Institut für Visualisierung und Interaktive Systeme
Abteilung Intelligente Systeme
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Diplomarbeit Nr. 2982

Background Subtraction in der Videoüberwachung

Sebastian Brutzer

Studiengang:	Informatik
Prüfer:	Prof. Dr. Gunther Heidemann
Betreuer:	Dipl.-Inf. Benjamin Höferlin
begonnen am:	14. November 2009
beendet am:	14. Mai 2010
CR-Klassifikation:	I.2.10, I.4.6, I.4.8, I.5.2

Inhaltsverzeichnis

Abbildungsverzeichnis

Verzeichnis der Algorithmen

1 Abstract

Background Subtraction Verfahren werden zur Extraktion von Vordergrundobjekten aus Videodaten eingesetzt. In den letzten Jahren wurden viele solcher Verfahren entwickelt. Arbeiten in denen diese ausführlich evaluiert werden sind jedoch selten. Diese werden benötigt um den aktuellen Stand der Forschung zu ermitteln und gegebenenfalls Problemfelder näher untersuchen und Lösungsansätze für diese entwickeln zu können.
In dieser Diplomarbeit wird eine pixelgenaue Untersuchung von *Background Subtraction* Verfahren bezüglich verschiedenen Problemen durchgeführt. Die Animationssoftware *Maya* wurde eingesetzt um eine realistische Überwachungsszene mit einer großen Menge an Vordergrundmasken zu erstellen, die für die Evaluation benötigt werden.
Für sechs Verfahren wurden optimale Parameter durch den Einsatz von *ROC*- Kurven bestimmt. Hierfür wurde eine komplexe Trainingsszene verwendet, die neben Vordergrundobjekten auch beispielsweise Hintergrundbewegungen, periodische Wiederholungen, Tarnungssituationen oder Reflektionen enthält. Zudem wurde ein Bildrauschen auf die Videobilder aufgetragen. Anschließend wurden die Verfahren mit den berechneten, optimalen Parameter unter verschiedenen Problemstellungen evaluiert. Zu diesen gehörten Beleuchtungsänderungen, Hintergrundbewegungen, eine Nachtszene mit starkem Rauschen, Tarnungssituationen sowie Hintergrundbewegungen. Zur Bewertung der Verfahren wurde das *F-Maß* verwendet.
Um zu messen, ob die Verfahren in der Lage sind, Schatten zu erkennen und diesen dem Hintergrund der Szene zuzuordnen, wurden spezielle Schattenmasken angefertigt und die Anzahl der fälschlicherweise als Vordergrund klassifizierten Schattenpixel berechnet.
Die Auswertung der Evaluationsergebnisse hat ergeben, dass keines der Verfahren für die bezüglich der Trainingssequenz optimalen Parameter unter allen Problemstellungen gute Resultate erzielen konnten. Bei der verwendeten Nachtsszene konnte keines der Verfahren gut abschneiden. Die Kombination aus Tarnungssituationen und Bildrauschen bereitet den Verfahren große Probleme. Zudem wurden unter diesen Parametereinstellungen viele Schattenpixel falsch klassifiziert.

2 Aufbau der Arbeit

Der Aufbau dieser Diplomarbeit ist in diesem Kapitel aufgeführt. In Kapitel 3 wird erläutert, was *Background Subtraction* Verfahren sind, wofür sie eingesetzt werden und wie diese Diplomarbeit zustande gekommen ist. Die für die Verfahren benötigten mathematischen Grundlagen sowie die im Verlauf der Ausarbeitung häufiger auftretenden Begriffe werden in Kapitel 4 aufgeführt und erläutert.
In Kapitel 5 werden arbeiten vorgestellt, die sich wie diese Diplomarbeit mit *Background Subtraction* Verfahren und deren Evaluation beziehungsweise Evaluierungsmöglichkeiten beschäftigen.
In der Praxis sind die *Background Subtraction* einer Vielzahl an Herausforderungen und Problemen ausgesetzt, mit denen sie zurecht kommen müssen, um gute Resultate erzielen zu können. Zu ihnen gehören :

1. Laufzeit
2. Trainingsdaten
3. Die Wahl der Parameter
4. Veränderungen der Szene
5. Beleuchtungsänderungen
6. Wetter
7. Tarnung
8. Schatten
9. Verdeckungen
10. Uninteressante Bewegungen, periodische Wiederholungen
11. Kamerabewegungen

Diese Probleme werden in den Teilabschnitten des Kapitels ausführlich diskutiert.
Die im Rahmen dieser Diplomarbeit evaluierten Verfahren, sowie weitere grundlegende Arbeiten, sind in Kapitel 7 aufgeführt. Neben der Grundidee dieser Verfahren werden auch ihre Eigenschaften wie beispielsweise Laufzeiten und Speicheranforderungen bezüglich O-Notation abgeschätzt. Zu diesen Verfahren gehören :

1. Differenzbild Verfahren
2. Mittelwert Verfahren
3. *Running Average*
4. *Running Gaussian*
5. *Median* Verfahren

6. Kantenbasiertes Verfahren von McKenna et al.
7. Kantenbasiertes Verfahren von Jabri et al.
8. *Mixture of Gaussian*
9. *Codebook* Verfahren
10. Verfahren von Li et al.

Kapitel 8 beschäftigt sich mit Nachbearbeitungsverfahren um die Resultate der *Background Subtraction* Verfahren, die Subtraktionsbilder, für eventuell anschließende Aufgaben zu verbessern.
Die durchgeführte Evaluatuion sowie die hierfür angewendeten Techniken und Metriken werden in Kapitel 9 diskutiert. Zudem sind hier die erhaltenen Resultate zu sehen.

3 Einleitung und Motivation

In diesem Kapitel wird erläutert, wie es zum Entstehen dieser Diplomarbeit kam. Zudem werden die Einsatzmöglichkeiten automatischer Videoüberwachungssysteme aufgezeigt. Dabei wird speziell auf folgende Fragen eingegangen :

1. Wofür wird Videoüberwachung betrieben? (Kapitel 3.1)
2. Weshalb wird automatische Videoüberwachung angestrebt? (Kapitel 3.2)
3. Was ist *Background Subtraction*? (Kapitel 3.3)
4. Wie können die Resultate der *Background Subtraction* Verfahren analysiert werden? (Kapitel 3.4)
5. Weshalb die ausführliche Evaluation der Verfahren? (Kapitel 3.5)

3.1 Wofür wird Videoüberwachung betrieben?

Der Einsatz von Videoüberwachungssystemen hat in den letzten Jahren stark zugenommen. Dies liegt hauptsächlich daran, dass die Leistungsfähigkeit der hierzu verwendeten Videokameras stark zugenommen hat. Heute übliche Einsatzgebiete sind beispielsweise:

- **Bewachung von Objekten, Gebieten, Personen:**
  Diese Bewachung erfüllt im Allgemeinen eine schützende Funktion. Objekte werden bewacht, damit sie nicht gestohlen oder beschädigt werden (Objektschutz) und Personen damit sie nicht verletzt werden (Personenschutz). Eine Überwachung von Gebieten wird häufig eingesetzt, um zu erkennen, ob unbefugte Personen versuchen, diese betreten.

- **Personenkontrolle an Gebäuden:**
  Diese Kontrolle dient dazu, Personen zu identifizieren beziehungsweise aufzufinden. Dadurch lässt sich beispielsweise verhindern, dass unbefugten Personen der Zugang zu diesem Gebäude gewährt wird. Desweitern lässt sich durch eine Identifikation der Zugang auch für einzelnen Personen verweigern (Durchsetzung eines Stadionverbotes, Aufspüren von Personen, die laut Angabe der Polizei Kontakte zu terroristischen Netzwerken haben).

- **Verkehrsüberwachung:**
  Sie dient der Prävention beziehungsweise der Aufklärung von Unfällen oder Verkehrsdelikten. Zudem lassen sich durch die Verkehrsüberwachung auch Staus frühzeitig erkennen, die dann zum Beispiel per Radiodurchsagen publik

gemacht werden können, so dass Verkehrsteilnehmer bei Bedarf auf alternative Wege zurückgreifen.

- **Überwachung der Umwelt:**
  Ähnlich wie bei der Überwachung von technischen Anlagen wird auch die Umwelt beobachtet um *Störungen* (Erdrutsche, Unwetterentwicklung) möglichst zu vermeiden, zu beheben oder vorhersagen zu können. Im Extremfall ist so eventuell eine Evakuierung von Personen, aus den gefährdeten Gebieten möglich.

- **Überwachung von Naturerscheinungen:**
  Hierbei werden Naturerscheinungen aufgenommen und nachträglich analysiert. Dies dient hauptsächlich der Beschaffung neuer Erkenntnisse und wird besonders häufig zu Forschungszwecken in der Wissenschaft eingesetzt.

Wie sich anhand dieser Auflistung erkennen lässt, spielt die Sicherheit, die man durch den Einsatz von Videoüberwachungssystemen erhöhen möchte, eine ganz zentrale Rolle. Besonders seit den Anschlägen vom 11.09.2001 ist die Angst vor Terroranschlägen in der Öffentlichkeit stark angestiegen. Szenarien, in denen Personen Bomben an öffentlichen Plätzen, Bahnhöfen oder Flughäfen hinterlassen und dann aus sicherer Entfernung zünden, beunruhigen viele Menschen seitdem besonders.

Durch den Einsatz von Überwachungssystemen wird versucht, kritische Situationen möglichst frühzeitig zu erkennen, so dass man auf diese geeignet reagieren kann. Die obige Liste zeigt zudem, dass die Einsatzmöglichkeiten nicht auf die Prävention von Terroranschlägen beschränkt sind, sondern sehr vielseitig sind.

3.2 Weshalb wird automatische Videoüberwachung angestrebt?

Der vorherige Abschnitt hat aufgezeigt, in welchen Gebieten Videoüberwachung zum Einsatz kommt. Der in den letzten Jahren beobachtbare Einsatz von Überwachungskameras führt jedoch auch zu einigen Problemen. Die Videoüberwachungsysteme erzeugen eine enorme Menge an visuellen Daten die ausgewertet werden müssen. Kameras die ein Gebiet überwachen, deren Überwachungsbilder aber nicht analysiert werden, erfüllen auch keine schützende Funktion und können im besten Fall nachträglich zu Aufklärungszwecken verwendet werden.
Der Einsatz von Überwachungspersonal zur Auswertung der durch Überwachungskameras aufgezeichneten Bilder ist sehr kostenintensiv, weshalb die Entwicklung automatisierter System vorangetrieben wird. Solche Systeme verwenden meist *Background Subtraction* Verfahren, die Vordergrundobjekte aus Überwachungsdaten extrahieren, damit diese verfolgt werden können und sich deren Verhalten analysieren lässt.

3.3 Was ist *Background Subtraction*?

Das Ziel sogenannter *Background Subtraction* Verfahren ist die Segmentierung von Vordergrundobjekten aus Videodaten. Eine Einteilung der Objekte in Vorder- und Hintergrundobjekte erfolgt aufgrund ihrer Dynamik in der Szene und nicht bezüglich ihrer Art. So wird ein Auto als Hintergrundobjekt eingestuft, falls es beispielsweise auf einem Parkplatz steht und sich daher nicht bewegt. Jedoch wird ein Auto das auf einer Straße fährt, als Vordergrundobjekt klassifiziert. Damit ist die der Separation zu Grunde liegende Idee, dass versucht wird, die für viele Analyseaufgaben interessanten Bereiche eines Videobildes von den uninteressanten zu trennen.
Nun könnte man auch jeden Pixel eines Videobildes auf andere Arten klassifizieren. Eine solche Aufteilung würde beispielsweise einen Pixel als Vordergrund einordnen, falls er zu einem Auto gehört. Gehört er hingegen nicht zu einem Auto sondern zu einem beliebigen anderen Objekt, so wird er als Hintergrund eingestuft. Das setzt jedoch voraus, dass das System auch jedes Auto erkennen und diese von anderen Objekten unterscheiden kann. Typische Klassifikationsverfahren, die für diese Aufgabe verwendet werden können, basieren auf sogenannten Trainingsbildern. Das sind Bilder, die nur die gewünschten Objekte aus verschiedenen Perspektiven, beinhalten. Jedoch existieren Autos in einer solchen Vielzahl von Variationen (Farbe, Größe, Form, etc.), dass eine Klassifikation eine enorme Anzahl an Trainingsbildern erfordern würde, so dass dieses Verfahren nicht praktikabel ist. Zudem ist das Aufsuchen gelernter Objekte relativ aufwändig, so dass das System nicht mehr in Echtzeit arbeiten würde. Außerdem lassen sich nur gelernte Objekte erkennen. Dieser Umstand setzt voraus, dass bekannt sein muss, nach welchen Objekten in den Videobildern gesucht werden soll, da das Verfahren sie sonst nicht finden kann. Daher werden solche Klassifikationsverfahren in der Regel nicht eingesetzt.

In Abbildung 3.1 ist der schematische Aufbau eines typischen *Background Subtraction* Verfahrens zu sehen. Es besteht aus folgenden den Komponenten :

- **Modell**
  Das Modell, oft auch Hintergrundmodell genannt, beinhaltet Informationen, die zur Klassifikation der Pixel in die Kategorien *Hintergrund* und *Vordergrund* verwendet werden können. Wie das Modell genau aufgebaut ist, das heißt welche Datenstrukturen es verwendet und auf welchen Konzepten es beruht dem jeweiligen Verfahren ab.

- **Subtraktionsbild**
  Das eingehende Videobild wird mit dem aktuellen Hintergrundmodell verglichen. Das Resultat dieses Vergleichs ist ein Binärbild, das jeden Pixel in *Hintergrund* oder *Vordergrund* aufteilt. Für diesen Vergleich werden Tests durchgeführt, deren Aussehen und Ablauf von dem jeweilig eingesetzten Verfahren abhängt. Zudem können sogenannte *Confidence Scores* berechnet werden, die angeben, wie sicher sich das Verfahren bezüglich der durchgeführten Klassifikation ist. Beispielsweise lassen sich so Vordergrundregionen aus dem Subtraktionsbild entfernen, wenn die zu ihm gehörenden Pixel einen niedrigen durchschnittlichen *Confidence Score* aufweisen.

- **Aktualisierung des Hintergrundmodells**

  Die Verfahren müssen ihr Modell regelmäßig aktualisieren, so dass es sich an eventuelle Änderungen in der Szene anpassen kann. Ändert sich beispielsweise im Laufe einer Überwachung die Helligkeit der Szene, so kommt es zu fehlerhaften Klassifikationen der Pixel, wenn das Modell diese Änderung nicht aufnimmt. Die eben angesprochenen Vergleiche würden Unterschiede zwischen dem Modell und den Hintergrundpixeln eines Videobildes feststellen und dadurch diese Pixel fälschlicherweise als Vordergrund klassifizieren. Die Aktualisierung erfolgt durch die eingehenden Videobilder. Manche Verfahren verwenden zusätzlich das berechnete Subtraktionsbild (siehe 4.7) für die Aktualisierung.

- **Nachbearbeitung** Um die Qualität der Subtraktionsbilder zu erhöhen, werden häufig Nachbearbeitungsverfahren verwendet. Solche Verfahren werden in Abschnitt 8 aufgeführt.

Das Subtraktionsverfahren liefert somit die oben beschriebene Klassifikation der Pixel. Das Resultat kann nun für Analyseaufgaben verwendet werden.

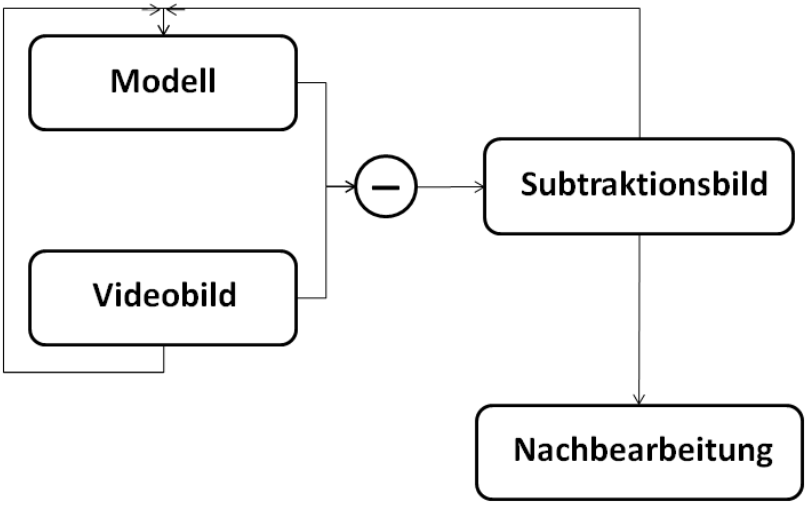

Abbildung 3.1: Schematischer Aufbau eines *Background Subtraction* Verfahrens.

3.4 Wie können die Resultate der *Background Subtraction* Verfahren analysiert werden?

Die Klassifikation der Pixel eingehender Überwachungsbilder, die ein *Background Subtraction* Verfahren durchführt, steht meist am Anfang einer Reihe von Bearbeitungsschritten. Sie stellt einen sogenannten *Low-Level Task* dar, das heißt eine Aufgabe auf unterster Ebene, der Pixelebene. Die Konzepte typischer Überwachungsaufgaben finden auf einer höheren Ebene statt und werden daher *High-Level Tasks* genannt. Sie analysieren und arbeiten auf den Resultaten der *Low-Level Tasks*.

In Abbildung 3.2 ist das *Left Luggage Detection* System schematisch dargestellt, das ich im Rahmen einer Studienarbeit implementiert und evaluiert habe. Dieses Beispiel soll der Beantwortung der Frage dieses Abschnittes dienen.

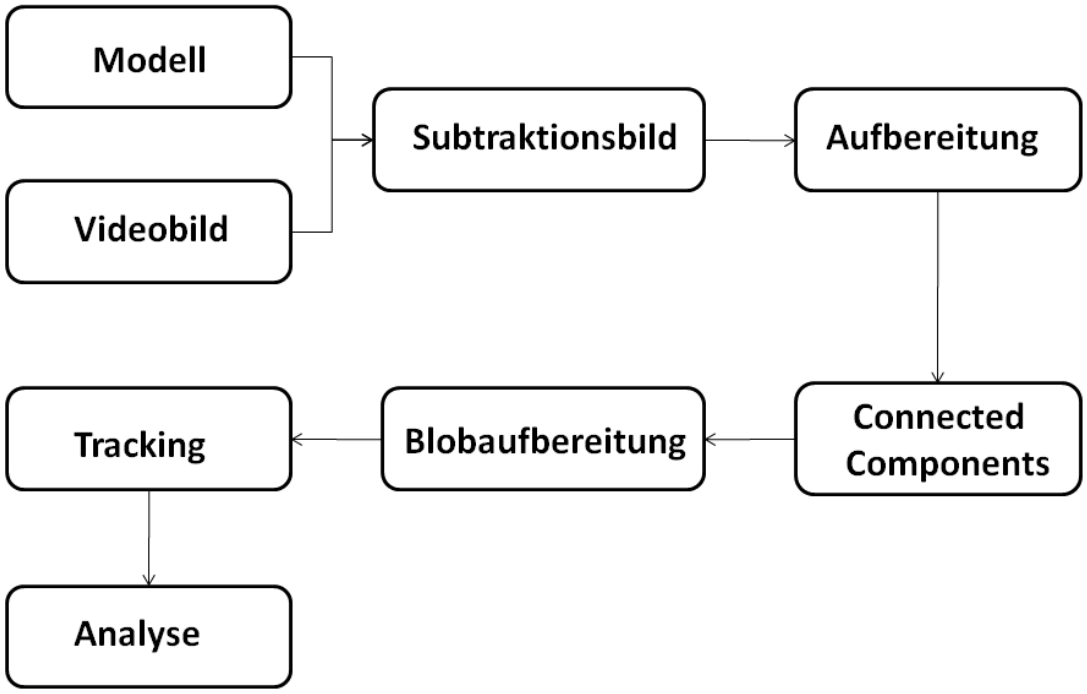

Abbildung 3.2: Schematischer Aufbau eines Systems zur *Left Luggage Detection*. Durch das sequentielle Arbeiten wirken sich Fehler im *Background Subtraction* Verfahren negativ auf die Analyse und somit auf das gesamte System aus. Damit die Resultate des Analyseschrittes zufriedenstellend sind, muss das verwendete *Background Subtraction* Verfahren so gut wie möglich arbeiten.

Das System besteht aus den folgenden Komponenten :

- **Hintergrundmodell**
  Basierend auf einer Trainingssequenz, das heißt einer im Prinzip beliebig langen Sequenz des Anfangs eines Überwachungsvideos, wird ein Hintergrundmodell berechnet. Dieses beinhaltet Informationen, um die im Subtraktionsschritt

gewünschte Klassifikation durchführen zu können. Im Rahmen der angesprochenen Studienarbeit wurden das Mittelwert- sowie das *Median* Verfahren implementiert. Diese werden in Abschnitt 7.2 beziehungsweise in Abschnitt 7.5 vorgestellt.

- **Subtraktion**
  Wie im vorherigen Abschnitt beschrieben, klassifiziert ein solches Verfahren jeden Pixel eines Videobildes in Vorder- beziehungsweise Hintergrund, also in für die Überwachungsaufgabe interessante oder uninteressante Pixel. Dazu wird das aktuelle Hintergrundmodell pixelweise mit dem eingehenden Videobild verglichen und auf Ähnlichkeit untersucht. Ist ein Pixel dem Hintergrund an der entsprechenden Stelle ähnlich, so wird dieser Pixel als Hintergrund eingestuft, ansonsten als Vordergrund. Das Resultat ist ein Binärbild, das für die weitere Analyse verwendet werden kann.

- **Aufbereitung**
  Das erzeugte Binärbild enthält, abhängig von Kamerarauschen, dem eingesetzten Verfahren sowie der Wahl seiner Parameter eine gewisse Anzahl falsch klassifizierter Pixel. Daher ist eine Aufbereitung notwendig, die eine Verbesserung der Qualität des Binärbildes zum Ziel hat. Eine Auflistung sowie Beschreibung hierfür geeigneter Verfahren, befindet sich in Kapitel 8. Für das implementierte System wurde eine dem morphologischen *Closing* ähnliche Methode sowie das Entfernen zu kleiner und zu großer Regionen verwendet.

- **Segmentierung**
  Nach der Aufbereitung werden zusammenhängende Regionen zu Einheiten, sogenannten *Blobs* (Binary Large Object Segment 4.4) zusammengefasst. Resultat des Segmentierungsschrittes ist eine Maske, die an jeder Pixelposition die Nummer des zu dieser Position gehörenden Blobs beinhaltet.

- **Blobeigenschaften**
  Nach der Segmentierung werden Eigenschaften der erzeugten *Blobs* berechnet. Die so berechnete Größe (gegeben durch die Anzahl der Pixel), sowie Ausmaße der *Bounding Box* (siehe hierfür Abschnitt 4.5), können für das *Tracking* verwendet werden.

- **Tracking**
  Das *Tracking* hat die Aufgabe, die detektierten *Blobs* innerhalb einer Videosequenz zu verfolgen und deren Bewegungsverläufe zu speichern, damit diese analsiert werden können.

- **Analyse**
  Im letzten Schritt des Systems werden die gespeicherten Bewegungsverläufe nach dem von *PETS* definierten *Left Luggage Event* durchsucht. Konnte ein solches erkannt werden, so wurde ein Gepäckstück in die Szene gebracht,

abgestellt und zurückgelassen.

Im Rahmen der erwähnten Studienarbeit hat sich gezeigt, wie wichtig ein gutes Binärbild, also das Resultat des eingesetzten *Background Subtraction* Verfahrens ist. Beinhaltet dieses viele falsch klassifizierte Pixel, so ist eine sinnvolle Segmentierung sowie ein anschließendes *Tracking* kaum möglich. Das bedeutet, dass sich früh entstandene Fehler negativ auf das Überwachungsystem auswirken und dieses im Extremfall sogar unbrauchbar machen. Daher ist ein gutes *Background Subtraction* Verfahren sowie eine gute Aufbereitung des Subtraktionsbildes für eine automatisierte Überwachungsaufgabe von großer Bedeutung.
Die oben aufgeführte Relevanz der *Background Subtraction* Verfahren im Bezug auf anstehende Analyseaufgaben, hat dazu geführt, dass diese Diplomarbeit, die sich ausführlich mit solchen Verfahren beschäftigt, zustande kam.

3.5 Weshalb die ausführliche Evaluation der Verfahren?

Der vorherige Abschnitt hat verdeutlicht, weshalb gute *Background Subtraction* Verfahren für den Einsatz vollautomatischer Überwachungssysteme wichtig sind. In den letzten Jahren ist eine Vielzahl solcher Verfahren entwickelt worden. Um den aktuellen Wissensstand in diesem, wie aber auch in jedem anderen Gebiet ermitteln zu können, ist eine Untersuchung der existierenden Verfahren nötig. Nur so kann festgestellt werden, welche Probleme schon gelöst sind und wo noch Forschungsarbeit zur Verbesserung aufgewendet werden muss.
Zwar evaluieren Forscherteams ihre entwickelten Verfahren, jedoch haben die veröffentlichten Ergebnisse häufig nur wenig Aussagekraft. Oft wird mittels einfachen Videos evaluiert, oder nur angemerkt, dass die erzielten Ergebnisse gut seien. Tests, die eine Vielzahl von Verfahren evaluieren und auf deren Stärken, Schwächen sowie Einsatzmöglichkeiten hin analysieren und vergleichen, sind selten und werden daher besonders benötigt. Es existieren zwar bislang Evaluationen die mehrere Verfahren einbeziehen, diese sind aber meist schone einige Jahre alt. Seitdem wurden weitere interessante Ansätze sowie Verbesserungen der Methoden entwickelt, die damals noch unbekannt waren und daher auch nicht berücksichtigt werden konnten. Einige dieser arbeiten sind in Kapitel 5 zusammengefasst.
Zudem existieren nur wenige Studien, die die Genauigkeit der Methoden exakt untersuchen. Die Mehrheit berechnet Metriken einzig auf Objektebene, aber nicht pixelweise. Eine exakte Untersuchung auch im Hinblick auf spezielle Probleme ist aber nötig, da beispielsweise durch Schatten die Statistiken der erkannten Objekte so stark verfälschen können, dass ein *Tracking* und eine anschließende Analyse des berechneten Bewegungsverlaufes nicht möglich ist. Einfache Tests, die nur auf dem Erkennen eines Objekts basieren, reichen daher kaum aus. Wie gut diese Verfahren bei speziellen Problemen letztlich genau abschneiden, ist bisher kaum untersucht.

Ziel dieser Diplomarbeit ist eine ausführliche, pixelgenaue Untersuchung neuer sowie bewährter *Background Subtraction* Verfahren, auf deren Abschneiden bezüglich Problemen, die typischerweise bei Überwachungsaufgaben auftreten.

4 Begriffe und Grundlagen

In diesem Kapitel werden die im Weiteren verwendeten Begriffe aufgeführt, sowie deren Bedeutung erläutert. Zudem werden hier benötigte Grundlagen aus den Bereichen Mathematik sowie Informatik, speziell Computer Vision, behandelt, so dass diese an entsprechender Stelle nicht aufgeführt werden müssen und bei Bedarf hier nachgelesen werden können.

4.1 PETS

PETS (Performance Evaluation of Tracking and Surveillance) ist eine IEEE-Konferenz die, sich mit den Einsatzmöglichkeiten und der Bewertung von automatisierten Überwachungssystemen beschäftigt. Auf die erste Konferenz, die am 31.3.2000 statt fand, folgten bis zum 25.06.2009 noch elf weitere. In der Regel widmet sich jede Konferenz einem speziellen Einsatzgebiet. So beschäftigte man sich im Jahre 2000 speziell mit dem *Tracking* von Personen und Fahrzeugen, 2004 mit dem erkennen spezieller menschlicher Aktivitäten, 2006 sowie 2007 mit *Left Luggage Detection* oder 2009 mit der Überwachung größerer Menschenmengen. Testvideos werden für Evaluationen und Vergleiche zur Verfügung gestellt.

4.2 Vorder- und Hintergrund

Als Vordergrund werden hier Objekte beziehungsweise Regionen verstanden, die für das Überwachungssystem von besonderem Interesse sind. So können beispielsweise Fahrzeuge in ein und der selben Szene als Vordergrund und Hintergrund klassifiziert werden, je nachdem, ob sie für die durchzuführende Analyseaufgabe von besonderem Interesse sind, oder nicht. Im Folgenden gelten die Regionen als interessant, die sich durch eine gewisse Dynamik von den statischen Regionen der Szene, dem sogenannten Hintergrund, abheben.

4.3 Ground Truth

Als *Ground Truth* Daten werden hier vorsegmentierte Videodaten bezeichnet. Häufig geschieht diese Segmentierung objektweise, das heißt, dass beispielsweise pro Objekt, das in einem Videobild vorhanden ist, dessen Mittelpunkt oder die Koordinaten seiner *Bounding Box* bekannt sind. Der zu leistende Aufwand einer pixelweisen Segmentierung ist allerdings so hoch, dass wenn überhaupt, nur wenige Videobilder auf diese Weise segmentiert werden.

4.4 Blob

Der Bergiff *Blob* steht für *Binary Large Object Segment* und bezeichnet Vordergrundregionen, solange diese nicht näher klassifiziert wurden.

4.5 Bounding Box

Eine *Bounding Box* ist ein rechteckiges Gebiet in einem Bild, welches eine detektierte Vordergrundregion vollständig umschließt. Zudem sollte eine *Bounding Box* einen möglichst kleinen Flächeninhalt besitzen und damit die Vordergrundregion möglichst eng begrenzt.

4.6 Connected Components

Ein *Connected Components* Algorithmus ordnet jedem Pixel eines Subtraktionsbildes, eine Nummer die angibt zu welchem Vordergrundobjekt er gehört, zu. Hintergrundpixel erhalten den Wert 0. Zwei Pixel erhalten die gleiche Nummer, falls sie zu der selben, zusammenhängenden Region gehören.

4.7 Selektion

Ein *Background Subtraction* ist selektiv, falls es in der Aktualisierungsphase nur die Pixelpositionen aktualisiert, an denen Hintergrundpixel detektiert werden konnten. Dadurch kann verhindert werden, dass Vordergrundpixel das Hintergrundmodell beeinträchtigen. Fehler in der Klassifizierung bleiben aber länger erhalten, da nicht jede Pixelposition zu jedem Zeitpunkt aktualisiert wird.

4.8 RGB-Farbraum

Der RGB-Farbraum ist additiv aufgebaut, das heißt, dass die durch ihn darstellbaren Farben durch Addieren seiner Grundfarben erzeugt werden können. Seine Grundfarben sind dabei Rot (R), Grün (G) und Blau (B). Farben werden nun durch eine Beschreibung angegeben, die die jeweiligen Anteile der Primärfarben beinhalten. Diese werden im Allgemeinen prozentual durch Dezimalzahlen im Bereich zwischen 0 und 1 angegeben. In der Bildverarbeitung werden die Farben häufig durch 8 Bit dargestellt, was eine Diskretiesierung in 256 unterschiedliche Farbwerte ermöglicht. Das heißt, dass der Anteil dort durch einen Wert zwischen 0 und 255 angegeben wird. Auch andere Bitgrößen sind hier möglich, die Bereiche müssen dann entsprechend angepasst werden. Üblich ist für die Beschreibung der Farben die Vektordarstelleung $Farbe = (R, G, B)$.
Durch diese vektorielle Darstellung lässt sich der RGB-Farbraum durch einen Würfel, den sogenannten RGB-Farbwürfel visualisieren. In Abbildung 4.1 ist dieser Würfel dargestellt. Links sind der Ursprung ($Schwarz = (0, 0, 0)$) sowie die drei Farbachsen

der Grundfarben zu sehen. In der Mitte ist eine Außenansicht des Würfels dargestellt. Rechts wird gezeigt wie sich spezielle Farben finden lassen, falls ihre Farbanteile bekannt sind beziehungsweise wie man die Farbanteile einer Farbe bestimmen kann. Um Farbabstände im RGB-Farbraum zu bestimmen, wird im Allgemeinen der euklidsche Abstand (siehe 4.11) zwischen zwei Farben berechnet.

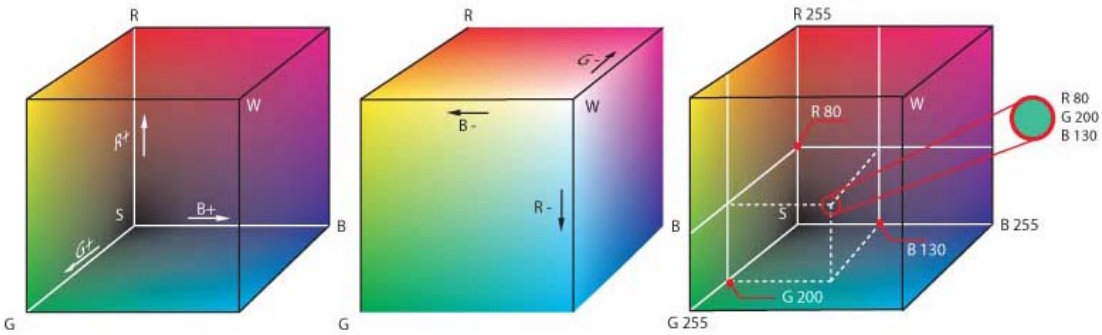

Abbildung 4.1: In dieser Abbildung ist der RGB-Farbraum durch einen Würfel dargestellt. In der linken Abbildung ist eine Innenansicht, in der mittleren dagegen eine Außenansicht des Würfel zu sehen. Im rechten Teil der Abbildung wird demonstriert, wie sich eine Farbe durch ihre speziellen RGB-Werte im Würfel auffinden lässt. (Quelle: [Wika])

4.9 HSV-Farbraum

Im HSV-Farbraum werden die darstellbaren Farben durch deren Farbton (engl. hue), ihrer Farbsättigung (engl. saturation) sowie deren Hellwerts (engl. value) angegeben. Die Koordinaten einer Farbe werden wie folgt bestimmt :

- **Farbton** : Wird durch einen Farbwinkel H auf dem Farbkreis. Dabei enstpicht 0° der Farbe Rot, 120° entspricht Grün und 240° entspricht Blau.
- **Sättigung** : Sie wird durch einen Prozentwert S im Intervall [0,1] anegegeben. 0 entspricht „neutralgrau“, 1 enstpricht einer reinen, gesättigten Farbe.
- **Hellwert** : Wird durch den Prozenwert V im Intervall [0,1] angegeben. 0 entsprich keiner Helligkeit, 1 entspricht voller Helligkeit.

In Abbildung 4.2 ist die schematische Darstellung des HSV-Raumes durch einen Kegel zu sehen. In Abbildung 4.3 wird dagegen veranschaulicht, wie sich eine spezielle Farbe in diesem Raum finden lässt, beziehungsweise wie sich die Koordinaten einer Farbe aus diesem bestimmen lassen. Der Farbton einer bestimmten Farbe ergibt sich dabei durch die Angabe eines Winkels des Farbkreises des HSV-Farbraumkegels aus Abbildung 4.2. Die Sättigung ergibt sich durch den prozentualen Abstand von der Mittelachse in Richtung des Kegelmantels. Ein Wert von 0 bedeutet, dass sich die Farbe auf der Mittelachse befindet. Entsprechend besagt ein Wert von 100, dass die Farbe auf dem Kegelmantel zu finden ist. Der Hellwert hingegen bestimmt sich durch eine Prozentangabe die widerspiegelt, auf welcher Höhe sich die Farbe im Kegel

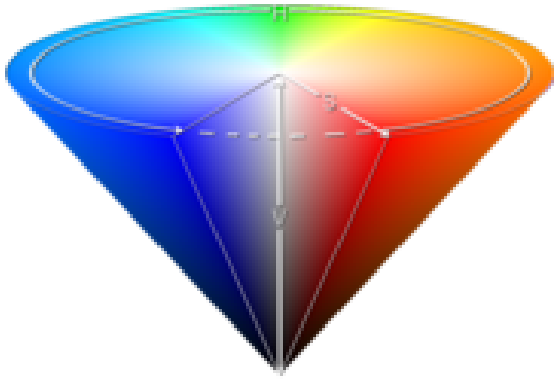

Abbildung 4.2: In dieser Abbildung ist der HSV-Farbraum durch eine kegelförmige Darstellung visualisiert. (Quelle: [Wikb])

befindet. Befindet sich die Farbe in der Spitze, so ist dieser Wert 0. Auf dem Farbkreis dagegen hat sie den Wert 1.

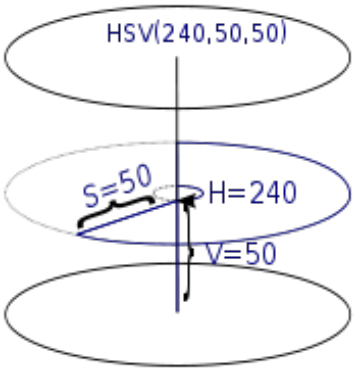

Abbildung 4.3: In dieser Abbildung wird veranschaulicht, wie sich die HSV-Werte für spezielle Farben im HSV-Farbraum ergeben. Der Farbton einer bestimmten Farbe ergibt sich dabei durch die Angabe eines Winkels des Farbkreises des HSV-Farbraumkegels (Abbildung 4.2). Sättigung sowie Hellwert werden dagegen in Prozent angegeben. Bei der Sättigung gibt dieser an, welchen prozentualen Abstand die Farbe von der Mittelachse bis zum Rand besitzt. Die Prozentzahl des Hellwertes gibt an, in welchem Verhältnis sich der Punkt auf der Mittelachse vom Boden entfernt, befindet. (Quelle: [Wikb])

Der Vorteil dieses Farbraumes ist die Ähnlichkeit zur menschlichen Farbwahrnehmung. So fällt es dem Menschen hier beispielsweise leichter, eine bestimmte Farbe im Raum zu finden.

4.10 Helligkeit

Die Helligkeit ist ein Maß dafür, wie viel Licht ein Bereich ausstrahlt. Zu einer Farbe im RGB-Farbraum kann die Helligkeit durch Addition ihrer drei Farbkomponenten bestimmt werden. Demnach gilt : $H_v = (R + G + B)$ für einen RGB-Farbvektor $v = (R, G, B)$. Farben gleicher Helligkeit haben in der Summe ihrer Farbkomponente den selben Wert. Alle Farben gleicher Helligkeit befinden sich im RGB-Würfel auf einer Ebene senkrecht zu der Geraden durch die Punkte $(0, 0, 0)$ und $(1, 1, 1)$.

4.11 Euklidscher Abstand

Der euklidsche Abstand d zwischen zwei Punkten $x = (x_1, ..., x_n)$ und $y = (y_1, ..., y_n)$ des n-dimensionalen Raumes $\mathbb{R}^n$ ist durch die euklidsche Norm des Differenzvektors der beiden Punkte x und y definiert.

$$d(x, y) = \|x - y\|_2 = \sqrt{(x_1 - y_1)^2 + ... + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Der euklidsche Abstand erfüllt die Bedingungen einer Metrik. Er stellt demnach eine Abbildung dar, die je zwei Elementen eines Raumes einen nicht negativen reellen Wert zuordnet, der als Abstand der Elemente angesehen werden kann. Somit werden folgende Bedingungen erfüllt:

1. $d(x, y) \geq 0$, das heißt dass zwei Punkte einen nicht negativen Abstand besitzen
2. $d(x, y) = 0 \Leftrightarrow x = y$, das heißt dass nur identische Punkte einen Abstand von 0 besitzen
3. $d(x, y) = d(y, x)$, damit ist die Abstandsbestimmung symmetrisch
4. $d(x, y) \leq d(x, z) + d(z, y)$, damit gilt die Dreiecksungleichung

4.12 Stochastik

4.12.1 Dichtefunktion

Die Dichtefunktion kann zur Bestimmung der Wahrscheinlichkeit, dass eine Zufallsvariable zwischen den reellen Zahlen a und b liegt, verwendet werden:
Sei Ω eine überabzählbare Ereignissmenge einer Zufallsvariablen X, ω ein Ereignis dieser Menge. Die Zufallsvariable sei durch die Abbildung $X : \Omega \to \mathbb{R}$ gegeben. Eine Funktion $f_x : \mathbb{R} \to \mathbb{R}$ wird Dichtefunktion genannt, falls gilt:

$$P([\omega : a \leq X(\omega) \leq b]) = \int_a^b f_x(t)dt$$

Damit lässt sich die Wahrscheinlichkeit, dass die Wahrscheinlichkeit eines Ereignisses ω zwischen zwei reellen Zahlen a und b liegt, durch das oben angegebene Integral

bestimmen. Dieses berechnet die Fläche zwischen der Dichtefunktion und der x-Achse im Intervall $[a, b]$.
Eine wichtige Eigenschaft der Dichtefunktion ist, dass diese normiert ist. Das heißt, die Fläche zwischen der Funktion und der x-Achse beträgt 1. Damit gilt:

$$\int_{-\infty}^{\infty} f_X(t)dt = 1$$

Diese Fläche entspricht einer Wahrscheinlichkeit von 100 Prozent.

4.12.2 Erwartungswert

Als Erwartungswert einer Zufallsvariablen versteht man denjenigen Wert, der sich bei oftmaligen Wiederholen des Zufallsexperiments als Mittelwert ergibt.
Sei X eine diskrete Zufallsvariable, die die Werte $(x_i)_{(i\in I)}$ mit den dazugehörigen Wahrscheinlickeiten $(p_i)_{i\in I}$ für eine abzählbare Indexmenge I besitzt. Dann berechnet sich der Erwartungswert, falls er existiert, zu :

$$E(X) = \sum_{i\in I}(x_i \cdot p_i) = \sum_{i\in I}(x_i \cdot P(X = x_i))$$

Besitzt die Zufallsvariable X eine Dichtefunktion f, dann hat sie einen endlichen Erwartungswert falls das Integral

$$\int_{-\infty}^{\infty} |x| \cdot f(x)dx$$

konvergiert. Dieser berechnet sich dann zu

$$E(X) = \int_{-\infty}^{\infty} x \cdot f(x)dx$$

4.12.3 Varianz und Standardabweichung

Die Varianz ist ein stochastisches Maß, das angibt, wie stark eine Zufallsvariable X von ihrem Erwartungswert $E(X)$ abweicht.
Sei $\mu = E(X) < \infty$ der Erwartungswert einer reellen eindimensionalen Zufallsvariablen X. Dann ergibt sich die Varianz von X zu :

$$Var(X) = E((X - \mu)^2)$$

Sei X eine diskrete Zufallsvariable, die die Werte $(x_i)_{(i\in I)}$ mit den dazugehörigen Wahrscheinlickeiten $(p_i)_{i\in I}$ für eine abzählbare Indexmenge I besitzt. Dann berechnet sich der Erwartungswert zu :

$$Var(X) = \sum_{i\in I}\left((x_i - \mu)^2 \cdot p_i\right)$$

Besitzt die Zufallsvariable X eine Dichtefunktion f, so berechnet sich die Varianz dagegen zu :

$$Var(X) = \int (x - \mu)^2 f(x)dx$$

Nachteilig für die Praxis ist der Umstand, dass die Varianz eine andere Einheit besitzt als der ihr zugrunde liegenden Daten. Wie aus obiger Formel ersichtlich ist, ist die berechnete Einheit das Quadrat der entsprechenden Datenmenge. Daher wird auch häufig die sogenannte Standardabweichung als Streuungsmaß verwendet. Diese ist definiert als die positive Quadratwurzel der Varianz. Damit gilt :

$\sigma_X = \sqrt{Var(X)}$ und somit ist ${\sigma_X}^2 = Var(X)$

4.12.4 Gauß-Verteilung, Normal-Verteilung

Die Gauß-Verteilung ist eine in der Stochastik wichtige Wahrscheinlichkeitsverteilung und ist auch unter den Namen Gaußfunktion oder Gauß-Glocke bekannt. Häufig wird sie zur Beschreibung zufälliger Abweichungen von einem vorgegebenen Wert verwendet.
Sei X ein Zufallsvariable mit der Dichtefunktion $f : \mathbb{R} \to \mathbb{R}^+$, $x \mapsto f(x)$. Diese Funktion heißt (μ, σ)-normalverteilt, wobei μ der Erwartungswert und σ die Standardabweichung von X ist, falls gilt :

$$f(x) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{1}{2} \cdot (\frac{x-\mu}{\sigma})^2}$$

Häufig schreibt man dann auch $X \sim \mathcal{N}(\mu, \sigma^2)$ oder einfach X ist (μ, σ)-normalverteilt. In Abbildung 4.4 sind die Schaubilder verschiedener Gauß-Verteilungen zu sehen. In ihnen sind auch die im Folgenden genannten Eigenschaften der Verteilungen ersichtlich.

- **Hochpunkt**
  Sei $f(x)$ wie oben definiert, dann berechnet sich die Ableitung nach der Kettenregel zu $f'(x) = -\frac{x-\mu}{\sigma^2} \cdot f(x)$. Die Gleichung besitzt dann für $x = \mu$ eine Lösung. Da $f''(\mu) < 0$ besitzt die Funktion an dieser Stelle einen Hochpunkt mit den Koordinaten $H(\mu|\frac{1}{\sigma \cdot \sqrt{2\pi}})$. Damit ist dieser eindeutig durch die Parameter μ sowie σ bestimmt. Je größer σ, desto kleiner ist auch seine y-Koordinate.

- **Symmetrie**
  Jede Gaußverteilung ist achsensymmetrisch zu einer parallelen der y-Achse mit der Gleichung $x = \mu$

- **Wendepunkt**
  Sei $f(x)$ wie oben definiert, dann berechnet sich die zweite Ableitung zu $f''(x) = \frac{1}{\sigma^2}(\frac{1}{\sigma^2}(x-\mu)^2 - 1) \cdot f(x)$. Für die Gleichung $f''(x) = 0$ ergeben sich die zwei Lösungen $x_1 = \mu + \sigma$ sowie $x_2 = \mu - \sigma$. Eingesetzt in $f'''(x)$ ergeben Werte $\neq 0$. Je größer σ, desto weiter liegen die zwei Wendepunkte voneinander entfernt.

- **Normierung**
  Die Fläche zwischen f(x) und der x-Achse beträgt 1. Daher gilt

  $$\int_{-\infty}^{\infty} f(x)dx = 1$$

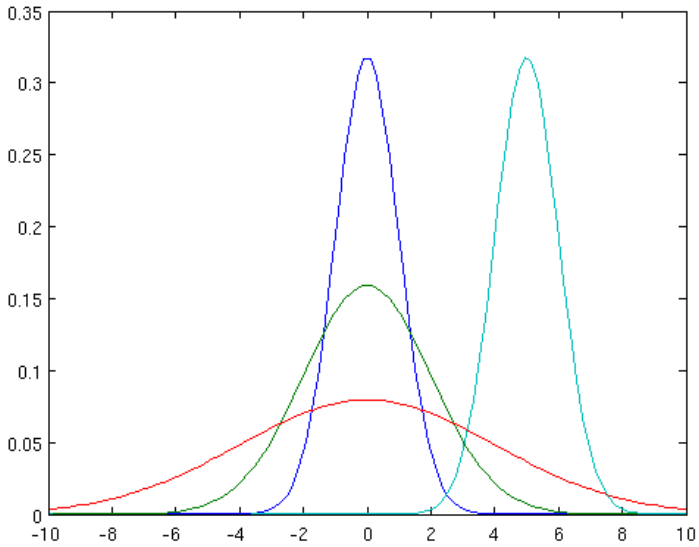

Abbildung 4.4: In dieser Abbildung sind verschiedene Gaußfunktionen zu sehen. Die Funktionen die zu dem dunkelblauen, dem roten sowie der grünen Graphen gehören, besitzen den gleichen Mittelwert. Ihre Varianzen nehmen in dieser Reihenfolge zu. Je größer die Varianz, desto breiter verläuft das Schaubild und desto kleiner ist die y-Koordinate des Hochpunktes. Die Varianz der Funktion die zu dem hellblauen Graphen gehört stimmt mit der des dunkelblauen überein, besitzt jedoch einen anderen Mittelwert

Aus diesen Eigenschaften lässt sich beispielsweise schlussfolgern, dass der Erwartungswert $E(x)$ die x-Koordinate des Hochpunktes direkt bestimmt. Je größer σ, desto kleiner ist dessen y-Koordinate und desto breiter ist der Verlauf des Graphen der Verteilung.
Die Gaußfunktion lässt sich als eine Dichtefunktion (Abschnitt 4.12.1) einer Zufallsvariablen interpretieren. Der komplette Flächeninhalt zwischen der Funktion und der x-Achse beträgt 1, welcher genau 100 Prozent entspricht. In der Nähe des Hochpunktes befindet sich bei jeder Gaußfunktion der Großteil der Fläche zwischen der Funktion und der x-Achse. In Abbildung 4.5 ist aufgeführt, wie viel Prozent dieser Fläche sich innerhalb eines von σ abhängigen Intervalls befindet. Es ergeben sich folgende Flächeninhalte :

- $\int_{\mu-\sigma}^{\mu+\sigma} f(x)dx \approx 0.682$
- $\int_{\mu-2\cdot\sigma}^{\mu+2\cdot\sigma} f(x)dx \approx 0.954$
- $\int_{\mu-2.5\cdot\sigma}^{\mu+2.5\cdot\sigma} f(x)dx \approx 0.988$
- $\int_{\mu-3\cdot\sigma}^{\mu+3\cdot\sigma} f(x)dx \approx 0.997$

So befinden sich beispielsweise 99,7 Prozent der Fläche zwischen der Gaußfunktion und der x-Achse innerhalb von $\pm 3 \cdot \sigma$ um den Mittelwert μ.

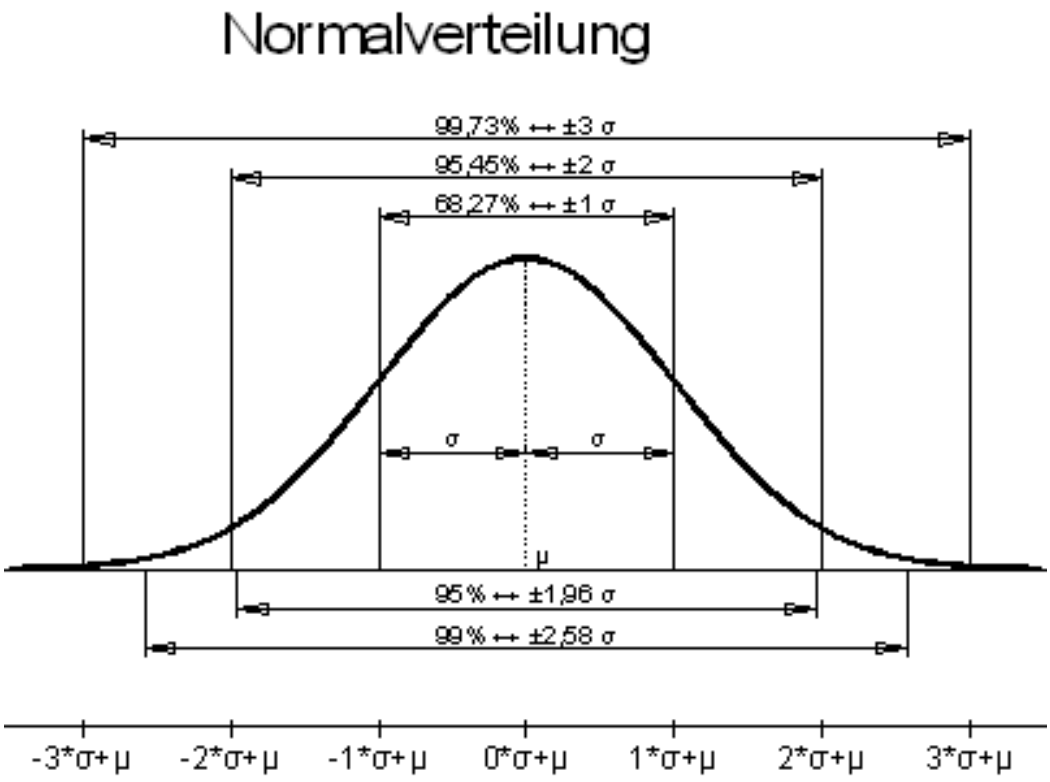

Abbildung 4.5: In dieser Abbildung ist die Größe der Flächeninhalte zwischen einer Gaußfunktion und der x-Achse in von der Standardabweichung abhängigen Intervallen um den Mittelwert zu sehen.(Quelle: [Rap])

5 Related Work

Durch den Einsatz automatisierter Überwachungssysteme fallen unausweichlich riesige Datenmengen an, die analysiert werden müssen. Algorithmen und Verfahren, die diese Analyse schnell und mit hoher Genauigkeit durchführen können, werden hierfür benötigt, da der Einsatz von Arbeitskräften sehr kostenintensiv ist. Eine Personaleinsparung hat nicht nur den Vorteil niedrigerer Kosten, sondern führt zudem zu einem erhöhten Einsatz solcher Systeme. Daher haben sich in den letzten Jahren viele Forschungsgruppen mit der Entwicklung automatisierter Überwachungssysteme beschäftigt. Die Arbeiten können im Allgemeinen folgenden Kategorien zugeordnet werden:

- Vorstellung eines neu entwickelten Analyseverfahrens oder einer hierfür benötigten Teilaufgabe, worunter auch die *Background Subtraction* Verfahren gehören
- Verbesserung eines bestehenden Verfahrens
- Behandlung eines typischen Problems bei Überwachungsaufgaben
- Verbesserungsmöglichkeiten der Subtraktionsbilder durch Nachbearbeitungsverfahren
- Evaluation eines oder mehreren Verfahren beziehungsweise Teilaufgaben

Veröffentlichte Arbeitden decken oftmals mehrere dieser Kategroien ab. Neuentwicklungen beziehungsweise Verbesserungen beinhalten in der Regel eine Evaluation, um deutlich zu machen, wie gut das Verfahren oder die Verbesserung tatsächlich ist.
Im nächsten Abschnitt dieses Kapitels, werden grundlegende Arbeiten dieser Kategorien kurz vorgestellt. Neben älteren, grundlegenden Arbeiten werden hier auch neue Arbeiten aufgeführt und deren Inhalte, Resultate oder Besonderheiten zusammengefasst. Sie geben einen Überblick über automatisierte Überwachungsysteme und betrachten insbesondere den Einsatz von *Background Subtraction* Verfahren.

5.1 Veröffentlichte Arbeiten zur automatisierten Videoüberwachung

Eine gute Übersicht über die Arbeitsweise und Herausforderungen automatisierter Überwachungssysteme sowie eine Vorstellung der hierfür eingesetzten Algorithmen, bietet das von Omar Javed und Mubarak Shah veröffentlichte Buch mit dem Titel *Automated Multi-Camera Surveillance - Algorithms and Practice* [JS08]. Im Wesentlichen werden folgende Themen behandelt:

- Automatisierte Überwachungssysteme, sowie die Vorstellung eines solchen

- Verfahren zur Detektion von Vordergrundobjekten sowie Möglichkeiten, diese zu kategorisieren
- Trackingverfahren, die mit einer oder mehreren Kameras arbeiten

Bezüglich der Detektion von Vordergrundobjekten, wofür üblicherweise *Background Subtraction* Verfahren eingesetzt werden, bieten die Autoren eine umfassende Auflistung bestehender Verfahren, wobei sie deren Konzepte hervorheben. Weiterhin erläutern sie kurz bei welchen Herausforderungen typischerweise Probleme auftreten und welche schon gut gelöst sind. Die Autoren heben hervor, dass neben Farbinformationen noch viele weitere Eigenschaften für die Berechnung der Subtraktionsbilder verwendet werden. Hierzu gehören unter anderem Helligkeits-, Kanten-, Textur- oder Tiefeninformationen. Auch der Einsatz von *Hidden Markov* Modellen wird hier erwähnt. Zudem stellen die Autoren ein *Mixture of Gaussian* Verfahren vor, welches um eine kantenbasierte Subtraktion erweitert wurde, um eine Verbesserung gegenüber Beleuchtungsänderungen zu erhalten.
Sie weisen abschließend darauf hin, dass das Überwachen von Menschenmengen sowie die Erkennung menschlicher Aktivitäten Einsatzmöglichkeiten automatisierter Überwachungssysteme sind, für die die bestehenden Verfahren noch verbessert werden müssen.

Massimo Piccardi veröffentlichte unter dem Titel *Background subtraction techniques: a review* [Pic04] eine Zusammenfassung der Arbeitsweisen sowie Eigenschaften gängiger *Background Subtraction* Verfahren. Zu diesen gehören das Differenzbildverfahren 7.1, die *Median-* 7.5, *Running Average-* 7.3, *Running Gaussian-* 7.4, *Mixture of Gaussian-* 7.8 Methoden, der Kernel-Dichteschätzer [EHD00], eine *Mean Shift* Schätzung [HCD04] sowie das *Eigenbackground* [ORP00] Verfahren. Es ist bei dieser Auswahl jedoch darauf hinzuweisen, dass diese Verfahren schon einige Jahre alt sind und daher viele aktuelle Methoden nicht beinhalten.
Der Autor wies in seiner Arbeit darauf hin, dass für eine ausführliche Evaluation gute Testdaten benötigt werden, um aussagekräftige Resultate zu erhalten. Diese würden aber bislang fehlen. Aus diesem Grund wurde im Rahmen dieser Diplomarbeit eine Überwachungsszene sowie Vordergrundmasken durch die Animationssoftware *Maya* erstellt.

Rita Cucchiara liefert unter dem Titel *People Surveillance* eine umfassende Übersicht über automatisierte Überwachungssysteme, deren Arbeitsweisen, generell auftretenden Problemen und Lösungsansätzen. Der Fokus liegt dabei auf der Überwachung von Personen. Dabei stellt sie auch einige grundlegende *Background Subtraction* Verfahren und geht auf deren Eigenschaften ein. Zudem widmet sie sich dem *Tracking* von Personen und den dabei auftretenden Problemen, wie beispielsweise Verdeckungen.

Weiming Hu et al. bieten in ihrer Arbeit [HTWM04] einen Überblick über Verfahren, die zur Analyse der Bewegungen von Objekten, speziell Menschen sowie Fahrzeuge, sowie deren Verhalten eingesetzt werden können. Dabei ist die Überwachung nicht auf eine Kamera beschränkt, sondern kann mittels mehreren erfolgen. Zudem zeigen sie Möglichkeiten zur Modellierung einer überwachten Umgebung auf. Häufig geschieht dies durch zweidimensionale Modelle, wie sie die *Background Subtraction* Verfahren verwenden. Mittels Tiefeninformationen können jedoch auch

dreidimensionale Modelle entwickelt werden, die eine höhere Robustheit gegenüber Verdeckungen liefern und Objektpositionen für eine Verhaltensanalyse liefern. In dieser Diplomarbeit werden jedoch nur die zweidimensionalen Modelle behandelt.

Ismail Haritaoglu et al. haben ein echtzeitfähiges Überwachungssystem entwickelt, mit dessen Hilfe sich Menschen in Videodaten detektieren und überwachen lassen [HHD98].
Das Besondere dieses Systems gegenüber anderen ist, dass es völlig ohne Farbinformationen auskommt, aber laut Autoren auch im Freien gute Resultate erzielt. Zur Personendetektion werden die Intensitätsinformationen der Pixel sowie zusätzliche Tiefeninformationen, die von speziellen Tiefenkameras stammen, verwendet. Dieses Verfahren wird im Rahmen dieser Diplomarbeit nicht evaluiert, da eine künstliche Überwachungsszene verwendet wird und somit die Tiefeninformationen nicht verwendet werden können. Zudem enthält Farbe wichtige Information für die Klassifikation, so dass ein Verfahren das nur bezüglich Grauwerten klassifizieren kann, im Nachteil wäre.

In ihrer Arbei beschäftigten sich Boult et al. [BMGE01] speziell mit den bei einem Einsatz von Videoüberwachungssystemen für militärische Szenarien auftretenden Problemen. Neben einer Vorstellung der auftretenden Probleme stellen sie ein zweischichtiges *Background Subtraction* Verfahren vor, das sie mit Hilfe von *ROC*- Kurven evaluieren (das Konzept der *ROC*- Kurven wird in Abschnitt 9.1.2 vorgestellt). In der in Kapitel 9 aufgeführten Evaluation, werden die *ROC*- Kurven zur Parameteroptimierung der Verfahren verwendet.
Background Subtraction Verfahren, die für militärische Zwecke im Freien eingesetzt werden sollen, müssen laut Autoren mit den folgenden Schwierigkeiten umgehen können:

- Die Beleuchtung im Freien kann sich langsam und kontinuierlich oder aber auch jederzeit sehr plötzlich ändern.
- Bäume, Büsche und Wolken bewegen sich über die Zeit, bleiben nicht konstant an einer Position. Diese Bewegungen sind uninteressant und dürfen die Analyse des Systems nicht negativ beeinflussen.
- In der Regel müssen Personen oder Fahrzeuge sehr früh und schnell detektiert werden, auch wenn diese noch weit entfernt sind und nur sehr klein innerhalb der Überwachungsbilder zu sehen sind. Das ist sehr problematisch, wenn die Videobilder verrauscht sind, da eventuell nicht zwischen Rauschen und korrekt detektiertem Vordergrundobjekt unterschieden werden kann.
- Die zu detektierenden Ziele wollen in der Regel nicht erkannt werden. Sie versuchen sich häufig zu Tarnen (englisch. *to camouflage*) so dass sie sich nicht stark vom Hintergrund der Szene abheben. Die eingesetzten Verfahren müssen demnach sehr sensitiv sein.
- Viele Ziele bewegen sich sehr langsam entweder weil sie sich noch sehr weit weg befinden oder aber keine Aufmerksamkeit auf sich lenken wollen. Die Adaption des Hintergrunds an die Szene darf nicht zu schnell erfolgen, so dass diese sich nicht zu schnell in den Hintegrund einarbeiten.

- Verdeckungen, besonders in bewaldeten Gebieten sind ebenfalls sehr problematisch, da ein einzelnes Vordergrundobjekt eventuell nicht als eine zusammenhängende Region wahrgenommen wird.

Der Fokus dieser Arbeit richtet sich somit speziell den Herausforderungen, die sich bei einem Einsatz in militärischen Szenarien ergeben. Die Einstellung der Parameter der *Background Subtraction* Verfahren wird in solchen Szenarien anders zu wählen sein als beispielsweise in Gebäuden oder städtischer Umgebungen. Ein sorgfältiges Wählen der Parameter in Abhängigkeit des Szenarios ist daher sehr wichtig. Viele der hier genannten Aspekte werden in der im Rahmen dieser Diplomarbeit durchgeführten Evaluation untersucht.

5.2 Evaluationen von Background Subtraction Verfahren

5.2.1 Robust Techniques for Background Subtraction in Urban videos

Sen-Ching S. et al. haben in ihrer Arbeit [KC04] verschiedene *Background Subtraction* Verfahren bezüglich ihrem Einsatz in städtischen Umgebungen im Freien, evaluiert. In den zur Evaluation verwendeten Testvideos sind Personen und Fahrezeuge, aber auch Hintergrundbewegungen, Beleuchtungsänderungen, verschiedene Wetterbedingungen wie Regen, Nebel oder Schnee, enthalten.
Folgende Verfahren wurden evaluiert:

1. Differenzbildverfahren (Kapitel 7.1)
2. *Approximated Median* Filter [MS95]
3. Kalman Filter [KB90]
4. Median Filter [CGPP03]
5. *Mixture of Gaussian* ([SG99])

Um zu evaluieren, wie gut die Verfahren bezüglich den verwendeten Überwachungsvideos abschneiden, wurden die *Precision* und *Recall* Metrik unter Variationen der Parameter der Verfahren berechnet (diese Metriken werden in Abschnitt 9.1.1 vorgestellt). *Ground Truth* Daten wurden für einige Videobilder gegen Ende der Sequenzen erzeugt und verwendet. In der folgenden Auflistung sind die erzielten Resultate sowie die Schlussfolgerungen die die Autoren aus diesen ziehen konnten, zu sehen:

- Das *Mixture of Gaussian* Verfahren schnitt am besten ab, dicht gefolgt von dem Median Filter. Das Differenzbildverfahren erzielte die schwächsten Resultate.
- Der *Approximated Median* Filter liefert erstaunlich gute Resultate, wenn man die Einfachheit seines Modells berücksichtigt. Jedoch ist dieses Verfahren nicht in der Lage, sich so schnell wie die anderen Verfahren an plötzliche Beleuchtungsänderungen oder allgemeine Szenenänderungen anzupassen.
- Der Kalman-Filter lieferte häufig schwache Subtraktionsbilder. Sich bewegende Vordergrundobjekte hinterlassen in diesen häufig einen *Schweif*, da das Hintergrundmodell durch sie häufig zu stark beeinträchtigt wird.

- Je nach Einsatzszenario kann es mal mehr und mal weniger sinnvoll sein, das Hintergrundmodell schnell an die Überwachungsszene anzupassen. Für die verwendeten Videos sollte die Anpassung nicht zu schnell erfolgen, da sonst beispielsweise Fahrzeuge die an roten Ampeln halten müssen in den Hintergrund aufgenommen werden. In der Regel haben die Parameter der Verfahren einen großen Einfluss auf die Adaptionsgeschwindigkeit und sollten daher sinnvoll gewählt werden.
- Zwar hat das *Mixture of Gaussian* Verfahren bei der Evaluation am besten abgeschnitten, jedoch besitzt es seine eigenen Probleme. Es ist rechenaufwändiger als die anderen Verfahren und schneidet bei plötzlichen Beleuchtungsänderungen relativ schwach ab, besonders wenn die Szene zuvor sehr statisch war, da dann die entsprechenden Varianzen relativ niedrig sind.

Insgesamt ist jedoch anzumerken, dass für die Evaluation keine Farbvideos verwendet wurden. Farbe liefert jedoch sehr wichtige Informationen, so dass für die in Kapitel 9 aufgeführte Evaluation nur Farbvideos verwendet wurden. Interessant wäre auch ein Vergleich des *Mixture of Gaussian* Verfahrens mit einem weiteren multimodalen Hintergrundmodell gewesen.

5.2.2 Herrero

Sonsoles Herrero und Jesús Bescós beschäftigten sich in ihrer 2009 veröffentlichten Arbeit mit dem Titel *Background Subtraction Techniques : Systematic Evaluation and Comparative Analysis* mit der Evaluation verschiedener *Background Subtraction* Verfahren. Dabei untersuchten sie speziell, wie schnell die einzelnen Verfahren arbeiten, für welche Parametereinstellungen sie besonders gute Ergebnisse liefern und welche Verfahren für welche Szenarien am besten geeignet zu sein scheinen.
Die evaluierten Verfahren sind der folgenden Auflistung zu entnehmen :

- Differenzbild-Verfahren [EF03]
- Median Filtering [KC04]
- Simple Gaussian [WADP97]
- Mixture of Gaussians [SG99]
- Gamma-Methode [Ebr05]
- Histogramm basierter Ansatz [MP04]
- Kernel-Dichteschätzer [EHD00]

Um ausreichend viele *Ground Truth* Daten (siehe Abschnitt 4.3) für die Tests zur Verfügung zu haben, wurden die in [TEBM]beschriebenen Datensätze verwendet. Für deren Erzeugung wurden Vordergrundobjekte in einem Studio so gefilmt, dass diese sich in den entstehenden Videobildern problemlos pixelgenau von dem dabei verwendeten Hintergrund segmentieren lassen. Diese Objekte wurden dann in verschiedene, separat aufgenommene Szenen integriert. Durch ein solches Vorgehen lassen sich Videos mit ausreichend vielen *Ground Truth* Daten, die für eine aussagekräftige Evaluation notwendig sind, generieren. Jedoch sollte dabei beachtet werden, dass die

Beleuchtungsverhältnisse der Szene keinen Einfluss auf die Vordergrundobjekte haben. Es lassen sich weder Schatten noch Lichteffekte oder ähnliche Phänomene auf diese Art erzeugen. Für die hier durchgeführte Evaluation wurden letztlich verschiedene Szenen ausgesucht, deren Hintergründe unterschiedlich starke Dynamik aufwiesen. Um diese Effekte sowie eine ausreichend große Menge an *Ground Truth* Daten zu erhalten, wurde für die in Kapitel 9 durchgeführte Evaluation ein Szene mittels einer Animationssoftware erzeugt.
Zunächst wurden die angesprochenen Verfahren in Matlab implementiert und auf die gewählten Testvideos angewendet. Dabei wurde die durchschnittliche Berechnungsdauer der Verfahren pro Videobild bestimmt. Die dabei erzielten Ergebnisse sind in der folgenden Tabelle aufgeführt.

Verfahren	FD	MF	SG	G	MoG	KDE	Hb
$\frac{Sekunden}{Videobild}$	0,0287	0,0299	0,0279	0,0127	4,8341	2,2027	13,6456

Tabelle 5.1: Berechungsgeschwindigkeiten der von Herrero evaluierten Verfahren

Es ist zu bemerken, dass die erzielten Ergebnisse durch den Einsatz von besserer Hardware oder durch trickreichere Implementierungen, falls diese möglich wären, steigerbar sind. Daher lassen sich diese Resultate nur zur groben Abschätzung der Laufzeiten verwenden. Es ist jedoch ersichtlich, dass die aufwendigeren Hintergrundmodelle offensichtlich mehr Rechenzeit benötigen.
Neben der Laufzeitberechnung wurden die Verfahren auch bezüglich ihrer Leistung evaluiert. Die hierfür eingesetzten Metriken sind in Abschnitt 9.1.3aufgeführt. Bei ihnen handelt es sich um eine Metrik zur Bestimmung der erzielten Genauigkeit P (engl. Precision) sowie einer zur Bestimmung der erzielten Trefferquote R (engl. Recall), die für die Berechnung des F-Maßes kombiniert werden. Diese Metriken basieren auf der Berechnung der in Abbschnitt 9.1.1 aufgeführten Metriken TP,TN,FP sowie FN und werden wie folgt bestimmt:

$$P_0 \leftarrow \frac{TN}{TN+FN}\ ,\ R_0 \leftarrow \frac{TN}{TN+FP}\ ,\ P_1 \leftarrow \frac{TP}{TP+FP}\ ,\ R_1 \leftarrow \frac{TP}{TP+FN}$$

Die Indizes 0 und 1 stehen dabei für den Hintergrund beziehungsweise Vordergrund. Die erhaltenen Werte werden nun zur Bestimmung des F-Maßes eingesetzt :

$$F_0 \leftarrow \frac{2 \cdot P_0 \cdot R_0}{P_0 + R_0}\ ,\ F_1 \leftarrow \frac{2 \cdot P_1 \cdot R_1}{P_1 + R_1}$$

Das F-Maß wurde nun für die Verfahren unter verschiedenen Parametereinstellungen berechnet. Als optimaler Parameter eines Verfahrens wurde dann derjenige gewählt, der den höchsten durchschnittlichen Wert für F, bezogen auf die zur Evaluation gewählten Testvideos, erzielen konnte.
Die Autoren haben zudem eine Tabelle veröffentlicht, in der die für die optimalen Parameter erzielten Werte für F für die jeweiligen Videos, aufgelistet wurden. Die Auswertung dieser Tabelle ergab im Wesentlichen folgende Resultate :

1. Die Gamma-Methode konnte im Durchschnitt am besten abschneiden

2. Bei starker Hintergrundbewegung lieferten das *Mixture of Gaussians Verfahren* und das *Median-Filtering* die besten Ergebnisse. Erstaunlich hierbei ist, dass das *Median-Filtering* eigentlich ein unimodales Hintergrundmodell besitzt, also nicht speziell für sehr dynamische Hintergründe entwickelt wurde.
3. Für weniger Hintergründe mit niedriger Dynamik schnitten das *Simple Gaussian Verfahren*, das *Mixture of Gaussian Verfahren* sowie die Gamma-Methode am besten ab.

Zuletzt führten die Autoren in ihrer Arbeit noch ihre subjektive Meinung bezüglich der visuellen Qualität der von den Verfahren erzielten Resultate. Hierfür wurde betrachtet, wie gut die einzelnen Verfahren die Vordergrundobjekte detektieren konnten, wie stark der Hintergrund der Szene durch Rauschen beeinträchtigt wurden und wie einfach die Einstellung geeigneter Parameter erfolgen konnte. Die folgenden Resultate sind einzig durch die Einschätzungen der Autoren entstanden und haben das Ziel, einen groben Eindruck über die erzielten Leistungen zu geben.

Verfahren	FD	MF	SG	G	MoG	KDE	Hb
Vordergrund	–	-	+	++	++	○	-
Hintergrundrauschen	○	○	-	-	-	+	+
Parameterwahl	++	++	○	-	-	-	+

Tabelle 5.2: Einschätzung der Leistung der von Herrero et al. evaluierten Verfahren. Die Angaben haben folgende Bedeutungen: sehr gut(++), gut(+), durchschnittlich(○), schwach(-)

5.2.3 Vergleich verschiedener *Background Subtraction* Verfahren für Szenen mit statischem Hintergrund

In ihrer Arbeit mit dem Titel *Comparison of Static Background Segmentation Methods* evaluierten Musatfa Karaman et al. [LGYS05] verschiedene gängige unimodale *Background Subtraction* Verfahren mittels *ROC*- Kurven und verschiedenen Metriken (diese sind in Kapitel 9 aufgeführt) evaluiert. Diese Methoden wurden in dieser Arbeit kurz vorgestellt und deren Arbeitsweisen visualisiert. Die Arbeiten folgender Autoren wurden für diese Evaluation herangezogen: Haritaoglu [HCD04], Francois [FM], Horprasert [HHD99], McKenna [MJD+00], Jabri [JDWR00], Cavallaro [CE02], Hong [HW03] und Shen [She04].
Die Autoren erzielten folgende Resultate, beziehungsweise konnten sie die folgenden Schlussfolgerungen aus diesen ziehen:

- Die erzielten Resultate der einzelnen Verfahren konnten, abhängig von den Videos, stark unterschiedlich sein.
- Die Verfahren von Shen, Cavallaro, Jabri und Horprasert konnten bezüglich des F-Maßes die besten Resulate erzielen. Das von Haritaoglu das schwächste.
- Die Metrik *Precision* kann dazu verwendet werden, die Tendenz zur Übersegmentierung eines Verfahrens zu erhalten. Die Methode von Shen konnte hier am besten abschneiden. Die Verfahren von Hong, Horprasert und McKenna

konnten noch gute Resultate erzielen. Die Methode von Haritaoglu schnitt dabei am schlechtesten ab.

- Durch die Metrik *Recall* kann dagegen der Hang zur Untersegmentierung eines Verfahrens ermittelt werden. Die Methoden von Cavallaro, Jabri und Fracois konnten hier die besten Resultate erzielen. Die Methode von Hong konnte hier das schwächste Resultat erzielen.
- Komplexere Methoden, die nicht nur Farbinformationen sondern auch beispielsweise Kanteninformationen für die Berechnung des Subtraktionsbildes heranziehen, konnten auch bessere Resultate erzielen. Eine geeignete und ausgewogene Auswahl an verwendeten Informationen ist daher laut Autoren besonders wichtig. Jedoch ist darauf hinzuweisen, dass auch einige der verwendeten Verfahren Nachverarbeitungsmethoden verwenden, so dass sie gegenüber denen die keine verwenden, im Vorteil sind.

Interessant wäre noch der Vergleich mit multimodalen *Background Subtraction* Verfahren, auch bezüglich nicht statischer Hintergrundszenen gewesen. Die in Kapitel 9 aufgeführte Evaluation beinhaltet eine Mischung aus unimodalen sowie multimodalen Hintergrundmodellen.

5.2.4 Evaluierung von Background Subtraction Algorithmen mit Post-Processing

Donovan Parks und Sidney Fels beschäftigten sich mit der Frage, wie stark sich die Resultate von *Background Subtraction* Verfahren durch Nachbearbeitungsschritte, sogenanntem *Post-Processing*, verbessern lassen. Dazu wählten sie zunächst einige gängige *Background Subtraction* Verfahren aus und evaluierten sie bezüglich 13 ausgewählten Testvideos mit verschiedenen Parametereinstellungen. Die Videos wurden so gewählt, dass viele gängige Herausforderungen in ihnen enthalten waren. Hierzu gehören unter anderem Bäume, die sich im Wind bewegen, Wasseroberflächen mit Reflektionen sowie Vordergrundobjekte die sich farblich nicht wesentlich vom Hintergrund der Szene abheben (Tarnung).
Zur Evaluation der Verfahren wurden sogenannte *Precission-Recall* Tests durchgeführt und zusätzlich das *F-Maß* bestimmt. Die Verfahren lassen sich so bezüglich ihrer dabei erzielten Ergebnisse vergleichen. Zudem lassen sich ihre jeweils besten Parametereinstellungen bestimmen. Diese ergeben sich direkt aus der Einstellung, die zum besten Ergebnis führt.
Folgende *Background Subtraction* Verfahren wurden evaluiert :

- *Running Gaussian* [WADP97]
- *Mixture of Gaussians* [SG99]
- *Mixture of Gaussians* mit einer adaptiven Anzahl Gaußfunktionen (AGMM) [ZH06]
- *Median Filtering* [SRPC06]
- *Approximated Median Filtering* (AMF) [ZH06]
- *Mediod Filtering* [CGPP03]

- *Eigenbackgrounds* [ORPoo]

Insgesamt ziehen die Autoren folgende Schlüsse :

- Das *Median Filtering* hat besonders starke Probleme bei Bäumen, die sich im Wind bewegen sowie bei der Tarnung
- Für hohe *Recall-* Werte ($Recall \geq 0,85$) liefert das *Mediod* Verfahren die besten Resultate
- Für $Recall \leq 0,8$ erzielten die beiden *Mixture of Gaussian* Verfahren die besten Ergebnisse
- Keines der Verfahren konnte konstant gute Subtraktionsbilder erzeugen. Diese Erkenntnis motivierte letztlich den Einsatz von Nachbearbeitungsverfahren.
- Multimodale Hintergrundmodelle arbeiten besser als unimodale
- Das Verwenden einer adaptiven Anzahl an Gaußfunktionen verbessert die Qualität der Subtraktionsbilder nicht wesentlich. Dafür konnten schnellere Berechnungszeiten sowie ein geringerer Speicherverbrauch festgestellt werden.
- Das *Approximated median Filtering* kann gute Resultate bei niedriger Laufzeit und geringem Speicherbedarf verbuchen
- Das *Eigenbackground* Verfahren erzielt zwar gute Resultate, jedoch ist zu beachten, dass es sich nicht an Änderungen des Hintergrunds der Szene anpassen kann.

Im Folgenden sind die verwendeten Nachbearbeitungsverfahren sowie deren Arbeitsweisen und eventuelle Parameter aufgeführt.

- Entfernen von Rauschen durch einen Filter, der einen Vordergrundpixel aus dem Subtraktionsbild entfernt, falls nicht mindestens ρ Pixel seiner Achternachbarschaft ebenfalls Vordergrundpixel sind. Für die Evaluation wurden verschiedene Werte für ρ verwendet.
- Morphologisches Schließen. Hierfür wurden unterschiedlich große quadratische Strukturelemente verwendet.
- Entfernen von Vordergrundregionen, die aus weniger als a Pixeln bestehen.
- Durchführung eines Salienztestes. Vordergrundobjekte, bei denen sich nicht mindestens γ Prozent ihrer Pixel deutlich vom Hintergrund abheben, werden aus dem Subtraktionsbild entfernt. Ein Vordergrundpixel unterscheidet sich deutlich vom Hintergrund, wenn er den Ähnlichkeitstest auch unter verschärften Bedingungen nicht bestehen würde. Verwendet ein Verfahren beispielsweise einen Schwellwert T, so kann hierfür ein Wert $T_{sal} = 2 \cdot T$ verwendet werden.
- Einsatz eines *Optical Flow Tests*. Bewegen sich Objekte für eine gewisse Zeit nicht, so werden sie in das Hintergrundmodell integriert. Bewegen sie sich nach einer solchen Integration wieder, so wird im Subtraktionsbild ein Vordergrundobjekt an dieser Stelle sichtbar, obwohl sich in der Szene dort keines mehr befindet. Man spricht hierbei von sogenannten *Geistern*. Da solche Geister sich nicht bewegen, können sie entfernt werden, wenn sich der Wert ihres *Optical Flows* nahe bei 0 befindet.

- Durch den Einsatz eines selektiven Aktualisierungsmechanismus (siehe Abschnitt 4.7 für nähere Informationen), lassen sich die eben beschriebenen Geister in den Subtraktionsbildern vermeiden.

Diese Nachbearbeitungsverfahren wurden auf die Subtraktionsbilder von vier der zuvor genannten Verfahren angewendet. Hierbei handelt es sich um das AGMM , Mediod , AMF , sowie um das *Eigenbackground* Verfahren. Zunächst wurden Nachbearbeitungsverfahren einzeln eingesetzt und evaluiert, dann wurden diese kombiniert (siehe hierfür Abbildung 5.1).

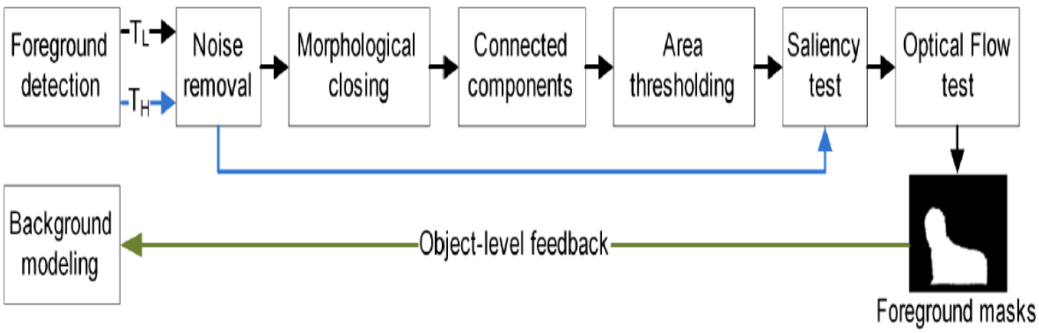

Abbildung 5.1: Hier ist eine vorgeschlagene sequentielle Kombination von Nachbearbeitungsverfahren zu sehen

Aus der durchgeführten Evaluation kamen die Autoren zu folgenden Schlussfolgerungen :

1. Der Rauschfilter sollte nicht zu stark filtern, das heißt ρ sollte relativ klein sein. Dann sind Verbesserungen erzielbar, wenn auch nur geringe.
2. Auch das morphologische Schließen führt zu etwas verbesserten Ergebnissen. Jedoch sollte die Größe des Strukturelements nicht zu groß gewählt werden.
3. Das Entfernen zu kleiner Regionen sollte mit einem Wert für a durchgeführt werden, der in etwa 25 Prozent der zu erwartenden Pixelanzahl der kleinsten Objekte entspricht. Jedoch wird ein zuvor durchgeführtes morphologisches Schließen dringend empfohlen, da Vordergrundobjekte häufig kleine nicht zusammenhängende Lücken aufweisen.
4. Der *Optical Flow* Test ist besonders nützlich, wenn man Objekte mit geringer Geschwindigkeit nicht in den Subtraktionsbildern haben möchte. Jedoch muss eine sorgfältige Wahl getroffen werden, wie hoch eine solche Geschwindigkeit anzusetzen ist. In den verwendeten Videosequenzen waren niedrige Werte notwendig.
5. Die Selektion bietet kaum Verbesserungen. Da sie jedoch geringe Laufzeit- und Speicheranforderungen stellt, kann sie bedenkenlos verwendet werden, zumal sie auch das Entstehen der Geister verhindert.
6. Ist die verfügbare Rechenleistung ausreichend groß, so sollten alle Nachbearbeitungsverfahren, wie in Abbildung 5.1 zu sehen, verwendet werden.

7. Einzig eine Optimierung der Parameter eines *Background Subtraction* Verfahrens ist nicht ausreichend. Eine optimale Wahl der bei den Nachbearbeitungsverfahren gewählten Parametern ist ebenfalls von großer Bedeutung.

Für die in Kapitel 9 aufgeführte Evaluation wurden keine Nachbearbeitungsverfahren eingesetzt. Jedoch wäre es interessant zu untersuchen, ob die Verfahren bei den beobachteten Problemen durch diese Nachbearbeitungsverfahren verbessern werden können.

5.2.5 Perturbation - Methode zum Evaluieren von *Background Subtraction* Verfahren

T.H. Chalidabhongse et al. veröffentlichten im Jahre 2003 ein Verfahren [Cha03], das zur Evaluation von *Background Subtraction* Verfahren eingesetzt werden kann. Dieses stellt fest, wie sensitiv ein Verfahren arbeitet. Genauer gesagt misst es, wie stark sich ein Vordergrundobjekt farblich vom Hintergrund einer Szene abheben muss, um erkannt werden zu können.
Zur Erzeugung von *ROC*- Kurven (diese werden in Kapitel 9 vorgestellt), die ebenfalls zur Evaluation von *Background Subtraction* Verfahren eingesetzt werden können, müssen *Ground Truth* Daten (siehe Abschnitt 4.3) der Szene vorhanden sein. Da die Erzeugung dieser Daten sehr aufwendig ist, werden diese häufig nur für wenige Videobilder einer Sequenz erzeugt. Für aussagekräftige Resultate wird jedoch eine genügend große Menge benötigt. Der Vorteil des von den Autoren entwickelten Verfahrens besteht genau darin, dass diese *Ground Truth* Daten für die Evaluation nicht benötigt werden.
Um herauszufinden wie stark sich ein Vordergrundobjekt von dem Hintergrund der Szene unterscheidet, muss ein Ausschnitt des berechneten Hintergrundmodells eines Verfahrens solange farblich verändert beziehungsweise *gestört* werden, bis dieser detektiert werden kann. Aus diesem Vorgehen leitet sich der Name dieses Verfahrens ab, da *to perturbate* übersetzt *stören* bedeutet. Zur Durchführung dieser Störung werden alle Farbwerte der Pixel des gewählten Ausschnittes um den Wert Δ in zufällige Richtungen innerhalb des RGB-Farbraumes verschoben. Δ wird sukzessiv erhöht und die Detektionsrate des Verfahrens bezüglich der so veränderten Farbwerte bestimmt.
Die Autoren haben nun 4 *Background Subtraction* Verfahren bezüglich der *Perturbation* Methode evaluiert. Diese sind das *Mixture of Gaussian* [SG99] und das *Codebook* Verfahren, eine kernelbasierte Methode [EHD00] sowie ein unimodales Verfahren [HHD99]. Die Tests haben ergeben, dass das Codebookverfahren und das unimodale Modell die besten Resultate lieferten. Das *Mixture of Gaussian* Verfahren lieferte die schwächsten Resultate.
Es ist darauf hinzuweisen, dass das Verfahren nur die Sensitivität der Verfahren misst. Auch Effekte wie Reflektionen oder Schatten, die durch Vordergrundobjekte entstehen, können nicht mit einbezogen werden. Bislang existieren keine Arbeiten, die sich mit dieser Evaluationsmethode ausgiebig beschäftigten. In der in Kapitel 9 aufgeführten Evaluation wurde daher diese Methode nicht verwendet.

6 Herausforderungen

Background Subtraction Verfahren, die bei der automatischen Videoüberwachung eingesetzt werden, müssen mit einer Vielzahl an Herausforderungen und Problemen klar kommen. Dieses Kapitel widmet sich diesen Problemen ausführlich und zeigt auf, weshalb sie den Verfahren Schwierigkeiten bereiten. Zudem dienen sie als Grundlage der Evaluation in Kapitel 9. Diese beschäftigt sich mit der Frage, wie gut die einzelnen *Background Subtraction* Verfahren unter diesen Herausforderungen abschneiden. Speziell soll geklärt werden, welche Probleme von keinem Verfahren in den Griff bekommen wird, so dass sich weitere Forschungsarbeiten gezielt auf diese konzentrieren können.

6.1 Laufzeit

Um automatisierte Videoüberwachungssysteme , beispielsweise zur Prävention von Straftaten, sinnvoll einsetzen zu können, ist es wie in Abschnitt 3.2 erläutert notwendig, dass diese beim Eintreten einer kritischen Situation möglichst früh Alarm schlagen, so dass beispielsweise Sicherheitskräfte rechtzeitig eingreifen können. Um dies zu ermöglichen, müssen die Systeme die erzeugten Videodaten in Echtzeit auswerten können. Kann dies ein System nicht leisten, so kann es höchstens zur Aufklärung verwendet werden und verfehlt dadurch präventive Zwecke.
Wie in Abschnitt 3.4 erwähnt, stehen die *Background Subtraction* Verfahren meist zu Beginn einer Sequenz von Arbeitsschritten, die zur Analyse der Videodaten benötigt werden. Wenn die Analyse in Echtzeit durchgeführt werden soll, so muss das eingesetzte *Background Subtraction* Verfahren so schnell arbeiten, dass für die noch folgenden Arbeitsschritte ausreichend Zeit bleibt, um die Echtzeitanforderung insgesamt erfüllen zu können.
Zu Beginn der Automatisierung der Überwachungssysteme wurden zunächst sehr einfache *Background Subtraction* Verfahren für relativ kleine Überwachungsbilder eingesetzt. Ein wesentlicher Grund hierfür ist in der damals verfügbaren Hardware zu finden, die den Einsatz etwas aufwendigerer und besserer Verfahren nicht zuließen. Jedoch ist der Einsatz solcher Systeme heute möglich, da sich die Hardwareleistung in den letzten Jahren kontinuierlich und signifikant erhöht hat.
Eine ausführliche Laufzeitmessung wurde im Rahmen dieser Diplomarbeit nicht umgesetzt. Eine solche würde erfordern, dass alle durchgeführten Tests unter den selben Rahmenbedingungen durchgeführt werden. Da die Hardware einen erheblichen Anteil an der Laufzeit hat, ist eine solcher Test wenig aussagekräftig. Würde man bessere Hardware verwenden, würde man auch bessere Laufzeiten erhalten. Auch die verwendete Programmiersprache spielt hier eine wichtige Rolle. Alle in dieser Diplomarbeit evaluierten Verfahren wurden in Matlab oder OpenCV implementiert. Die Laufzeiten in OpenCV liegen dabei weit unter denen von Matlab. Zudem können

eventuell kleinere Laufzeitverbesserungen durch trickreiche Implementierungen erhalten werden. Um aussagekräftige Evaluationen durchführen zu können, müssten solche Tricks für jedes oder für keines der Verfahren angewendet werden, falls solche existieren.
Daher werden für die implementierten Verfahren keine exakten Laufzeitmessungen durchgeführt, sondern Laufzeitabschätzungen, die die zugrunde liegenden Ideen in O-Notation ausdrücken, angegeben.

6.2 Trainingsdaten

Als Trainingsdaten werden bei den *Background Subtraction* Verfahren die Videobilder einer Videosequenz zu Beginn eines Überwachungsvideos bezeichnet. Eine solche Sequenz wird auch Trainingssequenz genannt und wird zur Berechnung eines initialen Hintergrundmodells verwendet. Die verschiedenen Verfahren haben auch unterschiedliche Anforderungen an die bereitgestellte Trainingssequenz, von denen einige in der folgenden Aufzählung vorgestellt werden.

- **Länge**
  Damit ein initiales Hintergrundmodell den Hintergrund auch möglichst gut widerspiegelt, muss die bereitgestellte Sequenz genügend Informationen für dessen Charakterisierung beinhalten. Um dies zu ermöglichen muss unter anderem die Länge der Sequenz ausreichend groß sein.
- **Verdeckungen**
  Ist der Hintergrund einer Szene eine längere Zeit beziehungsweise über die Sequenz häufiger von zum Beispiel durchlaufenden Personen bedeckt, so spiegelt sich dieser Umstand im berechneten initialen Hintergrundmodell wieder. In Abbildung 6.1 ist ein hierfür Beispiel zu sehen.

Abbildung 6.1: Ein gutes, initiales Hintergrundmodell in Trainingsseqeunzen bei starken Verdeckungsproblemen zu berechnen, ist ein schwieriges Problem.

In dem von PETS2007 (siehe Abschnitt 4.1) bereitgestellten Testvideo ist ein großer Teil des Hintergrunds über die komplette beziehungsweise nahezu komplette Sequenz nicht sichtbar. Daraus folgt, dass die Verfahren für diese Bereiche auch kein korrektes Modell berechnen können, da die Videobilder an den entsprechenden Stellen die dafür benötigten Informationen nicht bereit stellen.
In Abbildung 6.2 sind die Bilder zweier initialer Hintergrundmodelle zu sehen, die bezüglich einer Sequenz berechnet wurden, bei der im rechten Teil der Videobilder ein Teil des Hintergrunds nur kurzzeitig durch eine laufende Person verdeckt wurde. Im linken Bild ist das Modell des Mittelwertverfahren zusehen, im rechten das des Medianverfahrens. Während beim Mittelwertverfahren auch kurzzeitige Verdeckungen das Hintergrundmodell beeinflussen, muss der Hintergrund beim Medianverfahren mindestens 50 Prozent der Sequenzlänge verdeckt sein, bevor dieser verfälscht wird.

- **Beleuchtungsänderungen**
  Beleuchtungsänderungen werden in Abschnitt 6.5 in diesem Kapitel noch ausführlich besprochen. Ein Verfahren, das *Eigenbackground* Verfahren (siehe) das laut dessen Entwickler sehr gut auch mit plötzlichen und starken Beleuchtungsänderungen zurecht kommt, benötigt eine Trainingssequenz, die die Szene bezüglich verschiedenen Beleuchtungen beinhaltet. Erfüllt die Trainingssequenz diese Anforderung nicht so kommt das Verfahren mit diesen Beleuchtungsänderungen auch nicht zurecht. Das generieren einer geeigneten Trainingssequenz, gerade in Außenszenen, gestalltet sich als häufig ziemlich aufwendig und schwierig, da Beleuchtungsänderungen im Allgemeinen nicht beeinflusst werden können.

Abbildung 6.2: Links ist ein durch das Mittelwert Verfahren berechnetes initiales Hintergrundmodell zu sehen. Rechts eines, das durch das Median Verfahren bezüglich der selben Trainingssequenz berechnet wurde. Kurzzeitige Verdeckungen bereiten dem Median Verfahren weniger Probleme.

Somit spielen die Länge sowie der Inhalt der Trainingssequenz für die Güte der Verfahren eine wesentliche Rolle. Dadurch ergeben sich die folgenden Fragen. Darf eine Sequenz spezielle Ereignisse beinhalten oder muss dieses sogar Inhalt der Sequenz sein? Ab welcher Dauer bezüglich des Vorhandenseins eines speziellen Ereignisses wirkt sich dieses positiv beziehungsweise negativ auf das Verfahren aus?

6.3 Geeignete Wahl der Parameter

Die *Background Subtraction* Verfahren vergleichen eingehende Videobilder pixelweise mit den in dem Hintergrundmodell gespeicherten Werten auf Ähnlichkeit. Ist die Ähnlichkeit zwischen dem Pixel und dem Hintergrund gegeben, so wird dieser Pixel als Vordergrund, ansonsten als Hintergrund eingestuft. Die dabei durchzuführenden Tests arbeiten im Allgemeinen mit Hilfe von Paramtern, deren Werte im Prinzip frei gewählt werden dürfen. Jedoch haben sie auf das Ergebnis des Ähnlichkeitstest, dessen Ergebnis das Aussehen des Subtraktionsbildes bestimmt, einen starken Einfluss.
Das in Abschnitt 7.2 vorgestellte Mittelwertverfahren klassifiziert die Pixel in Vorder- beziehungsweise Hintergrundpixel, indem es den Farbabstand zwischen einem zu untersuchenden Pixel und dem in dem Hintergrundmodell an dieser Position gespeicherten Farbwert berechnet. Ist dieser Abstand nicht größer als ein bestimmter Schwellwert, so ist der Pixel dem Hintergrund ähnlich und wird daher als Hintergrundpixel klassifiziert. Ist der Abstand größer als der Schwellwert, so wird der Pixel als dem Hintergrund unähnlich und damit als Vordergrund eingestuft.
Der Wert des Schwellwertes, gibt damit direkt an, wie weit der Farbwert eines Pixels von dem ihm enstprechenden Hintergrundmodells im (RGB-Würfel, siehe Abschnitt 4.8) höchstens entfernt sein darf, um als Hintergrund deklariert zu werden. Die Wahl dieses Schwellwertes hat somit signiffikanten Einfluss auf das entstehende Subtraktionsbild.

6.4 Veränderungen der Szene

Im Rahmen meiner Studienarbeit habe ich das Mittelwert (siehe hierfür Abschnitt 7.2) sowie das *Median* Verfahren (siehe Abschnitt 7.5) für ein System zur *Left Luggage Detection* implementiert. Beide Verfahren wurden für die von PETS (siehe Kapitel 4.1) bereitgestellten Testvideos verwendet. Eine Aktualisierung die das Hintergrundmodell an Veränderungen in der Szene anpasst, wurde dabei nicht implementiert, da die Videosequenzen alle recht kurz waren und keine Szenenänderungen beinhalteten. Eine Aktualisierung des Hintergrundmodells wäre gerade für solch kurze Videos in diesem Kontext recht problematisch. Sobald ein Objekt in den Hintergrund eingearbeitet wurde, wird es solange es sich nicht bewegt, von dem System nicht mehr erkannt. Geschieht ein solches Einarbeiten zu schnell, kann das System ein abgestelltes Objekt nicht mehr erkennen. Da aber genau das die Aufgabe des Systems ist, darf eine solche Einarbeitung nicht zu schnell geschehen. Daher wurde bei solch kurzen Videos auf eine Aktualisierung verzichtet. Eine Aktualisierungsmöglichkeit kann dem System bei Bedarf jedoch einfach zugefügt werden.
Da Überwachungssysteme aber im Allgemeinen rund um die Uhr eingesetzt werden, müssen Änderungen der Szene, wie beispielsweise Beleuchtungsänderungen bei Szenen im Freien, regelmäßig in das Modell eingearbeitet werden. Die Geschwindigkeit der Aktualisierung ist applikationsabhängig. Manche Aufgaben verbieten eine rasche Adaption, andere wiederum benötigen diese. Wie eine Aktualisierung des Hintergrunds durchgeführt wird, ist vom jeweiligen Modell abhängig und wird jeweils in den entsprechenden Abschnitten in Kapitel 7 für die evaluierten Verfahren erläutert.
Zudem lassen sich durch regelmäßiges Aktualisieren des Hintergrundmodells Fehler,

die beispielsweise durch Verdeckungen während der Trainingsphase in das Modell gelangen können, wieder aus ihm entfernen.

6.5 Beleuchtungsänderungen

Beleuchtungsänderungen der zu überwachenden Szene, stellen eine weitere Herausforderung für die *Background Subtraction* Verfahren dar. Im wesentlichen unterscheidet man zwei Arten von Beleuchtungsänderungen. Zum einen die langsamen, schwachen Änderungen. Sie treten beispielsweise bei der Überwachung von Szenen im Freien im Laufe eines Tages auf und resultieren aus den Veränderungen des Sonnenstandes. Zum anderen kommen gerade im Freien häufig auch plötzliche, starke Beleuchtungsänderungen vor. Häufig treten diese durch Verdeckungen des Sonnenlichts durch Wolken auf. Schiebt sich eine Wolke vor die Sonne, so wird die Szene innerhalb kurzer Zeit deutlich dunkler. Verdeckt sie die Sonne nicht mehr, so wird die Szene plötzlich wieder wesentlich heller. Auch in Gebäuden kann es zu plötzlichen, stärken Beleuchtungsänderungen kommen, zum Beispiel durch das An- beziehungsweise Ausschalten von Lampen, oder wenn sich die überwachte Szene in der Nähe von Fenstern befindet.
In Abbildung 6.3 ist das aus diesem Umstand resultierende Problem zu sehen. In der Abbildung sind drei Frames eines Verkehrsüberwachungsvideos zu sehen, die zu verschiedenen Zeitpunkten aufgenommen wurden. Die komplette Szene beinhaltet keine Vordergrundobjekte, so dass das *Background Subtraction* Verfahren auch keinen Pixel als Vordergrund klassifizieren sollte. In der Szene ist jedoch eine Beleuchtungsänderung, resultierend aus dem Zusammenspiel zwischen der Sonne und einer Wolke zu, sehen. In der unteren Abbildung sind die berechneten Subtraktionsbilder zu den entsprechenden Zeitpunkten dargestellt. Hierbei wird offensichtlich, dass die plötzliche Beleuchtungsänderung zu vielen Fehlklassifikationen führt. Dies liegt an der Funktionsweise der *Background Subtraction* Verfahren, wie sie in Abschnitt 3.3 vorgestellt wurden. Pixelweise werden Tests durchgeführt, wie ähnlich der Farb- beziehungsweise Grauwert dem des Hintergrundmodells an dieser Stelle ist. Durch die Helligkeitsveränderung wird der Unterschied so groß, dass diese dem Hintergrund nicht mehr als ähnlich angesehen werden können und somit fälschlicherweise als Vordergrund klassifiziert werden.

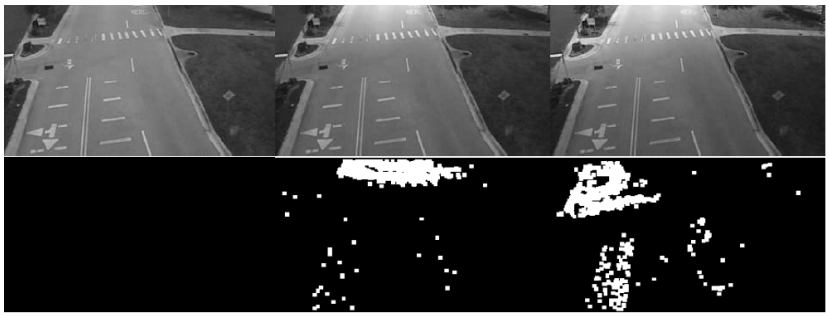

Abbildung 6.3: In dieser Abbildung ist zu sehen, wie durch Beleuchtungsänderungen Pixel fälschlicherweise als Vordergrund klassifiziert werden, da durch die Änderung der Ähnlichkeitstest zu keinem positiven resultat führt.

Die schwächeren und langsameren Beleuchtungsänderungen bereiten dagegen meist weniger große Probleme. Ändert sich die Beleuchtung nur wenig, so ändern sich auch die betroffenen Pixelwerte ebenfalls nur wenig, so dass sie die durchgeführten Ähnlichkeitstets noch bestehen. Zudem werden die kleineren Änderungen regelmäßig in das Hintergrundmodell mit eingearbeitet, da die Verfahren im Allgemeinen adaptiv sind, um sich an Veränderungen der Szene anpassen zu können (siehe hierzu Abschnitt 6.4).
Im Rahmen dieser Diplomarbeit werden die eingesetzten Verfahren auch auf deren Abschneiden bei den zwei vorgestellten Beleuchtungsänderungen untersucht.

6.6 Wetter

Auch verschiedene Wetterbedingungen können sich problematisch auf die Leistung der *Background Subtraction* Verfahren auswirken. In Abbildung 6.4 ist eine Verkehrsüberwachungsszene bei unterschiedlichem Wetter zu sehen. Die Videobilder wurden bei Schneefall und bei Schneefall aufgezeichnet, woraus zwei wesentliche Herausforderungen für das eingesetzte Verfahren resultieren.

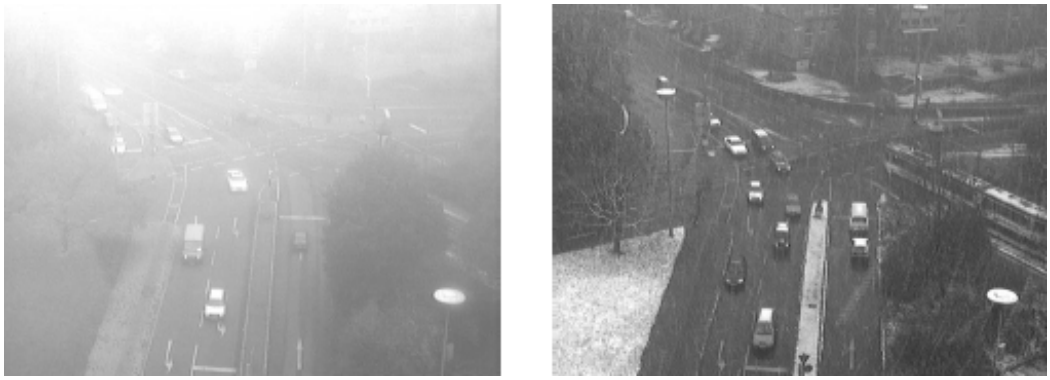

Abbildung 6.4: Schneefall, Regen oder ähnlich schlechte Wetterbedingungen erschweren die Arbeit der *Background Subtraction* Verfahren.

Die Schneeflocken *verunreinigen* das aufgezeichnete Bild dadurch, dass sie in ihm als weiße Flecke beziehungsweise Striche zu sehen sind. Hierbei verändert sich der Farb- beziehungsweise Grauwert an entsprechenden Stellen im Allgemeinen so stark, dass sich dies negativ auf die durchzuführenden Ähnlichkeitstests auswirkt. Dadurch werden an manchen Stellen eventuell Regionen fälschlicherweise als Vordergrund klassifiziert. Bei starkem Schneefall werden Objekte der Szene im Extremfall sogar verdeckt, so dass diese nicht mehr erkannt werden können. Auch beispielsweise weiße oder graue Autos können unter Umständen nicht mehr erkannt werden (Abschnitt 6.7). Regen führt zu einem ähnlichen Problem. Da dieser aber nicht weiß ist, wirken sich die Verunreinigungen im Bild nicht so drastisch aus, so dass dabei Objekte der Szene nicht komplett verdeckt werden. Da sich die bei leichtem Schneefall oder Regen entstehenden Verunreinigungen im Bild verrauschten Bildern ähnlich sind, bekommt ein Verfahren dieses Problem in den Griff, wenn es nicht besonders anfällig gegenüber Rauschen ist. Die im Rahmen dieser Diplomarbeit evaluierten Verfahren werden auch bezüglich ihrem abschneiden bei verrauschten Bildern untersucht.

Das zweite Problem das bei Schneefall auftritt, ist eine Änderung der Szene. Diese tritt beispielsweise dadurch auf, dass sich eine grüne Wiese mit der Zeit weiß färbt. Um dieses Problem in den Griff zu bekommen sollte das verwendete Verfahren eine Adaption beinhalten, so dass es sein Hintergrundmodell an die veränderte Szene anpassen kann.
Auch Nebel erschwert die Arbeit von *Background Subtraction* Verfahren. Dieser kann ähnlich wie starker Schneefall komplette Objekte verdecken, so dass diese nicht erkannt werden. Auch der Ähnlichkeitstest bei weißen oder grauen Objekten verläuft unter Umständen positiv, so dass sich diese nicht mehr vom Hintergrund unterschieden lassen können.

6.7 Tarnung

Ein Objekt gilt als getarnt, wenn sein Erscheinungsbild dazu führt, dass es in einer Szene nicht mehr oder nur durch großen Aufwand erkannt werden kann. Im allgemeinen unterscheidet man folgende Arten von Tarnung :

- **Optische Tarnung** : Wird zum Beispiel durch Tarnfarben erreicht.
- **Akustische Tarnung** : Hierunter fällt die Vermeidung von Geräuschen, zum Beispiel durch Bewegungsarmut oder Abschirmung der entstehenden Geräusche.
- **Thermische Tarnung** : Hierbei wird die Körperwärme oder Geräteabstrahlung beispielsweise durch spezielle Kleidung oder durch Abdeckungen, abgeschirmt.

Im Rahmen der *Background Subtraction* Verfahren ist natürlich speziell die optische Tarnung von besonderem Interesse. Ein typisches Einsatzgebiet hierfür ergibt sich durch das Militär. Hier werden Überwachungssysteme zur Detektion von Personen, Fahrzeugen oder sonstigen Objekten eingesetzt. Hierbei stehen die Systeme vor der Herausforderung, dass sie beispielsweise Personen erkennen sollen, die spezielle Kleidung angezogen oder ihren Körper so angemalt haben, dass sie sich farblich kaum von der Szene in der sie sich befinden, unterscheiden. Die Systeme klassifizieren die Personen dann häufig als Hintergrund, da sie dem Hintergrundmodell an ihrer Stelle im Videobild sehr ähnlich sind.
Tarnung führt jedoch nicht nur bei militärischen Einsatzgebieten zu Problemen, da sie in vielen Szenen auch ungewollt beziehungsweise unbewusst eintritt. Dieser Sachverhalt wird in Abbildung 6.5 verdeutlicht. In dieser ist links eine Person, die einen roten Pullover trägt und vor einer roten Mauer steht, zu sehen. Rechts hingegen ist das binäre Subtraktionsbild eines *Background Subtraction* Verfahrens zu sehen. Diese Fehlklassifikation durch den roten Pullover sind hier besonders deutlich erkennbar. Zudem sind weitere falsch klassifizierte Pixel aufgrund von Schattenbildungen in diesem Subtraktionsbild enthalten.

Durch die Wahl der Parameter eines eingesetzten *Background Subtraction* Verfahrens lässt sich zumindest in gewisser Weise regeln, wie gut das Verfahren ein getarntes Objekt erkennt. Genauer gesagt lässt sich dadurch steuern, wie stark sich das Objekt von dem Hintergrundmodell unterscheiden muss, dass es von dem Verfahren erkannt werden kann.
Verwendet man beispielsweise das in Abschnitt vorgestellte Mittelwert Verfahren, so ist der einzige veränderbare Parameter der Schwellwert t. Durch ihn kann geregelt

Abbildung 6.5: In dieser Abbildung ist zu sehen, wie Tarnung die Leistung der Verfahren beeinträchtigen kann. In der mittleren Abbildung ist ein Subtraktionsbild zu sehen, dass viele Vordergrundpixel nicht richtig klassifiziert.

werden, wie groß der Farabstand eines Pixels zum entsprechenden Hintergrundmodell sein muss, um diesen Pixel als Vordergrund zu klassifizieren. Je kleiner der Schwellwert gewählt wird, desto kleiner wird der Farbabstand der für diese Klassifikation verwendet wird. Gleichzeitig erhöht sich dabei die durch Rauschen verursachte Fehlklassifikation, so dass der Parameter bezüglich Rauschen und Tarnung eingestellt werden muss.
Das durch die Tarnung entstehende Problem kann in der Praxis zum Beispiel durch den zusätzlichen Einsatz von Wärmebildkameras, reduziert werden, da so farblich getarnte Objekte weiterhin detektiert werden können.

6.8 Schatten

Schatten stellen ein weiteres Problem, mit denen die verwendeten *Background Subtraction* Verfahren zu kämpfen haben, dar. Wie in Abschnitt 3.4 angesprochen und in Abbildung 3.2 zu sehen, stehen die *Background Subtraction* meist zu Beginn einer sequentiellen Folge von Arbeitsschritten. Werden durch sie Pixel falsch klassifiziert, so wirkt sich dies unter Umständen negativ auf spätere Bearbeitungsschritte, besonders auf die Analyse, aus.
Viele *Trackingverfahren*, deren Ziel die Verfolgung von Vordergrundobjekten in Videodaten ist, arbeiten aufgrund bestehender Echtzeitanforderungen mit Hilfe von Heuristiken. Häufig verwenden sie die Pixelanzahl die zu diesen Objekten gehört oder durch deren *Bounding Box*. Schatten können diese so stark vergrößern, dass die Heuristiken der *Trackingverfahren* nicht mehr korrekt arbeiten können.
Die Auswirkung von Schatten auf das entstehende Subtraktionsbild, lässt sich teilweise durch die Wahl der Parameter der Verfahren regeln. Erlaubt man durch die Parameter relativ große Unterschiede zu dem Hintergrundmodell, so lassen sich die negativen Auswirkungen durch Schatten bis zu einem gewissen Grad unterdrücken. Jedoch steigen dadurch im Allgemeinen die durch beispielsweise Rauschen verursachten Fehlklassifikationen. Ein Abwägen und sinnvolles Wählen der Parameter ist daher erforderlich.
In Abschnitt 8.5 wird ein Verfahren vorgestellt, dass die Anzahl der durch Schatten verursachten fehlklassifikationen zu reduzieren versucht.

6.9 Verdeckungen

Verdeckungen haben die negative Eigenschaft, dass durch sie zu detektierende Objekte in den Videobildern nicht, beziehungsweise nur teilweise sichtbar sind. Dadurch lassen sich vollständig verdeckte Objekte zumindest kurzzeitig nicht verfolgen und damit können deren Trajektorien nicht aktualisiert werden. Endet die Verdeckung, so lässt sich das Objekt auch wieder verfolgen.
Das Problem, mit dem ein Trackingverfahren dabei zu kämpfen hat ist, dass es erkennen muss ob sich ein Objekt schon einmal in der Szene befunden hat, oder nicht. Befand sich das Objekt schon einmal in der Szene, so kann es je nach späterer Analyseaufgabe sinnvoll sein, für dieses Objekt keine neue Trajektorie anzulegen, sondern die schon bestehende weiter zu nutzen und zu aktualisieren. Um dies zu gewährleisten muss das Trackingverfahren den Objekten ihre Trajektorien auch bei kurzzeitiger, vollständiger Verdeckungen sicher zuordnen können.
Auch die nicht vollständigen Verdeckungen können sich problematisch auswirken. Um die in Abschitt 6.1 geforderte Echtzeitfähigkeit zu erreichen, werden für viele Teilaufgaben eines Überwachungssystems sogenannte Heuristiken verwendet. Diese haben das Ziel, mit relativ geringem Wissen beziehungsweise wenigen Informationen, ihre Aufgaben möglichst gut zu lösen. So wird beispielsweise bei der Objektklassifikation die Anzahl der Pixel der detektierten Vordergrundonjekte verwendet. In Abbildung 6.6 ist hierfür ein Beispiel zu sehen. Das hier verwendete Überwachungssystem hat unter anderem die Aufgabe, die detektierten Vordergrundregionen in die Klassen *Fahrzeug* oder *Mensch* einzuordnen. Um diese Klassifikation möglichst schnell zu realisieren, wurden Regionen deren, Pixelanzahl oberhalb eines Grenzwertes liegt als *Fahrzeug* eingestuft. Die, deren Pixelanzahl darunter lagen, aber eine bestimmte Mindestgröße hatten, so dass sie mit großer Wahrscheinlichkeit nicht die Folge von Bildrauschen sind, folglich als *Mensch*. Wie in Abbildung zudem ersichtlich ist, führen die teilweisen Verdeckungen zu schlechten Ergebnissen solcher Heuristiken, da sie die Eigenschaften der detektierten Vordergrundregionen, hier speziell deren Pixelanzahl, stark verfälschen können.

Eine Möglichkeit die eingesetzten Videoüberwachungssysteme unempfindlicher gegenüber Verdeckungen zu machen, basiert auf dem Einsatz mehrerer Kameras. In Abbildung 6.7 ist die Arbeitsweise eines solchen Verfahrens zu sehen. In jeder Zeile ist eine andere Kameraperspektive der überwachten Szene aufgeführt. In der ersten Spalte ist jeweils das von der entsprechenden Kamera gelieferte Überwachungsbild zu einem bestimmten Zeitpunkt t zu sehen. In der zweiten Spalte wurden die detektierten Vordergrundobjekte, bezüglich des in der dritten Spalte gezeigten Subtraktionsbildes, grün eingefärbt. Mittels homographischer Transformationen können die detektierten Objekte in den von allen vier eingesetzten Kameras gemeinsam überwachten Bereich transformiert. In Abbildung 6.8 ist die Kombination dieser transformierten Objekte zu sehen. Durch die hier verwendeten Grauwerte, wird die Anzahl der Kameras in denen ein bestimmter Pixel sichtbar ist, codiert. Je dunkler der Grauwert, in desto mehr Kameras ist der enstprechende Objektpixel zu sehen. Um ein Objekt zu detektieren, reicht es aus, wenn ein Objekt von einer gewissen Anzahl an Kameras erfasst wurde. Diese Anzahl kann kleiner sein, als die der eingesetzten Kameras. Wird ein Objekt in einer Kamera beispielsweise durch eine Verdeckung nicht erkannt, so besteht die Möglichkeit, dass immerhin von anderen Kameras erkannt und verfolgt werden kann.

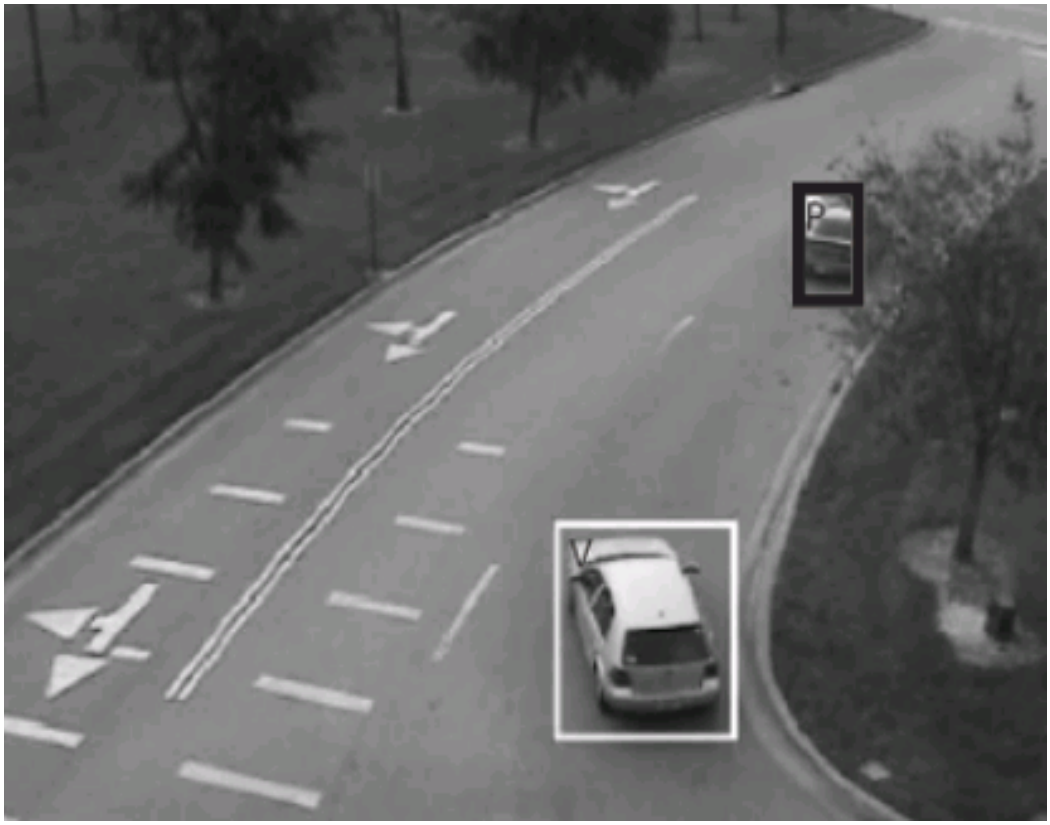

Abbildung 6.6: In dieser Abbildung ist zu sehen, wie Klassifikationsaufgaben fehlerhaft durchgeführt werden, wenn sie bezüglich der Anzahl der Pixel der jeweiligen Regionen durchgeführt wird. Durch Verdeckungen können exakte Pixelangaben nicht berechnet werden.

6.10 Uninteressante Bewegungen, periodische Wiederholungen

Wie in Abschnitt 4.2 erläutert, werden Regionen aufgrund ihrer Dynamik in der Szene als interessant beziehungsweise uninteressant eingestuft. Eine solche Einstufung dient einzig der schnellen Berechenbarkeit und erhebt keinen Anspruch auf Korrektheit. In vielen Fällen sind tatsächlich die sich bewegenden Regionen von speziellem Interesse. Jedoch existieren auch Bewegungen von für die durchzuführende Überwachungsaufgabe uninteressanten Regionen. Damit diese spätere Arbeitsschritte nicht unnötig erschweren beziehungsweise verfälschen, müssen die Verfahren so gut wie möglich erkennen können, welche Bewegungen interessant sind und welche nicht.
In der Praxis existieren eine Vielzahl an uninteressanten Bewegungen beziehungsweise Veränderungen der überwachten Szene. Einige in Überwachungsvideos häufiger auftretenden sind beispielsweise :

- Bewegungen von Blättern, Büsche und Ästen die durch den Wind verursacht werden
- Umschalten von Ampeln

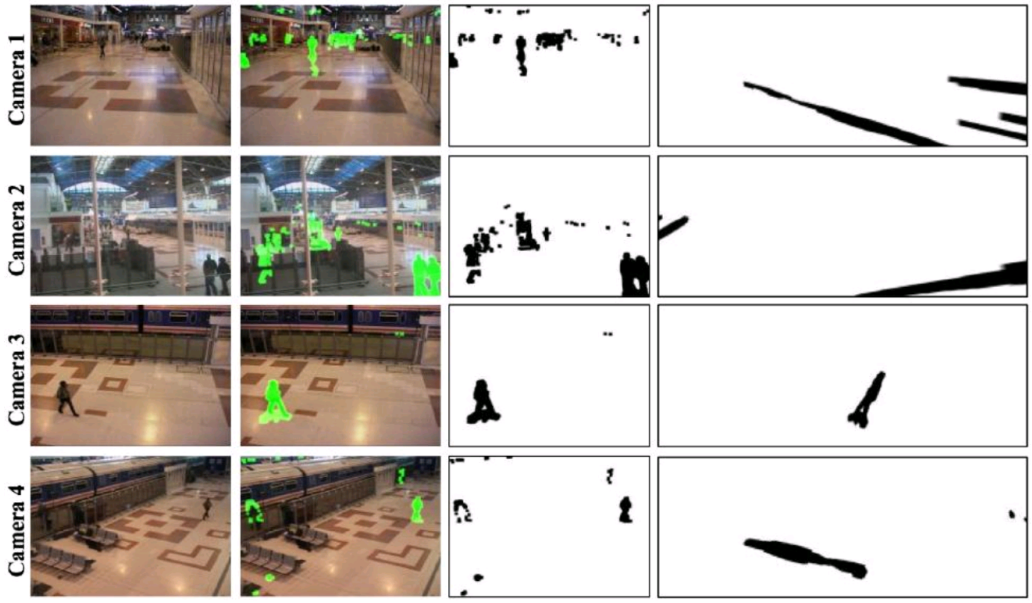

Abbildung 6.7: In dieser Abbildung ist die Arbeitsweise eines Verfahrens zu sehen, das mittels homographischen Transformationen Personen in mehreren Kameras verfolgen kann.

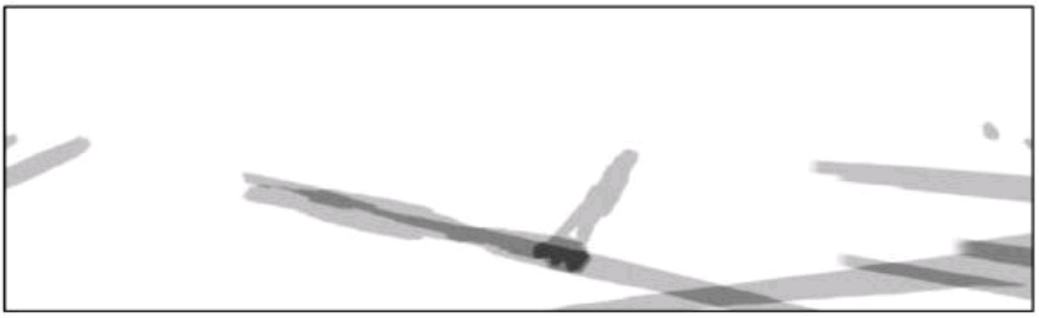

Abbildung 6.8: In dieser Abbildung ist das entstehende Fusionsbild zu sehen, in dem eine Person von mehreren Kameras detektiert werden konnte. Durch die Graustufen ist die Anzahl der Kameras codiert.

- Blinklichter die an Läden angebracht werden um auf diese aufmerksam zu machen
- Das Wehen von Fahnen
- Bewegungen von Schranken an Bahnübergängen

Die hier genannten Aspekte besitzen eine wichtige Gemeinsamkeit, sie sind in einem gewissen Sinne alle periodisch. Das bedeutet, dass sie sich regelmäßig wiederholen, wobei hier die Periodendauer nicht unbedingt einen festen Wert besitzt, sich häufig aber innerhalb eines gewissen Zeitrahmens bewegt. Man spricht hierbei auch von quasi-periodischer Wiederholung. In Abbildung 6.9 eine Grauwertverteilung eines Pixels innerhalb einer Videosequenz zu sehen, jedoch gehört der Pixel zu einem sich

bewegenden Hintergrund. Für eine solche Verteilung reicht die Beschreibung des Hintergrunds durch ein unimodales Modell nicht aus.

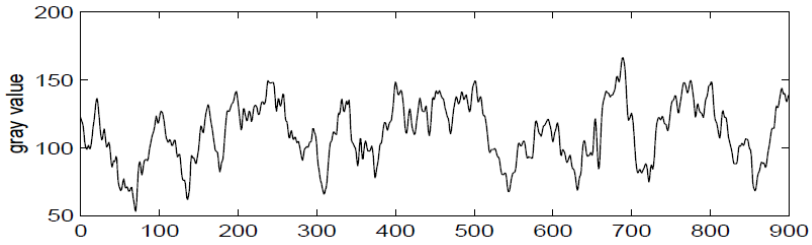

Abbildung 6.9: In dieser Abbildung ist die Grauwertverteilung eines sich quasi-periodisch Wiederholenden Hintergrundpixels zu sehen. die Beschreibung des Hintergrunds an einer solchen Position mit einem unimodalen Hintergrundmodell ist nicht ausreichend.

Um die periodischen Wiederholungen in ein Hintergrundmodell zu integrieren, erweitern viele Verfahren ihr Modell in der Hinsicht, dass sie pro Pixel mehrere Werte besitzen, bezüglich denen die Ähnlichkeitstests mit dem zu untersuchenden Pixel des Videobildes durchgeführt werden. Arbeitet ein Verfahren auf der oben genannten Videosequenz mit beispielsweise drei Hintergrundwerten pro Pixel, so resultieren daraus deutlich weniger falsch klassifizierte Pixel.
Hintergrundmodelle, die für jeden Pixel mehrere Hintergrundwerte führen, werden multimodale Hintergrundmodelle genannt.

6.11 Kamerabewegungen

Wie in Abschnitt 3.3 erläutert, führen die *Background Subtraction* Verfahren pixelweise Ähnlichkeitstests zwischen dem Hintergrundmodell und dem zu bearbeitenden Videobild, für eine entsprechende Klassifikation in Hinter- beziehungsweise Vordergrundpixel, durch. Für die durchzuführenden Tests ist es wichtig, dass sich die Positionen des Videobildes mit denen des Hintergrundmodells decken. Kommt es dagegen zu einer Kamerabewegung, so verschieben sich die Positionen des Videobildes gegenüber denen des Hintergrundmodells. Das führt letztlich dazu, dass die Ähnlichkeitstets für sich nicht entsprechenden Positionen durchgeführt werden.
In Abbildung 6.10 ist das durch Kamerabewegungen auftretende Problem bei Subtraktionsbildern veranschaulicht. Die Kanten zwischen farblich stark unterschiedlichen Regionen treten besonders deutlich hervor. Dies liegt an dem Umstand, dass der Ähnlichkeitstest durch die Verschiebung an eben solchen Stellen negativ ausfällt, während er in Regionen die farblich relativ homogen sind, noch immer zu einem positiven Resultat führt. Je inhomogener ein Videobild demnach ist, desto größer ist somit die Anzahl der aufgrund von Kamerabewegungen hervorgerufenen Fehlklassifikationen.

Im wesentlichen lassen sich zwei Arten von Kamerabewegungen unterscheiden.

1. Einsatz einer mobilen Kamera

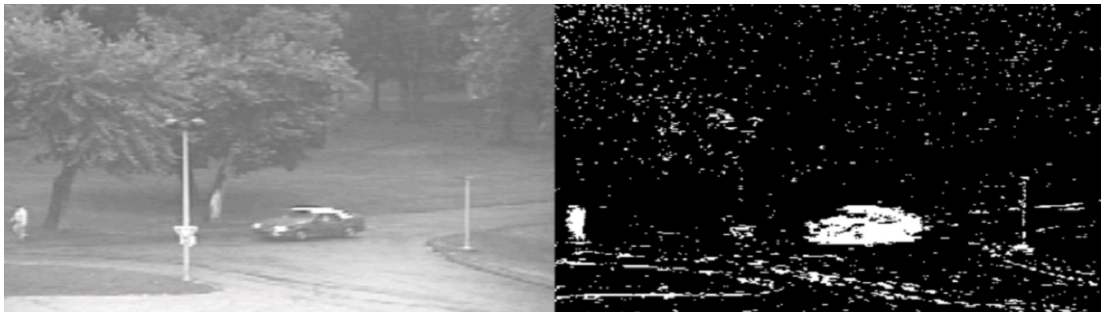

Abbildung 6.10: Herausforderung : Verwackeltes Videobild

2. Ungewollte Bewegungen, beispielsweise hervorgerufen durch Stöße oder Wind

Bei dem Einsatz mobiler Kameras lässt sich das auftretende Problem dadurch in den Griff bekommen, dass die durch Kamerabewegungen entstehende Verschiebung zwischen Hintergrundmodell und Videobild, berechnet wird. Dies ist hier relativ einfach möglich, da die durchgeführte Kamerabewegung nicht zufällig, sondern gewollt beziehungsweise geplant ist. Daher ist die Kamerabewegung bekannt und kann zur Berechnung der Verschiebung innerhalb der Überwachungsbilder herangezogen werden. Ist diese Verschiebung berechnet, so kann zu einer Bildposition der ihm enstprechende Wert des Hintergrundmodells für den Ähnlichkeitstest verwendet werden. Sind die Kamerabewegungen ungewollt, so ist die Berechnung der Verschiebung deutlich aufwendiger, da diese Bewegungen zufällig und nicht geplant erscheinen. Im allgemeinen werden hierfür Techniken der Bildregistrierung verwendet. Diese haben das Ziel, zwei oder mehr Bilder einer Szene aus unterschiedlichen Perspektiven bestmöglich in Übereinstimmung zu bringen.

7 Background Subtraction Verfahren

Dieses Kapitel widmet sich einer Vielzahl an *Background Subtraction* Verfahren die häufig in der Praxis eingesetzt werden oder interessante Ansätze beinhalten. In den einzelnen Abschnitten werden die Arbeitsweisen der Verfahren ausführlich vorgestellt, sowie deren Eigenheiten und Eigenschaften aufgezeigt und erläutert. Laufzeit- und Speicherabschätzungen zu den einzelnen Verfahren werden in O-Notation angegeben und können zum Vergleich der verschiedenen Verfahren verwendet werden. Berechnet ein Verfahren ein initiales Hintegrundmodell, so werden die Abschätzungen zur Berechnung diese Modells sowie der anschließenden Subtraktion getrennt betractet.
Das grundlegende Arbeitsprinzip der hier vorgestellten Verfahren wurde schon in Kapitel 3.3 aufgeführt und in Abbildung 3.1 schematisch dargestellt. Es besteht im Wesentlichen aus folgenden drei Komponenten: Hintergrundmodell, Subtraktionsbild sowie der Aktualisierung des Hintergrundmodells. Aus welchen Informationen der Hintergrund letztlich besteht, aus welchen Berechnungen sich die Ähnlichkeitstests zwischen den Videobildern und dem Hintergrundmodell zusammensetzen und wie die Aktualisierung des Hintergrundmodells durchgeführt wird, ist bei den einzelnen Verfahren verschieden und wird in den enstprechenden Abschnitten gezeigt.

7.1 Differenzbild Verfahren

Ein grundlegendes *Background Subtraction* Verfahren ist die Methode der Berechnung von Differenzbildern. Zur Berechnung eines Differenzbildes werden im Allgemeinen zwei gleichgroße Bilder pixelweise von einander subtrahiert. Sei D das Differenzbild, $Bild_1$ sowie $Bild_2$ die zur Berechnung von D verwendeten Bilder. D hat dabei die gleiche Bildgröße wie $Bild_1$ und $Bild_2$. Die Pixel des Differenzbildes berechnen sich nun wie folgt :

$$D(i,j) = Abstand(Bild_1(i,j) - Bild_2(i,j))$$

Das heißt, dass sich jeder Pixel des Differenzbildes aus dem Farb- oder Grauwertabstand der zu seiner Berechnung verwendeten Bilder, direkt ergibt. Bei Grauwertbildern verwendet man zur Abstandsberechnung einfach den Betrag der Subtraktion der entsprechenden Grauwerte. Bei Farbbildern kann beispielsweise die euklidsche Distanz (siehe Abschnitt 4.11) zwischen den zwei Farbwerten berechnet.
Um das Differenzbild Verfahren jedoch als *Background Subtraction* Verfahren einsetzen zu können, wird das Verfahren wie in Algorithmus 7.1 angegeben, verwendet. Zunächst werden nicht zwei beliebige Bilder zur Berechnung des Differenzbildes herangezogen, sondern zwei zeitlich direkt aufeinander folgende Videobilder $Frame_t$ sowie $Frame_{t+1}$ der Überwachungsszene. Das Hintergrundmodell besteht hier einfach aus dem Videobild $Frame_t$, also $Background_t = Frame_t$. Nach der Durchführung des Subtraktionsschrittes ersetzt $Frame_t$ das Hintergrundmodell. Somit gilt

Algorithmus 7.1 Differenzbild Verfahren

procedure FRAMEDIFFERENCING(*sequence*,*thresh*)
  for all ($FramesF_t$) **do**
    $F_t \leftarrow sequence(t)$;

    // compute subtraction result for each pixel
    for all $Pixels(x, y)$ **do**
      $distance \leftarrow dist(F_t(x, y), Background(x, y))$
      $SubtractionImage(x, y) \leftarrow \begin{cases} 1 & \text{falls } distance \geq thresh \\ 0 & \text{otherwise} \end{cases}$
    end for

    // update the background model
    $Background \leftarrow F_t$

    // store computed subtraction image at time t
    $save(SubtractionImage, t)$
  end for
end procedure

$Background_{t+1} = Frame_t$ Um das binäre Subtraktionsbild zu bestimmen, wird zunächst das Differenzbild D wie oben angegeben berechnet. Die Abstände werden jetzt noch binarisiert, indem die Abstände mit einem Schwellwert verglichen werden. Ist der Abstand größer als der Schwellwert, so ist keine Ähnlichkeit mit dem Hintergrund vorhanden und der Pixel wird entsprechend als Vordergrundpixel klassifiziert. Ist jedoch der Abstand kleiner als der gewählte Schwellwert, so gilt der Pixel aufgrund seiner Ähnlichkeit zum Hintergrund entsprechend als Hintergrundpixel.
Daher berechnet sich das Subtraktionsbild S an der Stelle (i, j) zu:

$$S(i,j) = \begin{cases} 1, & \text{falls } Abstand(Background_t, Frame_{t+1}) \geq Schwellwert \\ 0, & \text{sonst} \end{cases}$$

Zur Distanzberechnung werden im Allgemeinen die zuvor beschriebenen Methoden zur Berechnung des Unterschieds von Grau- oder Farbwerten, eingesetzt. Die Methode *save* wird zur Speicherung des berechneten Subtraktionsbildes verwendet.
Die Laufzeit dieses Verfahrens ist besonders gering. Zur Berechnung des Differenzbildes muss pro Pixel eine Subtraktion durchgeführt werden. Für die anschließende Binarisierung, das heißt der Berechnung des Subtraktionsbildes, muss pro Pixel der Abstand zwischen zwei Grau- beziehungsweise Farbwerten bestimmt werden. Die Aktualisierung wird durch eine direkte Ersetzung der Werte des aktuellen Hintergrundmodells durch die des gerade untersuchten Videobildes, durchgeführt.
Die Laufzeit des Verfahrens liegt daher in $\mathcal{O}(|Frames| \cdot |Pixel|)$
Auch der von diesem Verfahren benötigte Speicherbedarf ist sehr gering. Da das Hintergrundmodell zu jedem Zeitpunkt aus einem Videobild der zu untersuchenden Überwachungssequenz besteht und dieses mit einem weiteren Bild der Sequenz verglichen wird, benötigt das Modell den selben Speicher wie zwei Überwachungsbilder. Der Speicherbedarf des Verfahrens liegt demnach in $\mathcal{O}(|Pixel|)$
Da das Verfahren keine Initialisierungsphase zur Berechnung eines anfänglichen

Hintergrundmodells benötigt, kann das Verfahren ab dem ersten Videobild der Überwachungsszene verwendet werden.
Neben den geringen Laufzeit- und Speicheranforderungen besitzt dieses Verfahren jedoch noch einige negative Eigenschaften, die dazu führen, dass dieses Verfahren in der Praxis nicht eingesetzt wird. Das Hintergrundmodell aus einem unbearbeiteten Videobild der Überwachungsszene besteht, beinhaltet das Modell neben Hintergrundobjekten auch alle Vordergrundobjekte. Durch das Verfahren kann letztlich nur ermittelt werden, an welchen Pixeln sich der Farbwert innerhalb der zwei Videobilder ausreichend stark geändert hat. Bewegt sich ein Objekt in der Szene, das aus einer oder mehreren homogenen, das heißt farblich relativ ähnlichen Regionen besteht, so wird das Verfahren innerhalb dieser Regionen keine deutliche Änderung erkennen können. Das hat zur Folge, dass sich bewegende Objekte nicht vollständig erkennen lassen.
Dieses Problem kann negative Konsequenzen mit sich bringen. Wenn sich Objekte nur teilweise erkennen lassen, bedeutet das nicht automatisch, dass die erkennbaren Regionen des Objekts zusammenhängend sind. Im Extremfall werden statt einem Objekt dann mehrere kleine Objekte detektiert und verfolgt, so dass unbrauchbare Trajektorien entstehen. Zudem kann es passieren, dass solche Regionen durch Nachbearbeitungsschritte (siehe Abschnitt 8) aus dem Subtraktionsbild entfernt werden. Dies geschieht beispielsweise wenn deren Pixelanzahl relativ klein ist, so dass die Region möglicherweise durch Rauschen verursacht wurde.
Wie gut das Verfahren letztlich bewegende Objekte erkennen kann, hängt im wesentlichen von deren Bewegungsgeschwindigkeit ab. Ist diese gering, so ist bei einer Vielzahl an Pixeln keine ausreichend große Farbänderung erkennbar. Je schneller es sich dagegen bewegt, desto mehr Farbwerte ändern sich deutlich und werden von dem Verfahren erkannt. Jedoch werden nicht nur Pixel eines Objekts im gerade zu untersuchenden Videobild erkannt, sondern auch die des vorherigen Bildes, da das Hintergrundmodell aus diesem besteht. Im Subtraktionsbild sind somit Pixel aus der neuen sowie aus der alten Position des Objekts erkennbar.
Der Schwellwert der für die Klassifikation der Pixel in Vorder- beziehungsweise Hintergrund herangezogen wird, muss wie in Abschnitt 6.3 sorgfältig gewählt werden.

7.2 Mittelwert Verfahren

Das Mittelwert Verfahren besteht im Wesentlichen aus zwei Phasen. Zuerst wird innerhalb einer Initialisierungsphase ein anfängliches Hintergrundmodell berechnet. Wie der Name des Verfahrens andeutet, werden hierfür pixelweise die farblichen Mittelwerte bezüglich einer gegebenen Videosequenz, einer sogenannten Trainigssequenz, berechnet. Daher werden alle in der Sequenz vorkommenden Farbwerte eines Pixels $Frame_t(i,j)$, also an der Position (i,j) zum Zeitpunkt t, addiert und das Ergebnis durch die Anzahl der hierfür verwendeten Videobilder geteilt. Daher ergibt sich der Mittelwert zu :

$$\frac{1}{|Frames|}\sum_{t=1}^{|Frames|} Frame_t(i,j) = \sum_{t=1}^{|Frames|} \frac{1}{|Frames|} Frame_t(i,j)$$

Das Hineinziehen des Faktors $\frac{1}{|Frames|}$ in die Summe bietet bei einer Implementierung

Algorithmus 7.2 Initialisierung Mittelwert Verfahren

procedure INITAVERAGE(*sequence*) returns Background
  for all $(FramesF_t)$ **do**

    // update each pixel value
    for all $(Pixels(x,y))$ **do**
      $Background(x,y) \leftarrow Background(x,y) + \frac{1}{numberOfFrames} \cdot F_t(x,y);$
    end for
  end for
  return *Background*
end procedure

Algorithmus 7.3 Subtraktion Mittelwert Verfahren

procedure SUBTRACTAVERAGE(*sequence*, *Background*, *thresh*) returns
  for all $(FramesF_t)$ **do**

    // compute subtraction image
    for all (**do**$Pixels(x,y)$)
      $distance \leftarrow dist(F_t(x,y), Background(x,y))$
      $$SubtractionImage(x,y) \leftarrow \begin{cases} 1 & \text{falls } distance \geq thresh \\ 0 & \text{otherwise} \end{cases}$$
    end for

    // store computed subtraction Image
    $save(SubtractionImage, t)$
  end for
end procedure

den Vorteil, dass die zwischenzeitlichen Werte nicht den durch den Farbraum der Bilder vorgegebenen Wertebereich überschreiten. Bei längeren Sequenzen können sich die Werte ansonsten bis über den durch Programmiersprachen üblichen Wertebereich aufaddieren.
In Algorithmus 7.2 ist der Pseudocode für diese Initialisierung zu sehen.

In der zweiten Phase werden die Subtraktionsbilder dadurch erzeugt, dass das berechnete Hintergrundmodell pixelweise mit den Überwachungsbildern verglichen und auf Ähnlichkeit untersucht werden. Eine Ähnlichkeit liegt genau dann vor, wenn sich der farbliche Unterschied unterhalb eines Schwellwertes befindet. Im Allgemeinen beginnt die hierfür verwendete Sequenz zeitlich direkt nach der Trainingssequenz. In Algorithmus 7.3 ist der hierzu wesentliche Abschnitt in Pseudocode aufgeführt.

Die hierfür verwendeten Variablen sind die gleichen wie in dem Algorithmus zur Initialisierung des Hintergrundmodells.
Zur Berechnung der Farbdistanz kann die in Abschnitt 4.11 vorgestellte euklidsche Distanz verwendet.
Gegenüber dem in Abschnitt 7.1 vorgestellten Differenzbild Verfahren kann dieses

Verfahren nicht gleich ab dem ersten Videobild einer Sequenz die Subtraktionsbilder berechnen, sondern erst nachdem es ein initiales Hintergrundmodell berechnet hat. Der Zeitbedarf hierfür hängt im Wesentlichen von der Anzahl der Videobilder einer hierfür bereitgestellten Trainigssequenz so, wie der Größe der Bilder ab. Der hierfür benötigte Zeitaufwand liegt in $\mathcal{O}(|Frames| \cdot |Pixel|)$. Der Zeitaufwand zur Berechnung eines Subtraktionsbildes liegt in $\mathcal{O}(|Pixel|)$.
Bei einer naiven Implementierung liegt der Speicherbedarf dieses Verfahrens in $\mathcal{O}(|Frames| \cdot |Pixel|)$, da bei einer solchen die komplette Sequenz in den Speicher geladen wird. Jedoch lässt sich das Verfahren auch so implementieren, dass die Speicheranforderungen deutlich reduziert werden. Dazu wird nicht die ganze Sequenz in den Speicher geladen, sondern ein Videobild nach dem anderen. Da so zu jedem Zeitpunkt nur ein Videobild sowie die Zwischenwerte der Mittelwertberechnung im Speicher sind, liegt dieser dann in $\mathcal{O}(|Pixel|)$. Zur Berechnung des Subtraktionsbildes wird nur ein Videobild sowie das berechnete Hintergrundmodell im Speicher benötigt. Der Speicherbedarf liegt hier daher ebenfalls in $\mathcal{O}(|Pixel|)$.
Die Qualität des berechneten initialen Hintergrundmodells hängt im Wesentlichen von der Länge der Trainigssequenz sowie der Menge der sich darin bewegenden Objekte ab. Jede einzelne Farbänderung eines Pixels wirkt sich letztlich auf den berechneten Mittelwert aus. Je länger ein eigentlicher Hintergrundpixel dabei von einem solchen überdeckt wird und je stärker er sich farblich von einem solchen Objekt unterscheidet, desto stärker wird das Hintergrundmodell an den entsprechenden Stellen beeinträchtigt. Abhängig von der Sequenz kann eine Verbesserung des Hintergrundmodells dadurch erhalten werden, dass die Trainingssequenz erhöht wird. Eine Verbesserung tritt genau dann ein, wenn dadurch die prozentuale Verdeckung des Hintergrunds durch Vordergrundobjekte abnimmt. Auch Bildrauschen in Abhängigkeit der Länge der Trainingssequenz, aus dem Hintergrundmodell entfernt werden.
Nach der Berechnung des initialen Hintergrundmodells, ändert sich dieses nicht mehr. Somit kann sich das Modell nicht an Veränderungen der Szene anpassen. Weshalb eine Anpassung an Szenenänderungen wichtig ist, wurde in Kapitel 6.4 bereits erläutert. Aufgrund der zuvor geschilderten Problematik, dass die Qualität des Hintergrundmodells stark von der Trainingssequenz abhängig ist, benötigt das Verfahren eine Adaptionsfähigkeit, da ansonsten keine Möglichkeit besteht, Fehler aus dem Hintergrundmodell zu entfernen. Das im nächsten Abschnitt vorgestellte *Running Average* Verfahren stellt im Prinzip eine Verbesserung des Mittelwertverfahrens dar, so dass Szenenänderungen in das Hintergrundmodell regelmäßig eingearbeitet werden können.

7.3 Running Average

Ein wesentliches Problem des Mittelwertverfahrens, das in Abschnitt 7.2 vorgestellt wurde, war die fehlende Möglichkeit sich an Szenenänderungen anpassen zu können. Das *Running Average* Verfahren bietet diese Möglichkeit. Aktualisiert wird das Modell nach der folgenden Gleichung :

$$Background_{t+1} = \alpha \cdot Frame_t + (1 - \alpha) \cdot Background_t$$

Dabei stellt die Variable α ein prozentuales Gewicht dar, mit dem ein aktuell bearbeitetes Videobild in das Hintergrundmodell eingearbeitet wird. Häufig wird α auch

Algorithmus 7.4 Running Average

procedure RUNNINGAVERAGE($sequence, thresh, \alpha$) returns
  for all ($FramesF_t$ **do**
    for all ($Pixels(x,y)$) **do**
      $distance \leftarrow dist(background(x,y), F_t(x,y))$
      $$SubtractionImage(x,y) \leftarrow \begin{cases} 1 & \text{falls } distance \geq thresh \\ 0 & \text{otherwise} \end{cases}$$
      // update the background model
      $Background(x,y) \leftarrow \alpha \cdot F_t(x,y) + (1-\alpha) \cdot background(x,y)$
    end for

    // store computed subtraction image at time t
    $save(SubtractionImage, t)$
  end for
end procedure

als Lernrate bezeichnet, da durch sie gesteuert werden kann, wie schnell Szenenänderungen in dem Hintergrundmodell aufgenommen werden, also wie schnell diese Änderungen gelernt werden können. Zusätzlich bieten adaptive Verfahren die Möglichkeit, Fehler beziehungsweise Ungenauigkeiten, die sich in ein Hintergrundmodell eingearbeitet haben, durch Aktualisierungsschritte herauszuarbeiten. Das Verfahren ist in Algorithmus 7.4 als Pseudocode zu sehen.
Das Subtraktionsbild wird wie schon bei den vorherigen Verfahren durch die pixelweise Ermittlung des Farbabstandes zwischen dem Hintergrundmodell und dem aktuell zu untersuchenden Videobild berechnet.
Gegenüber dem Mittelwert Verfahren wird hier keine Initialisierung durchgeführt. Jedoch kann eine Initialisierung beispielsweise durch das verwenden des angesprochenen Mittelwert Verfahrens erfolgen. Dies würde den Vorteil bieten, dass schon zu Beginn einer Überwachungssequenz mit einem Hintergrundmodell gearbeitet werden kann, in dem die Vordergrundobjekte weitestgehend oder zumindest so gut wie möglich herausgerechnet wurden. Ohne Initialisierung benötigt das Modell eine gewisse Zeit, abhängig von der Lernrate α, bis diese Objekte in diesem Modell nicht mehr vorhanden sind, beziehungsweise nur noch relativ geringen Einfluss auf das Modell haben. Zudem besteht die Möglichkeit, dass eine Initialisierungsphase des Verfahrens künstlich erzeugt wird, indem man es auf eine Trainingssequenz ansetzt, ohne dabei Subtraktionsbilder zu erzeugen.
Vorteilhaft an diesem Verfahren sind die niedrigen Laufzeitkosten sowie der sehr geringe Speicherbedarf der während der Laufzeit benötigt wird. Die Laufzeit liegt in $\mathcal{O}(|Frames| \cdot |Pixel|)$. Der Platzbedarf liegt in $\mathcal{O}(|Pixel|)$. Während der Laufzeit muss nur jeweils eines der Videobilder, das Hintergrundmodell sowie das berechnete Subtraktionsbild im Speicher gehalten werden.
Das *Running Average* Verfahren bietet zwar eine schnelle Möglichkeit, Vordergrundobjekte aus Überwachungsaufnahmen zu extrahieren, jedoch bereiten ihm einige der in Kapitel 6 angesprochenen Herausforderungen offensichtlich Probleme. Ein grundelegendes Problem, das auch das zuvor erläuterte Mittelwert Verfahren aufweist, stellt die Anfälligkeit des Hintergrundmodells gegenüber sich bewegenden Objekten dar. Da jeder Pixel mit Hilfe jedes zu analysierenden Videobildes aktualisiert wird,

besteht die Gefahr, dass sich Farbwerte von Vordergrundobjekten mit denen von Hintergrundobjekten vermischen, diese im Extremfall sogar ersetzen. Wie stark diese Vermischung das Modell beeinflusst, beziehungsweise nach wievielen Videobildern eine vollständige Ersetzung stattgefunden hat, hängt dabei von den Farbabständen der beteiligten Vorder- und Hintergrundobjekten sowie von der Lernrate α ab. Um dieses Problem abzuschwächen, wird häufig die in Abschnitt 4.7 vorgestellte Selektion eingesetzt. Dadurch wird die Aktualisierung des Farbwertes des Hintergrundmodells genau dann durchgeführt, wenn es sich bei diesem um einen Hintergrundpixel handelt, ansonsten wird der bisherige Farbwert weiter geführt. Dadurch wird eine Verfälschung des Modells durch sich bewegende Objekte vermieden. Ob es sich lohnt die durch die Selektion auftretende Probleme, zum Beispiel dass das Hintergrundmodell keine Möglichkeit mehr besitzt um Vordergrundobjekte aufzunehmen selbst wenn dies eigentlich gewollt werden sollte (beispielsweise durch parkende Autos auf einem Parkplatz) oder das Modell wesentlich länger benötigt um in ihm enthaltene Fehler heraus zu rechnen, in Kauf zu nehmen, muss sorgfältig entschieden werden.
Durch den Schwellwert *thresh* lässt sich der Farbabstand regulieren, der zwischen einem Pixel und dem entsprechenden Hintergrundwert mindestens vorliegen muss, damit der Pixel nicht als Hintergrund, sondern als Vordergrund klassifiziert wird. Dieser Wert ist für jeden Pixel gleich, hängt also damit nicht von der Position und dem Farbverlauf an dieser Position ab. Damit das Verfahren gute Ergebnisse erzielen kann, muss dieser Parameter gut gewählt werden, damit möglichst wenig Pixel falsch klassifiziert werden.
Wie es sich bei einigen der wesentlichen Herausforderungen aus Kapitel 6 bewährt, ist Teil der in Rahmen dieser Diplomarbeit durchrgeführten Evaluation und ist in Kapitel 9 aufgeführt.

7.4 Running Gaussian

Das *Running Gaussian* Verfahren ist ein weiteres unimodales *Background Subtraction* Verfahren. Dieses Modell basiert auf den in Abschnitt 4.12.4 vorgestellten Gaußfunktionen. Das Graph einer Gaußfunktion wird durch die Parameter μ und σ festgelegt. Dabei bestimmt der Parameter μ die x-Koordinate des Hochpunktes der Funktion, σ dagegen die y-Koordinate sowie die Breite der Funktion. In der Stochastik werden die Gaußfunktionen häufig zur Beschreibung von Zufallsprozessen eingesetzt. Diese Prozesse werden hierbei durch einen Mittelwert, der angibt mit welchem Durchschnittswert man bei häufigem Wiederholen rechnen kann, sowie die sogenannte Varianz, die die Streuung um diesen Mittelwert widerspiegelt, beschrieben. Somit entspricht der Mittelwert dem Parameter μ der Gaußfunktion, die Varianz kann dagegen durch σ^2 ausgedrückt werden.
Jeder Pixel wird bei diesem Verfahren somit durch eine Gaußfunktion, also durch Angabe seines Mittelwertes μ und seiner Varianz beziehungsweise seiner Standardabweichung σ modelliert. Für eine Trainigsmenge, also einer Teilsequenz einer Überwachungsszene ergibt sich die Berechnung dieser Parameter für einen Pixel an der Stelle (x, y) für ein initiales Hintergrundmodell durch die folgenden Gleichungen

$$\mu = \frac{1}{t} \cdot \sum_{i=1}^{t} P_i(x, y)$$

$$\sigma = \frac{1}{t}\sum_{i=1}^{t}(P_i(x,y) - \mu)^2$$

$P_i(x,y)$ gibt hierbei den Farbwert des Pixels an der Stelle (x,y) zum Zeitpunkt i an.
Um das Hintergrundmodell an eventuelle Szenenänderungen anpassen zu können (siehe hierzu Abschnitt 6.4) wird das Modell pixelweise nach jeder Bearbeitung eines Videobildes aktualisiert. Diese Aktualisierungen werden durch die folgenden Gleichungen realisiert :

$$\mu_{t+1} = \alpha \cdot Frame_t + (1-\alpha) \cdot \mu_t$$

$$\sigma^2_{t+1} = \alpha \cdot (Frame_t - \mu_t)^2 + (1-\alpha) \cdot \sigma^2_t$$

Wie aus den Aktualisierungsgleichungen ersichtlich wird, ist schon nach der ersten Aktualisierung nicht mehr gewährleistet, dass das Hintergrundmodell den tatsächlichen Mittelwert oder Varianz besitzt. Dies liegt daran, dass der Parameter α das den aktuellen Farbwert gegenüber dem des Hintergrundmodells deutlich stärker gewichtet und nicht gleichmäßig, wie es für eine exakte Berechnung des Mittelwertes notwendig wäre. Da das Hintergrundmodell möglichst aktuell gehalten werden sollte, ist eine stärkere Gewichtung neuerer Videobilder durchaus sinnvoll. Wie schon bei dem *Running Average* Verfahren lässt sich durch den Parameter α steuern, wie schnell sich eventuelle Szenenänderungen in das Modell integrieren lassen.
Häufig wird auf die Berechnung eines initialen Hintergrundmodells verzichtet. Dann wird anstelle des Mittelwerts der Farbwert des ersten Videobildes der Überwachungsszene verwendet. Die Varianz wird dagegen auf einen beliebigen Anfangswert gesetzt. Sind die so gewählten Anfangswerte für die Szene nicht besonders gut, ergeben sich nach einigen Aktualisierungsschritten durch obige Gleichungen bessere.
Durch die Verwendung von Gaußfunktionen zur Modellierung des Hintergrunds entfällt die Klassifikation der Pixel mit Hilfe eines Schwellwertes wie beispielsweise bei dem *Running Average* Verfahren. Vorteilhaft ist dabei, dass die Klassifikation nicht für alle Pixel mit dem selben Wert durchgeführt werden muss, sondern mit Hilfe der Gaußfunktion durchgeführt werden kann, die sich für jeden Pixel aufgrund seines Farbverlaufes innerhalb der Videosequenz ergibt. Damit ist diese im wesentlichen von der Vergangenheit der Pixelwerte abhängig und damit deutlich flexibler. Die Sicherheit mit der ein Pixel dem Hintergrund beziehungsweise dem Vordergrund zugeordnet wird, hängt von der Farbdistanz zwischen ihm und dem Hintergrundwert an der entsprechenden Stelle, sowie von seiner Varianz ab. Nach Abbildung 4.5 liegen beispielsweise 98.8 Prozent der Fläche zwischen der Funktion und der x-Achse innerhalb des Intervalls $[\mu - 2 \cdot \sigma, \mu + 2 \cdot \sigma]$. Daher konzentriert sich der wesentliche Teil der Gesamtfläche um den Mittelwert mu. Ein Pixel wird als Hintergrundpixel eingestuft, falls sein Farbwert innerhalb des Intervalls $[\mu - k \cdot \sigma, \mu + k \cdot \sigma]$ liegt. Ansonsten ist die Distanz zu groß, eine Ähnlichkeit mit dem Hintergrund ist daher unwahrscheinlich.
In Algorithmus 7.5 ist das *Running Gaussian* Verfahren ohne spezielle Initialisierungsphase in Pseudocode aufgeführt.
Auch dieses unimodale *Background Subtraction* kann durch seine geringen Laufzeiten sowie Speicheranforderungen überzeugen. Die zur Berechnung der Subtraktionsbilder benötigte Zeit liegt dabei in $\mathcal{O}(|Pixel|)$, da für jeden Pixel eine Distanzberechnung sowie einen Vergleich durchgeführt werden muss. Auch die Aktualisierung dieses Modells ist mit dieser Schnelligkeit durchführbar, da hier pro Pixel zwei Zuordnungen erfolgen müssen. Für eine komplette Videosequenz liegt die Laufzeit somit in

Algorithmus 7.5 Running Gaussian

procedure RUNNING GAUSSIAN($sequence, \alpha, k$)
initModel();

// do subtraction

for all ($FramesF_t$) **do**
for all ($Pixels(x,y)$) **do**
$distance \leftarrow dist(Background(x,y), F_t(x,y))$
$$SubtractionImage(x,y) \leftarrow \begin{cases} 1 & \text{falls } distance \geq k \cdot \sigma(x,y) \\ 0 & \text{otherwise} \end{cases}$$
// update background model
$Background(x,y) \leftarrow \alpha \cdot F_t(x,y) + (1-\alpha) \cdot Background(x,y)$
$\sigma(x,y) \leftarrow \sqrt{\alpha \cdot (F_t(x,y) - Background(x,y))^2 + (1-\alpha) \cdot \sigma(x,y)^2}$
end for
// store computed subtraction Image
$save(SubtractionImage, t)$
end for
end procedure

$\mathcal{O}(|Frames| \cdot |Pixel|)$. Während der kompletten Laufzeit müssen die Mittelwerte, Varianzen sowie das gerade berechnete Subtraktionsbild im Speicher gehalten werden. Daher liegt der Speicherbedarf des *Running Average* Verfahrens in $\mathcal{O}(|Pixel|)$.

7.5 Median Verfahren

Das Median Verfahren berechnet für sein Hintergrundmodell, den Median der Farbwerte die ein Pixel innerhalb einer Trainingssequenz, also einer Teilsequenz des Überwachungsvideos, annimmt. Der Median berechnet sich dadurch, dass pixelweise die Farbwerte der Trainingssequenz komponentenweise sortiert werden und für jede Komponente das Element in der Mitte gewählt wird. Als Komponenten werden dabei die Farbwerte des zugrunde liegenden Farbsystems verstanden. Im Allgemeinen wird das in Abschnitt 4.8 vorgestellte RGB-Farbsystem verwendet. Daher werden die Rot-, Grün- und Blaukanäle in diesem Zusammenhang als Komponenten bezeichnet.
Mit $r_i(x,y)$ sei im Folgenden die Folge der Farbwerte der roten Farbkomponente, also $r_1(x,y), r_2(x,y,...,r_t(x,y))$, des Pixels an der Stelle (x,y) innerhalb der Trainingssequenz gemeint. Die Folge $r_{sorted,i}(x,y)$ sei diejenige Folge, die durch sortieren der Werte von $r_i(x,y)$ entsteht. Analog hierzu seien die Folgen $g_i(x,y)$, $b_i(x,y)$ sowie $g_{sorted,i}$ und $b_{sorted,i}$ definiert. Der Median ist als mittleres Element der sortierten Folgen definiert. Dieses befindet sich an der Stelle $\frac{t-1}{2}$. Demnach berechnet sich der Median zu :

$$M = (r_{Median}(x,y), g_{Median}(x,y), b_{Median}(x,y))$$

Besitzen die Folgen eine gerade Anzahl an Elementen so ist $t-1$ eine ungerade Zahl und entsprechend $\frac{t-1}{2}$ keine natürliche Zahl. In diesem Fall kann der Bruch einfach

ab- oder aufgerundet werden.
Um geeignet auf Änderungen innerhalb der Überwachungsszene reagieren zu können (siehe herzu Kapitel 6.4), muss das Hintergrundmodell des *Median* Verfahrens regelmäßig aktualisiert werden. Jedoch ist die Berechnung des Medians relativ aufwändig, da hierfür drei Zahlenfolgen sortiert werden müssen. Daher wird eine Aktualisierung benötigt, die deutlich schneller durchgeführt werden kann. McFarlan et al. [MS95] schlagen folgende Aktualisierung des Modells vor :

$$r_{Median}(x,y) \leftarrow \begin{cases} r_{Median}(x,y)+1 & \text{falls } r_{Median} \leq P_r(x,y) \\ r_{Median}(x,y) & \text{falls } r_{Median} = P_r(x,y) \\ r_{Median}(x,y)-1 & \text{falls } r_{Median} \geq P_r(x,y) \end{cases}$$

Dabei bezeichnet $P_r(x,y)$ den Wert der roten Farbkomponente des Pixels an der Stelle (x,y). Demnach wird keine Aktualisierung des Medians durchgeführt, falls er genau dem Wert des aktuell zu untersuchenden Pixels entspricht. Ist der zu untersuchende Pixel größer als der Wert des Hintergrundmodells, so wird der Hintergrundwert um eins erhöht, ansonsten um eins reduziert. Damit passt sich das Modell den Veränderungen innerhalb der Szene an. Die Geschwindigkeit in der diese Anpassung erfolgt hängt von dem Farbabstand zwischen dem aktuellen Pixel und dem Wert des Hintergrundmodells an dieser Stelle ab. Die Aktualisierung der Werte für die anderen zwei Farbkomponenten erfolgt analog.
Der Pseudocode des Median Verfahrens ist in Algorithmus 7.6 aufgeführt. Als Trainingssequenz werden hierfür die ersten t_{train} Videobilder der Überwachungssequenz verwendet.

Die Initialisierungsphase des *Median* Verfahrens ist besonders zeitintensiv, da hierfür drei Farbfolgen sortiert werden müssen. Häufig in der Praxis eingesetzte Sortieralgorithmen sortieren diese Farbfolgen mit einer Geschwindikeit, die in $\mathcal{O}(t \cdot log(t))$ liegt, wobei t angibt, aus wievielen Videobildern die gewählte Trainingssequenz besteht. Das initiale Hintergrundmodell lässt sich somit in $\mathcal{O}(t \cdot log(t) \cdot |Pixel|)$ berechnen.
Der Speicherbedarf des Verfahrens ist für die Initialisierungsphase ebenfalls recht hoch, da die komplette Trainingssequenz im Speicher gehalten werden muss. Gegenüber der Initialisierungsphase des Mittelwert Verfahrens (dieses ist in Abschnitt 7.2 aufgeführt) besteht hier leider nicht die Möglichkeit den recht hohen Speicherbedarf durch eine geschickte Implementierung zu reduzieren, da für die Medianberechnung die komplette Farbfolge der einzelnen Kanäle benötigt wird. Ein System welches das *Median* Verfahren als *Background Subtraction* Verfahren verwendet, muss dies beachten, da in Abhängigkeit der Größe sowie Kompressionsrate der Videobilder sowie des verfügbaren Speichers, keine beliebig große Trainingsphase zur Berechnung eines initialen Hintergrundmodells verwendet werden kann.
Bei Betrachtung des benötigten Rechen- und Speicherbedarfs dieses Verfahrens stellt sich die Frage, inwiefern sich dieser hohe Aufwand lohnt, also wo genau sich Vorteile ergeben. Zur Beantwortung dieser Frage sind in Abbildung 6.2 die Ergebnisse der Initialisierungsphasen des Mittelwert Verfahrens sowie des *Median* Verfahrens aufgeführt. Für beide Verfahren wurde die selbe Trainingsmenge verwendet, die sich bewegende Vordergrundobjekte beinhalteten. Demnach war der Hintergrund nicht über die komplette Sequenz sichtbar, sondern wurde zeitweise verdeckt. Bei der Betrachtung der zwei Ergebnisse wird deutlich, dass das *Median* Verfahren die Vordergrundobjekte weniger stark in das Modell einarbeitet. Das Mittelwert Verfahren

Algorithmus 7.6 Median Verfahren

procedure MEDIAN(*sequence*,*thresh*,$t_{training}$)

// init model with first $t_{training}$ frames of the sequence
for all ($TrainingFramesF_t$) **do**
for all ($Pixels$) **do**
$ComputeMedian(RedColorValues)$
$ComputeMedian(GreenColorValues)$
$ComputeMedian(BlueColorValues)$
end for
end for
for all ($Pixels(x,y)$) **do**
$background(r,c).setRedComponent(median(values(r,c,:,1)))$
$background(r,c).setGreenComponent(median(values(r,c,:,2)))$
$background(r,c).setBlueComponent(median(values(r,c,:,3)))$
end for

// compute subtraction image and update model
for all ($FramesF_t$) **do**
for all ($Pixels(x,y)$) **do**
$distance \leftarrow dist(F_t(x,y), Background(x,y))$

$$SubtractionImage(x,y) \leftarrow \begin{cases} 1 & \text{falls } distance \geq thresh \\ 0 & \text{otherwise} \end{cases}$$

// update background model

$$Background(x,y) \leftarrow \begin{cases} Background(x,y)+1 & \text{falls } Background(x,y) \leq F_t(x,y) \\ Background(x,y) & \text{falls } Background(x,y) = F_t(x,y) \\ Background(x,y)-1 & \text{falls } Background(x,y) \geq F_t(x,y) \end{cases}$$

end for

store computed subtraction image at time t

$saveImage(SubtractionImage, t)$
end for
end procedure

weist jedoch an einigen Stellen deutliche Fehler auf. Der Grund hierfür liegt in der Tatsache, dass die Vordergrundobjekte das Hintergrundmodell des Mittelwert Verfahren schon ab ihrem ersten Vorkommen beeinflussen und damit den eigentlichen Farbwert des Hintergrundmodells verfälschen. Daher ist das Resultat der Initialisierungsphase bei diesem Verfahren stark von der Länge der Trainingssequenz sowie der Anzahl der Vordergrundobjekte sowie deren Bewegungsgeschwindigkeit abhängig. Bei dem *Median* Verfahren ist der Einfluss eines Vordergrundobjektes weniger stark ausgeprägt. Aus der Berechnungsweise des Medians wird offensichtlich, dass ein Pixel bei mindestens 50 Prozent der Videobilder der Trainingsmenge von einem Vordergrundobjekt verdeckt sein muss, bevor das Hintergrundmodell an entsprechender Stelle einen Fehler aufweist. Es ist damit weniger sensibel gegenüber Ausreisern.

Die Qualität der erzeugten Subtraktionsbilder hängt bei diesem Verfahren maßgeblich von seinem einzigen Parameter, dem Schwellwert *thresh*, ab. Mit Hilfe dieses Wertes, der für jeden Pixel für alle Videobilder konstant ist, wird die Klassifikation in Hintergrund beziehungsweise Vordergrund durchgeführt. Um gute Ergebnisse zu erzielen, muss demnach dieser Parameter gut gewählt werden. Die Problematik die durch nicht optimale Wahl der Parameter eines *Background Subtraction* Verfahrens entsteht, ist in Abschnitt 6.3 aufgeführt.

7.6 McKenna

Dieser Abschnitt widmet sich dem von McKenna et al. vorgestellten *Background Subtraction* Verfahren [MJD+00]. Hierbei handelt es sich um ein Verfahren, welches das Prinzip des *Running Gaussian* Verfahrens (dieses ist in Abschnitt 7.4 aufgeführt) auf Chromazitäts- sowie Kantenwerte der einzelnen Pixel, anwendet.
Das Hintergrundmodell beinhaltet zunächst pro Pixel einen mittleren Farbwert μ sowie einen Varianzwert σ der angibt, wie sehr die Farbwerte der Pixel über die Zeit um den berechneten Mittelwert μ streuen. Eine solche Streuung kann beispielsweise aufgrund von Rauschen, quasi-periodischen Wiederholungen oder durch sich bewegende Vordergrundobjekte, die den Hintergrund verdecken, entstehen. Natürlich sind dabei Verfälschungen des Mittel- sowie des Varianzwertes durch Vordergrund nicht gewollt. Daher werden regelmäßig Aktualisierungen dieser Werte durchgeführt, um eventuelle Verfälschungen aus dem Hintergrundmodell herausrechnen zu können.
Die verwendeten Chromazitätswerte bieten dem Verfahren gegenüber dem Verwenden von RGB-Werten laut Autoren eine größere Robustheit gegenüber Schatten. Während die Intensitätswerte durch auftretende Schatten in der Szene bei RGB-Bildern stark abnehmen, bleiben sie dagegen bei Chromazitätsbildern oft nahezu unverändert. Wie in Abschnitt 6.8 erläutert, führen die Schatten bei Trackingverfahren, die auf den durch die *Background Subtraction* Verfahren berechneten Subtraktionsbildern arbeiten, zu einigen Problemen.
Die Chromazitätswerte werden im RGB-Farbraum wie folgt berechnet:

$$r_c = \frac{R}{R+G+B} \text{ sowie } g_c = \frac{G}{R+G+B}$$

Pixelweise werden die Chromazitäten durch Gaußfunktionen modelliert. Dazu werden die Mittelwerte μ_{r_c} und μ_{g_c} sowie die dazugehörigen Varianzen $\sigma^2_{r_c}$ und $\sigma^2_{r_c}$ berechnet. Für die Aktualisierung schlagen die Autoren die folgenden Gleichungen vor :

$$\mu_{t+1} = \alpha \cdot Frame_{t+1} + (1-\alpha) \cdot \mu_t$$

$$\sigma^2_{t+1} = (1-\alpha) \cdot (\sigma^2_t + (\mu_{t+1} - \mu_t)^2) + \alpha \cdot (Frame_{t+1} - \mu_{t+1})^2$$

Die Parameter μ sowie σ verändern sich im Allgemeinen über die Zeit. Sie stellen auch keine exakten Mittel- beziehungsweise Varianzwerte über die Zeit dar, da später auftretende Pixelwerte einen höheren Beitrag zu deren Berechnung leisten. Dieser Umstand ist gewollt, da sich nur so das Hintergrundmodell rasch an Änderungen in der Szene anpassen kann. Durch den Parameter α lässt sich regulieren, wie schnell die angesprochenen Szenenänderungen in das Hintergrundmodell aufgenommen werden.
Zwar sind die verwendeten Chromazitätswerte weniger anfällig gegenüber Schatten, dafür haben sie an anderer Stelle ihre Schwächen. So treten beispielsweise Probleme

auf, wenn eine Person, die eine schwarze Hose trägt auf einer grauen, asphaltierten Straße läuft, da sich hier die beteiligten Chromazitätswerte kaum unterscheiden. Dies führt in der Regel dazu, dass die durch Tarnung hervortretenden Probleme(siehe hierzu Abschnitt 6.7) stark zunehmen. Um dieses Problem ausgleichen zu können, wird das Verfahren erweitert, in dem zu den Chromazitätwerten zusätzlich Kantenwerte bezüglich der verwendeten Farbkomponenten in x-Richtung und in y-Richtung, berechnet werden. Auch diese werden durch Gaußfunktionen modelliert. Durch die oben aufgeführten Gleichungen werden daher noch die Parameter der Mittelwerte μ_{x_r}, μ_{y_r}, μ_{x_g}, μ_{y_g}, μ_{x_b} und μ_{y_b} sowie die Varianzen der Kantenstärken $\sigma^2_{g_r}$, $\sigma^2_{g_g}$ und $\sigma^2_{g_b}$ berechnet.
Der Ähnlichkeitstest für die Chromazitätskomponente gilt als bestanden, wenn die folgende Ungleichung erfüllt werden kann:

$$|chrom - \mu_{chrom}| \leq k \cdot \sigma_{chrom} \text{ für } chrom \in \{r_c, g_c\}$$

Der Test für die Kantenwerte gilt als bestanden, wenn die folgende Ungleichung für mindestens eine beteiligte Komponente erfüllt werden kann :

$$\sqrt{(color_x - \mu_{color_x})^2 + (color_y - \mu_{color_y})^2} \leq k \cdot \sigma_{g_{color}}$$

Für k kann im Prinzip ein beliebiger Wert verwendet werden. Da jedoch das Endresultat, also das zu berechnende Subtraktionsbild durch den Parameter direkt beeinflusst wird, sollte dieser möglichst gut gewählt werden. Die Autoren schlagen einen Wert von $k = 3$ vor. In Kapitel 9 wird unter anderem vorgestellt, wie gut dieser und andere mögliche Werte bei der im Rahmen dieser Diplomarbeit durchgeführten Evaluation auch gegenüber anderen Verfahren beziehungsweise Ansätzen, abschneidet. Ein Pixel wird als Vordergrund klassifiziert, falls für mindestens der Chromazitätstest oder der Kantentest für mindestens eine Komponente positiv ausfällt.
Das Hintergrundmodell führt pro Pixel genau 10 Parameter, die pro Videobild für die angesprochenen Test verwendet und anschließend aktualisiert werden. Durch die Angabe des von diesem Verfahren benötigten Laufzeit- und Speicherbedarfs in O-Notation wird diese Konstante unterdrückt. Die Laufzeit liegt in $\mathcal{O}(|Frames| \cdot |Pixel|)$ der Platzbedarf dagegen in $\mathcal{O}(|Pixel|)$.

7.7 Jabri

Das von S. Jabri et al. entwickelte *Background Subtraction* Verfahren [JDWRoo] arbeitet ähnlich wie das in Abschnitt 7.4 vorgestellte *Running Gaussian* Verfahren, ergänzt dieses jedoch um weitere Komponenten. Das Hintergrundmodell besteht pro Pixel aus einem Farbwert sowie je einem Wert für die Stärke horizontaler sowie vertikaler Kanten. Somit setzt sich das Hintergrundmodell aus einem Modell für die Farbe sowie aus einem Modell das die Kantenstärke widerspiegelt, zusammen.
Das Farbmodell besteht pixelweise aus einem Mittelwert sowie der Varianz die angibt, wie stark die Farbwerte der bisherigen Analyse um diesen Mittelwert verstreut lagen. Der Mittelwertbegriff ist dabei nicht wörtlich zu nehmen, da bei dessen Bestimmung nicht jeder Pixel für dessen Bestimmung gleichen Anteil beiträgt. Sie μ_t der Mittelwert eines beliebigen Pixels zum Zeitpunkt t, dann kann der Mittelwert durch die Gleichung $\mu_{t+1} = \alpha \cdot x_{t+1} + (1 - \alpha) \cdot \mu_t$ mit Hilfe des Farbwertes x zum Zeitpunkt

Algorithmus 7.7 Verfahren von McKenna

procedure McKenna(*sequence*)
  $InitBackground()$ as gaussians for color and gradients
  for all $FramesF_t$ **do**
    for all $Pixels(x, y) \in (R, G, B)$ **do**

      //compute chromacity values
      $r_c \leftarrow \frac{R}{R+G+B}$
      $g_c \leftarrow \frac{G}{R+G+B}$
      // compute gradients with sobel operators for each color component
      $r_x \leftarrow Sobel_x(Frame_R)$
      $r_y \leftarrow Sobel_y(Frame_R)$
      $g_x \leftarrow Sobel_x(Frame_G)$
      $g_y \leftarrow Sobel_y(Frame_G)$
      $b_x \leftarrow Sobel_x(Frame_B)$
      $b_y \leftarrow Sobel_y(Frame_B)$

      // check similarity as discussed at the end of chapter 7.6

      // check $|chrom - \mu_{chrom}| \leq k \cdot \sigma_{chrom}$ for $chrom \in \{r_c, g_c\}$

      // check $\sqrt{(color_x - \mu_{color_x})^2 + (color_y - \mu_{color_y})^2} \leq k \cdot \sigma_{g_{color}}$
      if (Pixel is similar to chromacity or graidients) **then**
        $Pixel(x, y) \in Foreground$
      else
        $Pixel(x, y) \in Background$
      end if
    end for
  end for
end procedure

$t + 1$ aktualisiert werden. Der Parameter α ist die sogenannte Lernrate, durch die beeinflusst werden kann, wie stark ein neuer Farbwert in das Hintergrundmodell einfließt. Daher haben zeitlich spätere Farbwerte einen größeren Einfluss auf das Modell, wodurch das Verfahren in der Lage ist, sich an Änderungen in der Szene anpassen zu können. Zur Berechnung eines initialen Mittelwertes kann beispielsweise das in Abschnitt 7.2 vorgestellte Mittelwertverfahren verwendet werden. Durch die aufgeführte Aktualisierung kann jedoch darauf verzichtet werden, da sich durch sie das initiale Hintergrundmodell schnell verändert und an die aktuellen Gegebenheiten der Szene anpasst. Dafür wird in der Regel der Farbwert des ersten Überwachungsbildes verwendet.
Die Varianz σ_t wird durch die Gleichung $\sigma^2_{t+1} = \alpha \cdot (x_{t+1} - \mu_{t+1})^2 + (1 - \alpha) \cdot \sigma^2_{t-1}$ aktualisiert. Als initialer Wert kann im Prinzip ein beliebiger verwendet werden, da sich dieser durch die Aktualisierung schnell anpasst.
Das Kantenmodell wird durch das Anwenden eines horizontalen sowie eines vertikalen Sobeloperators auf jede Farbkomponente bestimmt. Die drei horizontalen Kantenmodelle werden zu einem horizontalen Kantenbild H kombiniert. Analog

hierzu kombiniert man die drei vertikalen Kantenbilder zu einem Bild V. Nun lassen sich ein Mittelwert und die Varianz in gleicher Weise wie im Farbmodell bestimmen und aktualisieren.
In der Subtraktionsphase werden die Subtraktionsbilder berechnet. Dazu werden die Pixel eines Videobildes mit den gespeicherten Werten auf Ähnlichkeit überprüft und entsprechend in Vordergrund- oder Hintergrundpixel aufgeteilt. Zusätzlich berechnet dieses Verfahren noch einen Wert der angibt, wie sicher es sich ist , dass es sich bei einem Pixel um einen Vordergrundpixel handelt. Dieser Wert wird im folgenden *Confidence Score* genannt. Für jeden Farbkanal c wird nun ein solcher *Confidence Score* C^c nach folgender Gleichung berechnet:

$$C^c = \frac{\mid (x_c \mu_c) \mid - m_c \cdot \sigma}{M_c \cdot \sigma - m_c \cdot \sigma} \cdot 100$$

Zur Berechnung werden die zwei Schwellwerte m_c sowie M_c mit $m_c \leq M_c$ verwendet. Laut Autoren haben sich die Werte $m_c = 15$ und $M_c = 25$ bewährt. Ist $|x_c - \mu_c| \leq m \cdot \sigma$ so würde der Zähler des angegeben Quotienten negativ werden. Da durch den *Confidence Score* jedoch beschrieben werden soll, mit welcher Sicherheit das Verfahren den Pixel als einen Vordergrundpixel klassifiziert, sind Werte zwischen 0 und 100 Prozent besser geeignet. Aus diesem Grund wird der negative Wert auf 0 Prozent gesetzt. Ein Wert von 0 Prozent gibt letztlich an, dass das Verfahren überzeugt ist, dass es sich bei dem Pixel um einen Vordergrundpixel handelt, da es keine Ähnlichkeit zu dem Hintergrundmodell feststellen konnte. Gilt dagegen $x_c - \mu_c \geq M_c \cdot \sigma$ so wäre der Zähler größer als der Nenner und C^c würde einen Wert liefern, der über 100 Prozent liegt. Aus Zweckmäßigkeit wird in einem solchen Fall C^c auf 100 Prozent gesetzt. Für alle anderen Fälle gilt $0 \leq C^c \leq 100$.
Auch bezüglich den Kanten wird komponentenweise ein *Confidence Score* C^e berechnet, der angibt, wie sicher das Verfahren eine Kante als Vordergrundkante klassifizieren würde. Seien H und V das horizontale beziehungsweise vertikale Kantenbild einer Farbkomponente, H_t sowie V_t die zum Zeitpunkt t gültigen Mittelwerte. Dann werden zunächst pixelweise die vorliegenden Kantenunterschiede wie folgt bestimmt :

$$\Delta H = |H - H_t| \text{ und } \Delta V = |V - V_t|$$

Der vorliegende Kantengradient berechnet sich dann zu $\Delta G = \Delta H + \Delta V$. Damit die Änderung der Kantenstärke in zwei zeitlich direkt aufeinanderfolgenden unabhängig von der vorliegenden Kantenstärke ist, wird zunächst noch folgender Zuverlässigkeitswert (engl. reliabilty) R berechnet (eine Änderung der Kantenstärke um den Wert 10 würde bei einer vorliegenden Stärke von 50 genau 20 Prozent entsprechen, bei einer vorliegenden Stärke von 100 dagegen nur 10 Prozent). Es gilt : $R = \frac{\Delta G}{G_t^*}$, wobei $G = |H| + |V|$, $G_t = |H_t| + |V_t|$ und $G_t^* = max\{G, G_t\}$.
Der kantenbasierte $ConfidenceScore$ berechnet sich jetzt zu $C^e = \frac{R \cdot \Delta G - m_e \sigma}{M_e \cdot \sigma - m_e \cdot \sigma} \cdot 100$. Für die verwendeten Schwellwerte haben sich laut Autoren die folgenden Werte bewährt : $m_e = 3$, $M_e = 9$ Auch hier müssen letztlich zwei Fälle beachtet werden, damit sich der berechnete Prozentwert zwischen 0 und 100 befindet. Ist $R \cdot \Delta G \leq m_e \sigma$ so wird der Prozentwert auf 0 gesetzt. Ist dagegen $R \cdot \Delta G \geq M_e \cdot \sigma$, so wird der Prozentwert auf 100 gesetzt.
Zur entgültigen Klassifikation eines Pixels in Vorder- oder Hintergrund, werden die berechneten $ConfidenceScores$ C^c und C^e verwendet.
Der *Score* berechnet sich zu :

$$C = max(max(C^{c,r}, C^{c,g}, C^{c,b}), max(C^{e,r}, C^{e,g}, C^{e,b}))$$

Dabei stehen r, g, b für die Farbkomponenten der verwendeten RGB-Farbbilder. Wird als Subtraktionsbild ein Binärbild benötigt, so kann dieses durch einen Vergleich des $ConfidenceScores$ mit einem Schwellwert der bei mindestens 50 Prozent liegt, erfolgen. Das hier vorgestellte Verfahren ist ein unimodales *Background Subtraction* Verfahren, da es für jeden Pixel jeweils genau ein Farb- sowie Kantenmodell im Speicher hält. Mit sich quasi-wiederholenden Hintergrundbewegungen wird es daher sicherlich Probleme bekommen, weil diese sich nicht in das Modell integrieren lassen. Die Laufzeit ist dadurch jedoch ziemlich gering und kann in O-Notation durch $\mathcal{O}(|Frames| \cdot |Pixels|)$ angegeben werden. Die unterdrückten Konstanten sind jedoch deutlich größer als bei anderen Verfahren unimodalen Verfahren. Der Speicherbedarf liegt in $\mathcal{O}(|Pixel|)$, wobei auch hier im Vergleich zu anderen Verfahren die unterdrückte Konstante höher ist. Wie das Verfahren bei den in Kapitel 6 vorgestellten Herausforderungen abschneidet, kann bei den Evaluationen in Kapitel 9 nachgelesen werden.

7.8 Mixture of Gaussians

Dieser Abschnitt widmet sich dem von Staufer und Grimson entwickelten *Background Subtraction* Verfahren [SG99], das auf einer Mischung (engl. mixture) von Gaußfunktionen (die Gaußfunktionen wurden in Abschnitt 4.12.4 vorgestellt) beruht. Hierbei handelt es sich um eines der bekanntesten, meist zitierten und in der Praxis am häufigsten eingesetzten Verfahren. Aufgrund seines Bekanntheitsgrades eignet es sich daher besonders gut für die im Rahmen dieser Diplomarbeit durchgeführten Evaluation, da sich im Prinzip alle anderen in der Praxis eingesetzten Verfahren sich mit diesem Messen müssen. Diese Evaluation befindet sich in Kaptiel 9.
Die Arbeitsweise des Verfahrens ist in Algorithmus 7.8 in Pseudocode aufgeführt.

7.8.1 Mischung von Gaußfunktionen

In Abschnitt 7.4 wurde das *Running Gaussian*Verfahren vorgestellt. Es modelliert jeden Pixel durch eine Gaußfunktion, also durch Angabe eines Mittelwertes sowie der zu diesem Wert gehörenden Varianz, die die Streuung der Farbwerte um diesen Mittelwert angibt. Ist die Farbverteilung eines Pixels während einer Videosequenz der einer Gaußfunktion ähnlich, so kann der Hintergrundmodell problemlos durch eine solche Funktion modelliert werden. In Abbildung 6.9ist eine Verteilung zu sehen, die sich nicht gut durch eine Gaußfunktion annähern lässt, weil sie mehrere starke Ausprägungen besitzt. Solche Verteilungen kommen besonders häufig vor, wenn der Hintergrund nicht statisch ist, wie beispielsweise in Szenen, die sich im Wind bewegende Bäume, Sträucher, Fahnen oder ähnliches beinhalten.
Lässt sich der Hintergrund nicht ausreichend gut durch eine Gaußfunktion modellieren, kann eine Verbesserung der Modellierung durch das Verwenden mehrerer kombinierter Gaußfunktionen erfolgen. Diese Kombination wird in der Regel *Mischung* oder *Mixture* genannt und besteht aus einer einfachen , eventuell gewichteten Addition der beteiligten Funktionen. In Abbildung 7.1 ist hierzu ein Beispiel zu sehen, in dem drei Funktionen durch eine Addition kombiniert wurden

Durch die Kombination von mehr als zwei Gaußfunktionen, lassen sich auch komplexere Verteilungen modellieren. Häufig werden zur Beschreibung der Farbverteilung

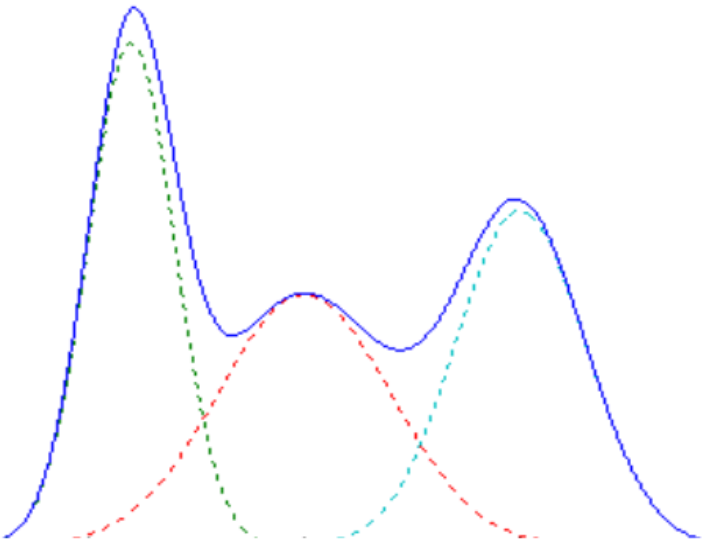

Abbildung 7.1: In dieser Abbildung ist die Mischung von drei Gaußfunktionen mittels einer Addition zu sehen. Durch solche Mischungen lassen sich Hintergründe modellieren, die mehrere starke Ausprägungen ihres Farbverlaufes besitzen.

eines Pixels bezüglich einer Videosequenz K Verteilungen verwendet, wobei in der Regel $k \in \{3,5\}$ gilt. Die Mischung noch größerer Anzahlen wäre möglich, jedoch hängt die benötigte Laufzeit von der Anzahl eingesetzter Gaußfunktionen ab und würde sich entsprechend erhöhen. Daher muss ein vernünftiges Verhältnis zwischen der Laufzeit und der Qualität der Ergebnisse gefunden werden, wobei berücksichtigt werden muss, dass das Verfahren die geforderten Echtzeitbedinungen erfüllen kann. Seien $g_1(x), ..., g_n(x)$ die eingesetzten Gaußfunktionen für eine Zufallsvariable x und seien $w_1, ..., w_n$ mit $w_i \geq 0$ sowie $\sum_i w_i = 1$ für $i = 1..n$ die verwendeten Gewichte. Dann ist durch $g(x) = \sum_i^n w_i \cdot g_i(x)$ eine gaußsche Mischungsfunktion bestimmt.

7.8.2 Arbeitsweise des Mixture of Gaussian Verfahrens

Da das *Mixture of Gaussian* Verfahren auf in dem RGB-Arbraum arbeitet und dessen Farbwerte durch einen Ausschnitt eines 3-dimensionalen Raumes beschrieben werden können (siehe hierfür den RGB-Farbraumwürfel aus Abschnitt 4.8), verwenden Stauffer und Grimson eine Erweiterung des Verfahrens für mehrdimensionale Normalverteilungen. Im Folgenden wird die Abreitsweise des Verfahrens beschrieben.
Sei I die zugrunde liegende Videosequenz eines Pixels $P(x_0, y_0)$. Dann ist zu an einem beliebigen Zeitpunkt t die Vergangenheit dieses Pixels, also dessen Farbwerte bis zu diesem Zeitpunkt, bekannt. Diese kann durch $\{X_1, .., X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\}$ angegeben werden. Sei K die Anzahl der verwendeten Gaußfunktionen. Dann berechnen Stauffer und Grimson die Wahrscheinlichkeit des Auftretens eines Pixels in Abhängig-

keit seiner Vergangenheit durch das Verwenden einer multivariaten Normalverteilung zu :

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \cdot \eta(X_t, \mu_{i,t}, \Sigma_i, t)$$

Hierbei werden die folgenden Parameter verwendet :

- t : Beschreibt den Zeitpunkt der Berechnung des Wahrscheinlichkeitswertes
- X_t : Gibt den Farbwert des Pixels zum Zeitpunkt t an
- K : Gibt die Anzahl der zur Beschreibung des Hintergrundmodells verwendeten Gaußfunktionen an
- $\omega_{i,t}$: Gibt das Gewicht für die i-te Gaußfuktion an. Je höher dieses Gewicht ist, desto häufiger konnte in den letzten Videobildern eine Ähnlichkeit zwischen den Pixelwerten und der Gaußfunktion ausgemacht werden. Die Gewichte werden nach jeder Aktualisierung normiert. Daher gilt $\sum_{i=1}^{K} \omega_{i,t} = 1$ zu Beginn und zu Ende jedes zu bearbeitenden Videobildes.
- $\eta(X_t, \mu_{i,t}, \Sigma_i, t)$: Steht für die multivariate Normal- beziehungsweise Gaußverteilung. Durch sie wird die Wahrscheinlichkeit bestimmt, wie groß aufgrund der bisher aufgetretenen Farbwerte, die Wahrscheinlichkeit eines jetzigen Auftretens von X_t ist.
- $\mu_{i,t}$: Durch diesen Parameter wird der mittlere Farbwert der i-ten Gaußverteilung angegeben.
- $\Sigma_{i,t}$: Steht für die sogenannte Kovarianzmatrix. Diese erweitert das Konzept der Varianzen auf mehrdimensionale Zufallsvariablen. Da die hier verwendete Zufallsvariable X_t drei Dimensionen besitzt, besteht die Kovarianzmatrix hier aus einer 3×3-Matrix.

Die in der Liste aufgeführte Dichtefunktion η lässt sich wie folgt berechnen :

$$\eta(X_t, \mu, \Sigma) = \frac{1}{\sqrt{(2 \cdot \pi)^n} \cdot \sqrt{|\Sigma|}} \cdot e^{-\frac{1}{2} \cdot (X_t - \mu_t)^T \cdot \Sigma^{-1} \cdot (X_t - \mu_t)}$$

Hierbei muss für die Determinante $|\Sigma| \neq 0$ gelten. Da das invertieren einer Matrix relativ rechenaufwändig ist, wird von den Entwicklern dieses Verfahrens vorgeschlagen, anstatt einer exakten Berechnung der Kovarianzmatrizen, die folgende Abschätzung zu verwenden :

$$\Sigma_{k,t} = \sigma_k^2 \cdot E_3 = \begin{pmatrix} \sigma_k^2 & 0 & 0 \\ 0 & \sigma_k^2 & 0 \\ 0 & 0 & \sigma_k^2 \end{pmatrix}$$

Hierbei stellt $\mathbb{R}^3$ die 3×3 Einheitsmatrix dar. Die aufgeführte Formel verwendet für die drei Dimensionen beziehungsweise Farbkanäle nur eine Varianz σ_k^2. Dadurch gelangt zwar ein gewisser Grad an Ungenauigkeit in das Modell, verglichen mit dem

sich so ergebenden Geschwindigkeitsvorteil, fällt dieser jedoch kaum ins Gewicht. Für die Invertierung der Matrix ergibt sich :

$$\Sigma_{k,t} = \begin{pmatrix} \sigma_k^2 & 0 & 0 \\ 0 & \sigma_k^2 & 0 \\ 0 & 0 & \sigma_k^2 \end{pmatrix} \overset{inv.}{\Longrightarrow} \Sigma_{k,t}^{-1} \begin{pmatrix} \frac{1}{\sigma_k^2} & 0 & 0 \\ 0 & \frac{1}{\sigma_k^2} & 0 \\ 0 & 0 & \frac{1}{\sigma_k^2} \end{pmatrix}$$

Aus der Dichtefunktion η lässt sich ein Distanzmaß zur Abstandsberechnung einer multivariaten Zufallsvariable X von einem Punkt der selben Dimensionalität, hier dem Mittelwert μ, herleiten. Dieses Maß heißt *Mahalanobis* Distanz und wird wie folgt berechnet :

$$d(X, \mu) = \sqrt{(X - \mu)^T \cdot \Sigma^{-1} \cdot (X - \mu)}$$

Die *Mahalanobis* Distanz entspricht somit dem Exponenten des e-Terms der oben genannten Dichtefunktion.
Zur Klassifikation eines Pixels in Vorder- beziehungsweise Hintergrund der Szene, arbeitet man nun wie folgt :

1. Die Gaußfunktionen werden nach den Quotienten $\frac{\omega}{\sigma}$ sortiert. Dieser Quotient ist besonders hoch wenn das Gewicht der Gaußfuktion hoch oder deren Standardabweichung relativ klein ist. Ein hohes Gewicht besitzt die Gaußfunktion, wenn sie bei der Analyse der letzten Videobilder häufig aktualisiert wurde. Die Varianz ist gering, wenn die Farbwerte, besonders während der letzten Videobilder, wenig um den Mittelwert verstreut lagen. Je höher dieser Quotient ist, desto eher handelt es sich bei der Funktion um einen wichtigen Hintergrund der Szene. Je kleiner der Quotient dagegen ist, desto unwichtiger erscheint der dazugehöriger Hintergrund, da er entweder schon einen längeren Zeitraum nicht mehr in der Szene zu sehen war, oder er durch eine große Varianz relativ wenig aussagekräftig ist.

2. Nach dem Sortieren werden die ersten b Gaußverteilungen als Hintergrundmodelle angesehen, so dass die Summe derer Gewichte größer als ein zu wählender Schwellwert T ist. Für das Hintergrundmodell B gilt bezüglich diesen b Verteilungen :

   $$B = \sum_{k=1}^{b} \omega_k \geq T$$

   Durch den Schwellwert T werden die besten Verteilungen, bis auf einen durch ihn bestimmten Prozentwert, für das Hintergrundmodell verwendet. Ist T klein, so ist das Modell in der Regel unimodal, da der Hintergrund dann durch nur eine Funktion beschrieben wird. Ist T dagegen sehr groß beziehungsweise gilt $T = 1$, dann wird der Hintergrund meisst durch alle Gaußfunktionen beschrieben. Das heißt, dass auch erst kürzlich erzeugt Funktionen schon komplett in das Modell integriert wurden. Häufig ist ein zu großer Wert für T nicht erwünscht, da Farbwerte erst dann in das Hintergrundmodell integrieren sollten, wenn sie bislang ausreichend lange beziehungsweise häufig in der Überwachungsszene aufgetreten sind.

3. Mittels *Mahalanobis* Distanz wird der Abstand eines Pixels mit dem Farbwert X_t zu den Gaußfunktionen die das Hintergrundmodell B bestimmen, berechnet. Dabei wird die Abstandsberechnung bezüglich den Mittelwerten der Funktionen durchgeführt. Gilt für eine dieser Funktionen $d(X_t, \mu_{i,t}) \leq k \cdot \sigma_{i,t}$, so konnte eine Ähnlichkeit zu einer Funktion die den Hintergrund beschriebt, festgestellt werden. Der Pixel wird entsprechend als Hintergrundpixel klassifiziert. Konnte dagegen keine Ähnlichkeit festgestellt werden, so gilt der Pixel als Vordergrund. Der Parameter k kann wie in Abbildung 4.5 gezeigt, ermittelt werden.

Damit das Verfahren Änderungen in der überwachten Szene, wie sie beispielsweise durch Beleuchtungsänderungen hervorgerufen werden (siehe hierzu Abschnitt 6.4), in sein Hintergrundmodell aufnehmen kann, muss dieses regelmäßig, im Allgemeinen nach jedem bearbeiteten Videobild, aktualisiert werden. Dabei wird nur die erste Gaußfunktion, nach der oben genannten Sortierung, zu der der Farbwert X_t ähnlich ist, aktualisiert. Eine solche Aktualisierung kann durch die sogenannte Lernkonstante α wie folgt durchgeführt werden :

$$\omega_{k,t+1} = \begin{cases} (1-\alpha) \cdot \omega_{k,t} + \alpha & \text{falls } d(X_t, \mu_{k,t}) \leq k \cdot \sigma_{k,t} \\ (1-\alpha) \cdot \omega_{k,t} & \text{sonst} \end{cases}$$

$$\mu_t = (1-\rho) \cdot \mu_{t-1} + \rho_{X_t}$$

$$\sigma_t^2 = (1-\rho) \cdot \sigma_{t-1}^2 + \rho \cdot (X_t - \mu_t)^T \cdot (X_t - \mu_t)$$

Wobei $\rho = \alpha \cdot \eta(X_t \| \mu_k, \sigma_k)$.
Häufig wird

$$\rho = \begin{cases} 1 \text{ falls } d(X_t, \mu_{k,t}) \leq k \cdot \sigma_{k,t} \\ 0 \text{ sonst} \end{cases}$$

verwendet, weil die verwendete Exponentialfunktion häufig relativ kleine Werte liefert. Kann keine Ähnlichkeit des Farbwertes X_t zu einer der Gaußfunktionen des Hintergrundmodells B festgestellt werden, so wird die am wenigstenn wahrscheinliche Funktion durch eine Neue ersetzt. Diese wird mit $\mu_t = x_t$ sowie einer relativ hohen Varianz σ_t^2 und einem kleinen Gewicht μ_t initialisiert. Dies hat zur Folge, dass die Funktion für die ersten Videobilder nach ihrer Erzeugung, noch nicht in das Hintergrundmodell B integriert wird. Wie schnell es sich letztlich genau in das Modell einarbeitet hängt von der Lernkonstante α sowie durch den Schwellwert T ab und lässt sich durch diese Parameter steuern.

7.8.3 Eigenschaften, Laufzeit- sowie Speicheranforderungen des Mixture of Gaussians Verfahrens

Das *Mixture of Gaussian* Verfahren besitzt ein sogenanntes multimodales Hintergrundmodell, da es für jeden Pixel durch mehr als eine Verteilung modelliert wird. Jedoch erhöht sich dadurch der benötigte Rechenzeit sowie Speicherbedarf. Die Laufzeit des Verfahrens, liegt in $\mathcal{O}(|gaussians| \cdot |Frames| \cdot |Pixel|)$, der dabei benötigte Speicher dagegen in $\mathcal{O}(|gaussians| \cdot |Pixel|)$. Durch das Verwenden einer einzigen Varianz für alle Farbkanäle, kann die Rechenzeit reduziert werden. Dies liegt hauptsächlich daran,

Algorithmus 7.8 Mixture of Gaussian Verfahren

procedure MIXTURE OF GAUSSIAN($sequence, \alpha, k, T$)
  $InitBackgroundmodel()$
  for all Frames F_t **do**
    for all $Pixels(x, y)$ **do**
      $match \leftarrow false$
      $Pixel \in Background$
      for all Gaussians $(\mu_k, \sigma_k^2) \in B$ **do**
        if $((MahalanobisDistance(X_t, \mu_t) \leq k \cdot \sigma) \wedge (match == true))$ **then**
          $match \leftarrow true$
          $UpdateGaussian(\mu_t, \sigma_k^2)$
        end if
      end for
      if $(match == true)$ **then**
        // delete least probable Gaussian
        $CreateGaussian(X_t, \sigma_{init})$
        $Pixel \leftarrow Foreground$
      end if
      Normalize gaussian weights so that $\sum_{i=1}^{K} \omega_{i,k} = 1$
      SortGaussians by $\frac{\omega_k}{\sigma_k}$
      Compute background modell $B = argmin_b(\sum_{k=1}^{b} \omega_k \geq T)$
    end for
  end for
end procedure

dass die zeitintensive Matrixinvertierung umgangen werden kann.
Durch die regelmäßigen Aktualisierungen des Hintergrundmodells kann sich das Verfahren an Änderungen der Szene anpassen sowie neue Objekte in das Modell integrieren. Durch die Parameter α sowie T kann die Anpassungsgeschwindigkeit des Verfahrens beeinflusst werden.

7.9 Codebook Verfahren

Dieser Abschnitt widmet sich dem Codebook Verfahren [KCHD04], das von Kyunungnam Kim et. al [KCHD04] entwickelt wurde. Es gehört zu den multimodalen *Background Subtraction* Verfahren.
Das Verfahren arbeitet in zwei Phasen. Einer Trainingsphase, in der bezüglich einer Trainingssequenz, also einer Teilsequenz des zugrunde liegenden Überwachungsvideos, ein initiales Hintergrundmodell mit Hilfe eines Clusterverfahrens berechnet wird. Dieses wird in der zweiten Phase verwendet, um die Subtraktionsbilder zu berechnen. Damit das Verfahren die Möglichkeit besitzt, Veränderungen der überwachten Szenen in sein Modell integrieren zu können, wird in dieser Phase zusätzlich eine Aktualisierung mit Hilfe des zuletzt bearbeiteten Videobildes durchgeführt.
In den nächsten Abschnitten wird die Arbeitsweise des *Codebook* Verfahrens sowie die von ihm verwendeten Parameter, Variablen und Funktionen vorgestellt und erläutert.

7.9.1 Variablen und Parameter des Codebook Verfahrens

Das initiale Hintergrundmodell des *Codebook* Verfahrens besteht pixelweise aus sogenannten *Clustern*, die bezüglich einer gegebenen Trainingssequenz berechnet werden. Unter einem Cluster ist in diesem Zusammenhang so etwas wie eine Gruppe ähnlicher Pixelwerte oder Äquivalenz- Klasse zu verstehen. Die Farbwerte eines Pixels beeinflußen dabei während der Initialisierungsphase das Aussehen desjenigen Clusterrepräsentanten, dem er aufgrund gewisser Eigenschaften, die durch spezielle Tests überprüft werden können, besonders ähnlich erscheint. Kann zu keinem Repräsentanten eine ausreichend große Ähnlichkeit nachgewiesen werden, so wird ein neuer angelegt. Die Menge der so entstehenden Cluster eines Pixels wird als *Codebook* bezeichnet, jeder einzelne dagegen als Codewort.
In diesem Unterkapitel werden die zur Erzeugung und Verwaltung benötigten Variablen beziehungsweise Parameter vorgestellt und näher erläutert.
Durch χ wird im weiteren die Trainingsseqquenz eines Pixels bezeichnet. Besteht die Sequenz aus n Videobildern, so besteht χ aus n RGB-Vektoren, das heißt $\chi = \{x_1, x_2, ..., x_n\}$, wobei $x_i \in RGB$. Mit $\mathcal{C} = \{c_1, c_2, ..., c_L\}$ sei das Codebook eines Pixel bezeichnet, das aus L Codewörten besteht. L ist jedoch nicht konstant. Demnach können zwei verschiedene Pixel eine unterschiedliche Anzahl an Codewörtern besitzen. Jedes einzelne Codewort setzt sich aus folgenden neun Parametern zusammen:

1. $\bar{R}$: Die Rotkomponente des RGB-Vektors v_i.
2. $\bar{G}$: Die Grünkomponente des RGB-Vektors v_i.
3. $\bar{B}$: Die Blaukomponente des RGB-Vektors v_i.
4. $\check{I}$: Die minimiale Helligkeit, die das Codewort akzeptiert.
5. $\hat{I}$: Die maximale Helligkeit, die das Codewort akzeptiert.
6. f : Die Frequenz mit der das Codewort bisher aktualisiert worden ist.
7. λ : Steht für das größte Zeitintervall in dem das Codewort während der Trainingsphase nicht aktualisiert wurde.
8. p : Gibt den Zeitpunkt an, zudem dieses Codewort angelegt wurde.
9. q : Gibt den Zeitpunkt an, an dem dieses Codewort zuletzt aktualisiert wurde.

Wie dieser Aufzählung zu entnehmen ist, werden die ersten drei Parameter zu einem RGB-Vektor v_i zusammengefasst, die weiteren Parameter werden dagegen zu einer Hilfsvariablen aux. Mit Hilfe dieser Parameter werden letztlich die Ähnlichkeitstests zwischen den Pixeln der zu untersuchenden Videobilder und dem Hintergrundmodell des Verfahrens durchgeführt. Wie diese Ähnlichkeitstest durchgeführt werden, ist Inhalt des nächsten Abschnittes.

Algorithmus 7.9 Initialisierung Codebook Verfahren

procedure INIT CODEBOOK(*trainingSequence*)
  for all Pixels **do**
    $L \leftarrow 0$
    $\mathcal{C} \leftarrow \emptyset$
    for all Frames **do**
      $x_t = (R, G, B)$
      $I = R + G + B$

      // find matching codeword v_i
      for all codewords **do**
        if $((dist(x_t, v_i) \leq \epsilon_1) \wedge (brightness(I, (\check{I}_i, \hat{I}_i))! = true))$ **then**
          $match \leftarrow true$
          $index \leftarrow i$
          break
        end if
      end for

      // create new codeword if there is no match
      if $(match == 0)$ **then**
        $L \leftarrow L + 1$
        $v_L \leftarrow (R, G, B)$
        $aux_L \leftarrow (I, I, 1, t-1, t, t)$
      end if

      // if there was a match, update the corresponding codeword
      if $(match == 1)$ **then**
        $m \leftarrow index$
        $v_m \leftarrow (\frac{f_m \cdot \bar{R}_m + R}{f_m + 1}, \frac{f_m \cdot \bar{G}_m + G}{f_m + 1}, \frac{f_m \cdot \bar{B}_m + B}{f_m + 1})$
        $aux_m = (min\{I, \check{I}_m\}, max\{I, \hat{I}_m\}, f_m + 1, max\{\lambda_m, t - q_m\}, p_m, t)$
      end if
    end for
  end for
  for all $(Codewordsc_i)$ **do**
    $\lambda_i \leftarrow max\{\lambda_i, numberOfFrames - q_i + p_i - 1\}$
  end for
end procedure

7.9.2 Initialisierungsphase des Codebook Verfahrens

Dieser Abschnitt widmet sich der Berechnung des initialen Hintergrundmodells des *Codebook* Verfahrens. Diese Initialisierungsphase ist in Algorithmus 7.9 in Pseudocode zu sehen. Die verwendeten Variablen beziehungsweise Parameter sind im vorherigen Abschnitt aufgeführt, wo auch deren Bedeutung nachgelesen werden kann.

Im Wesentlichen besteht der Algorithmus aus folgenden drei Komponenten :

1. Finde ein Codewort c_m aus dem Codebook $C = \{c_i \| 1 \leq i \leq L\}$ welches Ähnlichkeit zu dem aktuellen Pixel aufweist.
2. Falls es kein solches Codewort gibt, lege ein neues an.
3. Falls es jedoch ein ähnliches Codewort gibt, aktualisiere dieses mit Hilfe des untersuchten Pixels.

Wie sich dem oben aufgeführten Algorithmus entnehmen lässt, wird bei der Durchführung des Ähnlichkeitstest nicht nach demjenigen Cluster gesucht, der dem Pixel am ähnlichsten ist, sondern nach dem ersten, der bezüglich diesem Test zu einem positiven Ergebnis führt. Dies reduziert die Laufzeit des Verfahrens unter Umständen enorm, da die komplette Durchsuchung der Codebooks dadurch häufig nicht notwendig ist.
Die durchzuführenden Ähnlichkeitstest bestehen wiederum aus zwei Komponenten. Zum einen aus der Methode *dist*, die den Farbabstand zwischen dem Pixel und den in dem Farbwert v_i des Codeworts c_i berechnet. Zum anderen aus der Funktion *brightness* die überprüft, ob die Helligkeit eines Pixels innerhalb des von dem Codewort akzeptierten Helligkeitsbereiches liegt.
Die diesen beiden Methoden zugrundeliegende Theorie ist in Abbildung 7.2 zu sehen. Beobachtungen des Entwicklerteams ergaben, dass die Farbwerte eines Hintergrundpixels für viele Trainingssequenzen hauptsächlich in der Nähe der Geraden, die im RGB-Würfel (siehe Kapitel 4.8) durch die Farbwerte eines Codewortes und dem Ursprung konstruiert werden konnten, zu finden waren. Entsprechend sollte der Ähnlichkeitstest auch nur ein positives Resultat liefern, wenn der Farbwert eines Pixels in der Nähe einer so konstruierten Geraden liegt. Problematisch hierbei ist, dass jede solche Gerade durch dem Ursprung geht, der die Farbe Schwarz repräsentiert. Ohne Einschränkungen würde das eben beschriebene Verfahren eine Ähnlichkeit zwischen jeder sehr dunklen beziehungsweise schwarzen oder fast schwarzen Farbe zu jedem Codewort feststellen, da sich diese Farben alle in der Nähe dieser Geraden befinden. Um dieses Problem zu vermeiden, wird für die Tests nicht die kompletten Geraden verwenden, sondern nur Ausschnitte von ihnen, die durch den von dem entsprechenden Codewort akzeptierten Helligkeitsbereich definiert werden.

In den Variablen $\check{I}$ sowie $\hat{I}$ sind der minimale beziehungsweise der maximale Helligkeitswert gespeichert, die für das entsprechende Codewort bisher aufgetreten sind, gespeichert. Um einen Gewissen Spielraum gegenüber Beleuchtungsänderungen der Szene und gegenüber Schatten zu gewähren, wird die Helligkeit I eines Pixel genau dann akzeptiert, falls sie die Bedingung $I_{low} \leq I \leq I_{high}$ erfüllt, wobei $\check{I} \geq I_{low}$ und $\hat{I} \leq I_{high}$ gelten muss (Die Bestimmung von I_{low} sowie I_{high} ist weiter unten in dem Algorithmus 7.11 zu finden).
Der Parameter ϵ gibt dagegen an, wie weit der Farbwert x_t von dem des Codeworts v_i für eine Ähnlichkeit maximal entfernt sein darf. Zur Bestimmung des Abstandes δ wird der Farbwert x_t auf die Gerade zwischen dem Ursprung und v_i projiziert.
Demnach gilt ein Pixel mit dem Farbwert x_t als dem Codewort v_i ähnlich, falls er sich innerhalb des durch die Paramter I_{low} , I_{high} sowie ϵ aufgespannten Zylinders befindet. Zwar wurde in Abschnitt 4.10 erläutert, dass sich Farben konstanter Helligkeit auf einer Ebene senkrecht zur Diagonalen des RGB-Würfels befinden, jedoch befinden sich die so entstehenden Zylinder bezüglich diesen Ebenen etwas verschoben. Dies führt dazu, dass nicht für jeden Farbwert genau die selben Helligkeitsvariationen akzeptiert werden. Jedoch fällt dieser Umstand nicht besonders ins Gewicht und lässt

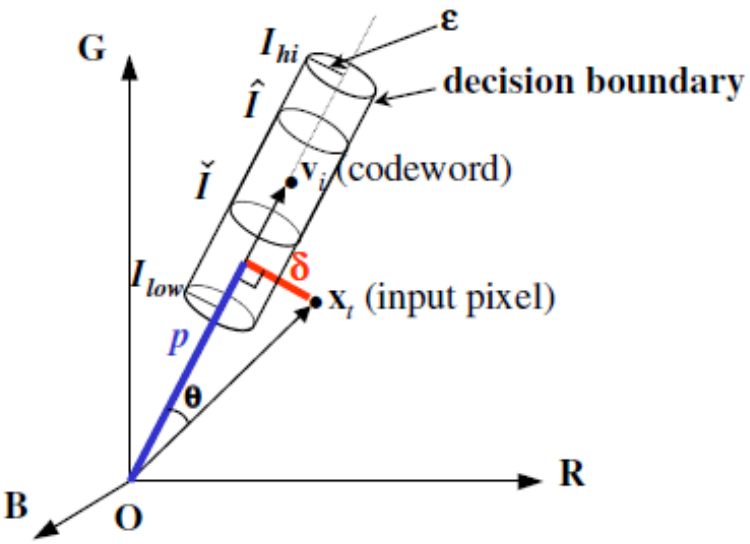

Abbildung 7.2: In dieser Abbildung ist das Klassifikationsmodell des *Codebook* Verfahrens zu sehen. Ein Pixelwert ist einem Codewort ähnlich, wenn er sich innerhalb eines Zylinder in dem das Codewort liegt, befindet

sich durch die Bestimmung von I_{low} sowie I_{high} regulieren, falls dies erforderlich wäre. Jedoch liefert diese etwas ungenaue Berechnungsmethode gegenüber einer exakten Geschwindigkeitsvorteile, die den angesprochenen Nachteil wett machen.
Die Methode *dist* zur Berechnung des Farbabstandes ist in Algorithmus 7.10 in Pseudocode aufgeführt. Die Berechnung der Methode *brightness* ist dagegen in Algorithmus 7.11 zu sehen.
Die Initialisierungsphase des Codebook Verfahrens ist somit für die pixelweise Berechnung der Anfangscluster, der sogenannten Codebooks, zuständig. Während der ersten Videobilder ist der Akzeptanzbereich der Codewörter noch nicht besonders groß, so dass häufig ein neues Codewort angelegt wird, wohingegen der selbe Pixel bei späteren Videobildern kein neues Codewort erzeugt hätte, weil der Akzeptanzbereich dann häufig deutlich größer ist. In der Regel werden daher gerade zu Beginn der Trainingsphase viele Codewörter erzeugt, die dann nicht mehr aktualisiert werden, da in dem Aktualisierungsschritt aus Geschwindigkeitsgründen nicht nach dem Codewort mit der größten Ähnlichkeit gesucht wird, sondern nach dem ersten das den Ähnlichkeitstes besteht. Diese Codewörter sollten nach der Trainingsphase aus den Codebooks entfernt werden, da sie die benötigte Rechenzeit im Subtraktionsphase unnötig in die Höhe treiben. Entfernt werden alle Codewörter, die während der Trainingsphase während mindestens $T_{\mathcal{M}}$ Videobilder nicht mehr aktualisiert wurden. Die maximale Zeit, die ein Codewort nicht Aktualisiert wurde ist in dem Parameter λ gespeichert. $T_{\mathcal{M}} = \frac{|Frames|}{2}$ hat sich laut Autoren für das Verfahren bewährt. Der Parameter f, also die Frequenz mit der ein Codewort während der Trainingsphase aufgetreten ist, könnte ebenfalls zum Filtern der Codewörter verwendet werden. Dies liefert aber nach Aussage der Autoren kein besseres Ergebnis.
Das initiale Hintergrundmodell sieht wie folgt aus :

$$\mathcal{M} = \{c_m \mid c_m \in \mathcal{C} \wedge \lambda_m \leq T_{\mathcal{M}}\}$$

Algorithmus 7.10 CodebookDist

procedure CODEBOOKDIST(x_t, v_i) returns distance
$\|x_t\|^2 = R^2 + G^2 + B^2$
$\|v_i\|^2 = \bar{R}_i^2 + \bar{G}_i^2 + \bar{B}_i^2$
$\langle x_t, v_i \rangle^2 = (\bar{R}_i \cdot R + \bar{G}_i \cdot G + \bar{B}_i \cdot B)^2$
$p^2 \leftarrow \frac{\langle x_t, v_i \rangle^2}{\|x_t\|^2 - p^2}$
$distance \leftarrow \sqrt{\|x_t\|^2 - p^2}$
end procedure

Algorithmus 7.11 CodebookBrightness

procedure CODEBOOKBRIGHTNESS($I, \langle \check{I}, \hat{I} \rangle$) return Boolean
$\alpha \in [0.4, 0.7]$
$\beta \in [1.1, 1.5]$
$I_{low} \leftarrow \alpha \cdot \hat{I}$
$I_{high} = min(\beta \cdot \hat{I}, \frac{\check{I}}{\alpha})$

if $(I_{low} \leq I \leq I_{hi})$ **then**
return $true$
else
return $false$
end if
end procedure

7.9.3 Subtraktionsphase des Codebook Verfahrens

In der zweiten Phase des *Codebook* Verfahrens, der Subtraktionsphase, werden die Subtraktionsbilder erzeugt, also die Pixel eingehender Videobilder in Vordergrund- beziehungsweise Hintergrundpixel eingeteilt. Diese Klassifikation wird durch die Methoden *dist* sowie *brightness* durchgeführt, die schon während der Initialisierungsphase verwendet wurden und in Algorithmus 7.10 sowie 7.11 aufgeführt sind. Diese zwei Funktionen überprüfen, ob sich ein Pixel in dem in Abbildung 7.2 dargestellten Zylinder befindet, oder nicht. Ist er in diesem zu finden, so ist er dem zugehörigen Codewort ähnlich und wird somit als Hintergrundpixel klassifiziert. Liegt er jedoch außerhalb, so kann keine ausreichend große Ähnlichkeit erkannt werden und der Pixel wird als Vordergrund eingestuft.
Die Subtraktionsphase ist in Algorithmus 7.12 in Pseudoceode aufgeführt. Zur Aktualisierung der Codewörter wird die Methode $update$ verwendet. Diese Methode aktualisiert das dem Pixel ähnliche Codewort genau so, wie auch in der Initialisierungspahe aktualisiert wurde (siehe 7.9). Dadurch kann sich das Hintergrundmodell an Veränderungen der Überwachungsszene regelmäßig anpassen. Jedoch werden dabei nur die Codewörter aktualisiert, die aktuell zu Hintergrundpixeln gehören. Befindet sich an einer Position gerade ein Vordergrundobjekt, so wird keines der existierenden Codewörter aktualisiert. Daher bietet das bisher beschriebene Verfahren auch keine Möglichkeit, Vordergrundobjekte die sich nicht mehr bewegen, in den

Algorithmus 7.12 CodebookSubtraction

```
procedure CodebookSubtraction(sequence)
    for all (Frames ∧ Pixels) do
        x ← (R, G, B)
        I ← R + G + B
        for all Codebooks c_i do
            if (dist(x, v_i)) ∧ (I, ⟨Ǐ_i, Î_i⟩ == true) then
                match ← true
                update(c_i, x, I)
            end if
        end for
        if (match == true) then
            pixel ← Background
        else
            pixel ← Foreground
        end if
    end for
end procedure
```

Hintergrund aufzunehmen. Solche Verfahren werden auch selektive Verfahren genannt. Bei einer Überwachung von Parkplätzen ist diese Eigenschaft jedoch dringen erforderlich. Eine Möglichkeit die statischen Vordergrundobjekte in das Modell zu integrieren wird im nächsten Abschnitt vorgestellt. Die dort aufgeführte Erweiterung des Verfahrens wurde fünf Jahre nach der ursprünglichen Version, veröffentlicht.

7.9.4 Eigenschaften des Codebook Verfahrens

Das *Codebook* Verfahren besitzt ein sogenanntes multimodales Hintergrundmodell, da es für jeden Pixel mehr als einen Farbwert besitzen kann. Jedoch müssen zur Durchführung des Ähnlichkeitstest im Extremfall alle Codewörter abgearbeitet werden. Die Anzahl der Codewörter die ein Pixel besitzt kann je nach Trainingsmenge stark variieren, ist demnach nicht konstant, wie beispielsweise bei dem *Mixture Of Gaussian* Verfahren (siehe Abschnitt 7.8). Sei L_{max} die maximale Anzahl der Codewörter die ein Pixel nach der Trainingsphase besitzt. Dann lässt sich die Laufzeit des Verfahrens in O-Notation durch $\mathcal{O}(L_{max} \cdot |Frames| \cdot |Pixel|)$ abschätzen. Der benötigte Speicherbedarf liegt in $\mathcal{O}(L_{max} \cdot |Pixels|)$. Die verwendeten Methoden *dist* sowie *brightness* haben durch die Abschätzungen in O-Notation keinen wesentlichen Einfluss auf die Laufzeit. Die Trainingsphase besitzt dagegen die selben Anforderungen an die Laufzeit. Da am Ende der Initialisierungsphase jedoch Codewörter ausgefiltert werden, die während dem Training für eine längere Zeit nicht aktualisiert wurden, kann hier die maximale Anzahl an Codewörtern L_{max} der Pixel jedoch deutlich höher liegen.
Wie das Verfahren bei einigen der in Kapitel 6 wesentlichen Herausforderungen zurecht kommt, ist Teil der im Rahmen dieser Diplomarbeit durchgeführten Evaluation und kann in Kapitel 9 nachgelesen werden.

7.9.5 Erweiterung des Verfahrens durch ein Schichtenmodell zur besseren Adaptivität

Das Verfahren, so wie es bisher beschrieben wurde, kann zwar Änderungen des Hintergrunds einer Szene in sein Modell integrieren, jedoch kann es keine Vordergrundobjekte in dieses aufnehmen, falls diese sich über einen längeren Zeitraum nicht bewegen und daher eine gewisse Zeit statisch sind. Für eine Vielzahl an möglichen Überwachungsszenarien ist dieser Umstand nicht nicht akzeptabel. Um die Vordergrundobjekte letztlich in das Hintergrundmodell zu integrieren, entwickelten die Autoren dieses Verfahrens ein Schichtenmodell. Dieses Arbeitet wie folgt :

1. Berechne während einer Trainingsphase (wie oben beschrieben) ein initiales Hintergrundmodell $\mathcal{M}$
2. Erzeuge nach der Trainingsphase ein zweites Modell $\mathcal{H}$, das als Puffer dient
3. Suche für den aktuellen Pixel X ein Codewort aus $\mathcal{M}$ dass diesem ähnlich ist. Falls es ein solches gibt, wird eine Aktualisierung des Codeworts durchgeführt.
4. Existiert ein solches Codewort nicht, dann suche eines in $\mathcal{H}$. Falls dieses existiert, aktualisiere es. Existiert es jedoch auch nicht, so lege ein neues Codewort an und füge es dem Puffer $\mathcal{H}$ zu.
5. Entferne diejenigen Codewörter aus $\mathcal{H}$, die seit längerer zeit nicht mehr Aktualisiert wurden. Es ergibt sich $\mathcal{H} \leftarrow \{h_i | h_i \in \mathcal{H} \wedge h_{i,\lambda} \leq T_{\mathcal{H}}\}$
6. Füge den Codewörtern des Hintergrundmodell $\mathcal{M}$ diejenigen des Puffers $\mathcal{C}$ hinzu, die ausreichend oft aktualisiert wurden. Es ergibt sich : $\mathcal{M} \leftarrow \mathcal{M} \cup \{h_i | h_i \in \mathcal{H} \wedge h_{i,f} \geq T_{add}\}$
7. Entferne letztlich diejenigen Wörter aus $\mathcal{M}$, die für eine ausreichend lange Zeit nicht mehr aktualisiert wurden. Somit ergibt sich: $\mathcal{M} \leftarrow \{c_i | c_i \in \mathcal{M} \wedge c_{i,\lambda} \leq T_{delete}\}$

Somit wird für Pixel, die zu keinem Codewort ähnlich sind, ein neues angelegt und dieses in einem Puffer gespeichert. Wird für die im folgenden eingehenden Pixel an dieser Position dann genügend häufig eine Ähnlichkeit festgestellt, so wird wird das Codewort aus dem Puffer in das Hintergrundmodell transferiert. Um den Speicherbedarf zu reduzieren und damit nicht unnötig uninteressante Codewörter verwaltet werden, müssen diese regelmäßig ausgefiltert werden. Hierbei wird der Parameter λ verwendet, der angibt, wie lange das entsprechende Codewort nicht mehr aktualisiert wurde. Ist schon seit längerem keine Aktualisierung mehr durchgeführt worden, so kann das Codewort aus dem Modell entfernt werden. Codewörter die sich noch im Puffer befinden, werden so ebenfalls entfernt.
Die verwendeten Schwellwerte $T_{\mathcal{H}}$, T_{add} sowie T_{delete} müssen für die zu bearbeitende Aufgabe entsprechend gewählt werden. gegenüber anderen Verfahren hat das Schichtenmodell den Vorteil, dass die Zeit innerhalb der ein Vordergrundobjekt in das Hintergrundmodell integriert beziehungsweise aus diesem entfernt werden soll, direkt durch die Schwellwerte eingestellt werden kann. Die Schichten verhindern zudem ein Vermischen von Hitergrund- und Vordergrundfarben bei der Adaption wie sie beispielsweise häufig bei den unimodalen Hintegrundmodellen auftritt.
Durch das Aussortieren wird letztlich der benötigte Laufzeit- und Speicherbedarf

reduziert, an den jeweiligen Abschätzungen in O-Notation ändert sich jedoch nichts. Sei L_{max} weiterhin die maximale Anzahl an möglichen Codewörtern pro Pixel. Dann lässt sich die Laufzeit weiterhin durch $\mathcal{O}(L_{max} \cdot |Frames| \cdot |Pixel|)$ und der Speicherbedarf durch $\mathcal{O}(L_{max} \cdot numPixels)$ angeben. L_{max} ist dabei durch T_{delete} nach oben beschränkt.

7.10 Das Verfahren von Li et al.

7.10.1 Grundlagen

Liyuan Li et al. stellen in ihrer Arbeit mit dem Titel *Foreground Object Detection from Videos Containing Complex Background* [LHGT03] ein *Background Subtraction* Verfahren vor, das laut deren Aussage Vordergrundobjekte auch bei schwierigen und komplexen Hintergründen detektieren kann. Es ist multimodal und führt seine Klassifikation der Pixel in Vorder- oder Hintergrund bezüglich einer Entscheidungsregel die auf dem Satz von Bayes basiert und einem sogenannten *Feature Vector*, durch. Generell lassen sich beliebige Eigenschaften, also *Features*, für die Klassifizierung verwenden, jedoch verwende die Autoren ausschließlich Farbe.
Sei v_t der besagte *Feature Vektor* der sich zum Zeitpunkt t aus einem Videobild an einer Stelle $s = (x, y)$ berechnen lässt. Nach dem Satz von Bayes lässt sich die Wahrscheinlichkeit P dass ein Pixel dem Vordergrund oder dem Hintergrund zugehört, wie folgt berechnen :

$$P(C|v_t, s) = \frac{P(v_t|C, s) \cdot P(C|s)}{P(v_t|s)}, C \in \{background, foreground\}$$

Ein Pixel gehört demnach dem Hintergrund an, falls die Wahrscheinlichkeit dass sein *Feature Vector* dem Hintergrund angehört größer ist, als dass er dem Vordergrund angehört. Demnach ist zu überprüfen, ob folgende Bedingung erfüllt ist :

$$P(background|v_t, s) \geq P(foreground|v_t, s)$$

Da ein Pixel entweder in Vorder- oder Hintergrund klassifizieren lässt, gilt folgende Beziehung :

$$P(v_t|s) = P(v_t|background, s) \cdot P(background|s) + P(v_t|foreground, s) \cdot P(foreground, s)$$

Aus diesen Beziehungen lässt eine Formel zur Klassifikation herleiten, die keine Aussagen über den Vordergrund beinhaltet. Diese lautet :

$$2 \cdot P(v_t|b, s) \cdot P(b|s) \geq P(v_t|s)$$

Das Verfahren sammelt und aktualisiert Werte, mit deren Hilfe sich nach der eben beschriebenen Formal prüfen lässt, ob ein Pixel dem Hintergrund der Szene angehört oder nicht.
Um schnellere Laufzeiten zu ermöglichen, wird der verwendete RGB-Farbraum (siehe Abschnitt 4.8) stärker diskretisiert. Prinzipiell ist eine beliebige Diskretisierung möglich, eine Aufteilung in 32 oder 64 Stufen hat sich laut Autoren als gut herausgestellt. Zur mathematische Beschreibung von $P(v_t|s)$ und $P(v_t|background, s)$ wählen die Autoren Histogramme mit der eben beschriebenen Diskretisierung.

Da jeder einzelne Pixel durch durch einen *Feature Vector* repräsentiert wird, lässt sich auch der Hintergrund der Szene durch solche Vektoren beschreiben. Der Hintergrund einer Szene zeichnet sich durch eine gewisse Statik aus, so dass ein relativ kleine Menge der Vektoren für dessen Beschreibung ausreicht, was bei Vordergrundobjekten dagegen nicht der Fall ist.
Seien im Folgenden $P(v_t^i|background, s)$ die ersten N Histogrammeinträge nach ihrer Wahrscheinlichkeiten absteigend sortiert. Beobachtungen der autoren ergaben, dass dann eine auf N bezogen deutlich kleiner Zahl N_1 existiert, die folgende Bedingungen erfüllt :

$$(\sum_{i=1}^{N_1} P(v_t^i|background, s) \geq M_1) \wedge (\sum_{i=1}^{N_1} P(v_t^i|foreground, s) \leq M_2)$$

M_1 und M_2 sind wählbare Prozentwerte. Sei beispielsweise $M_1 = 0.9$ und $M_2 = 0.1$, dann ergibt die Summe über die ersten N_1 sortierten Wahrscheinlichkeiten bezüglich des Hintergrunds einen Wert der größer als M_1 ist. Werden dagegen die ersten N_1 auf den Vordergrund bezogenen Wahrscheinlichkeiten aufaddiert, so wird der Wert M_2 dabei nicht überschritten. Diese Beaobachtung spiegelt den Sachverhalt wieder, dass der Hintergrund der Szene durch relativ wenige *Features* beschrieben werden kann.
Für den Hintergrund- sowie den Vordergrund wird eine Tabelle erstellt und während der Laufzeit aktualisiert, die die N_2 wahrscheinlichsten *Features* beinhalten. Es gilt $N_1 < N_2$ so dass ein Puffer entsteht, der zum lernen neuer Hintergrundwerte verwendet werden kann. Die Tabelle beinhaltet folgende Werte :

$$S_{v_t}^{s,t,i} = \begin{cases} p_v^{t,i} = P(v_t^i|s) \\ p_{vb}^{t,i} = P(v_t^i|b, s) \\ v_t^i = (a_1^i, .., a_n^i)^T \end{cases}$$

Der *Feature Vector* v_t besitzt dabei die Dimensionalität n.

7.10.2 Arbeitsweise

Die Arbeitsweise des Verfahrens gliedert sich in vier Komponenten. Diese sind :

1. Änderungsdetektion
2. Änderungsklassifikation
3. Segmentierung der Vordergrundobjekte
4. Aktualisierung und Lernen neue Hintergrundwerte

Sei $I(s, t)$ die Farbe des Pixels an der Stelle s eines Videobildes zum Zeitpunkt t. $B(s, t)$ sei dagegen die Farbe des Hintergrundmodells an der Stelle s zum Zeitpunkt t. Um eine Änderung zu Detektieren wird zunächst für jede Farbkomponente eine Bilddifferenz mit adaptivem Schwellwert nach [Roso2] durchgeführt. So werden zwei Differenzen generiert, eine Hintergrunddifferenz $F_{bd}(s, t)$ und eine zeitliche Differenz $F_{td}(s, t)$.
Nachdem die Änderungen erkannt wurde, werden diese Klassifiziert. Falls $F_{td}(s, t) = 1$, so gehört der entsprechende Pixel zu einem sich bewegenden Vordergrundobjekt.

Ansonsten ist es ein stationärer Pixel der zu einem stationären Objekt gehört. Diese Klassifikationen werden nun noch einmal unterteilt, um das Verfahren robuster zu machen.
Für einen stationären Pixel s wird ein *Feature Vector* $v_t = c_t = [r_t, g_t, b_t]^T$ mit $L = 64$ Quantisierungsstufen bestimmt. Für einen sich bewegenden Pixel wird dagegen $v_t = c_{tt} = [r_{t-1}, g_{t-1}, b_{t-1}, r_t, g_t, b_t]^T$ berechnet, jedoch mit $L = 32$. Der Vektor v_t kann nun mit den N_1 gelernten Vektoren verglichen werden, die in der Tabelle $S_{v_t}^{s,t,i}$ enthalten sind.
Die für den Ähnlichkeitstest benötigten Wahrscheinlichkeiten, lassen sich aus der Tabelle wie folgt bestimmen :

$$P(background|s) = p_b^{s,t}$$

$$P(v_t|s) = \sum_{j \in \mathcal{M}(v_t) p_{v_t}^{s,t,j}}$$

$$P(v_t|b,s) = \sum_{j \in \mathcal{M}(v,t) p_{vb}^{s,t,j}}$$

Die hierfür verwendete Menge $\mathcal{M}(v_t)$ bestimmt sich wie folgt :

$$\mathcal{M}(v_t) = \{k : \forall m \in \{1,..,n\}, \left|a_m - a_m^k\right| \leq \delta$$

In dieser Menge sind demnach diejenigen Vektoren, deren Farbabstand für jede Komponente nicht größer als δ ist. Die Autoren schlagen $\delta = 2$ vor. Eine Klassifikation in Vorder- und Hintergrund kann nun durch die berechneten Wahrscheinlichkeiten erfolgen.
Um das Resultat der Klassifikation zu verbessern, schlagen die Autoren den Einsatz morphologischer Operatoren und ein Entfernen zu kleiner Vordergrundregionen vor. Eine Aktualsierung des Modells wird wie folgt für die alle N_2 *Features* der Tabelle, also auch für die Elemente die sich im Puffer befinden, durchgeführt :

$$p_b^{s,t+1} = (1 - \alpha_2 \cdot p_b^{s,t} + \alpha_2 \cdot M_b^{s,t})$$

$$p_v^{s,t+1,i} = (1 - \alpha_2) \cdot p_v^{s,t,i} + \alpha_2 \cdot M_v^{s,t,i}$$

$$p_{vb}^{2,t+1,i} = (1 - \alpha_2) \cdot p_{vb}^{s,t,i} + \alpha_2 \cdot (M_b^{s,t} \wedge M_v^{s,t,i})$$

Dabei ist α_2 eine Lernkonstante durch die sich die Aktualisierungsgeschwindigleit steuern lässt. Zudem ist $M_b^{s,t} = 1$ falls s zum Zeitpunkt t als Hintegrund eingestuft wurde, ansonsten gilt $M_b^{s,t} = 0$. $M_v^{s,t,i} = 1$ gilt nur für den Wert in der Tabelle, der v_t am Ähnlichsten ist, für die anderen gilt $M_v^{s,t,i} = 0$.
Konnte kein *Feature* aktualisiert werden, so konnte auch zu keinem eine Ähnlichkeit festgestellt werden. Dann wird derjenige ersetzt, der die geringste Wahrscheinlichkeit aufweist. Dieser befindet sich immer an Position N_2, also im Puffer, da die Einträge der Tabelle nach ihren Wahrscheinlichkeiten sortiert werden.
Steigt die Wahrscheinlichkeit eines *Features* der sich im Puffer befindet genügend stark an, so tauscht er seine Position mit dem am wenigsten Wahrscheinlichen *Feature* aus dem Hintergrundmodell. Die hierfür benötigte Mindestwahrscheinlichkeit wird durch einen Schwellwert T geregelt.

Die Laufzeit des Verfahrens liegt in $mathcalO(|Frames| \cdot N_2 \cdot |Pixel|)$ und der Speicherbedarf lässt sich durch $\mathcal{O}(N_2 \cdot |Pixel|)$ abschätzen. Die Autoren schlagen einen Wert von $N_2 = 50$ vor. Von diesen 50 *Features* sind sich viele untereinander ähnlich, so dass nicht 50 verschiedene Hintergrundmodelle geführt werden. Die Anzahl der sich nicht ähnlichen Teilmengen hängt von dem Hintergrund der Szene und den daraus folgenden Wahrscheinlichkeitsberechnungen ab.
Die in der folgenden Tabelle aufgeführten Parameter wurden in der durchgeführten Evaluation (siehe Kapitel 9) optimiert und verwendet.

- α_1: Steuert, wie schnell sich Hintergrundpixel aus dem Modell entfernen lassen
- α_2: Steuert, wie schnell das Modell an Szenenänderungen angepasst werden soll
- α_3: Steuert ebenfalls Anpassungsgeschwindigkeiten des Modells
- δ: Wird für die Ähnlichkeitstests verwendet
- T: Prozentwert der angibt, ab welcher Wahrscheinlichkeit neue *Features* in das Hintergrundmodell aufgenommen werden.

8 Post Processing

Dieses Kapitel widmet sich typischen und in der Praxis häufig eingesetzten *Post Processing* Verfahren, die auf den durch *Background Subtraction* Verfahren erzeugten Subtraktionsbildern arbeiten, vorgestellt. Der Begriff *Post Processing* stammt aus dem Englischen und bedeutet soviel wie Nachbearbeitung. Das Ziel der hier vorgestellten Verfahren ist demnach das Nachbearbeiten der Subtraktionsbilder in einer solchen Weise, dass anschließende Aufgaben bessere Ergebnisse erzielen oder dass diese Aufgaben einfacher beziehungsweise schneller mit diesen Bildern erledigt werden können.
Die angesprochenen Subtraktionsbilder beinhalten, abhängig von dem verwendeten *Background Subtraction* Verfahren sowie den in der überwachten Szene auftretenden Herausforderungen (siehe hierfür Kapitel 6), eine gewissen Anzahl falsch klassifizierter Pixel. Diese führen zu Problemen bei anschließenden Analyseaufgaben. Häufig sind beispielsweise viele kleine Regionen, teilweise auch nur einzelne Pixel die durch Rauschen verursacht werden können, in den Subtraktionsbildern erkennbar. Werden diese nicht aus den Subtraktionsbildern entfernt, kommt es zu Fehlklassifikationen der Pixel. Ein solcher Umstand wirkt sich negativ auf anschließenden Verfahren aus. Nachbearbeitungsverfahren versuchen daher, die Fehler in den Subtraktionsbildern zu reduzieren.
Ein weiteres Problem stellen die Objekte beziehungsweise Personen geworfenen Schatten in der Szene dar. Eine Vielzahl anschließender Methoden verwenden zur Erfüllung ihrer Aufgaben Heuristiken, wie beispielsweise Objektgrößen, um mit wenig Rechenaufwand möglichst gute Ergebnisse erzielen zu können. Ein geringer Rechenaufwand ist aufgrund der geforderten Echtzeitfähigkeit notwendig, eine exakte Berechnung erfordert in der Regel zu viel Zeit. Falls das verwendete Verfahren relativ anfällig gegenüber Schatten ist, das heißt dass diese in den Subtraktionsbildern nahezu komplett oder zumindest größtenteils als Vordergrund enthalten sind, so liefern die verwendeten Heuristiken in der Regel schlechte Ergebnisse. Eine weitere Klasse der hier vorgestellten Verfahren beschäftigt sich daher mit der Reduktion der in der Szene vorkommenden Schatten.

8.1 Verfahren zum Entfernen von Rauschen - Medianfilter

Als Rauschen wird in der Bildverarbeitung die Störung von Bildern hinsichtlich der Farb- beziehungsweise der Helligkeitswerte einzelner Pixel der Bilder, bezeichnet. Die Stärke des Rauschens kann durch das sogenannte Signal-Rausch-Verhältnis beschrieben werden. Dieses ist durch den Quotienten $\frac{Nutzsignal}{Rauschsignal}$ definiert. Bezogen auf die Bildverarbeitung bedeutet ein kleiner Wert ein Bild mit starken Störungen. Das Vermeiden von Rauschen bei der Erzeugung von Bildern ist unvermeidlich. Daher kann nur versucht werden, es so gering wie möglich zu halten. Ist der Farb-

beziehungsweise Helligkeitswert eines Pixels verfälscht, so besteht in Abhängigkeit von der Stärke dieser Verfälschung die Möglichkeit, dass der Pixel bei der Berechnung des Subtraktionsbildes falsch klassifiziert wird.
Wird ein Pixel der zu einem Vordergrundobjekt gehört, fälschlicherweise als Hintergrund klassifiziert, so werden seine Statistiken wie beispielsweise seine Pixelanzahl verfälscht. Da spätere Verfahren auf solchen Statistiken arbeiten, werden sie abhängig vom Grad der Verfälschung, schlechte Resultate erzielen. Wird ein Hintergrundpixel dagegen fälschlicherweise als Vordergrund klassifiziert, so entsteht im Subtraktionsbild unter Umständen eine Region aus nur einem Pixel. Anschließende *Tracking-* sowie Analyseverfahren arbeiten auf den detektierten Regionen im Subtraktionsbild. Regionen die durch Bildrauschen entstanden sind erhöhen den Arbeitsaufwand dieser Verfahren und führen im Extremfall zu falschen Analyseergebnissen.
Verfahren zur Rauschunterdrückung lassen sich auch generell direkt auf die zu bearbeitenden Videobilder anwenden. Dabei muss jedoch darauf geachtet werden, dass diese nicht zu stark arbeiten und dadurch die Resultate der *Background Subtraction* Verfahren verschlechtern. Hierfür wird häufig der sogenannte Gaußfilter verwendet.
Der Median-Filter kann speziell zur Nachbearbeitung, also zur Verbesserung von Subtraktionsbildern, eingesetzt werden. Angewendet auf einen Pixel eines Grauwertbildes berechnet er den Median aller auftretenden Grauwerte innerhalb der durch die Filtergröße definierten Umgebung des Pixels. Der Median ist definiert als der mittlere dieser Grauwerte, nachdem diese sortiert wurden. Die Filtergröße wird dabei in x- und y-Richtung durch jeweils eine ungerade Anzahl an Pixeln definiert. Dies hat den Vorteil, dass die Umgebung des Pixels aus einer ungeraden Anzahl an Pixeln besteht und daher die Berechnung eines Medians eindeutig erfolgen kann. Zudem existiert eine zentrale Pixelposition genau in der Mitte des Filters. Diese Position stimmt mit der des zu bearbeitenden Pixels überein, so dass dessen Umgebung in x- und y-Richtung die selben Ausmaße besitzt und daher symmetrisch ist. An den Rändern des Bildes müssen Spezialbehandlungen erfolgen.
Bei dem Medianfilter handelt es sich um einen sogenannten Rangordnungsfilter, da er zur Berechnung alle Pixelwerte in der durch ihn definierten Umgebung sortieren muss. Dadurch ist er verglichen mit anderen Filtern sehr rechenintensiv.
Da die Subtraktionsbilder keine Graustufen- sonder Binärbilder sind, also nur aus den Farben Schwarz und Weiß. Das Ergebnis des Medianfilters hängt somit direkt von der Anzahl der schwarzen beziehungsweise weißen Pixeln in der zu untersuchenden Umgebung ab. Ist mehr als die Hälfte schwarz, so wird die Farbe des Pixels auch auf schwarz gesetzt. Ist dagegen mehr als die Hälfte weiß, so wird auch der Pixel auf weiß gesetzt. Hier arbeitet der Filter deutlich schneller, da keine Sortierung benötigt wird.

8.2 Entfernen zu kleiner und zu großer Regionen

Nach dem Anwenden eines *Connected Components* Algorithmus (siehe Abschnitt 4.6) sind alle zusammenhängenden Regionen eines Subtraktionsbildes bekannt und besitzen eine eindeutige Identifizierungsnummer. Durch Fehler im Hintergrundmodell oder durch Rauschen in den Videobildern kommt es häufig zu einer Vielzahl klei-

ner Regionen im Subtraktionsbild die zu keinen Vordergrundobjekten gehören. Im Allgemeinen geht dem *Connected Components* Algorithmus noch ein Verfahren zur Rauschunterdrückung voraus. Dieses filtert zunächst die kleinen Regionen aus dem Bild. Jedoch müssen auch noch die etwas größeren Regionen, von denen man mit hoher Wahrscheinlichkeit davon ausgehen kann dass es sich bei ihnen um keine Vordergrundobjekte handeln wird, aus dem Subtraktionsbild entfernen, da sie den Rechenaufwand späterer Arbeitsschritte in der Regel stark in die Höhe treiben und die Qualität verringern.
Starke und plötzliche Beleuchtungsänderungen in der Szene führen ebenfalls häufig zu vielen falsch klassifizierten Pixeln. Dies liegt daran, dass sich die Farbwerte der Pixel schnell verändern, so dass die Ähnlichkeitstest zu den Werten des Hintergrundmodells keine genügend große Ähnlichkeit mehr aufweisen können und daher fälschlicherweise als Vordergrundpixel klassifiziert werden. Gegenüber dem vorherig beschrieben Problem sind die zusammenhängenden Regionen hier dagegen oft sehr groß, im Extremfall kann sogar eine Region das ganz Bild abdecken. Daher werden auch große Regionen aus dem Subtraktionsbild entfernt, da diese mit hoher Wahrscheinlichkeit kein einzelnes Vordergrundobjekt repräsentieren, eventuell dafür mehrere die sich nicht mehr unterscheiden lassen. Zwar werden durch das Entfernen solcher Regionen keine falschen Trajektorien erzeugt beziehungsweise aktualisiert, jedoch können die von richtigen Objekten in der Szene ebenfalls nicht korrekt erzeugt oder aktualisiert werden. Dieses Problem lässt sich anhand des Subtraktionsbildes nicht vermeiden. Daher sollten Verfahren verwendet werden, die möglichst wenig anfällig gegenüber Beleuchtungsänderungen sind. Das hier beschriebene Verfahren verhindert nur die fehlerhafte Aktualisierung der Trajektorien.
Der besagte *Connected Components* Algorithmus liefert die einzelnen Regionen des Subtraktionsbildes. Da diese sich durch ihre jeweilige Identifizerungsnummer eindeutig voneinander unterscheiden lassen, lässt sich die Anzahl dieser Regionen sowie deren jeweilige Pixelanzahl durch abzählen bestimmen, falls sie nicht schon von dem Algorithmus mitgeliefert werden. Es müssen zwei geeignete Schwellwerte T_{min} sowie T_{max} gewählt werden, die zur Aussortierung der mit hoher Wahrscheinlichkeit falsch identifizierter Regionen verwendet werden können. Das Aussortieren entfernt nun diejenigen Regionen, deren Pixelanzahl kleiner als T_{min} oder größer als T_{max}.
Sei R die Menge der in einem Subtraktionsbild aufgefundenen Regionen, $|R_i|$ die Anzahl der Pixel einer Region $R_i \in R$. Das Entfernen zu großer beziehungsweise zu kleiner Regionen führt demnach zu $\bar{R} = \{R_i \mid R_i \in R \wedge T_{min} \leq |R_i| \leq T_{max}\}$
Die Schwellwerte T_{min} und T_{max} bestimmen das Ergebnis dieser Methode maßgeblich. Problematisch an ihnen ist, dass sie sich nicht allgemeingültig, das heißt für jede Überwachungsszene gut geeignet, wählen lassen. Dies liegt an dem Umstand, dass Vordergrundobjekte nicht in jeder Szene, im Allgemeinen nicht einmal innerhalb einer Szene, mit der selben Größe zu sehen sind. Dies liegt zum einen an den verschiedenen möglichen Objektgrößen sowie an der Lage, speziell Abstand sowie relativer Winkel, dieser Objekte bezüglich der verwendeten Überwachungskamera. Die Wahl der Parameter muss daher nach einer individuellen Abschätzung der Objektgrößen getroffen werden.
Das hier vorgestellte *Post Processing* Verfahren arbeitet relativ schnell. Das ist wichtig, da es ja zur Verbesserung der Resultate eines Verfahrens eingesetzt werden soll, welches in Echtzeit arbeitet. Falls das *Connected Components* Verfahren die benötigten Regionenangaben nicht liefert, müssen diese erst bestimmt werden, was im schlechtesten Fall durch abarbeiten jedes einzelnen Pixels erfolgen kann. Demnach liegt

die benötigte Rechenzeit in $\mathcal{O}(|Pixels|)$. Sind die Größenangaben bekannt, kann das Entfernen in $\mathcal{O}(|Regions|)$ erfolgen.

8.3 Morphologische Operatoren

Morphologische Operatoren verwenden sogenannte Strukturelemente um die Umgebung eines Pixels zu beschreiben und innerhalb dieser gewisse Operationen durchzuführen. Da das *Post Processing* auf den durch die *Background Subtraction* erzeugten binären Subtraktionsbildern arbeitet, enthält das Strukturelement an jeder seiner Positionen eine 0 oder 1. Wie bei Filtermasken wird das Strukturelement sukzessive auf alle Pixel des Bildes gelegt. An den Rändern des zu bearbeitenden Bildes werden Spezialbehandlungen durchgeführt.
Ein solcher morphologischer Operator ist die *Erosion*. Dieser Begriff lässt sich aus dem lateinischen Wort für abnagen, nämlich *erodere*, herleiten. Stimmt die Umgebung eines weißen Pixels an jeder Position mit den Werten des auf diesen angesetzten Strukturelements überein, so gehört der Pixel zur erodierten Menge und bleibt weiß. Gehört er nicht zu der erodierten Menge, so stimmt der wert an mindestens einer Position nicht überein und der Pixel wird auf schwarz gesetzt. Pixel deren Umgebung nicht mit dem Strukturelement übereinstimmen werden somit aus dem Subtraktionsbild entfernt. Das Problem hierbei ist, dass die Größe des Vordergrundobjektes abnimmt.
Ein weiterer solcher Operator stellt die *Dilation* dar. Der Begriff *Dilation* kommt aus dem Lateinischen und bedeutet soviel wie ausdehnen. Auch hier wird das Strukturelement auf jeden Pixel angesetzt. Falls es sich um einen weißen Pixel handelt, werden die Werte seiner Umgebung auf die Werte des Strukturelementes an entsprechender Position gesetzt. Das ursprüngliche Bild wird somit auf die Form des Strukturelementes gedehnt. Die Größe des Vordergrundobjektes nimmt hier dagegen zu.
Durch die Kombination dieser morphologischen Operatoren kann ein neuer Operator erzeugt werden, der in der Nachbearbeitung der Subtraktionsbilder häufig zum Einsatz kommt. Das sogenannte *Closing* führt eine *Dilation* gefolgt von einer *Erosion* mit dem selben Strukturelement durch. Durch diese Kombination ist es in der Lage, kleine Löcher in den Vordergrundobjekten zu schließen. Zudem wird das Aussehen der häufig ausgefranst wirkenden Ränder verbessert. Durch die Fransen entstehen häufig auch eine Vielzahl kleiner Regionen, die eventuell aus nur einem Pixel bestehen und spätere Analysaufgaben bezüglich Rechenzeit sowie Qualität der Resultate negativ beeinflussen können. Das *Closing* bewirkt, dass diese kleine Regionen mit dem Vordergrundobjekt verschmelzen, zu dem sie eigentlich gehören und in dessen Nähe sie sich auch befinden.

8.4 Salienztest

Der Begriff *Salienz* stammt ursprünglich aus dem Lateinischen und bedeutet so viel wie Hervorspringen. Bezogen auf die Bildverarbeitung wird mit *Salienz* in der Regel beschrieben, wie sehr sich ein Wert (oder auf was auch immer sich die *Salienz* bezieht) hervorhebt beziehungsweise gegenüber anderen Werten abhebt. Häufig wird damit beschrieben, wie sehr man davon überzeugt ist, dass der besagt Wert bestimmte Eigenschaften erfüllt.

Das Ziel des als *Post Porcessing* Verfahrens eingesetzten Salienztests ist demnach, herauszufinden wie Wahrscheinlich eine im Subtraktionsbild detektierte Region auch tatsächlich zu einem Vordergrundobjekt gehört. Dies erreicht der Test dadurch, dass er für jeden Pixel einer Region bestimmt, wie stark er sich von dem Hintergrund unterscheidet. Für die Klassifikation eines Pixels als Vordergrund reicht ein negatives Ergebnis bei den Ähnlichkleitstest zu dem entsprechenden Modell des Hintergrunds an dieser Position, aus. Für einen Salienztest wird der Ähnlichkeitstest für jeden Pixel noch einmal mit verschärften Bedingungen durchgeführt. Beispielsweise klassifiziert das in Abschnitt 7.3 vorgestellte *Running Average* Verfahren die Pixel durch die Berechnung der Farbdistanz zwischen ihnen sowie dem Hintergrundmodell und dem Anschließenden Vergleich mit einem Schwellwert. Für den Salienztest wird die Farbdistanz der Vordergrundpixel zusätzlich mit einem doppelt so großen Schwellwert verglichen. Erreicht nicht mindestens ein Prozentsatz γ der Pixel einer Region dabei ein positives Resultat, so wird die Region aus dem Subtraktionsbild gelöscht, da sich nicht ausreichend viele Pixel genügend stark vom Hintergrund abheben, sondern viele seiner Pixel nur knapp den ursprünglichen Ähnlichkeitstest bestanden haben.
Laut Donovan Parks und Sidney Fels, die die Verbesserung der Subtraktionsbilder durch verwenden des Salienztests evaluiert haben, hängt die Verbesserung stark von dem gewählten Prozensatz γ ab. Dieser sollte nicht zu hoch sein. Für eine gute Wahl des Parameters kann dieses Verfahren einige fälschlicherweise detektierte Vordergrundregionen erkennen und entfernen. Auf Grund der schnellen Berechenbarkeit kann sich der Einsatz dieses Verfahrens somit auch durchaus lohnen. Der Zeitbedarf wird nicht wesentlich erhöht, da einzig die Vordergrundobjekte mit einem oder mehreren weiteren Werten (die Anzahl hängt von dem durchzuführenden Ähnlichkeitstest ab) verglichen werden müssen.

8.5 Verbesserung gegenüber Schatten

Scott Tattersall sowie Kenneth Dawson-Howe beschäftigten sich mit der Frage, wie sich die Anzahl der durch Schatten verursachten falsch klassifizierten Pixel, reduzieren lässt. Die durch Schatten verursachten Probleme waren Inhalt des Abschnitts 6.8 und können dort nachgelesen werden. Im Wesentlichen sind zwei Arten von auftretenden Problemen zu unterscheiden. Überlagern sich die Schatten von beispielsweise zwei Personen, so können diese im Subtraktionsbild nicht mehr unterschieden werden, da diese dort als eine zusammenhängende Region zu sehen sind. Lassen sich diese nicht mehr auseinander halten, so können ihre Bewegungsverläufe durch ein *Tracking* Verfahren auch nicht korrekt aktualisiert werden. Anschließende Analyseaufgaben, die auf fehlerhaften Trajektorien arbeiten müssen, können so unter Umständen falsche Schlüsse aus diesen ziehen. Zudem werden durch Schatten viele Pixel falsch klassifiziert. Verfahren die auf den Subtraktionsbildern arbeiten, verwenden oft Heuristiken um möglichst schnell sein zu können, da das System in Echtzeit arbeiten muss um sinnvoll eingesetzt werden zu können. Einige dieser Heuristiken verwenden die Größe, speziell die Anzahl der Pixel, der im Subtraktionsbild detektierten Objekte. Durch die falsch detektierten Pixel werden die Objektgrößen teilweise so stark verfälscht, dass ein korrektes Arbeiten der Heuristiken nicht mehr möglich ist. Ein solches Beispiel ist in Abbildung 6.6
Um die Anzahl der durch Schatten falsch klassifizierten Pixel zu reduzieren entwickel-

ten sie ein Methode, die für jeden Pixel entscheidet, ob er zu einem Schatten gehört oder nicht. Das entstehende Ergebnis ist eine Binärmaske, hier *Shadow Point Mask* genannt, die folgende Form besitzt :

$$SPM(x,y) \leftarrow \begin{cases} 1, & \text{falls P(x,y) zu einem Schatten gehört} \\ 0, & \text{sonst} \end{cases}$$

Wird ein Pixel an der Position (x,y) als Vordergrundpixel klassifiziert und gilt zusätzlich an dieser Stelle $SPM(x,y) = 1$, so liegt ein Vordergrundpixel der zu einem Schatten gehört vor. Dieser Pixel wird im Subtraktionsbild auf den Wert 0 gesetzt und ist demnach im Folgenden als Hintergrundpixel zu betrachten. Im Subtraktionsbild sind daher nach dem Anwenden des Verfahrens nur noch diejenigen Pixel weiß, die zu einem Vordergrund gehören aber nicht als Schatten klassifiziert wurden. Für das Subtraktionsbild S gilt demnach an der Position (x,y) :

$$S(x,y) \leftarrow \begin{cases} 1, & \text{falls } S(x,y) = 1 \wedge SPM(x,y) = 0 \\ 0, & \text{sonst} \end{cases}$$

Als Eingabe benötigt das Verfahren das aktuelle Hintergrundmodell sowie Videobild benötigt. Das Ergebnis ist die Maske SPM die das Subtraktionsbild S beeinflusst.
Die Berechnung der Maske SPM, verwendet den in Abschnitt 4.9 vorgestellten HSV-Farbraum. In diesem Abschnitt sind auch die für die Videobilder benötigte Formeln für die Transformation aus dem RGB-Farbraum in den HSV-Farbraum aufgeführt. Die Berechnung der Maske beruht auf folgenden von den Autoren bei Schatten festgestellten Eigenschaften beziehungsweise Effekten :

1. Die Sättigung eine durch Schatten verdeckten Pixels steigt nie wesentlich an, kann aber dagegen stark abfallen.
2. Intensität eines solchen Pixels verringert sich in der Regel, erhöht sich jedoch niemals.
3. Die Intensität fällt zwar häufig ab, aber nicht beliebig stark. Daher lässt sich der Abfall durch einen Maximalwert begrenzen. Tests der Entwickler haben gezeigt, dass sich dieser prozentual von der Intensität des Hintergrundbildes abhängt. Je größer diese Intensität ist, desto größer ist der Einfluss des Schattens.

Durch $F_t(x,y)$ seien im Folgenden die HSV-Werte des Pixels an der Stelle (x,y) im Videobild zum Zeitpunkt t beschrieben. Analog beschriebt $B_t(x,y)$ die Werte des Hintergrundmodells zum Zeitpunkt t. Der Wert der Maske SPM an der Stelle (x,y) bestimmt nun wie folgt :

$$SP(x,y) \leftarrow \begin{cases} 1, & \text{falls } (\lambda \leq \frac{F_t^V(x,y)}{B_t^V(x,y)} \leq 1) \ \wedge \ (\left|F_t^S - B_t^S(x,y)\right| \leq 30) \\ 0, & \text{sonst} \end{cases}$$

Der bei den Sättigungunterschieden verwendete Schwellwert von 30 wurde von den Autoren empirisch ermittelt. Durch den Parameter λ lässt sich dagegen regeln, wie stark ein Pixel die Intensität verändern darf, um noch als Schatten klassifiziert zu werden. Dieser Parameter wird für eine überwachte Szene dynamisch bestimmt und kann zu verschiedenen Zeitpunkte auch verschiedene Werte besitzen. Dieser Umstand

hängt von der Beschaffenheit der überwachten Szene ab. Ein konstanter Wert für jeden Zeitpunkt ist aufgrund der sich mit der Zeit häufig ändernder Beleuchtungsverhältnisse oft nicht sinnvoll. Jedoch wird λ nicht pixelweise bestimmt, so dass der Wert an einem beliebigen Zeitpunkt für jeden Pixel der Szene verwendet wird.
Das von den Autoren vorgeschlagene dynamische Berechnungsverfahren des Parameters λ hat folgende zwei Ziele :

1. Das Verfahren soll für jede beliebige Szene arbeiten können. Speziell soll kein Vorwissen über beispielsweise Beleuchtungsverhältnisse in der Szene von Nöten sein.
2. Das Verfahren muss in gewissem Sinne adaptiv sein, so dass eine Anpassung an sich über die Zeit ändernden Gegebenheiten erfolgen kann.

Zur Bestimmung eines möglichst optimalen Wertes wird analysiert, wie stark eine Erhöhung von λ die Anzahl der durch die Maske SPM klassifizierten Schattenpixel beeinflusst. Tests haben gezeigt, dass ein optimaler Wert innerhalb des Bereiches zu finden ist, in dem die Anzahl der Klassifikationsänderungen zum ersten Mal abnimmt. Für einen Wert von $\lambda = 0$ existieren keine als Schatten klassifizierten Pixel. Für größer werdendes λ nimmt zunächst auch die Anzahl der Pixelklassifikationen zu. In der Nähe des optimalen Wertes nimmt die Pixeländerung weniger stark zu, da hier nahezu alle Schattenpixel der Szene als solche klassifiziert wurden und Objekte noch nicht fälschlicherweise als Schatten eingestuft werden. Ab einer gewissen Überschreitung des optimalen Wertes steigt die Anzahl der Klassifikationsänderungen wieder deutlich an, da von hier an Objektpixel fälschlicherweise als Schatten detektiert werden. Der zu wählende Wert ist demnach innerhalb des ersten monoton fallenden Gebietes zu finden.
Zur Evaluation verwendeten die Autoren zwei Metriken, mit deren Hilfe sie die Genauigkeit ihres Verfahren messen und diese mit denen anderer Verfahren zur Identifikation von durch Schatten verursachten Vordergrundpixeln vergleichen konnten. Dazu nahmen sie die folgende Metrik, die sogenannte *Shadow Discrimination Accuracy* Metrik die bei Prati et al. [PCMT01] vorgestellt wurde :

$$\xi \leftarrow \frac{TP_f - FP_s}{TP_f + FN_f}$$

Dabei stehen TP, FP, FN für *True Positive*, *False Positive* sowie *False Negative* (siehe hierfür Abschnitt 9.1.1), die sich durch den Vergleich der Subtraktionsbilder mit entsprechenden *Ground Truth* Daten, berechnen lassen. Demnach berechnet diese Metrik den Quotienten aus der Anzahl der korrekt detektierten Vordergrundpixel verringert um die Anzahl der fälschlich detektierten Schattenpixel sowie der Gesamtanzahl an Vorderrgrundpixeln der Szene. Resultat ist ein Wert im Intervall $[0, 1]$ und kann demnach als Prozentwert, der angibt wie gut das Verfahren Vordergrundpixel erkennt und diese von Schatten unterscheiden kann. Ein Wert von 1 beziehungsweise 100 Prozent kann nur erreicht werden, falls $FP_s = 0$ und $FN_f = 0$ gilt, also weder Objektpixel als Schatten detektiert wurde, noch Vordergrundpixel nicht erkannt werden konnten. Ein Wert von 0 wird nur dann erreicht, wenn alle Vordergrundpixel als Schatten klassifiziert wurden.
Zudem wurde noch eine zweite Metrik verwendet, die wie folgt berechnet wird :

$$\varphi = \frac{TP_f - FN_s}{TP_f + FN_f}$$

Diese Metrik bestimmt wie genau ein Verfahren die Vordergrundpixel der Szene erkennen konnte. Der berechnete Wert verringert sich, wenn die Anzahl der nicht erkannten Vordergrundpixel oder die Anzahl der als Vordergrund detektierten Schattenpixel zunimmt.
Das Verfahren wurde bezüglich einer Sequenz für die ausreichend große *Ground Truth*Daten vorhanden waren, mit anderen Verfahren die in [PCMT01] bezüglich dieser Sequenz evaluiert wurden, verglichen. Die von Tattersall und Dawson-Howe vorgeschlagene Schattenmaske schnitt dabei mit einem Wert von $\xi = 0,9451$ am besten ab. Für die anderen Verfahren ergab sich $\xi \in [0,8602; 0,9232]$ und damit schlechtere Ergebnisse. Bezüglich der zweiten Metrik wurde nur die in diesem Abschnitt aufgeführte Schattenmaske SPM evaluiert. Hierbei wurde $\varphi = 0,9777$ für die selbe Szene erreicht.

9 Evaluation

In diesem Kapitel wird die im Rahmen dieser Diplomarbeit durchgeführte Evaluation aufgeführt. Ziel dieser Arbeit ist eine pixelgenaue Untersuchung bestehender *Background Subtraction* Verfahren um zu messen, wie gut diese unter typischen Problemen beziehungsweise Herausforderungen die in Überwachungsvideos häufig auftreten, arbeiten. Dazu werden die Parameter der Verfahren zunächst bezüglich eines Trainingsvideos, das mehrere dieser Probleme beinhaltet, optimiert. Danach werden bezüglich diesen optimierten Parameter die Probleme speziell untersucht. Dadurch lassen sich die Schwachstellen der Verfahren aufdecken, so dass gezielt nach Verbesserungsmöglichkeiten gesucht werden kann.
Bisher existieren solche exakten, pixelgenauen Evaluationen, die eine Vielzahl an Problemen untersuchen und auf einer ausreichend großen Überwachungsvideos arbeiten, nicht. Der Grund hierfür liegt in dem großen Aufwand bei der Erzeugung der für solche Untersuchungen benötigten *Ground Truth* Daten (siehe Abschnitt 4.3). Bisherige Evaluationen haben daher meist die Leistung auf Objektebene gemessen und dafür untersucht, wie gut die Verfahren Vordergrundobjekte in der Szene detektieren konnten. Wenn pixelgenau untersucht wurde, dann nur für einzelne Videobilder, für die die *Ground Truth* Daten per Hand erzeugt wurden.
In Abschnitt 5.2.2 wurde eine Arbeit vorgestellt, die ebenfalls eine pixelgenaue Evaluation von *Background Subtraction* Verfahren durchgeführt hat. Die dabei verwendeten Videos wurden an Orten im Freien aufgenommen, die keine Vordergrundobjekte beinhalteten. Diese wurden in einem Studio separat aufgenommen und anschließend in die Videos integriert. So konnte zwar eine ausreichend große Menge an *Ground Truth* Daten erstellt werden, jedoch konnten einige typische in der Praxis auftretenden Probleme von diesen nicht beinhaltet werden. Beispielsweise waren keine Schatten von Vordergrundobjekten, sowie andere durch Licht verursachte Effekte in ihnen enthalten.
Daher wurde hier ein anderer Ansatz zur Erzeugung der Überwachungsvideos gewählt. Die verwendete Szene wurde komplett in *Maya* erstellt. Sie beinhaltet Personen, Fahrzeuge, eine Baum, Ampeln, Reflektionen an Fensterscheiben, sowie ein realistisches Beleuchtungs- sowie Schattenverhalten. Genaueres zu diesen Videos, welche Probleme sie beinhalten und wie die Leistung der *Background Subtraction* Verfahren gemessen wird, sind Themen dieses Kapitels.
Das Kapitel ist wie folgt aufgebaut. In Abschnitt 9.1.1 werden verschiedene Evaluationsmetriken- und techniken, die zur Untersuchung der *Background Subtraction* Verfahren verwendet wurden, aufgeführt und erläutert. Diese werden um einige weitere ergänzt, die ebenfalls häufig eingesetzt werden. Im darauf Folgenden Abschnitt 9.2 werden die eingesetzten Überwachungsszenen sowie deren Besonderheiten beziehungsweise Schwierigkeiten vorgestellt. Die Durchführung der Evaluation sowie die dabei erzielten Resultate sind in Abschnitt 9.3 aufgeführt.

9.1 Evaluationsmetriken und -techniken

9.1.1 Metriken

Metriken stellen Kennzahlen dar, durch die Größen einer Messung angegeben werden können. In diesem Abschnitt werden diejenigen Metriken vorgestellt, die bei Evaluationen häufig zum Einsatz kommen. Zudem werden die verwendeten *ROC*- Kurven erläutert, die auf diesen Metriken basieren.

	Positive-System	Negative-System
Positive-GT	TP	FP
Negative-GT	FN	TN

Abbildung 9.1: Hier sind die für die Evaluation benötigten Basismetriken aufgeführt. Die Fehlklassifikationen sind rot visualisiert. Die Metriken, die korrekte Klassifikationen widerspiegeln dagegen grün.

Die zur Evaluation der *Background Subtraction* Verfahren eingesetzten Basismetriken sind in Abbildung 9.1 zu sehen. Bei ihnen handelt es sich um:

- *TP* : Die Metrik *True Positive*. Durch sie wird die Anzahl der korrekt detektierten Vordergrundpixel angegeben
- *TN* : Die Metrik *True Negative* gibt an, wie viele Hintergrundpixel der Szene korrekt detektiert werden konnten
- *FP* : Durch die Metrik *False Positive* wird die Anzahl der Pixel angegeben, die fälschlicherweise als Vordergrund erkannt wurden
- *FN* : Durch die Metrik *False Negative* wird dagegen die Anzahl der fälschlicherweise als Hintergrund detektierten Pixel angegeben

Aus diesen grundlegenden Basismetriken lassen sich durch geeignete Kombination weitere aussagekräftige Metriken erzeugen. Zu diesen gehören :

- *P* : Die Metrik *Positive* repräsentiert die Anzahl der Vordergrundpixel in einem Videobild. Sie setzt sich zusammen aus den korrekt aufgefundenen Vordergrundpixeln sowie den fälschlicherweise als Hintergrund detektierten. Damit gilt :

  $$P = TP + FN$$

- *N* : Durch die Metrik *Negative* wird die Anzahl der Hintergrundpixel eines Videobildes angegeben. Diese setzt sich aus der Anzahl der korrekt detektierten Hintergrundpixel sowie aus der Anzahl der fälschlicherweise als Vordergrund klassifizierten Pixel zusammen. Damit gilt :

  $$N = TN + FP$$

- *TPR* : Die *True Positive Rate* gibt das Verhältnis der korrekt detektierten Vordergrundpixel zu der Anzahl der in dem Videobild vorhandenen Vordergrundpixel an. Somit berechnet es sich zu:

  $$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$$

  Die *True Positive Rate* wird oft auch *Recall* genannt

- *FPR* : Die *False Positive Rate* berechnet das Verhältnis zwischen den fälschlicherweise als Vordergrund detektierten Pixeln und der Gesamtzahl der Hintergrundpixel in der Szene. Damit gilt :

  $$FPR = \frac{FP}{N} = \frac{FP}{TN + FP}$$

Ebenfalls häufig eingesetzte Metriken, die für die hier durchgeführte Evaluation jedoch nicht eingesetzt wurden, sind :

- *TDR* : Die *Tracker Detection Rate* berechnet das Verhältnis zwischen Anzahl der korrekt detektierten Vordergrundpixeln sowie der Anzahl der insgesamt untersuchten Videobildern. Damit gilt :

  $$TDR = \frac{TP}{|Videobilder|}$$

  Diese Metrik kann zum direkten Vergleich zweier *Background Subtraction* Verfahren verwendet werden.

- *FAR* : Die Fehlalarmrate (*False Alarm Rate*) bestimmt das Verhältnis der fälschlicherweise als Vordergrund detektierten Pixel zur Gesamtzahl der als Vordergrund klassifizierten Pixeln. daher berechnet sie sich zu :

  $$FAR = \frac{FP}{TP + FP}$$

- *ACC* : Die Metrik *Accuracy* setzt die korrekt klassifizierten Pixel ins Verhältnis aller Pixel. Somit gilt :

  $$ACC = \frac{TP + TN}{|Pixel|} = \frac{TP + TN}{TP + TN + FP + FN}$$

9.1.2 ROC- Kurven

Die *ROC*- Kurven (*Receiver Operating Characteristic*) stellen eine Möglichkeit dar, *Background Subtraction* Verfahren zu bewerten und deren optimale Parameter zu bestimmen. Dabei sind diejenigen Parameter optimal, die bei der Bewertung des Verfahrens zu dem besten Ergebnis geführt haben.
Die Einstellung der Parameter der Verfahren haben einen starken Einfluss auf die erbrachte Leistung. Wird beispielsweise ein für einen Ähnlichkeitstest gewählter Schwellwert zu hoch gewählt, so werden Vordergrundobjekte die sich farblich nicht wesentlich von dem Hintergrund der Szene abheben, nicht erkannt.

Die Parameter lassen sich in der Regel nicht so einstellen, dass die durchgeführte Klassifikation jeden Pixel korrekt den Kategorien Vordergrund beziehungsweise Hintergrund zuordnet. Eine optimale Parameterwahl besitzt das beste Verhältnis zwischen der Anzahl der korrekt und der fehlerhaft klassifizierten Pixel.
Zur Bestimmung einer *ROC*- Kurve werden zwei Metriken berechnet, *FPR* welche auf der x-Achse eines Koordinatensystems abgetragen wird sowie *TPR* das auf der y-Achse abgetragen wird. Die x- sowie die y-Achse haben beide eine Länge von einer Einheit, was einem Wert von 100 Prozent entspricht. Optimiert werden kann immer nur einer der Parameter. Die Metriken werden unter verschiedenen Werten des zu optimierenden Parameters berechnet, die anderen werden konstant gehalten. Wurde der optimale Wert dieses Parameters bestimmt, so wird dieser konstant gehalten und ein anderer optimiert.
Zur Berechnung des optimalen Wertes existieren zwei Vorgehen. Ein analytisches sowie ein graphisches. Das analytische Verfahren berechnet den euklidschen Abstand zwischen einem Punkt der Kurve und dem theoretischen Optimum mit den Koordinaten $(0, 1)$. Dieser Punkt kann nur dann erreicht werden, wenn $FP = FN = 0$ gilt, also keine fehlerhaften Klassifizierungen aufgetreten sind. Optimal ist nun derjenige Wert, der den geringsten Abstand zu dem Optimum besitzt. Graphisch lässt sich dieser Wert dadurch bestimmen, dass auf der Kurve nach dem Punkt gesucht hat, der eine Tangente Parallel zu der Geraden durch $(0, 0)$ und $(1, 1)$ besitzt. Sind beide Achsen gleich lang gewählt, so besitzt diese Tangente einen Steigungswinkel von $45°$.
In Abbildung 9.2 ist die graphische Auswertungsmethode aufgezeigt.

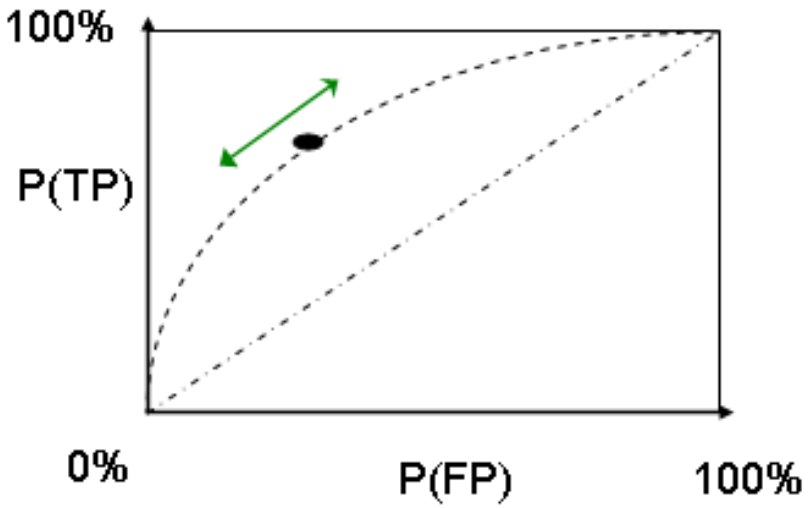

Abbildung 9.2: Hier ist die graphische Auswertung von *ROC*- Kurven dargestellt. Der optimale Punkt hat den geringsten Abstand zu dem theoretischen Optimum. Sind die Achsen des Schaubildes gleich lang, so besitzt der gesuchte Punkt eine Tangente mit einem Steigungswinkel von $45°$. (Quelle: [Wikc])

9.1.3 F-Maß

Der *F-Maß*, im Deutschen *F-Maß* genannt, beurteilt ebenfalls die Qualität eines Verfahrens, indem es zwei Metriken gegenseitig abwägt. Diese sind die Metriken *Precision* sowie *Recall*. *Precision* gibt an, wie genau ein Verfahren arbeitet. Sie ist definiert als :

$$Precision = \frac{TP}{TP + FP}$$

Es stellt demnach die Anzahl der korrekt als Vordergrund detektierten Pixel ins Verhältnis zu der Anzahl der als Vordergrund eingestuften Pixel. Je kleiner *FP*, desto größer ist die Genauigkeit.
Die Metrik *Recall* ist ein Maß, das die Trefferquote eines Verfahrens widerspiegelt. Sie ist definiert als :

$$Recall = \frac{TP}{TP + FN}$$

Je höher die Anzahl der fälschlicherweise als Hintergrund detektierten Pixel ist, desto kleiner wird die Trefferrate, da mehr Vordergrundpixel nicht als solche eingestuft werden konnten.
Die Kombination der beiden Metriken in Form eines harmonischen Mittels, führt zu dem besagten *F-Maß*. Es berechnet sich demnach zu :

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Alternativ hierzu lässt sich auch ein gewichtetes harmonisches Mittel verwenden, um entweder der Genauigkeit oder der Trefferquote einen größeren Einfluss zu geben. F berechnet sich dann bezüglich dem Gewicht α zu :

$$F_\alpha = \frac{(1 + \alpha) \cdot Precision \cdot Recall}{\alpha \cdot Precision + Recall}$$

Der Wert F_2 gewichtet beispielsweise die Trefferquote doppelt so stark wie die Genauigkeit. Bei $F_{\frac{1}{2}}$ ist das Gegenteil der Fall. Für die hier durchgeführte Evaluation wird eine gleichmäßige Gewichtung, also F_1 verwendet.

9.2 Überwachungsvideos

Zur Erzeugung der Überwachungsvideos wurde das Animationsprogramm *Maya* verwendet. In diesem Video ist eine typische Szene in einer Stadt zu sehen. Eines der Videobilder ist in Abbildung 9.3 aufgeführt. Zwei Teilausschnitte der Szene, die für spezielle Messungen verwendet wurden, sind durch die farbigen *Bounding Boxen* visualisiert. Die Szene besteht aus 700 Videobildern.

In dieser Szene sind verschiedene typische Herausforderungen enthalten, mit denen Überwachungssysteme, speziell *Background Subtraction* Verfahren in der Praxis zurecht kommen müssen. Zu ihnen gehören Veredeckungen, uninteressante Hintergrundbewegungen wie die eines Baumes dessen Blätter sich im Wind bewegen. Aber auch Beleuchtungsänderungen die durch das Umschalten von Ampeln entstehen. Zudem

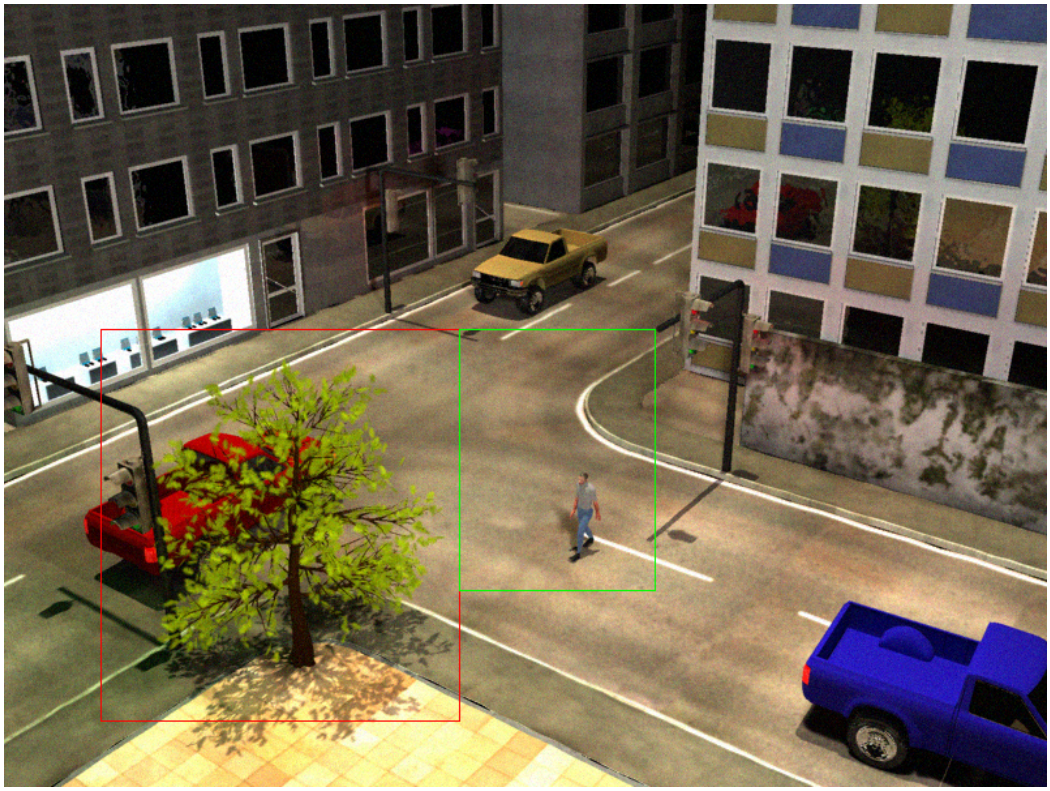

Abbildung 9.3: In dieser Abbildung ist ein Videobild der mittels Maya erzeugten Trainigssequenz zu sehen. Zwei speziell für die Evaluation gewählte Teilausschnitte sind durch das rote und grüne Rechteck hervorgehoben.

sind Reflektionen und Spiegelungen an Fensterscheiben enthalten. *Maya* erlaubt eine realistische Berechnung der Schatten innerhalb der Szene, so dass sich messen lässt, wie gut die Verfahren mit diesen zurecht kommen. Die Szene wurde zudem so erstellt, dass verschiedene Herausforderungen in ihr an- beziehungsweise wieder abgestellt werden konnten. So kann die Szene zur Evaluation verschiedener Probleme verwendet werden.

Durch die Rendertechnik *Vector* konnten sich die benötigten *Ground Truth* Daten die für die Berechnungen der Metriken gebraucht werden, erzeugen lassen. Ein solches *Ground Truth* Bild, ist in Abbildung 9.4 zu sehen.
Die folgenden Videos wurden für die durchgeführte Evaluation verwendet:

1. **Trainingsszene** : In dem Trainingsvideo, das zur Bestimmung der optimalen Parameter der Verfahren verwendet wird, sind viele der typischen Herausforderungen enthalten. Neben verschiedenen Bewegungsgeschwindigkeiten der Vordergrundobjekte sind hier Verdeckungen, Reflektionen, Hintergrundbewegungen, Schatten und stellenweise Beleuchtungsänderungen durch das Umschalten der Ampeln enthalten. Zudem tragen die Personen graue T-Shirts, die sich farblich wenig von der Straße unterscheiden, so dass die Szene durch Tarnungssituationen zusätzlich an Schwierigkeit zunimmt. Außerdem wurde mittels *Matlab* ein gaußsches Rauschen mit $\mu = 0$ und $\sigma^2 = 0,01$ hinzugefügt.

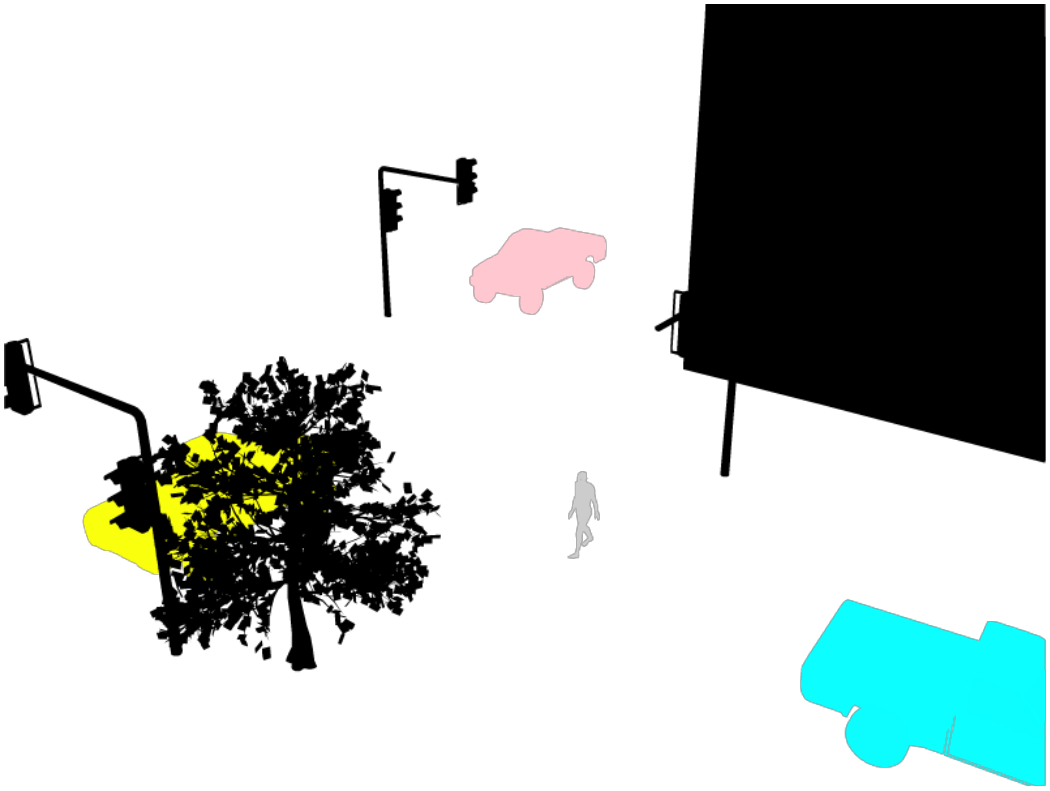

Abbildung 9.4: In dieser Abbildung ist ein *Ground Truth* Bild der mit Maya erstellten Überwachungsszene zu sehen. Weiße sowie schwarze Pixel repräsentieren den Hintergrund der Szene.

2. **Nachtszene** : Die Trainingsszene bei Nacht. Damit ist die Szene deutlich dunkler, Vordergrundobjekte lassen sich schwerer detektieren. Die durch die Ampeln auftretenden Beleuchtungsänderungen sind im Dunklen stärker ausgeprägt. Zudem wurde ein deutlich stärkeres Rauschen über die Szene gelegt, da Überwachungsvideos die im Dunkeln aufgenommen werden, ebenfalls oft stark verrauscht sind. Hierbei wurden $\mu = 0$ und $\sigma^2 = 0,1$ verwendet.
3. **Verdunklungsszene** : Die Trainingsszene ohne Rauschen. Dafür wird die Szene mit der Zeit dunkler, so dass sich die Verfahren an die Beleuchtungänderung anpassen müssen, um nicht zu viele Fehlklassifikationen zu erzeugen. Die Personen tragen blaue T-Shirts, so dass sie sich jetzt farblich deutlich von der Straße unterscheiden.
4. **Tarnungsszene** : Die Trainingsszene ohne Rauschen. So kann die Leistung der Verfahren bezüglich der Tarnung evaluiert werden, gerade im Bezug auf das Verdunklungsvideo, in dem die Personen blaue T-Shirts tragen.

In Abbildung 9.3 sind zur Evaluation verwendete Teilregionen zu sehen, in denen spezielle Probleme untersucht werden können. Der rot hervorgehobene Ausschnitt beinhaltet den kompletten Baum. Die in diesem Ausschnitt erzielten Resultate sind stark davon abhängig, wie gut das Verfahren mit dem Baum zurecht kommt. Der grün hervorgehobene Ausschnitt dagegen beinhaltet einen Teil der Straße. In ihm

lässt sich feststellen, wie gut die Verfahren unter den optimalen Parametern unter den Gesichtspunkten Tarnung und Schatten abschneiden.

9.3 Durchführung der Evaluation und Resultate

9.3.1 Parameteroptimierung - Training

Zur Parameteroptimierung der Verfahren werden die in Abschnitt 9.1.2 vorgestellten *ROC*- Kurven verwendet. Die Auswertung erfolgt hier analytisch und nicht graphisch. Das heißt, dass der Abstand eines Punktes auf der Kurve zu dem theoretischen Optimum an der Position $(0, 1)$ berechnet wird.
Besitzt ein Verfahren nur einen Parameter, so werden für ihn verschiedene Werte verwendet und bezüglich diesen die Metriken *FPR* sowie *TPR* und das optimale Paar gewählt. Besitzt das Verfahren mehrere einstellbare Parameter, so werden alle bis auf einen konstant gehalten und das optimale Paar für den veränderlichen Parameter berechnet. Danach wird dieser Parameter konstant gehalten, mit dem besten Wert, so dass sich ein anderer Parameter optimieren lässt. Das beste Resultat wird auf die zu testenden Probleme angewendet. Die Folgenden in Kapitel 7 vorgestellten *Background Subtraction* Verfahren wurden evaluiert :

1. *Codebook* Verfahren 7.9.3
2. *Mixture of Gaussian* Verfahren 7.8
3. Verfahren von Li et al. 7.10
4. Verfahren von McKenna et al. 7.6
5. *Median* Verfahren 7.5
6. *Running Average* Verfahren 7.3

9.3.2 Training

Die Optimierung wird bezüglich des im vorherigen Abschnitt beschriebenen Trainingsvideos durchgeführt. In der folgenden Aufzählung sind die erzielten Resultate zu sehen. Die Metriken *TPR*, *FPR* sowie die Distanz zum theoretischen Optimums werden zusätzlich aufgeführt.

Das Verfahren von McKenna et al. erzielte in der Trainingsphase das schlechteste Resultat. Die liegt zum einen daran, dass das Rauschen nicht gut gehandhabt werden konnte und zudem auch die Bewegungen des Baumes zu vielen fälschlicherweise als Vordergrund detektierten Pixeln führte.
Das *Codebook* Verfahren fiel ebenfalls durch viele fälschlicherweise als Vordergrund detektierten Pixel auf. Besonders stark waren dunkle Bereiche der Videobilder sowie die Region des Baumes betroffen. Die akzeptierten Helligkeitsvariationen sind gerade kurz nachdem ein neues Codewort erzeugt wurde, sehr gering, wobei dunkle Regionen besonders stark betroffen sind. Das hat zur Folge, dass sehr viele Codewörter erzeugt werden, bei denen der Ähnlichkeitstest auch bei nur geringen Helligkeitsunterschieden häufig negativ ausfällt.

$Verfahren$					
Li	$\alpha_1 = 0,01$	$\alpha_2 = 0,005$	$\alpha_3 = 0,05$	$\Delta = 2$	$thresh = 0,5$
MoG	$\alpha = 0,003$	$k = 2$	$\|gaussians\| = 5$	-	-
$Codebook$	$\delta = 25$	$\alpha = 0,7$	$\beta = 1,1$	-	-
$McKenna$	$\alpha = 0,001$	$k = 2$	$-$	-	-
$Median$	$thresh = 30$	-	-	-	-
$RunningAverage$	$\alpha = 0,005$	$thresh = 30$	$\|gaussians\| = 5$	-	-

Tabelle 9.1: In dieser Tabelle sind die optimalen Parameter der Verfahren bezüglich der Trainingssequenz aufgeführt. Diese werden für die weitere Evaluation verwendet.

Verfahren	FPR	TPR	Distanz
Li	$0,0639$	$0,9196$	$0,1025$
MoG	$0,0161$	$0,9117$	$0,0898$
$Codebook$	$0,0837$	$0,9417$	$0,1050$
$McKenna$	$0,1182$	$0,8454$	$0,194$
$Median$	$0,0452$	$0,9176$	$0,0,094$
$RunningAverage$	$0,0639$	$0,9196$	$0,1025$

Tabelle 9.2: In dieser Tabelle sind die von den Verfahren erzielten Resultate aufgeführt. Für das Training wurden die Metriken FPR und TPR verwendet. Zudem ist die Distanz zum theoretischen Optimum aufgelistet.

Das Verfahren von Li et al. zeigte die Besonderheit, dass es teilweise sehr wechselhafte Resultate erzielte. So konnte auf eine Sequenz in der sehr gut Ergebnisse erzielt wurden eine Sequenz folgen, in der sehr viele falsch klassifizierte Pixel vorhanden waren. Die anderen Verfahren unterlagen keinen so starken Schwankungen. In Kapitel 11.2 sind die erzielten *F-Maße* bezüglich der einzelnen Videobilder aufgeführt. Das geringe *F-Maß* bei Videobild 310 ist dort deutlich erkennbar. Zudem konnte das Verfahren für viele unterschiedliche Parametereinstellung nahezu identische Ergebnisse liefern, so dass das Verfahren wie von den Autoren bemerkt, nicht stark von den gewählten Parametern abhängig ist.

9.3.3 Die Nachtszene

Hier wurden die Verfahren bezüglich einer Szene bei Nacht evaluiert, wie sie in Abschnitt 9.2 beschrieben wurde. Die optimalen Parameter der Trainingsphase wurden hierfür verwendet. In der folgenden Tabelle sind die durchschnittlichen *F-Maß* Werte der Verfahren aufgeführt.

Verfahren	F-Maß
Codebook	0,0575
Mixture Of Gaussian	0,0467
Li	0,0472
McKenna	0,0588
Median	0,0624
Running Average	0,0629

Tabelle 9.3: Hier sind die durchschnittlichen F-Maße bezüglich der Nachtszene aufgeführt.

Die Resultate die die Verfahren hier erzielen konnten, sind negativ ausgefallen. Das verstärkte Rauschen bereitete besonders den Verfahren die bezüglich eines Schwellwertes klassifizieren besonders starke Probleme. Zudem heben sich die Vordergrundobjekte durch die Dunkelheit farblich weniger von dem Hintergrund der Szene ab, als dies in der Trainingsphase der Fall war. Das *Mixture of Gaussian* Verfahren, dass einen Pixel bezüglich der Varianz an seiner Position klassifiziert, kam als einziges Verfahren mit dem starken Rauschen zurecht. Jedoch bewirkten großen Varianzwerte, dass sehr viele Vordergrundpixel nicht als solche erkannt werden konnten, so dass auch hier nur ein sehr geringer durschnittlicher *F-Maß* erzielt werden konnte.
Die starke Verminderung der jeweiligen *F-Maß* führt zu folgenden Schlüssen :

- Verschlechtert sich die Bildqualität während einer Überwachung durch Bildrauschen merklich, so nimmt die Qualität der *Background Subtraction* Verfahren stark ab. Konstante Parametereinstellungen sind daher nicht zweckmäßig.
- Verfahren die bezüglich Varianzen klassifizieren, wie hier das *Mixture of Gaussian* Verfahren können zwar auch mit unterschiedlich starkem Rauschen zurecht kommen, haben dagegen starke Probleme wenn zu einem starken Rauschen auch Tarnungssituationen hinzukommen.
- Neue Verfahren oder Verbesserungen der hier vorgestellten werden benötigt, um bei einer Kombination aus starkem Rauschen und Tarnungssituationen gute Resultate erzielen zu können.

9.3.4 Verdunklungsszene

Diese Szene wurde in Abschnitt 9.2 vorgestellt. Die Helligkeit der Szene nimmt kontinuierlich ab, so dass es zu vielen fälschlicherweise als Vordergrund klassifizierten Pixeln kommt, falls die Verfahren mit den verwendeten Parametern nicht in der Lage sind, das Hintergrundmodell an diese Änderungen anzupassen.
In Tabelle 9.4 sind die durchschnittlich erzielten *F-Maße* der evaluierten Verfahren aufgeführt.

Die erzielten Werte in Abhängigkeit der einzelnen Videobilder sind in Kapitel 11.2 zu sehen.
Bei dieser Szene konnten sehr unterschiedliche Resultate erzielt werden. Während das

Verfahren	F-Maß
Codebook	0,2076
Mixture of Gaussian	0,5425
Li	0,5594
McKenna	0,2106
Median	0,4451
Running Average	0,1827

Tabelle 9.4: In dieser Tabelle sind die durchschnittlichen F-Maße der Verfahren bezüglich der Verdunklungsszene aufgeführt.

Verfahren von Li et al., das *Mixture of Gaussian* Verfahren und das *Median* Verfahren gute Resultate erzielen konnten, waren die Resultate der übrigen Verfahren deutlich schwächer.
Das *Codebook* Verfahren kommt mit der Beleuchtungsänderung nicht gut zurecht. Dies liegt an den Helligkeitsvariationen, die das Verfahren für die Durchführung der Ähnlichkeitstests verwendet. Diese fallen zu häufig negativ aus, so dass zu viele neue Codewörter erzeugt werden. Da ein neues Codewort in einem Puffer angelegt wird, repräsentiert es solange es sich in diesem befindet noch keinen gültigen Hintergrund. Daher kommt es zunächst zu vielen falschen Klassifikationen.
Das Verfahren von McKenna klassifiziert auffällig viele Vordergrundpixel nicht als solche. Da das Verfahren unimodal ist, kommt es in der Regionen des Baumes zu vielen fälschlicherweise als Vordergrund klassifizierten Pixeln.
Die verwendeten Parameter des *Running Average* Verfahrens führen ebenfalls zu vielen falschen Klassifikationen. Damit können die gewählten Parameter auch hier nicht für unterschiedliche Szenen eingesetzt werden.
Die erzielten Resultate führen zu folgenden Schlüssen :

- Nicht jedes Verfahren kann für unterschiedliche Szenen mit den selben Parametern verwendet werden.
- Das *Mixture of Gaussian* Verfahren, das Verfahren von Li et al. sowie das *Median* Verfahren liefern gute Resultate.
- Das *Codebook* Verfahren erzeugt zu viele Codewörter, wodurch es zu vielen fälschlicherweise positiv klassifizierten Pixeln kommt.

9.3.5 Tarnungsszene

Für die Tarnungsszene wurde die Trainingsszene verwendet, jedoch ohne Rauschen. Damit ist im Gegensatz zur Verdunklungsszene die Beleuchtung konstant. Zudem tragen die Personen graue T-Shirts, die sich farblich nur gering von der Straße unterscheiden.
Die in Tabelle 9.5 aufgeführten durchschnittlichen *F-Maßes* konnten von den evaluierten Verfahren erzielt werden. Die erzielten Werte in Abhängigkeit der einzelnen Videobilder sind in Kapitel 11.2 zu sehen.
Sortiert man die Resultate der Größe nach, so ergibt sich die selbe Reihenfolge wie

Verfahren	F-Maß
Codebook	0,360
Mixture of Gaussian	0,5638
Li	0,6640
McKenna	0,3899
Median	0,4254
Running Average	0,3371

Tabelle 9.5: In dieser Tabelle sind die durchschnittlichen F-Maße der Verfahren bezüglich der Tarnungsszene aufgeführt

in der Verdunklungsszene. Jedoch nahmen die Werte alle deutlich ab, da viele Vordergrundpixel fälschlicherweise als Hintergrund klassifiziert wurden. Dies ist auch der Grund, weshalb der *F-Maß* des *Median* Verfahrens besonders stark abgenommen hat. Der verwendete Schwellwert ist in der Lage, Rauschen zu unterdrücken. Jedoch kommt es so bei Tarnungssituationen zu vielen falschen Klassifikationen. Die Tarnung wird in dem nächsten Abschnitt genauer betrachtet.

9.3.6 Tarnung

Um zu messen, wie gut die Verfahren bezüglich der Herausforderung *Tarnung* arbeiten, wurden die Verdunklungsszene sowie die Tarnungsszene verwendet. Jedoch wurde ein Ausschnitt der Videos gewählt, der über die ganze Zeit Vorderrgrundobjekte zu sehen sind. Zusätzlich wurden die Videobilder der Sequenz von Bild 250 bis Bild 450 betrachtet, da sich hier nur die Personen in diesen Ausschnitten bewegen, die Fahrzeuge in diesen somit nicht vorkommen und damit die Statistiken nicht beeinträchtigen.
In Tabelle 9.6 sind die durchschnittlichen *F-Maße* der Verfahren aufgeführt.

Verfahren	Tarnung, lang	Dunkel, lang	Tarnung, kurz	Dunkel, kurz
Codebook	0,6953	0,6817	0,7122	0,7799
Mixture Of Gaussian	0,7105	0,7128	0,7150	0,7525
Li	0,6946	0,6907	0,7340	0,6839
McKenna	0,6606	0,6766	0,6651	0,6979
Median	0,6538	0,7056	0,6826	0,7649
Running Average	0,6640	0,7040	0,7002	0,7198

Tabelle 9.6: In dieser Tabelle sind die durchschnittlichen F-Maße bezüglich der Betrachtung des Tarnungsproblems aufgeführt

Die dritte und vierte Spalte der Tabelle werden zur Messung der Tarnung besonders betrachtet, da sie aus der Teilsequenz resultieren, die nur die Personen beinhalten.

Auffällig an diesen Resultaten ist, dass das Verfahren von Li in der Verdunklungsszene als einziges Verfahren schlechter als in der Tarnungsszene abgeschnitten hat. Eine genauere Untersuchung ergab, dass dies an den teilweise sehr wechselhaften Resultaten liegt, die dieses Verfahren erbringt. In der Trainingsphase 9.3.1 konnten ähnliche Effekte beobachtet werden. Rauschen sowie Beleuchtungsänderungen scheinen das Verfahren kurzzeitig stark zu beeinträchtigen.
Wie die Ergebnisse der kompletten Szenen vermuten ließen, bestätigt hier die genaue Betrachtung der Teilszenen, dass das *Codebook* Verfahren und das *Median* Verfahren mit ihren Parametern besonders starke Probleme bei Tarnungssituationen haben. Bei diesen beiden Verfahren ist der Unterschied des *F-Maßes* besonders groß.

9.3.7 Hintergrundbewegungen

Um zu messen wie gut die Verfahren bezüglich uninteressanten Hintergrundbewegungen abschneiden, wurde ein Ausschnitt des Tarnungsvideos gewählt, welches den kompletten Baum umschließt. Vordergrundobjekte bewegen sich hinter dem Baum, so dass diese von ihm teilweise verdeckt werden. Je besser der Baum in das Hintergrundmodell eingearbeitet wurde, desto höher ist der Wert der Metrik *Precision* da die Anzahl der fälschlicherweise positiv klassifizierten Pixel geringer ist. In der folgenden Tabelle sind die durchschnittlich erzielten *F-Maß* Werte der Verfahren aufgeführt.

Verfahren	F-Maß
Codebook	0,1666
Mixture of Gaussian	0,3249
Li	0,4176
McKenna	0,200
Median	0,2133
Running Average	0,1532

Tabelle 9.7: Hier sind die durchschnittlichen F-Maße der Verfahren bezüglich der Baumszene aufgeführt

Die erzielten Werte in Abhängigkeit der einzelnen Videobilder sind in Kapitel 11.3 zu sehen. Aus diesen Resultaten folgt, dass das *Codebook* Verfahren obwohl es multimodal ist, starke Probleme bei den uninteressanten Bewegungen die durch den Baum hervorgerufen werden, aufweist. Der Grund hierfür liegt auch hier bei den durch Helligkeitsvariationen hervorgerufenen Codeworterzeugungen, die hier besonders hoch sind. Neu angelegt Codewörter landen zunächst in einem Puffer, so dass sie keinen gültigen Hintergrund repräsentieren. Erst wenn genügend oft eine Aktualisierung stattgefunden hat, wird ein Codewort in das Modell aufgenommen. Eine Aktualisierung findet jedoch selten statt, da der Ähnlichkeitstest bezüglich der Helligkeit zu häufig negativ ausfällt. Die Einstellung der Parameter α und β hängen somit stark von der betrachteten Szene ab. Auch die Anzahl der Aktualisierungen bis ein Codewort den Hintergrund repräsentiert sowie die Anzahl der Videobilder die ein

Hintergrund nicht aktualisiert werden darf, damit er aus dem Modell entfernt wird, scheinen großen Einfluss auf die Leistung des Verfahrens zu haben. Für beide Werte wurden hier $t = 50$ gewählt. Eventuell sind andere Werte speziell für diese Videos besser geeignet, sind somit aber szenenabhängig.
Mit Ausnahme des *Codebook* Verfahrens zeigt die Tabelle, dass die multimodalen Hintergrundmodelle besser mit den uninteressanten Hintergrundbewegungen des Baumes zurechtkommen als die unimodalen Modelle.

9.3.8 Schatten

Um die Anzahl der Pixel zu bestimmen, die ein Verfahren auf Grund von Schatten fälschlicherweise als Vordergrund klassifiziert, wurden zunächst Schattenmasken der Videobilder berechnet. Das sind Binärbilder die an denjenigen Positionen den Wert 1 besitzen, die zu einem Schattenpixel gehören. Dazu wurde die Tarnungsszene ein zweites mal von *Maya* berechnet, dieses mal jedoch ohne Schatten. Mittels des euklidschen Abstands wurde ein Differenzbild erzeugt, welches pixelweise den Unterschied zwischen dieser Szene und der Tarnungsszene beinhaltet. Mit Hilfe der *Ground Truth* Daten wurden diejenigen Pixel aus dem Differenzbild entfernt die zu Vordergrundobjekten gehörten. Da es bei *Maya* besonders an den Rändern der Vordergrundobjekte zu Interpolationsfehlern kommt, wurde eine morphologische Dilatation auf das *Ground Truth* Bild angewendet, so dass diese Ränder entfernt werden konnten. Bei diesem Vorgehen werden zwar auch einige Schattenpixel entfernt, jedoch wurde die entstandene Maske auf alle Verfahren angewendet, so dass die erzielten Resultate vergleichbar sind.
Für jedes Verfahren wurde ein Histogramm bezüglich der Fehlklassifikationen in Abhängigkeit des durch den Schatten hervorgerufenen Intensitätsunterschiedes berechnet. Diese Histogramme sind in Abschnitt 11.1 zu sehen.
Auffällig hierbei ist, dass sich diese Histogramme nicht signifikant unterscheiden. Die Fehler im unteren Bereich der Distanzwerte resultieren ebenfalls aus Ungenauigkeiten beziehungsweise Interpolationsfehlern von *Maya*. Hinzu kommt, dass aufgrund der Beleuchtungsberechnung auch eine minimal unterschiedliche Farbabweichung vorliegen kann. Daher sollte das Histogramm erst ab einem bestimmten Farbabstand betrachtet werden (eine Distanz von 7 kann hier verwendet werden). Verfahren die viele Pixel fälschlicherweise als Vordergrund klassifizieren, haben bei niedrigen Distanzen besonders große Histogrammeinträge.
Die starken Schatten der Szene konnten von den Verfahren nicht als Hintergrund klassifiziert werden, was sich in einem starken Anstieg der Fehlklassifikationen ab einer Distanz von 70 bemerkbar macht.
Die Verfahren konnten jedoch nur eine geringe Anzahl an Schattenpixel als Hintergrund klassifizieren, die Anzahl der Fehler ist hoch. Dies gilt auch für Verfahren die laut Autoren weniger Probleme bezüglich Schatten haben. Daraus folgt, dass zumindest nicht für jede Parametereinstellung die Anzahl der Fehler bei Schatten gering ist.

9.3.9 Fazit

In der folgenden Tabelle werden die evaluierten *Background Subtraction* Verfahren nach ihrer durchschnitllich erreichten Platzierung sortiert. Grundlage sind die in den vorherigen Abschnitten aufgeführten *F-Maße* bezüglich den untersuchten Problemen.

Platzierung	Verfahren	Durchschnittliche Platzierung
1	*Mixture of Gaussian*	2,25
2	*Li*	2,75
3	*Median*	2,875
4	*Codebook*	3,375
5	*Running Average*	4,25
6	*McKenna*	4,625

Tabelle 9.8: In dieser Tabelle ist die durchschnittlich erreichte Platzierung der evaluierten Verfahren zu sehen.

Die Reihenfolge der durchschnittlichen Platzierung ändert sich nicht, wenn die Nachtszene, bezüglich der die Verfahren alle kein positives Resultat erzielen konnten, nicht mit einbezogen wird. Die besten Leistungen konnten mit dem *Mixture of Gaussian* Verfahren erzielt werden. Es ist verglichen mit den anderen Verfahren relativ unempfindlich gegenüber Rauschen und kann auch uninteressante Hintergrundbewegungen in sein Hintergrundmodell aufnehmen.
Das Verfahren von Li et al. kann ebenfalls gute Resultate erzielen, jedoch kommt es bei diesem Verfahren kurzzeitig zu vielen Fehlklassifikationen, besonders bei Bildrauschen und Beleuchtungsänderungen, so dass die durchschnittlichen *F-Maße* dadurch geringer ausgefallen sind. Dieser Effekt ist bei Videobild 310 der in Abschnitt 11.2 aufgeführten Diagramme zu sehen.
Die ersten beiden Verfahren konnten gute Resultate bei konstanten Parametern erzielen, die anderen Verfahren nicht. Es ist von Vorteil, wenn die Parametereinstellung nicht stark von der Szene abhängt, sondern für viele unterschiedliche Szenen verwendet werden kann. Ausnahme hiervon ist die Nachtszene.
Bezüglich der Nachtszene haben alle Verfahren schlecht abgeschnitten. In der Dunkelheit heben sich die Vordergrundobjekte farblich nicht so stark ab, wie bei Tageslicht. Es kommt zu tarnungsähnlichen Situationen. Die Kombination aus Tarnung und starkem Rauschen ist ein Problem, bei dem keines der evaluierten Verfahren gut abschneiden konnte.
Verschieden starkes Rauschen bereitet den Verfahren die bezüglich Schwellwerten klassifizieren starke Probleme. Die bezüglich eines Videos optimierten Parameter sind im Allgemeinen auch nur bezüglich der dort auftretenden Stärke des Rauschens optimal.
Hieraus folgt, dass die Leistung der Verfahren besonders stark von seinen Parametereinstellungen abhängig ist. Das Verfahren von Li et al. stellt hierbei eine Ausnahme dar. Es erzielt für sehr unterschiedliche Einstellungen sehr ähnliche Resultate. Erzielt es aufgrund von Szenenänderungen schlechtere Resultate, kann somit keine Verbesse-

rung durch die Wahl anderer Parametereinstellungen erfolgen.
Interessant ist, dass das *Codebook* Verfahren als multimodales Modell schlechter als das *Median* Verfahren abgeschnitten hat, welches nur ein unimodales Hintergrundmodell besitzt. Das *Codebook Verfahren* erzeugt während der Laufzeit besonders bei Bildrauschen und bei uninteressanten Hintergrundbewegungen zu viele neue Codewörter, so dass viele Pixel fälschlicherweise als Vordergrund klassifiziert werden.
Das *Running Average* Verfahren und das Verfahren von McKenna et al. schnitten bei der Evaluation am schlechtesten ab. Beide Verfahren erzeugen eine große Anzahl an fälschlicherweise als Vordergrund klassifizierten Pixeln, so dass nur geringe *F-Maß* Werte erreicht werden konnten.
Das Verfahren von McKenna et al. sowie das *Median* Verfahren haben Probleme bei Tarnungssituationen. Ihre jeweiligen *F-Maße* haben gegenüber der Szene in der keine Tarnungssitaution vorhanden war, deutlich abgenommen.
Die Fehlklassifikationen die durch Schatten verursacht wurden, waren bei allen Verfahren relativ gleich stark und hoch.

10 Ausblick

Verschiedene *Background Subtraction* Verfahren wurden für diese Diplomarbeit pixelgenau evaluiert. Dabei hat sich herausgestellt, dass die Leistung der meisten Verfahren stark von den gewählten Parametern abhängig ist. Keine Einstellung konnte für alle betrachteten Probleme konstant gute Resultate erzielen. Die Leistung aller Verfahren bei der eingesetzten Nachtszene war negativ. Die Kombination aus tarnungsähnlichen Verhältnissen und starkem Bildrauschen bereitete allen Verfahren starke Probleme. Die bezüglich Schwellwerten klassifizierenden Verfahren kommen bei verschieden starkem Rauschen nicht gut zurecht. Tritt ein stärkeres Rauschen in der Szene auf als in der für die Optimierung verwendeten Szene, so sinkt die Qualität der Verfahren stark ab. Das *Mixture of Gaussian* Verfahren kommt mit unterschiedlichen Stärken des Rauschens zurecht. Jedoch klassifiziert es viele Pixel von Vordergrundobjekten als Hintergrund, wenn diese sich farblich nicht stark von dem Hintergrund der Szene abheben. Insgesamt ist hier noch Forschungsbedarf nötig, da ein situationsbezogenes Einstellen der Parameter nicht zweckmäßig ist.
Für Parametereinstellungen der Verfahren bezüglich komplexen Szenen die viele Herausforderungen beinhalten, konnte keines der Verfahren besonders viele Schattenpixel als Hintergrund klassifizieren. Erweiterungen der Verfahren oder die Anwendung von Nachbearbeitungsverfahren könnten sich hierbei als vorteilhaft erweisen.
Daher könnte eine Erweiterung der hier durchgeführten Evaluation durch weitere Verfahren oder modifikationen der hier verwendeten Verfahren, erfolgen. Besonders für das *Mixture of Gaussian* Verfahren, welches hier besonders gut abschneiden konnte, existieren viele Erweiterungen.
Ebenso wäre es interessant zu untersuchen, ob die Leistung der Verfahren bei der Nachtszene durch den Einsatz von Nachbearbeitungsverfahren verbessert werden kann und wie stark eine eventuelle Verbesserung durch solche Verfahren ausfallen würde.

11 Anhang

11.1 Durch Schatten fehlerhaft klassifizierte Pixel

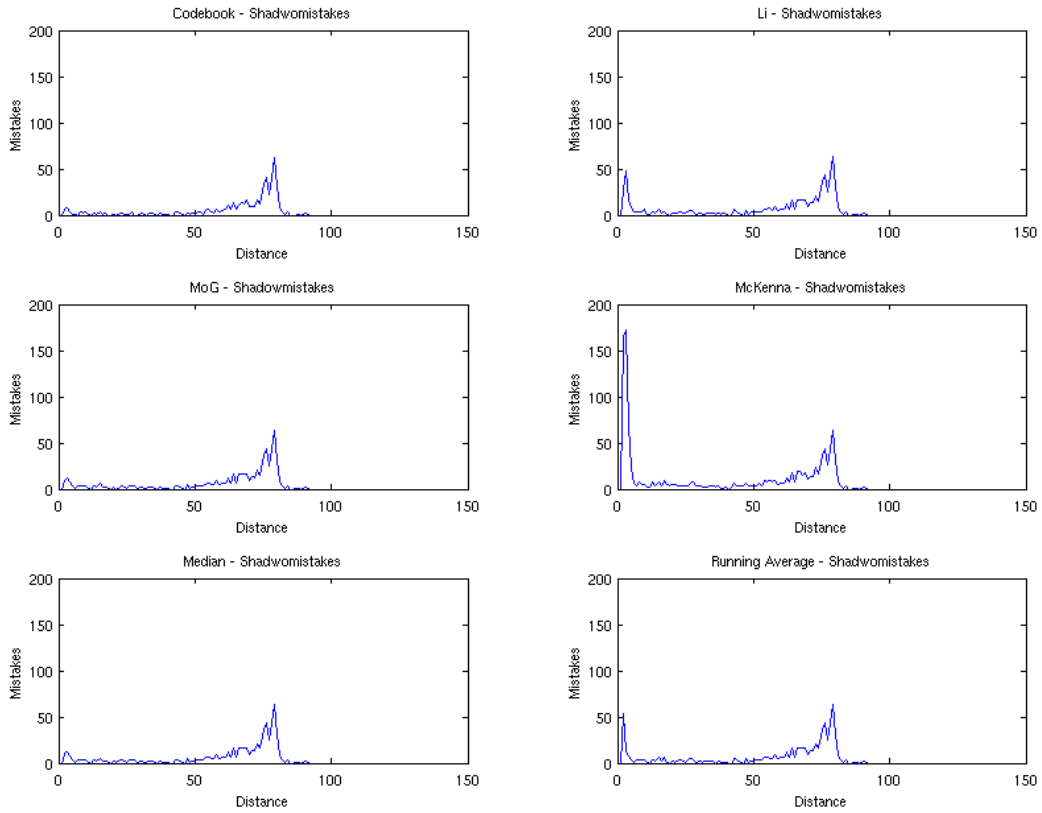

Abbildung 11.1: In diesen Abbildungen sind Histogramme der falsch klassifizierten Schattenpixel in Abhängigkeit der Schattenstärke zu sehen.

11.2 Evaluation der Verdunklungs- und Tarnungsszene

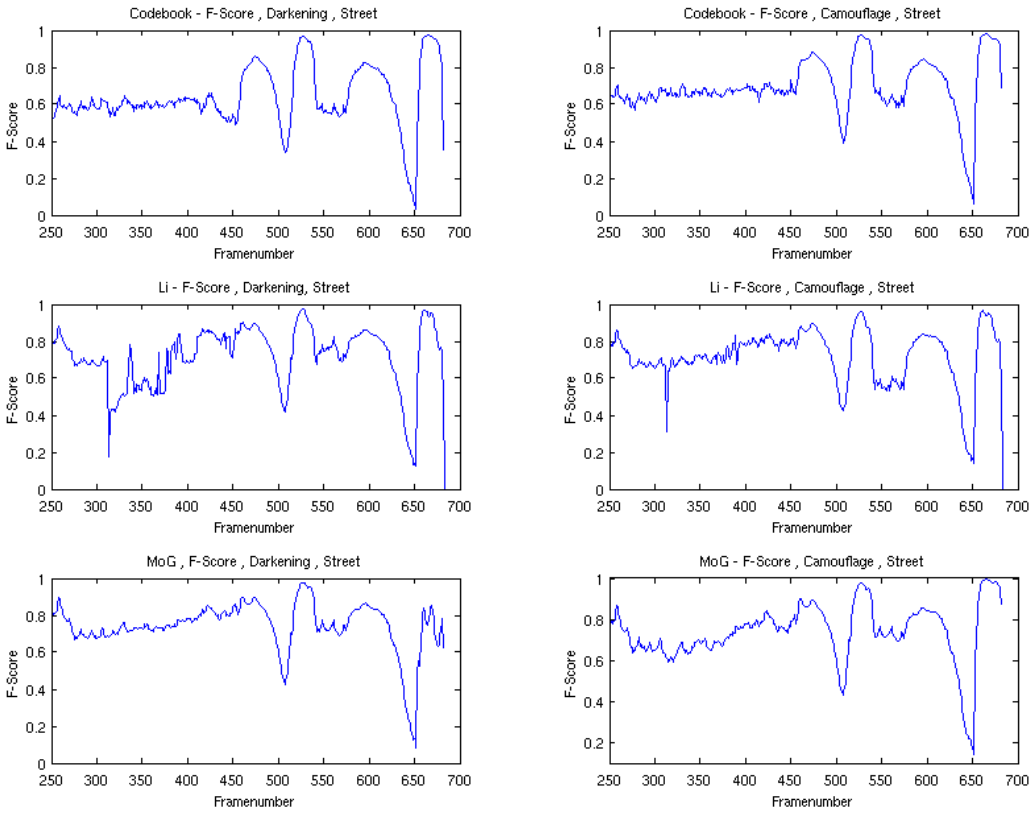

Abbildung 11.2: Hier sind die erreichten F-Maße der multimodalen Verfahren bezüglich der Verdunklungs- sowie der Tarnungsszene zu sehen. In der Verdunklungsszene existieren keine Tarnungssituationen.

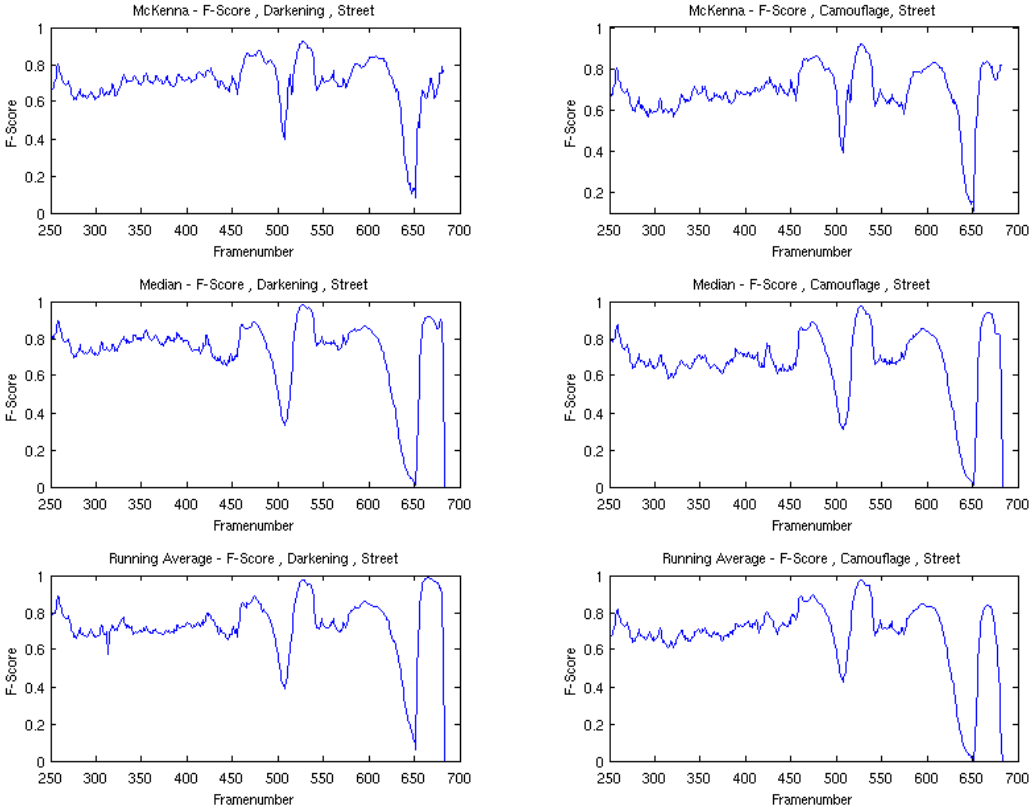

Abbildung 11.3: Hier sind die erreichten F-Maße der unimodalen Verfahren bezüglich der Verdunklungs- sowie der Tarnungsszene zu sehen. In der Verdunklungsszene existieren keine Tarnungssituationen.

11.3 Baum

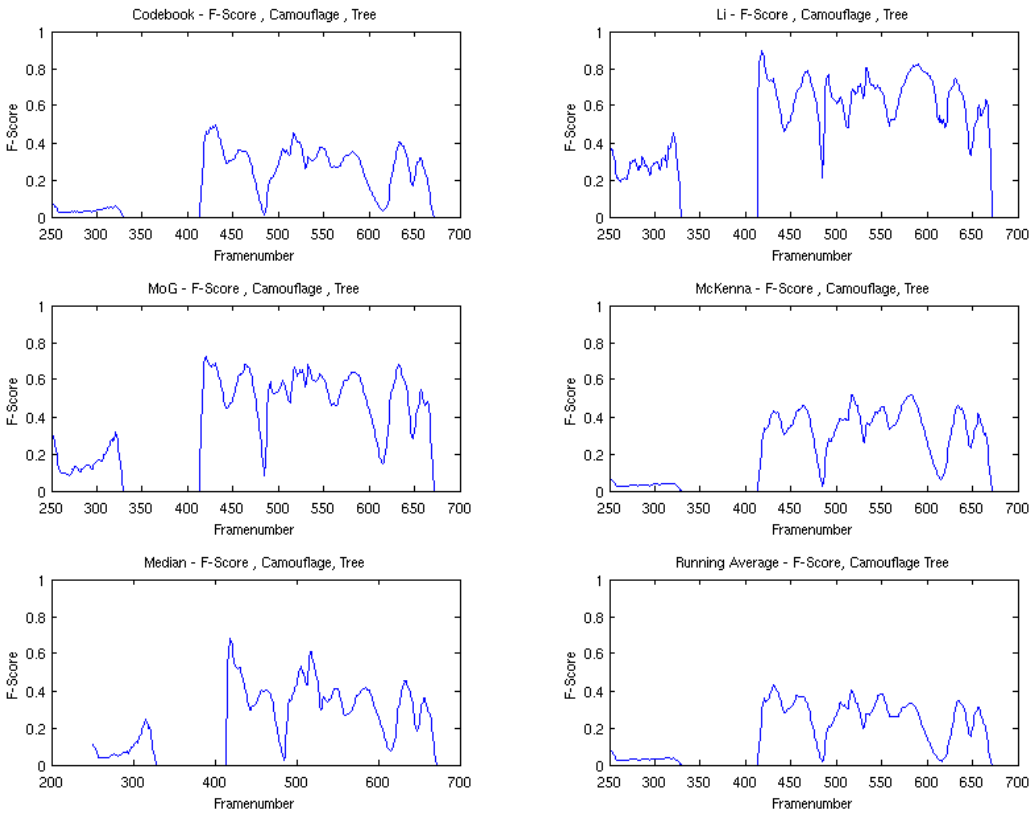

Abbildung 11.4: Hier sind die erreichten F-Maße der Verfahren bezüglich der Baumszene aufgeführt

Literaturverzeichnis

[BMGE01] Boult, T. E. ; Micheals, R. J. ; Gao, X. ; Eckmann, M.: Into the Woods: Visual Surveillance of Noncooperative and camouflaged Targets in Complex Outdoor Settings. In: *Proceedings of the IEEE* 89 (2001), October, Nr. 10, S. 1382–1402 (Zitiert auf Seite 31)

[CE02] Cavallaro, A. ; Ebrahimi, T.: Accurate video object segmentation through change detection. In: *Proc. of IEEE International Conference on on Multimedia and Expo*, 2002 (Lecture Notes in Computer Science) (Zitiert auf Seite 35)

[CGPP03] Cucchiara, Rita ; Grana, Costantino ; Piccardi, Massimo ; Prati, Andrea: Detecting Moving Objects, Ghosts, and Shadows in Video Streams. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (2003), S. 1337–1342 (Zitiert auf den Seiten 32 und 36)

[Cha03] Chalidabhongse, Thanarat H.: A perturbation method for evaluating background subtraction algorithms. In: *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, 2003 (Zitiert auf Seite 39)

[Ebr05] Ebrahimi, A. Cavallaro ; O. Steiger; T.: Semantic Video Analysis for Adaptive Content Delivery and Automatic Description. In: *IEEE Transactions on Circuits and Systems for Video Technology* Bd. 15, 2005, S. 1200–1209 (Zitiert auf Seite 33)

[EF03] Ewerth, Ralph ; Freisleben, Bernd: Frame Difference Normalization: An Approach to Reduce Error Rates of Cut Detection Algorithms for MPEG Videos. In: *In Proc. of IEEE International Conference on Image Processing*, 2003, S. 1009–1012 (Zitiert auf Seite 33)

[EHD00] Elgammal, Ahmed ; Harwood, David ; Davis, Larry: Non-parametric model for background subtraction. In: *FRAME-RATE Workshop, IEEE*, 2000, S. 751–767 (Zitiert auf den Seiten 30, 33 und 39)

[FM] François, Re R. J. ; Medion, Gérard G.: *Adaptive Color Background Modeling for Real-Time Segmentation of Video Streams* * (Zitiert auf Seite 35)

[HCD04] Han, Bohyung ; Comaniciu, Dorin ; Davis, Larry S.: *Sequential kernel density approximation through mode propagation: applications to background modeling*. 2004 (Zitiert auf den Seiten 30 und 35)

[HHD98] Haritaoglu, Ismail ; Harwood, David ; David, Lary S.: W^4S: A Real-Time System for Detecting And Tracking People in $2\frac{1}{2}$ D. In: *In Proc. 5th European Conf. Computer Vision*, Springer Verlag, 1998, S. 877–892 (Zitiert auf Seite 31)

[HHD99] Horprasert, Thanarat ; Harwood, David ; Davis, Larry S.: A statistical approach for real-time robust background subtraction and shadow detection. In: *ICCV Frame-Rate WS*, 1999, S. 1–19 (Zitiert auf den Seiten 35 und 39)

[HTWM04] Hu, Weiming ; Tan, Tieniu ; Wang, Liang ; Maybank, Steve: A survey on visual surveillance of object motion and behaviors. In: *IEEE Transactions on Systems, Man and Cybernetics* 34 (2004), S. 334–352 (Zitiert auf Seite 30)

[HW03] Hong, Dongpyo ; Woo;, Woontack: A background subtraction for a vision-based user interface. In: *Information, Communications and Signal Processing* 1 (2003), S. 263 – 267 (Zitiert auf Seite 35)

[JDWR00] Jabri, Sumer ; Duric, Zoran ; Wechsle, Harry ; Rosenfeld, Azriel: Detection and location of people in video images using adaptive fusion of color and edge information. In: *In Proc. 15th Int'l Conf. on Pattern Recognition*, 2000, S. 627–630 (Zitiert auf den Seiten 35 und 67)

[JS08] Javed, Omar ; Shah, Mubarak: *Automated Multi-Camera Surveillance: Algorithms and Practice*. Springer Publishing Company, Incorporated, 2008 (Zitiert auf Seite 29)

[KB90] Karmann, K.P. ; Brandt, A.: *Moving object recognition using an adaptive background memory*. 1990 (Zitiert auf Seite 32)

[KC04] Kamath, C. ; Cheung, Sen-Ching S.: Robust techniques for background subtraction in urban traffic video. In: *Visual Communications and Image Processing* 5308 (2004), S. 881–892 (Zitiert auf den Seiten 32 und 33)

[KCHD04] Kim, Kyungnam ; Chalidabhongse, Thanarat H. ; Harwood, David ; Davis, Larry: Background modeling and subtraction by codebook construction. In: *In International Conference on Image Processing*, 2004, S. 3061–3064 (Zitiert auf Seite 75)

[LGYS05] Lutz, Mustafa K. ; Goldmann, Lutz ; Yu, Da ; Sikora, Thomas: *Comparison of Static Background Segmentation Methods*. 2005 (Zitiert auf Seite 35)

[LHGT03] Li, Liyuan ; Huang, Weimin ; Gu, Irene Y. H. ; Tian, Qi: Foreground object detection from videos containing complex background. In: *In MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, ACM Press, 2003, S. 2–10 (Zitiert auf Seite 83)

[MJD+00] McKenna, Stephen J. ; Jabri, Sumer ; Duric, Zoran ; Rosenfeld, Azriel ; Wechsler, Harry: Tracking Groups of People. In: *Computer Vision and Image Understanding* 80 (2000), S. 42–56 (Zitiert auf den Seiten 35 und 66)

[MP04] Mittal, Anurag ; Paragios, Nikos: Motion-Based Background Subtraction Using Adaptive Kernel Density Estimation. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, S. 302–309 (Zitiert auf Seite 33)

[MS95] McFarlane, N. ; Schofield, C.: Segmentation and tracking of piglets in images. In: *Machine Vision and Applications* 8 (1995), May, Nr. 3, S. 187–193 (Zitiert auf den Seiten 32 und 64)

[ORP00] Oliver, N.M. ; Rosario, B. ; Pentland, A. P.: A Bayesian Computer Vision System for Modeling Human Interactions. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* Bd. 22, 2000, S. 831 – 843 (Zitiert auf den Seiten 30 und 37)

[PCMT01] Prati, Andrea ; Cucchiara, Rita ; Mikic, Ivana ; Trivedi, Mohan M.: Analysis and Detection of Shadows in Video Streams: A Comparative Evaluation. In: *In Proc. Int. Conf. Computer Vion and Pattern Recognition*, 2001, S. 571–576 (Zitiert auf den Seiten 93 und 94)

[Pic04] Piccardi, Massimo: Background subtraction techniques: a review. In: *IEEE International Conference on Systems Man and Cybernetics IEEE Cat* 4 (2004), S. 3099–3104 (Zitiert auf Seite 30)

[Rap] `http://www.ulrcih-rapp.de/stoff/pc/tabkal/Normalverteilung_Diagramm.png` (Zitiert auf Seite 27)

[Ros02] Rosin, Paul L.: Thresholding for change detection. In: *Computer Vision and Image Understanding* 86 (2002), Nr. 2, S. 79–95 (Zitiert auf Seite 84)

[SG99] Stauffer, Chris ; Grimson, W.E.L.: Adaptive Background Mixture Models for Real-Time Tracking. In: *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* 2 (1999), S. 2246 (Zitiert auf den Seiten 32, 33, 36, 39 und 70)

[She04] Shen, J.: Motion detection in color image sequence and shadow elimination. In: *PROCEEDINGS- SPIE THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING* 5308 (2004), S. 731–740 (Zitiert auf Seite 35)

[SRPC06] Simone, Calderara ; Rudy, Melli ; Prati, Andrea ; Cucchiara, Rita: Reliable background suppression for complex scenes. In: *VSSN '06: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, 2006, S. 211–214 (Zitiert auf Seite 36)

[TEBM] Tiburzi, F. ; Escudero, Marcos ; Besos, Jesus ; Martinez, Jose M.: *A Corpus for Motion-based Video-object Segmentation* (Zitiert auf Seite 33)

[WADP97] Wren, Christopher ; Azarbayejani, Ali ; Darrell, Trevor ; Pentland, Alex: Pfinder: Real-Time tracking of the Human Body. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1997), S. 780–785 (Zitiert auf den Seiten 33 und 36)

[Wika] `http://de.wikipedia.org/wiki/RGB-Farbrauml` (Zitiert auf Seite 21)

[Wikb] `http://de.wikipedia.org/wiki/HSV-Farbraum` (Zitiert auf Seite 22)

[Wikc] `http://de.wikipedia.org/wiki/Receiver_Operating_Characteristic` (Zitiert auf Seite 98)

[ZH06] Zivkovic, Z. ; Heijden, F. van d.: Efficient adaptive density estimation per image pixel for the task of background subtraction. In: *Pattern Recognition Letters* 27 (2006), Nr. 7, S. 773–780 (Zitiert auf Seite 36)

Alle URLs wurden zuletzt am 11.05.2010 geprüft.

Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst und nur die angegebenen Quellen benutzt zu haben.

(Sebastian Brutzer)