, der dazu dient, Bild-zu-Bild-Transformationen in einer Bildfolge zu schätzen.

Abschnitt 4.2 behandelt ein auf Multilaterationsmessungen basierendes Flugzeugtracking. Das Flugzeugtracking gehört zur Klasse der Punktzieltracker, bei denen ein Objekt als Punkt modelliert wird. Eine Modellierung als Punktziel ist nur dann nützlich, wenn keine weiteren Analysen des beobachteten Objekts notwendig sind. In einem solchen Fall kann die Gestalt des Objekts wichtig sein.

Ein Beispiel hierfür ist das Tracking der Iris des menschlichen Auges. Aus der Position und Gestalt der Iris lassen sich verschiedene Informationen ableiten. Zum Beispiel kann der Blickwinkel geschätzt werden, zum Anderen kann auch der Öffnungswinkel des Auges bestimmt werden, indem der Anteil der sichtbaren Iris, eingerahmt von den Augenlidern betrachtet wird.

Um die Gestalt eines Objekts tracken zu können, wird die Technik des Trackings ausgedehnter Objekte eingesetzt. In Abschnitt 4.3 wird das Tracking ausgedehnter Objekte am Beispiel eines Iris-Trackers eingeführt. Ein weiterer Anwendungsfall für das Tracking ausgedehnter Objekte wird im Abschnitt 4.4.3 diskutiert. Hier wird das Tracking des Gesichts untersucht, wobei hier die Notwendigkeit von Nebenbedingungen für das Gesichtsmodell herausgearbeitet wird.

4.1 Statisches Tracking: Homographieschätzung zur Bildregistrierung

Für ein internes Projekt des Fraunhofer IOSB wurde ein statisches Trackingverfahren entwickelt, mit dem sogenannte Homographien für Bildsequenzen geschätzt werden können. Homographien beschreiben eine projektive Abbildung zwischen zwei Bildern einer Bildsequenz, in der eine Bewegung der Kamera durchgeführt wurde. Das heißt, eine Homographie setzt einen Punkt im Ursprungsbild mit einem korrespondierenden Punkt im Zielbild in Zusammenhang. Somit ist eine Grundvoraussetzung für die Berechnung der Homographie eine ausreichend große Menge von Korrespondenzen zwischen Punkten, die sowohl im Ursprungsbild als auch im Zielbild vorhanden sind. Das Finden solcher Punktkorrespondenzen ist ein fundamentales Problem der Bildverarbeitung und findet große Beachtung im Zusammenhang mit der Verarbeitung von Stereobildern oder des optischen Flusses.

Im Zusammenhang mit dem internen Projekt des Fraunhofer IOSB tritt dieses Korrespondenzproblem im Zusammenhang mit Infrarotsequenzen auf. Die im Projekt verwendeten Infrarotsequenzen wurden mit Hilfe des Deutschen Zentrums für Luft- und Raumfahrt (DLR) in Braunschweig angefertigt. Dazu wurde eine mit Infrarot-Kameras bestückte Messkugel an einem Helikopter angebracht. Mit Hilfe des Helikopters wurden Anflüge auf Objekte durchgeführt, die in einer Waldumgebung aufgestellt wurden. Der hierdurch entstehende Nachteil resultiert in Bildern mit geringem Kontrast und somit erhöhter Schwierigkeit Punktkorrespondenzen zu finden. Aus diesem Grund wurde ein Verfahren entwickelt, mit dem bekannte GPS-Koordinaten in das Bild projiziert und somit Punktkorrespondenzen festgelegt werden konnten. Die Schwierigkeit bestand darin, eine Abbildung zu finden, mit der sich GPS-Koordinaten in das Bildkoordinatensystem abbilden lassen. Zur Verdeutlichung der Kontrastsituation in den verwendeten Bilddaten zeigt Abbildung 4.2 das erste und letzte Bild eines Anflugszenarios. In der Mitte des ersten Bildes ist mit geringem Kontrast die Fläche (Schneise im Wald) im letzten Bild zu sehen, auf dem sich die gesuchten Objekte finden.

(a) Erstes Frame **(b)** Letztes Frame

Abbildung 4.2: Detailvergleich zwischen dem ersten Frame (a) und dem letzten Frame (b). Die Markierungen zeigen besondere Punkte, die zur Registrierung verwendet werden.

Betrachtet man Abbildung 4.2, erkennt man, dass Objekte, die im ersten Bild als Punktziel repräsentiert werden im letzten Bild als ausgedehntes Objekt zu sehen sind. Das heißt, das aus einem einzigen Pixel für ein Objekt ein ganzer Pixelhaufen entstanden ist. Ebenso lässt sich erkennen, dass die Kontrastverhältnisse sehr stark vom Hintergrund des Objekts und dem Detailgrad der betrachteten Szene abhängen. Aus diesem Grund werden die Homographien nur zwischen aufeinanderfolgenden Bildern der Bildsequenz berechnet, um solche Effekte abzufangen und aufzuweichen.

In der klassischen Schätzung von Homographien werden sehr viele Korrespondenzkandidaten ermittelt mit Interestpoint Detektoren, wie dem Förstner-Filter von Förstner in [Fös87]. Modernere Verfahren wie das von Lowe in [Low99] vorgestellte SIFT Verfahren erstellen hochdimensionale Deskriptorvektoren für Korrespondenzkandidaten, um ein genaueres Matching der Punkte zu ermöglichen. Die Berechnung der Homographie erfolgt dann über wohlbekannte Methoden, die beispielsweise in Hartley et al. [Har04] nachgelesen werden können. In dem hier vorgestellten Verfahren werden GPS Fixpunkte in jedes Bild der Sequenz projiziert, um somit feste Korrespondenzen zu erzeugen und somit das Korrespondenzproblem zu umgehen.

Im folgenden Abschnitt wird die Methode hergeleitet mit der sich die GPS-Fixpunkte in die Bildsequenz projizieren lassen.

4.1.1 Projektion von GPS-Punkten in das Bildkoordinatensystem

GPS steht für Global-Positioning-System. Punkte dieses Systems bestehen aus zwei Winkelgrößen und einer Höhe. Die Winkelgrößen sind die Latitude und Longitude, die die Lage des Objekts bezüglich der Längen- und Breitengrade des WGS-84 Erdellipsoiden beschreiben. Das Ziel dieses Abschnitts ist die Projektion von einzelnen GPS-Punkten in das Bildkoordinatensystem. Diese projizierten Punkte werden als Punktkorrespondenzen zur Berechnung von Homographien verwendet. Um die GPS-Punkte in das Bild projizieren zu können, wird eine Abbildung vom GPS-Koordinatensystem in das Bildkoordinatensystem benötigt. Im Folgenden wird das GPS-Koordinatensystem als Weltkoordinatensystem bezeichnet. Die gesuchte Abbildung wird durch

$$f : \underline{x}_{GPS} \rightarrow \underline{x}_{CAM} \tag{4.4}$$

definiert. $\underline{x}_{GPS}$ ist ein Vektor im Weltkoordinatensystem und $\underline{x}_{CAM}$ ein Vektor im Bildkoordinatensystem. Bei den Weltkoordinaten handelt es sich um Kugelkoordinaten. Das Zielkoordinatensystem ist ein kartesisches Koordinatensystem. Durch Rotationen werden die Latitude und Longitude in das Kamerakoordinatensystem überführt. Durch

$$\mathbf{R}_{\text{lat}}(x) = \begin{bmatrix} \sin(x) & 0 & -\cos(x) \\ 0 & 1 & 0 \\ \cos(x) & 0 & \sin(x) \end{bmatrix}, \tag{4.5}$$

die Latitude so ausgerichtet, das sie der x-Koordinate des Kamerasystems entspricht. Mit

$$\mathbf{R}_{\text{lon}}(y) = \begin{bmatrix} \cos(y) & \sin(y) & 0 \\ -\sin(y) & \cos(y) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

wird die Longitude gedreht, dass sie der y-Koordinate der Kamera entspricht. Durch Multiplikation von (4.5) und (4.6) können die beiden Rotationen zu einer Gesamtrotation

$$\mathbf{R}_{\text{world}\rightarrow\text{cam}} = \mathbf{R}_{\text{lat}} \cdot \mathbf{R}_{\text{lon}} \tag{4.7}$$

kombiniert werden. Durch die Rotation wird ein Vektor $\underline{x}_{\text{GPS}}$ des Weltkoordinatensystems derart rotiert, dass die Kamera im Koordinatenursprung liegt. Dadurch befinden sich sowohl die Kamera als auch die Weltkoordinaten im gleichen Koordinatensystem. Durch die Rotation können Koordinaten im Weltkoordinatensystem in das Kamerakoordinatensystem überführt werden. Die Projektion in das Bildkoordinatensystem ist noch nicht erfolgt.

Es muss geprüft werden, ob sich die angegebenen GPS-Koordinaten innerhalb des Blickwinkels der Kamera befinden. Zunächst wird das Kamerakoordinatensystem so rotiert, dass die Ausrichtung des Systems der Blickrichtung der Kamera entspricht. Im Messsystem ist ein Inertialsystem verbaut. Das Inertialsystem misst Winkelgeschwindigkeiten. Unter Verwendung eines Kalman-Filters im Bootstrap-Algorithmus werden die Lagewinkel (Roll, Nick und Gier) bestimmt. Durch α wird der Rollwinkel bezeichnet, durch β der Nickwinkel und durch γ der Gierwinkel. Die Reihenfolge der Rotationen muss eingehalten werden. Bevor die Lagewinkelrotationen definiert werden, müssen zunächst die Achsenrichtungen definiert werden, sodass die x- und y-Achsen dem Bildkoordinatensystem entsprechend ausgerichtet werden. Im Bildkoordinatensystem befindet sich der Ursprung in der unteren, linken Ecke. Somit wird die x-Achse so ausgerichtet, dass sie vom Ursprung aus nach rechts zeigt. Die y-Achse zeigt nach oben im Bild, und muss entsprechend ausgerichtet werden. Die z-Achse zeigt in die Blickrichtung der Kamera. Die Achsenausrichtung wird durch

$$\mathbf{R}_S = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \tag{4.8}$$

auf die Koordinatenachsen angewendet. Nach Ausrichtung der Koordinatenachsen werden Roll-, Nick- und Gierwinkel verwendet, um die Ausrichtung des Kamerakoordinatensystems an die tatsächliche Lage der Kamera anzupassen. Für jeden der Lagewinkel wird eine Rotationsmatrix definiert. Die Rotationsmatrix für den Rollwinkel ist durch

$$\mathbf{R}_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{4.9}$$

definiert. Durch

$$\mathbf{R}_\beta = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{4.10}$$

wird die Rotationsmatrix für den Nickwinkel beschrieben. Mit

$$\mathbf{R}_\gamma = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

wird die Rotationsmatrix des Gierwinkels bestimmt. Die Reihenfolge der Winkel ergibt sich aus der Definitionsreihenfolge der Winkel. Multipliziert man die Rotationsmatrizen entsprechend dieser Reihenfolge kann die vollständige Rotation bestimmt werden. Mit der Rotation

$$\mathbf{R}_{\text{cam}} = \mathbf{R}_S \cdot \mathbf{R}_\alpha \cdot \mathbf{R}_\beta \cdot \mathbf{R}_\gamma \cdot \mathbf{R}_{\text{world}\to\text{cam}} \tag{4.12}$$

können Koordinaten im Weltkoordinatensystem in das dem Blickwinkel der Kamera entsprechende Kamerakoordinatensystem überführt werden.

Nachdem die GPS-Koordinaten jetzt in das ausgerichtete Kamerakoordinatensystem transformiert werden können, müssen diese Koordinaten nur noch in das Bildkoordinatensystem projiziert werden. Hierzu wird eine weitere Matrix benötigt: die Kameramatrix. Die Kameramatrix setzt sich aus zwei Teilen zusammen, der Kalibriermatrix und einer Translation. Um die Kalibriermatrix aufzustellen, müssen die Brennweite der Kamera und der Bildhauptpunkt bekannt sein. Die Brennweite des verwendeten Kamerasystems lässt sich durch Vermessung des Kamerachips ermitteln. Aufgrund der rechteckigen Form ergeben sich unterschiedliche Brennweiten f_x und f_y für die x- und y-Richtungen. Durch Anwendung der Epipolargeometrie [Har04] lässt sich der Bildhauptpunkt $\left[p_x, p_y\right]^T$ des Kamerasystems ermitteln. Somit lässt sich die Kalibriermatrix mit

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_x \\ 0 & 0 & 1 \end{bmatrix} \tag{4.13}$$

zusammensetzen. Die Kameramatrix benötigt noch eine weitere Komponente, eine Translation. Diese Translation wird als Verschiebung des Koordinatenursprungs angesehen. Der Koordinatenursprung des Kamerasystems muss im Zentrum der Kamera liegen. Damit wird durch

$$\mathbf{K}_t = \left[\mathbf{I}\right| - \underline{x}^c\right] = \begin{bmatrix} 1 & 0 & 0 & -x^c \\ 0 & 1 & 0 & -y^c \\ 0 & 0 & 1 & -z^c \end{bmatrix} \tag{4.14}$$

die Translation des Koordinatenursprungs beschrieben. Die vollständige Projektion wird durch

$$\mathbf{P}_t = \mathbf{K} \cdot \mathbf{R}_{\text{cam}} \cdot \mathbf{K}_t \tag{4.15}$$

definiert. Durch Anwendung der Projektionsmatrix $\mathbf{P}_t$ lassen sich GPS-Koordinaten in das Bild projizieren. Aufgrund der Anpassung an die aktuelle Rotation der Kamera kann dies für jedes einzelne Bild einer kompletten Bildsequenz durchgeführt werden. Somit lassen sich auch nachträglich in der Nachbearbeitung GPS-Punkte in das Bild projizieren, da auch hier für jedes Bild die aktuelle Position der Kamera bekannt ist. Daher können in jedem Bild korrespondierende Punkte generiert werden, anhand derer sich die Bild-zu-Bild Transformationen, die Homographien, berechnen lassen. Diese Homographien können nun für unterschiedliche Anwendungsfälle verwendet werden: Zum Beispiel lässt sich das Bild nachträglich stabilisieren, zusätzlich können so Markierungen im Bild gesetzt werden, die über die gesamte Laufzeit der Bildsequenz stabil an einem Ort gehalten werden können. Im Zusammenhang mit dieser stabilen Platzierung von Markierungen wurden Experimente getätigt, die zeigen sollten, dass der gewählte Ansatz Vorteile generiert im Vergleich zu gängigen Korrespondenztrackern, wie SIFT oder SURF-Merkmalen. Als Anwendungsfall für die Experimente wurden Beobachterversuchsdaten verwendet.

4.1.2 Experimente zur GPS basierten Homographieberechnung

Für die Experimente wurden insgesamt zwei Bildsequenzen verwendet. Die erste Bildsequenz wurde bei Tag aufgenommen und die zweite Bildsequenz bei Nacht. In beiden Sequenzen wurden drei signifikante Objekte als Referenzpunkte verwendet: eine Stadt (t_1), ein Turm (t_2) und eine Kaserne (t_3). Alle drei Objekte wurden in jedem Bild durch Experten annotiert. Zur Berechnung der Homographien für die beiden Sequenzen wurde die RANSAC (Random Sample Consensus) Prozedur von Fischler et al. [Fis81] verwendet. Die notwendigen Punktkorrespondenzen wurden mit der vorgestellten Methode (GPS), anhand von SIFT-Merkmalen (SIFT) und anhand von SURF-Merkmalen bestimmt. Für alle drei Methoden wurde die durch Experten annotierte Position der drei Objekte im ersten Bild der Sequenz verwendet. Die Objektlokalisationen wurden mittels der auf den Korrespondenzen berechneten Homographien bis zum letzten Bild der Sequenz transformiert. In jedem Bild wurde der

Tabelle 4.1: Ergebnisse der durch die Homographien induzierten Pixelfehler in der bei Tag aufgenommenen Bildsequenz.

Objekt	Methode	Mittelwert	Std	Median	Max
t_1	GPS	1.613	1.215	1.366	12.094
	SIFT	1.099	0.956	0.882	13.121
	SURF	1.112	0.960	0.891	13.122
t_2	GPS	2.300	1.725	1.883	13.418
	SIFT	2.024	1.602	1.638	13.019
	SURF	2.036	1.610	1.642	12.945
t_3	GPS	1.793	1.141	1.595	14.092
	SIFT	1.282	0.994	1.065	13.986
	SURF	1.294	1.992	1.070	13.687

Pixelfehler jedes einzelnen Objekts bestimmt. In Tabelle 4.1 werden die Mittelwerte der Pixelfehler, sowie die Standardabweichungen, der Median und der Maximalwert der Pixelfehler für die Tagsequenz aufgelistet.

Beim Betrachten der Tabelle 4.1 fällt auf, dass die mittleren und medianen Fehlerwerte zwar etwas größer sind als die der Methoden des Stands der Technik, aber dennoch Ergebnisse in der gleichen Größenordung zu erwarten sind. Erwähnenswert ist auch, dass das vorgeschlagene Verfahren auf eine geringere Anzahl an Korrespondenzpunkten zurückgreift, wohingegen die Vergleichsmethoden eine deutlich höhere Anzahl an Korrespondenzpunkten ansprechen. Die mittleren Fehler sind dennoch etwas höher als man es bei hohem Kontrast erwartet. Allerdings heben sich die interessierenden Objekte nicht sehr stark vom Hintergrund ab, da dieser auch einen sehr hohen inneren Kontrast aufweist.

Ein vergleichbares Experiment wurde unter Verwendung einer Nachtsequenz durchgeführt. Aufgrund der fehlenden Sonneneinstrahlung und weiteren Effekten wie den *kalten Himmel* ist der Kontrast einer Nachtsequenz deutlich geringer im Vergleich zu einer Tagsequenz. Aus diesem Grund wird die Ermittlung von Punktkorrespondenzen zwischen zwei Bildern einer Sequenz deutlich schwieriger. Objekte die eine höhere Kerntemperatur vorweisen, emittieren natürlich in der Nacht in gleichem Maße IR-Strahlung wie auch bei Tag. Somit steigt konsequenterweise der Kontrast zu diesen Objekten, wobei der Hintergrund mit einem deutlich schwächerem Kontrast nahezu verschwindet.

Tabelle 4.2: Ergebnisse des Homographie-Experiments mit einer Nachtsequenz.

Objekt	Methode	Mittelwert	Std	Median	Max
t_1	GPS	1.231	0.994	0.963	6.799
	SIFT	0.771	0.644	0.559	3.837
	SURF	0.792	0.644	0.595	3.911
t_2	GPS	1.074	0.883	0.765	5.278
	SIFT	0.859	0.802	0.556	4.697
	SURF	0.870	0.803	0.563	4.740
t_3	GPS	1.040	0.684	0.892	5.589
	SIFT	0.729	0.525	0.590	4.103
	SURF	0.756	0.550	0.613	4.467

Es ist also zu erwarten, dass die Methoden des Stands der Technik hauptsächlich Korrespondenzen im Bereich der Objekte mit hohem Kontrast ermitteln können und dass alle Methoden erfolgreich in der Lage sind gute Ergebnisse zu erzielen. Die Abweichungsergebnisse können in Tabelle 4.2 betrachtet werden.

Die Verbesserung des Objektkontrasts verbessert die Ergebnisse aller Methoden signifikant gegenüber der Tagsequenz. Auch hier weißt der Stand der Technik bessere Ergebnisse gegenüber der vorgeschlagenen Methodik vor. Dennoch bleibt zu sagen, dass mit der vorgeschlagenen Methode (GPS) vergleichbare Ergebnisse mit 6 projizierten Punkten im Vergleich zum Stand der Technik erreicht werden können.

In einem dritten Experiment wird das Driftverhalten der Homographien aus den drei Methoden untersucht. In diesem Versuch wurde im ersten Bild der jeweiligen Sequenz eine Markierung auf ein beliebiges Objekt gesetzt. Die korrekte Objektposition ist für jedes Bild der Bildsequenz bekannt. Die Position der Markierung wird von Bild zu Bild transformiert. Für die Transformation der Markierung werden die Homographien verwendet, die für die Sequenz mittels der vorgeschlagenen Methode (GPS) und der Methoden des Stands der Technik (SIFT) und (SURF) erzeugt wurden. In jedem Bild wurde die euklidische Pixeldistanz zwischen der jeweiligen transformierten Markierung und der originalen Objektposition bestimmt. Der Driftfehler wurde als Mittelwert der bestimmten euklidischen Distanzen berechnet. Zusätzlich wurden Standardabweichung, Median und Maximum der euklidischen Distanzen für

Tabelle 4.3: Bewertung des Driftfehlers für die drei Methoden zur Homographieberechnung.

Methode	Fehler	Median	Maximum
GPS	2.95 ± 1.68	2.71	13.46
SIFT	6.41 ± 3.25	5.92	20.81
SURF	10.80 ± 7.52	9.59	30.89

jede Methode bestimmt. Die Ergebnisse der Homographien sind in Tabelle 4.3 zusammengefasst.

Die Betrachtung des Driftfehlers zeigt, dass der mittlere Driftfehler für die präsentierte Methode deutlich geringer ist. Sowohl SIFT als auch SURF erzeugen größere Fehler in dieser Testreihe. Bei der Betrachtung mehrerer hoch kontrastiger Objekte waren die Methoden SIFT und SURF für jedes Objekt etwas genauer als die GPS-basierte Methodik. Im Fall einer Markierung beliebiger Objekte im Bild konnte die Stärke des GPS-basierten Ansatzes etwas deutlicher gezeigt werden. In allen Kategorien vom Mittelwert bis hin zum Median und Maximum des Pixelfehlers waren die Werte für den GPS-basierten Ansatz in Tabelle 4.3 geringer als die der Vergleichsmethoden.

Nachdem nun die Stabilität, sowie das Driftverhalten der errechneten Homographien beurteilt wurden, bleibt ein weiterer Vorteil der vorgestellten Methodik zu benennen. Sowohl SIFT als auch SURF bestimmen für alle Kandidaten von Punktkorrespondenzen einen Deskriptorvektor. Dieser Deskriptorvektor ist hochdimensional und die Berechnung aller Komponenten in diesem Vektor ist sehr aufwändig. Zusätzlich ist die Anzahl der Punktkorrespondenzen sehr hoch, wodurch ein großes überbestimmtes System zur Berechnung der Homographien gelöst werden muss. Für den vorgestellten Algorithmus mit GPS-basierten Punktprojektionen genügt es eine geringe Anzahl von GPS-Punkten in das Bild zu projizieren. Durch die Kenntnis der Projektionsparameter kann in jedes Bild der Sequenz ein Objekt an die korrekte Position eingefügt werden. Hierdurch werden keine hochdimensionalen Deskriptoren benötigt. Daraus ergibt sich ein geringerer Speicheraufwand für die vorgestellte Methode. Neben dem geringeren Speicheraufwand wird zusätzlich der Aufwand der Berechnung bewertet. Für die Bestimmung des Rechenaufwands wurde eine Simulation in Matlab R2013b mit einem Intel Core i7 Prozessor durchgeführt. Die Rechenzeiten wurden mit einer Sequenz mit 2300 Einzelbildern ermittelt und sind in Tabelle 4.4 zu sehen.

Tabelle 4.4: Bewertung der Rechenzeit für die Bestimmung von Homographien in einer 2300 Bilder umfassenden Sequenz.

Methode	Rechenzeit in Sekunden	Pro Bild
GPS	6.9	0.003
SIFT	5518.6	2.4
SURF	221.3	0.1

Die Zeiten in Tabelle 4.4 zeigen die von jeder Methode benötigte Zeit um alle Homographien der gesamten Sequenz zu berechnen. Insbesondere fällt die sehr hohe Rechenzeit für die SIFT Methode auf. Der SIFT Algorithmus benötigt sehr viel Zeit für die Bestimmung der Punktkorrespondenzen und benötigt deshalb die meiste Zeit für die Berechnung der Homographien. Obgleich die SURF Methode im Vergleich zur SIFT Methode deutlich weniger Zeit benötigt, ist sie dennoch deutlich langsamer als der GPS-basierte Ansatz. Allerdings ist der Zeitaufwand für die Bestimmung der Projektionsmatrizen hier nicht beachtet worden. Er ist dennoch vernachlässigbar im Vergleich zu den Rechenzeiten der SIFT und SURF Methodik.

Mit der GPS-basierten Projektion von Objekten in Kamerabildern konnte die Berechnung von Homographien für Bildsequenzen qualitativ verbessert und beschleunigt werden. Es konnte gezeigt werden, dass die erreichte Genauigkeit ähnlich gute Ergebnisse erzielt hat wie auch der Stand der Technik. Im Drift-Experiment konnte sogar ein deutlich besseres Ergebnis im Vergleich zum Stand der Technik erzielt werden.

4.2 Tracking von Punktzielen: Flugzeugtracking anhand von Multilaterationsmessungen

Das Tracking von Punktzielen lässt die Ausdehnung des beobachteten Objekts außer acht. Eine Modellierung eines bewegten Punktziels lässt ebenfalls außer acht, dass die Bewegung durch die Masse des Objekts beeinflusst wird. Für den Fall des Multilaterationstrackings wird das beobachtete Flugzeug als Punktobjekt modelliert. Diese Art der Modellierung erlaubt die Verwendung

eines einfachen Systemmodells. Etwaige Ungenauigkeiten bei der Modellierung können durch entsprechend modellierte Rauschkovarianzmatrizen charakterisiert werden. In dem vorliegenden Anwendungsfall wird von verteilten Bodenstationen ausgegangen, die dazu verwendet werden die Flugzeugposition zu bestimmen. In diesem Aufbau sendet das Flugzeug Signale aus, die zu verschiedenen Zeitpunkten von den Bodenstationen empfangen werden. Die unterschiedlichen Ankunftszeiten des Signals an den Basisstationen ist durch die große räumliche Verteilung der Basisstationen begründet. Die aktuelle Position des Flugzeugs ist unbekannt, daher ist auch der Sendezeitpunk des Signals unbekannt. Somit kann die genaue Sendezeit nicht direkt bestimmt werden. Es liegen nur die Empfangszeiten an den Basisstationen vor.

Ohne Beschränkung der Allgemeinheit wird davon ausgegangen, dass $n \in \mathbb{N}$ Basisstationen in allgemeiner Position zueinander vorliegen. Die Ankunftszeiten des Flugzeugsignals werden durch

$$t_k^i = \frac{\|\underline{s}_i - \underline{x}_k\|}{c} + t_k^0 \tag{4.16}$$

beschrieben, wobei der Index $i \in \{1, \dots, n\}$ die i-te Basistation bezeichnet. Der Vektor $\underline{s}_i$ beschreibt die Position des i-ten Sensors und $\underline{x}_k$ beschreibt die unbekannte Position des Flugzeugs zum Zeitpunkt k. Mit c wird die Lichtgeschwindigkeitskonstante im leeren Raum bezeichnet, es gilt $c = 299792.458\text{km/s}$. Der Ausdruck $\|\underline{s}_i - \underline{x}_k\|$ beschreibt den Abstand des Flugzeugs zur Basisstation, somit kann die Dauer zwischen der Signalaussendung und dem Empfangszeitpunkt bestimmt werden. Mit Hilfe des Sendezeitpunkts t_k^0 kann der Empfangszeitpunkt in der Basistation bestimmt werden. Die Gleichung enthält zwei Unbekannte: die gesuchte Position des Flugzeugs $\underline{x}_k$ und den Zeitpunkt der Aussendung des Signals t_k^0. Eine Möglichkeit die Anzahl der Unbekannten zu reduzieren ist die Verwendung von Differenzen der Ankunftszeit (engl. Time differences of Arrival (TDOA)) als Messungen. Für n Basisstationen resultiert die Verwendung von TDOAs als Messungen in $n-1$ einzelnen Messungen. Durch die Verwendung von TDOAs wird die Ankunftszeit eliminiert. Eine TDOA beschreibt anschaulich einen Hyperboloid im Raum. Um die

Position des Flugzeugs zu bestimmen, sind somit mindestens 3 TDOA Messungen notwendig. Insofern die Messungen keiner Störung unterliegen, kann die aktuelle Position durch den Schnitt von mindestens drei Hyperboloiden bestimmt werden. Ein Hyperboloid kann aus zwei Körpern bestehen, somit können mehrere Schnittpunkte vorliegen. Ein alternativer Ansatz besteht in der direkten Verwendung der Ankunftszeiten (4.16). Die Empfangszeit entspricht geometrisch einem Kegel. Somit kann die Position des Flugzeugs durch den Schnitt mehrerer Kegel bestimmt werden. Ein Nachteil der Verwendung der Ankunftszeiten ist der unbekannte Sendezeitpunkt t_k^0. Der unbekannte Sendezeitpunkt bedingt, dass die Höhe des Kegels unbekannt ist. Dennoch ist der Schnitt von Kegeln wohl definiert.

Die Verwendung der Ankunftszeiten als Messabbildung für einen Trackingalgorithmus erfordert eine Behandlung des unbekannten Sendezeitpunkts. Da die Position des Flugzeugs ermittelt werden soll, werden zur Vorbereitung die Ankunftszeiten in Pseudoabstandsmessungen transformiert. Zu diesem Zweck wird (4.16) auf beiden Seiten der Gleichung mit der Schallgeschwindigkeit c multipliziert. Somit kann der Abstand des Flugzeugs zum i-ten Sensor durch

$$y_k(i) = c \cdot t_k^0 = \|\underline{s}_i - \underline{x}_k\| + \underbrace{c \cdot t_k^0}_{r_k} \tag{4.17}$$

berechnet werden. Der unbekannte Sendezeitpunkt geht in den Offset $r_k = c \cdot t_k^0$ ein. Unter der Annahme, dass der Empfangszeitpunkt bekannt ist kann der Abstand des Flugzeugs zur i-ten Basisstation durch (4.17) bestimmt werden. Dabei ist die Flugzeugposition durch $\underline{x}_k$ gegeben. Für den Tracker werden die beiden Unbekannten $\underline{x}_k$ und r_k in den Systemzustand aufgenommen.

4.2.1 Modellierung des Systems und Zustandsschätzung

Der Systemzustand wird mit Hilfe der aktuellen Flugzeugposition und dem unbekannten Offset durch

$$\underline{\xi}_k = \begin{bmatrix} \underline{x}_k \\ r_k \end{bmatrix} \tag{4.18}$$

definiert. Eine Messung umfasst die Pseudoabstandsmessungen zwischen der aktuellen Flugzeugposition und allen vorhandenen Basisstationen. Zur vollständigen Definition des Messmodells müssen alle Unsicherheiten, die die Messungen beeinflussen können, in Erwägung gezogen werden. Die Positionen der Basisstationen gelten als unsicher, da GPS-Messgeräte für die Positionsbestimmung eingesetzt werden. Durch ungenaue Uhren und Sensordrift ergeben sich somit Fehler bei der Positionsbestimmung. Dieser Fehler wird durch einen additiven Rauschanteil modelliert. Durch

$$\underline{S}_i = \left(\hat{\underline{S}}_i + \underline{\rho}_k\right) \tag{4.19}$$

wird die unsichere Position der i-ten Basisstation angegeben, wobei für den additiven Rauschanteil $\underline{\rho}_k \sim \mathcal{N}\left(\underline{0}, \mathbf{C}_k^{\rho}\right)$ gilt. Die Kovarianzmatrix $\mathbf{C}_k^{\rho}$ bezieht sich auf die Unsicherheit bei der Positionsbestimmung der Basisstationen. Es wird angenommen, dass die Positionsunsicherheit der Basisstationen unkorreliert ist. Die Messunsicherheit ergibt sich einzeln zu $\sigma_{\rho_0}^2$. Die Varianz für jede einzelne Basisstation ergibt sich in Abhängigkeit der Anzahl an Basisstationen n und kann durch

$$\sigma_{\rho}^2 = \frac{1}{n^2} \cdot \sigma_{\rho_0}^2 \tag{4.20}$$

angegeben werden. Aufgrund der angenommen Unkorreliertheit ergibt sich eine diagonale Kovarianzmatrix $\mathbf{C}_k^{\rho} = \operatorname{diag}\left(\sigma_{\rho}^2, \ldots, \sigma_{\rho}^2\right)$.

Ein weiterer Unsicherheitsfaktor entsteht durch Störungen der Messungen. Die Messung der Empfangszeit wird ebenfalls durch einen Zeitmesser durchgeführt. Ein solcher Zeitmesser unterliegt einer Drift, ebenso wie ein GPS-Messgerät. Der zeitliche Anteil wird durch σ_t^2 angegeben. Das Signal unterliegt ebenfalls weiteren physikalischen Störeinflüssen, dazu gehören Diffraktion, Reflexion und Streuung. Diese Rauschanteile werden durch σ_s^2 definiert. Die beiden Rauschanteile des Signals werden als additiver Rauschterm modelliert. Daraus ergibt sich durch

$$\mathbf{C}_k^{\upsilon^a} = \operatorname{diag}\left(\{\sigma_s^2 + \sigma_t^2\}, \ldots, \{\sigma_s^2 + \sigma_t^2\}\right) \tag{4.21}$$

die Kovarianzmatrix für den additiven Rauschanteil der Signalmessungen. Das additive Rauschen wird als mittelwertfreies, normalverteiltes Rauschen modelliert. Somit gilt $\underline{\upsilon}_k^a \sim \mathcal{N}\left(\underline{0}, \mathbf{C}_k^{\upsilon^a}\right)$. Neben einem additiven Rauschanteil des Signals wird ein multiplikativer Rauschanteil

$$\underline{\upsilon}_k^m \sim \mathcal{N}\left(\underline{1}, \mathbf{C}_k^{\upsilon^m}\right) \tag{4.22}$$

definiert. $\underline{\upsilon}_k^m$ ist ein Zufallsvektor mit mittelwertbehafteter Normalverteilung. Die Definition der drei Rauschanteile ermöglicht die vollständige Angabe der Messfunktion, basierend auf n Basisstationen. Durch

$$\underline{y}_k = \underline{h}\left(\underline{\xi}_k, \underline{\upsilon}_k^m\right) + \underline{\upsilon}_k^a = \begin{bmatrix} \|\left(\hat{\underline{S}}_1 + \underline{\rho}_k\right) - \underline{x}_k\| \\ \vdots \\ \|\left(\hat{\underline{S}}_n + \underline{\rho}_k\right) - \underline{x}_k\| \end{bmatrix} \cdot \underline{\upsilon}_k^m + \underline{1} \cdot r_k + \underline{\upsilon}_k^a \tag{4.23}$$

wird das nichtlineare Messmodell definiert, in dem der Abstand der aktuellen, geschätzten Position $\underline{x}_k$ zu den n Basisstationen $\hat{\underline{S}}_i + \underline{\rho}_k$ berechnet wird. Der

multiplikative Rauschanteil wird auf die Abstände zu den Basisstationen angewandt, wobei der Offset r_k und ein additiver Rauschterm zu der Messung hinzuaddiert werden.

Das Systemmodell modelliert die physikalische Bewegung des beobachteten Objekts. Das vorliegende Objekt ist ein Passagierflugzeug, das sich vorrangig mit konstanter Geschwindigkeit fortbewegt. Im Fall von Kursänderungen werden in kurzen Passagen Kurven zur Kurskorrektur geflogen. Somit reicht ein einfaches Modell zur Beschreibung von Bewegungen mit konstanter Geschwindigkeit nicht aus. Aus diesem Grund wird ein Kombinationsmodell (engl. Interacting Multiple Model) (IMM) verwendet. IMM werden in Bar-Shalom et al. [Bar02] beschrieben. Im Fall des Multilaterationstrackings kommen zwei unterschiedliche Modelle für die Bewegung des Flugzeugs zum Einsatz: Ein konstantes Geschwindigkeitsmodell (engl. Constant-Velocity-Model) (CV) und ein konstantes Positionsmodell (engl. Constant-Position-Model) (CP).

Der Systemzustand wurde um den unbekannten Offset erweitert. Daher muss neben der Flugzeugbewegung ein Systemmodell generiert werden, dass die dynamische Entwicklung des Offsets beschreibt. Da keine Annahme über die Dynamik des Offsets getroffen werden kann, wird das konstante Positionsmodell (engl. Constant-Position-Model) (CP) verwendet. Im CP Modell geht man davon aus, dass der Offset konstant ist. Da dieses Modell keine genaue Beschreibung des tatsächlichen Verhaltens des Offsets ist, enthält das Modell einen additiven Rauschterm, der die Ungenauigkeit des Offsets beschreibt. Durch

$$r_{k+1} = r_k + w^r \tag{4.24}$$

ist das CP Modell für den unbekannten Entfernungsoffset beschrieben, wobei w^r ein mittelwertfreies, Gauß'sches Rauschen ist.

Für die Beschreibung der Flugzeugdynamik wird das CV Modell eingesetzt. Das CV Modell ist ein lineares Modell, in dem die aktuelle Position und die

aktuelle Geschwindigkeit des beobachteten Objekts in der Zustandsbeschreibung enthalten sein müssen. Die Flugzeugposition wird in einer Ebene beschrieben, da eine Schätzung der Höhe des Flugzeugs aufgrund der ebenen Verteilung der Basisstationen zu fehleranfällig ist. Die Höhe des Flugzeugs wird über Höhenmesser bestimmt, da diese Information sicherheitsrelevant ist. Somit können Position und Geschwindigkeit des Flugzeugs jeweils durch zwei Koordinaten angegeben werden. Der Anteil des Systemzustands zur Beschreibung des aktuellen Zustands des Flugzeugs ist durch

$$\underline{x}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T \tag{4.25}$$

definiert. Ausgehend von dem Aufbau des Systemzustands kann das CV Systemmodell durch

$$\mathbf{A}_x = \begin{bmatrix} \mathbf{I} & T \cdot \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{4.26}$$

angegeben werden, wobei $\mathbf{I}$ die Einheitsmatrix ist, $\mathbf{0}$ die Nullmatrix und T der diskrete Zeitschritt. Mit der Systemmatrix und dem Systemzustand kann die Systemgleichung definiert werden. Für das CV Modell ergibt sich eine lineare Systemfunktion durch

$$\underline{x}_{k+1} = \mathbf{A}_x \cdot \underline{x}_k + \underline{w}_k^x \,, \tag{4.27}$$

wobei $\underline{w}_k^x$ ein korreliertes Gauß'sches Rauschen mit Kovarianzmatrix

$$\mathbf{C}_x^w = \begin{bmatrix} \frac{T^3}{3} \cdot \mathbf{Q} & \frac{T^2}{2} \cdot \mathbf{Q} \\ \frac{T^2}{2} \cdot \mathbf{Q} & T \cdot \mathbf{Q} \end{bmatrix} \tag{4.28}$$

beschreibt. Die Matrix $\mathbf{Q} = \text{diag}\{\sigma_x^2, \sigma_y^2\}$ enthält die Positionsunsicherheiten auf der Hauptdiagonalen.

Im vorliegenden Multilaterationstracker werden simultan der unbekannte Offset und die Position des Flugzeugs geschätzt. In den vorangehenden Abschnitten wurden die Systemmodelle für den Offset und die Flugzeugbewegung unabhängig voneinander definiert. Für eine simultane Schätzung müssen beide Modelle miteinander kombiniert werden. Dies kann durch Erweiterung des Systemzustands und zusammengesetzte Systemkovarianzmatrix und Systemmatrix erreicht werden. Die zusammengesetzte Systemmatrix für das Systemmodell kann durch reguläre Erweiterung der Systemmatrix definiert werden. Die kombinierte Systemmatrix kann durch

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_x & \underline{0} \\ \underline{0}^T & 1 \end{bmatrix} \tag{4.29}$$

angegeben werden. Die Matrix $\mathbf{A}_x$ ist die Systemmatrix des CV Modells. Für die Erweiterung mit dem Offsetsystemmodell wird die Matrix um eine Zeile und eine Spalte vergrößert. Die Hauptdiagonale enthält die Identitätsabbildung für den Offset. Da der Offset unabhängig von der Position und Geschwindigkeit ist, werden die restlichen Felder mit Nullvektoren aufgefüllt. Äquivalent wird die Kovarianzmatrix für das Systemrauschen erweitert. Das Ergebnis der Erweiterung ist durch

$$\mathbf{C}_k^w = \begin{bmatrix} \mathbf{C}_x^w & \underline{0} \\ \underline{0}^T & \sigma_r^2 \end{bmatrix} \tag{4.30}$$

gegeben. Der erweiterte Systemzustand ist durch

$$\underline{\xi}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k, r_k]^T \tag{4.31}$$

gegeben und entsteht durch Erweiterung um den unbekannten Offset r_k. Durch die Modellierung des Systemzustands, der Systemabbildung und der Messabbildung kann die Definition des Schätzers für das Multilaterationstracking erfolgen. Die Modellkomponenten können in die Gleichungen für das stochastische Filter eingesetzt werden. Durch die Definition einer nichtlinearen Messabbildung muss ein nichtlineares Filter als Schätzer gewählt werden.

4.2.2 Schätzerdesign

Da ein nichtlineares Filter eingesetzt werden muss, bietet sich die Verwendung eines Sample-basierten Verfahrens an. Zu den Sample-basierten Verfahren gehören das Particle-Filter [Aru02], sowie das Unscented Kalman Filter (UKF) [Jul99, Jul00]. Die beiden Verfahren unterscheiden sich in der Art des verwendeten Samplings. Im Particle-Filter wird ein zufallsbasiertes Sampling durchgeführt, wohingegen das UKF ein deterministisches Sampling verwendet. Durch das zufallsbasierte Sampling wird eine Vielzahl an Partikeln für das Particle-Filter benötigt. Determinischte Sampling Verfahren greifen für das Sampling auf das erste und zweite Zentrale Moment der A-Priori-Zustandsverteilung zurück. Filter, die für das Sampling die A-Priori-Zustandsverteilung verwenden, werden unter der Bezeichnung lineare Regressions-Kalman-Filter zusammengefasst. Ein weiterer Vertreter dieser Klasse ist das durch Huber et al. in [Hub08] beschriebene Gaußfilter. Das Gaußfilter und das UKF führen beide ein deterministisches Sampling der Zustandsverteilung aus. In Huber et al. [Hub08] konnte gezeigt werden, dass mit dem veränderten Samplings des Gaußfilters eine verbesserte Schätzung der posterioren Verteilung durchgeführt werden kann. Aus diesem Grund wird für das Multilaterationstracking auf den Gaußfilter zurückgegriffen.

Die Systemmatrix (4.29) ist linear. Aus diesem Grund muss kein Sampling der prioren Verteilung durchgeführt werden. Somit kann der Prädiktionsschritt des Standard Kalman Filters verwendet werden. Durch

$$\underline{\xi}_{k+1}^{p} = \mathbf{A} \cdot \underline{\xi}_{k}^{e} \tag{4.32}$$

$$\mathbf{C}_{k+1}^{p} = \mathbf{A} \cdot \mathbf{C}_{k}^{e} \mathbf{A}^{T} + \mathbf{C}_{k}^{w} \tag{4.33}$$

werden der Systemzustand $\underline{\xi}_{k}$ und die Systemkovarianzmatrix $\mathbf{C}_k$ durch Anwendung von (4.29) und (4.30) prädiziert.

Die Durchführung des Filterschritts erfordert die Berechnung des Kalman Gains $\mathbf{K}_k$, des geschätzten Systemzustands $\underline{\xi}_{k}^{e}$ und der geschätzten Systemkovarianzmatrix $\mathbf{C}_{k}^{e}$. Dazu müssen folgende Gleichungen gelöst werden:

$$\mathbf{K}_k = \mathbf{C}_{k}^{\xi,y} \cdot \left(\mathbf{C}_{k}^{y}\right)^{-1} \tag{4.34}$$

$$\underline{\xi}_{k}^{e} = \underline{\xi}_{k}^{p} + \mathbf{K}_k \cdot \left(\underline{\hat{y}}_{k} - \underline{\mu}_{k}^{y}\right) \tag{4.35}$$

$$\mathbf{C}_{k}^{e} = \mathbf{C}_{k}^{p} - \mathbf{K}_k \cdot \mathbf{C}_{k}^{y} \cdot \mathbf{K}_{k}^{T} \tag{4.36}$$

Die Größen $\underline{\mu}_{k}^{y}$, $\mathbf{C}_{k}^{y}$ und $\mathbf{C}_{k}^{\xi,y}$ können nur unter Verwendung des nichtlinearen Messmodells (4.23) bestimmt werden. Wegen der Nichtlinearität können die Ausdrücke nicht analytisch ausgewertet werden. Daher ist eine Approximation des prädizierten Zustands notwendig. Zur Approximation des prädizierten Zustands wird das deterministische Sampling des Gaußfilters eingesetzt. Bevor das Sampling angewendet wird, erfolgt eine Zerlegung des Systemzustands. Dabei wird der Systemzustand in einen Anteil zerlegt, der in der Messfunktion verwendet wird, und in einen Anteil, der nicht in der Messfunktion verwendet wird. Die Zerlegung ist durch

$$\underline{x}_{k}^{a} = \left[x_k, y_k, z_k, r_k\right]^{T} ,$$

$$\underline{x}_{k}^{b} = \left[\dot{x}_k, \dot{y}_k, \dot{z}_k\right]^{T}$$

gegeben. Die Geschwindigkeit geht zwar indirekt durch das Systemmodell in die aktuelle Position des Flugzeuges ein, ist aber dennoch kein direkter Bestandteil des Messmodells. Aus diesem Grund wird die Geschwindigkeit in einem separaten Teil untergebracht. Die Zerlegung des Systemzustands ist gültig, weil die prädizierte Dichte entsprechend der Formel von Bayes durch

$$f\left(\underline{\xi}_k\right) = f\left(\underline{x}_k^b \mid \underline{x}_k^a\right) \cdot f\left(\underline{x}_k^a\right) \tag{4.37}$$

faktorisiert werden kann. Es wird angenommen, dass $\underline{\xi}_k \sim \mathcal{N}\left(\underline{\xi}_k, \mathbf{C}_k\right)$ gilt. Unter dieser Annahme, kann die Kovarianzmatrix entsprechend der Dichte zerlegt werden. In

$$\mathbf{C}_k = \begin{bmatrix} \mathbf{C}_k^a & \mathbf{C}_k^{a,b} \\ \mathbf{C}_k^{b,a} & \mathbf{C}_k^b \end{bmatrix} \tag{4.38}$$

ist die Zerlegung dargestellt. Da die prädizierte Dichte eine Gaußdichte ist, können entsprechend der Formel von Bayes alle Teildichten als Gaußdichten angegeben werden. Die Teildichte für $f\left(\underline{x}_k^a\right)$ kann durch

$$\mathcal{N}\left(\underline{x}_k^a - \underline{\mu}_k^a, \mathbf{C}_k^a\right) \tag{4.39}$$

angegeben werden. Beutler et al. haben in [Beu09] gezeigt, dass $f\left(\underline{x}_k^b \mid \underline{x}_k^a\right)$ ebenfalls eine Gaußdichte ist. Die vollständige Dichte kann durch

$$\mathcal{N}\left(\underline{x}_k^b - \left[\underline{\mu}_k^b + \mathbf{C}_k^{b,a} \cdot \left(\mathbf{C}_k^a\right)^{-1} \cdot \left(\underline{x}_k^a - \underline{\mu}_k^b\right)\right], C_k^b - \mathbf{C}_k^{b,a} \cdot \left(\mathbf{C}_k^a\right)^{-1} \cdot \mathbf{C}_k^{a,b}\right) \tag{4.40}$$

bestimmt werden. Die Teildichten des zerlegten Systemzustands werden benötigt, um die Kovarianzmatrizen $\mathbf{C}_k^{\xi,y}$ und $\mathbf{C}_k^{y}$ zu approximieren. Da die Messungen ausschließlich von dem Teilzustand $\underline{x}_k^a$ abhängen, genügt die Approximation von $f\left(\underline{x}_k^a\right)$ durch eine Dirac-Mischdichte. Huber et al. haben in [Hub08] beschrieben, das eine Gaußdichte durch

$$f\left(\underline{x}_k^a\right) \approx \frac{1}{L}\sum_{i=1}^{L} \delta\left(\underline{x}_k^a - \underline{\mu}_i^a\right) \tag{4.41}$$

mit einer Dirac-Mischdichte approximiert werden kann. Die $\underline{\mu}_i^a$ entsprechen den Positionen der einzelnen Dirac-Komponenten der Mischdichte. Die Dirac-Komponenten repräsentieren die Samples der zu approximierenden Dichte. Für die Bestimmung der Positionen der Samplepunkte wird auf Huber et al. [Hub08] verwiesen.

Das Messmodell in (4.23) wird dazu verwendet die Samplepunkte $\underline{\mu}_i^a$ in den Messraum zu transformieren. Die transformierten Samplepunkte werden durch

$$\underline{\mu}_i^y = \underline{h}\left(\underline{\mu}_i^a\right) \tag{4.42}$$

repräsentiert. Die transformierten Samples $\underline{\mu}_i^y$ werden als Dirac-Komponenten einer transformierten Dirac-Mischdichte angenommen. Somit ist durch die Dirac-Mischdichte eine Approximation der Messdichte $f\left(\underline{y}_k\right)$ gegeben. Um den Mittelwert und die Kovarianzmatrix der Messdichte rechnerisch zu approximieren, werden nach Huber et al. in [Hub08] die Stichprobenmittelwert-

und die Stichprobenkovarianzfunktion verwendet. Mit Hilfe der Stichprobenmittelwertfunktion ergibt sich durch

$$\underline{\mu}_k^y = \frac{1}{L} \sum_{i=1}^{L} \underline{\mu}_i^y \tag{4.43}$$

der Mittelwertvektor der Messdichte, wobei L die Anzahl der Dirac-Komponenten $\underline{\mu}_i^y$ ist. Unter Anwendung der Stichprobenkovarianzfunktion wird durch

$$\mathbf{C}_k^y = \frac{1}{d} \sum_{i=1}^{L} \left(\underline{\mu}_i^y - \underline{\mu}_k^y \right) \cdot \left(\underline{\mu}_i^y - \underline{\mu}_k^y \right)^T + \mathbf{C}_k^v \tag{4.44}$$

die Kovarianzmatrix der Messdichte berechnet, wobei d die Dimension des Systemzustands ist. Da die Schätzung der Messkovarianzmatrix erwartungstreu ist, kann hier der Faktor $\frac{1}{d}$ verwendet werden. Nachdem die Messkovarianzmatrix $\mathbf{C}_k^y$ durch Anwendung der Stichprobenkovarianzfunktion berechnet wurde, kann jetzt unter Verwendung der approximierten Dichten die Kreuzkovarianzmatrix bestimmt werden. Unter Verwendung der Zustandsdichte $f\left(\underline{\xi}_k\right)$ ergibt sich durch

$$\mathbf{C}_k^{\xi,y} = \iint \left(\underline{\xi}_k - \underline{\mu}_k^{\xi} \right) \cdot \left(\underline{y}_k - \underline{\mu}_k^y \right)^T \cdot f\left(\underline{y}_k | \underline{\xi}_k \right) f\left(\underline{\xi}_k \right) \mathrm{d}\underline{\xi}_k \, \mathrm{d}\underline{y}_k \tag{4.45}$$

die Kreuzkovarianzmatrix. Um die approximierten Dichten verwenden zu können, wird die Kreuzkovarianzmatrix zerlegt. Die Zerlegung erfolgt in

einen beobachtbaren und nichtbeobachtbaren Teil. Die resultierende Zerlegung wird durch

$$\mathbf{C}_k^{\xi,y} = \begin{bmatrix} \mathbf{C}_k^{a,y} \\ \mathbf{C}_k^{b,y} \end{bmatrix} \tag{4.46}$$

angegeben. Der beobachtbare Anteil $\mathbf{C}_k^{a,y}$ kann unter Anwendung der Stichprobenkovarianzfunktion mit den Samples der Messdichte und der A-Priori-Dichte berechnet werden. Somit kann mit

$$\mathbf{C}_k^{a,y} = \frac{1}{d} \cdot \sum_{i=1}^{L} \left(\underline{\mu}_i^a - \underline{\mu}_x^a \right) \cdot \left(\underline{\mu}_i^y - \underline{\mu}_k^y \right)^T \tag{4.47}$$

die Kreuzkovarianzmatrix des beobachtbaren Anteils des Systemzustands berechnet werden. Die Bestimmung des nichtbeobachtbaren Anteils der Kreuzkovarianzmatrix $\mathbf{C}_k^{b,y}$ kann analytisch erfolgen. Hierzu muss

$$\mathbf{C}_k^{b,y} = \iint \underline{x}_k^b \cdot \underline{h}\left(\underline{x}_k^a\right)^T \cdot f\left(\underline{x}_k^b | \underline{x}_k^a\right) f\left(\underline{x}_k^a\right) \mathrm{d}\underline{x}_k^a \mathrm{d}\underline{x}_k^b - \underline{\mu}_k^b \cdot \left(\underline{\mu}_k^y\right)^T \tag{4.48}$$

ausgewertet werden. Aufgrund der Unabhängigkeit der Variablen $\underline{x}_k^b$ von den Messungen kann diese Variable durch Marginalisierung eliminiert werden. Durch geschicktes Umsortieren der Integrale und der analytischen Berechnung der Integrale kann der Ausdruck für die Berechnung von $\mathbf{C}_k^{b,y}$ vereinfacht werden. Die resultierende Berechnungsvorschrift ist durch

$$\begin{aligned} \mathbf{C}_k^{b,y} = {} & \underline{\mu}_k^b \cdot \left(\underline{\mu}_k^y\right)^T + \mathbf{C}_k^{b,a} \cdot \left(\mathbf{C}_k^a\right)^{-1} \cdot \left[\mathbf{C}_k^{a,y} + \underline{\mu}_k^a \cdot \left(\underline{\mu}_k^y\right)^T \right] \\ & - \left[\mathbf{C}_k^{b,a} \cdot \left(\mathbf{C}_k^a\right)^{-1} \cdot \underline{\mu}_k^a \cdot \left(\underline{\mu}_k^y\right)^T \right] - \underline{\mu}_k^b \cdot \left(\underline{\mu}_k^y\right)^T \end{aligned} \tag{4.49}$$

gegeben. Durch Berechnungen kann (4.49) in

$$\mathbf{C}_k^{b,y} = \mathbf{C}_k^{b,a} \cdot \left(\mathbf{C}_k^{a}\right)^{-1} \cdot \mathbf{C}_k^{a,y} \tag{4.50}$$

vereinfacht und umgeformt werden. Durch die Bestimmung der beobachtbaren und nichtbeobachtbaren Anteile der Kreuzkovarianzmatrix ist (4.46) vollständig bestimmt. Somit können (4.34), (4.35) und (4.36) bestimmt werden. Durch eine geschickte Zerlegung des Systemzustands konnte der Anteil der zu approximierenden Zustandsdichte signifikant reduziert werden. Ebenso konnte der Berechnungsaufwand der Kreuzkovarianzmatrix durch analytische Berechnungen optimiert werden. In Simulationen wird die Funktionalität des vorgeschlagenen Modells demonstriert.

4.2.3 Simulationsergebnisse

In drei unterschiedlichen Simulationen wird die Wirksamkeit des vorgestellten Multilaterationstrackers im Vergleich zum Stand der Technik bewertet. Es wird grundsätzlich zwischen analytischen Methoden und stochastischen Trackingverfahren unterschieden. Die analytischen Methoden sind die Spherical Interpolation Methode (SI), die Abel et al. in [Abe87] beschrieben haben und die hyperbolische Positionierungsmethode (CH), die in [Cha94] von Chan et al. publiziert wurde. Als weiteres stochastisches Trackingverfahren kommt ein Unscented Kalman Filter (UKF) basierend auf TDOA Messungen zum Einsatz.

Es wurden sieben Sensoren modelliert, die in diskreten Zeitabständen Signale von einem Flugzeug empfangen. Tabelle 4.5 fasst die Positionen der Sensoren zusammen.

Die Trajektorie des Flugzeugs wird einmalig festgelegt. Abbildung 4.3 zeigt die planare Projektion der Flugzeugtrajektorie. In den Simulationen wird die Schätzung der Flughöhe vernachlässigt. Die Sensorpositionen in Tabelle 4.5 weisen im Vergleich zum Flugzeug nahezu identische Höhen auf, sodass eine akkurate Schätzung der Höhe unmöglich ist.

Tabelle 4.5: Sensorpositionen für die Simulation des Multilaterationstrackings.

Sensor	x/km	y/km	z/km
A	10	10	0.150
B	30	120	0.360
C	100	35	0.220
D	175	110	0.060
E	200	75	0.140
F	5	80	0.420
G	170	10	0.270

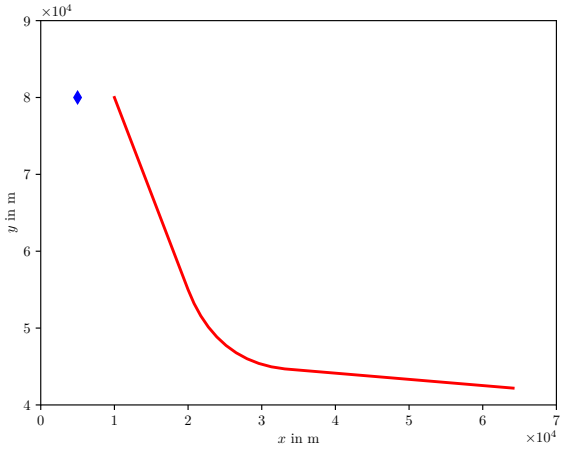


Abbildung 4.3: Planare Projektion der simulierten Flugzeugtrajektorie.

Tabelle 4.6: Resultierende RMSE Werte inklusive der Standardabweichungen in m für den Multilaterationstracker (GF), den TDOA-basierten UKF (UKF), die Spherical Interpolation Methode (SI) und die hyperbolische Lokalisierung (CH) aus der Simulation mit variierenden Rauschstärken.

Methode	0.3 m	3 m	7.5 m	15 m
GF	0.35 ± 0.24	2.65 ± 1.78	5.64 ± 3.69	10.08 ± 6.21
UKF	0.35 ± 0.26	2.89 ± 2.07	6.51 ± 4.68	11.98 ± 8.43
SI	0.70 ± 0.47	7.04 ± 4.73	17.60 ± 11.83	35.20 ± 23.65
CH	0.43 ± 0.32	5.87 ± 11.54	22.69 ± 43.62	35.75 ± 48.91

Die Trajektorie umfasst 250 Zeitschritte und wird für alle drei Simulationen verwendet. Die drei Simulationen behandeln unterschiedliche Szenarien. Das erste Szenario variiert die Standardabweichungen des Sensorrauschens für alle Sensoren. Das vorgestellte Trackingverfahren und die Vergleichsmethoden entsprechend dem Stand der Technik werden unter dem Einfluss der unterschiedlichen Rauschstärken untersucht und bewertet. In der zweiten Simulation wird die Anzahl der Sensoren verändert. Dabei liegt der Fokus darauf, wie sich die Positionsgenauigkeit der verwendeted Methoden unter verschiedenen Sensoranzahlen verändert. In der dritten Simulation werden Sensorausfälle untersucht. Dabei werden in zwei Durchläufen 10 Prozent und 20 Prozent der Sensoren zufällig ausgewählt, die keine Signale empfangen. Dieser Umstand verursacht Unterraummessungen. Da nur das vorgestellte Trackingverfahren und der Unscented Kalman Filter Unterraummessungen verarbeiten können, werden hier nur diese Trackingverfahren miteinander verglichen.

Für die erste Simulation wurden für verschiedene Rauschstärken jeweils 1000 Monte-Carlo-Simulationen mit der oben beschriebenen Trajektorie durchgeführt. Zur Beurteilung des resultierenden Tracking-Fehlers wurden die RMSE-Werte über diesen 1000 Durchläufen gemittelt und die Standardabweichung für alle vier Methoden berechnet. Die Standardabweichungen für die verschieden Rauschstärken wurden in vier Schritten erhöht. Nacheinander wurden mit Standardabweichungen von 0.3 m, 3 m, 7.5 m und 15 m jeweils 1000 Monte-Carlo-Simulationen durchgeführt. Tabelle 4.6 fasst die Ergebnisse des Tests zusammen.

Abbildung 4.4 zeigt die relativen Fehlerniveaus der einzelnen Methoden bezogen auf die verwendeten Standardabweichungen. Man erkennt, dass die beiden Tracking-Verfahren GF und UKF für jedes Fehlerniveau einen mittleren

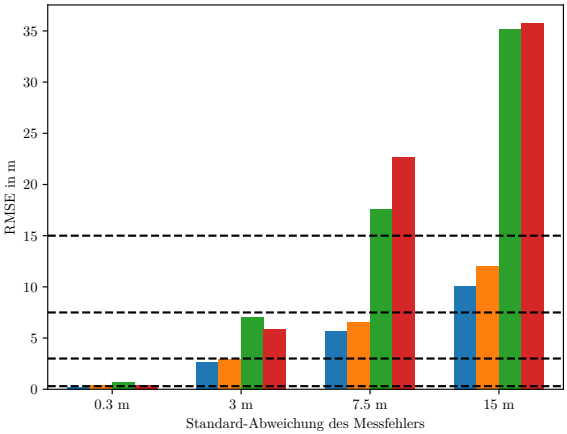


Abbildung 4.4: Balkengrafik mit den RMSE Mittelwerten der verwendeten Trackingmethoden.

RMSE unterhalb der Standardabweichung des Messrauschens aufweisen. Bei den beiden analytischen Vergleichsmethoden steigt der erreichte Trackingfehler bezogen auf das Fehlerniveau sehr stark an.

Für die zweite Simulation wurde ein festes Fehlerniveau mit einer Standardabweichung von 7.5 m für das Messrauschen festgelegt. Ziel dieser Simulation war die Beurteilung der Stabilität des Trackingergebnisses bezogen auf unterschiedliche Anzahlen von Sensoren. Es wurde zwischen 3, 4, 5, 6, und 7 Sensoren variiert. Für jede Anzahl von Sensoren wurden erneut 1000 Monte-Carlo-Simulationen durchgeführt, um durchschnittliche RMSE Werte, sowie zugehörige Standardabweichungen zu ermitteln. Die so bestimmten RMSE Mittelwerte und Standardabweichungen sind in Tabelle 4.7 aufgeführt.

Abbildung 4.5 stellt die mittleren RMSE Werte für die zweite Simulation in Abhängigkeit der Sensoranzahlen und der verwendeten Methoden dar. Es wird eine logarithmische Skala für die RMSE Werte verwendet, damit verdeutlicht werden kann, wie viele Größenordnungen gerade in den Tests mit geringer Sensoranzahl zwischen den analytischen Methoden und den Trackingverfahren liegen. Dieser Test hat gezeigt, dass die Trackingverfahren GF und UKF robuste Schätzungen vorweisen können, auch wenn nur eine geringe Anzahl

Tabelle 4.7: Simulationsergebnis der Variation der Sensorenanzahl. RMSE-Werte inklusive der Standardabweichungen in m werden angegeben für jede der verwendeten Methoden.

Methode	#3	#4	#5
GF	9.4 ± 6.36	7.31 ± 4.27	5.91 ± 3.33
UKF	19.2 ± 17.1	10.5 ± 7.91	8.59 ± 6.33
SI	$1.68x10^5 \pm 8.24x10^5$	$1.02x10^5 \pm 4.42x10^5$	$223.5 \pm 1.41x10^3$
CH	$4.77x10^5 \pm 4.28x10^6$	$4.74x10^5 \pm 1.17x10^5$	$6.48x10^3 \pm 2.23x10^4$
Methode	#6	#7	
GF	4.95 ± 2.71	4.64 ± 2.62	
UKF	5.96 ± 3.77	5.44 ± 3.22	
SI	22.8 ± 28.3	10.6 ± 6.85	
CH	$854.7 + 4.12x10^3$	$14.9 + 38.9$	

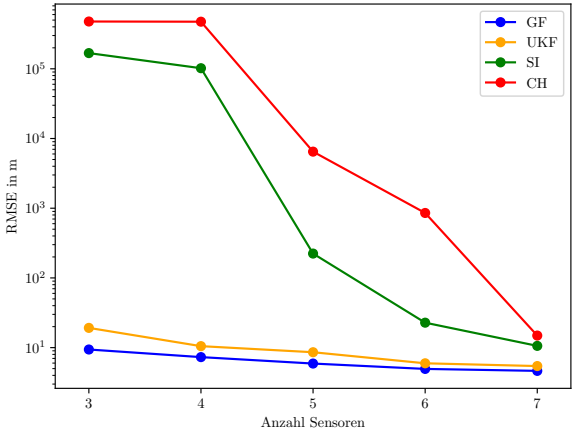


Abbildung 4.5: Vergleich der Fehlerverläufe mit logarithmischer Skala zur Verdeutlichung der Stabilität der verwendeten Methoden.

von Sensorstationen vorliegt. Die etwas höheren Trackingfehler des UKF mit TDOA Messungen können dadurch begründet werden, dass hier aufgrund des Abziehens der Ankunftszeiten $n-1$ Messungen vorliegen, wobei n die Anzahl der Sensoren ist.

Tabelle 4.8: Simulation des Sensorausfalls (in %) für die beiden filterbasierten Methoden. Evaluiert werden die RMSE-Werte inklusive der Standardabweichungen.

Methode	10%	20%
GF	4.35 ± 2.08	4.76 ± 2.06
UKF	5.44 ± 2.61	6.21 ± 2.60

Die dritte Simulation bewertet die Trackingverfahren GF und UKF bezüglich ihrer Robustheit bei Sensorausfall. Für diese Simulation wird von sieben Sensoren mit einer festen Standardabweichung von 7.5 m für das Messrauschen ausgegangen. Es werden zwei Monte-Carlo Durchläufe mit jeweils 1000 Wiederholungen durchgeführt. Im ersten Durchlauf wird ein Ausfall von 10% der Sensoren simuliert. Im zweiten Durchlauf beläuft sich der Ausfall auf 20% der Sensoren. Durch diesen Ausfall entstehen sogenannte Unterraum-Messungen. Die analytischen Methoden SI und CH besitzen keine Möglichkeit diese Unterraum-Messungen zu verarbeiten und werden aus diesem Grund für diese Simulation ausgeschlossen. Die Ergebnisse werden wie bereits in den vorherigen beiden Simulationen durch RMSE Mittelwerte und Standardabweichungen quantisiert. Tabelle 4.8 fasst die Ergebnisse zusammen.

Vergleicht man die Ergebnisse in Tabelle 4.8 miteinander, zeigt das vorgeschlagene Verfahren GF eine erhöhte Robustheit gegenüber dem UKF mit TDOA Messungen. Zur Verdeutlichung der Unterschiede sind in Abbildung 4.6 die Ergebnisse als Balkendiagramm aufgetragen.

Abbildung 4.6 verdeutlicht nochmals die Überlegenheit des vorgeschlagenen Modells gegenüber einem UKF mit TDOA Messungen. Auch bei vorhandenen Subraummessungen verringert sich der Fehler des vorgeschlagenen Trackers GF nur geringfügig im Vergleich zu einem Versuch in dem alle Sensoren funktionieren. Somit kann mit der vorgeschlagenen Methode GF ein robustes und verlässliches Trackingverfahren präsentiert werden.

4.2.4 Diskussion der Ergebnisse

Nach dem Stand der Technik werden zumeist TDOA Messungen verwendet, da somit der unbekannte Sendezeitpunkt eliminiert werden kann. Die hier

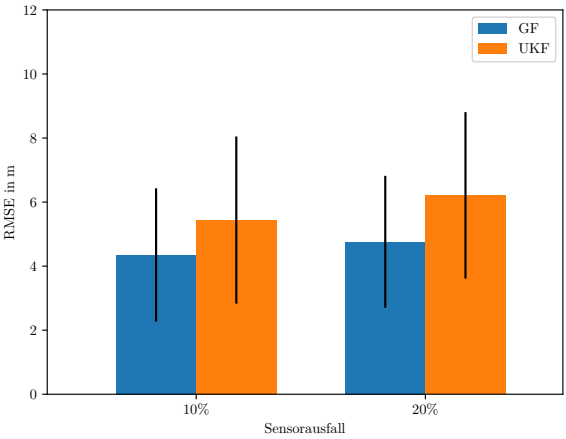


Abbildung 4.6: Balkendiagramm mit den mittleren RMSE-Werten bei unterschiedlichen Sensorausfallszenarien für GF und UKF

vorgestellte Methode ermöglicht eine direkte Verwendung von Empfangszeiten und zeigt zugleich, das eine simultane Schätzung des Sendezeitpunkts als Entfernungsoffset zu den Basisstationen eine robuste Schätzung ermöglicht. Zugleich kann der geometrische Hintergrund der Flugzeugpositionsbestimmung durch diese Methode vereinfacht werden. Die Positionsbestimmung durch TDOA Messungen wird durch den Schnitt von Hyperboloiden repräsentiert. Diese Schnittberechnung kann analytisch durchgeführt werden, besitzt allerdings eine hohe Komplexität. Durch die Verwendung der direkten Empfangszeiten konnte die Positionsbestimmung zu einem Schnitt von Kegeln vereinfacht werden.

Die Verwendung eines stochastischen Filters erlaubt die Verwendung von linearen Bewegungsmodellen. Für den Filterschritt wurde eine Zerlegung des Systemzustands vorgestellt, in einen beobachtbaren und einen indirekt beobachtbaren Anteil. Es wurde herausgestellt, dass nur der beobachtbare Teil des Systemzustands in die nichtlinearen Messfunktionen eingeht. Somit musste nur ein kleiner Teil des Systemzustands durch Sampling repräsentiert werden. Für das Sampling und die Filterung wurde das Gaußfilter von Huber et

al. [Hub08, Beu09] eingesetzt. Die Dekomposition hatte den zusätzlichen Vorteil, dass so die Berechnungskomplexität für das Sampling deutlich reduziert werden konnte. Aufgrund der Dekomposition musste auch die Kreuzkovarianzmatrix zerlegt, berechnet und anschließend zusammengesetzt werden.

In den vorgestellten Simulationen konnte gezeigt werden, dass das Verfahren mit unterschiedlichen Rauschstärken umgehen kann. Ebenso konnte gezeigt werden, dass das Verfahren eine erhöhte Robustheit bei verschiedenen Sensoranzahlen besitzt. Außerdem ist das Verfahren robust und zuverlässig im Fall von Sensorausfällen.

Zusammenfassend stellt die Möglichkeit der simultanen Schätzung des unbekannten Offsets und der gesuchten Flugzeugposition eine für das Flugzeugtracking gut geeignete Methode dar. In der Simulationsumgebung stellt das Verfahren unter unterschiedlichen Störungseinflüssen eine gute und zuverlässige Schätzung der aktuellen Flugzeugposition bereit.

4.3 Tracking eines ausgedehnten Objekts: die menschliche Iris

Das menschliche Auge ist ein guter Indikator, wenn es darum geht herauszufinden wohin sich die Aufmerksamkeit eines Menschen richtet. Das Auge eines Menschen setzt sich aus mehreren sichtbaren Bereichen zusammen: der weißen Lederhaut (lat. Sclera), der Regenbogenhaut (lat. Iris), sowie der Pupille. Alle Teile des menschlichen Auges sind in Abbildung 4.7 dargestellt.

Die Abbildung zeigt, dass die Unterscheidung zwischen der Sclera und der Iris aufgrund des Helligkeitsunterschieds einfach zu bewerkstelligen ist. Etwas schwieriger ist die Unterscheidung zwischen Pupille und der Iris. Die Pupille verändert ihre Größe, je nachdem auf was das Auge fokussiert oder welche Helligkeit vorherrscht. Die Iris verändert ihre äußere Form nicht und behält ihren Durchmesser bei. Der innere Durchmesser wird durch die Pupille bestimmt und nimmt ab und zu, je nach Dilatation der Pupille. Der Mittelpunkt

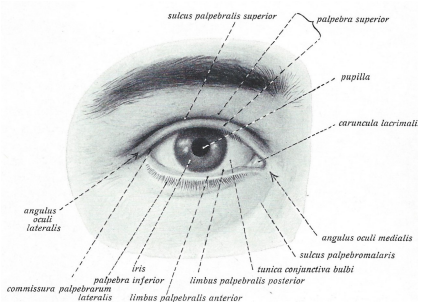


Abbildung 4.7: Abbildung des Sinnesorgans Auge aus Sobotta et al. [Sob62]

der Iris entspricht auch gleichzeitig dem Mittelpunkt der Pupille. Die Blickrichtung eines Menschen kann somit anhand der Position dieses Mittelpunkts bestimmt werden. Diesen Umstand machen sich gängige Eye-Tracker Systeme zunutze. Mit Infrarotreflexen auf der Iris können Eye-Tracker Systeme mit vorheriger Kalibrierung den Blickpunkt auf einem Monitor bestimmen. Damit ein Eye-Tracker die Richtung bestimmen kann, ist eine vorherige Detektion des Mittelpunkts und die Verfolgung des Mittelpunkts zu jeder Zeit der Beobachtung notwendig. Eine Möglichkeit den Mittelpunkt zu verfolgen besteht durch das Tracking der Iris.

Um aus dem Tracking heraus den Mittelpunkt der Pupille berechnen zu können, genügt ein Tracking als Punktziel nicht aus. Somit muss eine passende Repräsentation als ausgedehntes Objekt gefunden werden, mit der das Tracking durchgeführt wird. Ein geeignetes Formmodell für die Iris ist ein Kreis. Bei der Detektion der Iris muss also darauf geachtet werden, dass die Parameter des Kreises ausgehend von der Messung geschätzt werden können. Um das ausgedehnte Objekt dann in einem Trackingalgorithmus verwenden zu können, muss eine Parameterform für den Kreis gefunden werden. Im Folgenden wird zunächst die Detektion des Auges beschrieben, dann die Segmentierung des Rands der Iris. Anschließend wird beschrieben, wie aus den Randmessungen die Parameter des Iris-Modells geschätzt werden. Abschließend wird der Trackingalgorithmus angepasst, damit mit den Modellparametern das Tracking der Iris durchgeführt werden kann.

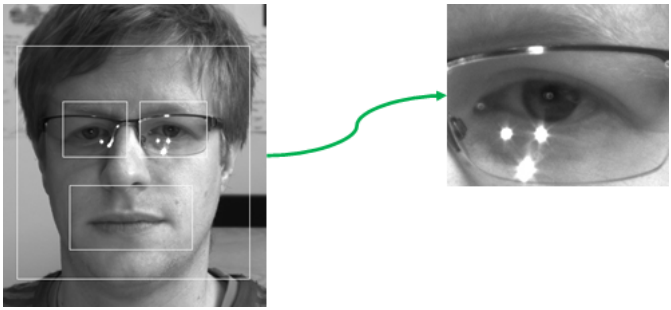


Abbildung 4.8: Extraktion der Augen unter Verwendung des Kaskaden-Klassifikators der OpenCV-Bibliothek.

4.3.1 Detektion und Segmentierung

Ausgehend von einer Aufnahme des gesamten Gesichts wird mit Hilfe des Cascade-Classifiers der OpenCV-Bibliothek detektiert. Dazu werden verschiedene, vorher trainierte Kaskaden verwendet. Die OpenCV-Bibliothek bietet dazu Kaskaden zur Detektion des Gesichts, sowie der Augen an. Der Cascade-Classifier bietet eine sehr schnelle Implementierung an und liefert Ergebnisse mit einer akzeptablen Genauigkeit. Um eine verbesserte Genauigkeit zu erhalten, sollte man selbst Kaskaden für den Klassifikator trainieren. Mit den Kaskaden von OpenCV ergeben sich Rechtecke, die den Bereich des gesuchten Objekts umranden. Das Ergebnis einer solchen Detektion für das Gesicht, beide Augen und den Mund, sowie der vergrößerte Bereich des linken Auges sind in Abbildung 4.8 zu sehen.

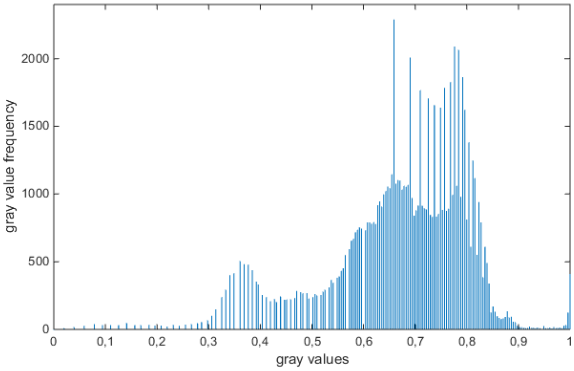
Ein extrahiertes Augenbild, wie es in Abbildung 4.8 dargestellt ist, ist der Ausgangspunkt für die Segmentierung des Rands der Iris. Um den Rand der Iris zu detektieren, wird ein Vorgehen aus der Computergrafik angewendet. Dabei geht man von einem Punkt im Inneren des zu segmentierenden Bereichs aus und läuft auf Strahlen dem Rand des Bilds entgegen; bei einer großen Grauwertänderung wird dann eine Messung generiert. So geht man auf sehr vielen Strahlen ausgehend von diesem Punkt aus, um so Randpunkte der Iris zu finden. Ein guter Ausgangspunkt für ein solches Verfahren ist die Pupille. Mit Hilfe einer Adaption des Algorithmus von Asadifard et al. [Asa10] wird

der Mittelpunkt der Pupille markiert und somit der Ausgangsgpunkt für die Detektion des Rands der Iris generiert.

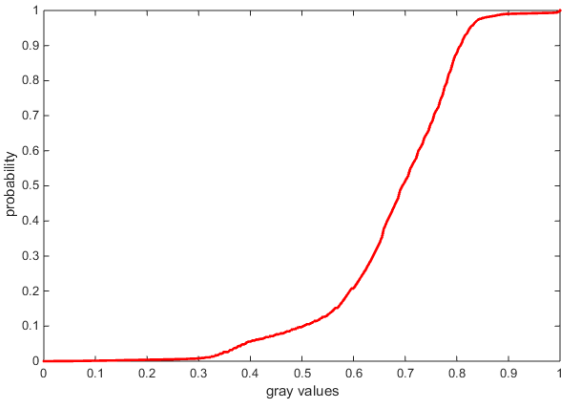
Ziel dieser Segmentierung ist die Detektion des Rands der Iris. Da die Iris nicht der dunkelste Bereich des Auges ist und auch nicht der hellste, muss zunächst eine Transformation des Bilds durchgeführt werden, die den Bereich der Iris hervorhebt. Bei der Iris handelt es sich im Grauwertbild um einen dunkleren Bereich im Vergleich zur Sclera. Somit wird eine Transformation benötigt, die dunkle Bereiche im Bild verstärkt. Eine solche Verstärkung kann durch Anwendung einer logarithmischen Bildtransformation erreicht werden. Nachdem die dunklen Bereiche hervorgehoben wurden, muss weiterhin die Iris erkannt werden. Dazu wird ein Schwellwert bestimmt, mit dem durch eine Binarisierung des Farbbereichs genau die Pupille übrig bleibt, um so einen Ausgangspunkt zur Detektion der Iris zu erlangen. In einem ersten Schritt werden die Grauwerte des Bilds in das Intervall $[0, 1]$ transformiert. Anschließend wird das Histogramm der transformierten Grauwerte bestimmt, um hieraus die kumulative Verteilungsfunktion der Grauwerte zu erlangen. Das resultierende Grauwerthistogramm sowie die kumulative Verteilung zu diesem Histogramm ist für ein beliebiges Augenbild in Abbildung 4.9 dargestellt.

Die kumulative Verteilung der Grauwerte in 4.9 ist die Grundlage für die Schwellwertbestimmung zur Segmentierung der Iris. In Anlehnung an das Vorgehen in Asadifard et al. [Asa10] wird der Bereich der Verteilungsfunktion betrachtet, in dem die niedrigsten Grauwerte enthalten sind. Die Pupille ist der dunkelste Bereich im Auge. Die Iris ist im Vergleich zur Pupille im Grauwertbild etwas heller. Somit muss der Schwellwert etwas höher gewählt werden. Bei der aktuellen Skalierung des Auges umfasst die Fläche der Iris ca. 5% der Bildfläche. Somit wird der Schwellwert gewählt, bei dem die kumulative Verteilung 5% der gezählten Grauwerte enthält. Basierend auf diesem Schwellwert wird ein Binärbild erzeugt. Abbildung 4.10 stellt das Binärbild dar.

Im Binärbild 4.10 sind Teile des Brillenrands und die Iris sichtbar. Iris und Brillenrand unterscheiden sich grundsätzlich in ihrer Form. Die Iris kann durch ein Rechteck mit ähnlich großen Kantenlängen umrandet werden. Bei einer perfekten Abbildung der Iris wäre die umrandende Form ein Quadrat. Im Fall



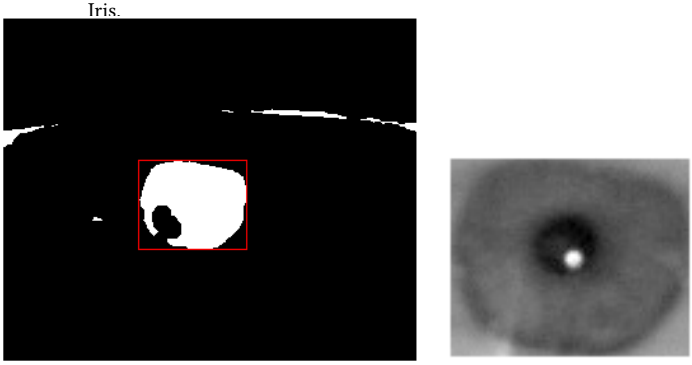
(a) Histogramm



(b) CDF

Abbildung 4.9: Grauwerthistogramm und kumulative Verteilung (CDF) eines logarithmisch transformierten Augenbilds.

Abbildung 4.10: Binärbild als Resultat der adaptiv ermittelten Schwelle zur Segmentierung der Iris.



(a) **(b)**

Abbildung 4.11: Binärbild mit ROI um die Iris-Region (a) und extrahiertes Iris-Abbild in (b).

des Brillenrands wäre das umschließende Rechteck ein Rechteck mit sehr großen Unterschieden zwischen den beiden Kantenlängen. Somit können die resultierenden Kandidaten gut unterschieden werden und das korrekte Rechteck identifiziert werden. In Abbildung 4.11 ist das Binärbild dargestellt mit (a) dem umschließenden Rechteck im Gesamtbild und (b) dem ausgeschnittenen Iris-Bereich des Auges.

In Abbildung 4.11 (b) ist nicht die gesamte Fläche der Iris enthalten. Das ist damit zu begründen, dass es in diesem Bild helle Reflexionen auf der Iris gibt. Diese konnten nicht durch Vorverarbeitung eliminiert werden. Es genügt jedoch den Rand der Iris zu erfassen. Damit alle Randpunkte erfasst werden

(a) Iris-Maske **(b)** Iris

Abbildung 4.12: Iris Maske nach zwei-stufiger Schwellwertanwendung (a) und resultierendes Binärbild zur Extraktion der Iris (b).

können, wird empfohlen das umschließende Rechteck isotrop zu skalieren und somit zu vergrößern. Der so gewonnene Ausschnitt wird auf das logarithmisch transformierte Bild angewendet. Innerhalb dieses Ausschnitts wird in einem nächsten Schritt die Pupille detektiert, um einen weiteren Schwellwert zu generieren. Dieser Schwellwert wird auf Abbildung 4.11 (b) angewendet. Das resultierende Binärbild wird invertiert, wodurch eine Maske entsteht, mit der die Pupille ausgeblendet wird. Mit Hilfe der Iris-Maske in Abbildung 4.12 (a) kann jetzt die Iris vollständig aus dem Bild extrahiert werden. In Abbildung 4.12 (b) ist der segmentierte Bereich der Iris im vergrößerten Ausschnitt als Binärbild dargestellt.

Das Binärbild in Abbildung 4.12 (b) kann jetzt dazu verwendet werden, Randpunkte der Iris zu lokalisieren. In der weiteren Modellierung wird zunächst angenommen, dass die Iris ein Kreis ist. Die Abbildung einer Iris bildet jedoch selten einen vollständigen Kreis. Dies ist durch perspektivische Verzerrungen und durch die Verdeckung durch die Augenlider zu erklären. Aus diesen Grund werden zunächst die Randpunkte im Binärbild extrahiert. Die Punkte werden dann dazu verwendet, die konvexe Hülle dieser Punkte zu bestimmen. Alle Punkte, die auf der konvexen Hülle liegen, werden als Messwerte des Iris-Rands interpretiert. Die resultierenden Messpunkte werden als Messung für das Tracking verwendet. Zur Visualisierung der erfolgten Segmentierung wurden die Messpunkte als Polygon zusammengefasst und in das Ausgangsbild eingezeichnet. Abbildung 4.13 zeigt das Ergebnis der Segmentierung.

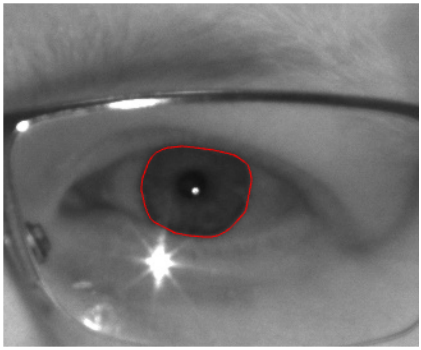


Abbildung 4.13: Segmentierte Iris mit markiertem Iris-Rand basierend auf Abbildung 4.11

Der segmentierte Iris-Rand wird als Messung im Iris-Tracker verwendet. Dazu müssen die Unsicherheiten der Randpunktmessungen beurteilt und beschrieben werden.

4.3.2 Messunsicherheiten der Iris-Messungen

Aufgrund von verrauschten Kamerabildern, sowie von Unschärfe, die durch Bewegung des Probanden entstehen können, kann es zu unscharfen Darstellungen des Iris-Rands kommen. Diese Unsicherheiten müssen modelliert werden, damit eine Verwendung der Randpunkte als Messung im Tracker durchgeführt werden kann. Unscharfe Ränder führen zu verwischten Gradienten. Zur Quantifizierung des Messrauschens wurden in einem empirischen Versuch die Gradienten bestimmt und deren Länge gemessen. Zusätzlich wurde dabei eine Segmentierung der Iris durchgeführt und mit annotierten Bildern verglichen. Durch den empirischen Versuch ergibt sich eine mittlere Abweichung von 5 Pixeln für die Randpunkte von der wahren Position. Abbildung 4.14 zeigt exemplarisch eine solche Abweichung dargestellt.

Zur Charakterisierung des Messrauchens muss die Kovarianzmatrix einer Messung bestimmt werden. Hierzu wird zunächst die Annahme getroffen, dass alle Messpunkte unabhängig voneinander sind. Das hat den Vorteil, dass die Kovarianzmatrix als Diagonalmatrix definiert werden kann. Ausgehend

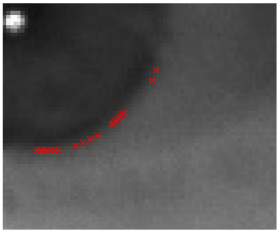


Abbildung 4.14: Ungenauigkeit der extrahierten Randpixel aufgrund des Bildrauschens.

von der vorherigen Analyse des Fehlers, basierend auf den verwischten Gradienten kann durch

$$\mathbf{C}_v = \text{diag}\left\{\sigma_v^2, \cdots, \sigma_v^2\right\} \; \sigma_v^2 = 25 \tag{4.51}$$

die zu den Messungen gehörige Kovarianzmatrix als Diagonalmatrix angegeben werden. Die Kovarianzmatrix wird benötigt, um die Messungen im Filterschritt des Trackers verwenden zu können.

4.3.3 Tracker Definition

Im Modell wird die Iris in Form eines Kreises repräsentiert. Mit einer punktförmigen Repräsentation kann zum Beispiel nur der Mittelpunkt der Iris getrackt werden. Damit können keine weiteren Informationen aus dem Tracking abgeleitet werden. Zum Beispiel kann mit einem Kreismodell das Tracking mit einer gleichzeitigen Identifikation der Iris verbunden werden. Ebenso ist es mit einer ausgedehnten Betrachtung möglich abzuleiten, wohin ein Mensch blickt. Für ein Tracking mit ausgedehnten Objekten muss das klassische Tracking zunächst angepasst werden, insbesondere im Filterschritt. Es muss eine Repräsentation definiert werden, die die gesamte Iris als Objekt enthält.

Weiter oben wurde bereits erwähnt, dass die Gestalt der Iris einem Kreis entspricht. Somit muss das Shape-Modell ein Kreis sein. Wichtig für ein solches Shape-Modell ist die Darstellung in Parameterform. Durch

$$\Phi_x(s) = \begin{bmatrix} x_M + r \cdot \cos(s) \\ y_M + r \cdot \sin(s) \end{bmatrix} \tag{4.52}$$

ist eine Parameterform gegeben, wobei das Subskript x auf die Abhängigkeit von den Parametern hindeutet. Als Parameter werden der Mittelpunkt $\underline{x}_M = [x_M, y_M]^T$ und der Radius r des Kreises verwendet. Durch $s \in [0, 2\pi)$ wird ein Laufparameter definiert, der die Punkte auf der Kreisoberfläche definiert. Der Parameter s wird benötigt, damit die Randpunkte der Iris in Zusammenhang mit dem Formmodell gebracht werden können. Die Koordinaten des Mittelpunkts und der Radius des Kreises bilden zusammen den Systemzustand für den Tracker. Somit kann durch

$$\underline{x}_k = [x_M, y_M, r]^T \tag{4.53}$$

der Systemzustand des Iris-Trackers definiert werden. Mittelpunkt und Radius sind veränderlich, da sie stark von der Kameraposition und dem Abstand der Person zur Kamera abhängen. Zusätzlich zu den äußeren Parametern führt die Person autonome Bewegungen mit dem Augen durch, wodurch keine koordinierte Bewegung der Iris angenommen werden kann. Somit können koordinierte Modelle wie das Constant Velocity Modell nicht verwendet werden. Da keine Annahme über das Bewegungsmodell getroffen werden kann, wird in diesem Fall ein Constant Position Modell eingesetzt. Mit

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.54}$$

kann das Bewegungsmodell für das Trackingsystem angegeben werden. Mit dem Systemmodell wird die erwartete systematische Veränderung des Systemzustands beschrieben. Diese systematische Veränderung unterliegt Unsicherheiten, da die Natur nicht vollständig korrekt durch das Systemmodell beschrieben werden kann. Über eine unkorrelierte Rauschkovarianzmatrix wird diese Unsicherheit des Systems beschrieben. Der Systemzustand enthält zwei Größen, den Mittelpunkt der Iris und den Radius der Iris, wobei der Mittelpunkt durch zwei Komponenten beschrieben wird. Der Mittelpunkt der Iris kann sich durch Bewegungen des Auges, und des Kopfes im beobachteten Bereich verschieben. Ebenso kann der Radius der Iris durch Veränderung der Beobachtungsentfernung Änderungen unterliegen. Die schnellste mögliche Bewegung des Auges ist die sogenannte Sakkade. Eine Sakkade ist eine sehr schnelle Bewegung des Auges, die während der Durchführung der Bewegung keine Informationsverarbeitung enthält. Eine Sakkade kann Winkelgeschwindigkeiten von bis zu 900″ erreichen. Aufgrund dieser möglichen Fehlerquellen muss die Kovarianzmatrix des Systemrauschens die Therme σ_x^2 und σ_y^2 enthalten. Wie bereits weiter oben erwähnt unterliegt auch der Radius der Iris möglichen Veränderungen. Zum Beispiel durch Erhöhung oder Verringerung des Abstands zur Kamera. Daraus leitet sich der Therm σ_r^2 ab. Mit der Definition dieser Größen ergibt sich durch

$$\mathbf{C}_w = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_r^2 \end{bmatrix}, \tag{4.55}$$

die Kovarianzmatrix des Systemrauschens. Die Unabhängigkeit der Systemkomponenten kann dadurch begründet werden, dass eine isolierte Translation der x-Komponente des Mittelpunktes auftreten kann, ebenso wie eine isolierte Translation des y-Wertes. Außerdem verändert sich durch eine reine Translation des Mittelpunktes der Radius der Iris nicht. Durch eine reine Erhöhung oder Verringerung des Radius, wird sich die Position des Mittelpunktes nicht verändern.

Im Hinblick auf die Experimente für das Tracking der Iris kann der Messaufbau zur Quantifizierung der Rauschterme verwendet werden. Es kommt eine Kamera mit einer Bildwiederholrate von 60 Bildern pro Sekunde zum Einsatz. Ausgehend von physiologisch ermittelten Bewegungsgeschwindigkeiten [Sch00] ergibt sich eine Standardabweichung von 15 Grad im Bild bei gegebener Bildwiederholrate, das unter Verwendung der optischen Eigenschaften der Kameraoptik und einer angenommenen festen Entfernung von 80 cm zur Kamera eine positionelle Veränderung des Mittelpunktes einer Fläche von 22px^2 entspricht. Somit können die Rauschterme durch $\sigma_x = \sigma_y = 22$ angegeben werden. Der ermittelte Radius der Iris unterliegt aufgrund der Schätzung einem weiteren Fehler, dieser kann mit einer Standardabweichung von $\sigma_r = 3.5$ Pixeln angegeben werden.

Mit der Definition der Kovarianzmatrix des Systemrauschens kann durch

$$\underline{x}_{k+1}^{p} = \mathbf{A} \cdot \underline{x}_{k}^{e} \tag{4.56}$$

$$\mathbf{C}_{k+1}^{p} = \mathbf{A} \cdot \mathbf{C}_{k}^{e} \mathbf{A}^{T} + C_{k}^{w} \tag{4.57}$$

der prädizierte Systemzustand $\underline{x}_{k+1}^{p}$ sowie die prädizierte Systemkovarianzmatrix $\mathbf{C}_{k+1}^{p}$ berechnet werden. Dieser Schritt entspricht einem linearen Prädiktionsschritt, wie er für ein lineares Kalman Filter in Abschnitt 3.2 definiert wurde.

4.3.4 Messmodell für das Iris-Tracking

Mit dem Systemzustand wird ein ausgedehntes Objekt beschrieben. Als Messungen werden Punkte auf dem Rand der Iris verwendet. Das Messmodell muss die gemessenen Randpunkte der Iris mit dem ausgedehnten Objekt in Verbindung bringen. Beim Tracking ausgedehnter Objekte, wie es zum Beispiel in Zea et al. [Zea14] oder Faion et al. [Fai15] angewandt wird, kommt das sogenannte Greedy Association Model (GAM) zum Einsatz. Das GAM geht davon aus, dass jeder gemessene Punkte eine generierende Quelle besitzen

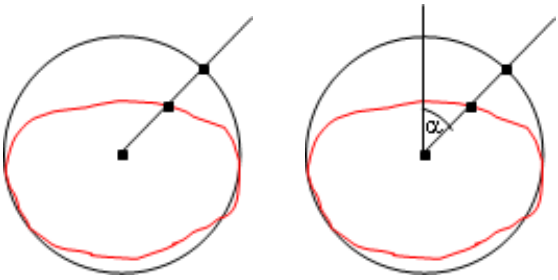


Abbildung 4.15: Erläuterung des GAM Prinzips (a) und Bestimmung des Winkels (b) anhand einer Iris-Messung.

muss. Eine generierende Quelle ist zum Beispiel ein Punkt, der für die Emission einer Messung zuständig ist. Das heißt, jedem gemessenen Punkt muss die potentiell zugehörige generierende Quelle zugeordnet werden. Mathematisch kann das GAM als Abbildung aufgefasst werden, die jeder Messung y seine generierende Quelle s zuordnet. Durch

$$\pi\left(\underline{y}\right) = s \tag{4.58}$$

wird diese Abbildung definiert. Für das ausgedehnte Iris-Modell, einem Kreis, entspricht die generierende Quelle s dem Winkel, in dessen Richtung eine Gerade durch den Ursprung des Kreises und die Kreisoberfläche verläuft. Der Verlauf einer solchen Geraden, sowie der gesuchte Winkel sind in Abbildung 4.15 dargestellt.

In Abbildung 4.15 sind ein gemessener Iris-Rand, der geschätzte Mittelpunkt des Kreises und der zugehörige Kreis eingezeichnet. Ausgehend von einem Punkt auf dem gemessenen Iris-Rand wird die Gerade mit dem Mittelpunkt und dem Punkt auf dem Iris-Rand bestimmt. Der gesuchte Winkel s ergibt sich als Winkel zwischen der Geraden, die in Richtung 0 Grad, ausgehend vom Mittelpunkt, den Kreis durchstößt und der neuen Geraden durch den Iris-Randpunkt. Unter Anwendung des GAM kann jetzt für jede Randpunktmessung $\underline{y}_i$ der zugehörige Winkel s_i bestimmt werden, sodass die Messungen

direkt im Filterschritt eines Filters verwendet werden können. Da das Messmodell nichtlinear ist, kann kein lineares Kalman Filter verwendet werden. Daher wird das UKF zur Ausführung des Filterschritts angewendet.

Durch den Prädiktionsschritt sind der Mittelwert $\underline{x}_k^p$ und die Kovarianzmatrix $\mathbf{C}_k^p$ bekannt. Nach der Definition von Huber et al. [Hub08] kann die Annahme getroffen werden, dass die prädizierte Dichte als Gaußdichte vorliegt, da diese vollständig durch Mittelwert und Kovarianzmatrix bestimmt werden kann. Durch Anwendung des Vorgehens in 3.2.4, werden basierend auf $f\left(\underline{x}\right) = \mathcal{N}\left(\underline{x} - \underline{x}_k^p, \mathbf{C}_k^p\right)$ skalierte, symmetrische Sigmapunkte $\underline{x}_i$ berechnet. Die Skalierung und die symmetrische Verteilung der Sigmapunkte entspricht dem resultierenden Kovarianzellipsoid. Somit finden sich Kovarianzmatrix und Mittelwertvektor in den Sigmapunkten wieder. Jedem Sigmapunkt $\underline{x}_i$ ist ein Gewichtungsfaktor w_i zugeordnet. Mit den generierenden Quellen s_j aus der Randmessung der Iris für den Filterschritt, werden mit den Messmodell Sigmapunkte der Messdichte bestimmt. Mit

$$\underline{y}_i = \phi_{x_i}\left(\underline{s}\right) \tag{4.59}$$

wird ein Sigmapunkt für die prädizierte Messung erzeugt, wobei $\underline{s} = [s_1, \cdots, s_n]^T$ alle potentiellen Quellen enthält. Somit ergibt sich ein Messvektor. Für jeden der Sigmapunkte der prädizierten Dichte werden einzelne Messvektoren basierend auf den potentiellen Quellen bestimmt. Die so entstandenen Sigmapunkte der Messdichte können jetzt dazu verwendet werden, den Mittelwert und die Kovarianzmatrix der Messdichte zu bestimmen. Unter Verwendung des Stichprobenmittelwerts und der Stichprobenkovarianzmatrix werden die Momente der Messdichte berechnet. Die zugrundeliegenden Gewichte werden von der prädizierten Dichte übernommen. Unter Verwendung von (3.36) und (3.37) können die Messkovarianzmatrix sowie die Kreuzkovarianzmatrix bestimmt werden. Diese beiden Matrizen werden

dazu benötigt, um das Kalman-Gain aus (3.38) zu berechnen. Mit Hilfe des Kalman-Gains, sowie der prädizierten Messung

$$\underline{y}_k^p = \sum_{i=1}^{L} \omega_i \cdot \underline{y}_i \tag{4.60}$$

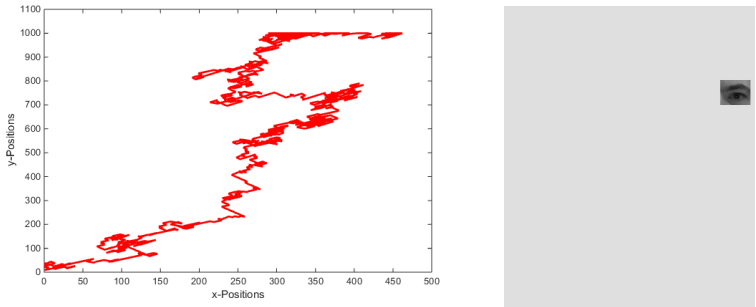
können jetzt der Mittelwert und die Kovarianzmatrix der geschätzten Dichte ermittelt werden. Dazu werden (3.39) und (3.40) angewendet. Hierdurch ergibt sich eine verbesserte Schätzung des Systemzustands, wodurch die Position und Größe der Iris geschätzt werden können.

Der Tracker wird rekursiv angewandt, um die Iris über die Zeit tracken zu können. Liegt einmal keine Messung vor, so wird die prädizierte Dichte als Schätzung verwendet. Sobald erneut eine Messung vorliegt, erfolgt eine Verbesserung der Schätzung durch Anwendung eines erneuten Filterschritts.

Der entwickelte Tracker für die menschliche Iris wird in Simulationen erprobt. Der folgende Abschnitt behandelt diese Simulationen.

4.3.5 Iris-Tracking: Experimente

Als Datengrundlage für das Experiment wird eine Bildsequenz eines menschlichen Gesichts verwendet. Die verwendete Kamera ist vom Typ Flea 3 USB 3 der Firma Point Grey. Diese Kamera wird mit 60 Frames pro Sekunde (FPS) betrieben. Die Kamera wurde an einer Stelle fixiert und so ausgerichtet, dass sie das Gesicht im Zentrum des Bildes aufnehmen konnte. Zur Beleuchtung der Szene wurden drei LED-Arrays eingesetzt. Die LED-Arrays beleuchten die Szene im nahen Infrarot-Spektralbereich. Das menschliche Auge ist in nahen Infrarotbereich gut vom restlichen Gesicht zu unterscheiden, insbesondere ist die Pupille sehr gut sichtbar. Ein weiterer Grund für die Nutzung dieses Spektralbereichs ist die Sensitivität der Kamera. Die Kamera ist eine Grauwertkamera. Die LED-Arrays sind Teil eines Eye-Trackers, der am Fraunhofer IOSB entwickelt wurde. Die LED-Arrays wurden so konfiguriert, dass sie alternierend aufleuchten. Somit ist keine konstante Beleuchtung vorhanden.



(a) **(b)**

Abbildung 4.16: Zufällige Trajektorie entlang der die extrahierten Augenbilder in Einzelbilder eingebettet wurden (a). Ein Beispiel für ein Einzelbild (b) aus dieser Serie.

Hierdurch weisen die entstandenen Bilddaten unterschiedliche Beleuchtungssituationen auf.

Für die Aufnahmen wurde ein Proband gebeten, mit offenen Augen vor der Kamera zu verweilen ohne zu blinzeln. Es wurden mehrere Sequenzen aufgenommen. Unter diesen Sequenzen wurde die Sequenz gewählt, bei der es dem Probanden gelang, die Augen am längsten aufzulassen. Die gewählte Aufnahme beinhaltet 282 Einzelbilder. Um die Grundgesamtheit der Bilddaten zu erhöhen, werden wiederholte, zufällige Permutation der 282 Bilder gesammelt. Durch dieses Vorgehen resultiert eine Videosequenz mit 1128 Einzelbildern. Weiterhin werden die Einzelbilder vorverarbeitet. Aus jedem der Gesichtsbilder wird das linke Auge des Probanden extrahiert. Aus der Extraktion des Auges sind Bildausschnitte mit variabler Größe entstanden. Die kleinsten Ausschnitte besitzen eine Größe von 112 x 116 Pixeln, wohingegen der größte Ausschnitt eine Fläche von 136 x 140 Pixeln misst. Diese entstandenen Bildausschnitte wurden entlang einer randomisierten Trajektorie in einen grauen Hintergrund eingebettet. Für jeden Bewegungspunkt wurde ein neues Bild generiert. Da auch die Augenbilder variable Beleuchtungssituationen beinhalten, wurde der Grauwert des Hintergrunds zufällig für jedes Bild aus dem Intervall $[200, 255]$ gewählt. Die zufällige Trajektorie ist in Abbildung 4.16 (a) abgebildet. Abbildung 4.16 (b) zeigt eines der generierten Bilder.

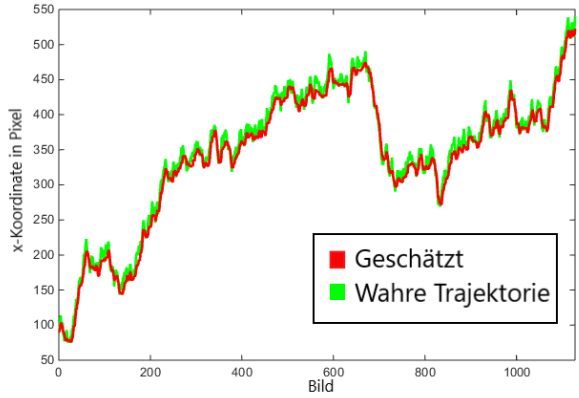
Mit diesem Experiment wird gezeigt, dass der vorgeschlagene Algorithmus der Iris auf einer zufälligen Trajektorie folgen kann. Die Bilder werden so positioniert, dass der Mittelpunkt der Pupille entlang der Trajektorie bewegt wird. Somit wird ein direkter Vergleich des geschätzten Mittelpunkts der Iris mit der tatsächlichen Position möglich. Zusätzlich ist für jedes Bild der Sequenz die Größe der Iris bekannt. Der Vergleich der realen Trajektorie mit der aus der Schätzung resultierenden Trajektorie ist in Abbildung 4.17 dargestellt. Zur Verdeutlichung der Trackingqualität werden in Abbildung 4.17 (a) der Verlauf des x-Wertes der Trajektorie mit der Schätzung verglichen und in Abbildung 4.17 (b) der Verlauf des y-Wertes der Trajektorie mit der Schätzung.

Die in Abbildung 4.17 gezeigte Schätzung ist gut an den wahren Verlauf der Trajektorie angepasst. In der x-Richtung ist eine größere Abweichung von der wahren Trajektorie zu erkennen als bei der y-Trajektorie. Zur Verdeutlichung des Positionierungsfehlers werden der x- und y-Fehler unter Anwendung der euklidischen Distanzfunktion zusammengefasst. Der resultierende Positionierungsfehler ist in Abbildung 4.18 dargestellt.

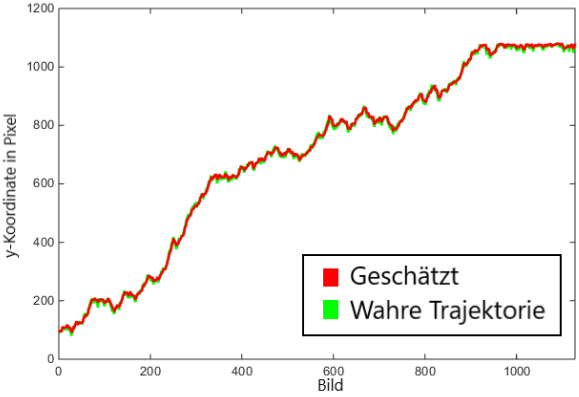
Durch Mittelung der Positionierungsfehler wurde eine mittlere Fehlerdistanz von $\epsilon_{\text{pos}} = 10.1 \pm 4.9\text{px}$ bestimmt.

Neben der Bestimmung des Iris-Mittelpunkts erfolgt eine Schätzung des Radius der Iris. Zur Bestimmung des wahren Radius wurde für jedes Bild der Rand der Iris manuell annotiert. Unter Verwendung der Annotation erfolgte die Schätzung des kleinsten Kreises, der den Rand enthält. Basierend auf der Kreisschätzung wurde so der wahre Radius geschätzt. Zur Ermittlung des Fehlers bei der Radius-Schätzung wurden die ermittelten Werte direkt mit dem geschätzten Radius verglichen. Das Ergebnis der Schätzung des Radius, sowie der wahre Radius sind in Abbildung 4.19 (a) dargestellt. Abbildung 4.19 (b) stellt den resultierenden Fehler dar.

Nach Auswertung des Radius-Fehlers ergibt sich ein systematischer Fehler von 5.6 Pixeln mit einer Standardabweichung von 1.6 Pixeln für die Schätzung des Radius. Somit ist der mittlere Fehler für die Schätzung des Radius sehr viel geringer als der Positionierungsfehler. Zusätzlich ist sichtbar, dass der geschätzte Radius immer etwas größer ist als der wahre Radius. Dieser Fehler



(a)



(b)

Abbildung 4.17: Vergleich der geschätzten und wahren Trajektorie der dargestellten Iris in (a) x-Richtung und (b) y-Richtung.

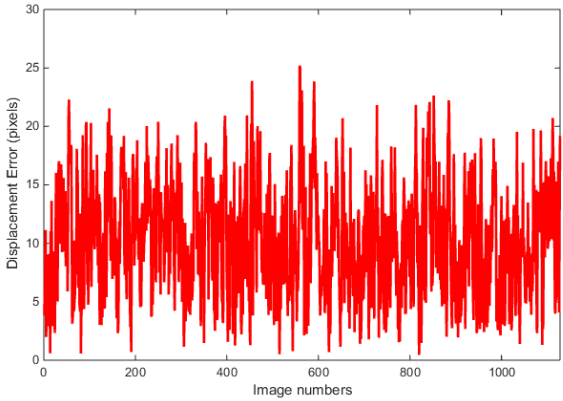
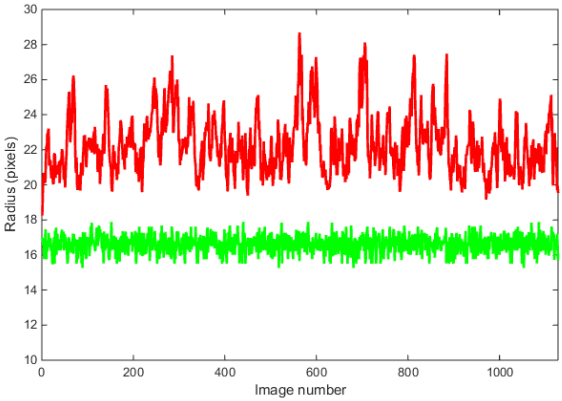


Abbildung 4.18: Fehler der Positionsbestimmung des Iris-Mittelpunktes für jedes Bild der Sequenz.

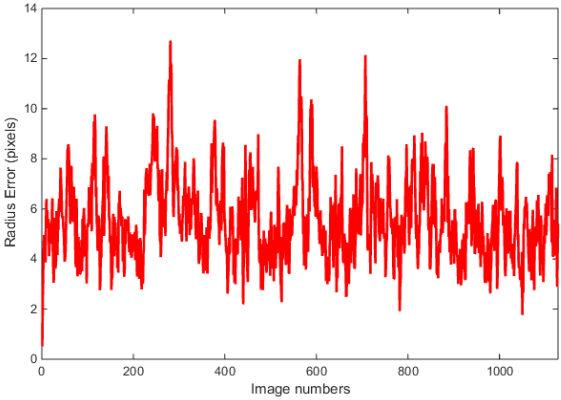
scheint vernachlässigbar, da hierdurch kein Nachteil entsteht. Zusätzlich ist durch Verwendung eines ausgedehnten Objekts als Zustand garantiert, dass die vollständige Iris während des Trackings innerhalb des Modells enthalten ist. Die resultierenden Fehler für das Tracking zeigen, dass der vorgeschlagene Tracker in der Lage ist, der Iris zu folgen.

4.3.6 Iris-Tracking: Zusammenfassung

Mit dem vorgestellten Trackingansatz wurde gezeigt, dass durch die Verwendung eines ausgedehnten Objekts eine genaue Schätzung der Position des verfolgten Objekts möglich ist. Zudem wurde gezeigt, dass eine geschickte Wahl einer ausgedehnten Objektbeschreibung weitere Verarbeitungsschritte des Trackingergebnisses ermöglicht. In dem beschriebenen Experiment wurde eine Trajektorie für das linke Auge simuliert. Diese Trajektorie entspricht vermutlich nicht dem physiologischen Bewegungsmuster eines Auges. Dennoch war der Tracker unabhängig von der gewählten Bewegung in der Lage der Iris zu folgen. Weiter konnte gezeigt werden, dass zu keinem Zeitpunkt die Gefahr bestand, dass der Tracker die Iris verliert. Grundsätzlich schien der Tracker die Größe der Iris leicht zu überschätzen; allerdings entspricht



(a)



(b)

Abbildung 4.19: Vergleich zwischen dem geschätzten Radius (rot) und dem wahren Radius (grün) in (a) und der resultierende Fehler in (b).

das bei einem systematischen Fehler von 5.6 Pixeln mit einer Standardabweichung von 1.6 Pixeln ungefähr dem Rauschen des Bilds. Somit kann davon ausgegangen werden, dass der Tracker bei geringerem Bildrauschen sowie sehr gut eingestellten Parametern für die Rauschterme eine weitaus genauere Schätzung liefern kann.

Abschließend bleibt zu sagen, dass die Verwendung einer ausgedehnten Repräsentation des Systemzustands in Verbindung mit dem GAM eine gut funktionierende Kombination ist. Aus diesem Grund wird eine ähnliche Vorgehensweise für das Tracking des Gesichts gewählt.

4.4 Tracking des menschlichen Gesichts

Um weitere Analysen des menschlichen Gesichts in einer dynamischen Umgebung zu ermöglichen, ist das Tracking des Gesichts notwendig. Diese weiteren Analysen erfordern ein ausgedehntes Beschreibungsmodell für das Gesicht. Ein Landmarkenmodell, wie es in Abschnitt 3.1 eingeführt wurde, stellt eine geeignete Darstellungsform dar. Dieses Landmarkenmodell umfasst 68 einzelne Punkte. Die Punkte können in Zwischenpunkte und echte Landmarken unterteilt werden. Unter echten Landmarken versteht man Punkte im Gesicht, die prominent sind. Dies können zum Beispiel die Augenwinkel und Mundwinkel sein. Die Zwischenpunkte liegen auf Kurven, die die echten Landmarken miteinander verbinden. Die Zwischenpunkte werden durch diese Kurven interpoliert und in gleichmäßigen Abständen platziert, siehe Cootes et al. [Coo00]. Die Landmarken werden unter Verwendung von Detektionsmethoden extrahiert. Beispiele für Detektionsmethoden sind in Abschnitt 3.1 aufgelistet. Für das hier beschriebene Tracking wird der von Qu et al. in [Qu15a] beschriebene Detektionsalgorithmus verwendet. Eine genauere Beschreibung des Algorithmus kann in Abschnitt 3.1.1 nachgelesen werden. Der Detektionsalgorithmus passt das Landmarkenmodell mit 68 Punkten an ein neues Gesicht an. Die Ausgabe mit angepassten Landmarken ist am Beispiel eines Gesichts mit erfreutem Ausdruck in Abbildung 4.20 dargestellt. Das Bild enstammt der Cohn-Kanade+ Datenbank (CK), die von Lucey et al.

Abbildung 4.20: Subjekt S055 der CK Datenbank (©J. Cohn) mit an das Gesicht angepasstem 68 Punkt-Landmarkenmodell.

[Luc10b] veröffentlicht wurde. Das Bild zeigt Subjekt S055 bei vollem Ausdruck der Emotion Freude. Die CK-Datenbank enthält eine Sammlung von Bildern in verschiedenen Emotionen. Grundsätzlich sind dort folgende Emotionen enthalten: *Wut*, *Ekel*, *Furcht*, *Freude*, *Traurigkeit* und *Überraschung*. Diese sechs Emotionen entsprechen den sogenannten Basisemotionen nach Ekman in [Ekm99]. Die Bilder in der CK-Datenbank sind in Sequenzen angeordnet. Jede dieser Sequenzen beginnt mit dem neutralen Ausdruck und endet mit einem vollen emotionalen Gesichtsausdruck.

Die weiteren Analysen haben das Ziel, den emotionalen Zustand der beobachteten Person auswerten zu können. Daher muss das Trackingsystem die Form des Gesichts erhalten und mit einer hohen Genauigkeit wiedergeben. Grundlegend, um eine hohe Genauigkeit zu erlangen, ist die Verwendung eines zuverlässigen Extraktionsalgorithmus. Der Extraktionsalgorithmus wird verwendet, um eine Beobachtung oder Messung des Gesichts vorzunehmen. Eine Messung wird durch 68 Punkte repräsentiert. Das verwendete Medium zur Beobachtung des Gesichts ist eine Kamera. Es liegen keine Tiefeninformationen vor. Die Punkte werden als zweidimensionale Vektoren

$$\underline{p}_i = [x_i, y_i]^T \tag{4.61}$$

repräsentiert, wobei $i \in \{0,1,\ldots,67\}$ gilt. Die direkte Beobachtung des Gesichts durch eine Kamera erfordert eine Kommunikationssituation zwischen einem Menschen und einem Computersystem oder Roboter. Das Gesicht ist in einer solchen Situation zumeist frontal sichtbar, wobei zeitweise Verdeckungen durch Hände oder Haare vorkommen können. Zur Erhaltung der durch das Modell gegebenen Form ist es notwendig, Randbedingungen für den Trackingalgorithmus zu formulieren.

Die weiteren Unterabschnitte des Gesichtstrackings bauen sich wie folgt auf: zunächst werden das grundlegende Trackingmodell mit System- und Messmodell und die zu beschreibenden Randbedingungen definiert. Dann erfolgt die Definition eines einfachen Trackers auf Basis eines Kalman Filters. Im darauffolgenden Abschnitt wird das Modell um die definierten Randbedingungen, die zur Erhaltung der Form beitragen, erweitert. Abschließend erfolgt eine Evaluation des vorgestellten Modells und eine Zusammenfassung des Gesichtstrackings.

4.4.1 Trackingmodell für das Gesichtstracking

Durch das Tracking der Gesichtslandmarken wird zu jeder Zeit eine Repräsentation der Gesichtsform beobachtet. Damit ein funktionaler Zusammenhang zwischen zwei Zeitschritten aufrechterhalten werden kann, muss das Landmarkenmodell in ein Referenzkoordinatensystem überführt werden. Unter Verwendung der Procrustes Analyse [Ken89] können beliebige, durch Punktmengen definierte Formen unter Erhalt der eigenen Form in ein Referenzkoordinatensystem transformiert werden. In der Procrustes Analyse werden dazu drei Transformationen ausgeführt: Translation zum Ursprung des Koordinatensystems, isotrope Skalierung und Rotation zur Ausrichtung einer mittleren Form. Die zusammengesetzte Transformation entspricht einer geometrischen Ähnlichkeitstransformation. Für die Verwendung der Procrustes Analyse wird zunächst eine mittlere Form benötigt. Diese wird anhand einer Trainingsmenge von extrahierten Landmarken generiert, indem alle vorher beschriebenen Schritte durchgeführt werden. Eines der verwendeten Trainingssamples wird zufällig als mittlere Form angenommen. Die Form wird

nach den Schritten solange durch die Mittelung der resultierenden Formen ersetzt, bis die neue mittlere Form sich nicht mehr von der vorher verwendeten unterscheidet. Die drei benötigten Transformationen werden im Folgenden beschrieben.

Zur Ermittlung des Verschiebungsvektors wird der Mittelwert der Punkte berechnet und durch

$$\underline{p}'_j = \underline{p}_j - \frac{1}{n}\sum_{i=1}^{n}\underline{p}_i \;\forall j \; in \{1,n\}, \tag{4.62}$$

werden die einzelnen Punkte des Modells verschoben, wobei $n = 68$ gilt. Im nächsten Schritt erfolgt die isotrope Skalierung des Modells. Der Skalierungsfaktor wird durch Berechnung der mittleren Root Mean Squared Distance (RMSD) der Punkte bestimmt. Durch

$$s = \sqrt{\frac{1}{n}\cdot\sum_{i=1}^{n}(x_i-\hat{x})^2+(y_i-\hat{y})^2} \tag{4.63}$$

wird der Skalierungsfaktor s berechnet, wobei durch $[\hat{x},\hat{y}]^T$ der Mittelwert der Punkte beschrieben wird, der durch die Skalierung im Ursprung liegt, d.h. $[\hat{x},\hat{y}]^T = [0,0]^T$. Durch den Skalierungsfaktor s wird dazu verwendet, um zu garantieren, dass die RMSD aller zum translatierten Mittelpunt 1 beträgt. Im letzten Schritt wird das gesamte Modell rotiert und an die mittlere Form angepasst. Die Rotationskorrektur wird durch Anwendung einer Rotationsmatrix

$$\mathbf{R} = \begin{bmatrix}\cos(\phi) & -\sin(\phi)\\ \sin(\phi) & \cos(\phi)\end{bmatrix} \tag{4.64}$$

durchgeführt, wobei der Parameter ϕ der Rotationswinkel zur Anpassung des Modells an die mittlere Form ist. Der Rotationswinkel ϕ muss für jede Form durch

$$\phi = \tan^{-1}\left(\frac{\sum_{i=1}^{n}(w_i \cdot y_i - z_i \cdot x_i)}{\sum_{i=1}^{n}(w_i \cdot x_i + z_i \cdot y_i)}\right) \tag{4.65}$$

zwischen zwei Formen P und W, wobei $\underline{p}_i = [x_i, y_i]^T \in P$ mit $i \in [1, n]$ und $\underline{w}_i = [w_i, z_i]^T \in W$ mit $i \in [1, n]$ berechnet werden. Die Form P entspricht der mittleren Form und die Form W ist die anzupassende Form. Für die Ermittlung der mittleren Form muss die Procrustes Distanz berechnet werden. Wird hier ein Schwellwert unterschritten, kann die Auswahl der mittleren Form beendet werden. Die Procrustes Distanz kann durch

$$d^{PC} = \sqrt{\sum_{i=1}^{n}\left[(w_i - x_i)^2 + (z_i - y_i)^2\right]} \tag{4.66}$$

berechnet werden, wobei durch $[x_i, y_i]^T$ die Punkte der mittleren Form repräsentiert werden und $[w_i, z_i]^T$ die Punkte der angepassten Form sind. Nachdem die beobachtete Form durch die Procrustes Analyse in das Referenzkoordinatensystem überführt wurde, kann der Systemzustand definiert werden.

Ausgehend von den 68 Punkten des Landmarkenmodells können 136 einzelne Koordinaten aus dem Modell extrahiert werden. Diese 136 Werte setzen sich aus den jeweils 68 x- und y-Werten der Landmarken zusammen. Die ersten 68 Einträge beinhalten die x-Werte und in den restlichen 68 befinden sich die y-Werte. Die Werte unterliegen einer festen Sortierung, die auf einer festen

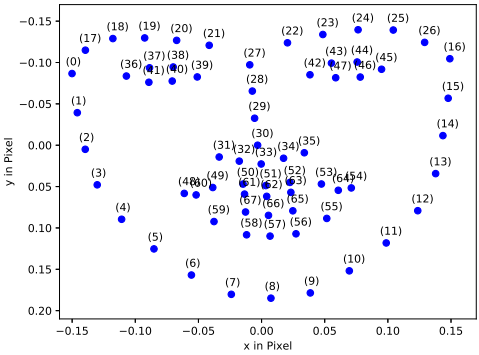


Abbildung 4.21: Das Landmarkenmodell mit 68 geordneten und gezählten Punkten als mittlere Form aus einer Menge mehreren Gesichtern.

Nummerierung der Landmarken im Modell basiert. Die Nummerierung der Landmarken ist in Abbildung 4.21 dargestellt. Der Systemzustand kann durch

$$\underline{x}_k = [x_1, \cdots, x_n, y_1, \cdots, y_n]^T \tag{4.67}$$

angegeben werden. Für das Tracking wird ein Bewegungsmodell benötigt, dass die Annahme über wahrscheinliche Bewegungen des Gesichts modelliert. Da es schwierig ist, eine koordinierte Bewegung für das Gesicht anzunehmen, wird in diesem Fall das konstante Positionsmodell (CP) aus Abschnitt 3.2.2 verwendet. Die Dimension des Zustandsvektors in (3.14) muss von 2 auf 136 Elemente angepasst werden, wodurch sich das Systemmodell durch eine Matrix $\mathbf{A} = \mathbf{I} \in \mathbb{R}^{136x136}$ beschreiben lässt. Das Systemrauschen wird als unkorreliertes, weißes Rauschen mit Kovarianzmatrix

$$\mathbf{C}_w = \text{diag}\{\sigma_w^2, \cdots, \sigma_w^2\} = \mathbf{I} \cdot \sigma_w^2 \tag{4.68}$$

angenommen.

Das Messmodell des Trackers entspricht einer direkten Abbildung des Systemzustands und entspricht, wie das Systemmodell der Einheitsmatrix. Die

realen Beobachtungen werden unter Verwendung des Landmarkendetektors von Qu et al. [Qu15a] bereitgestellt. Die realen Messungen müssen durch die Procrustes Analyse in das Referenzkoordinatensystem überführt werden. Die Matrix $\mathbf{H} = \mathbf{I}$ repräsentiert das Messmodell. Das Messrauschen wird als unkorreliertes, weißes Rauschen modelliert, wodurch die Kovarianzmatrix durch

$$\mathbf{C}_v = \text{diag}\left\{\sigma_v^2, \cdots, \sigma_v^2\right\} = \mathbf{I} \cdot \sigma_v^2 \tag{4.69}$$

angegeben werden kann.

Um den Filterschritt durchführen zu können, muss eine reale Beobachtung $\underline{\hat{y}}_{k+1}$ des Landmarkendetektors vorliegen. Unter Verwendung des Messmodells können im Filterschritt die Messkovarianzmatrix (3.21) und der Innovationsvektor (3.20) bestimmt werden. Hierzu wird mit dem Messmodell die prädizierte Messung

$$\underline{y}_k^p = \underline{x}_k^p \tag{4.70}$$

berechnet. Um mit der Messung die aktuelle Schätzung des Filters zu verbessern, muss zunächst das Kalman-Gain durch (3.22) bestimmt werden. Das Kalman-Gain wird dazu verwendet, die aktuelle Messung mit dem prädizierten Zustand zu verbinden und eine verbesserte Schätzung des aktuellen Zustands zu erhalten. Die aktuelle Schätzung wird unter Verwendung des Kalman-Gains und den Gleichungen (3.23) und (3.24) verbessert. Das Resultat sind der geschätzte Systemzustand $\underline{x}_{k+1}^e$ und die geschätzte Kovarianzmatrix $\mathbf{C}_{k+1}^e$. Durch das Kalman Filter wird ein rekursiver Algorithmus definiert, wodurch in jedem Zeitschritt das gleiche Vorgehen verwendet werden kann.

Abbildung 4.22 zeigt den Ablaufplan für den Gesichtslandmarkentracker. Der Tracker liefert durch den Prädiktionsschritt $\underline{x}_k^p$ basierend auf dem aktuellen Zustand zurück. Der Extraktionsalgorithmus von Qu et al. liefert die aktuelle Messung $\underline{\hat{y}}_k$ ausgehend von einem Kamerabild. Das Gating prüft, ob sich die gelieferte Messung von der mittleren Form unterscheidet. Hierzu wird eine

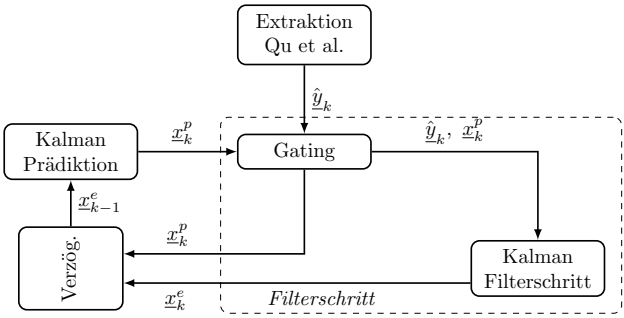


Abbildung 4.22: Ablaufplan des Trackers für das menschliche Gesicht.

Ähnlichkeitstransformation bestimmt und es wird geprüft, ob sich die berechnete Transformation von der Einheitsmatrix unterscheidet. Falls dies nicht der Fall ist, so wird der prädizierte Zustand zurückgeliefert und als aktuelle Schätzung an den Prädiktionsschritt geliefert. Ist der Unterschied ausreichend groß, werden die aktuelle Messung und der prädizierte Zustand dem Kalman Filterschritt übergeben und der Filterschritt bestimmt die aktuelle Schätzung $\underline{x}_k^e$.

Das vorliegende Trackingmodell geht von unabhängigen, unkorrelierten Zustandsvariablen aus. Der Erhalt der Form kann nicht garantiert werden. Es müssen zusätzlich Randbedingungen eingeführt werden, wodurch die Nutzung des linearen Kalman Filters nicht mehr ausreichend ist.

4.4.2 Randbedingungen für einen Tracker für ausgedehnte Objekte

Um die Randbedingung für die Erhaltung der Form zu definieren, wird zunächst das von Cootes et al. [Coo00] beschriebene Landmarkenmodell betrachtet. Dieses Modell ist so definiert, dass Variationen des Modells, wie zum Beispiel unterschiedliche Gesichtsausdrücke, zugelassen werden können. Um diese Variabilität zuzulassen, müssen viele verschiedene Gesichtsausdrücke in möglichst vielen Varianten als Landmarkenmodell vorliegen. Unter Anwendung der Hauptkomponentenanalyse (PCA) werden die Hauptachsen des

Modells bestimmt. Dazu ist die PCA so parametrisiert, dass 98% der Varianz erhalten bleiben. Damit dies gelingt wird zunächst durch

$$\underline{\hat{x}} = \frac{1}{N} \sum_{i=1}^{N} \underline{x}_i \tag{4.71}$$

der Mittelwert der Trainingssamples ermittelt, wobei N die Anzahl aller Trainingssamples ist. Der Mittelwertes $\underline{\hat{x}}$ wird verwendet, um durch

$$\mathbf{S} = \frac{1}{N-1} \sum_{i=1}^{N} \left(\underline{x}_i - \underline{\overline{x}}\right) \cdot \left(\underline{x}_i - \underline{\overline{x}}\right)^T \tag{4.72}$$

die Kovarianzmatrix über allen Samples zu bestimmen. Zur weiteren Analyse werden die Eigenvektoren $\underline{\phi}_i$ und zugehörigen Eigenwerte λ_i der Kovarianzmatrix $\mathbf{S}$ bestimmt. Die Eigenwerte und Eigenvektoren werden absteigend nach der Größe der Eigenwerte sortiert. Durch

$$\sum_{i=1}^{t} \lambda_i \geq 0.98 \cdot V_T \tag{4.73}$$

werden so viele Eigenwerte aufsummiert, dass die resultierende Varianz 98% der gesamten Varianz V_T entspricht [Coo00]. Durch die Summierung werden die relevanten Eigenvektoren ausgewählt. Die zugehörigen Eigenvektoren werden zur Matrix ⊠ zusammengesetzt, wodurch mit

$$\underline{x} \approx \underline{\overline{x}} + \Phi \underline{b} \tag{4.74}$$

jedes Sample $\underline{x}$ der Trainingsmenge approximiert werden kann. Der Vektor $\underline{b}$ wird durch

$$\underline{b} = \Phi^T \cdot \left(\underline{x} - \overline{\underline{x}}\right) \tag{4.75}$$

bestimmt und stellt eine Projektion in das durch die Hauptkomponenten bestimmte Koordinatensystem dar. Unter Verwendung von (4.74) und (4.75) können beliebige Samples zurückprojiziert und in der korrekten Form fixiert werden. Dieser Vorgang wird durch einen erweiterten Filterschritt zur Korrektur der aktuellen Schätzung $\underline{x}_k^e$ implementiert.

In [Jul07] haben Julier et al. einen Formalismus definiert, mit dem nichtlineare Randbedingungen in das Kalman Filter eingeführt werden können. Der Formalismus basiert auf der Verwendung der Unscented Transformation (UT) ohne eine zusätzliche Verwendung von Rauschtermen. Auf Basis der Schätzung $\underline{x}_k^e$ und $\mathbf{C}_k^e$ werden Sigma-Punkte bestimmt. Jeder Sigma-Punkt wird durch (4.75) zunächst auf das Hauptkomponentensystem projiziert und durch (4.74) rückprojiziert. Das gleiche Vorgehen wird für die reale Messung $\underline{\hat{y}}$ ausgeführt. Die rückprojizierten Sigmapunkte werden entsprechend des Messmodells propagiert. Basierend auf den Sigmapunkten erfolgt durch (3.37) die Berechnung der Kreuzkovarianzmatrix $\mathbf{C}_k^{xy}$ und durch (3.36) die Berechnung der Messkovarianzmatrix $\mathbf{C}_k^{\tilde{y}}$. Die beiden Kovarianzmatrizen werden dazu verwendet, das Kalman-Gain $\mathbf{K}_k^+$ entsprechend (3.38) zu berechnen. Der korrigierte Systemzustand kann durch

$$\underline{x}_k^{e+} = \underline{x}_k^e + \mathbf{K}_k^+ \cdot \left(\underline{\hat{y}} - \underline{\tilde{y}}\right) \tag{4.76}$$

bestimmt werden. Entsprechend wird durch

$$\mathbf{C}_k^{e+} = \mathbf{C}_k^e - \mathbf{K}_k^+ \cdot \mathbf{C}_k^{\tilde{y}} \cdot \left(\mathbf{K}_k^+\right)^T \tag{4.77}$$

die korrigierte Schätzung der Kovarianzmatrix bestimmt. In Cootes et al. [Coo00] finden sich Beispiele, die die Korrektur der Eingangsformen belegen. In diesem Algorithmus wird diese Korrektur dazu verwendet, damit die Rahmenbedingungen der Filtergleichung eingehalten werden können.

4.4.3 Tracking des menschlichen Gesichts

In diesem Experiment werden drei Verfahren miteinander verglichen: der vorgestellte Tracker, der Landmarkenextraktor von Qu et al. [Qu15a] und der Landmarkenextraktor der Dlib von King [Kin09]. Es werden drei unterschiedliche Sequenzen aus der CK-Datenbank entnommen. Grundlage für die Auswahl ist eine Sequenzlänge von mindestens 10 Bildern. Für alle Bilder der Sequenzen wurde die wahre Position der Landmarken durch den Landmarkenextraktor von Qu et al. [Qu15a] bestimmt. Es wurden wiederholt Schätzungen des Trackers bestimmt und durch Mittelung dieser Schätzungen wurde die Grundwahrheit für jedes Bild erzeugt.

Für das Experiment wurden alle Bilder identisch vorbereitet. Zunächst wird das Gesicht detektiert. Dazu wird ein neuronales Netz aus der Opensource Bibliothek OpenCV [1] verwendet. Das Gesicht wird ausgeschnitten und so skaliert, dass das Bild 300 x 400 Pixel umfasst. Zusätzlich erfolgt eine Aufhellung des Bilds durch eine Gamma Anpassung, mit einem Gammawert von $\gamma = 0.5$. Das vorverarbeitete Bild wird an den Landmarkenextraktor übergeben.

Im Livebetrieb als Tracker wird jedes Eingangsbild äquivalent zu der weiter oben beschriebenen Vorgehensweise vorverarbeitet. Der Tracker verwendet das Gating aus Abschnitt 4.4.1. Als Landmarkenextraktor verwendet der Tracker den von Qu et al. in [Qu15a] entwickelten Algorithmus. Der Tracker verwendet die erste Messung als Startzustand. Danach wird für das Tracking sowie zur Anwendung der Nebenbedingung ein Unscented Kalman Filter eingesetzt. Der Trackingalgorithmus wird verglichen mit der rohen Verwendung des Landmarkenextraktors von Qu et al., sowie des Algorithmus von King aus der Dlib [Kin09].

[1] https://opencv.org/about.hmtl

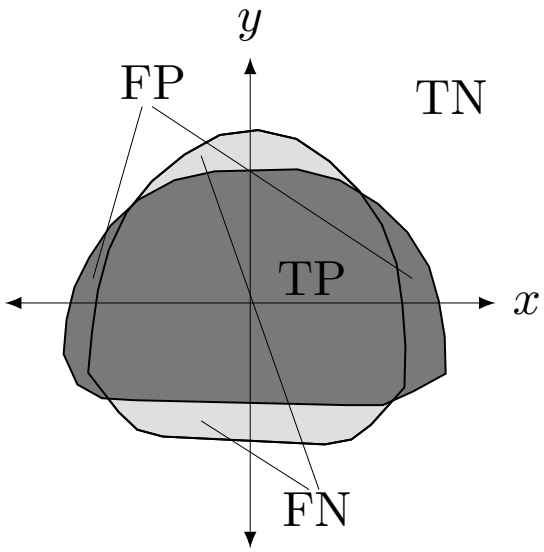


Abbildung 4.23: Überlappung zweier konvexer Polygone zur Bestimmung der Werte für TP, FP, TN und FN.

Zur Bewertung der Trackingqualität wird die konvexe Hülle des geschätzten Landmarkenmodells für jeden Algorithmus bestimmt. Dann wird ein Vergleich mit der konvexen Hülle der Referenzmessung durchgeführt. Die Unterschiede zwischen den Flächen werden in die Kategorien richtig positiv (TP), falsch positiv (FP), falsch negativ (FN) und richtig negativ (TN) eingeordnet. Das Kategorisieren wird entsprechend Abbildung 4.23 durchgeführt.

Der TP-Wert entspricht der Schnittfläche zwischen beiden Polygonen. Die Fläche des unteren Polygons, die nicht durch die Schätzung abgedeckt wird, wird als FN kategorisiert. Alle Flächen des geschätzten Polygons, die außerhalb des wahren Polygons sind, werden als FP einsortiert. Als TN wird der Hintergrund verwendet. Die Klassifizierung der Flächen als TP, FP, FN und

Tabelle 4.9: Ergebnisse für zehn Durchläufe des Trackings über 15 Bilder der Testperson S010 aus der CK Datenbank. Als Qualitätsmaß ϕ_t ist das Überlappungsmaß angegeben.

Durchlauf	Tracker	Qu et al. [Qu15a]	DLib [Kin09]
1	0.9925 ± 0.0144	$\mathbf{0.9931 \pm 0.0146}$	0.9740 ± 0.0035
2	0.9900 ± 0.0301	$\mathbf{0.9921 \pm 0.0202}$	0.9740 ± 0.0035
3	0.9986 ± 0.0008	$\mathbf{1.0000 \pm 0.0000}$	0.9740 ± 0.0035
4	0.9986 ± 0.0008	$\mathbf{1.0000 \pm 0.0000}$	0.9740 ± 0.0035
5	0.9983 ± 0.0009	$\mathbf{0.9997 \pm 0.0010}$	0.9740 ± 0.0035
6	$\mathbf{0.9983 \pm 0.0014}$	0.9961 ± 0.0146	0.9740 ± 0.0035
7	$\mathbf{0.9904 \pm 0.0100}$	0.9834 ± 0.0238	0.9740 ± 0.0035
8	$\mathbf{0.9980 \pm 0.0023}$	0.9960 ± 0.0149	0.9740 ± 0.0035
9	0.9977 ± 0.0033	$\mathbf{0.9987 \pm 0.0050}$	0.9740 ± 0.0035
10	$\mathbf{0.9912 \pm 0.0127}$	0.9900 ± 0.0186	0.9740 ± 0.0035

TN ist an die Vorgehensweise von Čehovin et al. in [Ceh16] angelehnt. Hier werden die Flächenverhältnisse dazu verwendet, um das Überlappungsmaß

$$\phi_t = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{4.78}$$

zu berechnen. Das Maß ϕ_t ist an das F-Maß angelehnt, dass einen Faktor 2 zu TP multipliziert. Durch $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$ wird die Präzision ausgehend von den Flächen berechnet.

Mit einer Sequenz von 15 Einzelbildern wird das Experiment durchgeführt. Die Sequenz wird zehn Mal durchlaufen, um herauszufinden, ob die gelieferten Ergebnisse schwanken oder stabil sind. Tabelle 4.9 fasst die Ergebnisse der drei Methoden zusammen. Es fällt sofort auf, dass der Landmarkendetektor der DLib [Kin09] in jedem Durchlauf das exakt selbe Ergebnis liefert. Dieser Umstand legt die Vermutung nahe, dass die DLib für das Training des Detektors die verwendeten Bildsequenzen verwendet. Aus diesem Grund wird der Detektor der DLib nicht weiter bewertet. Dennoch sind die Ergebnisse in den Abbildungen 4.25 und 4.24 enthalten. Man erkennt aber leicht, dass die Kurven für die DLib identisch sind.

Der Detektor von Qu et al. [Qu15a] zeigt sehr gute Ergebnisse. Allerdings sind diese zu erwarten, da der Detektor zur Erstellung der Referenzflächen

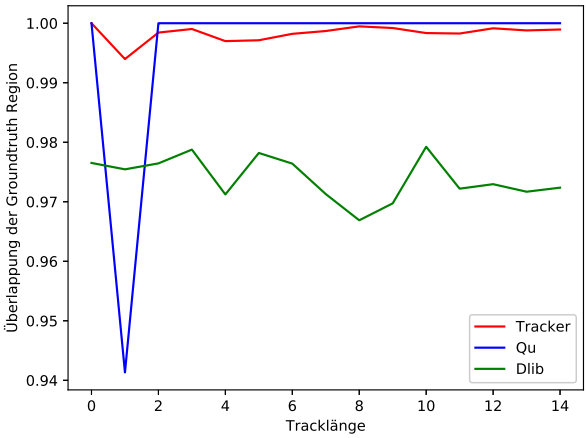


Abbildung 4.24: Verlauf des Überlappungsmaßes der drei Methoden mit einer Fehlmessung im zweiten Zeitschritt des Trackings.

verwendet wurde. In einem weiteren Test wurde geprüft, wie sich der Tracker verhält, wenn eine Messung der Sequenz ausbleibt. Da der Tracker den Detektor von Qu et al. nutzt, fällt die fehlende Messung auch im Verlauf des Landmarkendetektors auf. Der niedrige Wert beschreibt die Überlappung der mittleren Form zur wahren Form. Betrachtet man Abbildung 4.24 genauer, kann jedoch abgeleitet werden, dass die Verwendung eines Trackers den Fehler durch das Ausbleiben der Messung reduziert.

In Abbildung 4.24 ist zu sehen, dass der Landmarkendetektor für alle übrigen Messungen perfekte Ergebnisse liefert. Dadurch kann belegt werden, dass ein qualitativ hochwertiger Algorithmus als Messsystem verwendet wird. Abbildung 4.25 zeigt eine für den Landmarkendetektor von Qu et al. [Qu15a] perfekte Sequenz.

Die dargestellten Verläufe zeigen eine hohe Qualität des verwendeten Landmarkendetektors. Die Experimente zeigten weiter, dass die verwendeten Sequenzen Teil der Trainingsdaten des Detektors der DLib waren. Der Detektor von Qu et al. [Qu15a] ergibt nicht immer die gleichen Ergebnisse, jedoch

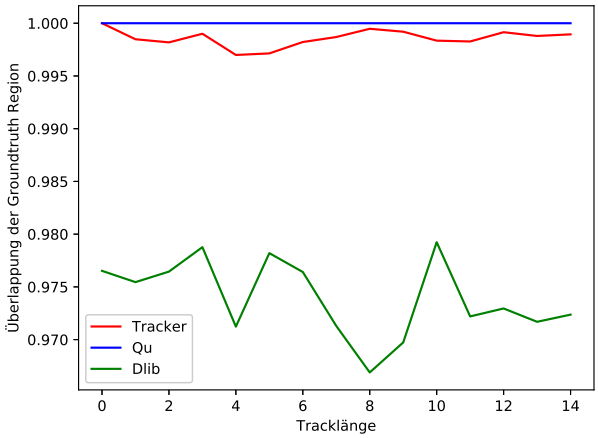


Abbildung 4.25: Verlauf des Überlappungsmaßes der drei Methoden mit einem perfekten Erkennungsverlauf für den Landmarkendetektor von Qu et al. [Qu15a]

sind die Ergebnisse zumeist von sehr hoher Qualität. Sowohl der Landmarkendetektor von Qu et al. [Qu15a], als auch der Detektor der DLib erzielen sehr gute Ergebnisse. Für die weitere Arbeit wird der Detektor von Qu et al. [Qu15a] verwendet.

4.5 Zusammenfassung

In diesem Kapitel wurden verschiedene Möglichkeiten des Trackings eingeführt. In Abschnitt 4.1 wurde ein statischer Fall von Tracking vorgestellt. Hier war es das Ziel, Bild zu Bild Transformationen zu bestimmen, mit denen Objekte über eine Bildsequenz hinweg sicher markiert werden können. Das Tracking ist ein Fall von statischem Tracking, da hier kein Objekt im Bild verfolgt wurde, sondern vielmehr die Parameter zur Bestimmung der Homographie zwischen zwei Bildern. Im Abschnitt 4.2 wurde ein Tracking für Punktziele am Beispiel von Flugzeugtracking vorgestellt. Für das Tracking eines Flugzeugs wurden Messungen eines passiven Systems verwendet. Signale, die vom Flugzeug ausgingen, wurden von am Boden verteilten Basisstationen

empfangen und in Pseudoabstände umgerechnet. Diese Messungen wurden dann mit einem linearen Regressions Kalman Filter zur simultanen Positions- und Offsetschätzung eingesetzt. Abschnitt 4.3 zeigt am Beispiel der Iris des menschlichen Auges wie eine Form dazu verwendet werden kann, ein ausgedehntes Objekt zu tracken. Das Tracking von ausgedehnten Objekten hat Vorteile gegenüber dem Tracking von Punktzielen. Durch die ausgedehnte Beschreibung des Objekts können weitere Informationen über das Objekt extrahiert werden. Im Fall der Iris könnten das zum Beispiel biometrische Informationen sein. Als Messungen für dieses System wurden aus Augenbildern Randpunkte der menschlichen Iris detektiert. Unter Verwendung eines Greedy Association Models konnten die Randpunkte mit dem verwendeten Formmodell in Verbindung gebracht werden. Mit Hilfe des Unscented Kalman Filters war es dann möglich, die Iris-Schätzung durch Filterung zu verbessern. In Abschnitt 4.4.3 wurde das Tracking von ausgedehnten Objekten auf ein Landmarkenmodell für das menschliche Gesicht angewendet. Dieses Landmarkenmodell enthält insgesamt 68 Einzelpunkte, sodass der Zustandsvektor aus genau 136 Elementen besteht. Somit wurde jeder einzelne Punkt für den Systemzustand verwendet. Da kein einfaches parametrisches Modell existiert, das dieses Formmodell erzeugt, konnte keine einfachere Formulierung für den Systemzustand gefunden werden. Für das Tracking der einzelnen Landmarkenpunkte des Modells wurde ein Unscented Kalman Filter eingesetzt. Da kein koordiniertes Bewegungsmodell für das Landmarkenmodell gefunden werden konnte, wurde ein Constant Position Modell eingesetzt. Durch dieses Modell wird jedoch ein Fehler eingeführt, der zu einer Deformation des Landmarkenmodells führt. Aus diesem Grund wird eine Nebenbedingung eingeführt, die eine Rückführung auf das nächste korrekte Landmarkenmodell erzwingt. Dazu wird das Landmarkenmodell zunächst so in Hauptkomponenten zerlegt, dass 98% der beschriebenen Varianz erhalten bleiben. In diesen 98% sind nur korrekte Fälle enthalten und hochfrequentes Rauschen wird entfernt. Diese werden dann rückprojiziert, sodass ein Landmarkenmodell zurückgegeben wird. Zusätzlich wird ein Gating verwendet, falls falsche Messungen zurückgeliefert werden. In Experimenten konnte gezeigt werden, dass die Verwendung eines Trackingalgorithmus insbesondere dann Vorteile gegenüber der rohen Verwendung eines Landmarkendetektors liefert, wenn

es zu Fehlmessungen des Landmarkendetektor kommt. Das ist insbesondere dann von Vorteil, wenn zum Beispiel Emotionen abgeleitet werden sollen.

5 Analyse des Beobachtertrainings

Es gibt viele Anforderungen bei der Beobachtung und Analyse von Szenen. Bei einer hohen Zahl an öffentlichen Kameras kommt es zu einer immer größer werdenden Zahl an Bildaufnahmen, wobei die Anzahl der beobachtenden Personen nicht im gleichen Maß wächst. Somit müssen Beobachter für diese Aufgabe trainiert werden und die Leistung einzelner Beobachter gemessen und über einen Verlauf eines Trainings getrackt werden. Diese Messungen sollten Indikatoren enthalten, über die ein Lernfortschritt festgestellt werden kann.

In zwei Studien wird ein Training vorgestellt, mit dem sich die Weiterentwicklung der Beobachtungsleistung einzelner Probanden bestimmen lässt. Zunächst werden Maßzahlen abgeleitet, die für die Feststellung von Leistungsindikatoren verwendet werden können. Dann werden Trainingsprozeduren entwickelt, mit denen die Indikatoren bestimmt werden können. Anhand von simulierten Bilddaten werden Probanden trainiert und mit vorher-nachher-Tests wird ein möglicher Trainingseffekt untersucht. Die zweite Studie wendet die Ergebnisse der ersten Studie an um zu zeigen, ob die Verwendung eines automatischen Zielerfassungssystems einen Vorteil für die Beobachtungsleistung liefert.

In Abschnitt 5.1 wird untersucht ob der Mensch für das Tracking von Objekten in Videos trainiert werden kann. Abschnitt 5.2 beschreibt den Effekt der durch die Verwendung eines unterstützenden, automatischen Zielerfassungssystem verursacht wird. Die Diskussion der Ergebnisse zur Bewertung der Beobachtungsleistung von menschlichen Beobachtern wird in Abschnitt 5.3 besprochen.

5.1 Nachweis eines Trainingseffekts für menschliche Tracker

Menschen sind von Natur aus in der Lage Objekte zu tracken und zu klassifizieren. In diesem Kapitel wird untersucht, ob sich durch ein Training die Leistung des Trackings und der Klassifikation von Objekten verbessern. Dabei wird das Potential, die eigene Tracking- und Klassifikationsleistung zu steigern, durch Anwendung von Beobachterversuchen analysiert.

Ein System zur automatischen Zielerkennung kann durch Techniken des maschinellen Lernens für diese Aufgabe vorbereitet werden. Der Mensch besitzt jedoch von klein auf die Fähigkeit Objekte zu erkennen, sowie diese in eigener Bewegung zu verfolgen. Um ein technisches System zur Bewältigung einer solchen kognitiven Aufgabe korrekt bewerten zu können, muss auch der Vergleich mit menschlichen Beobachtern durchgeführt werden. Zu diesem Zweck wird in dieser Untersuchung zunächst bewertet, ob Menschen für die Tätigkeit der Erkennung und Verfolgung von speziellen Objekten trainiert werden können. Diese speziellen Objekte sind in diesem Fall Avatare mit Rucksack, die sich zwischen weiteren Avataren ohne Rucksack bewegen. Diese Avatare sind Teil von simulierten Sequenzen, sogenannten Crowd Simulationen. Die Crowd-Simulationen enthalten eine festgelegte Anzahl unterschiedlicher Avatare. Es kann ein fester Anteil von Avataren mit Rucksäcken gewählt werden, die dann zufällig zu verschiedenen Zeitpunkten die Fläche überqueren. Außerdem kann der Betrachtungswinkel auf verschiedene Werte für jede Sequenz festgelegt werden. In Abbildung 5.1 ist ein Ausschnitt einer Sequenz zu sehen. In dieser Szene befinden sich Avatare mit und ohne Rucksack.

Abbildung 5.2 enthält zwei unterschiedliche Teilversuche: Zum einen Videoversuche, in denen Probanden die Aufgabe haben Avatare mit Rucksäcken zu finden und zu markieren, zum anderen ein Training, das Beobachter auf die Detektion von Avataren mit Rucksäcken vorbereitet.

Es wird eine Standardprozedur definiert, mit der der Vorgang der Markierung ermöglicht wird. Den Probanden wird zwei Minuten lang eine Fläche gezeigt,

Abbildung 5.1: Bildausschnitt aus einer Crowd-Simulation mit rucksacktragenden Avataren im Bild.

Abbildung 5.2: Geplanter Verlauf des Versuchs zur Existenzprüfung eines Trainingseffekts.

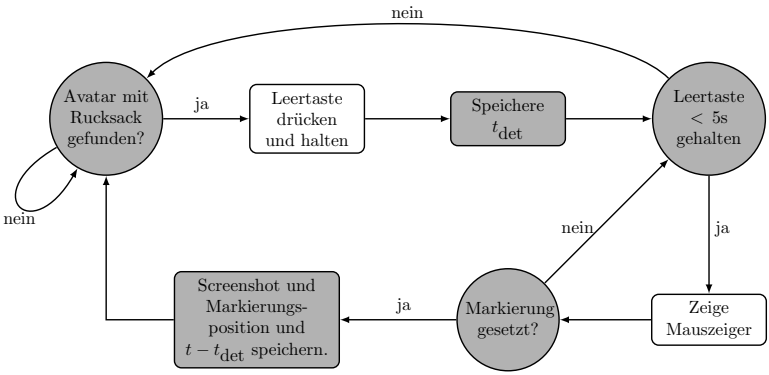


Abbildung 5.3: Verlauf einer Detektion eines Avatars mit Rucksack bei einem Video Versuch.

über die sich insgesamt 150 Avatare bewegen, davon 15 mit Rucksack. Zu jedem Zeitpunkt kann es sein, dass entweder kein Avatar mit Rucksack auf dem Platz ist oder einer oder mehrere Avatare mit Rucksack. Die Markierung eines Avatars mit Rucksack geschieht über Drücken der Leertaste. Mit dem Drücken der Leertaste wird die Detektionsprozedur gestartet und der Zeitpunkt t_{det} gespeichert. Ab diesem Zeitpunkt können die Probanden innerhalb von maximal 5 Sekunden mit der Maus eine Markierung auf den Avatar mit Rucksack setzen. Innerhalb dieser Zeit muss die Leertaste gehalten werden, wodurch auch das Bild des Videos angehalten wird. Durch Loslassen der Leertaste kann die Detektionsprozedur abgebrochen werden und eine Fehldetektion wird gespeichert. Wird während des Haltens der Leertaste ein Avatar markiert, wird zunächst die Dauer der Markierung $t - t_{\text{det}}$ gespeichert. Ein Screenshot wird mit der Markierung an der angeklickten Position angefertigt und die Markierungsposition wird gespeichert. Der Prozess der Markierung eines Avatars mit Rucksack ist in Abbildung 5.3 dargestellt.

Im Training werden den Probanden 176 Bilder in zufälliger Reihenfolge gezeigt. Das Training ist als sogenannte Ja/Nein-Prozedur definiert. Das bedeutet, dass die Probanden bei jedem Bild mit Ja oder Nein antworten müssen. Ein Ja steht dabei für die Zustimmung zu der Frage, ob ein Avatar mit Rucksack im vorliegenden Bild zu finden ist. Für jedes Bild bekommen die Probanden eine Zeit von maximal 10 Sekunden, um zu einer Entscheidung zu kommen.

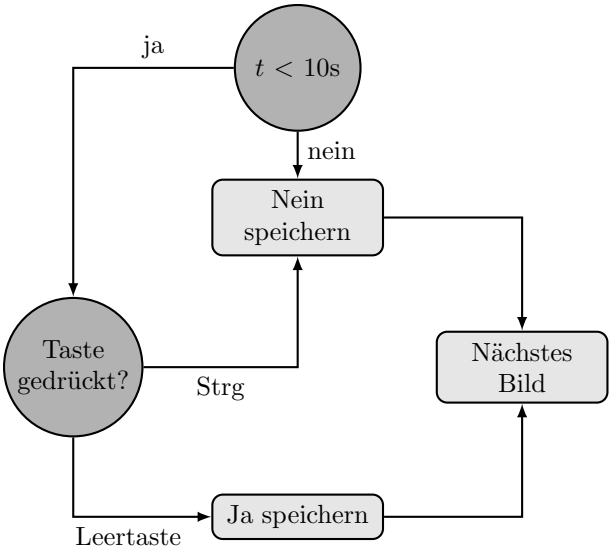


Abbildung 5.4: Ja/Nein-Entscheidungsprozedur für jedes Bild während des Trainings.

Läuft die Zeit ab, wird das Bild automatisch mit Nein bewertet. Es sind zwei Tasten konfiguriert, mit denen sich eine Antwort geben lässt. Mit der Taste *Strg* antworten die Probanden mit Nein und mit der *Leertaste* mit Ja. Das Ja/Nein-Vorgehen für jedes Bild ist in Abbildung 5.4 grafisch dargestellt.

In beiden Prozeduren werden Daten gespeichert. Für das Training wird für jedes Bild die Dauer Δ_t^{det} bis zur Antwort und ja ($x_{\text{dec}} = 1$) oder nein ($x_{\text{dec}} = 0$) gespeichert. Für die Videoversuche werden für jede Markierungsoperation der Zeitpunkt des Drückens auf die Leertaste t_{det} und für den Fall einer Markierung die Position der Markierung $\underline{x}_{\text{det}}$, die Dauer zwischen dem Druck der Leertaste und dem Setzen der Markierung $\Delta_{\text{det}} = t - t_{\text{det}}$ und ein Screenshot gespeichert. Liegt nur t_{det} vor, so wird automatisch eine Fehldetektion erfasst.

Mit diesem Versuch wird untersucht, ob ein Trainingseffekt für diese Beobachtungsaufgabe nachgewiesen werden kann. Es werden drei Haupthypothesen und fünf Nebenhypothesen definiert. Tabelle 5.1 fasst die drei Haupthypothesen zusammen. Die definierten Haupthypothesen in Tabelle 5.1 werden

Tabelle 5.1: Drei Haupthypothesen für den Nachweis eines Trainingseffekts.

Hypothese	Beschreibung
H1	Durch das Training wird die Detektionszeit verringert.
H2	Nach dem Training wird die Anzahl von falschen Detektionen verringert.
H3	Die Anzahl der korrekten Detektionen erhöht sich durch die Teilnahme beim Training.

nach den beiden Hauptversuchen bewertet. Die fünf Nebenhypothesen werden zwischen den Trainingseinheiten ausgewertet. Die Ergebnisse dieser Untersuchung sind in Abschnitt 5.1.1 beschrieben.

Die Nebenhypothesen haben den Zweck, den Einfluss des Trainings zu beurteilen. Hierzu wird die Receiver-Operating-Characteristic (ROC) Analyse verwendet. Die ROC-Analyse erzeugt zunächst die sogenannte ROC-Kurve. In der ROC-Kurve wird die Falsch-Positiv-Rate gegen die Sensitivität aufgetragen. Um die Kurve zu erzeugen, muss ein Parameter variiert werden, sodass die Sensitivität und Falsch-Positiv-Rate sich verändern, solange bis beide bei 100% angelangt sind. Die Kurve kann dazu verwendet werden, um einen optimalen Parameter für das Klassifikationsproblem zu ermitteln. In diesem Fall wird die ROC-Kurve dazu verwendet, um die Hypothesen auszuwerten. In den fünf Hypothesen werden Sensitivität, Spezifität, der Positiv-Prädiktive-Wert (PPV), der Negativ-Prädiktive-Wert (NPV) und der Sensitivitätsindex (d') ausgewertet. Die Sensitivität wird als Verhältnis der korrekten Zuordnungen zu der Gesamtanzahl der Rucksackbilder durch

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5.1}$$

berechnet, wobei TP die korrekten Zuordnungen, FN die nicht zugeordneten Rucksackbilder und TPR die Sensitivität bezeichnet. Zur Berechnung der

Spezifizität müssen die korrekt und falsch zugeordneten Bilder ohne Rucksack gezählt werden. Die Spezifizität FPR ergibt sich durch

$$\mathrm{FPR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}\,, \tag{5.2}$$

wobei TN die korrekt zugeordneten Bilder ohne Rucksack, und FP die falsch zugeordneten Bilder ohne Rucksack bezeichnet. Unter weiterer Verwendung von TP, FN, TN und TP können PPV und NPV durch

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{5.3}$$

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} \tag{5.4}$$

berechnet werden. Der Sensitivitätsindex d' kann berechnet werden, wenn das Signal-zu-Rausch-Verhältnis für das Problem bekannt ist. Das ist in diesem Fall unbekannt. Durch Verwendung der ROC-Statistik kann der Wert approximiert werden. Dann kann durch

$$d' = \sqrt{2} \cdot Z\,(\mathrm{AUC}) \tag{5.5}$$

der Sensitivitätsindex d' berechnet werden, wobei AUC der Area-Under-Curve-Wert der ROC-Kurve und $Z\,(\cdot)$ die Inverse der kumlativen Verteilungsfunktion der Normalverteilung sind. Mit diesen Größen werden die fünf Nebenhypothesen in Tabelle 5.2 definiert.

Im folgenden Abschnitt wird der Versuch zur Untersuchung von unterstützenden Trackingverfahren beschrieben.

Tabelle 5.2: Fünf Nebenhypothesen zur Bewertung der Trainingsveränderungen zwischen jeder Trainingseinheit.

Hypothese	Beschreibung
SH1	Die Sensitivität (TPR) der Detektionsaufgabe verbessert sich.
SH2	Die Spezifität (FPR) der Detektionsaufgabe verbessert sich.
SH3	Der Positiv-Prädiktive-Wert (PPV) verbessert sich durch das Training.
SH4	Der Negativ-Prädiktive-Wert (NPV) verbessert sich durch das Training.
SH5	Die Detektionsleistung, ermittelt durch den Sensitivitätsindex (d'), verbessert sich durch das Training.

Tabelle 5.3: Population des Versuchs zur Prüfung der Existenz eines Trainingseffekts für Beobachtungsaufgaben.

Altersbereich	weiblich	männlich
< 30	3	1
$[30, 40)$	7	0
$[40, 50)$	3	3
$[50, 60)$	2	2
$>= 60$	1	0
Summe	16	6

5.1.1 Resultate der Untersuchung über die Existenz eines Trainingseffekts

Der Versuch zum Trainingseffekt für menschliche Beobachter wurde am Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB) in Ettlingen durchgeführt. Die Teilnehmer dieses Experiments waren Mitarbeiter dieses Forschungsinstituts. Insgesamt haben 22 Probanden teilgenommen. Die Alters- und Geschlechtsverteilung ist in Tabelle 5.3 angegeben.

Tabelle 5.4: Ergebnisse des Basis-Versuchs zur Ermittlung des Ist-Zustandes um einen Basiswert für die Ermittlung der Existenz eines Trainingseffekts zu erlangen. Die Messwerte werden zusätzlich mit Standardabweichung angegeben.

	Messwert	$\sum$	min	x_{med}	max
Korrekte Detektionen	12.09 ± 2.07	266	8	12.50	15
Mittlere Detektionszeit	$10.22\text{ s} \pm 3.76\text{s}$	224.82s	3.61s	10.27s	18.49s
Sensitivität	0.81 ± 0.14	17.73	0.53	0.83	1
Spezifität	1 ± 0	22	1	1	1
PPV	1 ± 0	22	1	1	1
NPV	0.98 ± 0.02	21.54	0.95	0.98	1

Alle Probanden führten das Experiment in einer festgelegten Reihenfolge durch (siehe Abbildung 5.2). Für alle Probanden musste zunächst der Ist-Zustand ermittelt werden. Alle Probanden bekamen ein Video zu sehen, in dem sie die Aufgabe hatten rucksacktragende Avatare aufzuspüren und zu markieren. Die durchschnittliche Zeit von der Detektion bis zur Markierung wurde gemessen. Zusätzlich wurde erfasst, wie viele der Avatare mit Rucksack gefunden und korrekt markiert wurden und wie viele Fehldetektionen es gab. Daraus werden die definierten Kennwerte abgeleitet. Tabelle 5.4 fasst die Ergebnisse des Basisversuchs zusammen.

Tabelle 5.4 zeigt, dass im Schnitt 12.09 ± 2.07 Avatare von insgesamt 15 Avataren mit Rucksack gefunden wurden. Es gab Probanden, die alle Avatare finden konnten. Die mittlere Detektionszeit beläuft sich auf 10.22 ± 3.76 Sekunden. Die Probanden weisen eine hohe, mittlere Sensitivität auf. Die Spezifität und der PPV erzielen im Mittel ein perfektes Ergebnis. Der NPV erzielt ein sehr gutes Ergebnis mit 0.98 ± 0.02. Mit dieser Ausgangsbasis ist eine grundsätzliche Verbesserung nur noch in der Anzahl der korrekten Detektionen und der mittleren Detektionszeit zu erwarten.

Um eine Verbesserung zu erlangen, sollten die Probanden einen Trainingsprozess durchlaufen. Mit einem Tag Abstand wurden die Probanden mit Einzelbildern trainiert. Bei jedem Bild mussten die Probanden sich möglichst schnell

Tabelle 5.5: Ergebnisse der drei Trainingseinheiten zur Verbesserung der Detektionsleistung. Alle Messwerte werden mit Standardabweichung angegeben.

	Messwert	$\sum$	min	x_{med}	max
Sensitivität (T1)	0.82 ± 0.12	18.25	0.58	0.87	0.96
Spezifität (T1)	$\mathbf{0.49 \pm 0.19}$	**10.83**	**0.17**	**0.50**	**0.94**
PPV (T1)	$\mathbf{0.94} \pm 0.02$	**20.61**	**0.91**	**0.94**	**0.99**
NPV (T1)	$0.29 \pm \mathbf{0.10}$	6.28	**0.15**	0.30	0.46
d' (T1)	1.04 ± 0.37	**22.92**	0.36	1.05	**1.80**
Sensitivität (T2)	0.87 ± 0.11	19.11	0.62	0.92	**1.00**
Spezifität (T2)	0.42 ± 0.23	9.28	0.00	0.42	0.83
PPV (T2)	0.93 ± 0.02	20.50	0.90	0.93	0.97
NPV (T2)	$\mathbf{0.32} \pm 0.18$	**7.07**	0	0.29	**1.00**
d' (T2)	$\mathbf{1.06 \pm 0.21}$	21.28	**0.67**	**1.08**	1.41
Sensitivität (T3)	$\mathbf{0.89 \pm 0.10}$	**19.60**	**0.68**	**0.93**	0.99
Spezifität (T3)	0.34 ± 0.19	7.39	0.06	0.28	0.72
PPV (T3)	0.92 ± 0.02	20.31	0.90	0.92	0.97
NPV (T3)	0.31 ± 0.12	6.87	**0.15**	**0.32**	0.63
d' (T3)	0.90 ± 0.31	19.82	0.26	0.93	1.50

entscheiden, ob ein Avatar mit Rucksack im Bild war. Dazu mussten die Probanden bei der Entscheidung Nein die linke Strg-Taste drücken, oder die Leertaste bei Ja. Basierend auf den Antworten werden zur Ermittlung der Detektionsleistung verschiedene Werte gemessen: die Sensitivität, Spezifität, PPV, NPV und d'. Die Werte wurden für jeden Probanden ermittelt und es wurden Mittelwert mit Standardabweichung, die Summe der Messwerte, Minimal- und Maximalwert, sowie der Median der gemessenen Werte berechnet. Die Ergebnisse der Trainingseinheiten sind in Tabelle 5.5 aufgeführt.

Betrachtet man Tabelle 5.5 genauer, konnte ausschließlich die Sensitivität über die Trainingseinheiten gesteigert werden. Die Spezifität und PPV nehmen kontinuierlich zwischen jeder Trainingseinheit ab. Die Werte NPV und d' verbessern sich zwischen den Trainingseinheiten T1 und T2 und verschlechtern sich nach dem dritten Training. Der Wert d' ist beim dritten Training minimal. Für eine weitere Auswertung werden für das Training fünf Nebenhypothesen bewertet. Unter Betrachtung der Nebenhypothesen (siehe Tabelle 5.2) zeigt sich, dass die Nebenhypothese SH1 bestätigt werden kann, da die Sensitivität sich im Verlauf des Trainings verbessert hat. Die Nebenhypothesen SH2 und SH3 müssen verworfen werden, da eine kontinuierliche Verschlechterung von Spezifität und PPV besteht. Die

Tabelle 5.6: Ergebnisse des finalen Versuchs zur Ermittlung des Zustands nach Training zur Ermittlung der Existenz eines Trainingseffekts. Alle Messwerte werden mit Standardabweichung angegeben.

	Messwert	$\sum$	min	x_{med}	max
Korrekte Detektionen	10.64 ± 1.79	234	7	10	14
Mittlere Detektionszeit	$10.75\,\text{s} \pm 5.00\text{s}$	236.47s	2.84s	9.95s	31.94s
Sensitivität	0.71 ± 0.12	15.6	0.47	0.67	0.93
Spezifität	1 ± 0	22	1	1	1
PPV	1 ± 0	22	1	1	1
NPV	0.97 ± 0.01	21.32	0.94	0.96	0.99

Nebenhypothesen SH4 und SH5 werden zwischen T1 und T2 bestätigt und müssen unter Einbezug von T3 verworfen werden. Somit werden vier der fünf Hypothesen verworfen. Für das Training kann keine Verbesserung der Detektionsleistung nachgewiesen werden.

Zum Beweis eines Trainingseffekts muss noch der finale Videoversuch ausgewertet werden. Für den zweiten Versuch werden analog des ersten Videoversuchs die Anzahl der korrekten Detektionen, die mittlere Detektionszeit, Sensitivität, Spezifität, PPV und NPV für jeden Probanden im Videoversuch ermittelt. Die Messwerte aller Ergebnisse werden statistisch ausgewertet. Die ausgewerteten Ergebnisse sind in Tabelle 5.6 dargestellt.

Um die Haupthypothesen zu bestätigen, müssen die Ergebnisse des Basisversuchs mit den finalen Versuchsergebnissen verglichen werden. Für die Haupthypothese H1 werden die mittleren Detektionszeiten miteinander verglichen. Es zeigt sich eine geringfügige mittlere Zunahme, wobei eine stärkere Schwankung gemessen wird. Das Minimum der gemessenen mittleren Zeit und der Median verringern sich zwischen beiden Versuchen. Die maximale Zeit ist deutlich gestiegen. Es wurde keine signifikante Verbesserung erreicht, weswegen Hypothese H1 verworfen werden muss. Hypothese H2 kann für diesen Versuch nicht ausgewertet werden, da keine Falschdetektionen verzeichnet wurden. Hypothese H3 muss verworfen werden, da keine Erhöhung der korrekten Detektionen erfasst wurde.

Für diese Untersuchung kann weder ein eindeutiger Hinweis für die Existenz eines Trainingseffekts nachgewiesen, noch ausgeschlossen werden. Es muss die geringe Anzahl von 22 Probanden berücksichtigt werden, die für statistisch valide Ergebnisse weitaus größer sein muss. Für diesen Versuch konnten nicht mehr Probanden akquiriert werden. Demnach bleibt die Frage nach einer statistisch abgesicherten Antwort auf die Frage nach der Existenz eines Trainingseffekts für diese Aufgabe offen.

5.2 Überprüfung des Effekts von unterstützenden Trackingsystemen

Mit einer steigenden Anzahl an Kamerasystemen an öffentlichen Plätzen erhöht sich der Überwachungsaufwand. Beobachter bekommen häufig die Aufgabe, verdächtige Personen in einer Menschenmenge zu erkennen. In unterschiedlichen Bereichen, in denen Menschen beobachtet werden, wird darüber nachgedacht, Methoden aus dem maschinellen Lernen zu verwenden, um den Beobachtern die Aufgabe zu erleichtern. Zu diesen Verfahren gehören Zieldetektionssysteme in Kombination mit speziellen Trackingalgorithmen. Solche Algorithmen erzeugen durch ihr Zielerkennungssystem hypothetische Ziele. Diese werden durch einen Rahmen markiert und Beobachter können gezielt diese Objekte absuchen, um gesuchte Zielobjekte zu finden. Dafür darf die Anzahl der dargestellten Rahmen nicht zu groß werden. Sonst muss ein Beobachter zu viele Objekte betrachten und verliert den Überblick. Bei einer zu geringen Anzahl an Rahmen würde der Fokus von nicht erkannten Zielen ablenken. In beiden Fällen würden möglicherweise zu wenige Zielobjekte gefunden. In der Praxis hängt die optimale Anzahl an Rahmen stark von der Aufgabe und dem aktuellen Geschehen ab.

In diesem Abschnitt wird untersucht, ob ein automatisches Zielerfassungssystem einen positiven Einfluss auf die Detektionsleistung von Probanden ausübt. Es werden Crowd-Simulationsvideos verwendet analog zu Abschnitt 5.1. In diesem Versuch wurden die Videos mit der Software Maya Autodesk

Tabelle 5.7: Überblick über die verwendeten Sequenzen für die Untersuchung des Einflusses von Markierungen auf die Detektionsleistung von Probanden. Im Fall von 20 Markierungen bei 150 Avataren bedeutet der Eintrag, dass sechs der 15 Avatare markiert wurden und ein Blickwinkel von 72° verwendet wurde.

Markierungen	100 Avatare	150 Avatare	200 Avatare
0	0/10 (0°)	0/15 (288°)	0/20 (144°)
5	1/10 (72°)	2/15 (144°)	2/20 (216°)
10	2/10 (144°)	3/15 (0°)	4/20 (288°)
20	4/10 (216°)	6/15 (72°)	8/20 (0°)
40	8/10 (288°)	12/15 (216°)	16/20 (72°)

2015 und Golem Crowd 2015 erstellt. Die Markierungen wurden mit Adobe After Effects CC 2015 manuell gesetzt. Eine automatische Trackingsoftware sorgte für die Verfolgung der markierten Avatare. Es wurden insgesamt 15 Videos verwendet. Bei diesen Videos wurden der Blickwinkel auf den simulierten Platz, die Zahl der Avatare und die Zahl der angezeigten Markierungen variiert. Es wurden fünf verschiedene Blickwinkel gewählt: $\{0°, 72°, 144°, 216°, 288°\}$. In den Videos können drei unterschiedlich große Anzahlen von Avataren enthalten sein: $\{100, 150, 200\}$. Die Anzahl der angezeigten Rahmen konnte zwischen keinem und 40 Markierungen variieren. Die tatsächliche Zusammensetzung der Parameter in den 15 Videosequenzen ist in Tabelle 5.7 dargestellt. In Abbildung 5.5 ist ein Einzelbild mit 200 Avataren und 40 Markierungen dargestellt.

Um den Einfluss der Markierungen auf die Arbeitslast der Probanden zu ermitteln, wird eine zweite Aufgabe für die Probanden in den Versuch integriert. In zufälligen Intervallen wurden den Probanden akustische Signale vorgespielt, während sie die Videoversuche durchführten. Die Probanden mussten auf dieses akustische Signal reagieren, indem sie das akustische Signal durch einen Tastendruck quittierten. Ausgehend von der Signalquittierung wurde die Reaktionszeit auf den Stimulus gespeichert. Für den Videoversuch wurden Detektionszeiten und -raten der gesuchten Avatare gespeichert. Das Ziel des Versuchs war es herauszufinden, ob eine steigende Avataranzahl, sowie eine steigende Markierungsanzahl einen stärkeren, negativen Effekt auf die Detektionsleistung der Probanden ausübt. Dazu werden bezüglich der steigenden Avataranzahl und der steigenden Markierungsanzahl unterschiedliche Hypothesen generiert. Die erzeugten Hypothesen beziehen die Detektionszeiten,

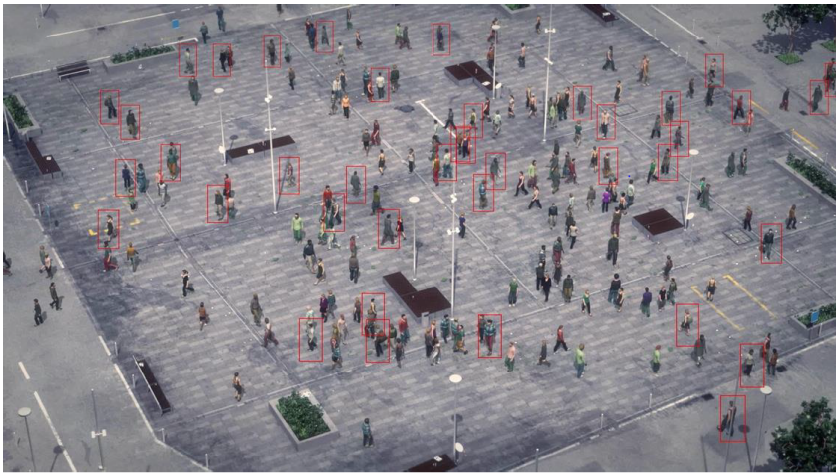


Abbildung 5.5: Beispielszene aus einer Sequenz mit 200 Avataren und 40 Markierungen.

Tabelle 5.8: Getestete Hypothesen für die Studie zur Evaluierung des Einflusses der Bildanalysealgorithmenqualität auf die Detektionsleistung von Probanden.

Hypothese	Erklärung
H1	Die Detektionszeit (DT_{TG}) erhöht sich mit steigender Avataranzahl.
H2	Die Detektionsrate (DR_{TG}) verringert sich mit steigender Avataranzahl.
H3	Die Stimulus-Reaktionszeit (RT_{ST}) steigt mit steigender Avataranzahl.
H4	Die Stimulus-Trefferrate (HR_{ST}) sinkt mit steigender Avataranzahl.
H5	Die Detektionszeit (DT_{TG}) steigt mit steigender Markierungsanzahl.
H6	Die Detektionsrate (DR_{TG}) sinkt mit steigender Markierungsanzahl.
H7	Die Stimulus-Reaktionszeit (RT_{ST}) steigt mit steigender Markierungsanzahl.
H8	Die Stimulus-Trefferrate (HR_{ST}) sinkt mit steigender Markierungsanzahl.

Detektionsraten, Stimulus-Reaktionszeiten und Stimulus-Trefferraten ein. Tabelle 5.8 fasst die Hypothesen für den Versuch zusammen.

Tabelle 5.9: Ergebnisse der Detektionszeiten und -raten, sowie der Reaktionszeiten und Trefferraten für die akustischen Stimuli bei einer steigenden Anzahl von Avataren. Die Vergleichswerte sind als Median der gemessenen Verteilungen gegeben.

	Anzahl der Avatare			
	A100	**A150**	**A200**	p-Wert
DT_{TG} **(s)**	5.21	6.63	7.13	0.06^{*}
DR_{TG}	0.54	0.43	0.40	$< 0.000^{*}$
RT_{ST} **(ms)**	544	563	576	0.482
HR_{ST}	0.97	0.96	0.97	0.74

5.2.1 Resultate der Untersuchung über den Einfluss von Bildanalysesoftware auf die Detektionsleistung

Der Versuch zur Bewertung des Einflusses von Bildanalysesoftware zur Unterstützung von Beobachtern wurde am Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB) in Ettlingen, Deutschland durchgeführt. Für den Versuch konnten 26 Freiwillige akquiriert werden, 7 weibliche Beobachter und 19 männliche Beobachter. Die Versuche wurden an zwei aufeinanderfolgenden Tagen durchgeführt. Die Sequenzen wurden in drei unterschiedlichen Permutationen aneinandergereiht, sodass immer zwei nebeneinandersitzende Probanden unterschiedliche Reihenfolgen der Sequenzen sahen. Während der Betrachtung der Videos mussten die Probanden auf zufällig auftretende akustische Signale durch das Drücken einer Taste reagieren.

Zur Evaluation der Ergebnisse wurden Methoden der deskriptiven Statistik und der Inferenzstatistik eingesetzt. Die Daten können nicht als statistisch unabhängig angenommen werden. Zusätzlich wurden die Verteilungen der gemessenen Detektionszeiten und der Reaktionszeiten auf den akustischen Stimulus untersucht. Diese Werte waren nicht normalverteilt. Aufgrund dieser Beobachtungen wurde auf nichtparametrische, statistische Tests zurückgegriffen. In den Versuchen werden wiederholte Messungen miteinander verglichen. Der Friedman-Test wurde mit einem Signifikanzniveau $p < 0.1$ verwendet. Mit der Software PSPP 0.10.4 wurden die statistischen Analysen ausgeführt. Die Analyse wurde bezüglich der steigenden Avataranzahl untersucht. Die Ergebnisse dieser Untersuchung sind in Tabelle 5.9 zusammengefasst.

Die Untersuchung der Detektionszeit DT_{TG} in Tabelle 5.9 zeigt eine Zunahme bei Vergrößerung der Avataranzahl. Der resultierende p-Wert liegt unter dem

Signifikanzniveau; daher ist diese Steigerung der Detektionszeit als signifikant zu bezeichnen. Damit kann die Hypothese H1 bestätigt werden. Ebenso verringert sich die Detektionsrate DR_{TG} bei steigender Avataranzahl in signifikanter Weise, wodurch auch Hypothese H2 bestätigt wird. Wie erwartet, steigerte sich auch die Reaktionszeit auf den sekundären Stimulus RT_{ST}. Allerdings liegt der p-Wert deutlich über dem Signifikanzniveau, wodurch Hypothese H3 verworfen werden muss. Die erwartete Verringerung der Trefferrate konnte nicht festgestellt werden, was zum Verwerfen von Hypothese H4 führt. Die verworfenen Hypothesen H3 und H4 deuten darauf hin, dass sich durch Erhöhen der Anzahl an Avataren die Arbeitslast der Probanden sich zwar objektiv erhöht hat, hieraus allerdings keine Überforderung der Probanden resultiert ist.

Ein weiterer Aspekt dieser Untersuchung ist der Effekt von computergesteuerten Markierungsvorschlägen durch ein Computersystem. Die Anzahl der vorgeschlagenen Markierungen wurde in den einzelnen Videos variiert. Die Anzahlen wurden in fünf unterschiedliche Stufen unterteilt: In keine Markierung M0, fünf Markierugen M5, zehn Markierungen M10, zwanzig Markierungen M20 und vierzig Markierungen M40. Unter Berücksichtigung der Markierungsanzahl werden die Hypothesen H5 bis H8 untersucht mit dem Ziel die Veränderung der Detektionsleistung unter den unterschiedlichen Markierungsanzahlen, sowie die Beeinflussung der Arbeitslast durch die Anzahl der Markierungen zu untersuchen. Die Ergebnisse sind in Tabelle 5.10 aufgetragen. Die Tabelle ist in vier Abschnitte unterteilt. Es gibt je einen Abschnitt für die Detektionszeit DT_{TG}, die Detektionsrate DR_{TG}, die Reaktionszeit auf den Stimulus RT_{ST} und die Trefferrate auf die Stimuli HR_{ST}.

Die Ergebnisse sind einzeln für jede Avataranzahl aufgeführt und werden zu einem Durchschnittswert für die Markierungsanzahlen zusammengefasst. Die Detektionszeit zeigt keine eindeutige Steigerung in Abhängigkeit der steigenden Markierungsanzahl. Vielmehr ist ein u-förmiger Zusammenhang zu beobachten. Dieser Zusammenhang ist in Abbildung 5.6 dargestellt.

Die u-Form in Abbildung 5.6 zeigt ein Minimum der Detektionszeit für eine Markierungsanzahl von 10 Markierungen im Bild. Dies ergibt eine Widerlegung der Hypothese H5. Dieses Minimum ist unabhängig von der Anzahl der

Tabelle 5.10: Ergebnisse der Auswertung der Hypothesen H5 bis H8 bezogen auf eine steigende Anzahl an Markierungen in den Videos. Alle Werte sind als Median der gemessenen Werte gegeben.

		Anzahl Avatare			
		A100	**A150**	**A200**	Mittelwert
	M0	6.10	6.26	7.24	6.61
	M5	5.48	5.74	8.00	6.35
DT_{TG} (**s**)	**M10**	4.39	5.83	5.71	5.29
	M20	6.96	6.74	7.12	7.03
	M40	4.87	7.21	5.37	5.99
	p-Wert	0.03*	0.15	0.37	0.61
	M0	0.60	0.47	0.40	0.47
	M5	0.60	0.40	0.35	0.45
DR_{TG}	**M10**	0.50	0.40	0.40	0.40
	M20	0.50	0.40	0.40	0.45
	M40	0.60	0.50	0.50	0.50
	p-Wert	0.03*	0.21	0.002*	$< 0.000^*$
	M0	543	575	537	544
	M5	531	536	562	543
RT_{ST} (**ms**)	**M10**	521	556	526	526
	M20	581	556	550	558
	M40	515	578	589	564
	p-Wert	0.25	0.59	0.84	0.35
	M0	1.00	1.00	1.00	1.00
	M5	1.00	0.94	1.00	1.00
HR_{ST}	**M10**	1.00	1.00	1.00	1.00
	M20	1.00	1.00	1.00	1.00
	M40	1.00	1.00	1.00	1.00
	p-Wert	0.54	0.55	0.75	0.89

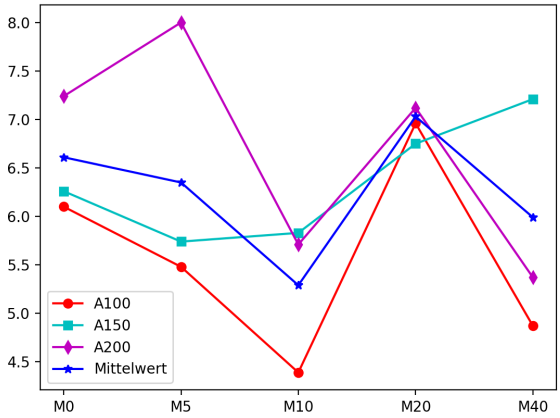


Abbildung 5.6: Entwicklung des Wertes DT_{TG} für verschiedene Markierungsstufen und die verschiedenen Avataranzahlen, sowie den mittleren Verlauf über allen Avataranzahlen.

Avatare und erscheint stabil. Allerdings ist die Anzahl der Probanden sehr gering, sodass hier nur von einer Tendenz gesprochen werden kann.

Die Detektionsrate in Abhängigkeit der Markierungsanzahl DR_{TG} zeigt eine leichte u-Form für jede Avataranzahl. Die Unterschiede werden insgesamt als signifikant angezeigt. Es zeigt sich ein Abfall der Detektionsrate mit steigender Avataranzahl. Die Hypothese H6 ist aufgrund der u-Form zu verwerfen. Die Reaktionszeit RT_{ST} zeigt einen u-förmigen Verlauf über der Markierungsanzahl. Dieser Verlauf ist unabhängig von der Avataranzahl. Somit kann H7 verworfen werden, da sich keine kontinuierliche Steigerung der Reaktionszeit auf den externen Stimulus zeigt. Hier zeigt sich ein Optimum bezüglich der Markierungsanzahl. Das Optimum stellt sich zwischen 10 und 20 Markierungen ein. Die letzte Hypothese kann verworfen werden, da die Detektionsrate der Stimuli konstant bei 1 verbleibt. Ein Ausreißer wurde aufgezeichnet, bei dem nur 94 Prozent der Stimuli quittiert wurden. Das ermittelte Optimum bei 10 bis 20 Markierungen deckt sich mit einer Studie von Huber et al. [Hub15].

Nur zwei der aufgestellten Hypothesen konnten bestätigt werden H1 und H2. Alle weiteren Hypothesen mussten verworfen werden. Es zeigte sich bei der Untersuchung der Hypothesen H5 bis H8 eine nützliche Erkenntnis. Die Anzahl der Markierungen zeigte einen Einfluss auf das Detektionsvermögen der Probanden. Eine Anzahl von 10 bis 20 Markierungen hatte einen positiven Einfluss auf die Detektionszeiten und Detektionsraten für die Zielobjekte und auch die Reaktionszeit des sekundären Stimulus. Der sekundäre Stimulus wurde gemessen, um den Arbeitsaufwand der Probanden nachzuverfolgen. Somit konnte eine Tendenz nachgewiesen werden, die ein Minimum bei einer Arbeitslast von 10 bis 20 Markierungen bedeutet. Die Untersuchung der algorithmischen Unterstützung bei Beobachtungsaufgaben zeigt einen positiven Einfluss auf die Detektionsleistung bei moderater Unterstützung durch Markierungen. Menschliche Beobachter können gezielt durch Algorithmen unterstützt werden, ohne zu einem Störfaktor zu werden. Es ist darauf zu achten, dass die Genauigkeit des Algorithmus hinreichend hoch ist.

5.3 Zusammenfassung

In diesem Kapitel erfolgte eine Untersuchung der menschlichen Leistungsfähigkeit bei Beobachtungsaufgaben. Im ersten Versuch wurde ein möglicher Trainingseffekt durch das vorherige Zeigen von Einzelbildern untersucht. Mit den Einzelbildern wurde wiederholt ein Ja-Nein-Experiment durchgeführt, um herauszufinden, ob die Probanden eine verbesserte Detektionsleistung über die drei Trainingseinheiten entwickeln. Außerdem wurde ein Vorher-Nachher-Experiment durchgeführt, bei dem die Probanden in wiederholten Videosequenzen Avatare mit Rucksack entdecken und markieren sollten. In diesem Experiment konnte kein eindeutiger Hinweis auf einen Trainingseffekt nachgewiesen werden.

In einem zweiten Versuch wurde untersucht, ob der Einsatz von unterstützenden Algorithmen einen positiven Einfluss auf die Beobachtungsleistung ausübt. Es wurde festgestellt, dass eine Anzahl von 10 bis 20 Markierungen hilfreich für Beobachter sein kann. Bei diesen Anzahlen konnte ein positiver Effekt auf die gemessenen Kennzahlen verzeichnet werden.

In beiden Versuchen konnte nur auf eine geringe Anzahl an Probanden zurückgegriffen werden, sodass möglicherweise dennoch ein Trainingseffekt vorhanden sein könnte. Daher ist eine erneute Überprüfung der Ergebnisse mit ausgebildeten Videobeobachtern ratsam.

Der im zweiten Versuch festgestellte positive Effekt bei 10 bis 20 Markierungen auf die Detektionsleistung kann aufgrund der geringen Beobachterzahl nur als Tendenz bewertet werden. Dieser Effekt ist erklärbar, da der Abfall der Kennzahlen bei einer größeren Anzahl von Markierungen auf eine Übereizung hindeutet. Durch eine gezielte Untersuchung mit einer feineren Abstufung der Markierungsanzahl könnte hier ein Nachweis gelingen.

6 Erkennung von emotionalen Gesichtsausdrücken

Diverse Emotionen können einen signifikanten Einfluss auf die menschliche Leistungsfähigkeit oder Motivation ausüben. In der Kommunikation, insbesondere der nonverbalen, spielen Emotionen eine sehr wichtige Rolle. Nonverbale Kommunikation findet statt durch Gestik und Körperhaltung sowie vor allem durch den Gesichtsausdruck des Gegenübers. Denn dieser liefert eine sichtbare Darstellung des aktuellen emotionalen Zustands und bietet somit dem Kommunikationspartner eine erweiterte Interpretationsmöglichkeit. Dem Menschen fällt es leicht, einem Gesichtsausdruck die korrekte emotionale Bedeutung zuzuordnen. Es gibt eine Vielzahl unterschiedlicher Gesichtsausdrücke, die sich in spezifische Klassen einteilen lassen. In der Erkennung von emotionalen Gesichtsausdrücken werden die sechs bekannten Basisemotionen nach Ekman [Ekm99] verwendet: *Wut* (A), *Ekel* (D), *Furcht* (F), *Freude* (H), *Traurigkeit* (S) und *Überraschung* (U). Für diese sechs Emotionsklassen lassen sich exakte Entsprechungen in den Gesichtsausdrücken finden. Abbildung 6.1 zeigt unterschiedliche emotionale Gesichtsausdrücke.

Für ein automatisches System ist der Einsatz eines Klassifikators eine gute Wahl. Die sechs unterschiedlichen Klassen für die Emotionen können als diskrete Klassen aufgefasst werden.

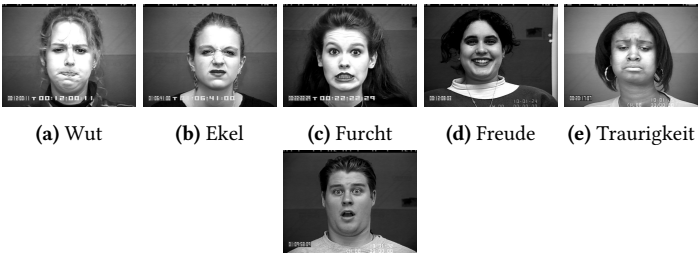
Zur Emotionserkennung muss zunächst eine auf Merkmalen basierte Beschreibung der Gesichtsausdrücke gefunden werden. Diese Beschreibung muss eine hinreichende Unterscheidung der Gesichtsausdrücke nach den sechs Klassen zulassen. Das ist die Grundvoraussetzung für die Nutzung eines Algorithmus zur automatischen Klassifikation der Gesichtsausdrücke.

Die hohe Varianz an Gesichtern stellt eine Herausforderung für die Unterscheidung der Gesichtsausdrücke dar. Zur Veranschaulichung der Varianz emotionaler Gesichtsausdrücke zeigt Abbildung 6.2 acht unterschiedliche Gesichter mit Gesichtsausdrücken der Emotion Freude (H) aus dem Cohn-Kanade+ Datensatz (CK) von Lucey et al. [Luc10a].

Die Bilder zeigen eine hohe Variabilität zwischen den Gesichtern, die sich auch in den zugehörigen Landmarken widerspiegelt. Verwendet man die Techniken aus Kapitel 4: die Extraktion der Landmarken und die Procrustes Analyse, so können die resultierenden Landmarken übereinander gelegt werden. Abbildung 6.3 zeigt überlagerte Landmarken der acht Gesichter aus Abbildung 6.2.

Abbildung 6.4 fasst die Problemstellung der Klassifikation eines emotionalen Gesichtsausdrucks graphisch zusammen.

In den folgenden Abschnitten werden Merkmale abgeleitet, die zur Klassifikation der Gesichtsausdrücke verwendet werden.



(a) Wut **(b)** Ekel **(c)** Furcht **(d)** Freude **(e)** Traurigkeit

(f) Überraschung

Abbildung 6.1: Sechs Beispielbilder mit Gesichtsausdrücken zu den sechs Basisemotionen aus der CK Datenbank. Die Gesichter gehören zu den Probanden S052-A (a), S055-D (b), S074-F (c), S124-H (d), S125-S (e) und S132-U (f) ©Jeffrey Cohn.

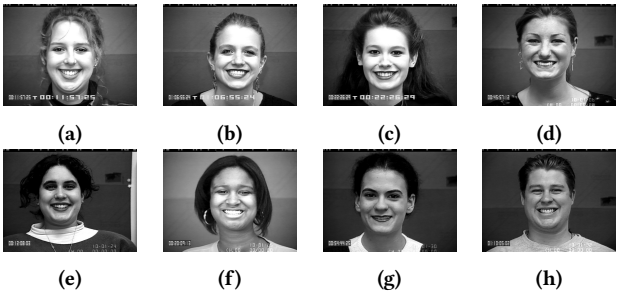


Abbildung 6.2: Acht Beispielbilder mit Gesichtsausdrücken der Emotion H aus der CK Datenbank. Die Gesichter gehören zu den Probanden S052 (a), S055 (b), S074 (c), S106 (d), S124 (e), S125 (f), S130 (g) und S132 (h) ©Jeffrey Cohn.

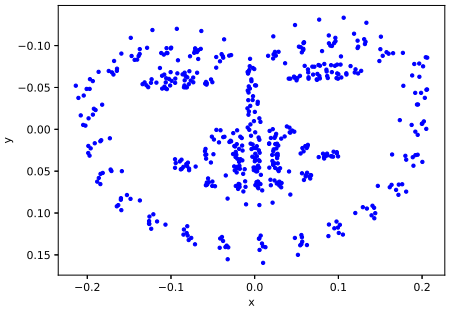


Abbildung 6.3: Mittels Procrustes Analyse übereinandergelegte Landmarken von acht Beispielbilder aus der CK Datenbank mit Gesichtsausdruck der Emotion Freude (H).

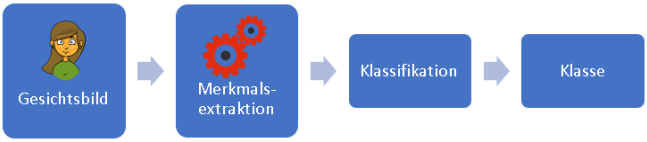


Abbildung 6.4: Prozesskette für die Verarbeitung von Gesichtsbildern in einem Klassifikationsprozess, um die emotionale Klasse dem Bild zuzuordnen.

6.1 Klassifikation von Gesichtsausdrücken

Die Verwendung von Grauwertmustern wie in LBP oder LQP enthält wenig strukturelle Information über das Gesicht und die zugrundeliegenden Gesichtsausdrücke. In dieser Arbeit werden Merkmale entwickelt, die die inhärente Struktur des Gesichtsausdrucks verwenden. Mit der inhärenten Struktur des Gesichtsausdrucks ist grob die Form des Gesichtsausdrucks gemeint. Eine gute Repräsentation dieser Struktur ist durch sogenannte Gesichtslandmarken gegeben. Gesichtslandmarken bieten eine Repräsentation des Gesichtsausdrucks durch eine feste Anzahl an Punkten. Zu diesem Zweck existieren verschiedene Landmarkenmodelle. In dieser Arbeit wird das Landmarkenmodell verwendet, dass auch in der Landmarken-Anpassung von Qu et al. [Qu15a] Verwendung findet. Das Landmarkenmodell umfasst insgesamt 68 einzelne Punkte, wobei zwischen Eckmerkmalen und interpolierten Zwischenpunkten zu unterscheiden ist. In Cootes et al. [Coo00] ist dieser Umstand näher beschrieben. Eckmerkmale sind *echte* Landmarken, die durch prägnante Strukturen des Gesichts gegeben sind, wie zum Beispiel der Mundwinkel oder die Augenwinkel. Zwischenpunkte werden entlang einer Trajektorie zwischen den echten Landmarken interpoliert. In Abschnitt 3.1 wurde das Landmarkenmodell bereits eingeführt.

Ausgehend von diesem Landmarkenmodell werden Merkmale zur Beschreibung des emotionalen Gesichtsausdrucks extrahiert. Die Merkmale beinhalten Informationen über Winkel, sowie über Größen. Die Winkelinformationen werden mittels Schnitt von Geraden ermittelt, die auf Basis der Landmarken konstruiert werden. Die Größeninformation wird mittels Ellipsen approximiert, die anhand von einer Menge von Landmarken interpoliert werden. Dieser Merkmalssatz wird als Angle- and Size Feature Set (ASF) [Dun18a] bezeichnet. Die Ableitung der Winkelinformationen, sowie die Entscheidung für spezielle Winkel wird im folgenden Abschnitt beschrieben. Weiter werden die Größeninformationen mit dem Verhältnis der Ellipsenhalbachsen bestimmt. Die Herleitung dieser Größeninformationen wird im übernächsten Abschnitt betrachtet.

6.1.1 Extraktion von Winkelinformationen

Es werden Geraden aus Punktpaaren des Landmarkenmodells gebildet. Diese Geraden schneiden sich und bilden Winkelinformationen in den Schnittpunkten. Eine Auswahl an Geraden und Schnitten zwischen den Geraden muss getroffen werden, die eine hinreichend große Varianz zwischen den verschiedenen zu beschreibenden Gesichtsausdrücken besitzen. Zur Unterscheidung dieser sieben Ausdrücke werden zunächst sieben Abbildungen der Landmarken eines Subjekts aus der CK Datenbank in Abbildung 6.5 dargestellt. Die Landmarken sind mittels Procrustes Analyse translatiert, skaliert und rotiert worden.

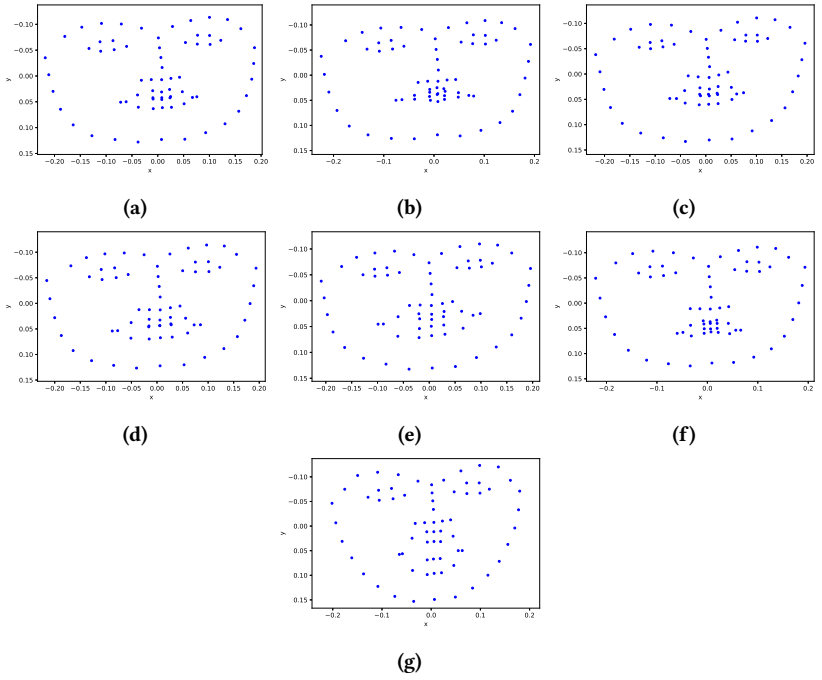


Abbildung 6.5: Landmarken für die sieben emotionalen Klassen (a) Neutral, (b) Ärger, (c) Ekel, (d) Furcht, (e) Freude, (f) Trauer und (g) Überraschung.

Beim Vergleich der Landmarken zwischen den einzelnen Emotionsklassen sind Unterschiede erkennbar. Ein direkter Vergleich der Landmarken ist möglich und wurde in der Literatur bereits durchgeführt. In dieser Arbeit wird der Ansatz verfolgt, die Unterschiede der Landmarken weiter hervorzuheben. Hierzu werden Merkmale entwickelt, die eine weitergehende Unterscheidung und eine gute Grundlage für die Diskriminierung der Klassen in einem Klassifikator bilden. Um weitere Merkmale zu extrahieren, müssen zusammenhängende Informationen herausgestellt werden. Der emotionale Gesichtsausdruck besitzt verschiedene Merkmalsmoden: die Augen spiegeln einen gewissen Anteil des Ausdrucks wieder, ebenso spielt der Mund eine große Rolle bei einem Ausdruck. Die Nase und der Gesichtsrand sind nur bezüglich ihrer Position gegenüber den anderen beiden Bereichen für die visuelle Erkennung des Ausdrucks wichtig. Aus diesem Grund werden die Augenregionen betrachtet. Die Öffnung der Augen unterscheidet sich zwischen den einzelnen Emotionsklassen. Daher werden Winkelinformationen bezogen auf die Öffnung der Augen und die Relation zwischen den Augen und Augenbrauen als Merkmale verwendet.

Um Winkelinformationen zu extrahieren werden Punkte paarweise zusammengefasst und durch Geraden verbunden. Als Beispiel kann der äußere Eckpunkt als Ausgangspunkt für zwei Punktpaare gesehen werden. Die Nachbarpunkte des oberen und unteren Lids bilden jeweils den zweiten Referenzpunkt für eine der beiden Geraden. Der Winkel zwischen diesen beiden Geraden approximiert den Öffnungswinkel zwischen den Augenlidern. Die Geraden sind so gewählt, dass sie sich schneiden. Der Winkel wird mittels

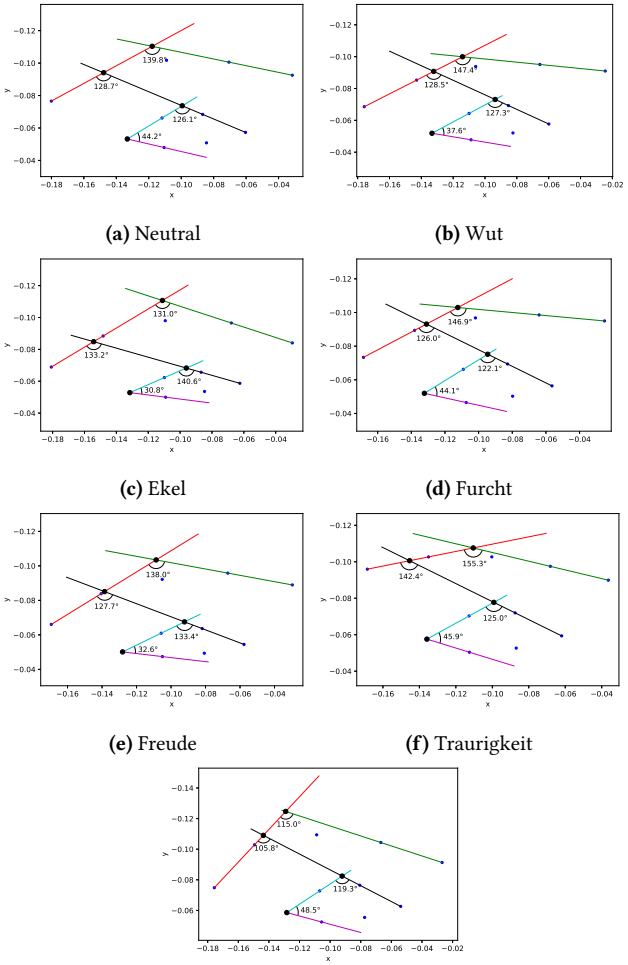
$$\alpha = \cos^{-1}\left(< \underline{g}_0, \underline{g}_1 >\right) \tag{6.1}$$

berechnet, wobei $< \cdot,\cdot >$ das Skalarprodukt ist. Die Vektoren $\underline{g}_i$ mit $i \in \{0,1\}$ stehen für die normierten Richtungsvektoren der sich schneidenden Geraden. Außerdem entsteht der Eindruck, dass die Augenbrauen stärkeren Bewegungen unterliegen. Daraus ergibt sich, dass Winkel zwischen den Augenlid-Geraden und den Augenbrauen als Merkmal geeignet sind. Für die Augenbrauen werden zwei Geraden verwendet, um die Krümmung der Augenbrauen durch den Winkel zwischen beiden Geraden erfassen zu können. Zur Auswahl der Winkel wurden die Gesichtsausdrücke der CK-Datenbank ausgewertet. Dafür wurde die Varianz aller Winkelvarianten ausgewertet. Die stärksten wurden ausgewählt. Insgesamt wurden 26 Winkel ausgewählt, um als Merkmale für die Unterscheidung der Emotionsklassen verwendet zu werden. Um die Veränderlichkeit der Varianzen zu verdeutlichen, werden die Winkel eines Auges für alle sieben Emotionsklassen dargestellt. Die Winkelmerkmale für alle Klassen sind in Abbildung 6.6 wiedergegeben.

Die Abbildungen 6.6 (a) - (g) zeigen eine gute Unterscheidbarkeit der Winkel zwischen den einzelnen Klassen und legen somit eine gute Unterscheidbarkeit zwischen den Klassen nahe. Der ASF Merkmalsvektor beinhaltet 26 einzelne Winkel aus dem Schnitt unterschiedlicher Geraden. Bei der Auswahl der Winkel wurde Wert darauf gelegt, alle Regionen des Gesichts in Betracht zu ziehen. Das heißt, zunächst wurden einzelne Regionen isoliert betrachtet und verschiedene Landmarken mittels Geraden und Winkeln in Beziehung gesetzt. In einem weiteren Schritt wurden verschiedene Regionen in Beziehung gesetzt. Es wurden sich schneidende Geraden aus beiden Regionen ausgewählt, um den Winkel zwischen diesen Geraden zu berechnen.

Die ersten acht Winkel setzen sich aus den Winkeln der beiden Augen und Augenbrauen zusammen. In Abbildung 6.7 (a) sind die Winkel des rechten Auges abgebildet und in Abbildung 6.7 (b) die Winkel des linken Auges.

Die Auswahl der Winkel für die Augenregionen ist nicht vollständig symmetrisch. Die Asymmetrie wurde gewählt, um Zusatzinformationen zu gewinnen. Die Winkel zwischen Geraden des Auges und der Augenbraue wurden gewählt, um Zusammenhänge zwischen den Augen und den Augenbrauen zu modellieren. Hiermit wird das Ziel verfolgt, das gesamte Gesicht als zusammenhängendes System zu modellieren. Die Mundregion beinhaltet 11



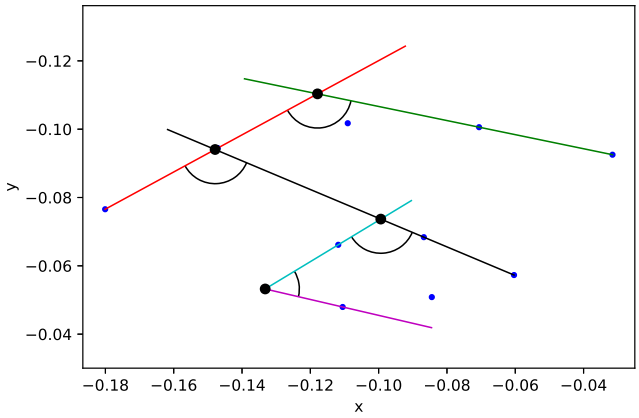
(a) Neutral **(b)** Wut

(c) Ekel **(d)** Furcht

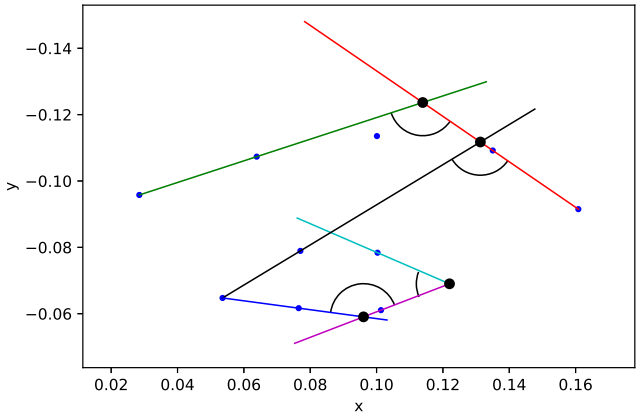
(e) Freude **(f)** Traurigkeit

(g) Überraschung

Abbildung 6.6: Verwendete Geraden und extrahierte Winkel für die Winkelmerkmale des ASF Merkmalssatzes für Augen und Augenbrauen. Die Landmarken wurden von Subjekt S055 der CK-Datenbank extrahiert.



(a) rechtes Auge



(b) linkes Auge

Abbildung 6.7: Geraden und die extrahierten Winkel für die Augenregion. (a) zeigt die extrahierten Winkel für das rechte Auge und (b) die extrahierten Winkel für das linke Auge.

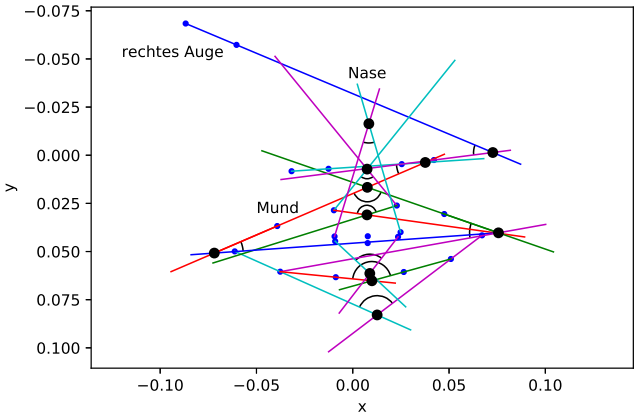


Abbildung 6.8: Geraden der Mundregion, sowie zwischen Augen, Nasen und Mundregion. Die extrahierten Winkel sind durch Kreisbögen dargestellt.

einzelne Winkel. Dabei wurden bereits Kombinationen zwischen dem Mund und weiteren Regionen verwendet. Dazu gehört die Nasenregion, ebenso wie die Region des rechten Auges. Die Geraden und extrahierten Winkel für die Mundregion sind in Abbildung 6.8 dargestellt.

Abbildung 6.8 verdeutlicht die modellierten Zusammenhänge zwischen den einzelnen Regionen. Diese Zusammenhänge sind den Verbindungen in der mimischen Muskulatur des Gesichts nachempfunden. Damit sind Bewegungszusammenhänge gemeint, wie zum Beispiel ein ausgeprägtes Grinsen einen Einfluss auf die Wangen und somit auf die Öffnung der Augen ausübt. Die Schnittwinkel zwischen den Geraden des Auges und des Mundes beschreiben diese Zusammenhänge. Zusätzlich gibt es Winkel, die durch Punkte des Munds und der Nase bestimmt werden. Hier werden Zusammenhänge bereits durch die Geraden beschrieben, die implizit in den Winkelmerkmalen enthalten sind. Mit den Winkelmerkmalen des ASF Merkmalsvektors wird ein zusammenhängendes Modell definiert. Um letztlich weitere Zusammenhänge zwischen allen Gesichtsbereichen zu modellieren gibt es sieben weitere kombinierte Winkel, die ausschließlich durch den Schnitt regionenübergreifender Geraden extrahiert werden. Diese kombinierten Winkel sind in Abbildung 6.9 wiedergegeben.

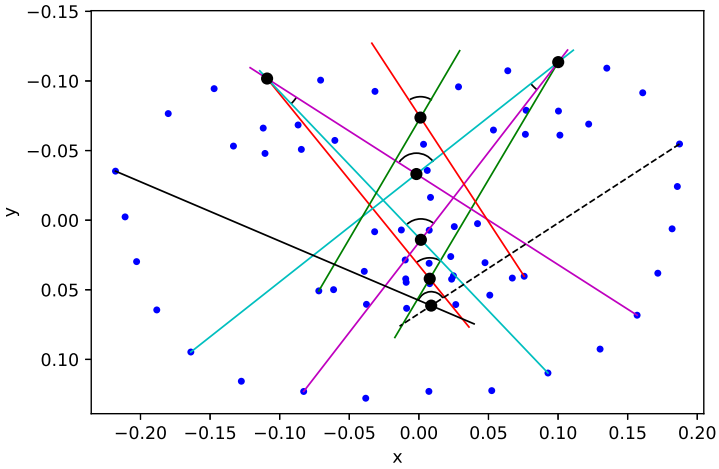


Abbildung 6.9: Regionen überspannende Geraden zur Modellierung des Zusammenspiels der Gesichtsregionen bei der Darstellung von emotionalen Gesichtsausdrücken.

Mit den kombinierten Winkeln aus Abbildung 6.9 sind alle 26 verwendeten Winkel des ASF Merkmalsvektors beschrieben. Die Winkel unterliegen Veränderungen, wenn sich die Konfiguration des Landmarkenmodells verändert. Die Winkel beschreiben eine Musterausprägung für den jeweiligen Gesichtsausdruck.

Um durch die Winkelmerkmale einen robusten Merkmalsvektor zu beschreiben, sollten die verwendeten Merkmale invariant unter geometrischen Transformationen sein. Das Landmarkenmodell ist so definiert [Coo00], dass es unter einer Ähnlichkeitstransformation invariant ist. Unter der Anwendung von Translationen, Rotationen und isotroper Skalierung verändert sich die Konfiguration der Landmarken nicht. Da die Berechnung der Geraden von den geordneten Landmarken abhängt, sind die extrahierten Winkel unter einer Ähnlichkeitstransformation invariant. Die Beleuchtung der Gesichtsbilder wird im Rahmen der Vorverarbeitung normalisiert. Unter der Normalisierung der Beleuchtung werden die Merkmale als invariant angenommen, da die Landmarken als Grundlage für die Extraktion der Merkmale verwendet werden.

(a) Wut **(b)** Überraschung

Abbildung 6.10: Darstellungen der Emotionen (a) Wut und (b) Überraschung von Subjekt S055 der CK-Datenbank (©Jeffrey Cohn)

Unter der Betrachtung der zu den Emotionsklassen gehörenden Bildern wurde beobachtet, dass es Größenunterschiede bestimmter Gesichtsregionen unter verschiedenen Emotionsklassen gibt. Zum Beispiel verändert sich die Augengröße durch das Aufreißen der Augen, oder die Mundgröße durch Zusammenziehen oder Spreizen der Lippen. Aus diesen Grund wurden zusätzliche Größenmerkmale untersucht.

6.1.2 Extraktion von Größeninformationen

Es wird angenommen, dass sich die Größe des Munds und der Augen bei verschiedenen Emotionen unterscheiden. Die Annahme kann durch Vergleich der beiden emotionalen Gesichtsausdrücke in Abbildung 6.10 belegt werden.

Die Emotionsdarstellungen in Abbildung 6.10 verdeutlichen den Eindruck, dass Augen und Mund unterschiedlich groß dargestellt werden. Daher wird eine Modellierung benötigt, um diese Größe zu beschreiben. Dazu wird eine Approximation der Formen für Auge und Mund benötigt, die schnell berechnet werden kann und die die Größe zuverlässig wiedergibt. Eine günstige Approximation für eine Form die durch Randpunkte gegeben ist, stellt die Ellipse dar. Die Ellipse besitzt wegen der beiden Halbachsen zwei Größenkomponenten. Um eine unter Ähnlichkeitstransformationen invariante Größenbeschreibung zu erlangen, kann das Verhältnis der beiden Halbachsen verwendet werden. Dieses Verhältnis invariant unter Translation, Rotation und Skalierung.

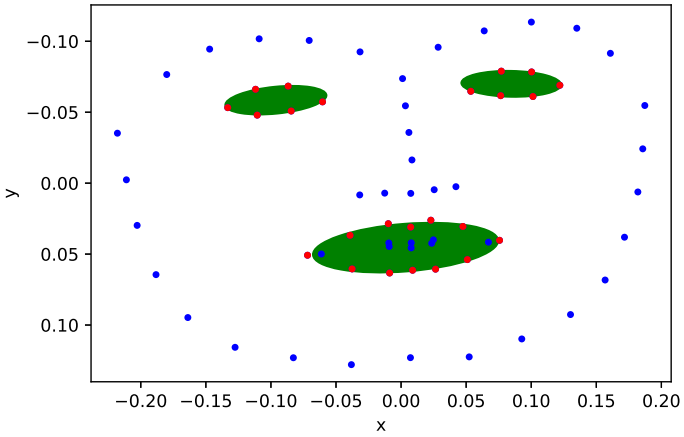


Abbildung 6.11: Berechnete Ellipsen zur Extraktion der Größenmerkmale für den ASF Merkmalssatz.

Dementsprechend bleiben unter Verwendung eines solchen Größenmaßes die Invarianzen gültig.

Es werden drei Ellipsen benötigt, um die drei Größenkomponenten des Gesichts zu beschreiben. Je eine Ellipse wird unter Verwendung der sechs Augenpunkte für jedes Auge bestimmt. Die äußeren Punkte des Munds bilden die Grundlage für die Ellipse zur Größenbeschreibung des Munds. Zur Berechnung der Ellipsen wird ein Least-Squares Fit verwendet. Grundlage ist die Ellipsengleichung

$$a \cdot x^2 + b \cdot x \cdot y + c \cdot y^2 + d \cdot x + e \cdot y + f = 0\,, \tag{6.2}$$

ein Polynom zweiter Ordnung mit sechs Parametern. Für den Least-Squares Fit werden die Parameter so angepasst, dass der quadratische Abstand zu den eingesetzten Punkten minimiert wird. Die berechneten Ellipsen für die Augen und den Mund sind in Abbildung 6.11 abgebildet.

Die Ellipsen für die Augen und den Mund lassen erkennen, dass sie die Ursprungsformen approximieren. Die Abstände zu den Punkten sind sehr gering. Maßgeblich für die Extraktion der Größeninformationen sind die Halbachsen der Ellipse. Die Parameterform der Ellipsengleichung wird durch

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_0 + a\cos t\cos\alpha - b\sin t\sin\alpha \\ y_0 + a\cos t\sin\alpha + b\sin t\cos\alpha \end{pmatrix}, \text{ mit} t \in [0, 2\pi) \tag{6.3}$$

definiert, wobei $[x_0, y_0]^T$ der Mittelpunkt der Ellipse ist, a und b sind die beiden Halbachsen der Ellipse, α ist der Rotationswinkel und t ist der Parameter. Zur Berechnung der Größeninformation werden nur die beiden Halbachsen benötigt. Das Merkmal Größe ergibt sich aus dem Verhältnis der beiden Variablen. Für die Berechnung wird die kleinere Halbachse ins Verhältnis zur größeren Halbachse gesetzt, so dass der Wert des Merkmals im Intervall $[0, 1]$ liegt. Somit ergibt

$$s = \frac{b}{a} \tag{6.4}$$

das Verhältnis, wobei a die große Halbachse und b die kleine Halbachse ist. Dieser Zusammenhang wird in Abbildung 6.12 anhand einer Augenellipse verdeutlicht.

Nach diesem Vorgehen werden alle drei Größenmerkmale berechnet. Die Verwendung des Verhältnisses ermöglicht die Beibehaltung der Invarianzen, wie bereits weiter oben argumentiert wurde.

Zu den 26 Winkeln aus dem vorherigen Abschnitt werden die drei Größeninformationen hinzugenommen. Es ergibt sich ein Merkmalsvektor der Größe 29 und eine signifikante Reduktion der Dimensionalität im Vergleich zur der direkten Nutzung der Landmarken als Merkmalsvektor, was einer Größe von 136 Elementen entspricht. Nachdem der Merkmalsvektor fertiggestellt ist, muss entschieden werden mit welchem Klassifikator die besten Ergebnisse erzielt werden können.

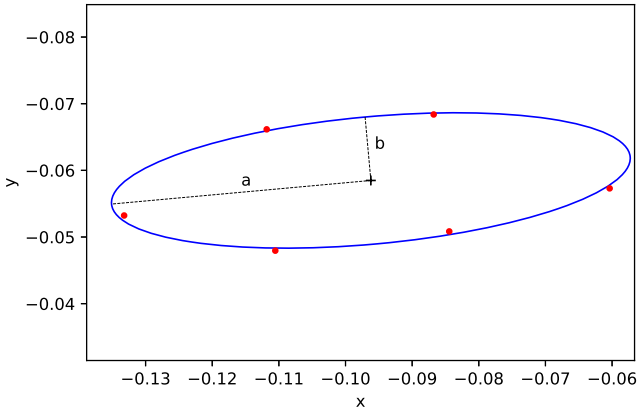


Abbildung 6.12: Rotierte Ellipse des rechten Auges mit eingezeichneten großer Halbachse a und kleiner Halbachse b.

6.1.3 Klassifikatorauswahl

Der ASF Merkmalssatz beschreibt einen Deskriptor für emotionale Gesichtsausdrücke. Daraus lässt sich ableiten, dass der Deskriptor eine Unterscheidung von statischen Gesichtsausdrücken zulässt. Mit Hilfe der Merkmale werden emotionale Gesichtsausdrücke einer diskreten Klasse zugeordnet. Die Zuordnung zu einer diskreten Klasse setzt die Verwendung eines Klassifikationsalgorithmus voraus. Die Klassifikationsalgorithmen gehören zu den überwachten Lernverfahren. Die Auswahl des besten Verfahrens für das Klassifikationsproblem hängt von den verwendeten Daten ab. Handelt es sich nur um eine geringe Menge von Daten, deren Dimension gering ist, empfiehlt sich die Verwendung einer linearen Support-Vektor-Maschine und des Naive Bayes Klassifikators. Ist Erklärbarkeit der Klassifikationsentscheidung wichtig, fällt die Wahl auf den Entscheidungsbaum oder die logistische Regression. Im Fall der ASF Merkmale wird ein Klassifikationsverfahren benötigt,

dass eine hohe Genauigkeit und hohe Geschwindigkeit garantiert. Verwendet werden können Klassifikationsmodelle wie die Kernel Support-Vektor-Maschinen, Neuronale Netze, Random Forests und Gradient Boosting Tree-Verfahren. Betrachtet man vergangene Klassifikationswettbewerbe, beispielsweise bei der Online-Community Kaggle, so weisen die Gradient Boosting Tree-Verfahren sehr hohe Genauigkeiten bei sehr hoher Geschwindigkeit auf, die vergleichbare Leistungen liefern wie Verfahren, die auf künstlichen Neuronalen Netzen basieren. Aus diesem Grund wurde in dieser Arbeit für die ASF Merkmale das XGBoost Klassifikationsverfahren von Chen und Guestrin [Che16] ausgewählt. Das Verfahren ist in Abschnitt 3.3.2 beschrieben.

Der ASF-Merkmalsatz wird mit einem künstlichen Neuronalen Netz (KNN) verglichen. Dazu wird in dieser Arbeit das mit Imagenet-Gewichten initialisierte VGG-16 Netz verwendet. Die Klasse der KNN wird in Abschnitt 3.3.3 eingeführt. Um einen gerechten Vergleich der Klassifikatoren und der Erstellung der Merkmalsvektoren zu erreichen, wird das KNN ohne größere Vorverarbeitungsschritte verwendet.

6.2 Experimente zur Emotionserkennung

Der ASF Merkmalssatz wird anhand verschiedener Datenbanken getestet. Dazu gehören die Cohn-Kanade Plus (CK) Datenbank, sowie die Oulu-Casia (OC) Datenbank der Finnischen Universität Oulu-Casia. Die CK Datenbank ist eine Erweiterung der ursprünglichen Cohn-Kanade Datenbank. Der usprüngliche Zweck dieser Datenbank bestand in der Codierung der Bilder mit dem Facial Action Coding System (FACS) von Paul Ekman [Ekm78]. In der vorherigen Variante waren die Emotionslabels zwar gegeben, allerdings waren diese nicht Validiert. Mit der Erweiterung erfolgte eine Reevaluierung dieser Labels mit Hilfe des Emotionspräsiktionstabelle aus dem FACS Buch von Ekman et al. [Ekm02]. Mit Hilfe dieser Tabelle können erkannte Kombinationen von Facial Actions (FA) auf diskrete Emotionen abgebildet werden. Diese Tabelle enthält alle Basisemotionen bis auf Gleichgültigkeit. In dieser Datenbank sind alle Emotionen durch Schauspieler dargestellt, bis auf wenige Sequenzen mit einem Lächeln. Diese Emotionsdarstellungen waren spontan und

wurden ebenfalls aufgezeichnet. Außerdem enthält die CK Datenbank ausschließlich Frontalbilder. Die Zuordnung der Emotionsklassen wurde durch Lucey et al. [Luc10a] mit Hilfe eines Active-Appearance-Models und einer Support-Vector-Maschine getestet.

Die Oulu-Casia Datenbank wurde von der Finnischen Universität Oulu-Casia 2008 und 2009 erzeugt. Bei der Erzeugung der Datenbank lagen unterschiedliche Beleuchtungsszenarien im Vordergrund. Deshalb wurden die Bilddaten bei zusätzlicher Beleuchtung, leichter Beleuchtung mit dem Computerdisplay und ohne Beleuchtung erfasst. Die Gesichter wurden frontal mit einer visuellen Kamera und einer NIR-Kamera aufgenommen. Die Emotionen sind alle dargestellt, wobei ein Teil an der Universität Oulu-Casia und ein weiterer Teil an der Universität von Beijing aufgenommen wurde. Die verwendeten Kameras besitzen eine Auflösung von 320 x 240 Pixeln. Der Kontrast der Daten ist gering verglichen mit den Daten der CK Datenbank.

Zur Prüfung der Leistungsfähigkeit der von den Landmarken abgeleiteten Merkmale erfolgten zwei grundlegende Experimente: Vergleiche zu Benchmarks auf bekannten Datenbanken und ein direkter Vergleich der Leistungsfähigkeit mit Deep Learning Ansätzen. Mit den Experimenten wird gezeigt, dass der gewählte Merkmalssatz gleiche Leistungsfähigkeiten besitzt wie der Stand der Technik. Außerdem werden geschwindigkeitsabhängige Metriken betrachtet und evaluiert. Im folgenden Abschnitt wird mit dem Benchmark-Test begonnen.

6.2.1 Benchmarkuntersuchung auf bekannten Datenbanken

Um die Leistung des ASF Merkmalssatzes mit dem Stand der Technik zu vergleichen, erfolgen Tests auf bekannten Datenbanken: der CK Datenbank und der OC Datenbank. Zunächst werden die Datenbanken isoliert betrachtet. Hierzu wird ein Kreuzvalidierungsschema verwendet, in dem fünf Durchläufe ausgeführt werden. Die Menge der Merkmale wird in ein fünftel Testdaten und vier fünftel Trainingsdaten aufgeteilt. Für die ASF Merkmale wird ein XGBoost Klassifikator eingesetzt.

Es wird eine Parameteroptimierung mittels Gittersuche durchgeführt. Für jeden einzustellenden Parameter wird eine Menge von möglichen Parametern vorgegeben. Dabei wurde schrittweise nach der Anleitung von Analytics Vidhya [1] vorgegangen. Bei der Gittersuche wird für den zu optimierenden Parameter eine Werteliste festgelegt. Diese Wertliste wird in einer Schleife durchlaufen und der Klassifikator wird mit dem Parameterwert per Kreuzvalidierung bewertet. Während jedes Schleifendurchlaufs wird die Genauigkeit der Prädiktion bewertet. Nach Abschluss der Kreuzvalidierung wird die durchschnittliche Genauigkeit für den Parameterwert berechnet. Der Parameterwert mit maximaler Genauigkeit wird als optimierter Parameter gewählt.

Die Parameteroptimierung wird für jede der Datenbanken einzeln und für beide gemeinsam ausgeführt. Mit den optimierten Parametern werden die Versuche mit den Datenbanken durchgeführt. Zur Auswertung werden die Kennzahlen Genauigkeit, Präzision, Sensitivität und F_1-Score berechnet. Für die Experimente werden die Bilddaten vorbereitet. Es werden nur Bilder mit dem vollständig ausgeprägten Gesichtsausdruck ausgewählt. In beiden Datenbanken sind Sequenzen von Einzelbildern gespeichert. Jede Sequenz beginnt mit einem neutralen Gesichtsausdruck und endet mit einem vollständig ausgeprägten Gesichtsausdruck. Die Anzahl der Bilder wird maximiert, indem aus jeder Sequenz die drei letzten Bilder und das jeweils erste Bild ausgewählt werden. Damit wurde eine ausreichend große Datenmenge erlangt. Für den Versuch stehen 1441 Bilder der CK Datenbank und 1517 Bilder der OC Datenbank zur Verfügung. Tabelle 6.1 zeigt die Verteilung der Bilder auf die Emotionsklassen.

6.2.1.1 XBoost Klassifikator Parameter

In der Literatur wird zwischen dem Sechs- und Siebenklassenproblem unterschieden. Beim Sechsklassenproblem wird auf die Verwendung der neutralen

[1] https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/ (abgerufen am 23.05.2020)

Tabelle 6.1: Verteilung der Einzelbilder der CK und OC Datenbanken auf die sieben Basisemotionen *Neutral*, *Wut*, *Ekel*, *Furcht*, *Freude*, *Traurigkeit* und *Überraschung*.

Emotionsklasse	**Anzahl CK**	**Anzahl OC**
Neutral	112	80
Wut	180	240
Ekel	198	240
Furcht	198	240
Freude	270	240
Traurigkeit	225	240
Überraschung	258	237
Summe	1441	1517

Klasse verzichtet. Für den Versuch wird der XGBoost Klassifikator verwendet. Als Optimierungskennzahl wird die Genauigkeit x_{acc}

$$x_{\text{acc}} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FN} + \text{FP})} \tag{6.5}$$

verwendet, um den für den Datensatz optimalen Parameterwert zu ermitteln. Der Prozess der Parameteroptimierung wird anhand des Verlaufs der Genauigkeit dargestellt. Die Parameteroptimierung wird systematisch durchgeführt. Es wird mit einem Standardparametersatz begonnen und es werden schrittweise die korrekten Parameter eingestellt, sowie die optimale Anzahl an Bäumen.

Die Parameter des Klassifikators lassen sich in drei Gruppen aufteilen: allgemeine Parameter, Booster Parameter und Lern Parameter.

Allgemeine Parameter Hier kann der Booster des Klassifikators beeinflusst werden. Gewählt werden können der Baum Booster und der lineare Booster. Außerdem kann die Zahl der zu verwendenden Threads eingestellt werden. Ebenso lässt sich der Detailgrad der Ausgabe einstellen.

Booster Parameter Zu den Booster Parametern gehört die Variable η, die ähnlich verwendet wird wie die Lernrate in Optimierungsverfahren. Zusätzlich gibt es noch die minimale Summe der Gewichte, die in einem

Kindbaum gesammelt wird. Dieser Parameter beeinflusst das Overfitting. Die maximale Tiefe für Bäume in XGBoost dient ebenfalls der Kontrolle des Overfitting. Die maximale Anzahl der terminalen Blätter kann anstatt des Parameters der maximalen Baumtiefe verwendet werden. Der Parameter Gamma liefert einen Grenzwert für die Reduktion der Loss-Funktion, die minimal um diesen Wert reduziert werden muss, damit ein Baumsplit akzeptiert wird. Der Parameter Max Delta Step wird verwendet, um den Update Schritt konservativer zu machen. Generell wird dieser Parameter nicht verwendet, außer im Fall von schlecht balancierten Datensätzen für die logistische Regression. Mittels Subsample wird der Anteil der Beobachtungen eingegrenzt, der als Zufallssample ausgewählt wird. Damit soll Overfitting vermieden werden, allerdings ist Underfitting möglich. Der Parameter Colsample By Tree gibt den Anteil der Spalten eines Merkmalsvektors an, aus dem zufällig Spalten gezogen werden. Dieser Parameter ist vergleichbar mit der Angabe der maximalen Anzahl an Merkmalen in anderen Algorithmen. Der Parameter Colsample By Level ist eine Alternative zu Colsample By Tree und Subsample. In dieser Arbeit werden Subsample und Colsample By Tree verwendet. Der L_2 Regularisierungsterm λ wird verwendet um Overfitting zu kontrollieren. Der Parameter α ist ein L_1-Regularisierungsterm und wird zur Beschleunigung der Konvergenz bei hoch-dimensionalen Merkmalen verwendet. Mit dem Parameter Scale Positive Weights werden schlecht balancierte Klassensamples höher gewichtet.

Lern Parameter Die Lern Parameter setzen sich zusammen aus den Parametern Objective, Evaluationsmetrik und Random Seed. Der Parameter Objective bezieht sich auf die Zielfunktion für den Baumoptimierer. Für die binäre Klassifikation wählt man *binary:logistic*, für die Multiklassen Klassifikation *multi:softmax* für Multiklassen Klassifikation und für die Ausgabe von Wahrscheinlichkeiten bei der Multiklassen Klassifikation *multi:softprob*. Für die vorliegende Problemstellung wird die Einstellung *multi:softmax* für die reine Klassifikation und *multi:softprob* für die Verwendung einer Reject-Option eingesetzt. Mit der Evaluationsmetrik wird die Lossfunktion bestimmt. Hier muss zwischen Regressions-

und Klassifikationslossfunktionen gewählt werden. Durch den Random Seed Parameter kann für reproduzierbare Ergebnisse gesorgt werden.

6.2.1.2 Klassifikation mit der CK Datenbank

Für die Parameteroptimierung werden nacheinander einzelne Parameter optimiert. Zunächst werden die Parameter Maximale Tiefe und Minimale Summe der Gewichte gemeinsam optimiert. Dafür werden je fünf Werte für beide Parameter eingestellt und paarweise durchlaufen, sodass jede Kombination verarbeitet wird. Im nächsten Schritt erfolgt die Optimierung des Parameters Gamma, wobei hier auf fünf Parameter zurückgegriffen wird. Der dritte Parameterschritt umfasst die Optimierung der Parameter Subsample und Colsample By Tree mit jeweils fünf möglichen Parameterwerten. Der vierte Schritt optimiert den ersten Regularisierungsparameter α zunächst grob aus fünf Parameterwerten. Im fünften Schritt wird α mit fünf weiteren möglichen Parameterwerten fein eingestellt. Der sechste und siebte Schritt stellt den Regierungsparameter λ ein. Es wird eine fünffache Kreuzvalidierung für die Optimierung ausgeführt. Das heißt, in jedem Schritt werden 20 Prozent der Samples als Testmenge und 80 Prozent der Samples als Trainingsmenge verwendet. Während jedes Schritts bleiben die restlichen Parameter auf dem bis dahin ermittelten Wert fixiert. Für die erste Optimierung wird ein Initialisierungsparametersatz laut Tabelle 6.2 verwendet. Die verwendete Implementierung des XGBoost Klassifikators entstammt DMLC [1].

Nach jedem Optimierungsschritt wird die Genauigkeit als Kennzahl ausgewertet. Die Entwicklung der Genauigkeit über dem Verlauf der Parameteroptimierungsschritte ist in Abbildung 6.13 abgebildet. Die Ergebnisse der Parameteroptimierungen sind für das Sechsklassenproblem und das Siebenklassenproblem getrennt dargestellt. In der Abbildung ist zu erkennen, dass die Genauigkeit mit jedem Optimierungsschritt für beide Klassifikationsprobleme steigt.

[1] http://xgboost.ai

Tabelle 6.2: Initialisierungsparameter für den XGBoost Klassifikator.

Parameter	Wert
Lernrate (η)	0.1
Anzahl Schätzer	1000
Maximale Tiefe	5
Minimale Summe der Gewichte	1
γ	0
Subsample	0.8
Colsample By Tree	0.8
Scale Positive Weight	1
Random Seed	27

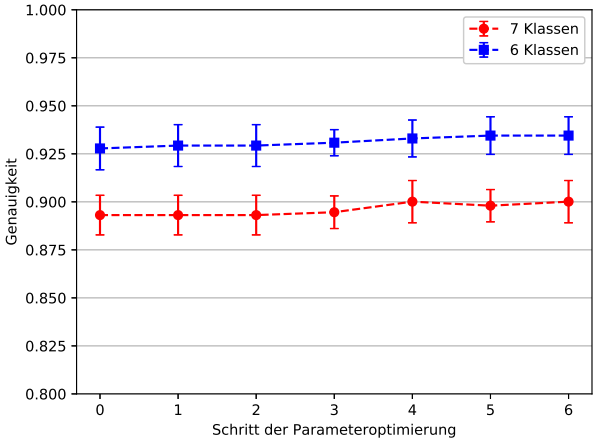


Abbildung 6.13: Entwicklung der Zielgröße Genauigkeit unter der Parameteroptimierung für die Cohn-Kanade Datenbank. Die Entwicklung der Genauigkeit für das Sechs- und Siebenklassenproblem.

Tabelle 6.3: Optimierte Parameter für das Sechs- und Siebenklassenproblem auf der CK Datenbank.

Parameter	**6 Klassen**	**7 Klassen**
Maximale Tiefe	9	5
Minimale Summe der Gewichte	1	1
γ	0	0
Subsample	0.6	0.65
Colsample By Tree	0.55	0.7
α	0	0
λ	1	1
Anzahl Schätzer	182	294

Tabelle 6.4: Klassifikationsergebnisse für das sechs und sieben Klassenproblem auf der CK Datenbank mit dem XGBoost Klassifikator und den optimierten Parametern.

Problem	**Klasse**	**Sensitivität**	**Präzision**	**F_1-Score**
6 Klassen	A	0.96 ± 0.02	0.89 ± 0.02	0.92 ± 0.02
	D	0.95 ± 0.04	0.96 ± 0.03	0.96 ± 0.01
	F	0.82 ± 0.06	0.90 ± 0.04	0.86 ± 0.04
	H	0.96 ± 0.03	0.95 ± 0.02	0.95 ± 0.01
	S	0.93 ± 0.04	0.92 ± 0.04	0.92 ± 0.02
	U	0.97 ± 0.02	0.98 ± 0.02	0.98 ± 0.02
	Total	0.93 ± 0.05	0.93 ± 0.03	0.93 ± 0.04
7 Klassen	N	0.50 ± 0.12	0.85 ± 0.10	0.61 ± 0.08
	A	0.94 ± 0.04	0.80 ± 0.04	0.87 ± 0.02
	D	0.94 ± 0.03	0.93 ± 0.02	0.94 ± 0.02
	F	0.79 ± 0.04	0.83 ± 0.03	0.81 ± 0.02
	H	0.95 ± 0.03	0.93 ± 0.01	0.94 ± 0.02
	S	0.95 ± 0.01	0.88 ± 0.05	0.91 ± 0.03
	U	0.96 ± 0.03	0.98 ± 0.02	0.97 ± 0.02
	Total	0.86 ± 0.16	0.89 ± 0.06	0.86 ± 0.11

Die resultierende Parametermenge für beide Probleme ist in Tabelle 6.3 dargestellt.

Unter Verwendung der optimierten Parameter für das Sechs- und Siebenklassenproblem erfolgte eine Kreuzvalidierung auf der CK Datenbank. Für das Sechsklassenproblem wurden die Bilder für die Klasse Neutral ausgelassen. Für die Ergebnisdarstellung werden neben der Sensitivität die Präzision und der F_1-Score verwendet. Die Tabelle 6.4 zeigt die Ergebnisse für das Sechs- und Siebenklassenproblem.

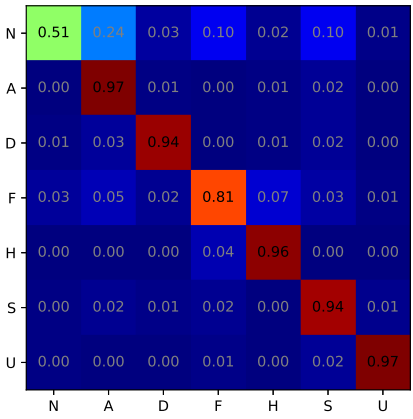


Abbildung 6.14: Konfusionsmatrix des Siebenklassenproblems für die CK-Datenbank.

Vergleicht man die erreichte Sensitivität für das Sechs- und Siebenklassenproblem in Tabelle 6.4, so fällt auf, dass die neutrale Klasse mit Abstand das schwächste Ergebnis erreicht. Durch die geringe Sensitivität der neutralen Klasse verringern sich die erreichten Sensitivitäten der übrigen Klassen. Dies kann durch eine große Ähnlichkeit des neutralen Gesichtsausdrucks zu vielen der übrigen Klassen begründet werden. Die Konfusionsmatrix in Abbildung 6.14 verdeutlicht diesen Umstand.

Insbesondere Samples der Klassen A, F und S werden der Klasse N zugeordnet. Diese konnten nicht der eigentlichen Klasse zugeordnet werden. Um eine höhere Genauigkeit zu erreichen, besteht die Möglichkeit eine unbekannte Klasse hinzuzufügen. Im Hinblick auf die dynamische Erkennung von Emotionen und die Verfolgung der Emotionsklassen kann so mit Fehlklassifikationen umgegangen werden. Wendet man eine unbekannte Klasse an, muss ein Kriterium eingeführt werden, um Objekte dieser Klasse zuzuordnen. Zu diesem Zweck muss die Prädiktion unter Angabe der bedingten Klassenwahrscheinlichkeiten erfolgen. Durch Auswahl eines Grenzwerts für die bedingte Wahrscheinlichkeit kann dann die Klasse zugeordnet werden. Der Grenzwert

bildet eine untere Grenze, die übertroffen werden muss, damit die Klasse zugeordnet werden kann. Unterschreiten alle bedingten Wahrscheinlichkeiten diese Grenze, erfolgt die Zuordnung zur unbekannten Klasse. Die Ausgabe des Klassifikators wird durch

$$\arg\max_{x \in \mathcal{K}} f\left(x|\underline{y}\right) \tag{6.6}$$

gegeben, wobei x eine Klasse aus der Menge $\mathcal{K} = \{N, A, D, F, H, S, U\}$ ist und $\underline{y}$ den beobachteten Merkmalsvektor bezeichnet. $f(\cdot)$ beschreibt die bedingte Wahrscheinlichkeit als Ausgabe des Klassifikators. Unter Verwendung einer Bedingung für die Auswahl einer Klasse muss eine Ablehnungsoption eingeführt werden. Bei der Ablehnungsoption (engl. Rejection Option) wird ein Schwellwert gesetzt, der die Klassenzuweisung zurückweist, wenn die maximale posteriore Wahrscheinlichkeit kleiner ist als dieser Schwellwert. Die Wahl des Schwellwerts kann eingegrenzt werden. In dem vorliegenden Problem gibt es $k = 7$ Klassen. Für die posterioren Wahrscheinlichkeiten gilt

$$\sum_{i=1}^{k} f\left(x = \mathcal{K}_k|\underline{y}\right) = 1\,, \tag{6.7}$$

wobei $\mathcal{K}_k$ der k-ten Klasse aus der Menge $\mathcal{K}$ entspricht und $f\left(x|\underline{y}\right)$ die posteriore Wahrscheinlichkeit ist, wenn $\underline{y}$ zur Klasse x gehört. Aufgrund von (6.7) gilt, dass bei Verwendung eines Schwellwerts $\theta = 1/k$ jedes Sample angenommen wird, da diese Wahrscheinlichkeit entweder immer erreicht oder übertroffen wird. Bei der Verwendung des Schwellwerts $\theta = 1$ wird jedes Sample abgelehnt, da

$$\max_k f\left(x|\underline{y}\right) < 1 \tag{6.8}$$

Tabelle 6.5: Genauigkeiten der Einzelklassenexperimente mit der CK-Datenbank. Für das Siebenklassenproblem wurde ebenfalls eine Ablehnungsoption mit $\theta = 0.5$ verwendet.

Anwendung	Genauigkeit
6 Klassen	0.93 ± 0.01
7 Klassen	0.89 ± 0.01
7 Klassen + Reject	0.91 ± 0.01

gilt. Demnach muss der Schwellwert θ in dem offenen Intervall $(1/k, 1)$ enthalten sein. Durch die Verwendung der Ablehnungsoption konnte die Genauigkeit verbessert werden. In Tabelle 6.5 sind die Genauigkeiten des Einzelklassenexperiments aufgelistet. Als Zusatz ist in der Tabelle die Genauigkeit unter Verwendung eines Schwellwerts von 0.5 enthalten.

Unter Verwendung der Ablehnungsoption konnten die Genauigkeit und die Erkennungswahrscheinlichkeit für die neutrale Klasse erhöht werden. Abgelehnte Samples werden der Klasse *Unbekannt* zugeordnet. Die abgelehnten Samples gehen nicht in die Bewertung der Genauigkeit mit ein. Aus diesem Grund werden nur jene Samples verwendet, die nicht abgelehnt wurden.

6.2.1.3 Klassifikation mit der OC Datenbank

Neben der CK Datenbank erfolgten Experimente auf der OC Datenbank der finnischen Oulu-Casia Universität. Wie bereits im vorherigen Abschnitt berichtet, wurden 1517 Bilder aus der OC Datenbank extrahiert und für die Auswertung mit dem behandelten Merkmalssatz vorbereitet. In ähnlicher Weise wie bei der CK Datenbank erfolgten Auswertungen des Sechs- und Siebenklassenproblems. Äquivalent zum vorherigen Abschnitt erfolgte für beide Problemklassen eine Optimierung der Parameter für den XGBoost Klassifikator. Die optimierten Parametereinstellungen für den XGBoost Klassifikator sind in Tabelle 6.6 enthalten.

Die Ergebnisse des Sechs- und Siebenklassenproblems sind in Tabelle 6.7 aufgeführt. Enthalten sind die Kennzahlen Sensitivität, Präzision und F_1-Score pro Klasse und wurden über allen Klassen gemittelt.

Tabelle 6.6: Optimierte Parameter für das Sechs- und Siebenklassenproblem auf der OC Datenbank.

Parameter	**6 Klassen**	**7 Klassen**
Maximale Tiefe	12	5
Minimale Summe der Gewichte	1	1
γ	0	0
Subsample	0.8	0.65
Colsample By Tree	0.75	0.7
α	0	0
λ	0.001	1
Anzahl Schätzer	49	294

Tabelle 6.7: Klassifikationsergebnisse für das Sechs- und Siebenklassenproblem auf der OC Datenbank mit dem XGBoost Klassifikator und den optimierten Parametern.

Problem	**Klasse**	**Sensitivität**	**Präzision**	**F_1-Score**
6 Klassen	A	0.63 ± 0.04	0.90 ± 0.05	0.74 ± 0.03
	D	0.77 ± 0.08	0.58 ± 0.25	0.62 ± 0.13
	F	0.75 ± 0.13	0.69 ± 0.21	0.68 ± 0.07
	H	0.74 ± 0.06	0.94 ± 0.02	0.82 ± 0.04
	S	0.72 ± 0.07	0.88 ± 0.06	0.79 ± 0.04
	U	0.71 ± 0.05	0.93 ± 0.03	0.80 ± 0.04
	Total	0.72 ± 0.04	0.82 ± 0.14	0.74 ± 0.07
7 Klassen	N	0.08 ± 0.05	0.40 ± 0.20	0.12 ± 0.07
	A	0.67 ± 0.05	0.85 ± 0.03	0.74 ± 0.02
	D	0.72 ± 0.12	0.67 ± 0.26	0.65 ± 0.13
	F	0.81 ± 0.15	0.57 ± 0.20	0.62 ± 0.07
	H	0.73 ± 0.06	0.90 ± 0.03	0.81 ± 0.04
	S	0.72 ± 0.06	0.85 ± 0.06	0.78 ± 0.05
	U	0.71 ± 0.04	0.92 ± 0.03	0.80 ± 0.03
	Total	0.63 ± 0.23	0.74 ± 0.18	0.65 ± 0.22

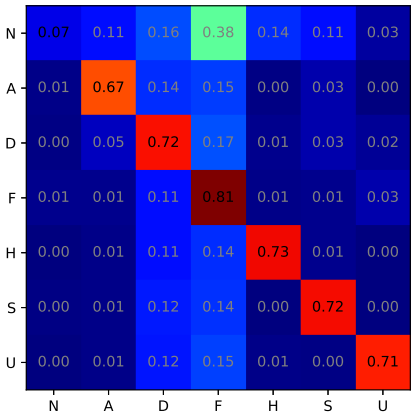


Abbildung 6.15: Konfusionsmatrix mit den relativen Sensitivitäten der einzelnen Klassen und Fehlzuordnungen für den Siebenklassentest mit der OC Datenbank.

Die Tabelle 6.7 zeigt schwächere Ergebnisse im Vergleich zu den Ergebnissen der CK-Datenbank. Dies kann mit der geringeren Auflösung der Bilddaten erklärt werden. Aufgrund der geringeren Auflösung sind die Details schlechter zu erkennen. Dieser Umstand kann verbessert werden, indem ein optimierter Algorithmus zur Detektion von Landmarken verwendet wird. Um den Vergleich weiter zu führen, muss das schlechte Ergebnis der Erkennung für den neutralen Zustand betrachtet werden. Bei einer Verwendung von sieben Klassen ist eine durchschnittliche Sensitivität von acht Prozent sehr gering. Zur Analyse des Ergebnisses wird die Konfusionsmatrix zur Betrachtung hinzugezogen. Die Konfusionsmatrix des Siebenklassenproblems ist in Abbildung 6.15 dargestellt.

Die meisten Fehlzuordnungen von Samples der Klasse Neutral entstanden in der Klasse F (Furcht). Ungefähr 38 Prozent der Samples wurden falsch zugeordnet. An zweiter Stelle wurden 16 Prozent der Samples der Klasse D (Ekel) und 14 Prozent der Samples der Klasse H (Freude) zugeordnet. In Summe entspricht das bereits 68 Prozent der Samples. Um dieses Ergebnis zu erläutern, kann auf die Verteilung der Samples auf die Klassen zurückgegriffen werden. Bis auf die Klasse N sind für alle weitere Klassen 240 Bilder vorhanden. Nur

ein Drittel dieser Anzahl, 80 Bilder, sind aus der Klasse N vorhanden. Somit ist die Klasse stark unterrepräsentiert. Durch eine Erhöhung des Anteils an neutralen Bildern könnte hier eine Verbesserung erreicht werden. Eine weitere Erhöhung der Trainingssamples könnte die Ergebnisse weiter stabilisieren. Hierbei handelt es sich um Empfehlungen für eine Erweiterung der Untersuchung. Bisher wurde diese Untersuchung noch nicht durchgeführt.

Da der Stand der Technik hauptsächlich auf Methoden aus den Fundus der Tiefen Neuronalen Netze besteht, ist der Vergleich mit ebendiesen Methoden unabdingbar. Der folgende Abschnitt behandelt den Vergleich mit dem Stand der Technik.

6.2.2 Vergleich mit Deep Learning Ansätzen zur Emotionserkennung

Tiefen Neuronalen Netzen liegt die Annahme zugrunde, dass ein solches Netz selbständig relevante Merkmale aus Eingabedaten bezieht, um eine korrekte Klassifikation durchzuführen. In der idealen Vorstellung wird davon ausgegangen, dass ohne Verwendung von Vorverarbeitungsschritten eine gute Klassifikation möglich ist. Betrachtet man die aktuelle Literatur zum Stand der Technik in Abschnitt 2.3, gibt es viele Methoden, die auf künstliche Neuronale Netze zurückgreifen. Die meisten Veröffentlichungen zu Tiefen Neuronalen Netzen für die Emotionserkennung unterscheiden sich in der Vorverarbeitung. Die Netze besitzen nur leichte Unterschiede. Aus diesem Grund ist der Vergleich eines klassischen auf Merkmalen basierten Verfahrens mit Tiefen Neuronalen Netzen wichtig.

Der hier beschriebene Merkmalssatz basiert auf der Berechnung spezialisierter Merkmale, die aus der Struktur des Gesichtsausdrucks extrahiert werden. Die Merkmalsextraktion ist vom verwendeten Klassifikator unabhängig. Der verwendete Klassifikator kann auf Basis der gewünschten Genauigkeit ausgewählt werden. Aus diesem Grund wurde im vorherigen Kapitel der XGBoost Klassifikator verwendet. In diesem Kapitel werden Ergebnisse nach dem Stand

der Technik mit den ermittelten Werten für den in dieser Arbeit entwickelten Merkmalssatz verglichen. Neben Ergebnissen, die im Rahmen von Veröffentlichungen angegeben wurden, werden in dieser Arbeit eigene Ergebnisse unter Verwendung eines VGG-16 Netzes, basierend auf einer Keras Implementierung ermittelt. Das Netz wurde von Simonyan et al. in [Sim14] veröffentlicht. Das Netz wurde mit vorab trainierten Gewichten initialisiert. Die Gewichte resultierten aus einem Training des Netzes mit dem ImageNet Datensatz von Deng et al. [Den09]. Die Anpassung des Netzes auf das Sechsklassenproblem der Emotionserkennung erfolgt durch Feinabgleich. Dazu wird die Ausgabeschicht durch eine Softmax-Schicht mit sechs Ausgabeknoten ersetzt und die Gewichte der darunterliegenden Schichten werden fixiert. Um Overfitting zu vermeiden erfolgte der Einsatz einer Dropout-Schicht vor der Ausgabe-Schicht.

In der Vorverarbeitung der Bilder erfolgten mehrere Schritte:

- Ausschneiden der Gesichtsregion des Bildes
- Anpassung der Größe des Ausschnittes auf 224 x 224 Pixel
- Anpassung der Helligkeit durch Gamma-Korrektur mit Exponent 0.5

Die Vorverarbeitung der Bilder erfolgte für beide Methoden identisch. Die Bilder werden direkt als Eingabe in das VGG-16 Netz gegeben. Für den ASF Merkmalssatz erfolgt zunächst die Extraktion der Merkmale aus den Bildern. Die Merkmalsextraktion im neuronalen Netz erfolgt durch Traversion durch das gesamte Netz. In diesem Experiment wurden die sechs Basisemotionen verwendet. Im ersten Experiment wird die CK-Datenbank ausgewertet. Die Ergebnisse des VGG-16 Netzes sind in Tabelle 6.8 für die CK-Datenbank und in Tabelle 6.9 für die OC-Datenbank gegeben. In Tabelle 6.8 fällt sofort auf, dass die Sensitivität für die Klasse A sehr gering ist und die Präzision sehr hoch. Dieses Ergebnis zeigt, dass die Zuordnungen zu Klasse A sehr eng beieinander liegen. Zusätzlich fällt eine hohe Sensitivität mit geringer Präzision bei den Klassen F und S auf. Aufgrund des ähnlichen Aussehens der Klassen

Tabelle 6.8: Vergleichsergebnisse für den CK Datensatz.

Methode	Klasse	Sensitivität	Präzision	F_1-Score
VGG-16	A	0.51 ± 0.09	0.98 ± 0.04	0.66 ± 0.08
	D	0.88 ± 0.07	0.95 ± 0.06	0.91 ± 0.03
	F	0.98 ± 0.02	0.77 ± 0.02	0.86 ± 0.02
	H	0.92 ± 0.04	0.97 ± 0.03	0.94 ± 0.03
	S	0.98 ± 0.02	0.79 ± 0.02	0.87 ± 0.01
	U	0.98 ± 0.02	0.97 ± 0.02	0.97 ± 0.01
	Total	0.88 ± 0.18	0.90 ± 0.10	0.87 ± 0.11

Tabelle 6.9: Ergebnisse zum Vergleich der Leistung von VGG-16 auf dem OC Datensatz.

Methode	Klasse	Sensitivität	Präzision	F_1-Score
VGG-16	A	0.23 ± 0.19	0.87 ± 0.17	0.31 ± 0.17
	D	0.65 ± 0.11	0.49 ± 0.07	0.55 ± 0.04
	F	0.52 ± 0.05	0.43 ± 0.12	0.46 ± 0.08
	H	0.50 ± 0.11	0.58 ± 0.18	0.51 ± 0.06
	S	0.10 ± 0.10	0.48 ± 0.26	0.16 ± 0.15
	U	0.96 ± 0.31	0.57 ± 0.18	0.69 ± 0.13
	Total	0.49 ± 0.31	0.57 ± 0.16	0.45 ± 0.19

A, S und F kann davon ausgegangen werden, dass ca. 50 Prozent der Samples aus Klasse A sich auf die Klassen F und S verteilt haben. Für den direkten Vergleich können die Ergebnisse in Tabelle 6.4 für den ASF Merkmalssatz herangezogen werden.

Das VGG-16 Netz erreicht hervorragende Ergebnisse in der Sensitivität der Klassen F, S und U, wohingegen die Präzision schwächer ausfällt. Im direkten Vergleich schneidet der ASF Merkmalssatz in Tabelle 6.4 unter Verwendung des XGBoost Klassifikators besser ab, da sowohl Sensitivität, als auch Präzision in ähnlicher Höhe vorliegen. Die schwächste Klasse des VGG-16 Netzes ist A.

Die Ergebnisse des Tests mit der OC Datenbank sind in Tabelle 6.9 enthalten. Hier fallen die grundsätzlich geringeren Werte für die Sensitivität und Präzision auf. Diese entstehen aufgrund des geringen Kontrasts der Bilddaten der OC-Datenbank. Besonders gravierend ist der Sensitivitätswert für die Klasse S. Die geringe Sensitivität zeigt, dass nur zufällig Samples korrekt der Klasse S zugeordnet werden. Das lässt sich aufgrund der großen Ähnlichkeit der Bilder der Klasse A, S und F erklären.

Tabelle 6.10: Ergebnisse zum Vergleich der Leistung von ASF und VGG-16 auf den vereinten Datensätzen CK und OC.

Methode	Klasse	Sensitivität	Präzision	F_1-Score
ASF	A	0.78 ± 0.07	0.76 ± 0.12	0.76 ± 0.05
	D	0.83 ± 0.06	0.69 ± 0.17	0.74 ± 0.08
	F	0.75 ± 0.06	0.89 ± 0.15	0.76 ± 0.07
	H	0.85 ± 0.02	0.93 ± 0.03	0.89 ± 0.02
	S	0.79 ± 0.03	0.88 ± 0.03	0.83 ± 0.03
	U	0.83 ± 0.03	0.96 ± 0.02	0.89 ± 0.01
	Total	0.81 ± 0.03	0.84 ± 0.10	0.81 ± 0.06
VGG-16	A	0.33 ± 0.11	0.82 ± 0.10	0.46 ± 0.10
	D	0.67 ± 0.11	0.74 ± 0.11	0.69 ± 0.06
	F	0.38 ± 0.10	0.75 ± 0.06	0.49 ± 0.08
	H	0.70 ± 0.10	0.87 ± 0.03	0.77 ± 0.06
	S	0.83 ± 0.06	0.76 ± 0.06	0.79 ± 0.04
	U	1.00 ± 0.00	0.49 ± 0.09	0.66 ± 0.07
	Total	0.65 ± 0.26	0.74 ± 0.13	0.64 ± 0.14

Das VGG-16 Netz schneidet auf dem OC Datensatz sehr schlecht ab, obwohl für die Klasse U eine sehr hohe Sensitivität erreicht wird. Auch hier ist es auffällig, dass die Präzision sehr viel schwächer ist als die Sensitivität, wodurch das Gesamtergebnis sehr schlecht wird. Die schwächste Klasse ist S, wo nur knapp 10 Prozent der Samples korrekt erkannt werden. Die Erkennungsraten bei dem ASF Merkmalssatz in Tabelle 6.7 weisen eine hohe Präzision auf und sind in einem ähnlichen Bereich verteilt. Lediglich die Klassen D und F besitzen schwache Präzisionsraten.

Um stabilere Ergebnisse zu erreichen, ist ein Experiment unter Verwendung der gemischten Datenbanken CK und OC durchgeführt worden. Das ergab eine Menge von 2766 Bildern, die entsprechend der Tabelle 6.1 verteilt sind. Die Ergebnisse des Tests in den vereinten Datenbanken sind in Tabelle 6.10 für beide Methoden ASF und VGG-16 dargestellt.

Ausgehend von den Ergebnissen der OC Datenbank in Tabelle 6.9 konnten die Ergebnisse für beide Methoden signifikant verbessert werden. Die Ergebnisse der CK Datenbank werden nicht erreicht, jedoch ist die Erkennungsrate für die Klasse U für das VGG-16 Netz perfekt. Die Präzision liegt ungefähr bei 50 Prozent. Das bedeutet, dass deutlich mehr Samples der anderen Klassen der Klasse U zugeordnet wurden.

Tabelle 6.11: Genauigkeit der betrachteten Verfahren unter Verwendung der einzelnen Datenbanken CK und OC und der vereinten Datenbank CK mit OC.

Modell	CK	OC	CK und OC
ASF	0.93 ± 0.01	0.72 ± 0.03	0.81 ± 0.01
VGG16	0.89 ± 0.02	0.49 ± 0.06	0.66 ± 0.05
Lopes et al. [Lop17]	0.97	–	–
Liu et al. [Liu15]	0.92	–	–
Mollahosseini et al. [Mol16]	0.93	–	–
Liu et al. [Liu16]	0.95	0.79	–

Im Vergleich zur alleinigen Verwendung des OC Datensatzes sind die Erkennungsraten des ASF Merkmalssatzes gestiegen. Dabei konnten hohe Raten bei der Präzision beibehalten werden. Für die Klassen D und F konnten die Präzisionsraten verbessert werden. Insgesamt ergaben sich demnach für beide Methoden verbesserte Raten in allen Bereichen. Zur vollständigen Betrachtung der Ergebnisse werden die Genauigkeiten verglichen. In Tabelle 6.11 sind die Genauigkeiten für alle Datenbanken und Methoden dargestellt.

Aus den Genauigkeitswerten ist erkennbar, dass die ASF Merkmale mit den Ergebnissen aus dem Deep Learning Umfeld äquivalent sind. In einzelnen Fällen konnten bessere Ergebnisse erziehlt werden. Im Fall der OC Datenbank sind vergleichbare Ergebnisse erreicht worden. Die guten Ergebnisse der Deep Learning Verfahren werden mit einem hohen Aufwand für die Vorverarbeitung erkauft. Der Aufwand ist viel höher als der Aufwand der Vorverarbeitung, der für die ASF Merkmale betrieben wird. Des Weiteren hängt die Genauigkeit der Klassifikation basierend auf den ASF-Merkmalen von der Güte der Landmarkenextraktion ab.

6.3 Zusammenfassung

In Kapitel 6 wurde die Erkennung von emotionalen Gesichtsausdrücken in statischen Bilddaten behandelt. Dazu wurde zunächst die Problemstellung erläutert und die Erkennung des emotionalen Gesichtsausdrucks als Mustererkennungsproblem erläutert. Die Darstellung als Mutererkennungsproblem basiert darauf, dass aus den Bildern Merkmale extrahiert werden, aus denen

bezogen auf die zuzuordnenden Klassen Muster gelernt werden. Die Wiedererkennung der Muster resultiert in der Zuordnung des Merkmalsvektors zu einer der Klassen entsprechend den Basis-Emotionen nach Ekman [Ekm99]. Das Kapitel behandelt die in dieser Arbeit verwendeten Merkmale, die zur Klassifikation der emotionalen Gesichtsausdrücke verwendet werden. Die Merkmale werden in zwei Schritten generiert: der erste Schritt umfasst die Extraktion von Landmarken aus den Bildern; im zweiten Schritt erfolgt die Extraktion von Winkel- und Größenmerkmale.

Für die Extraktion der Landmarken wird zunächst eine Reihe von Vorverarbeitungsschritten ausgeführt, zunächst die Erkennung des Gesichts und die Reduzierung des Ausschnitts auf das Gesicht. Dann wird der Ausschnitt skaliert, um eine einheitliche Größe zu garantieren. Als abschließender Prozessschritt erfolgt eine Korrektur des Gammawerts, um die Helligkeit des Bildes anzupassen. Ausgehend von diesen Bildtransformationen erfolgt die Anpassung des 68-Punkte umfassenden Landmarkenmodells an das Eingabebild.

Der zweite Schritt, die Extraktion und Berechnung der Merkmale, wird unter Verwendung der Landmarken ausgeführt. Das strukturelle Wissen über das Gesicht und die Bewegungen der Gesichtsmuskulatur werden ausgenutzt, um Winkelmerkmale zu generieren. Die Winkel werden aus dem Schnitt von Geraden berechnet. Die Geraden dienen der Approximation verschiedener Gesichtsstrukturen sowie der Modellierung verschiedener Verbindungen zwischen Gesichtsregionen. Um eine fundierte Auswahl der Winkel zu ermöglichen, erfolgte zunächst eine Analyse der Varianz zwischen den einzelnen Emotionsklassen. Dieser Analyse-Schritt ermittelte die Winkel, die die höchste Varianz vorwiesen und somit eine hinreichend große Variabilität zwischen den Klassen ermöglicht. Neben Winkelmerkmalen wurden aufgrund der Form der emotionalen Gesichtsausdrücke Größenmerkmale extrahiert. Betrachtet man verschiedene Gesichtsausdrücke, fällt auf, dass die Größe des Munds und der Augen starken Variationen unterliegt. Die Größe der Augen bezieht sich hierbei auf die Größe des sichtbaren Auges. Um die Größe dieser beiden Gesichtsregionen zu extrahieren, erfolgt eine Approximation dieser Formen mittels Ellipsen. Hierzu werden im Fall der Augen alle Punkte des Auges verwendet und im Fall des Munds nur jene Punkte ausgewählt, die den äußeren Rand

der Lippen betreffen. Durch die Least-Squares-Methode erfolgt die Schätzung der Ellipsenparameter. Zur Berechnung der Größe wird das Verhältnis der beiden Halbachsen verwendet. Hierzu wird die Länge der kurzen Halbachse durch die Länge der großen Halbachse geteilt, um einen Wert kleiner eins zu erhalten. Letztlich beschließt die Extraktion der Größen beider Augen und des Mundes die Merkmalsgenerierung.

Durch die Extraktion der Winkel- und Größenmerkmale erfolgt eine Reduktion des Merkmalsvektors. Geht man von dem 68 Punkte umfassenden Landmarkenmodell aus, so besteht der zugehörige Merkmalsvektor aus 136 Einzelmerkmalen, aufgrund der x- und y-Koordinaten der Landmarken. Aus diesen 68 Punkten wurden insgesamt 26 Winkel- und drei Größenmerkmale gewonnen. Das entspricht 29 Einzelmerkmalen für den Merkmalsvektor. Der extrahierte Merkmalsvektor entspricht einer Reduktion der Dimension auf 21.32 Prozent des ursprünglichen Merkmalsvektors.

Für die Klassifikation wird in dieser Arbeit der XGBoost-Klassifikator eingesetzt. Um bestmögliche Ergebnisse zu erzielen, wurde in der Experimentsektion eine Parameteroptimierung etabliert, bei der unter Verwendung von Kreuzvalidierung die Parametereinstellung optimiert wurde. In den Experimenten wurden zwei Datenbanken mit emotionalen Gesichtsausdrücken verwendet: die Cohn-Kanade+ Datenbank (CK) und die Oulu-Casia Datenbank (OC). Beide Datenbanken enthalten Sequenzen von Bildern, in denen vom neutralen Gesichtsausdruck auf einen der sechs Gesichtsausdrücke der Basisemotionen übergegangen wird. Aus diesen Sequenzen wurden jeweils die drei letzten Bilder entnommen und für die Klassifizierung vorbereitet. Zusätzlich wurde von jedem Probanden, der in der Datenbank enthalten ist, ein neutrales Bild entnommen. Somit konnte das Sechs- und Siebenklassenproblem bearbeitet werden, wobei im Sechsklassenproblem alle sechs Basisemotionen enthalten sind und im Siebenklassenproblem die Neutrale Klasse hinzugefügt wird. Für jede Emotionsklasse wurde jeweils ein Sechs- und ein Siebenklassenmodell generiert. Es wurden vier Modelle erzeugt. Der Modellerzeugung geht immer eine Parameteroptimierung voran. Zusätzlich erfolgte zu den Einzelklassentests noch ein Experiment mit einer Rejection-Option als Vorbereitung für die Verwendung des Klassifikators im dynamischen Modell zur

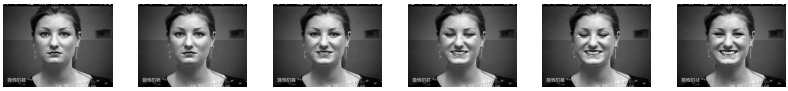
Emotionserkennung. Zum Abschluss der Experimentsektion erfolgte ein Vergleichstest mit einem neuronalen Netz zur Emotionsklassifikation. In diesem Vergleichstest wurde das VGG-16 Netz von Simonyan et al. [Sim14] verwendet. Das Netz wurde mit den ImageNet-Gewichten initialisiert und die letzte Schicht wurde durch eine Dropout-Schicht und eine voll vernetzte Softmax-Schicht ersetzt. Die ImageNet-Gewichte wurden in den unteren Schichten fixiert und zur Feinjustierung des Netzes wurden die Gewichte der oberen Schichten mit den neuen Daten trainiert. In den Experimenten hat sich gezeigt, dass ein großes Maß an Vorverarbeitungsschritten notwendig ist, um den Stand der Technik nachzubilden. Es muss sehr viel Aufwand betrieben werden, um gute Ergebnisse zu erzielen. Der in dieser Arbeit etablierte Merkmalssatz konnte vergleichbare Ergebnisse erzielen wie der Stand der Technik. Mit einer fundierten Merkmalsgenerierung können vergleichbare, bis hin zu besseren Ergebnisse erzielt werden, als es mit neuronalen Netzen möglich ist. Die Ergebnisse sind damit kontrollierbarer und es kann auf verschiedene Klassifikatoren zurückgegriffen werden. Zusätzlich muss deutlich weniger Aufwand aufgebracht werden, um die Daten vorzuverarbeiten.

Dieses Kapitel hat gezeigt, dass mit Hilfe von Methoden der Mustererkennung sehr gute Ergebnisse für die Aufgabe der Emotionserkennung auf Basis von Gesichtsbildern erreicht werden können. Außerdem konnte gezeigt werden, dass die Ergebnisse vergleichbar sind mit dem Stand der Technik, der durch neuronale Netze dominiert wird. Ausgehend von dem Merkmalsmodell sind somit die Grundsteine gelegt, um Beobachtungen für ein dynamisches Modell zur Verfolgung des emotionalen Zustands zu generieren. Das folgende Kapitel widmet sich auf Basis dieser Beobachtungen der Erzeugung eines dynamischen Modells zur Verfolgung des emotionalen Zustands.

7 Tracking von Emotionszuständen

Im Verlauf eines Gesprächs kann sich der emotionale Zustand eines Menschen ändern. Eine solche Änderung kann aufgrund einer traurigen Nachricht während eines Gesprächs entstehen, oder aufgrund der Mitteilung eines Gewinns. Damit man einfühlend auf den Gesprächspartner eingehen kann, sollte die Änderung des emotionalen Zustands erfasst werden. Der neue emotionale Zustand wird sich in einer fließenden Bewegung entwickeln. Eine solche fließende Bewegung ist in Abbildung 7.1 dargestellt. Darin wird die Entwicklung ausgehend von dem neutralen Gesichtsausdruck in den emotionalen Ausdruck für Freude gezeigt.

Damit ein automatisches System in der Lage ist, die Anbahnung einer Emotion oder den Verlauf einer Emotion zu tracken muss der dynamische Kontext der Emotion betrachtet werden. Im Stand der Technik in Abschnitt 2.4 werden multimodale Merkmale verwendet, um ein Tracking des emotionalen Zustands zu erreichen. Im Gegensatz dazu verwendet diese Arbeit ausschließlich Gesichtsmerkmale für die Erkennung des emotionalen Zustands. Die verwendeten Merkmale sind die ASF-Merkmale aus dem Abschnitt 2.3, die einen aus 29 Komponenten bestehenden Deskriptor für den Gesichtsausdruck eines Menschen definieren. Die Einzelmerkmale setzen sich zusammen aus Winkeln und Größenverhältnissen zwischen einzelnen Teilbereichen eines Gesichts. Ein weiterer Unterschied zum Stand der Technik besteht in der Verwendung eines Zustandsraummodells für die Realisierung des Trackers



(a) (b) (c) (d) (e) (f)

Abbildung 7.1: Transfer des neutralen Gesichtsausdrucks in einen freudigen Gesichtsausdruck, dargestellt von Subject S106 aus der Cohn-Kanade+-Datenbank (©Jeffrey Cohn).

für emotionale Zustände. Für das Zustandsraummodell wird der emotionale Zustand im Valenz-Erregungs-Raum (VA-Raum) repräsentiert. Die Abbildung einer ASF-Messung in den VA-Raum wird durch einen Gaußprozess erreicht. Das Training des GP erfolgt überwacht. Die durch den GP in den VA-Raum übertragenen Messungen werden in einem Kalman Filter verarbeitet. Um Drift zu verhindern, wird eine Nebenbedingung mit der Unscented Transformation auf den Zustandsraum angewendet.

Im Folgenden wird das formelle Problem der dynamischen Zustandsschätzung für emotionale Zustände definiert. In Abschnitt 7.2 wird die auf Gaußprozessen basierende Abbildung der ASF-Merkmale in den VA-Raum behandelt. Die eigentliche Zustandsschätzung und die Einführung der Nebenbedingung sind Gegenstand von Abschnitt 7.3. In Abschnitt 7.4 werden numerische Ergebnisse zur Bewertung des dynamischen Ansatzes präsentiert. Das Kapitel schließt mit der Diskussion ab.

7.1 Problemformulierung der dynamischen Zustandsschätzung für emotionale Zustände

Die Daten für die dynamische Zustandsschätzung beinhalten Sequenzen, die die Initiierung einer Emotion zeigen. Die Initiierung bezeichnet die Sequenz ausgehend von einem neutralen Gesichtsausdruck, die in einem vollen emotionalem Gesichtsausdruck endet. Der hier präsentierte Tracker soll in der Lage sein, den emotionalen Zustand während der gesamten Sequenz zu verfolgen. Das hier verwendete Trackingmodell verwendet die sechs durch Ekman in [Ekm99] definierten Basisemotionen: *Wut* (A), *Ekel* (D), *Furcht* (F), *Freude* (H), *Traurigkeit* (S) und *Überraschung* (U).

Um eine vereinfachte Zustandsrepräsentation zu erhalten, wird das Vorgehen aus Al-Hamadi et al. [Al-16] verwendet. Darin werden Emotionsklassen durch spezifische Punkte im Valenz-Erregungs-Raum (VA-Raum) repräsentiert. Der aktuelle emotionale Zustand kann durch einen zweidimensionalen

Vektor $\underline{x}_k = [v, a]^T \in \mathbb{R}^2$ dargestellt werden, wobei v für den Valenzwert und a für den Erregungswert stehen. Die Darstellung im VA-Raum entspricht einem kontinuierlichen Zustandsvektor. Eine diskrete Zustandsrepräsentation müsste um eine Komponente erweitert werden, die die Stärke der Emotion repräsentiert. Dieser Umstand ist implizit in der VA-Raum-Darstellung enthalten. Zusätzlich bietet die kontinuierliche Darstellung den Vorteil eines robusteren Trackings.

Das eigentliche Trackingproblem wird als Bayes'sches Problem der Zustandsschätzung definiert. In dieser Problemklasse wird angenommen, dass der Zustand ein Zufallsvektor $\underline{\mathbf{x}}$ mit zugehöriger Wahrscheinlichkeitsverteilung $P(\underline{\mathbf{x}})$ ist. Für den Tracker von emotionalen Zuständen wird angenommen, dass die zugehörige Wahrscheinlichkeitsverteilung des Zustands eine Normalverteilung ist, sodass

$$\underline{\mathbf{x}} \sim \mathcal{N}\left(\underline{\mu}, \mathbf{C}\right) \tag{7.1}$$

gilt. $\underline{\mu}$ ist der Mittelwert der Verteilung und $\mathbf{C}$ die zugehörige Kovarianzmatrix.

Die Schätzung des emotionalen Zustands erfolgt in zwei alternierenden Schritten: einem Prädiktions- und einem Filterschritt. Im Filterschritt wird der Schätzwert $\underline{\mathbf{x}}^e_{k-1}$ des Zeitschritts $k-1$ in den Zeitschritt k überführt, wobei für den diskreten Zeitschritt $k = 0,1,\ldots$ gilt. Im Sinne der verwendeten Bildsequenzen korrespondiert der diskrete Zeitpunkt k mit dem k-ten Bild der Sequenz. Das Resultat des Prädiktionsschritts wird durch den prädizierten Zustand $\underline{\mathbf{x}}^p_k$ repräsentiert. Durch $\underline{\mathbf{x}}^p_k$ ist die Eingabe des Filterschritts definiert. Im Filterschritt wird die Information, die aus dem k-ten Bild extrahiert wurde, dazu verwendet um zusammen mit dem prädizierten Zustand eine verbesserte Schätzung $\underline{\mathbf{x}}^e_k$ zu erzeugen.

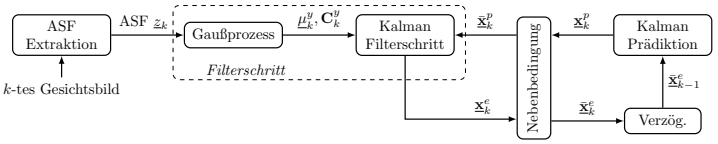


Abbildung 7.2: Ablaufplan des Trackers für emotionale Zustände.

In Abbildung 7.2 wird der Bayes'sche Filterprozess dargestellt. Um diesen Prozess vollständig zu realisieren müssen einige Anforderungen erfüllt werden. Zunächst müssen die ASF-Messungen in dem VA-Raum abgebildet werden. Im Anschluss muss ein Messmodell abgeleitet werden, durch das die VA-Messungen auf den Systemzustand abgebildet werden können. Es muss ein Systemmodell erzeugt werden, durch dass sich der Zustand in den nächsten Zeitschritt überführen lässt. Letztendlich muss die Ungleichheitsnebendingung eingeführt werden, durch die eine Drift des Systemzustands verhindert wird.

7.2 Transformation der ASF-Merkmale in den Valenz-Erregungs-Raum

Für die Zustandsschätzung ist eine kontinuierliche Zustandsrepräsentation vorteilhaft. Daher ist sie einer diskreten Klassenrepräsentation vorzuziehen. Durch das in Al-Hamadi et al. [Al-16] vorgeschlagene Vorgehen können die sechs Grundemotionen Wut (A), Ekel (D), Angst (F), Freude (H), Traurigkeit (S) und Überraschung (U) und zusätzlich die neutrale Emotion (N) als zweidimensionale Vektoren $\underline{x}$ im VA-Raum repräsentiert werden. Die zugehörigen Vektoren werden durch Tabelle 7.1 festgelegt. In Abbildung 7.3 sind die Punkte im Einheitskreis eingezeichnet.

Die Messungen liegen in Form von ASF-Merkmalsvektoren $\underline{z} \in \mathbb{R}^{29}$ vor, wie sie in Abschnitt 2.3 eingeführt wurden. Diese Merkmalsvektoren werden aus den Bildern extrahiert und müssen dann in den VA-Raum abgebildet werden. Es gibt keine analytisch erklärbare Funktion, die diese Abbildung beschreibt. Aus diesem Grund wird ein GP verwendet, der die Abbildung eines

Tabelle 7.1: Zuweisung der Valenz und Erregungs-Werte zu den sieben Basis-Emotionsklassen.

Emotion	$\underline{\mathbf{p}} = [\mathbf{v}, \mathbf{a}]^{\mathbf{T}}$
Neutral (N)	$[0, 0]$
Wut (A)	$[-0.89, 0.45]$
Ekel (D)	$[-0.89, -0.45]$
Furcht (F)	$[-0.45, 0.89]$
Freude (H)	$[0.89, 0.45]$
Trauer (S)	$[-0.45, -0.89]$
Überraschung (U)	$[0.24, 0.97]$

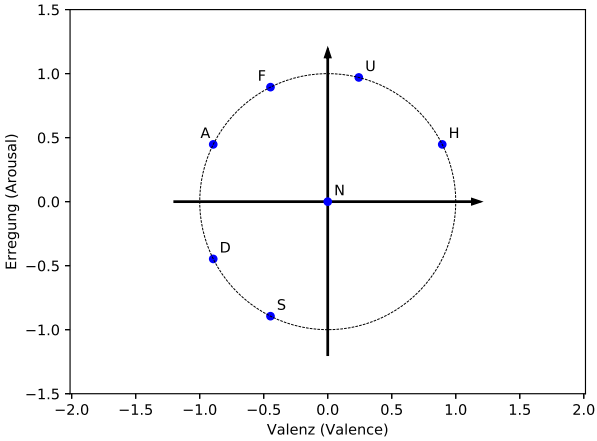


Abbildung 7.3: Abbildung der Basis-Emotionen auf den Valenz-Erregungs-Raum (VA-Raum) zur Ermittlung kontinuierlicher Messwerte für die Emotionserkennung.

ASF-Vektors $\underline{z}$ in den VA-Raum lernt. Durch dieses Vorgehen kann eine Messung im VA-Raum generiert werden, die direkt mit dem emotionalen Systemzustand $\underline{\mathbf{x}}$ verarbeitet werden kann. Hierdurch kann die Verwendung eines linearen Bayes'schen Schätzers, wie zum Beispiel dem Kalman Filter, verwendet werden. Hier muss kein approximativer, nichtlinearer Bayes'scher Schätzer verwendet werden.

Die Problemstellung, einen ASF-Vektor $\underline{z}$ in einen VA-Vektor $\underline{x}$ abzubilden, kann als Regressionsproblem betrachtet werden. Die ASF-Vektoren bilden einen kontinuierlichen Vektorraum, ebenso wie auch der VA-Raum ein kontinuierlicher Vektorraum ist. Zur Lösung des Regressionsproblems wird in dieser Arbeit die GP Regression verwendet. Der Vorteil der GP Regression besteht darin, dass ein GP Regressor Unsicherheiten erfassen kann, die aufgrund von Rauschen oder partiellen Überdeckungen resultieren.

Ein GP besitzt eine Vielzahl von günstigen Eigenschaften. Zum Beispiel kann ein GP als Distribution über Funktionen interpretiert werden, da er aus einer unendlichen Anzahl von Gaußverteilten Zufallsvariablen zusammengesetzt wird. Daraus resultiert die praktische Eigenschaft, dass jede marginale Verteilung eines GP wiederum Gaußverteilt ist. Eine gründliche Einführung in GPs findet sich in Abschnitt 3.4.

Ein GP ist vollständig durch eine Mittelwertfunktion $m(.)$ und eine Kovarianzfunktion $\kappa(.,.)$ definiert. Durch die Kovarianzfunktion können weitere nützliche Eigenschaften hinzugefügt werden, wie Glattheit oder Nicht-Stationarität. In diesem Fall wird der GP dazu verwendet, um die Funktion

$$\underline{\mathbf{y}} = \underline{g}(\underline{z}) + \underline{\boxtimes} \tag{7.2}$$

wird aus Trainingsdaten $\mathcal{D} = \left\{\left(\underline{z}_1, \underline{\mathbf{y}}_1\right), \ldots, \left(\underline{z}_n, \underline{\mathbf{y}}_n\right)\right\}$ abgeleitet und $\underline{g}(.) = [g_1(.), g_2(.)]^T$ ist die latente Abbildung der ASF-Vektoren $\underline{z}$ auf VA-Messungen $\underline{\mathbf{y}}$, wobei $\underline{\boxtimes} \in \mathcal{N}\left(\underline{0}, \mathbf{C}^{\epsilon}\right)$ ein gaußverteiltes, weißes Rauschen mit einer diagonalen Kovarianzmatrix $\mathbf{C}^{\epsilon} = \text{diag}\left(\sigma_1^2, \sigma_2^2\right)$ ist. Die GP Funktionen $g_1(.)$ und $g_2(.)$ sind individuelle Funktionen für die Abbildung auf die Valenz und auf

die Erregung. Das Training erfolgt überwacht anhand der Trainingsdaten $\mathcal{D}$ unabhängig für jeden der beiden GPs. Eine genaue Abhandlung über das Training eines GP kann in Abschnitt 3.4 nachgelesen werden.

Mit der trainierten GP Abbildung $\underline{g}(.)$ kann aus einem gemessenen ASF Vektor $\underline{z}$ eine VA-Messung $\underline{\mathbf{y}} \sim \mathcal{N}\left(\underline{\mu}^y, \mathbf{C}^y\right)$ abgeleitet werden. Die Ableitung erfolgt über den Mittelwertvektor $\underline{\mu}^y = \left[\mu_1^y, \mu_2^y\right]^T$ und die Kovarianzmatrix $\mathbf{C}^y = \text{diag}\left(\left(\sigma_1^y\right)^2, \left(\sigma_2^y\right)^2\right)$. Die Komponenten des Mittelwertvektors werden durch

$$\mu_i^y = \underline{k}^T \cdot \mathbf{K}_i^{-1} \cdot \underline{y}_{\mathcal{D},i} \tag{7.3}$$

bestimmt, wobei $i = 1{,}2$ gilt. Es gilt weiter $\underline{k}^T = \left[\kappa\left(\underline{z}_1, \underline{z}\right), \ldots, \kappa\left(\underline{z}_n, \underline{z}\right)\right]$, sowie $\underline{y}_{\mathcal{D},i}^T = \left[y_{1,i}, \ldots, y_{n,i}\right]$. $\mathbf{K}_i$ wird durch $\mathbf{K}_i = \mathbf{K} + \sigma_i^2 \cdot \mathbf{I}_n$ berechnet. $\mathbf{K}$ ist eine Kernelmatrix, die durch $(\mathbf{K})_{a,b} = \kappa\left(\underline{z}_a, \underline{z}_b\right)$ erzeugt wird, wobei $\forall \underline{z}_a, \underline{z}_b \in \mathcal{D}$ gilt. $\mathbf{I}_n$ ist eine $n \times n$-Einheitsmatrix. Die Kernelmatrix wird durch Anwendung der Kernelfunktion $\kappa(.,.)$ auf paarweise ASF-Vektoren aus der Trainingsmenge $\mathcal{D}$ erzeugt. Die Varianz $\left(\sigma_i^y\right)^2$ kann durch

$$\left(\sigma_i^y\right)^2 = \kappa\left(\underline{z},\underline{z}\right) - \underline{k}^T \cdot \mathbf{K}_i^{-1} \cdot \underline{k} \tag{7.4}$$

bestimmt werden. Durch (7.3) und (7.4) können ASF-Vektoren in den VA-Raum transformiert werden.

Alternativ könnte der GP verwendet werden, den Systemzustand im VA-Raum direkt auf eine ASF-Messung abzubilden. Dazu wäre eine Regression vom niedrig-dimensionalen VA-Raum in den hoch-dimensionalen ASF-Raum notwendig. Eine solche Abbildung erfordert genaue Kenntnis der VA-Werte, die den ASF-Messungen entsprechen. Da diese nicht vorliegen, wird in dieser Arbeit darauf verzichtet, diese Transformation als weitere Alternative zu betrachten.

7.3 Herleitung des Trackers mit Nebenbedingung

Der emotionale Zustand wird mit einem Bayes'schen Schätzer getrackt. In einem Bayes'schen Schätzer wird der Systemzustand als Zufallsvektor $\underline{\mathbf{x}}_k$ repräsentiert. k bezeichnet einen diskreten Zeitindex. Es wird angenommen, dass der Systemzustand $\underline{\mathbf{x}}_k$ für jeden Zeitpunkt k gaußverteilt ist. Es existieren ein Mittelwert $\underline{\mu}_k$ und eine Kovarianzmatrix $\mathbf{C}_k$ für jeden Systemzustand. In dieser Arbeit wird ein Kalman Filter für die Bayes'sche Zustandsschätzung verwendet. Mit dem Kalman Filter werden der Mittelwert und die Kovarianzmatrix für den emotionalen Zustand geschätzt. Als Messungen werden Messungen im VA-Raum verwendet, die zuvor durch den GP aus Abschnitt 7.2 aus dem ASF-Raum in den VA-Raum transformiert wurden.

Im Filterschritt wird der prädizierte Systemzustand $\underline{\mathbf{x}}_k^p \sim \mathcal{N}\left(\underline{\mu}_k^p, \mathbf{C}_k^p\right)$ aus dem vorherigen Prädiktionsschritt mit einer Messung im VA-Raum aktualisiert. Der Filterschritt resultiert in einem geschätzten Systemzustand $\underline{\mathbf{x}}_k^e \sim \mathcal{N}\left(\underline{\mu}_k^e, \mathbf{C}_k^e\right)$.

7.3.1 Filterschritt des Trackers

Die Messabbildung kann den Systemzustand direkt mit Messungen im VA-Raum in Verbindung bringen, weil der GP aus dem vorherigen Abschnitt Messungen im VA-Raum liefert.

$$\underline{\mu}_k^y = \underline{\mathbf{x}}_k + \underline{\mathbf{v}}_k \tag{7.5}$$

ist die Messabbildung mit Messrauschen $\underline{\mathbf{v}}_k \sim \mathcal{N}\left(\underline{0}, \mathbf{C}_k^y\right)$. Die Kovarianzmatrix $\mathbf{C}_k^y$ ist ein Resultat der Transformation der ASF-Messung in die VA-Messung $\underline{\mu}_k^y$ mit Kovarianzmatrix $\mathbf{C}_k^y$ durch den GP. Durch Einsetzen der Messabbildung (7.5) in die Filtergleichung des Kalman-Filters kann

durch (3.23) und (3.24) die verbesserte Zustandsschätzung $\underline{\mathbf{x}}_k^e \sim \mathcal{N}\left(\underline{\mu}_k^e, \mathbf{C}_k^e\right)$ berechnet werden, wobei

$$\underline{\mu}_k^e = \underline{\mu}_k^p + \mathbf{G}_k \cdot \left(\underline{\mu}_k^y - \underline{\mu}_k^p\right) \tag{7.6}$$

$$\mathbf{C}_k^e = \mathbf{C}_k^p - \mathbf{G}_k \cdot \mathbf{C}_k^p \tag{7.7}$$

den Mittelwertvektor $\underline{\mu}_k^e$ und die Kovarianzmatrix $\mathbf{C}_k^e$ berechnen. Die Matrix $\mathbf{G}_k$ ist das sogenannte Kalman-Gain aus (3.22).

7.3.2 Prädiktionsschritt des Trackers

Es ist kein Modell bekannt, das die zeitliche Entwicklung des emotionalen Zustands abbildet. Deswegen wird das Constant-Positions Modell (CP-Modell)

$$\underline{\mathbf{x}}_{k+1}^p = \underline{\mathbf{x}}_k^e + \underline{\omega}_k \tag{7.8}$$

mit dem Rauschterm $\underline{\omega}_k \sim \mathcal{N}\left(\underline{0}, \mathbf{C}_k^\omega\right)$ verwendet. Durch das CP-Modell wird die brownsche Molekularbewegung modelliert. Das CP-Modell ist in Abschnitt 3.2.2 beschrieben.

Der Kalman Prädiktionsschritt (3.18) und (3.19) wird auf (7.8) angewendet, um eine Schätzung des prädizierten Zustands $\underline{\mathbf{x}}_{k+1}^p$ aus der aktuellen Schätzung $\underline{\mathbf{x}}_k^e$ zu erhalten. Mittelwert und Kovarianzmatrix von $\underline{\mathbf{x}}_{k+1}^p$ werden durch

$$\underline{\mu}_{k+1}^p = \underline{\mu}_k^e \tag{7.9}$$

$$\mathbf{C}_{k+1}^p = \mathbf{C}_k^e + \mathbf{C}_k^w \tag{7.10}$$

berechnet.

7.3.3 Nebenbedingungen für den Tracker

Aufgrund von verrauschten Messungen kann es vorkommen, dass der geschätzte Zustand den Einheitskreis verlässt, in dem der VA-Raum definiert ist. Ein solcher Zustand muss korrigiert werden, damit der Einheitskreis eingehalten wird. Darum wird durch

$$\underline{\mathbf{x}}^T \cdot \underline{\mathbf{x}} \leq 1 \tag{7.11}$$

eine nichtlineare Nebenbedingung definiert. Der Zustandsvektor $\underline{\mathbf{x}}$ ist ein reeller Zufallsvektor. Deswegen kann (7.11) nicht auf einfachem Weg angewendet werden. Die Herleitung einer analytischen Transformation, die garantiert, dass der größte Teil der Wahrscheinlichkeitsmaße die Nebenbedingung erfüllt, ist im Allgemeinen sehr schwer zu erreichen. Um die Nebenbedingung trotzdem anwenden zu können, wird der von Kandepu et al. in [Kan08] vorgeschlagene Ansatz zur Anwendung rechteckiger Nebenbedingungen in einem Unscented Kalman Filter angewendet.

Zunächst werden Sigmapunkte $\underline{\mathcal{X}}_i$ aus den prädizierten und geschätzten Zuständen $\underline{\mathbf{x}}_k^p$ und $\underline{\mathbf{x}}_k^e$ berechnet. Die Berechnung der Sigmapunkte kann in Abschnitt 3.2.4 nachgelesen werden. Es wird geprüft, ob jeder Sigmapunkt $\underline{\mathcal{X}}_i$ die Nebenbedingung (7.11) erfüllt. Falls ein Sigmapunkt sich außerhalb des Einheitskreises befindet, wird dieser auf den Rand des Einheitskreises projiziert. Durch

$$\underline{\bar{\mathcal{X}}}_i = \frac{\underline{\mathcal{X}}_i}{\|\underline{\mathcal{X}}_i\|} \tag{7.12}$$

wird die Projektion für alle Sigmapunkte außerhalb des Einheitskreises ausgeführt. Somit entsteht eine Menge von korrigierten Sigmapunkten, die durch den Stichprobenmittelwert und die Stichprobenkovarianz korrigierte Schätzungen für $\underline{\mathbf{x}}_k^p$ und $\underline{\mathbf{x}}_k^e$ berechnen.

Die Nebenbedingung wird zweimal angewendet: nachdem der Zustand in den nächsten Zeitschritt prädiziert wurde und nach dem Filterschritt. Dieses Vorgehen garantiert, dass der größte Teil der Wahrscheinlichkeitsmasse des Systemzustands die Nebenbedingung erfüllt.

7.4 Experimente zum Tracking von emotionalen Zuständen

Wie bereits im vorherigen Kapitel werden die beiden Datenbanken Cohn-Kanade+ und Oulu-Casia für die Durchführung der Experimente verwendet. Beide Datenbanken bestehen aus Sequenzen, die mit einem neutralen Gesichtsausdruck beginnen und in einem emotionalen Gesichtsausdruck enden.

7.4.1 Training des GPs

Für das Training des GP wurden 5536 ASF-Vektoren verwendet. Die 5536 ASF-Vektoren wurden aus beiden Datenbanken extrahiert. Es wurden ausschließlich voll ausgebildete emotionale Ausdrücke verwendet. Die Einzelbilder wurden korrekt annotiert. Jeder Emotion wurde der entsprechende VA-Vektor aus Tabelle 7.1 zugeordnet. Die ASF-Vektoren wurden unter Verwendung der Hauptkomponentenanalyse vorverarbeitet, um numerisch stabile Werte zu erhalten. Es wurde eine Hyperparamteroptimierung durch Minimierung der negativen marginalen Log-Likelihood unter Verwendung der Trainingsdaten durchgeführt. Die Trainingsvektoren wurden durch den trainierten GP in den VA-Raum projiziert. Das Ergebnis dieses Tests ist in Abbildung 7.4 abgebildet.

Die ASF-Vektoren werden alle in die direkte Umgebung der korrespondierenden VA-Werte durch den trainierten GP abgebildet. Die Verteilung der Samples um den korrespondierenden VA-Wert kann zur Auswertung der Genauigkeit des GPs herangezogen werden. Beide Datenbanken enthalten keine Informationen über den korrekten VA-Wert für ihre Bilder. Deswegen können

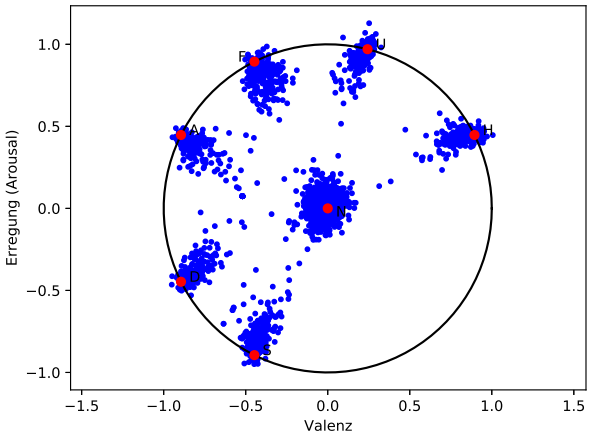


Abbildung 7.4: Abbildung der Trainingssamples durch den GP zum Test des Trainings.

hier keine Bilder verwendet werden, die nur teilweise Gesichtsausdrücke enthalten. Die VA-Werte für die Zwischenbilder konnten nur geschätzt werden und sind daher dünn besetzt. Damit kann eine weitere Begründung für die Entscheidung den GP nicht als Messmodel zu verwenden gegeben werden. Aufgrund der vorgestellten Einschränkungen ist eine Verwendung des GPs zur Abbildung der ASF-Vektoren in den VA-Raum passender.

7.4.2 Tracking des emotionalen Zustands

Zur Auswertung der experimentellen Ergebnisse wird der vorgestellte Tracker für emotionale Zustände mit der Abkürzung GPET bezeichnet. Der GPET wird auf alle Sequenzen der beiden Datenbanken angewendet. Die Recherchen zu dieser Studie haben ergeben, dass kein weiterer Tracker für emotionale Zustände ausschließlich Gesichtsausdrücke verwendet. Daher werden keine Vergleiche zum Stand der Technik aufgeführt.

Abbildung 7.5 zeigt das Ergebnis der Schätzung des GPET mit einer Sequenz aus der Cohn-Kanade+ (CK) Datenbank. Die Sequenz beginnt im neutralen Zustand (N) und entwickelt sich in Richtung der emotionalen Zustands Wut

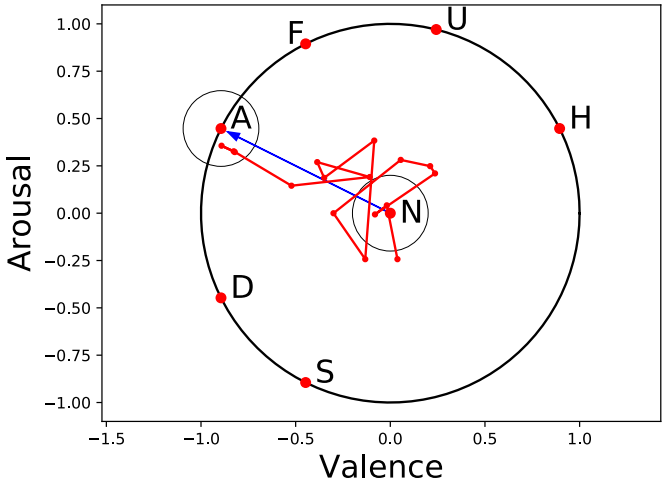


Abbildung 7.5: Durch den GPET geschätzte Sequenz der CK Datenbank.

(A). Die Sequenz wird durch eine stückweise, lineare Funktion dargestellt. Die Punkte der Sequenz stellen die Zustandsschätzungen des GPET dar. Es wird davon ausgegangen, dass der wahre Entwicklungsverlauf einer Emotion einer linearen Funktion entspricht. Im Fall von Abbildung 7.5 würde demnach der wahre Verlauf annähernd eine Gerade zwischen dem neutralen Zustand (N) und dem emotionalen Zustand (A) sein. Zur Auswertung der Güte der Schätzung wird der Abstand der Schätzungen von der angenommenen Geraden bestimmt. Der dargestellte Verlauf hat generell einen geringen Abstand von der angenommenen geraden Linie zwischen A und N.

Die generelle Auswertung des Tests mit den Datenbanken erfolgte durch Bestimmung des Mittelwerts aller Distanzen der individuellen Schätzungen des emotionalen Zustands bezogen auf die angenommene Gerade zwischen dem neutralen und dem finalen emotionalen Zustand. In diesem Experiment wird das Ergebnis des GPET mit dem Ergebnis eines genutzten GPs als Schätzer für den emotionalen Zustand verglichen. Damit sollte der Vorteil der Verwendung eines Bayes'schen Schätzers zusammen mit einem GP verdeutlicht werden. Die Ergebnisse der Schätzung der 443 Sequenzen der CK Datenbank sind in Tabelle 7.2 aufgetragen.

Tabelle 7.2: Ergebnisse der Schätzung des emotionalen Zustands unter Verwendung der CK Datenbank.

Emotion	N	Fehler GPET	Fehler GP
Wut (A)	60	0.1907 ± 0.0974	0.1911 ± 0.0931
Ekel (D)	66	0.1447 ± 0.0736	0.1510 ± 0.0680
Furcht (F)	66	0.1637 ± 0.0981	0.1571 ± 0.0902
Freude (H)	90	0.1467 ± 0.0848	0.1454 ± 0.0778
Traurigkeit (S)	75	0.1790 ± 0.0910	0.1723 ± 0.0819
Überraschung (U)	86	0.0929 ± 0.0511	0.0879 ± 0.0472

Tabelle 7.3: Ergebnisse der Auswertung des emotionalen Trackings der Sequenzen der OC-Datenbank.

Emotion	N	Fehler GPET	Fehler GP
Wut (A)	80	0.2076 ± 0.1005	0.2541 ± 0.1164
Ekel (D)	80	0.1905 ± 0.0927	0.2175 ± 0.0978
Furcht (F)	80	0.1588 ± 0.0834	0.1815 ± 0.0902
Freude (H)	80	0.1620 ± 0.0818	0.2019 ± 0.0978
Traurigkeit (S)	80	0.1737 ± 0.0884	0.1988 ± 0.1013
Überraschung (U)	80	0.1532 ± 0.0668	0.1663 ± 0.0717

Die Auswertung der 480 Sequenzen der OC Datenbank ist in Tabelle 7.3 dargestellt.

Es wurden Kreuzvalidierungen zur Auswertung der mittleren Fehler der Tracker durchgeführt. Die Tabellen 7.2 und 7.3 zeigen, dass eine kombinierte Nutzung eines Bayes'schen Schätzers mit GP Regression ähnliche oder gar signifikant reduzierte Fehler für die Schätzung der Sequenzen erzeugt im Vergleich zur alleinigen Nutzung der GP Regression. Im Vergleich mit der GP Regression sind die Ergebnisse in Tabelle 7.3 erfolgversprechender. Für jeden emotionalen Zustand besitzt die Schätzung des GPET einen signifikant niedrigeren Fehler als die alleinige Nutzung der GP Regression.

Für einen weiteren Test wird der XGBoost Klassifikator aus dem vorherigen Kapitel für die Bild-zu-Bild Schätzung des emotionalen Zustands verwendet. Um die Ergebnisse vergleichen zu können werden verschiedene Maße berechnet:

- Die Bildnummer des ersten Bildes der Sequenz, das korrekt klassifiziert wurde. Der Wert wird in Prozent angegeben, sodass kleinere Werte besser sind als große.

- Die Anzahl der korrekt klassifzierten Bilder einer Sequenz in Prozent. Die ersten 30% einer Sequenz werden der neutralen Klasse zugeordnet und die restlichen dem finalen emotionalen Zustand der Sequenz.
- Die Anzahl der Klassenwechsel innerhalb einer Sequenz, kleinere Werte sind besser.
- Die Anzahl der Sequenzen bei denen der Tracker kein Ergebnis lieferte, kleinere Werte sind besser.

Die VA Schätzungen des GPET werden durch Ermittlung der Mahalanobis-Distanz zwischen dem geschätzten Systemzustand und allen anderen Klassen im VA-Raum auf die Klassen abgebildet. Die Mahalanobis-Distanz ist durch

$$d\left(\underline{x},\underline{y}\right) = \sqrt{\left(\underline{x}-\underline{y}\right)^{T} \cdot \Sigma^{-1} \cdot \left(\underline{x}-\underline{y}\right)} \tag{7.13}$$

gegeben. Die Mahalanobis-Distanz gewichtet den Abstand mit der inversen der Kovarianzmatrix Σ. Eine Schätzung wird der Klasse zugeordnet, für die die berechnete Mahalanobis-Distanz am geringsten ist. Die Ergebnisse des XGBoost Klassifikators für beide Datenbanken sind in Tabelle 7.4 aufgetragen.

Zum Vergleich der Ergebnisse ist die Auswertung der Kennzahlen für den GPET in Tabelle 7.5 enthalten.

Der XGBoost Tracker hat die meisten fehlgeschlagenen Sequenzen, insbesondere wenn man die OC-Datenbank Ergebnisse betrachtet, siehe Tabelle 7.4. Im Vergleich dazu schneidet der GPET besser ab. Die Anzahl der korrekt klassifizierten Einzelbilder einer Sequenz ist höher als bei dem XGBoost basierten Ansatz. Sehr auffällig ist die geringe Fehleranzahl für die OC-Datenbank. Allerdings benötigt GPET länger bis die korrekte Klasse getrackt wird. Diese Geschwindigkeit wäre höher, wenn für die Zwischenbilder eine korrekte Zuordnung des emotionalen Zustands im VA-Raum zugeordnet wären. In diesem Fall könnte ein genaueres Training der GP Regression erfolgen, wodurch

Tabelle 7.4: Auswertung der Trackingqualität des XGBoost Klassifikators für beide Datenbanken inklusive der Standardabweichungen.

Klasse	Erste korrekte Schätzung	Anteil korrekter Bilder in %	Klassen-wechsel	Fehler
1-CK	59.46 ± 20.00	39.13 ± 10.61	5.67 ± 5.67	52
2-CK	59.37 ± 18.55	60.62 ± 19.66	4.24 ± 4.24	12
3-CK	50.14 ± 23.64	65.41 ± 19.68	4.77 ± 4.77	22
4-CK	40.12 ± 16.42	68.90 ± 21.40	4.23 ± 2.45	4
5-CK	52.88 ± 22.89	45.00 ± 14.72	5.49 ± 4.54	43
6-CK	53.40 ± 21.26	64.95 ± 23.02	3.66 ± 2.42	25
1-OC	33.93 ± 17.71	52.74 ± 14.70	6.25 ± 5.48	73
2-OC	49.00 ± 17.23	45.90 ± 15.28	5.31 ± 3.00	49
3-OC	49.15 ± 13.70	50.00 ± 17.48	4.56 ± 2.87	64
4-OC	41.53 ± 12.91	47.58 ± 16.42	4.01 ± 2.24	46
5-OC	40.37 ± 9.72	53.92 ± 17.30	4.96 ± 3.22	73
6-OC	50.75 ± 13.61	46.95 ± 14.66	3.61 ± 2.21	65

Tabelle 7.5: Auswertung der Trackingqualität für den GPET inklusive der Standardabweichungen.

Klasse	Erste korrekte Schätzung	Anteil korrekter Bilder in %	Klassen-wechsel	Fehler
1-CK	51.48 ± 25.01	49.83 ± 17.93	5.43 ± 4.30	26
2-CK	51.57 ± 22.31	57.26 ± 18.25	3.59 ± 1.78	19
3-CK	55.76 ± 21.23	57.67 ± 18.90	3.94 ± 3.49	33
4-CK	50.46 ± 20.24	71.94 ± 19.48	3.52 ± 2.46	19
5-CK	47.26 ± 24.12	40.25 ± 10.18	4.60 ± 3.35	45
6-CK	52.83 ± 18.29	77.86 ± 16.80	2.42 ± 1.19	7
1-OC	58.46 ± 21.33	57.79 ± 13.88	6.44 ± 3.42	3
2-OC	54.61 ± 23.37	57.40 ± 15.68	6.36 ± 3.24	2
3-OC	54.50 ± 21.19	62.27 ± 15.45	6.78 ± 3.31	2
4-OC	54.08 ± 18.58	66.98 ± 15.66	5.24 ± 2.74	3
5-OC	56.89 ± 23.14	56.33 ± 15.54	6.70 ± 3.46	0
6-OC	54.57 ± 19.40	67.92 ± 18.88	5.19 ± 2.74	1

eine Verbesserung des Trackings mit GPET erreicht werden könnte. Basierend auf den präsentierten Ergebnissen ist eine kombinierte Verwendung von Bayes'scher Zustandsschätzung mit GP Regression zu empfehlen.

7.5 Zusammenfassung

Das siebte Kapitel stellt einen neuen Ansatz zum Tracking von emotionalen Zuständen basierend auf der Beobachtung von Gesichtsausdrücken vor. Die ASF-Merkmale aus Kapitel 6 werden aus Bildern extrahiert und mit Gaußprozess Regression in den Valenz-Erregungs-Raum abgebildet. Die so abgebildeten Merkmale werden als Messungen für einen Bayes'schen Zustandsschätzer verwendet. Es wird eine nichtlineare Nebenbedingung in Form einer Ungleichung in den Zustandsschätzer eingebracht, um die Drift des Zustands in einem Bereich außerhalb des Definitionsbereichs des Valenz-Erregungs-Raums zu verhindern.

In Experimenten mit Sequenzen der beiden Emotionsbilddatenbanken Cohn-Kanade+ und Oulu-Casia konnte gezeigt werden, dass der vorgestellte Bayes'sche Schätzer in Kombination mit einer GP Regression einen robusten Trackingalgorithmus darstellt. Es konnte gezeigt werden, dass die Trackingergebnisse stabiler und größtenteils besser waren als die dazu im Vergleich erzielten Ergebnisse mit dem Bild-zu-Bild-Tracker auf Basis einer alleinigen Gaußprozess Regression und einem XGBoost-Klassifikator.

Das Ziel bereits die Anbahnung eines emotionalen Gesichtsausdrucks zu erfassen ist ab 50% der Sequenz bei allen Emotionen gelungen. Allerdings zeigt die Unsicherheit der VA-Raum Abbildung insbesondere im Anfangsstadium einer Emotionssequenz, dass hier noch Verbesserungspotential nötig ist. Durch ein validiertes Labeling der Anbahnungssequenzen könnte hier eine deutliche Verbesserung erreicht werden.

In diesem Kapitel wurde somit ein erstes funktionierendes Tracking-Verfahren von Emotionen in Gesichtsbildern vorgestellt.

8 Zusammenfassung

Menschen werden in vielen Situationen beobachtet. Die vorliegende Arbeit *Tracking von Menschen und menschlichen Zuständen* stellt Ansätze für die Beobachtung von Menschen und ihren Zuständen vor. Der Fokus der Arbeit liegt dabei auf der Kommunikationsebene. Menschliche Kommunikation findet häufig nonverbal statt und hierfür ist der Gesichtsausdruck von großer Bedeutung. Daher werden im Verlauf der Arbeit zunächst Ansätze für das Gesichtstracking geliefert. Hierfür wurde ein Modell aufgestellt, das eine Verfolgung eines 68 Punkte umfassenden Landmarkenmodells ermöglicht. In einem stochastischen Filtermodell wurden die Landmarken als Zustandsvariablen eingesetzt. Um eine Gleichheits-Nebenbedingung einzuführen, wurde das Unscented Kalman Filter verwendet, welches verhindert, dass die Landmarken in unkoordinierter Weise auseinander driften. Per Hauptkomponentenanalyse wurde ein Normmodell trainiert, das ausreichend Varianz enthält, um alle möglichen Gesichtsausdrücke zu erhalten. Die Landmarken wurden auf den nächsten passenden Gesichtsausdruck projiziert. Somit konnte ein robuster Schätzer generiert werden, der immer ein gültiges Landmarkenmodell enthält.

Im Kapitel 5 wurde die Beobachtungsleistung von Probanden beurteilt und als Aspekt der menschlichen Leistungsfähigkeit verwendet. Mit dem Ziel, die Existenz eines Trainingseffekts nachzuweisen, wurden zwei Studien durchgeführt. In der ersten Studie wurde die Leistung vor und nach einem Training beurteilt. Die zweite Studie verwendet künstliche Bilderkennungsalgorithmen, um Entscheidungshilfen bereitzustellen; zusätzlich wurde hier die Aufmerksamkeit durch eine sekundäre Aufgabe beurteilt. Die Studien konnten den Trainingseffekt nicht eindeutig nachweisen. Allerdings konnten Indikatoren abgeleitet werden, die bei der Erstellung von Beobachtungstrainings helfen.

Das Extrahieren von Informationen aus dem Landmarkenmodell wurde im Kapitel 6 auf die Erkennung von emotionalen Zuständen abgebildet. Zur Charakterisierung des emotionalen Zustands wurde insgesamt eine Menge von 26 Winkeln und 3 Größeninformationen extrahiert. Dieser Merkmalssatz lieferte einen robusten Ansatz zur Schätzung des emotionalen Zustands vom Ausgangspunkt eines Landmarkenmodells, das aus statischen Bilddaten extrahiert wurde. In Zusammenarbeit mit dem XGBoost-Klassifikator liefert diese Informationsquelle sehr gute Ergebnisse. Diese wurden in Vergleichsexperimenten zum Stand der Technik verglichen, insbesondere zu Deep Learning Modellen.

Im letzten Kapitel dieser Arbeit wurde der Merkmalsansatz zur Erkennung von Emotionen auf ein dynamisches Modell zum Tracking des emotionalen Zustands erweitert. Der Stand der Technik zeigt keine Arbeiten, die ausschließlich auf dem Gesichtsausdruck ein Tracking des emotionalen Zustands durchführen. Die meisten Arbeiten basieren auf der multimodalen Analyse von Sprache und Körpersprache. Hier wurde ein Ansatz gewählt, der ausschließlich mit dem Gesichtsausdruck arbeitet. Zunächst wurde eine Übertragung des emotionalen Zustands auf eine kontinuierliche Größe entwickelt. Es wurde ein Gaußprozess trainiert, der die Emotionen auf den Valenz-Erregungsraum abbildet. Diese Abbildung hat ein Trackingproblem ermöglicht. Der Systemzustand konnte durch das Valenz-Erregungs-Wertepaar repräsentiert werden. Für das Tracking wurde ein Kalman Filter mit einer Ungleichheitsnebenbedingung formuliert.

Die Sigmapunkte der Unscented Transformation wurden auf den Einheitskreis projiziert, sofern sie sich außerhalb diesem befanden. Hierdurch konnten gute Trackingergebnisse ermittelt werden, die robust bei der Anbahnung eines emotionalen Gesichtsausdrucks funktionierten.

Die Formulierung eines Trackingmodells für die Verfolgung des emotionalen Gesichtsausdrucks beschließt diese Arbeit. Es zeigt, dass es möglich ist, ein Framework basierend auf dem 68 Punkte umfassenden Landmarkenmodell zu formulieren. Damit lassen sich verschiedene Aufgaben erfüllen von der Erkennung des emotionalen Zustands bis hin zur Verfolgung des emotionalen Zustands. Im Vergleich mit dem State of the Art gibt es Methoden, die

eine etwas höhere Genauigkeit liefern, jedoch ausschließlich für den Klassifikationsgebrauch genutzt werden. In dieser Arbeit konnte ein Modell für viele Aufgaben eingesetzt werden.

Für zukünftige Arbeiten kann das Modell beschleunigt werden, um eine Echtzeitauswertung der genannten Zustände zu ermöglichen. Ebenso können einzelne Komponenten des Frameworks durch bessere Modelle ersetzt werden, um die Leistung zu verbessern. Der Emotionsklassifikator kann mit weiteren Daten geprüft werden. Zusätzlich kann man mit mehr Trainingsdaten die Stabilität der Ergebnisse erhöhen.

Literatur

[A S64] A Sonquist, John and Morgan, James: „The Detection of Interaction Effects: A Report on a Computer Program for the Selection of Optimal Combinations of Explanatory Variables“. In: *SERBIULA (sistema Librum 2.0)* (Jan. 1964) (siehe S. 42).

[Abe87] Abel, J. and Smith, J.: „The spherical interpolation method for closed-form passive source localization using range difference measurements“. In: *ICASSP '87. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Bd. 12. Apr. 1987, S. 471–474. doi: 10.1109/ICASSP.1987.1169674 (siehe S. 87).

[Al-16] Al-Hamadi, Ayoub; Saeed, Anwar; Niese, Robert; Handrich, Sebastian and Neumann, Heiko: „Emotional Trace: Mapping of Facial Expression to Valence-arousal Space“. In: *British Journal of Applied Science & Technology* 16.6 (2016) (siehe S. 188, 190).

[Aru02] Arulampalam, M. S.; Maskell, S.; Gordon, N. and Clapp, T.: „A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking“. In: *IEEE Transactions on Signal Processing* 50.2 (Feb. 2002), S. 174–188. doi: 10.1109/78.978374 (siehe S. 81).

[Asa10] Asadifard, Mozhdeh and Shanbehzadeh, Jamshid: „Automatic Adaptive Center of Pupil Detection Using Face Detection and CDF Analysis“. In: *International MultiConference of Engineers and Computer Scientists 2010 (IMECS 2010)*. 2010 (siehe S. 96, 97).

[Bai08] Bailenson, Jeremy N.; Pontikakis, Emmanuel D.; Mauss, Iris B.; Gross, James J.; Jabon, Maria E.; Hutcherson, Cendri A. C.; Nass, Clifford and John, Oliver: „Real-time Classification of Evoked Emotions Using Facial Feature Tracking and Physiological Responses“. In: *Int. J. Hum.-Comput. Stud.* 66.5 (Mai 2008),

S. 303–317. DOI: 10.1016/j.ijhcs.2007.10.011. URL: http://dx.doi.org/10.1016/j.ijhcs.2007.10.011 (siehe S. 12).

[Bar02] Bar-Shalom, Yaakov; Kirubarajan, Thiagalingam and Li, X.-Rong: Estimation with Applications to Tracking and Navigation. New York, NY, USA: John Wiley & Sons, Inc., 2002 (siehe S. 78).

[Beu09] Beutler, F.; Huber, M. F. and Hanebeck, U. D.: „Gaussian Filtering using state decomposition methods“. In: *2009 12th International Conference on Information Fusion*. Juli 2009, S. 579–586 (siehe S. 83, 94).

[Bra98] Bradski, G. R.: „Real time face and object tracking as a component of a perceptual user interface“. In: *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV'98 (Cat. No.98EX201)*. Okt. 1998, S. 214–219. DOI: 10.1109/ACV.1998.732882 (siehe S. 5).

[Bre01] Breiman, Leo: „Random Forests“. In: *Machine Learning* 45.1 (2001), S. 5–32. DOI: 10.1023/A:1010933404324 (siehe S. 42, 44).

[Ceh16] Cehovin, L.; Leonardis, A. and Kristan, M.: „Visual Object Tracking Performance Measures Revisited“. In: *IEEE Transactions on Image Processing* 25.3 (März 2016), S. 1261–1274. DOI: 10.1109/TIP.2016.2520370 (siehe S. 126).

[Cha94] Chan, Y. T. and Ho, K. C.: „A simple and efficient estimator for hyperbolic location“. In: *IEEE Transactions on Signal Processing* 42.8 (Aug. 1994), S. 1905–1915. DOI: 10.1109/78.301830 (siehe S. 87).

[Che16] Chen, Tianqi and Guestrin, Carlos: „XGBoost: A Scalable Tree Boosting System“. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD16*. 2016, S. 785–794 (siehe S. 42, 43, 166).

[Coo00] Cootes, T.F. and Taylor, C.J.: Statistical Models of Appearance for Computer Vision. 2000 (siehe S. 26, 114, 121, 122, 124, 154, 161).

[Cra02] Cramer, J.S.: „The Origins of Logistic Regression“. In: *Tinbergen Institute, Tinbergen Institute Discussion Papers* (Jan. 2002). doi: 10.2139/ssrn.360300 (siehe S. 42).

[Den09] Deng, J.; Dong, W.; Socher, R.; Li, L. J.; Li, K. and Fei-Fei, L.: „ImageNet: A large-scale hierarchical image database“. In: *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009 (siehe S. 52, 180).

[Dun10] Dunau, Patrick; Packi, Ferdinand; Beutler, Frederik and Hanebeck, Uwe D.: „Efficient multilateration tracking with concurrent offset estimation using stochastic filtering techniques“. In: *2010 13th International Conference on Information Fusion*. Juli 2010, S. 1–8. doi: 10.1109/ICIF.2010.5712059 (siehe S. 13).

[Dun14] Dunau, Patrick: „Exploitation of GPS-control points in low-contrast IR-imagery for homography estimation“. In: *Forum Bildverarbeitung 2014*. Hrsg. von Puente Leon, Fernando and Heizmann, Michael. Regensburg: KIT Scientific Publishing, Nov. 2014, S. 97–106 (siehe S. 15, 16).

[Dun15a] Dunau, Patrick; Fitz, Daniel and Beyerer, Jürgen: „Homography estimation for low-contrast IR image sequences utilizing GPS control points“. In: *tm - Technisches Messen* 85.5 (Apr. 2015), S. 262–272 (siehe S. 16, 18).

[Dun15b] Dunau, Patrick; Fitz, Daniel and Stein, Karin U.: „Evaluation of statistical methods for the evaluation of observer trials for the assessment of the effectiveness of signature measures“. In: *Proceedings of SPIE Volume 9653, Target and Background Signatures, Toulouse, 2015*. 2015 (siehe S. 17).

[Dun16a] Dunau, P. and Beyerer, J.: „Iris tracking using extended object tracking“. In: *2016 19th International Conference on Information Fusion (FUSION)*. 2016, S. 1735–1742 (siehe S. 18).

[Dun16b] Dunau, Patrick; Huber, Samuel W.; Stein, Karin U. and Wellig, Peter: „Asynchronous threat awareness by observer trials

using crowd simulation“. In: *Proc. SPIE 9997, Target and Background Signatures II*. Hrsg. von Stein, Karin U. and Schleijpen, Ric H. M. A. SPIE Proceedings. SPIE, 2016, 99970K. doi: 10.1117/12.2241981 (siehe S. 20, 21).

[Dun18a] Dunau, Patrick; Bonny, Mike; Huber, Marco F. and Beyerer, Jürgen: „Reduced Feature Set For Emotion Recognition Based On Angle And Size Information“. In: *Intelligent Autonomous Systems 15 Proceedings of the 15th International Conference IAS-15*. 2018 (siehe S. 22, 23, 154).

[Dun18b] Dunau, Patrick; Huber, Marco F. Huber and Beyerer, Jürgen: „Comparison of Angle and Size Features with Deep Learning for Emotion Recognition“. In: *Proceedings of the 23rd Iberoamerican Congress on Pattern Recognition*. 2018 (siehe S. 23).

[Dun19] Dunau, Patrick; Huber, Marco F. and Beyerer, Juergen: „Gaussian Process based Dynamic Facial Emotion Tracking“. In: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2019)*. Taipei, Taiwan, 2019 (siehe S. 24).

[Ekm02] Ekman, P; Friesen, W and Hager, J: „Facial Action Coding System: Research Nexus“. In: *Network Research Information* 2 (2002), S. 3 (siehe S. 166).

[Ekm78] Ekman, P. and Friesen, W.: Facial Action Coding System: A Technique for the Measurement of Facial Movement. Palo Alto: Consulting Psychologists Press, 1978 (siehe S. 166).

[Ekm99] Ekman, Paul: „Basic Emotions“. In: *Handbook of Cognition and Emotion*. Hrsg. von Dalgeish, T. and Power, M. John Wiley & Sons Ltd., 1999, S. 45–60 (siehe S. 8, 115, 151, 184, 188).

[Fai15] Faion, F.; Zea, A.; Baum, M. and Hanebeck, U. D.: „Partial likelihood for unbiased extended object tracking“. In: *2015 18th International Conference on Information Fusion (Fusion)*. Juli 2015, S. 1022–1029 (siehe S. 105).

[Fis81] Fischler, Martin A. and Bolles, Robert C.: „Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography“. In: *Commun. ACM* 24.6 (Juni 1981), S. 381–395. doi: 10.1145/358669.358692. url: https://doi.org/10.1145/358669.358692 (siehe S. 69).

[Fös87] Föstner, M. A. and Gülch, E.: „A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centers of Circular Features“. In: *ISPRS Intercommission Workshop*. Interlaken, Switzerland, 1987 (siehe S. 64).

[Fri02] Friedman, Jerome H.: „Stochastic gradient boosting“. In: *Computational Statistics & Data Analysis* 38.4 (2002). Nonlinear Methods and Data Mining, S. 367–378. doi: https://doi.org/10.1016/S0167-9473(01)00065-2. url: http://www.sciencedirect.com/science/article/pii/S0167947301000652 (siehe S. 28, 42, 44).

[Gre84] Green, P. J.: „Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and some Robust and Resistant Alternatives“. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 46.2 (1984), S. 149–192. url: http://www.jstor.org/stable/2345503 (siehe S. 27).

[Har04] Hartley, R. and Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, 2004 (siehe S. 64, 68).

[Hay98] Haykin, Simon: Neural Networks: A Comprehensive Foundation (2nd Edition) Neural Networks: A Comprehensive Foundation. Jan. 1998 (siehe S. 42).

[Hay99] Haykin, Simon: Neural Networks - A Comprehensive Foundation. Pearson Prentice Hall, 1999 (siehe S. 49).

[Hua14] Huang, Xiaohua: „Methods For Facial Expression Recognition With Applications In Challenging Situations“. Diss. University of Oulu, Finland, 2014 (siehe S. 9).

[Hub08] Huber, Marco F. and Hanebeck, Uwe D.: „Gaussian Filter based on Deterministic Sampling for High Quality Nonlinear Estimation“. In: *IFAC Proceedings Volumes* 41.2 (2008). 17th IFAC World Congress, S. 13527–13532. doi: https://doi.org/10.3182/20080706-5-KR-1001.02291. url: http://www.sciencedirect.com/science/article/pii/S1474667016411572 (siehe S. 81, 84, 94, 107).

[Hub15] Huber, Samuel and Wellig, Peter: „Human factors of target detection tasks within heavily cluttered video scenes“. In: *Proceedings of SPIE Volume 9653, Target and Background Signatures, Toulouse, 2015*. 2015 (siehe S. 148).

[Hub17] Huber, Samuel; Dunau, Patrick; Wellig, Peter and Stein, Karin: „Dependency of human target detection performance on clutter and quality of supporting image analysis algorithms in a video surveillance task“. In: *Conference on Target and Background Signatures, SPIE Security And Defence 2017*. Bd. 10432. 2017, S. 10432 - 10432 –6. doi: 10.1117/12.2278342. url: https://doi.org/10.1117/12.2278342 (siehe S. 21).

[Hus12] Hussain, Sibt Ul; Napoléon, Thibault and Jurie, Frédéric: „Face Recognition using Local Quantized Patterns“. In: *British Machive Vision Conference*. Guildford, United Kingdom, Sep. 2012, 11 pages. url: https://hal.archives-ouvertes.fr/hal-00806104 (siehe S. 9).

[Jul00] Julier, S.; Uhlmann, J. and Durrant-Whyte, H. F.: „A new method for the nonlinear transformation of means and covariances in filters and estimators“. In: *IEEE Transactions on Automatic Control* 45.3 (März 2000), S. 477–482. doi: 10.1109/9.847726 (siehe S. 81).

[Jul02] Julier, Simon J. and Industries, Idak: „The scaled unscented transformation“. In: *in Proc. IEEE Amer. Control Conf.* 2002, S. 4555–4559 (siehe S. 38).

[Jul07] Julier, S. J. and LaViola, J. J.: „On Kalman Filtering With Nonlinear Equality Constraints“. In: *IEEE Transactions on Signal Processing* 55.6 (Juni 2007), S. 2774–2784. doi: 10.1109/TSP.2007.893949 (siehe S. 123).

[Jul99] Julier, Simon J. and Uhlmann, Jeffrey K.: „A New Extension of the Kalman Filter to Nonlinear Systems“. In: *Proc. SPIE* 3068 (Feb. 1999). doi: 10.1117/12.280797 (siehe S. 29, 37, 81).

[Kal10] Kalal, Z.; Mikolajczyk, K. and Matas, J.: „Face-TLD: Tracking-Learning-Detection applied to faces“. In: *2010 IEEE International Conference on Image Processing*. Sep. 2010, S. 3789–3792. doi: 10.1109/ICIP.2010.5653525 (siehe S. 5, 6).

[Kál60] Kálman, Rudolf E.: „A New Approach to Linear Filtering and Prediction Problems“. In: *Journal of Basic Engineering* 82.5 (M 1960), S. 35–45. doi: 10.1115/1.3662552 (siehe S. 29, 37).

[Kan08] Kandepu, Rambabu; Imsland, Lars and Foss, Bjarne A.: „Constrained State Estimation Using the Unscented Kalman Filter“. In: *Proceedings of the 16th Mediterranean Conference on Control and Automation*. 2008 (siehe S. 196).

[Kaz14] Kazemi, V. and Sullivan, J.: „One millisecond face alignment with an ensemble of regression trees“. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Juni 2014, S. 1867–1874. doi: 10.1109/CVPR.2014.241 (siehe S. 6, 7, 26, 28, 29).

[Ken89] Kendall, David G.: „A Survey of the Statistical Theory of Shape“. In: *Statistical Science* 4.2 (1989), S. 87–99. url: http://www.jstor.org/stable/2245331 (siehe S. 116).

[Kim08] Kim, Minyoung; Kumar, S.; Pavlovic, V. and Rowley, H.: „Face tracking and recognition with visual constraints in real-world videos“. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. Juni 2008, S. 1–8. doi: 10.1109/CVPR.2008.4587572 (siehe S. 5, 6).

[Kin09] King, Davis E.: „Dlib-ml: A Machine Learning Toolkit“. In: *J. Mach. Learn. Res.* 10 (Dez. 2009), S. 1755–1758. URL: http://dl.acm.org/citation.cfm?id=1577069.1755843 (siehe S. 6, 124, 126).

[Kiv04] Kivinen, Jyrki; Smola, Alexander and C. Williamson, Robert: „Learning With Kernels“. In: *IEEE Transactions on Signal Processing* 52 (Aug. 2004), S. 2165–2176. DOI: 10.1109/TSP.2004.830991 (siehe S. 42).

[Liu15] Liu, Mengyi; Li, Shaoxin; Shan, Shiguang; Wang, Ruiping and Chen, Xilin: „Deeply Learning Deformable Facial Action Parts Model for Dynamic Expression Analysis“. In: *Computer Vision – ACCV 2014*. Hrsg. von Cremers, Daniel; Reid, Ian; Saito, Hideo and Yang, Ming-Hsuan. Cham: Springer International Publishing, 2015, S. 143–157 (siehe S. 9, 183).

[Liu16] Liu, M.; Shan, S.; Wang, R. and Chen, X.: „Learning Expressionlets via Universal Manifold Model for Dynamic Facial Expression Recognition“. In: *IEEE Transactions on Image Processing* 25.12 (Dez. 2016), S. 5920–5932. DOI: 10.1109/TIP.2016.2615424 (siehe S. 183).

[Lop17] Lopes, A. T.; Aguiar, E. de; De Souza, A. F. and Oliveira-Santos, T.: „Facial expression recognition with Convolutional Neural Networks: Coping with few data and the training sample order“. In: *Pattern Recognition* 61 (2017), S. 610–628 (siehe S. 10, 183).

[Low99] Lowe, D. G.: „Object recognition from local scale-invariant features“. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Bd. 2. Sep. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410 (siehe S. 27, 64).

[Luc10a] Lucey, P.; Cohn, J. F.; Kanade, T.; Saragih, J.; Ambadar, Z. and Matthews, I.: „The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression“. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. 2010, S. 94–101. DOI: 10.1109/CVPRW.2010.5543262 (siehe S. 152, 167).

[Luc10b] Lucey, Patrick; Cohn, Jeffrey F.; Kanade, Takeo; Saragih, Jason; Ambadar, Zara and Matthews, Iain: „The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression“. In: *Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010)*, San Francisco, USA, 2010, S. 94–101 (siehe S. 115).

[Mal11] Malandrakis, N.; Potamianos, A.; Evangelopoulos, G. and Zlatintsi, A.: „A supervised approach to movie emotion tracking“. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mai 2011, S. 2376–2379. doi: 10.1109/ICASSP.2011.5946961 (siehe S. 11).

[Met11] Metallinou, A.; Katsamanis, A.; Wang, Y. and Narayanan, S.: „Tracking changes in continuous emotion states using body language and prosodic cues“. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mai 2011, S. 2288–2291. doi: 10.1109/ICASSP.2011.5946939 (siehe S. 11, 12).

[Met13] Metallinou, Angeliki; Katsamanis, Athanasios and Narayanan, Shrikanth: „Tracking continuous emotional trends of participants during affective dyadic interactions using body language and speech information“. In: *Image and Vision Computing* 31.2 (2013). Affect Analysis In Continuous Input, S. 137–152. doi: https://doi.org/10.1016/j.imavis.2012.08.018. url: http://www.sciencedirect.com/science/article/pii/S0262885612001710 (siehe S. 12).

[Mol16] Mollahosseini, A.; Chan, D. and Mahoor, M. H.: „Going deeper in facial expression recognition using deep neural networks“. In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. März 2016, S. 1–10. doi: 10.1109/WACV.2016.7477450 (siehe S. 10, 183).

[Pen16] Peng, Min; Wang, Chongyang; Chen, Tong and Liu, Guangyuan: „NIRFaceNet: A Convolutional Neural Network for Near-Infrared Face Identification“. In: *Information* 7.4 (2016). doi: 10.3390/info7040061. url: http://www.mdpi.com/2078-2489/7/4/61 (siehe S. 49).

[Qu14] Qu, C.; Monari, E.; Schuchert, T and Beyerer, J.: „Fast, robust and automatic 3D face model reconstruction from videos“. In: *Proc.IEEE International Conference on Advanced Video and SignalBased Surveillance (AVSS)*. 2014, S. 113–118 (siehe S. 7).

[Qu15a] Qu, C.; Gao, H.; Monari, E.; Beyerer, J. and Thiran, J. P.: „Towards robust cascaded regression for face alignment in the wild“. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2015, S. 1–9. doi: 10.1109/CVPRW.2015.7301348 (siehe S. 7, 22, 26–28, 114, 120, 124, 126–128, 154).

[Qu15b] Qu, C.; Herrmann, C.; Monari, E.; Schuchert, T. and Beyerer, J.: „3D vs. 2D: on the importance of registration for hallucinating faces under unconstrained poses“. In: *Proc. Conference on Computerand Robot Vision (CRV)*. 2015, S. 139–146 (siehe S. 7).

[Qu15c] Qu, C.; Monari, E.; Schuchert, T. and Beyerer, J: „Realistic texture extraction for 3D face models robust to self-occlusion“. In: *Proc. SPIE, Image Processing:Machine Vision Applications VIII,vol. 9405*. 2015, 94050P (siehe S. 7).

[Qu15d] Qu, C.; Monari, E.; Schuchert, T. and Beyerer, J.: „Adaptive contour fitting for pose-invariant 3D face shape reconstruction“. In: *Proc. British Machine Vision Conference (BMVC)*. 2015, S. 87.1–87.12 (siehe S. 7).

[Qu17] Qu, C.; Herrmann, C.; Monari, E.; Schuchert, T. and Beyerer, J.: „Robust 3D patch-based face hallucination“. In: *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2017, S. 1105–1114 (siehe S. 7).

[Qu18] Qu, Chengchao: „Facial Texture Super-Resolution by Fitting 3D Face Models“. Diss. Karlsruhe Institute of Technology, KIT, 2018 (siehe S. 7).

[Ras06] Rasmussen, Carl Edward and Williams, Christopher K. I.: Gaussian Processes for Machine Learning. MIT Press, 2006 (siehe S. 52, 53, 55–57).

[Ros58] Rosenblatt, F.: „The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain“. In: *Psychological Review* (1958), S. 65–386 (siehe S. 45).

[Sch00] Schmidt, Robert F.; Thews, Gerhard and Lang, Florian: Physiologie des Menschen. Springer-Verlag Berlin-Heidelberg, 2000 (siehe S. 105).

[Sim06] Simon, Dan: Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches. John Wiley & Sons, Inc., Jan. 2006. doi: 10.1002/0470045345 (siehe S. 29).

[Sim14] Simonyan, K. and Zisserman, A.: „Very Deep Convolutional Networks for Large-Scale Image Recognition“. In: *CoRR* abs/1409.1556, arXiv (2014) (siehe S. 51, 180, 186).

[Sob62] Sobotta, J. and Becher, H.: Atlas der Anatomie des Menschen. Urban & Schwarzenberg, München-Berlin, 1962 (siehe S. 95).

[Val12] Valstar, M. F. and Pantic, M.: „Fully Automatic Recognition of the Temporal Phases of Facial Actions“. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42.1 (2012), S. 28–43 (siehe S. 8).

[Vio04] Viola, Paul and Jones, Michael J.: „Robust Real-Time Face Detection“. In: *International Journal of Computer Vision* 57.2 (2004), S. 137–154. doi: 10.1023/B:VISI.0000013087.49260.fb (siehe S. 19).

[Vur07] Vural, Esra; Cetin, Mujdat; Ercil, Aytul; Littlewort, Gwen; Bartlett, Marian and Movellan, Javier: „Drowsy Driver Detection Through Facial Movement Analysis“. In: *Human–Computer Interaction*. Hrsg. von Lew, Michael; Sebe, Nicu;

Huang, Thomas S. and Bakker, Erwin M. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, S. 6–18 (siehe S. 7).

[Zea14] Zea, A.; Faion, F. and Hanebeck, U. D.: „Tracking extended objects using extrusion Random Hypersurface Models“. In: *2014 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. Okt. 2014, S. 1–6. doi: 10.1109/SDF.2014.6954722 (siehe S. 105).

Eigene Publikationen

Dieser Abschnitt enthält ein vollständiges Verzeichnis der eigenen Veröffentlichungen.

[1] Dunau, Patrick; Packi, Ferdinand; Beutler, Frederik and Hanebeck, Uwe D.: „Efficient multilateration tracking with concurrent offset estimation using stochastic filtering techniques“. In: *2010 13th International Conference on Information Fusion*. Juli 2010, S. 1–8. doi: 10.1109/ICIF.2010.5712059.

[2] Dunau, Patrick: „Exploitation of GPS-control points in low-contrast IR-imagery for homography estimation“. In: *Forum Bildverarbeitung 2014*. Hrsg. von Puente Leon, Fernando and Heizmann, Michael. Regensburg: KIT Scientific Publishing, Nov. 2014, S. 97–106.

[3] Dunau, Patrick; Fitz, Daniel and Beyerer, Jürgen: „Homography estimation for low-contrast IR image sequences utilizing GPS control points“. In: *tm - Technisches Messen* 85.5 (Apr. 2015), S. 262–272.

[4] Dunau, Patrick; Fitz, Daniel and Stein, Karin U.: „Evaluation of statistical methods for the evaluation of observer trials for the assessment of the effectiveness of signature measures“. In: *Proceedings of SPIE Volume 9653, Target and Background Signatures, Toulouse, 2015*. 2015.

[5] Dunau, P. and Beyerer, J.: „Iris tracking using extended object tracking“. In: *2016 19th International Conference on Information Fusion (FUSION)*. 2016, S. 1735–1742.

[6] Dunau, Patrick; Huber, Samuel W.; Stein, Karin U. and Wellig, Peter: „Asynchronous threat awareness by observer trials using crowd simulation“. In: *Proc. SPIE 9997, Target and Background Signatures II*.

Hrsg. von Stein, Karin U. and Schleijpen, Ric H. M. A. SPIE Proceedings. SPIE, 2016, 99970K. doi: 10.1117/12.2241981.

[7] Huber, Samuel; Dunau, Patrick; Wellig, Peter and Stein, Karin: „Dependency of human target detection performance on clutter and quality of supporting image analysis algorithms in a video surveillance task". In: *Conference on Target and Background Signatures, SPIE Security And Defence 2017*. Bd. 10432. 2017, S. 10432 - 10432 –6. doi: 10.1117/12.2278342. url: https://doi.org/10.1117/12.2278342.

[8] Dunau, Patrick; Bonny, Mike; Huber, Marco F. and Beyerer, Jürgen: „Reduced Feature Set For Emotion Recognition Based On Angle And Size Information". In: *Intelligent Autonomous Systems 15 Proceedings of the 15th International Conference IAS-15*. 2018.

[9] Dunau, Patrick; Huber, Marco F. Huber and Beyerer, Jürgen: „Comparison of Angle and Size Features with Deep Learning for Emotion Recognition". In: *Proceedings of the 23rd Iberoamerican Congress on Pattern Recognition*. 2018.

[10] Dunau, Patrick; Huber, Marco F. and Beyerer, Juergen: „Gaussian Process based Dynamic Facial Emotion Tracking". In: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2019)*. Taipei, Taiwan, 2019.

Abbildungsverzeichnis

Tabellenverzeichnis