

Technische Universität München  
Ingenieurfaculty Bau Geo Umwelt  
Lehrstuhl für Methodik der Fernerkundung

# **Robustes Tracking mithilfe des Partikel Filters - Ein merkmalsbasierter Ansatz und dessen Anwendung im Bereich der Verkehrsüberwachung**

**Tobias Koch**

## **Masterarbeit**

Geodäsie und Geoinformation

**Bearbeitung:** 27.05.2013 – 02.01.2014

**Betreuer:** Prof. Dr.-Ing. habil. Richard Bamler, TUM  
Dr.-Ing. Markus Ulrich, MVTec  
Dr.-Ing. Friedrich Fraundorfer, TUM

**Kooperation:** 



Technische Universität München  
Ingenieur fakultät Bau Geo Umwelt  
Masterarbeit  
WS 2013/14

# **Robustes Tracking mithilfe des Partikel Filters**

**Ein merkmalsbasierter Ansatz und dessen Anwendung im Bereich der Verkehrsüberwachung**

Themensteller:

Prof. Dr. Richard Bamler<sup>1</sup>  
Masterarbeit  
Ingenieur fakultät Bau Geo Umwelt  
Technische Universität München  
Arcisstraße 11, 80333 München

Bearbeitet von:

Tobias Koch  
Matrikelnummer: 3602730 , Fachsemester: 11  
Betreuer: Dr.-Ing. Markus Ulrich<sup>2</sup>  
Dr.-Ing. Friedrich Fraundorfer<sup>1</sup>

Abgabetermin: 02. Januar 2014

---

<sup>1</sup>Ingenieur fakultät Bau Geo Umwelt, Technische Universität München, Arcisstraße 21, (80333 München)  
<sup>2</sup>MVTec Software GmbH, Neherstr. 1, 81675 München

---

---

## Selbstständigkeitserklärung

---

Erklärung gemäß §18 Absatz 9 APSO der Technischen Universität München:

**„Ich versichere, dass ich diese Masterarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.“**

---

Tobias Koch  
München, den 02.01.2014



---

## Abstract

This Master's Thesis provides a road traffic tracking algorithm based on particle filtering using static cameras. A template-based object model was applied using color histograms and edge orientation histograms as features. A multipart appearance model can individually weight single parts of the object in case of occlusions. This provides a robust tracking for multi object tracking to a high degree of occlusion. Furthermore, a prediction of a totally occluded object can be achieved. Two different motion models allow a stable tracking for different points of view. The tracking performance is evaluated using real traffic scenes. Finally, an automatic traffic surveillance system is presented, combining the tracking algorithm with an object detector based on background subtraction, from which real traffic statistics can be extracted.

## Zusammenfassung

Diese Arbeit beschäftigt sich mit der Entwicklung eines Verkehrsüberwachungssystems mit Hilfe eines Partikel Filters bei Verwendung von stationären Kameras. Dieser merkmalsbasierte Ansatz verwendet Farb- und Kantenmerkmale in Form von Templates, sowie eine Unterteilung des Objektmodells in mehrere Subregionen. Eine dynamische Gewichtung dieser Regionen bei simultanem Tracking mehrerer Objekte ermöglicht ein robustes Tracking bis zu einem hohen Verdeckungsgrad. Ebenso kann eine Vorhersage komplett verdeckter Objekte über einen gewissen Zeitraum erfolgen. Die Verwendung zweier unterschiedlicher Bewegungsmodelle gewährt zudem ein stabiles Tracking für unterschiedliche Aufnahmeperspektiven. Die Fähigkeiten des Verfahrens werden an einigen realen Sequenzen untersucht. Mit der Kombination eines Objektdetektors wird ein vollautomatisches Trackingsystem vorgestellt, aus dem Statistiken für Verkehrsanalysen extrahiert werden können.



---

---

## Danksagungen

---

Zunächst möchte ich mich an dieser Stelle bei all denjenigen bedanken, die mich während der Anfertigung dieser Masterarbeit unterstützt und motiviert haben.

Mein besonderer Dank gebührt Herrn Dr.-Ing. Markus Ulrich für das Bereitstellen dieses interessanten Themas der Masterarbeit und der tatkräftigen Unterstützung und Betreuung während der Bearbeitung. Ebenso danke ich Herrn Dr.-Ing. Friedrich Fraundofer für die viele Zeit, die er sich für fachliche Gespräche genommen hat. Ohne ihn wäre diese Zusammenarbeit zwischen der Technischen Universität München und MVTec Software GmbH nicht möglich gewesen.

Insbesondere möchte ich auch Tobias Böttger für seinen fachlichen Rat danken. Er hat viel Zeit dafür geopfert hat, die Ergebnisse meiner Arbeit zu begutachten und weitere hilfreiche Anregungen zu geben.

Ebenso möchte ich Christina Eisenhofer, Matthias Meßner, Andreas Pumpf und Roman Moie danken, die neben einer umfangreichen Unterstützung ebenfalls zu einer sehr entspannten und kreativen Arbeitsatmosphäre beigetragen haben.

Nicht zuletzt gilt mein besonderer und tiefer Dank meiner Familie, die mich nicht nur während der Masterarbeit, sondern während meines gesamten Studiums intensiv unterstützt und ermutigt haben.





<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Aufgabenstellung . . . . .	2
1.3	Aufbau der Arbeit . . . . .	3
<b>2</b>	<b>Stand der Forschung</b>	<b>5</b>
2.1	Ansätze zur Verkehrsüberwachung in Videos . . . . .	5
2.1.1	Modellbasiertes Tracking . . . . .	5
2.1.2	Regionenbasiertes Tracking . . . . .	6
2.1.3	Konturbasiertes Tracking . . . . .	7
2.1.4	Merkmalsbasiertes Tracking . . . . .	8
2.2	Grundlagen merkmalsbasierter Objektverfolgung . . . . .	8
2.3	Angewandte Methode und Mehrwert der Arbeit . . . . .	13
<b>3</b>	<b>Mathematische Grundlagen des Partikel Filters</b>	<b>15</b>
3.1	Rekursive Bayes-Filter . . . . .	15
3.2	Kalman Filter . . . . .	19
3.3	Partikel Filter . . . . .	21
<b>4</b>	<b>Merkmalsextraktion</b>	<b>29</b>
4.1	Farbe . . . . .	30
4.1.1	Histogramm Binning . . . . .	30
4.1.2	Einführung einer Gewichtungsfunktion . . . . .	31
4.1.3	Multi-Part Farbhistogramme . . . . .	32
4.2	Kanten . . . . .	33
4.2.1	Kantenrichtungshistogramm . . . . .	33
4.2.2	Multi-Part Kantenrichtungshistogramme . . . . .	36
<b>5</b>	<b>Anwendung eines Partikel Filters für die Verkehrsüberwachung</b>	<b>39</b>
5.1	Modellbildung . . . . .	40
5.2	Ablauf des Partikel Filters . . . . .	40
5.3	Prädiktion mittels Bewegungsmodellen . . . . .	44
5.3.1	Konstantes Geschwindigkeitsmodell . . . . .	45
5.3.2	Einbeziehung einer Rotation . . . . .	46

---

5.4	Korrektur mittels Beobachtungsmodellen . . . . .	48
5.4.1	Farbe . . . . .	48
5.4.2	Kanten . . . . .	51
5.5	Behandlung von Verdeckungen . . . . .	52
5.5.1	Verdeckung mit dem Bildhintergrund . . . . .	53
5.5.2	Verdeckung zweier Objekte . . . . .	55
5.6	Externe Schätzung der Skalierung . . . . .	56
<b>6</b>	<b>Evaluierung</b>	<b>59</b>
6.1	Einfluss der Parameter des Partikel Filters auf Laufzeit und Robustheit . . . . .	60
6.1.1	Wahl des Prozessrauschens . . . . .	60
6.1.2	Wahl der Merkmalsparameter . . . . .	62
6.1.3	Anzahl der Partikel . . . . .	65
6.2	Testen unterschiedlicher Szenen . . . . .	66
6.2.1	Aufnahmen vom Straßenrand . . . . .	67
6.2.2	Autobahnsequenzen . . . . .	70
6.2.3	Kreuzungsaufnahmen . . . . .	80
6.3	Einfluss externer Faktoren . . . . .	80
6.3.1	Einfluss der Bildauflösung . . . . .	82
6.3.2	Einfluss der Bildfrequenz . . . . .	82
<b>7</b>	<b>Anwendung eines automatischen Verkehrsüberwachungssystems</b>	<b>83</b>
7.1	Ziele . . . . .	83
7.2	Detektion . . . . .	84
7.3	Tracking . . . . .	85
7.4	Ergebnisanalyse . . . . .	85
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>89</b>
8.1	Zusammenfassung . . . . .	89
8.2	Ausblick . . . . .	90

### 1.1 Motivation

Dank der rasanten Entwicklung der Computertechnologie innerhalb der letzten Jahrzehnte und den immer kompakter und preiswerter werdenden Digitalkameras, vermehrt sich der Wunsch einer automatischen, echtzeitfähigen Videoauswertung. Das Forschungsgebiet der *Computer Vision* beschäftigt sich seit einigen Jahren innerhalb des *Video Trackings* damit, Objekte in Videosequenzen möglichst robust und in Echtzeit zu verfolgen. Dies gewährt eine zunehmend steigende Anwendungsvielfalt, von privaten bis hin zu militärischen Zwecken. Sogenannte *Smart Interfaces* erlauben eine intuitive Steuerung eines Computers mittels Gesten und Augenbewegungen und wird bereits ansatzweise in Fahrassistenzsystemen in der Automobilindustrie angewendet. In der Robotik ermöglicht die Detektion und das Verfolgen von Objekten eine bessere Umgebungswahrnehmung für Roboter und kann zur Navigation angewendet werden. Klassischere Anwendungsgebiete finden sich im Überwachungsbereich, sei es für Privatgrundstücke, staatliche bzw. militärische Anlagen oder im Straßenverkehr. Mit Hilfe von einfachen Digitalkameras an fest installierten Punkten kann eine vollautomatische Überwachung der Verkehrslage an besonders kritischen Stellen erfolgen. Dies kann für eine verbesserte, schnellere Stauwarnung, aber auch zur Analyse des Verkehrsflusses verwendet werden. So können Statistiken wie Geschwindigkeiten, Zählen von Verkehrsteilnehmern und Spurdichten extrahiert, sowie eine Klassifikation von Verkehrsobjekten durchgeführt werden, um mögliche Verbesserungen am Straßenverlauf sowie zur Steuerung des Verkehrsflusses vorzunehmen.

Die Entwicklung eines robusten und zugleich echtzeitfähigen Verkehrsüberwachungssystems stellt dabei immer noch eine große Herausforderung dar, die mit sehr vielen Schwierigkeiten behaftet ist. Was einem Menschen leicht fällt, kann dabei zu einer großen Herausforderung für einen Computer werden, wenn es darum geht ein Objekt innerhalb eines Videos zu verfolgen das aufgrund zahlreicher Einflüsse im Laufe dieser Sequenz ganz unterschiedliche Erscheinungsbilder

aufweisen kann. Beleuchtungsänderungen, Rotationen und Verdeckungen seien hier als Beispiele genannt. Besonders letzteres stellt ein kritisches Problem innerhalb des Video Trackings dar. So müssen Modelle entwickelt werden, die es ermöglichen, das Wiedererscheinen verschwundener Objekte, wie beispielsweise nach Tunneldurchfahrten von Vehikeln, korrekt zu detektieren und anschließend von anderen, ähnlichen Objekten unterscheiden zu können. Ebenso können sich Objektbewegungen aus unterschiedlichen Aufnahmeperspektiven aufgrund der Projektion von 3D-Weltkoordinaten in die 2D-Bildebene stark unterscheiden und in der Regel nicht mit einem einheitlichen Modell dargestellt werden.

Damit eine möglichst stabile Objektverfolgung realisiert werden kann ist es daher notwendig für unterschiedliche Anwendungen passende Modelle zu entwickeln und diese in einem Gesamtkonzept zu integrieren, damit eine benutzerfreundliche Anwendung eines Trackingverfahrens gewährleistet werden kann.

### 1.2 Aufgabenstellung

Ziel dieser Arbeit ist es, ein Trackingverfahren zu entwickeln, das dafür ausgelegt sein soll, verschiedenste Objekte im Straßenverkehr (PKW, LKW, Motorrad, Fahrräder, Passanten) in unterschiedlichen Szenarien robust über einen möglichst langen Zeitraum verfolgen zu können. Dabei soll das Verfahren so gewählt sein, dass es ohne großen Aufwand für verschiedene Szenen und damit unter unterschiedlichen Bedingungen einsetzbar ist.

Die zur Evaluierung dienenden selbst erstellten Videos sollen dabei möglichst realitätsnah an Überwachungskameras in Verkehrsszenen herankommen. Dies bedeutet daher auch, dass eine stationäre Kamera verwendet wurde. Dies gewährt zusätzliches Wissen über das Bewegungsverhalten der Objekte und soll in das Tracking mit einfließen. Zudem erlaubt es eine einfache Detektion der Fahrzeuge und kann für die Initialisierung des Trackers dienen. Die Kopplung mit einem Detektor ermöglicht damit ein vollautomatisches Verkehrsüberwachungssystem, welches ohne manuelles Eingreifen Statistiken über den Verkehrsfluss einer Szene liefert. Die meisten gängigen Verkehrsüberwachungssysteme basieren auf einem Hintergrundschätzer. Die Detektion des Vordergrundes in jedem Frame liefert detektierte Objekte, die den Objekten des vorherigen Frames zugeordnet werden müssen. Im Unterschied zu diesen regionenbasierten Trackingverfahren soll in dieser Arbeit ein merkmalsbasierter Ansatz verwendet werden. Abgesehen von der Initialisierung der Objekte, soll das Tracking ohne Verwendung eines Hintergrundmodells auskommen. Für verschiedene Szenarien müssen unterschiedliche Vorgehensweisen herangezogen werden, die stark abhängig vom Erscheinungsbild und dem Bewegungsverhalten der in der Szene vorkommenden Objekte ist. Es soll daher auf ein möglichst modulares Konzept zurückgegriffen werden, das eine schnelle Anpassung des Trackers auf verschiedenen Szenarien ermöglicht.

Beim Verfolgen von Objekten im Straßenverkehr müssen u.a folgende Probleme berücksichtigt werden:

- Informationsverlust durch die Projektion der 3D-Welt in eine 2D-Bildebene

- Mehrere, ähnliche Objekte können sich sehr nahe kommen und gegenseitig verdecken.
- Bewegungsunschärfe
- Beleuchtungsänderungen innerhalb der Szene
- Vehikel können bei Detektion bereits sehr schnell sein.

Folgendes Vorwissen, das sich zum Teil auf die Verwendung einer stationären Kamera bezieht, kann herangezogen werden:

- Fahrzeuge können als farblich homogene, starre Körper angenommen werden.
- Bewegungen von Vehikeln sind meist konstant und vorhersehbar.
- Größen- und Geschwindigkeitsänderungen der Objekte abhängig von oftmals bekannter Bildgeometrie.

### 1.3 Aufbau der Arbeit

Im folgenden Kapitel wird ein Überblick über bisherige Ansätze der Objektverfolgung in Verkehrsaufnahmen dargestellt. Zudem werden die Grundlagen des Trackings erläutert und verschiedene Trackingstrategien vorgestellt, aus denen schließlich das hier verwendete Verfahren abgeleitet werden kann.

Das dritte Kapitel stellt die mathematischen Grundlagen eines Partikel Filters vor. Neben der Herleitung eines Bayes-Filters wird das verwandte Kalman Filter und schließlich das Partikel Filter erläutert.

Das vierte Kapitel beschreibt die in dieser Arbeit verwendeten Bildmerkmale, sowie deren Extraktion.

Kapitel fünf stellt dar, wie ein Partikel Filter für die Objektverfolgung realisiert werden kann und zeigt wie Erweiterungen die Robustheit bei Anwendung in Verkehrsszenen erhöhen können. Im sechsten Kapitel wird das entwickelte Trackingverfahren auf mehrere reale Sequenzen angewendet. Dabei soll das Verhalten, sowie Vor- und Nachteile verschiedener Bildmerkmale und deren Parameterwahl erläutert werden. Zudem werden Einflüsse von Auflösung und Bildfrequenz auf die Robustheit des Systems getestet.

In Kapitel sieben wird schließlich ein automatisches Verkehrsüberwachungssystem basierend auf den in Kapitel fünf vorgestellten Partikel Filter dargestellt. Dieses ermöglicht eine automatische Objekterkennung und -verfolgung, sowie eine Analyse der aktuellen Verkehrssituation. Schließlich erfolgt eine allgemeine Zusammenfassung und der Ausblick auf mögliche Verbesserungen des Verfahrens.



## 2.1 Ansätze zur Verkehrsüberwachung in Videos

Für die Entwicklung eines Verkehrsüberwachungssystems gibt es eine Vielzahl unterschiedlicher Verfahren.

Die ersten sogenannten ITS (*Intelligent Transportation Systems*) basierten ausschließlich auf Objektdetektoren, die mit teilweise unterschiedlichen Sensoren arbeiten. Beispiele solcher Anwendungen sind AUTOSCOPE, CCATS, TAS, IMPACTS und TrafficCam [15]. Optische Verkehrsüberwachung erfolgte dabei durch das Registrieren von Intensitätsänderungen in manuell definierten Bildregionen, die durch bewegte Objekte hervorgerufen werden. Ein Tracking der Fahrzeuge konnte durch diese Ansätze nicht vollzogen werden, sondern lediglich das Registrieren der Vehikel in definierten Bildregionen.

Durch explizites Tracking der Verkehrsteilnehmer in der gesamten Szene können wesentlich mehr Informationen über den aktuellen Verkehrsfluss extrahiert werden wie beispielsweise Geschwindigkeiten, Spurwechsel, sowie einzelne Trajektorien der Objekte. Hiermit kann das Verhalten von Verkehrsteilnehmern in überwachten Straßenabschnitten aufschlussreicher analysiert werden.

Zeitgemäße Methoden versuchen daher präsente Objekte zu jedem Zeitpunkt im Bild zu lokalisieren. Diese Verfahren unterscheiden sich in ihrer Herangehensweise und können in *modellbasierte*, *regionenbasierte*, *konturbasierte* und *merkmalsbasierte* Trackingverfahren unterteilt werden [15].

### 2.1.1 Modellbasiertes Tracking

In diesen *top-down*-Verfahren werden detaillierte geometrische zwei- oder dreidimensionale Objektmodelle der Vehikel generiert und möglichst genau auf die in den Videos auftretenden Fahrzeuge projiziert (siehe Abbildung (2.1a)). Lepetit und Fua stellen in [36] eine umfangreiche Übersicht über die Vorgehensweise modellbasierter Trackingverfahren dar. Weitere Anwendun-



(a) Modellbasiert [36] (b) Regionenbasiert [11] (c) Konturbasiert [32] (d) Merkmalsbasiert [46]

**Abbildung 2.1:** Unterschiedliche Trackingverfahren für Verkehrsüberwachung

gen dieses Ansatzes in Bezug auf Verkehrsaufnahmen finden sich u.a. in [31], [49], [3]. Diese können ein sehr detailliertes Tracking durch die Verwendung exakter Modelle gewährleisten und eignen sich durch die bekannte Objektform auch im Falle von Verdeckungen. Die meisten Verfahren beziehen sich allerdings auf die Verfolgung einzelner, vorher gut initialisierter Objekte und setzen eine kalibrierte Kamera voraus, die in Verkehrsaufnahmen nicht immer zur Verfügung steht. Die hohe Rechenintensität verhindert ein echtzeitfähiges Tracking mehrerer Objekte bei dicht befahrenen Straßen. Zudem müsste eine sehr große Datenbank an Objektmodellen von sämtlichen am Straßenverkehr teilnehmenden Vehikeln erstellt werden und eine zusätzliche Klassifikation der detektierten Objekte vorgenommen werden. Für ein automatisches Verkehrsüberwachungssystem eignen sich diese Ansätze daher nicht.

### 2.1.2 Regionenbasiertes Tracking

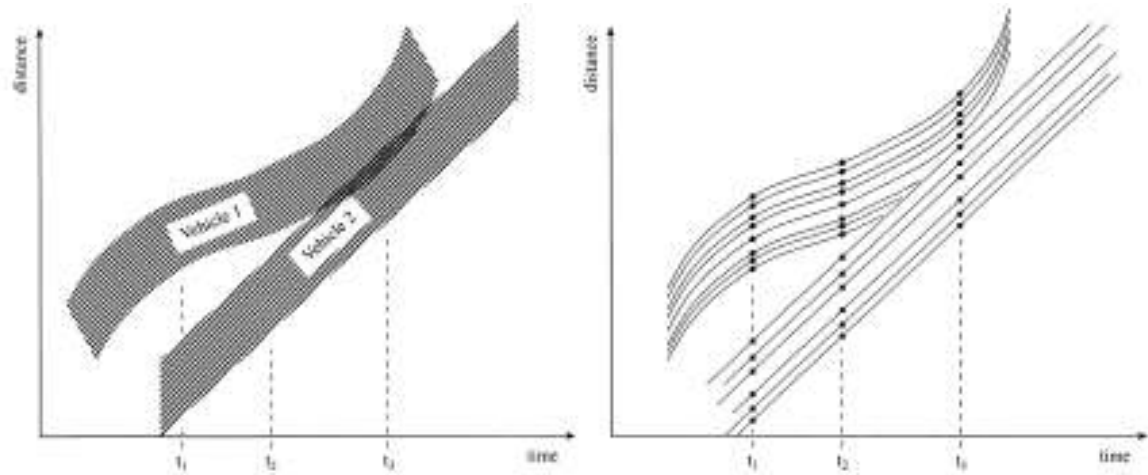
Diese Verfahren stellen die am häufigsten verwendeten Methoden dar und gehören zur Klasse der *bottom-up*-Verfahren, da das Tracking ohne Verwendung eines Objektmodells auf Basis der Bildwerte erfolgt. Dabei werden in jedem Bild zusammenhängende Regionen, so genannte *Blobs*, identifiziert (siehe Abbildung (2.1b)). Eine Zuordnung dieser Blobs zu bestehenden Objekten erfolgt durch einen Vergleich (deterministisch oder probabilistisch) ihrer Position und Merkmale mit den detektierten Blobs des vorherigen Bildes.

Grundlage hierfür bilden in der Regel Hintergrundbilder der Szene die durch die Verwendung einer stationären Kamera gebildet werden können. Mit Hilfe sogenannter *Background-Subtraction*-Methoden können zwischen dem aktuellen Bild und dem Hintergrundbild Intensitätsänderungen der Pixel festgestellt werden die durch bewegte Objekte entstehen. Überschreiten diese einen gewissen Schwellwert, werden sie als Vordergrund klassifiziert. Durch eine anschließende Segmentierung der Vordergrundpixel und die Verwendung morphologischer Operatoren entstehen die oben erwähnten Blobs. Eine Vorstellung, sowie ein Vergleich unterschiedlicher *Background-Subtraction*-Methoden findet sich beispielsweise in [9].

Die meisten regionenbasierten Trackingverfahren beziehen sich dabei auf die korrekte Zuweisung der detektierten Objekte mit denen des vorherigen Frames. Dies wird auch als *Tracking-by-Detection* bezeichnet. Anwendungen dieser Trackingmethode finden sich u.a in [34], [30], [28], [10], [35].

Nachteil dieser Trackingstrategie ist es, dass es auf einen guten Hintergrundschätzer angewie-





**Abbildung 2.2:** Vergleich unterschiedlicher Trackingverfahren bei Verdeckungen [15]. Links: regionenbasiertes Tracking kann zwei sich verdeckende Objekte nicht mehr trennen. Rechts: merkmalsbasiertes Tracking kann das verdeckte Objekt (Vehikel 1) trotz zum Teil verschwundener Merkmale eindeutig identifizieren.

sen ist. Die Qualität des Trackings ist dabei abhängig von der Genauigkeit und Robustheit des Hintergrundbildes, dessen Bildung zum Aufgabenbereich der Objektdetektion gehört. Die größte Schwierigkeit ergibt sich, wenn sich zwei oder mehrere Objekte überlappen bzw. gegenseitig verdecken. Dies führt dazu, dass beide Objekte zu einem zusammenhängenden Blob verschmelzen. Die Trennung überlappender Objekte stellt bis heute ein großes Problem dar. Der linke Bildausschnitt in Abbildung (2.2) stellt dieses Phänomen im eindimensionalen Fall dar. Eine Trennung beider Blobs stellt für die Zeitpunkte  $t_1$  und  $t_2$  kein Problem dar. Zum Zeitpunkt  $t_3$  wird jedoch ein Teil von Vehikel 1 von Vehikel 2 verdeckt; somit verschmelzen beide Objekte, deren Trennung auf Pixelbasis eine große Schwierigkeit darstellt.

Eine weiteres Problem besteht in der Behandlung von Schatten. Gerade bei tiefstehender Sonne können sehr lange Schatten entstehen, die ebenfalls als Vordergrund registriert werden. Eine Trennung dieser Schatten vom Objekt stellen bisher noch große Herausforderungen dar. Weiterhin ist dieser Ansatz auf die Verwendung einer sehr stabilen Kamera angewiesen, da ein Verwackeln des Bildes (beispielsweise durch Wind) zu sehr vielen Fehldetektionen führen könnte.

### 2.1.3 Konturbasiertes Tracking

Das Tracking basiert in dieser Strategie auf dem Verfolgen der Objektkontur (siehe Abbildung (2.1c)). Auf Grundlage einer bekannten Objektkontur kann diese im Laufe der Sequenz dynamisch aktualisiert werden und auf Variationen des Erscheinungsbildes (zum Beispiel aufgrund von Rotationen) reagieren. Trucco und Plakas stellen in ihrer Übersicht über Video Tracking verschiedene konturbasierte Ansätze vor [51]. Diese meist sehr rechen-effizienten Methoden haben aber immer noch Probleme bei verdeckten Objekten [46]. Eine vorausgehende Segmentierung des

Objektes, woraus die Konturberechnung resultiert, kann zu schlechten Modellen führen falls sich mehrere Objekte überlagern bzw. verdecken. Der *Condensation Algorithmus* in [26] stellt einen Ansatz der Konturverfolgung dar, der nur auf Grundlage einer Initialisierung der Objektkontur erfolgt, ohne weitere Objektsegmentierungen im Laufe der Sequenz durchführen zu müssen. Mit diesem Ansatz ist es möglich das Objekt im Falle von Verdeckungen weiterzuverfolgen.

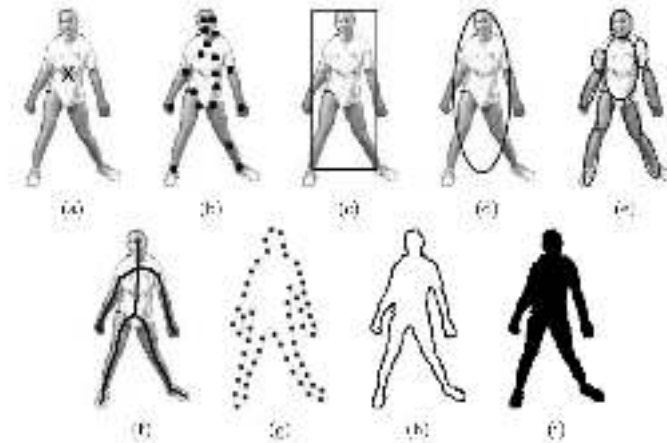
#### 2.1.4 Merkmalsbasiertes Tracking

Alternative Trackingverfahren auf Basis von Merkmalen (engl. *features*) unterscheiden sich zu den bisherigen Ansätzen dadurch, dass das Objekt nicht mehr als Ganzes betrachtet wird, sondern das Tracking auf Basis von Objektmerkmalen oder Sub-Merkmalen, wie beispielsweise markanten Punkten oder Linien auf dem Objekt, erfolgt. Für die Wahl der Objektmerkmale können verschiedene Ansätze zum Einsatz kommen. Je eindeutiger diese Merkmale das Objekt beschreiben können, desto besser eignen sie sich für die Behandlung von Verdeckungen. Im Vergleich zu regionenbasierten Ansätzen können trotz Teilverdeckung noch einige Merkmale sichtbar sein, anhand derer das Objekt verfolgt werden kann.

Der rechte Bildausschnitt in Abbildung (2.2) stellt dieses Vorgehen für das so eben beschriebene Beispiel dar. Jedes Vehikel kann anhand eindeutiger Merkmale beschrieben werden (dargestellt als Kreise für Vehikel 1 und Rechtecke für Vehikel 2). Im Fall der Verdeckung zum Zeitpunkt  $t_3$  werden zwar ebenfalls einige Merkmale des Vehikels 1 verdeckt, jedoch kann es anhand der restlichen Merkmale weiterhin eindeutig beschrieben und von Vehikel 2 getrennt werden. Häufig basiert die Merkmalswahl auf der Beschreibung des Objekts durch markante Punkte, die in den Folgebildern gesucht werden [46]. Diese können Eckpunkte, aber auch Scheinwerfer bzw. Rückleuchten des Fahrzeuges sein [12]. Durch die Darstellung von Punkten fällt jedoch die Bestimmung der Fahrzeuggröße, die in manchen Anwendungen von Interesse sein kann, schwer. Es gibt jedoch eine Vielzahl anderer Merkmale die verwendet werden können, welche im nachfolgenden Kapitel beschrieben werden.

## 2.2 Grundlagen merkmalsbasierter Objektverfolgung

In dieser Arbeit wird ein merkmalsbasierter Trackingalgorithmus beschrieben, da diese in der Regel bessere Ergebnisse bei Verdeckungen erzielen [29], welche ein häufiges Problem in Verkehrsaufnahmen darstellen. Dabei soll für den Trackingalgorithmus auf die Verwendung eines Hintergrundbildes wegen oben genannter Probleme verzichtet werden. Punktbasierte Trackingverfahren eignen sich zwar, wie bereits beschrieben, für eine robuste Verfolgung der Objekte, können jedoch die Größe des Objekts nicht wahrnehmen. Hierfür können weitere Bildmerkmale herangezogen werden die innerhalb von Bildregionen, sogenannten *Templates*, berechnet werden und somit eine räumliche Anordnung der Pixel berücksichtigen. Da die verwendeten Bildmerkmale oft mit der Darstellungsart des Objektes zusammenhängen sollte zunächst geklärt werden, wie das zu verfolgende Objekt dargestellt werden soll. In einer sehr ausführlichen Übersicht über das Tracking in Videosequenzen unterteilen Yilmaz et al. das Videotracking in die Unterpunkte



**Abbildung 2.3:** Verschiedene Darstellungsmöglichkeiten von Objektformen [56].

### Objektrepräsentation, Objektdetektion und Objektverfolgung [56].

**Objektrepräsentation** In der Objektrepräsentation soll ein Modell über die Erscheinung des Objektes erstellt und die Wahl der Bildmerkmale geklärt werden. Für die Objekterscheinung gibt es, wie in Abbildung (2.3) dargestellt, unterschiedliche Arten. Die Wahl der Darstellungsform für das Objektmodell soll so ausgerichtet sein, dass das Objekt möglichst gut repräsentiert werden kann. Dies hängt insbesondere davon ab, wie detailliert das Objekt während des Trackings dargestellt werden kann und welche Analysen damit im Nachhinein folgen sollen. Die Komplexität der Berechnung des Trackingverfahrens ist allerdings ebenfalls von der gewählten Darstellungsform abhängig und sollte daher bei der Wahl ebenfalls berücksichtigt werden.

Die Beschreibung eines Objektes anhand seines Schwerpunktes (a) oder mehrerer markanter Punkte (b) eignet sich, wie bereits beschrieben, für die Darstellung von Fahrzeugen nur bedingt, da sie keine Information über die Größe des Objekts liefert. Rechtecke (c) oder Ellipsen (d), die das Objekt umschließen, können rechen-effizient dargestellt werden und die Größe des Objekts beschreiben. Sie können daher für diese Problemstellung herangezogen werden. Eine Merkmalsextraktion erfolgt dabei innerhalb dieser Bildregionen. Eine Unterteilung des Objektes in mehrere zusammenhängende Regionen (e) dient der Approximation komplexer Objektformen. Die in Verkehrsaufnahmen auftretenden Objekte können allerdings als starre Körper angenommen werden, daher findet diese Objektform keine Anwendung in dieser Arbeit. Dies gilt ebenso für die Verwendung von Objektskeletten (f). Durch die Beschreibung der Objektkontur (g-i) kann die wahre Objektform erhalten bleiben. Dies ermöglicht eine sehr genaue Objektbeschreibung und hat den Vorteil gegenüber einfachen geometrischen Formen, dass es sich um keine Approximation der Objektform handelt und dadurch weniger Hintergrundinformation in das Modell einfließt. Jedoch kann die Objektform mathematisch nicht mehr trivial erfasst werden. Dies führt zu Problemen bei variierender Objektgröße, die in der Regel Bestandteil von Verkehrsaufnahmen sind.

Wurde eine Objektform ausgewählt, kann eine hierfür geeignete Merkmalsextraktion erfolgen. Für diese Problemstellung eignet sich eine templatebasierte Merkmalswahl. Dies bedeutet, dass die Merkmalsextraktion innerhalb des verwendeten Templates (hier: Rechteck) erfolgt. Die Wahl der Bildmerkmale sollte so gewählt sein, dass sich das Objekt durch diese möglichst eindeutig beschreiben und von anderen, eventuell ähnlichen, Objekten unterscheiden lässt. Die Beschreibung des Objekts erfolgt damit im Merkmalsraum. Um ein möglichst echtzeitfähiges Trackingsystem zu entwickeln, sollte bei der Wahl der Bildmerkmale eine rechen-effiziente Implementierung dieser Merkmalsextraktion möglich sein, da dieser Schritt in der Regel der rechen-intensivste ist.

Ein beliebtes Bildmerkmal, welches eine Verwendung in den meisten Trackingsystemen findet, ist die **Farbe** eines Objekts. Oftmals können Objekte allein anhand ihrer Farbe vom Hintergrund oder auch von anderen Objekten differenziert werden. Zudem ist sie invariant gegenüber Skalierungen und Rotationen des Objekts. In der Regel werden Farbverteilungen anhand von Farbhistogrammen zur Objektbeschreibung verwendet. Diese sind effizient zu berechnen da die Pixel bereits als Farbwerte vorliegen und nicht mehr vorverarbeitet werden müssen.

Weitere oft verwendete Bildmerkmale stellen **Kanten** dar. Kanten werden in Bildern durch starke Intensitätsänderungen benachbarter Pixel sichtbar. Je höher diese Intensitätsänderung ist, desto stärker erscheint die Kante im Bild. Starke Objektkanten können dem Tracking als wertvolle Merkmale dienen, sofern diese im Laufe der Sequenz sichtbar bleiben. Besonders durch ihre Unempfindlichkeit gegenüber Beleuchtungsänderungen können sie bei dynamischen Lichtverhältnissen während einer Szene gute Ergebnisse erzielen. Konturbasierte Verfahren konzentrieren sich auf die Kanten im Objekturnriss, während histogrammbasierte Verfahren die Kanten ebenfalls innerhalb eines Templates verwenden.

Sollen Bildmerkmale verwendet werden, die auf Geschwindigkeiten basieren, können Ansätze eines **optischen Flusses** eingesetzt werden. Ein optischer Fluss beschreibt ein Vektorfeld das die Bewegungsrichtung- und geschwindigkeit jedes Bildpunktes angibt. Dies eignet sich besonders bei Verwendung einer stationären Kamera. Die hohe Rechenintensität und die schwierige Abgrenzung benachbarter Objekte seien hier als Nachteil genannt und eignen sich daher nicht für diese Problemstellung.

Zuletzt sei die **Textur** als Bildmerkmal erwähnt. Diese gilt als Maß für die Intensitätsänderung einer Objektoberfläche und beschreibt dabei, wie glatt und regelmäßig diese verläuft. Da Fahrzeuge relativ wenig Textur aufweisen und die Berechnung lokaler Texturstrukturen sehr rechenintensiv ist, werden auch sie innerhalb dieser Arbeit nicht verwendet.

**Objektdetektion** Ein Objektdetektor übernimmt die Aufgabe, ein Objekt bei Erscheinen in der Szene zu registrieren und den Tracker zu initialisieren. Hierfür lokalisiert er neue Objekte in einer vorab definierten Bildregion und stellt das verwendete Modell für dieses Objekt auf. In templatebasierten Trackingansätzen soll also eine möglichst gute Approximation des Objektes mit dem Template erfolgen und die Bildmerkmale innerhalb dieser Region extrahiert werden. Die Objektlokalisierung und die Objektmerkmale werden anschließend an den Tracker übergeben. Die Objektdetektion in Verkehrsaufnahmen erfolgt in der Regel, wie bereits erwähnt, über



**Abbildung 2.4:** Links: Deterministische *Single Hypothesis Localisation*-Verfahren verwenden Bildmerkmale um die prädiizierte Zustandsschätzung iterativ zu verbessern. Rechts: *Multiple Hypothesis Localisation*-Verfahren generieren mehrere Hypothesen, wobei die Zustandsschätzung aus der Qualität aller Hypothesen erfolgt. Abbildung aus [39].

*background subtraction*-Verfahren. Jedoch kann ebenfalls eine manuelle Initialisierung durch Angeben der Bildregion um das Fahrzeug als Objektdetektion verwendet werden.

**Objektverfolgung** Nachdem das Objekt bei Erscheinen in der Szene erfolgreich detektiert und die Bildmerkmale extrahiert wurden, besteht nun die Aufgabe des Trackers das Objekt in jedem neuen Bild anhand dieser Bildmerkmale zu lokalisieren. Es gibt zahlreiche verschiedene Methoden des Trackings, die im Rahmen dieser Arbeit nicht alle ausführlich erläutert werden können. Es werden daher nur gängige merkmalsbasierte Ansätze vorgestellt die sich für templatebasierte Darstellungsformen eignen. Diese können in *Single-Hypothesis-Localisation* (SHL) und *Multiple-Hypothesis-Localisation* (MHL) unterteilt werden [39].

**Single-Hypothesis-Localisation** Charakteristisch für SHL-Trackingverfahren ist, dass nur eine Hypothese pro Zeitpunkt verfolgt wird. Bei deterministischen SHL-Methoden werden im aktuellen Frame Bildmerkmale an der zuletzt geschätzten Objektposition extrahiert. Mit Hilfe einer Ähnlichkeitsfunktion (engl. *score-function*) wird die Übereinstimmung der extrahierten Bildmerkmale des Bildausschnittes mit den Referenzmerkmalen ermittelt. Die Objektposition wird in einem iterativen Vorgehen nach und nach verbessert bis der Tracker in einem Maximum dieser Ähnlichkeitsfunktion konvergiert (siehe linken Bildausschnitt in Abbildung (2.4)). Die Richtung der Verbesserung gibt ein ableitbarer Kernel im Suchbereich vor, sodass die Verbesserungen immer in Richtung des stärksten aufsteigenden Gradienten zeigen. Dieses Optimierungsproblem wird in der Regel mit Hilfe des Gradientenabstiegsverfahrens [16],[38] oder dem Erwartungsmaximierungs-Algorithmus [27] gelöst. Aufgrund der Verwendung dieser Kernel werden diese Verfahren auch als kernelbasierte Trackingverfahren bezeichnet. *Kanade-Lucas-Tomasi* (KLT)-Tracker [5], [48]. und *Mean Shift* (MS) [16], [53], [37], [55] sind Beispiele kernelbasierter Trackingverfahren und finden häufig Verwendung in Trackinganwendungen. Generell konvergieren SHL-Methoden nach wenigen Iterationen und sind daher auch ohne großen Rechenaufwand durchführbar. Dies spricht für Echtzeitanwendungen.

Maggio und Cavallaro stellen allerdings die Grenzen dieser Systeme dar [39]. Als nachteilig

erweist sich zum einen die Schwäche bei schnellen Objektbewegungen. Da als Initialisierung jeweils die Position des letzten Zeitpunkts verwendet wird, kann ein schnelles Objekt außerhalb des Kernels liegen, sodass der Tracker niemals im globalen Maximum konvergieren kann und das Objekt somit verliert. Ebenso liefert dieser Ansatz schlechte Ergebnisse bei Verdeckungen. Sollte das Objekt kurzzeitig verschwinden konvergiert der Tracker in beliebigen Bereichen um das gesuchte Objekt oder driftet ab. Ein Wiederfinden bei Erscheinen des Objekts kann durch den meist begrenzten Suchbereich nicht garantiert werden. Weiterhin kann die merkmalsbasierte *score-function* multi-modal werden falls sich zwei ähnliche Objekte nebeneinander befinden. Dies wird durch SHL-Methoden nicht berücksichtigt und der Tracker konvergiert in einem Maximum, das abhängig vom Startpunkt, nicht zwingend vom gesuchten Objekt stammen muss und damit ein lokales und kein globales Maximum darstellt.

Eine Möglichkeit diese Probleme zu beheben besteht in der Kombination dieser Verfahren mit probabilistischen Methoden in Form von Bayes-Filter. Ein bekanntes Bayes-Filter stellt das Kalman Filter dar. Mit Hilfe eines zugrunde liegenden Bewegungsmodells kann eine gezielte Vorhersage der Objektposition anhand der letzten Objektbewegungen getroffen werden. Durch eine zusätzliche Unsicherheitsschätzung kann zudem der Suchraum des Trackings variiert werden. Dies eignet sich insbesondere bei schnellen Objektbewegungen [16], [6]. Die Qualität eines Trackings mittels Kalman Filterung ist jedoch stark von der gewählten Anwendung abhängig. Nur unter bestimmten Voraussetzungen kann ein Kalman Filter gute Ergebnisse liefern, während die Fähigkeit mit Verdeckungen umgehen zu können sehr limitiert ist [39]. Das Kalman Filter wird in Kapitel 3.2 detaillierter vorgestellt.

**Multiple-Hypothesis-Localisation** Im Gegensatz zu SHL werden bei MHL mehrere Hypothesen simultan verfolgt. Jede Hypothese stellt dabei einen diskreten Zustand (für Video Tracking gibt dieser Auskunft über die Position des Objektes im Bild) dar. Nach einer Merkmalsextraktion für jeder dieser Hypothesen und einem anschließendem Vergleich mit den Referenzmerkmalen kann die Qualität jeder dieser Zustände ermittelt werden. Es folgt eine Gewichtung der Hypothesen anhand ihrer Ähnlichkeiten zu den Referenzmerkmalen, woraufhin der aktuelle Zustand des Objekts anhand dieser Gewichte ermittelt werden kann.

*Grid-based*-Filter decken mit den Hypothesen ein regelmäßiges Gitter über den Zustandsraum (Bildregion) ab. Nach der Merkmalsextraktion für all diese Zustände und der Verwendung des Ähnlichkeitsmaßes wird anschließend der Zustand mit dem höchsten Ähnlichkeitsmaß als aktueller Zustand gewählt. Für die Objektverfolgung in Videos kann ein *Templatematching* als einfacher *grid-based*-Filter verstanden werden. Diese eignen sich aufgrund ihrer hohen Rechenintensität nicht für Echtzeitanwendungen.

*Partikel Filter* versuchen durch geeignetes Sampling die Zustandshypothesen um den erwartenden aktuellen Zustand zu fokussieren. Dieses kann zwar abhängig von der Dimensionalität (Rechenaufwand steigt exponentiell mit der Anzahl der Zustansparameter) immer noch sehr rechen-intensiv sein, ist aber dennoch effizienter als ein *grid-based*-Filter [39].

Die Vorteile der MHL-Verfahren im Video-Tracking liegen in der Fähigkeit durch simultanes Verfolgen mehrerer Hypothesen kurzzeitig verdeckte Objekte wiederfinden zu können. Zudem können multi-modale Verteilungen, die innerhalb der Objektverfolgung häufig im Falle von Verdeckungen vorkommen, dargestellt und behandelt werden.

In [43] wurden Mean-Shift, Mean-Shift in Kombination mit einem Kalman Filter und Partikel Filter gegenübergestellt. Es zeigte sich, dass ein Partikel Filter bei abrupten Bewegungen und schwierigen Bedingungen, wie Teilverdeckungen und Skalierungsänderungen die besten Ergebnisse erzielen kann.

## 2.3 Angewandte Methode und Mehrwert der Arbeit

In dieser Arbeit fiel die Wahl auf einen merkmalsbasierten Trackingalgorithmus um die Probleme von Verdeckungen in Verkehrsszenen zu behandeln. Als Bildmerkmale werden Farbe und Kanten verwendet um ein echtzeitfähiges Tracking in verschiedenen Situationen zu gewährleisten. Die Bildmerkmale werden dabei innerhalb von rechteckigen Templates berechnet, welche für die Darstellung von Fahrzeugen ausreichen. In Szenen, in denen die Objekte nur schlecht durch Rechtecke repräsentiert werden können, kann mit einer zusätzliche Gewichtungsfunktion der Fokus auf die relevanten Bereiche des Objekts gelegt werden.

Es wird ein Trackingalgorithmus für Verkehrsszenen vorgestellt, der im Gegensatz zu den häufig verwendeten *Tracking-by-Detection*-Verfahren unabhängig von der Verwendung eines Hintergrundschätzers ist. Angestrebt wird dabei ein modulares Trackingsystem bei dem lediglich die Detektion und Initialisierung der Objekte durch einen Objektdetektor erfolgt. Die Detektion kann dabei durch einen Hintergrundschätzer erfolgen. Dieser muss jedoch nur in manuell definierten Bildregionen und nicht im gesamten Bild aktiv sein.

Als Trackingverfahren wird ein probabilistischer Ansatz anhand eines Partikel Filters gewählt. Das Multihypothesentracking eignet sich besonders im Fall von kurzzeitigen Verdeckungen, in denen teilweise mehrere Haupthypothesen simultan verfolgt werden können. Zudem eignet es sich bei Verwendung eines entwickelten, nicht-linearen Bewegungsmodells (*CVR*), das zusätzlich die Rotation eines Objekts berücksichtigt. Schließlich eignet sich der strikt getrennte Aufbau des Partikel Filters um ohne großen Aufwand auf unterschiedlich verwendete Modelle zurückgreifen zu können, damit das Tracking auf verschiedene Anwendungen angepasst werden kann.

Neben einem neuen Bewegungsmodell werden detailliertere Beobachtungsmodelle verwendet, die auf eine Unterteilung der Objektmodelle in mehrere Blöcke basieren. Diese ermöglichen ein robustes Schätzen der Objektgröße. Durch das simultane Tracking mehrerer Objekte können Teil- und Kompletterdeckungen direkt auf das neue Modell angewendet werden, indem verdeckte Bereiche von der Merkmalsextraktion ausgeschlossen werden.

Zusätzlich soll der Vorteil einer stationären Kamera in das Tracking einfließen. Durch eine geometrische Berechnung der Objektgröße in jeder Bildposition kann auf eine Schätzung der

Objektskalierung innerhalb des Trackingverfahrens verzichtet werden. Ebenso soll eine robuste Vorhersage komplett verdeckter Objekte erfolgen, die auf der bekannten Bildgeometrie basiert.



---

### Mathematische Grundlagen des Partikel Filters

---

In diesem Kapitel sollen die theoretischen Grundlagen eines Partikel Filters erarbeitet werden. Ausgehend von einem rekursiven Bayes-Filter, welches die Grundlage für probabilistische Filterungen bildet, werden Kalman Filter und Partikel Filter erläutert und auf deren Unterschiede, sowie Stärken und Schwächen eingegangen. Die Anwendung eines Partikel Filters für Objektverfolgung (engl: *tracking*) ist neben den Bereichen Statistik und Robotik nur ein Beispiel. Dieses Kapitel widmet sich der mathematischen Beschreibung des Verfahrens, die Anwendung des Partikel Filters für das Trackingverfahren erfolgt in Kapitel 5.

#### 3.1 Rekursive Bayes-Filter

Die Einsatzgebiete von Bayes-Filtern sind vielseitig. Prinzipiell kann es überall dort angewendet werden, wenn es darum geht, über ein rekursives Schema Parameter zu schätzen, die einer weitestgehend bekannten Dynamik unterliegen und Beobachtungen des Systems, auf deren Grundlage diese Zustandsparameter geschätzt werden, mit einer gewissen Unsicherheit behaftet sind. Dabei können vergangene Zustände geglättet, aktuelle geschätzt und sogar zukünftige prädiziert werden. Dies kann für die Vorhersage von Finanzindizes, eine robuste Positionierung von Robotern, in der Navigation oder auch für die Objektverfolgung in Videosequenzen verwendet werden. Rekursiv im Sinne der Objektverfolgung bedeutet hier, dass die Lokalisierung Bild für Bild durchgeführt wird und nur abhängig vom Zustand des letzten Zeitpunktes ist. Dies gewährt eine rechen-effiziente Implementierung und ermöglicht Echtzeitfähigkeit. Notwendig für den Einsatz eines Bayes-Filters ist es zum einen, Informationen über das Dynamikmodell und das Beobachtungsmodell zu besitzen. Dabei beschreibt das Dynamikmodell die Evolution des Systems mit der Zeit und das Beobachtungsmodell in welchem Zusammenhang Beobachtungen bzw. Messungen mit dem System stehen. Zum anderen müssen diese beide Modelle in einer probabilistischen Form ausgedrückt werden können. Ausführliche Literatur über Bayes-Filter findet

sich u.a. in [22] auf dessen Grundlage die folgenden Herleitungen basieren.

Das Dynamik-, oder auch Bewegungsmodell, beschreibt, wie sich der Zustandsvektor zwischen zwei Zeitpunkten  $k$  entwickelt und hat die Form

$$x_k = f_k(x_{k-1}, v_{k-1}), \quad (3.1)$$

wobei  $k > 0$  nur gültige Zeitpunkte zu Beginn der Messung annehmen kann.  $f_k$  stellt eine bekannte (lineare oder nicht-lineare) Funktion dar.  $\{v_{k-1}, k \in \mathbb{N}\}$  beschreibt das Prozessrauschen als Form eines weißen Rauschens und damit die Unsicherheit des zugrunde liegenden Dynamikmodells. Eine äquivalente Formulierung von Gleichung (3.1) in Form einer probabilistischen Beschreibung stellt  $p(x_k|x_{k-1})$  dar. Dies wird oft auch als *Übergangsdichte* bezeichnet. Für den einfachen Fall, dass  $f$  linear und  $v$  einer Gaußverteilung folgt, ist auch  $p(x_k|x_{k-1})$  gaußverteilt.

Das Beobachtungsmodell stellt den Zusammenhang zwischen neuen Beobachtungen und dem aktuellen Zustand her und hat die Form

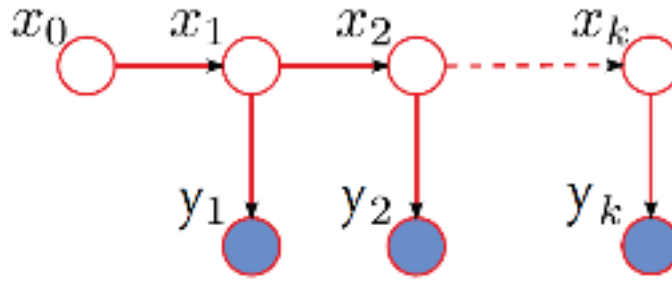
$$y_k = h_k(x_k, w_k), \quad (3.2)$$

wobei auch hier  $k > 0$  gelten muss.  $y_k$  stellt dabei einen Vektor neuer Beobachtungen zum Zeitpunkt  $k$  für den Zustand  $x_k$  dar und  $h_k$  ist wiederum eine bekannte (lineare oder nicht-lineare) Funktion, wobei  $w_k$  als Messrauschen bezeichnet wird und die Messunsicherheit darstellt. Auch dieses kann wieder als weißes Rauschen definiert werden. In probabilistischer Form kann Gleichung (3.2) als  $p(y_k|x_k)$  geschrieben werden. Auch hier gilt für den Spezialfall, falls  $h_k$  linear ist und  $w_k$  einer Gaußverteilung folgt, wird auch  $p(y_k|x_k)$  eine Gaußverteilung annehmen.

Sind die Funktionen des Bewegungsmodells  $f_k$  und des Beobachtungsmodells  $h_k$  bekannt, muss zusätzlich eine Initialisierung der Wahrscheinlichkeitsdichte definiert werden. Dies beinhaltet die Vorgabe der A-Priori Verteilung  $p(x_0)$  zum Zeitpunkt  $k = 0$ , also zu dem Zeitpunkt zu dem noch keine Beobachtungen verfügbar sind. Die gesamte probabilistische Beschreibung des Problems kann mit Hilfe dieser drei Zusammenhänge,  $p(x_0)$ ,  $p(x_k|x_{k-1})$  und  $p(y_k|x_k)$ , beschrieben werden.

Ein *Bayes-Schätzer* beschreibt die A-Posteriori Wahrscheinlichkeitsdichtefunktion (engl. *probability density function*: pdf)  $p(x_k|y_{1:k})$  eines Zustandsvektors  $x_k$  unter Berücksichtigung aller verfügbaren Beobachtungen  $y_{1:k}$  bis zum Zeitpunkt  $k$ .

Ausgehend von einer bekannten Wahrscheinlichkeitsdichtefunktion  $p(x_{k-1}|y_{1:k-1})$  kann diese mit Hilfe eines *rekursiven Bayes-Filter* sequentiell unter Verwendung des Dynamikmodells ( $p(x_k|x_{k-1})$ ) für den nächsten Zeitpunkt propagiert und bei Verfügbarkeit von neuen Beobachtungen mit Hilfe des Beobachtungsmodells ( $p(y_k|x_k)$ ) aktualisiert werden. Ziel ist es also die A-Posteriori Wahrscheinlichkeitsdichtefunktion ( $p(x_k|y_{1:k})$ ) zum Zeitpunkt  $k$  zu bestimmen. Dieses Vorgehen erfolgt im Sinne eines Bayes'schen Ansatzes in zwei Schritten, namentlich *Prädiktion* und



**Abbildung 3.1:** Darstellung eines *Hidden Markov Modells* für die Zustände  $x_{0:k}$  und den Beobachtungen  $y_{1:k}$ . Pfeile geben die Richtung der Abhängigkeiten an. Abbildung aus [39].

*Korrektur.*

**Annahme eines Hidden-Markov-Modells** Für die Herleitung eines Bayes-Filters wird von der Bedingung ausgegangen, dass die Struktur des Modells auf einem *Hidden Markov Modell* beruht. Diese Annahme geht zum einen davon aus, dass sich die Prädiktion des Objektzustands zum Zeitpunkt  $k$  ausschließlich aus dem Objektzustand des letzten Zeitpunktes  $k - 1$  ableiten lässt. Formal stellt sich dies wie folgt dar:

$$p(x_k | x_{0:k-1}) = p(x_k | x_{k-1}) \quad (3.3)$$

Die Tatsache, dass man somit keine Information über die gesamte Zustandshistorie benötigt, ermöglicht eine rechen- und speichereffiziente Schätzung des aktuellen Zustands unabhängig von der Länge der Datensequenz.

Die zweite Annahme beinhaltet die Unabhängigkeit der Beobachtungen. Neue Beobachtungen  $y_k$  hängen nur vom aktuellen Zustand  $x_k$  ab und werden von allen vorangegangenen Zuständen als stochastisch unabhängig angesehen. Mathematisch lässt sich dies folgendermaßen darstellen:

$$p(y_{1:k} | x_{0:k}) = p(y_k | x_k) \quad (3.4)$$

Abbildung (3.1) stellt die Annahmen des *Hidden-Markov Modells* graphisch dar. Die Zustände werden dabei als weiße Kreise ausgedrückt, Beobachtungen dagegen als blauen Kreise. Die Pfeile zeigen in die Richtung der Abhängigkeiten für die jeweiligen Zustände, bzw. Beobachtungen. Die obere Reihe repräsentiert den Übergang der Zustände, also die erste Annahme des *Hidden Markov Modells*, die untere Reihe die Unabhängigkeit der Beobachtungen und damit die zweite Annahme.

**Prädiktion** Ist die A-Posteriori Dichte  $p(x_{k-1} | y_{1:k-1})$  zum Zeitpunkt  $k - 1$  bekannt (für  $k = 0$  gilt:  $p(x_{k-1} | y_{1:k-1}) = p(x_0)$ ), so kann im ersten Schritt, der Prädiktion, mit Hilfe des Dynamikmodells aus Gleichung (3.1) die Entwicklung der Wahrscheinlichkeitsverteilung für den nächsten Schritt  $k$  prädiziert werden. Dabei wird noch keine aktuelle Beobachtung zum Zeitpunkt  $k$  berücksichtigt sondern nur eine Vorhersage aufgrund des bekannten Vorgängerzustandes sowie ein

Modell welches den zu Grunde liegenden Prozess möglichst genau beschreibt. Daraus resultiert die A-Priori Verteilung zum Zeitpunkt  $k$  und lässt sich folgendermaßen darstellen:

$$p(x_k|y_{1:k-1}) = \int_{-\infty}^{\infty} p(x_k, x_{k-1}|y_{1:k-1}) dx_{k-1} \quad (3.5)$$

Durch Verwenden der Kettenregel für Wahrscheinlichkeitsdichtefunktionen erhält man

$$p(x_k|y_{1:k-1}) = \int_{-\infty}^{\infty} p(x_k|x_{k-1}, y_{1:k-1}) p(x_{k-1}|y_{1:k-1}) dx_{k-1} \quad (3.6)$$

Unter Annahme der ersten Markov-Eigenschaft, dass der propagierte Zustand ausschließlich vom vorherigen Zustand abhängig ist, erhält man schließlich die *Chapman-Kolmogorov*-Gleichung

$$\underbrace{p(x_k|y_{1:k-1})}_{\text{Prior zu } k} = \int \underbrace{p(x_k|x_{k-1})}_{\text{Dynamik}} \underbrace{p(x_{k-1}|y_{1:k-1})}_{\text{Posteriori zu } k-1} dx_{k-1} \quad (3.7)$$

**Korrektur** Erhält man zum Zeitpunkt  $k$  eine neue Messung  $y_k$ , wird diese für den anschließenden Korrekturschritt verwendet um schließlich die A-Posteriori Wahrscheinlichkeitsdichte  $p(x_k|y_{1:k})$  zu berechnen. Dies geschieht im zweiten Schritt des Bayes-Filters, der sog. Korrektur. Diese basiert auf Grundlage des Bayes-Theorems, welches die Berechnung zweier bedingter Wahrscheinlichkeiten  $A$  und  $B$  in bekannter Form formuliert:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \quad (3.8)$$

In Bezug auf die Problemstellung eines Bayes-Filters kann dies folgendermaßen dargestellt werden:

$$p(x_k|y_{1:k}) = \frac{p(y_{1:k}|x_k)p(x_k)}{p(y_{1:k})} \quad (3.9)$$

Aufteilung von  $p(y_{1:k})$  in  $p(y_k, y_{k-1})$  führt zu

$$p(x_k|y_{1:k}) = \frac{p(y_k, y_{1:k-1}|x_k)p(x_k)}{p(y_k, y_{k-1})}$$

Durch Anwenden des Multiplikationssatzes für bedingte Wahrscheinlichkeitsdichtefunktionen ( $p(a, b|c) = p(a|b, c) \cdot p(b|c)$ ) kann der Zähler dargestellt werden als

$$p(x_k|y_{1:k}) = \frac{p(y_k|y_{1:k-1}, x_k)p(y_{1:k-1}|x_k)p(x_k)}{p(y_k, y_{1:k-1})}$$

Für den Nenner gilt  $p(a, b) = p(a|b) \cdot p(b)$

$$p(x_k|y_{1:k}) = \frac{p(y_k|y_{1:k-1}, x_k)p(y_{1:k-1}|x_k)p(x_k)}{p(y_k|y_{1:k-1})p(y_{1:k-1})}$$

Anwenden des Bayes-Theorems auf  $p(y_{1:k-1}|x_k)$  führt zu

$$p(x_k|y_{1:k}) = \frac{p(y_k|y_{1:k-1}, x_k)p(x_k|y_{1:k-1})p(y_{1:k-1})p(x_k)}{p(y_k|y_{1:k-1})p(y_{1:k-1})p(x_k)}$$

Durch Kürzen von  $p(x_k)$  und  $p(y_{1:k-1})$  erhält man

$$p(x_k|y_{1:k}) = \frac{p(y_k|y_{1:k-1}, x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}$$

Unter Annahme der zweiten Markov-Eigenschaft von unabhängigen Beobachtungen aus Gleichung (3.4), erhält man schließlich die optimale Bayes'sche Lösung:

$$\underbrace{p(x_k|y_{1:k})}_{\text{Posteriori}} = \underbrace{p(y_k|x_k)}_{\text{Likelihood}} \underbrace{p(x_k|y_{1:k-1})}_{\text{Prior}} / \underbrace{p(y_k|y_{1:k-1})}_{\text{Normalisierungs- konstante}} \quad (3.10)$$

Die Normalisierungseigenschaft des Nenners kann für folgende Betrachtungen unberücksichtigt bleiben. Fügt man schließlich Gleichung (3.7) in Gleichung (3.10) ein und vernachlässigt den Normalisierungsterm, kann die Bayes-Filterung in folgender kompakten Form dargestellt werden.

$$p(x_k|y_{1:k}) = p(y_k|x_k) \int_{-\infty}^{\infty} p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \quad (3.11)$$

Gleichung (3.11) liefert in Bezug auf die Berechnung der gesuchten A-Posteriori Wahrscheinlichkeitsdichtefunktion die optimale Bayes'sche Lösung. Um diese Gleichung zu lösen bedarf es einer geeigneten Repräsentation dieser Wahrscheinlichkeitsdichten, die ausreichend genau, schnell und einfach zu bestimmen sind. In einigen Fällen können die enthaltenden Integrale unter bestimmten Voraussetzungen analytisch mit Hilfe eines Kalman Filters gelöst werden. Sollte dies nicht möglich sein, können diese Integrale beispielsweise mit einem Partikel Filter approximiert werden.

## 3.2 Kalman Filter

Kalman Filter gehören zu den wohl bekanntesten Vertretern zur Zustandsschätzung dynamischer Systeme mit Hilfe von Bayes-Filtern. Für lineare Systeme liefert dieses erwartungstreue Filter eine optimale Lösung der Gleichung (3.11) und zielt auf die Minimierung hinsichtlich des Fehlerquadrates (bzw. Fehlerkovarianz) ab. Dies ist allerdings nur unter bestimmten Voraussetzungen gültig.

Für das System muss gelten, dass sowohl Bewegungsmodell, als auch Beobachtungsmodell in linearer Form vorliegen. Zudem müssen Prozess- und Messrauschen normalverteilt und die zu schätzende A-Posteriori Wahrscheinlichkeitsverteilung ebenfalls normalverteilt sein. Sind diese Voraussetzungen erfüllt, entsprechen alle auftretenden Wahrscheinlichkeitsverteilungen (egal zu welchem Zeitpunkt) Normalverteilungen. Ausführliche Literatur über Kalman Filter findet sich

u.a. in [23].

Die Zustandsschätzung mittels Kalman Filterung wird durch folgende Gleichungen bestimmt.

$$x_k = Ax_{k-1} + Bu_k + w_k \quad (3.12)$$

$$y_k = Hx_k + v_k \quad (3.13)$$

Gleichung (3.12) beschreibt dabei das Bewegungsmodell mit dem Zustandsvektor  $x$ , einer Transitionsmatrix  $A$ , einer Stellmatrix  $B$  mit zugehörigen Steuerparameter  $u$  (die im Falle von Tracking keine Verwendung finden) und einem Prozessrauschen  $w$ , das einer mehrdimensionalen Normalverteilung nach  $N(0, Q)$  mit Kovarianzmatrix  $Q$  entspricht.

Das Beobachtungsmodell in Gleichung (3.13) wird letztlich durch die Messmatrix  $H$  festgelegt, welche die aktuellen Messungen auf den Zustandsvektor beziehen und von einem Messrauschen  $v$  nach  $N(0, R)$  mit zugehöriger Kovarianzmatrix  $R$  überlagert wird.

Ziel des Kalman Filters ist es nun, die Schätzung zum Zeitpunkt  $k$  als lineare Kombination der vorangegangenen Schätzungen mit der neuen Beobachtung  $y_k$  zu formulieren. Durch Berücksichtigung des *Hidden Markov Modells* kann dies in einer rekursiven Struktur erreicht werden, da die Schätzung zum Zeitpunkt  $k - 1$  alle Informationen der vorangegangenen Beobachtungen  $y_{1:k-1}$  enthält. Anlehnend an die bereits erwähnte Struktur eines Bayes Filters, erfolgt auch der Ablauf eines Kalman Filters in zwei separaten Schritten: Prädiktion und Korrektur.

**Prädiktion** Im ersten Schritt wird die vorangehende Zustandsschätzung zum Zeitpunkt  $k - 1$  unter Verwendung des Dynamikmodells auf den aktuellen Zeitpunkt  $k$  prädictiert um eine Voraussage für den aktuellen Zeitpunkt zu erhalten. Da bei Trackingaufgaben keine Steuerparameter zu schätzen sind, werden diese in der folgenden Darstellung ignoriert. Die prädictierte A-Priori Schätzung erfolgt durch

$$\hat{x}_{[k|k-1]} = A\hat{x}_{k-1} \quad (3.14)$$

Für die Kovarianz gilt

$$\hat{P}_{[k|k-1]} = A\hat{P}_{k-1}A^T + Q_{k-1} \quad (3.15)$$

Die Schreibweise  $[k|k-1]$  drückt dabei aus, dass es sich bei den Parametern um prädictierte A-Priori Schätzungen handelt.

**Korrektur** Die Vorhersage wird mit der aktuellen Beobachtung korrigiert und es ergeben sich die A-Posteriori Schätzungen für den Zustandsvektor

$$\hat{x}_k = \hat{x}_{[k|k-1]} + \hat{K}_k I_k, \quad (3.16)$$

sowie für die Kovarianz

$$\hat{P}_k = \hat{P}_{[k|k-1]} - \hat{K}_k S_k \hat{K}_k^T. \quad (3.17)$$

mit der Innovation  $I_k$

$$\tilde{I}_k = y_k - H\hat{x}_{[k|k-1]} \quad (3.18)$$

$\hat{K}_k$  bezeichnet dabei das Kalman-Gain, welcher als Art Proportionalitätsfaktor zwischen Unsicherheit der Messwerte und Unsicherheit der Schätzung zu verstehen ist. Berechnet wird es durch folgende Gleichung

$$\hat{K}_k = \hat{P}_{[k|k-1]} H^T S_k^{-1}. \quad (3.19)$$

$S_k$  stellt die Residualmatrix dar

$$S_k = H\hat{P}_{[k|k-1]} H^T + R_k \quad (3.20)$$

Die Innovation  $I_k$  in Gleichung (3.18) ist ein Maß dafür, wie genau die prädizierte Schätzung den aktuellen Beobachtungswert mit Hilfe der verwendeten Beobachtungsgleichung beschreibt. Für gute Vorhersagen fällt die Innovation klein aus, für schlechte Vorhersagen kann sie dagegen sehr große Werte annehmen. Dementsprechend müssen auch die Korrekturen an dem Zustandsvektor klein bzw. groß ausfallen.

Die Tatsache, dass das Kalman Filter auf lineare Zusammenhänge und normalverteiltes Prozess- und Messrauschen beruht kann bei Anwendungen, die diese Restriktionen nicht erfüllen zu schlechten Ergebnissen führen. Abhilfe können hier verschiedene Erweiterungen des Kalman Filters verschaffen.

Eine Erweiterung, die nichtlineare Modelle berücksichtigt, stellt das *Erweiterte Kalman Filter* (EKF) dar. Hier wird um den aktuellen Mittelwert und die aktuelle Kovarianz linearisiert. Genauer bedeutet dies, dass eine Taylorapproximation des Modells vorgenommen wird, die nach dem ersten Glied abgebrochen wird.

Für stark nicht-lineare Modelle kann ein *Unscented Kalman Filter* (UKF) verwendet werden.

Das Kalman Filter, sowie deren Erweiterungen sind jedoch alle auf normalverteilte Wahrscheinlichkeitsdichten angewiesen. Sollte die gesuchte Dichte nicht einer Normalverteilung entsprechen, liefern Kalman Filter schlechte Lösungen und es muss auf ein weiteres Verfahren zurückgegriffen werden.

### 3.3 Partikel Filter

Das Partikel Filter, Vertreter der sequenziellen Monte-Carlo-Methoden (SMC), bietet eine umfassendere Lösung des Problems aus Gleichung (3.11) als das Kalman Filter. Es gliedert sich in die Gruppe der sub-optimalen Filter ein und ist in der Lage die Probleme des eben erwähnten Kalman Filters zu umgehen.

Grundidee eines Partikel Filters ist es, beliebige Wahrscheinlichkeitsverteilungen durch gewich-

tete, diskrete Einzelpunkte (*Samples*, bzw. *Partikel*) zu approximieren und alle daraus notwendigen Schätzungen (Mittelwert, Mode, Varianz, ...) anhand dieser Partikel abzuleiten. Wenn die Anzahl der Partikel sehr groß wird, kann diese Monte-Carlo Charakterisierung als äquivalente Darstellung zur üblichen funktionalen Beschreibung und ein Partikel Filter als optimaler Bayes-Schätzer angesehen werden. Vorteil dieser Methode ist es, dass ein Partikel Filter in der Lage ist, Schätzungen mit nichtlinearen Modellen durchzuführen und durch die Darstellung beliebiger Wahrscheinlichkeitsverteilungen nicht auf Normalverteilungen angewiesen ist. Daher ist es auch möglich „schiefe“ oder selbst multi-modale Verteilungen darzustellen und diese gesondert zu behandeln. Dies bietet einen großen Vorteil im Bereich des Videotrackings, da hier, wie in Kapitel 6 dargestellt, häufig multi-modale Verteilungen durch Störobjekte in der Umgebung auftreten können. Die folgende Beschreibung des Partikel Filters basiert auf den ausführlichen Beschreibungen in [45], [2], [20], [19], [14].

Für die folgende Darstellung sei  $x_{0:k}$  die Zustandshistorie  $\{x_0, \dots, x_k\}$  und  $y_{1:k}$  die Beobachtungshistorie  $\{y_1, \dots, y_k\}$ . Ausgehend von der Beschreibung der *Monte-Carlo-Integration* und dem *Importance Sampling* wird die Grundlage aller Partikel Filter hergeleitet, das *Sequential Importance Sampling* (SIS). Darauf basierend können Vereinfachungen gewählt werden aus dem das *Sequential Importance Resampling* (SIR) Filter resultiert, auf dem der hier gewählte Tracking Algorithmus basiert.

**Monte Carlo Integration** Gesucht sei die A-Posteriori Wahrscheinlichkeitsverteilung von  $p(x_k|y_{1:k})$ . Weiterhin ist es das Ziel, dessen Erwartungswert bzw. Maximum abzuleiten, das wiederum den geschätzten A-Posteriori Zustandsvektor beschreibt, welcher beim Tracking der neuen geschätzten Position des Objekts im Bild entspricht.

Ist  $p(x_k|y_{1:k})$  bekannt, lässt sich der Erwartungswert  $E$  der Verteilung im kontinuierlichen Fall auf herkömmliche Weise berechnen:

$$E[x_k|y_{1:k}] = \int x_k p(x_k|y_{1:k}) dx_k \quad (3.21)$$

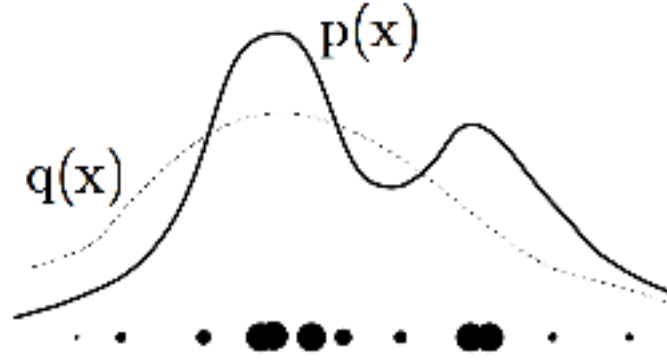
Weiterhin ist bekannt, dass sich die A-Posteriori Wahrscheinlichkeitsverteilung im Sinne einer Monte-Carlo Approximation als diskrete Verteilung darstellen lässt. Hierfür werden unabhängige, zufällige Samples gezogen, für die  $x_k^i \sim p(x_k|y_{1:k})$  gelten soll.  $p(x_k|y_{1:k})$  lässt sich dann folgendermaßen approximieren:

$$p(x_k|y_{1:k}) \approx \frac{1}{N} \sum_{i=1}^N \delta(x_k - x_k^i) \quad (3.22)$$

$\delta(\cdot)$  ist dabei die Delta-Dirac Funktion und  $N$  die Anzahl der Partikel. Für den Erwartungswert unter Annahme einer Monte-Carlo Approximation gilt demnach:

$$E[x_k|y_{1:k}] \approx \frac{1}{N} \sum_{i=1}^N x_k^i \quad (3.23)$$





**Abbildung 3.2:** Prinzip des *importance samplings* im eindimensionalen Fall. Approximation von  $p(x)$  erfolgt durch diskrete, gewichtete Samples aus  $q(x)$ . Gewichte werden durch ihre Punktgröße beschrieben. Abbildung aus [52]

Kann  $p(x_k|y_{1:k})$  als diskrete Verteilung dargestellt werden, ist es also möglich den Erwartungswert und damit den geschätzten Zustand aus diskreten Stichproben dieser Verteilung zu bestimmen. Problematisch ist es jedoch, Samples aus  $p(x_k|y_{1:k})$  zu ziehen, weil die Verteilung zunächst unbekannt ist. Einzig möglich ist die Berechnung von Likelihood-Wahrscheinlichkeiten  $p(y_k|x_k)$  für beliebige (gültige) Zustände  $x_k$ . Um daraus  $p(x_k|y_{1:k})$  zu approximieren, müsste der komplette Zustandsraum mit Partikel abgedeckt werden. Dies ist extrem rechenaufwändig und ineffizient, da  $p(y_k|x_k)$  für die meisten Zustände gegen Null geht. Wenn es möglich ist die wichtigen Teile der Wahrscheinlichkeitsverteilung vorherzusagen, kann der Suchraum für  $p(x_k|y_{1:k})$  eingeschränkt, und mit Hilfe von wenigen Samples dargestellt werden.

**Importance sampling (IS)** Ein Verfahren welches für diesen Zweck eingeführt werden kann ist *importance sampling*. Dies beinhaltet eine Einführung einer zusätzlichen, bekannten Wahrscheinlichkeitsverteilung, der sog. *importance sampling distribution*,  $q(x_{0:k}|y_{1:k})$ , die zu  $p(x_{0:k}|y_{1:k})$  möglichst proportional sein soll. Zudem sollte sie möglichst so gewählt sein, dass aus ihr einfach Stichproben gezogen werden können, beispielsweise normalverteilt. Anstatt nun Samples aus  $p(x_{0:k}|y_{1:k})$  zu ziehen, zieht man sie aus  $q(x_{0:k}|y_{1:k})$ . Durch eine Gewichtung der einzelnen Samples kann die Auswertung dann aus  $q(x_{0:k}|y_{1:k})$  und nicht aus  $p(x_{0:k}|y_{1:k})$  erfolgen (siehe auch Abbildung (3.2)). Hierfür werden nun unabhängige, zufällige Samples aus der *importance sampling distribution* gezogen  $x_k^i \sim q(x_k|y_{1:k})$ . Allgemein berechnet sich der Erwartungswert  $E[x_k|y_{1:k}]$  mit Hilfe des *importance samplings* folgendermaßen:

$$\begin{aligned}
 E[x_k|y_{1:k}] &= \int x_k p(x_k|y_{1:k}) dx_k \\
 &= \int \left[ x_k \frac{p(x_k|y_{1:k})}{q(x_k|y_{1:k})} q(x_k|y_{1:k}) \right] dx_k \\
 &= \int x_k w(x_k) q(x_k|y_{1:k}) dx_k
 \end{aligned} \tag{3.24}$$

mit  $w(x_k)$  als sogenannte *importance weights*

$$w(x_k) \approx \frac{p(x_k|y_{1:k})}{q(x_k|y_{1:k})} \quad (3.25)$$

Im Sinne einer Monte-Carlo-Approximation kann somit der Erwartungswert aus  $N$  diskreten, gewichteten Samples der *importance sampling distribution* bestimmt werden:

$$E[x_k|y_{1:k}] \approx \frac{1}{N} \sum_{i=1}^N x_{0:k}^i w_k^i \approx \sum_{i=1}^N x_{0:k}^i \tilde{w}_k^i, \quad (3.26)$$

wobei  $\tilde{w}_k^i$  die normierten *importance weights* darstellen, welche sich zu eins aufsummieren

$$\tilde{w}_k^i = \frac{w(x_{0:k}^i)}{\sum_{j=1}^N w(x_{0:k}^j)} \quad (3.27)$$

Somit kann die A-Posteriori Wahrscheinlichkeitsverteilung  $p(x_{0:k}|y_{1:k})$  anhand diskreter, gewichteter Samples der *importance sampling distribution* approximiert werden:

$$p(x_{0:k}|y_{1:k}) \approx \sum_{i=1}^N \tilde{w}_k^i \delta(x_{0:k}^i - x_{0:k}) \quad (3.28)$$

Mit Hilfe des IS können nun unzureichend bekannte Wahrscheinlichkeitsdichten auf Basis von bekannten ähnlichen Verteilungen abgeleitet werden. Jedoch ist die Effizienz des IS abhängig von der Wahl der *importance sampling distribution*. Diese muss so gewählt werden, dass die bedeutenden Teile von  $p(x_{0:k}|y_{1:k})$  durch  $q(x_{0:k}|y_{1:k})$  abgedeckt werden. Sollte der Erwartungswert von  $p(x_{0:k}|y_{1:k})$  nicht innerhalb der *importance sampling distribution* liegen, kann dieser auch nicht bestimmt werden, und damit keine erwartungstreue Approximation erfolgen. Somit sollte die Wahl der *importance sampling distribution* mit Bedacht gewählt werden.

**Sequential importance sampling (SIS)** Die Darstellung in Gleichung (3.28) liefert jedoch noch keine rekursive Berechnung von  $p(x_k|y_{1:k})$ . Bisher werden pro Zeitpunkt alle Beobachtungen und Zustände von  $k = 0$  bis  $k$ , sowie die Berechnung aller *importance weights* über die gesamte Zustandssequenz benötigt, damit  $p(x_{0:k}|y_{1:k})$  berechnet werden kann. Dies zieht zum einen eine immer zeitaufwändigere Berechnung nach sich und erfordert zum anderen die Speicherung aller Zustände  $x_{0:k}$  und Beobachtungen  $y_{0:k}$  bis zum Zeitpunkt  $k$ . Aus diesem Grund wird eine rekursive Darstellung von  $p(x_k|y_{1:k})$  angestrebt, für dessen Berechnung unter Berücksichtigung der Markov Eigenschaften nur die Zustände  $x_{k-1}$  zur Verfügung stehen müssen und die Beobachtungen lediglich vom aktuellen Zustand abhängen.

Dies kann durch das *sequential importance sampling* (SIS) erreicht werden, welches die Grundlage für die meisten Partikel Filter Variationen bildet.

Hierfür kann die *importance sampling distribution* folgendermaßen umgeformt werden:

$$q(x_{0:k}^i|y_{1:k}) = q(x_{0:k-1}^i|y_{1:k-1})q(x_k^i|x_{0:k-1}^i, y_{1:k}) \quad (3.29)$$

Die in Gleichung (3.29) und nachfolgend verwendeten Indizes  $i$  deuten an, dass nun  $i = 1, \dots, N$  konkrete Stichproben der Verteilungen betrachtet werden.

Ähnlich zur Herleitung des Korrektur-Schrittes eines Bayes-Filters in Kapitel 3.1 kann  $p(x_{0:k}^i | y_{1:k})$  mit Hilfe des Bayes-Theorems faktorisiert werden:

$$p(x_{0:k}^i | y_{1:k}) = p(x_{0:k-1}^i | y_{1:k-1}) \frac{p(y_k | x_{0:k}^i, y_{1:k-1}) p(x_k^i | x_{k-1}^i, y_{1:k-1})}{p(y_k | y_{1:k-1})} \quad (3.30)$$

Unter Annahme der Markov Eigenschaften und Vernachlässigung des Normierungsterms vereinfacht sich Gleichung (3.30) zu

$$\begin{aligned} p(x_{0:k}^i | y_{1:k}) &= p(x_{0:k-1}^i | y_{1:k-1}) \frac{p(y_k | x_k^i) p(x_k^i | x_{k-1}^i)}{p(y_k | y_{1:k-1})} \\ &\propto p(x_{0:k-1}^i | y_{1:k-1}) p(y_k | x_k^i) p(x_k^i | x_{k-1}^i) \end{aligned} \quad (3.31)$$

Für die rekursive Darstellung der *importance weights* kann Gleichung (3.31) und Gleichung (3.29) in Gleichung (3.25) eingesetzt werden und man erhält:

$$w_k^i \propto \frac{p(x_{0:k-1}^i | y_{1:k-1}) p(y_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_{0:k-1}^i | y_{1:k-1}) q(x_k^i | x_{k-1}^i, y_{1:k})}, \quad (3.32)$$

woraus

$$w_k^i = w_{k-1}^i \frac{p(y_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, y_{1:k})} \quad (3.33)$$

folgt. Die rekursive Darstellung ist nun ersichtlich. Die Berechnung der *importance weights* für den aktuellen Zeitpunkt  $k$  ist nur noch abhängig von den normierten *importance weights* des vorangehenden Zeitpunkts  $w_{k-1}^i$ .

Jedoch haben Kong et al. in [33] festgestellt, dass, egal welche *importance sampling distribution* verwendet wird, die Gewichte mit der Zeit degenerieren. Dies führt dazu, dass nach einiger Zeit die Gewichte fast aller Partikel nahezu null sind, und nur noch ein Partikel ein hohes Gewicht erhält. Das hat zur Folge, dass die Wahrscheinlichkeitsverteilung nicht mehr ausreichend approximiert werden kann und die Varianz der Schätzung zu jedem Zeitpunkt ansteigt. Dies wird auch als *degeneracy problem* bezeichnet. Um dieses Problem zu beheben, gibt es verschiedene Methoden:

- Wahl einer optimalen *importance sampling distribution*. Dies ist in den meisten Fällen allerdings nicht realisierbar, da die gesuchte Verteilung nicht ausreichend bekannt ist.
- Erhöhen der Partikelanzahl schwächt diesen Effekt der Degenerierung ab, kann ihm aber mit der Zeit nicht entgegenwirken. Zudem ist eine hohe Partikelanzahl in den meisten Anwendungen rechentechnisch nicht effizient.
- Ignorieren von Partikeln mit niedrigem Gewicht und vervielfältigen derjenigen mit hohem Gewicht. Dies wird als *Resampling* bezeichnet.

**Sequential importance resampling (SIR)** Im Falle eines *Sequential importance resampling* (SIR) Filters wird dem oben bereits dargestellten SIS Filter ein zusätzlicher Schritt, das sogenannte *Resampling*, hinzugefügt. Dieser dient zur Bewältigung des *degeneracy problems*. Nachdem die Schätzung aus der Wahrscheinlichkeitsverteilung erfolgt ist werden Partikel, die ein niedriges Gewicht erhalten haben und damit schlechte Schätzungen der A-Posteriori Wahrscheinlichkeitsverteilung beschreiben, für den nächsten Zeitpunkt verworfen. Dagegen werden Partikel, die ein hohes Gewicht und damit einer guten Schätzung entsprechen, dupliziert. Hierfür können die normierten Gewichte  $\tilde{w}_k^i$  als Wahrscheinlichkeiten angesehen werden. Durch ein „Ziehen mit Zurücklegen“ werden Partikel entsprechend ihrer Gewichte gezogen und bilden schließlich einen neuen, reduzierten Partikelsatz  $\{\tilde{x}_k^j, \tilde{w}_k^j\}_{j=1}^N$ , der ausschließlich Partikel enthalten soll, die wichtige Teile der Wahrscheinlichkeitsverteilung repräsentieren. Dabei können auch Partikel mit sehr hohen Gewichten öfters gezogen und damit dupliziert werden. Die Gewichte der gezogenen Partikel werden nach dem Resampling gleichverteilt  $\tilde{w}_k^j = N^{-1}$ . Mit Hilfe des Prozessrauschens werden diese redundanten Partikel im nächsten Prädiktionsschritt gestreut und in die Richtung verschoben, in der die Wahrscheinlichkeitsverteilung für den nächsten Zeitpunkt erwartet wird. Es existieren verschiedene Verfahren des Resamplings. Die bekanntesten darunter sind *Systematic Resampling*, *Residual Resampling*, *Multinomial Resampling*, und *Stratified Resampling*. In dieser Arbeit findet das *Systematic Resampling* Anwendung. Die Ergebnisse mit anderen Resampling Verfahren haben keine großen Veränderungen gezeigt und werden daher in der Evaluierung auch nicht untersucht. Für eine genauere Beschreibung der unterschiedlichen Methoden sei auf [18] verwiesen.

Die Implementierung des *Systematic Resamplings* ist in Algorithmus 1 dargestellt.

---

**Algorithmus 1 : SYSTEMATIC RESAMPLING**

---

**Input :** Partikelsatz  $\{x_k^i, w_k^i\}_{i=1}^N$  zum Zeitpunkt  $k$  mit Partikelzuständen  $x_k^i$  und zugehörigen normierten Partikelgewichten  $w_k^i$  und Partikelanzahl  $N$

**Output :** Neuer, geresamelter Partikelsatz  $\{\tilde{x}_k^j, \tilde{w}_k^j\}_{j=1}^N$

- 1 Initialisiere kumulative Verteilungsfunktion (KVF):  $c_1 = w_k^1$
- 2 **for**  $i \leftarrow 2$  **to**  $N$  **do**
- 3     Generiere KVF:  $c_i = c_{i-1} + w_k^i$
- 4 **end**
- 5 Generiere Laufindex für KVF:  $i = 1$
- 6 Generiere Startwert:  $u_1 \sim U[0, N^{-1}]$
- 7 **for**  $j \leftarrow 1$  **to**  $N$  **do**
- 8      $u_j = u_1 + N^{-1}(j - 1)$
- 9     **while**  $u_j > c_i$  **do**
- 10          $i = i + 1$
- 11     **end**
- 12     Weise neuen Zustand zu:  $\tilde{x}_k^j = x_k^i$
- 13     Weise neues Gewicht zu:  $\tilde{w}_k^j = N^{-1}$
- 14 **end**

---

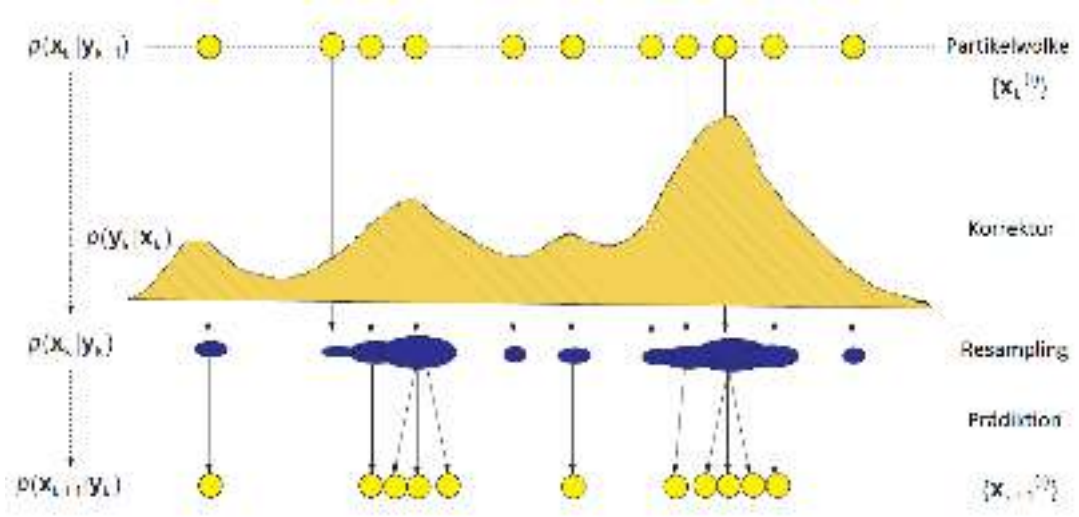


Abbildung 3.3: Ablauf eines Iterationsschrittes eines SIR-Filters

**Bootstrap Filter** Im Sinne eines *bootstrap filters* wird das Bewegungsmodell (aus dem Prädiktionsschritt) als *importance sampling distribution* verwendet:

$$q(x_k^i | x_{1:k-1}^i, y_{1:k}) = p(x_k^i | x_{k-1}^i) \quad (3.34)$$

Dadurch vereinfacht sich Gleichung (3.33) zu

$$w_k^i = w_{k-1}^i p(y_k | x_k^i) \quad (3.35)$$

Wird der Resampling-Schritt für jeden Zeitpunkt  $k$  (nach der Zustandsschätzung) angewendet, werden die Gewichte nach jedem Resampling-Schritt gleichverteilt. Dadurch vereinfacht sich die Berechnung der *importance weights* aus Gleichung (3.35) weiterhin zu

$$\tilde{w}_k^i = p(y_k | x_k^i) \quad (3.36)$$

Damit fließt Information aus dem Bewegungsmodell in die *importance distribution* ein und die neue Berechnung der *importance weights* ist nur noch abhängig von den aktuellen Likelihood-Wahrscheinlichkeiten.

Die Auswertung der A-Posteriori Verteilung kann auf unterschiedliche Arten erfolgen. In der Regel werden die Schätzverfahren MMSE (*Minimum-Mean-Square-Error*) und MAP (*Maximum-A-Posteriori*) verwendet:

$$\hat{x}_k^{MMSE} = E[x_k | y_{1:k}] \approx \sum_{i=1}^N x_k^i \tilde{w}_k^i \quad (3.37)$$

$$\hat{x}_k^{MAP} = \arg \max_{x_k} p(x_k | y_{1:k}) \approx \max_{x_k} w_k^i \quad (3.38)$$

Welchen Einfluss die unterschiedlichen Schätzverfahren auf die Auswertung von  $p(x_k|y_{k-1})$  haben wird in Kapitel 5.2 dargestellt.

Abbildung (3.3) zeigt den Ablauf eines Iterationsschrittes eines *bootstrap filters*. Gesucht ist die kontinuierliche Wahrscheinlichkeitsverteilung (gelb). Ausgehend von einer gleichverteilten prä-dizierten A-Priori Verteilung  $p(x_k|y_{k-1})$ , welche diskret durch den Partikelsatz  $\{x_k^i\}$  dargestellt wird (gelbe Punkte im oberen Bildausschnitt), wird allen Partikeln im Korrektur-Schritt ein Gewicht zugewiesen, welches auf der Likelihood-Wahrscheinlichkeit  $p(y_k|x_k^i)$  beruht. Dies stellt die A-Posteriori Verteilung dar. Die Größe der blauen Punkte stellt die Höhe der jeweiligen Partikelgewichte dar. Die gesuchte Wahrscheinlichkeitsverteilung kann nun anhand dieser Partikel mit ihren Gewichten approximiert werden und Schätzungen aus ihr berechnet werden. Anschließend werden durch das Resampling Partikel mit hohen Gewichten mehrfach gezogen, während Partikel mit niedrigen Gewichten seltener oder auch gar nicht gezogen werden. Die gezogenen Partikel werden anschließend durch das Bewegungsmodell für den nächsten Zeitpunkt prä-diziert. Durch das Prozessrauschen streuen sich die redundanten Partikel (gelbe Punkte im unteren Bildausschnitt). Diese bilden wiederum die A-Priori Verteilung, für den Zeitpunkt  $k+1$  und der Ablauf der Partikel Filters für einen Zeitpunkt ist beendet.

Die Anwendung des in Kapitel 3 vorgestellten Partikelfilters für die Objektverfolgung benötigt neben Initialisierung und einem Bewegungsmodell ein Beobachtungsmodell für die Berechnung der Partikelgewichte.

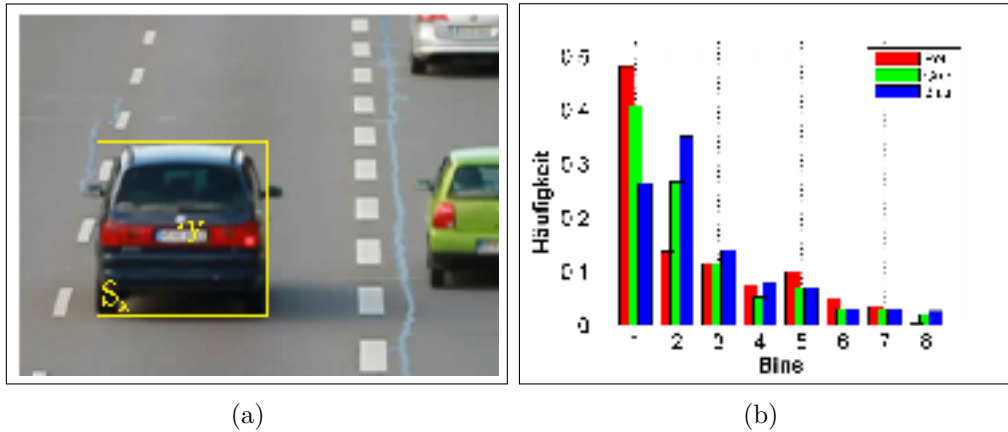
Es muss demnach ein Modell aufgestellt werden, welches für jeden Zustand (Position im Bild) ein Bildmerkmal extrahiert das mit dem gesuchten Objekt verglichen werden muss. Die Wahl der Bildmerkmale kann prinzipiell beliebig und pro Szene individuell gewählt werden. Die Einbindung in das Partikel Filter erfolgt, sofern ein einheitliches Ähnlichkeitsmaß zwischen den Bildmerkmalen eines Partikels und des Referenzobjekts besteht, für alle Bildmerkmale gleich.

Daher widmet sich dieses Kapitel zunächst ausschließlich der Extraktion dieser Bildmerkmale, welche in Kapitel 5 für die Anwendung des Partikel Filters verwendet werden.

Ausschlaggebend für die Wahl der Bildmerkmale ist eine möglichst eindeutige Beschreibung des gesuchten Objekts im Merkmalsraum. Dabei muss berücksichtigt werden, dass sich das Erscheinungsbild des Objekts im Laufe der Bildsequenz durch äußere Einflüsse (Beleuchtungsunterschiede, Verdeckungen), sowie durch Veränderungen des Objekts an sich (Rotationen) verändern kann. Dies sollte bereits bei der Wahl der Bildmerkmale beachtet werden.

Zusätzlich sollte die Merkmalsextraktion möglichst rechen-effizient durchführbar sein, da diese pro Partikel durchgeführt werden muss. Die Bildmerkmale sollten in vektorieller Form dargestellt werden, damit diese später in unkomplizierter Art mit denen des Referenzobjekts verglichen werden können.

In dieser Arbeit werden Bildmerkmale in Form von Farbe als Farbhistogramme und Kanten als Kantenrichtungshistogramme verwendet, die innerhalb von Bildregionen, sogenannten Templates, extrahiert werden. Die Extraktion der Bildmerkmale innerhalb der Templates wird nachfolgend erläutert. Im Folgenden sei  $S_x$  eine rechteckige Bildregion, die durch den Zustand  $x$  definiert wird, wobei  $x = \begin{bmatrix} r & c & w & h \end{bmatrix}$  die Position des Templates durch Bildzeile  $r$ , Bildspalte  $c$ , sowie dessen Ausmaße durch Breite  $w$  und Höhe  $h$  darstellt.



**Abbildung 4.1:** Merkmalsextraktion mittels Farbhistogrammen. (a) Darstellung eines Objekts mit Hilfe einer achsparallelen Bildregion  $S_x$  mit Mittelpunkt  $y$ . (b) Histogrammdarstellung für  $c = 3$  Farbkanäle und  $B = 8$  Bins für die Bildregion  $S_x$ .

## 4.1 Farbe

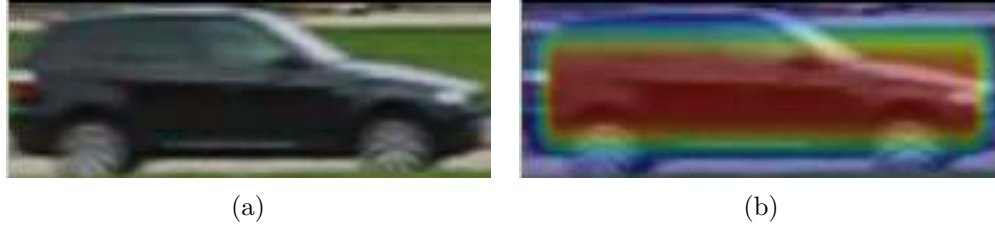
Ein intuitives Bildmerkmal zur Beschreibung eines Objekts ist dessen Farbverteilung, die zugleich rechen-effizient darstellbar ist, da die Videos in der Regel bereits als R,G,B-Bilder vorliegen. Aus diesen Gründen findet sie in den meisten gebräuchlichen Trackingverfahren Verwendung.

### 4.1.1 Histogramm Binning

Farbverteilungen werden dabei durch Histogramme beschrieben, worunter man grundsätzlich die Häufigkeit einzelner Farbwerte in einem Bild, bzw. Bildregionen versteht. Die Farbverteilung als Bildmerkmal hat den Vorteil einer sehr geringen Empfindlichkeit gegenüber Skalierung, Rotation und Translation des Bildinhaltes. Dabei wird die Verteilung oft in diskrete Intervalle eingeteilt, sogenannte Bins, statt die Häufigkeit für jeden einzelnen Farbwert zu berechnen. Eine große Anzahl an Bins zieht zwei Nachteile mit sich. Zum einen muss für jedes Partikel ein Histogrammvergleich zwischen Objektkandidat und Referenzobjekt berechnet werden, welches für eine große Anzahl an Bins einen erhöhten Rechenaufwand bedeutet. Zum anderen können durch leichte Veränderungen im Erscheinungsbild Drifts einiger Intensitätswerte zu benachbarten Bins starke Histogrammunterschiede aufweisen. Dadurch kann das Objekt nicht mehr präzise verfolgt werden. Aus diesen Gründen nimmt man eine Diskretisierung der Histogramme in eine geringe Anzahl an Bins vor. Auf der anderen Seite hat eine zu geringe Anzahl an Bins ebenfalls auch einen negativen Einfluss auf die Genauigkeit. Hier läuft man Gefahr, dass Objekte nicht mehr ausreichend genau beschrieben werden können und benachbarte Objekte oder gar der Hintergrund ähnliche Histogramme liefern. Der Einfluss der Binanzahl wird in Kapitel 6 untersucht.

Die Farbverteilung wird durch ein B-Bin Histogramm ausgedrückt, dessen Komponenten normiert werden, so dass die Summe aller Bins Eins ergibt. Für eine Region  $S_x$  im Bild ergibt sich das Farbhistogramm  $h_{S_x}^{c,j}$  des Farbkanals  $c$  durch Zuweisung des Intensitätswertes jedes Pixels





**Abbildung 4.2:** Einführung einer Gewichtungsfunktion für das Objektmodell (a) mit Hilfe eines Epanechnikov Kernels (b). Höhere Gewichtung der roten Bereiche bei der Erstellung der Farbbistogramme.

$x_i \in S_x$  zu einem Bin  $j = 1, \dots, B$

$$h_{S_x}^{c,j} = C_N \sum_{x_i \in S_x} \delta_j(b_{x_i}^c), \quad j = 1 \dots B, \quad (4.1)$$

wobei  $b_{x_i}^c \in \{1, \dots, B\}$  der Bin-Index ist, in das der Intensitätswert des Pixels an der Position  $x_i$  fällt.  $\delta(\cdot)$  stellt die *Kronecker-Delta* Funktion dar und  $C_N$  ist die Normalisierungskonstante damit  $\sum_{j=1}^B h_{S_x}^{c,j} = 1$  gilt [8].

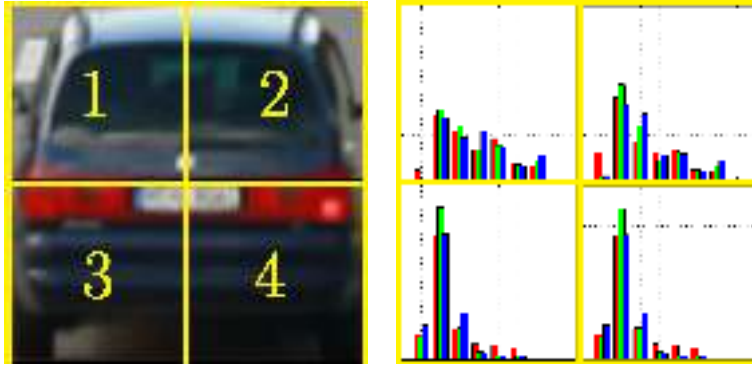
In Abbildung (4.1) werden beispielhaft die Farbbistogramme bei Verwendung von acht Bins (b) einer Bildregion  $S_x$  (a) innerhalb eines Bildes dargestellt.

#### 4.1.2 Einführung einer Gewichtungsfunktion

Durch die Beschränkung der Templatedarstellung in Form eines Rechteckes wird die wahre Form des Objektes nur in den seltensten Fällen vollständig erreicht werden. Dies hat zur Folge, dass viele Pixel des Hintergrundes mit in das Modell und damit in die Histogrammberechnung mit einbezogen werden können und dieses dadurch verfälschen. Um dem entgegenzuwirken, kann eine optionale Erweiterung des Farbbistogrammes durch die zusätzliche Einführung einer Gewichtungsfunktion verwendet werden [41]. Diese gewichtet Pixel die näher an dem Mittelpunkt der Region, und damit in der Regel mit größerer Sicherheit vom Objekt stammen höher, wobei Pixel die sich am Rand der Region befinden häufig vom Hintergrund stammen, und damit ein geringeres Gewicht erhalten. Als Gewichtungsfunktion kann beispielsweise ein *Epanechnikov Kernel* verwendet werden, wie in Abbildung (4.2) dargestellt :

$$k(d_r, d_c) = \begin{cases} \frac{3}{4} (1 - ((d_r/\sigma)^2 + (d_c/\sigma)^2)) & \text{falls } |(d_r/\sigma)^2 + (d_c/\sigma)^2| \leq 1 \\ 0 & \text{sonst} \end{cases} \quad (4.2)$$

wobei  $d_r$  und  $d_c$  den Abstand des Pixels zum Mittelpunkt der Region in zeilen- und spaltenrichtung beschreibt und  $\sigma$  die Bandbreite des Kernels darstellt. Besonders bei seitlichen Aufnahmen eines Autos wird deutlich, dass viele Hintergrundpixel in das Modell aufgenommen werden und die Einführung einer Gewichtungsfunktion hier sinnvoll ist (siehe Abbildung (4.2)). Ist die Form des Objekts im vornherein bekannt, können beliebige Kernels erstellt werden um ein besseres Modell des Objekts zu erhalten. Dies zielt allerdings auf eine Klassifizierung des Objekts nach



**Abbildung 4.3:** (a) Unterteilung der Bildregion  $S_x$  aus Abbildung (4.1) in  $T = 4$  Subregionen. (b) Dazugehörige Histogrammdarstellung für  $c = 3$  Farbkanäle und  $B = 8$  Bins.

der Detektion ab, die in dieser Arbeit nicht berücksichtigt wird.

Bei Verwendung einer Gewichtungsfunktion muss in Gleichung (4.1) der zusätzliche Kernel berücksichtigt werden:

$$h_{S_x}^{c,j} = C_N \sum_{x_i \in S_x} k \left( \frac{\|r_y - r_{x_i}\|}{h}, \frac{\|c_y - c_{x_i}\|}{w} \right) \delta_j(b_{x_i}^c), \quad j = 1 \dots B, \quad (4.3)$$

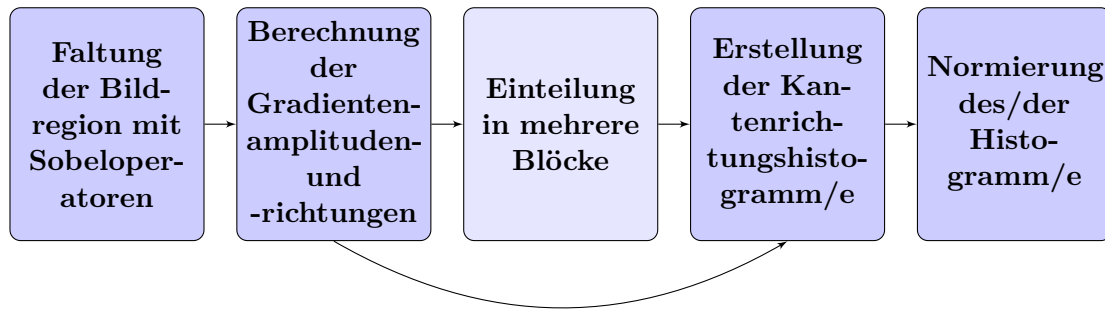
wobei  $r_y$  und  $c_y$  Zeile und Spalte des Mittelpunkts der Region  $S_x$ ,  $r_{x_i}$  und  $c_{x_i}$  Zeile und Spalte des untersuchten Bildpunktes und  $h$  und  $w$  die Höhe bzw. Breite des Templates darstellt.

### 4.1.3 Multi-Part Farbhistogramme

In Anlehnung an *FragTrack* [1] kann ein detaillierteres Modell des Objekts zu erstellt werden, indem Multi-Part Histogramme verwendet werden. Dabei wird die Bildregion  $S_x$  in  $t = 2, \dots, T$  gleichgroße Subregionen  $S_x^t$  unterteilt, für die jeweils ein Farbhistogramm berechnet wird. Gleichung (4.1) wird damit zu:

$$h_{S_x}^{c,j,t} = C_N \sum_{x_i \in S_x^t} \delta_j(b_{x_i}^c), \quad j = 1 \dots B, \quad (4.4)$$

Der Merkmalsvektor  $h_{S_x}^{c,j,t}$  hat damit die Länge  $c \cdot B \cdot T$  und ist damit  $T$ -mal so lang wie ohne Unterteilung in Subregionen. Dies gewährt zum Einen mehr Informationen über das Objekt, erhöht allerdings die Rechenzeit, wie in Kapitel 5 gezeigt wird. Abbildung (4.3) stellt die Unterteilung der Region  $S_x$  aus Abbildung (4.1) in vier Subregionen mit den dazugehörigen Histogrammen für 8x8 Bins der Farbkanäle R,G,B dar. Die Unterteilung in Subregionen bietet nicht nur eine bessere Beschreibung der räumlichen Anordnung der Farbverteilung, welche sich positiv auf die Schätzung der wahren Größe des Objekts auswirkt, es gewährt zudem die Möglichkeit gezielt mit Verdeckungen umgehen zu können. Da es im Straßenverkehr vermehrt zu Teil- und Kompletterdeckungen kommt, können durch die Verwendung von Multi-Part Histogrammen verdeckte Regionen bei der Suche des Objekts niedriger gewichtet, oder sogar ausgeschlossen werden. Nähere Details dazu werden in Kapitel 5 vorgestellt.



**Abbildung 4.4:** Ablauf zur Erstellung der Kantenrichtungshistogramme.

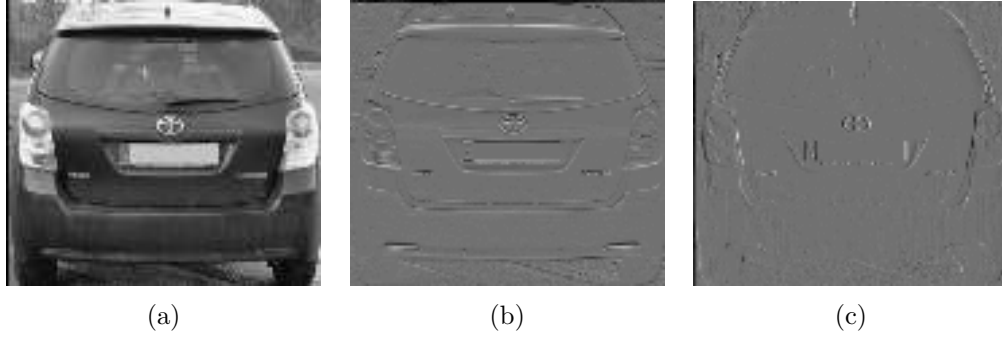
## 4.2 Kanten

Um die räumliche Struktur des Objekts zu berücksichtigen, kann ein weiteres Bildmerkmal herangezogen werden, welches sich auf die Gradienten des Objekts bezieht. Hierfür werden Kantenrichtungshistogramme des Objekts berechnet. Es wird dabei das Auftreten von Gradientenorientierungen in der Bildregion  $S_x$  gezählt. Ähnlich der Verwendung von Farb- oder Grauwert-histogrammen werden die Richtungen der Kanten einer Bildregion in Bins eingeteilt und entsprechend ihrer Kantenstärke aufsummiert; dadurch erhält man ein Maß der ausgeprägtesten Kantenrichtungen im Bildausschnitt. Dieses Bildmerkmal eignet sich besonders in Szenen in denen Objekte nicht mehr eindeutig durch ihre Farbverteilung dargestellt werden können, insbesondere bei Beleuchtungsänderungen in der Szene. Aber auch für Objekte die eine ähnliche Farbverteilung wie der Hintergrund aufweisen, können durch die Repräsentation ihrer Kanten eindeutiger dargestellt werden. Allerdings geht hierbei die Information der Objektfarbe verloren und eine Störung durch benachbarte, räumlich ähnlich ausgeprägte Objekte erhöht die Gefahr das gesuchte Objekt zu verlieren. Um die Robustheit zu erhöhen und detaillierte räumliche Aspekte zu erhalten, können, ähnlich den Multi-Part Histogrammen, Kantenrichtungshistogramme in mehrere Blöcke unterteilt werden.

### 4.2.1 Kantenrichtungshistogramm

Im Folgenden werden die einzelnen Schritte dieser Merkmalsextraktion erläutert. Die Vorgehensweise orientiert sich dabei an das vorgestellte Verfahren in [41]. Abbildung (4.4) stellt den schematischen Ablauf dar.

**Faltung der Bildregion mit Sobeloperatoren** Für die Bestimmung der Gradienten, muss die Bildregion  $S_x$  zunächst mit einer Ableitungsmaske gefaltet werden. Hierfür werden horizontale und vertikale Sobeloperatoren  $K_c$  und  $K_r$  verwendet, mit denen die Bildregion gefaltet wird.



**Abbildung 4.5:** Faltung der Bildregion (a) mit Sobeloperatoren. (b) Horizontales- und (c) vertikales Gradientenbild.

Diese besitzen folgende Form:

$$K_c = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, K_r = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}, \quad (4.5)$$

Die Gradientenbilder in horizontaler und vertikaler Richtung  $G_c$  und  $G_r$  erhält man durch Faltung der Bildregion  $S_x$  mit den Sobeloperatoren:

$$G_c(r, c) = K_c * S_x(r, c) \quad (4.6)$$

$$G_r(r, c) = K_r * S_x(r, c) \quad (4.7)$$

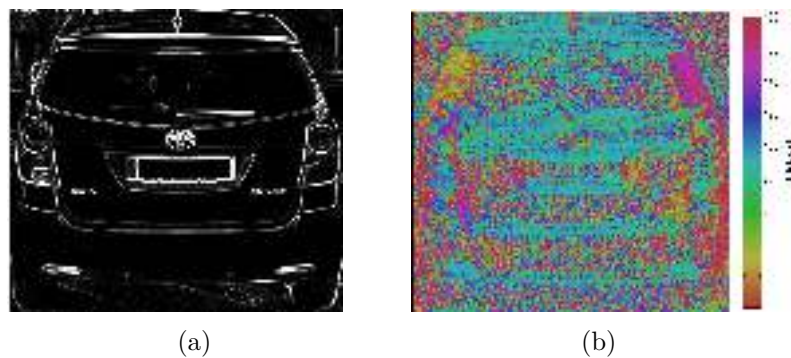
Abbildung (4.5) stellt die daraus resultierenden Gradientenbilder für eine Bildregion  $S_x$  dar. Dabei muss die Randbehandlung beachtet werden. Für Filtermasken der Größe  $[3 \times 3]$  gehen die Informationen des äußeren Randes (1 Pixel) verloren. Da diese Bereiche oft starke Kanten aufweisen, sollte die Größe der untersuchten Bildregion um einen Pixel pro Rand vergrößert werden.

**Berechnung der Gradienten** Nach der Faltung können pro Pixel der Bildregion die Gradientenamplitude  $M(r, c)$  und -orientierung  $\Theta(r, c)$  berechnet werden.

$$M(r, c) = \sqrt{G_r^2(r, c) + G_c^2(r, c)} \quad (4.8)$$

$$\Theta(r, c) = \arctan \frac{G_c(r, c)}{G_r(r, c)} \quad (4.9)$$

Ungerichtete Gradientenorientierung liefern robustere Ergebnisse. Somit werden nur Winkel zwi-



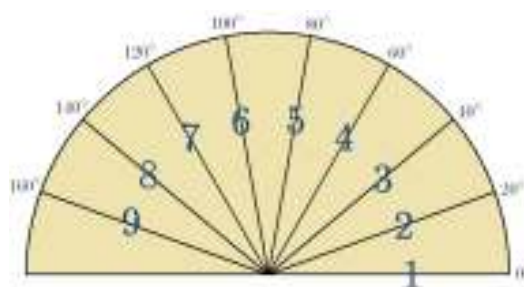
**Abbildung 4.6:** Gradientenamplituden- (a) und orientierungsbild (b).

schen  $0^\circ \leq \Theta \leq 180^\circ$  betrachtet:

$$\Theta(r, c) = \begin{cases} \Theta(r, c), & \text{falls } \Theta(r, c) > 0^\circ \\ \Theta(r, c) + 180^\circ, & \text{falls } \Theta(r, c) < 0^\circ \end{cases} \quad (4.10)$$

Abbildung (4.6) stellt das Gradientenamplituden- sowie orientierungsbild dar.

**Berechnung der Gradientenamplitudenhistogramme** Im nächsten Schritt wird pro Pixel die Orientierung entsprechend ihres Winkels einem Bin zugewiesen. Jeder Bin,  $j$ , deckt damit ein Winkelintervall gleicher Größe ab. Die Wahl der Bin-Anzahl kann zwar manuell bestimmt werden, jedoch empfiehlt sich eine Einteilung in neun Histogrammbins. Abbildung (4.7) zeigt die Aufteilung der Histogrammbins für  $j = 9$ . Aufaddiert in den entsprechenden Bin wird die Gradientenamplitude des Farbkanals mit der größten Gradientenstärke. Dies setzt voraus, dass für jeden Farbkanal eine Faltung und Berechnung der Gradientenamplituden- und orientierungen vorausgegangen ist. Alternativ können zur Effizienzsteigerung die Gradienten des Grauwertbildes verwendet werden. In diesem Fall muss die Bereitstellung des Gradientenamplituden- und orientierungsbildes nur noch einmal statt dreimal durchgeführt werden. In Kapitel 6 wird untersucht, welchen Einfluss es hat, alle Farbkanäle zu berücksichtigen. Damit Aliasing Effekte vermieden werden, wird die Gradientenstärke entsprechend ihres Abstandes ihrer Orientierung zu beiden benachbarten Bins aufgeteilt. Ist beispielsweise  $M = 10$  und  $\Theta = 85^\circ$  wird dem Bin  $j_5$  der Wert  $\frac{15}{20}M = 7.5$  aufaddiert, für  $j_6$  ergibt sich demnach  $\frac{5}{20}M = 2.5$ .



**Abbildung 4.7:** Einteilung der Gradientenorientierungen in 9 Bins

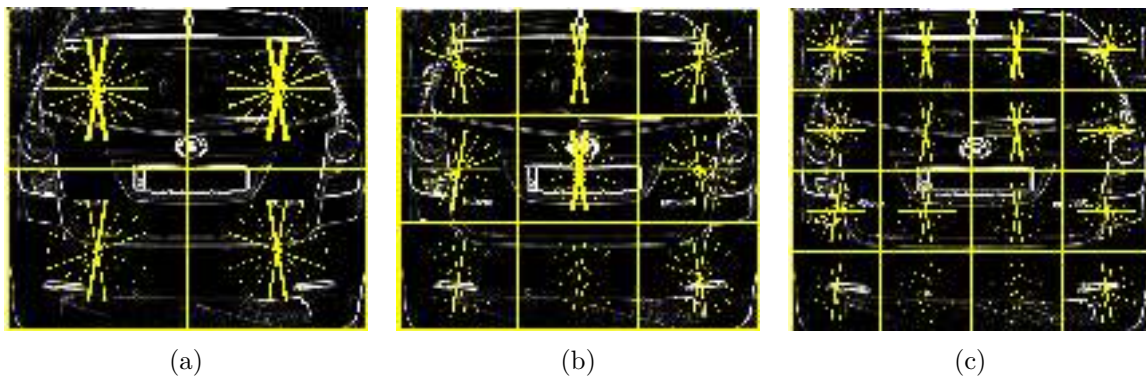


**Abbildung 4.8:** Kantenrichtungshistogramm für die Bildregion aus Abbildung (4.5) mit  $T = 1$  Block. Pfeilrichtung stellen die Kantenrichtungen für  $n = 9$  Bins dar, die Breite der Pfeile dagegen die jeweilige Kantenstärke.

Anschließend wird das Histogramm normiert, sodass die Summe der Bins Eins ergibt. Man erhält schließlich einen Vektor mit der Länge der Anzahl der Bins, wobei jedes Element die Stärke der auftretenden Kante für das jeweilige Intervall der Kantenrichtungen darstellt. Abbildung (4.8) stellt das Kantenrichtungshistogramm der untersuchten Bildregion für  $j = 9$  Bins dar. Je breiter die Pfeile, desto ausgeprägter sind die Kanten für die jeweilige Richtung. Es wird ersichtlich, dass horizontale und vertikale Kanten die am stärksten ausgeprägtesten des Objektmodells sind.

#### 4.2.2 Multi-Part Kantenrichtungshistogramme

Um einen genaueren räumlichen Eindruck des Objektes zu erhalten, kann, ähnlich dem Multi-Part Farbhistogramm, das Objekt in  $T$  Regionen, im folgenden Blöcke genannt, unterteilt werden. Die Berechnung der Kantenrichtungshistogramme erfolgt dann für jeden Block getrennt. Man erhält dadurch einen Merkmalsvektor mit der Länge der Anzahl der Bins multipliziert mit der Blockanzahl. Diese Darstellung ähnelt einem *Histogram of Oriented Gradients* (HOG), jedoch ohne den dort verwendeten Normierungsschritt [17]. Für die Normierung eignet es sich die Bins aller Blöcke gesamt zu normieren, sodass sich der gesamte Merkmalsvektor auf Eins summiert. Dadurch werden die einzelnen Blöcke nicht getrennt voneinander betrachtet, son-



**Abbildung 4.9:** Kantenrichtungshistogramme bei Verwendung einer unterschiedlichen Anzahl an Blöcken. (a)  $T = 4$ , (b)  $T = 9$  und (c)  $T = 16$ .

dern starke Kanten in einzelnen Blöcken haben Einfluss auf den gesamten Merkmalsvektor. Dies stabilisiert das Ergebnis und eignet sich bei Verdeckungen. Abbildung (4.9) stellt Kantenrichtungshistogramme für eine unterschiedliche Anzahl an Blöcken dar. Es wird eine deutlich detailliertere Beschreibung des Objekts durch die Erhöhung der Block-Anzahl ersichtlich.





---

### Anwendung eines Partikel Filters für die Verkehrsüberwachung

---

Folgendes Kapitel stellt die Vorgehensweise dar, die in Kapitel 4 vorgestellten Bildmerkmale in den Partikel Filter zu integrieren und daraus ein Trackingverfahren zu realisieren, welches sich auf die Verfolgung von Objekten im Straßenverkehr konzentriert.

Zu Beginn wird die Modellierung der Objektlokalisierung im Zustandsraum erläutert, gefolgt vom allgemeinen Ablauf eines rekursiven Filterungsschrittes des Partikel Filters. Anschließend werden Bewegungs- und Beobachtungsmodelle detaillierter behandelt und schließlich Vorgehensweisen für Teil- und Kompletต์verdeckungen, sowie ein Ansatz zur externen Bestimmung der Objektgröße vorgestellt.

Das in Kapitel 3 vorgestellte SIR-Filter stellt das Grundkonzept für die nachfolgenden Betrachtungen dar. Soll ein Partikel Filter für das Tracking verwendet werden, müssen folgende Modelle definiert werden:

- Erscheinungsbild des Objekts und zu schätzender Zustandsvektor  $x_k$
- Initialisierung zu  $k = 0$ : Vorgabe der A-Priori-Verteilung  $p(x_0)$  durch manuelle oder automatische Angabe der Objektposition  $x_0$  im ersten Frame, sowie Merkmalsextraktion  $q$  des Referenzobjektes.
- Wahl des Bewegungsmodells  $p(x_k|x_{k-1})$ : Prädiziert Partikel basierend auf den Zustand zum letzten Zeitpunkt  $k - 1$  für den nächsten Zeitpunkt  $k$ .
- Wahl des Beobachtungsmodells  $p(y_k|x_k)$ : Berechnet Partikelgewichte basierend auf den prädizierten Partikelpositionen aus  $p(x_k|x_{k-1})$  mit Hilfe von Ähnlichkeitsmaßen zwischen Objekthypothesen und Referenzobjekt für die gewählten Bildmerkmale.

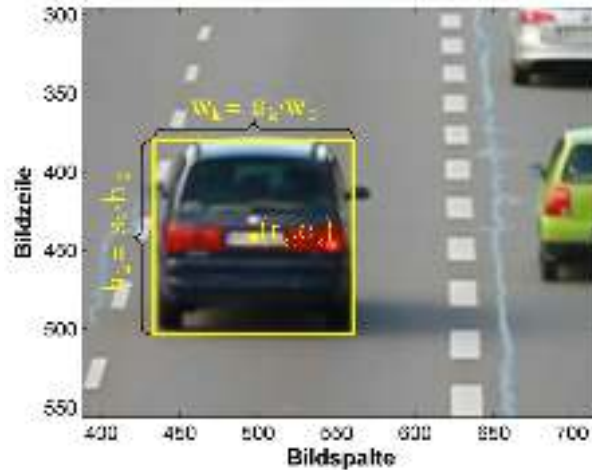


Abbildung 5.1: Darstellung des Objektmodells für den Zeitpunkt  $k$ .

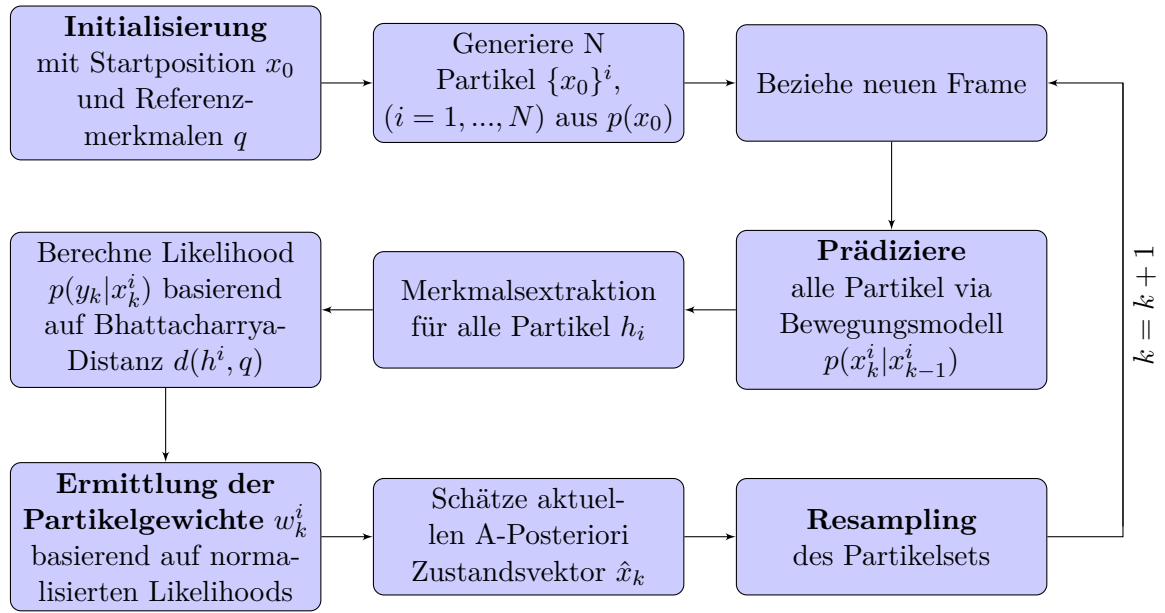
## 5.1 Modellbildung

Zunächst muss ein Modell des zu verfolgenden Objekts gebildet und ein Zustandsvektor definiert werden, mit dessen Hilfe das Objekt im Zustandsraum lokalisiert werden soll. Für die Wahl der Objektform gibt es, wie in Kapitel 2 bereits vorgestellt, mehrere Möglichkeiten. Das in dieser Arbeit vorgestellte Verfahren zur Objektverfolgung basiert auf einem merkmalsbasiertem Trackingalgorithmus. Die Merkmalsextraktion erfolgt dabei innerhalb von Bildregionen, den sogenannten Templates. Die Form des Templates kann theoretisch beliebig gewählt werden. Für die Anwendung im Straßenverkehr eignen sich in den meisten Fällen achsenparallele Rechteckmasken, da diese die Form von Vehikeln generell gut abdecken können. Die Größe der Objekte kann sich allerdings abhängig von der Szene und Aufnahmegeometrie ändern und muss zusätzlich berücksichtigt werden. Die Proportionen von Breite und Höhe des Templates können dabei als konstant angenommen werden, wodurch die Größenänderung durch einen Skalierungsfaktor  $s$  ausgedrückt werden kann.

Somit erfolgt die Lokalisation des Objektes im Bild  $k$  durch den Parametersatz  $\mathbf{x}_k = [r_k \ c_k \ s_k]^T$  mit Mittelpunkt in Bildzeile  $r_k$  und Bildspalte  $c_k$ , sowie der Breite  $w_k = s_k \cdot w_0$  und Höhe  $h_k = s_k \cdot h_0$ , wobei  $h_0$  und  $w_0$  die Objekthöhe und Breite bei Initialisierung des Objekts darstellt. Abbildung (5.1) stellt diesen Zusammenhang nochmals graphisch dar. Um zusätzlich die Geschwindigkeit des Objektes in der Bildebene zu berücksichtigen, kann der Zustandsvektor um die beiden Geschwindigkeiten  $\dot{r}_k$  und  $\dot{c}_k$  erweitert werden und ergibt sich schließlich zu  $\mathbf{x}_k = [r_k \ c_k \ \dot{r}_k \ \dot{c}_k \ s_k]^T$ .

## 5.2 Ablauf des Partikel Filters

Im Folgenden wird eine Übersicht über die Anwendung des in Kapitel 3.3 vorgestellten Partikel Filters für die Objektverfolgung vorgestellt. Dies dient einer kurzen Zusammenfassung über den Ablauf des Filters, dessen Details in den Folgekapiteln behandelt werden. Ein Ablaufdiagramm



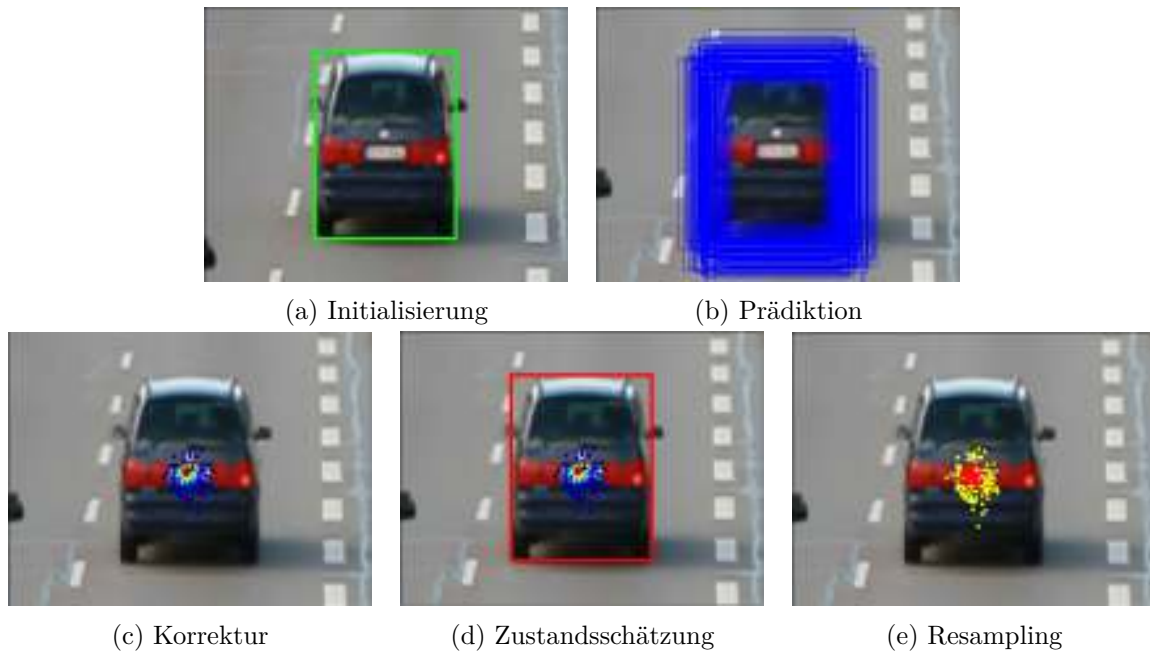
**Abbildung 5.2:** Ablauf des Partikel Filters (SIR) für Trackinganwendungen

das im Folgenden näher erläutert wird ist in Abbildung (5.2) dargestellt.

**Initialisierung** Zu Beginn eines jeden Trackers muss dieser zunächst initialisiert werden. Hierfür muss zunächst der Zustandsvektor zum Zeitpunkt  $k = 0$  vorgegeben werden. Dies kann entweder manuell oder mit Hilfe eines Detektors geschehen.  $r_0$  und  $c_0$  können anhand des Mittelpunkts des detektierten Objekts bestimmt werden (siehe auch Abbildung (5.3a)). Die Geschwindigkeiten  $\dot{r}_0$  und  $\dot{c}_0$  können bei unbekannter Bewegung des Objektes mit 0 initialisiert werden, wenn sich das Objekt nicht zu schnell bewegt und die Streuung der Partikel durch das Prozessrauschen ausreicht um das Objekt im nächsten Bild wiederfinden zu können. Innerhalb weniger Bilder kann der Tracker die Geschwindigkeit des Objekts anpassen. Ein robusteres Ergebnis bei schnellen detektierten Objekten kann durch die Vorgabe von  $\dot{r}_0$  und  $\dot{c}_0$  durch den Detektor erreicht werden. Dies kann beispielsweise durch die Berechnung des optischen Flusses geschehen. Dadurch kann eine präzisere Prädiktion der Partikel in den ersten Bildern erreicht, und die Gefahr das Objekt zu verlieren verringert werden.

Die Skalierung  $s_0$  wird zu Beginn als 1 definiert. Dies impliziert, dass Objektbreite und -höhe des Referenzobjektes zu  $k = 0$  für jeden Zeitpunkt an den Tracker überliefert werden muss, da die neue Objektgröße ausgehend von der Referenzgröße und Skalierung  $s_k$  geschätzt wird.

Weiterhin müssen die Bildmerkmale des Objekts bezogen werden. Diese dienen im Laufe des Trackings als Referenzmerkmale, nach denen in den Folgebildern gesucht werden soll. Die Wahl der Bildmerkmale kann für jede Anwendung unterschiedlich sein und theoretisch jegliche Art von Merkmalsextraktion beinhalten. Diese Arbeit beschränkt sich auf die in Kapitel 4 vorgestellten Merkmale in Form von Farb- bzw. Graustufen- und Kantenrichtungshistogrammen. Pro verwendetes Bildmerkmal erhält man einen Merkmalsvektor  $q$ . Die Dimension ist dabei abhängig



**Abbildung 5.3:** Visualisierung eines Iterationsschrittes des Partikel Filters für Trackinganwendungen. (a) Detektion des Objekts (grün) und Merkmalsextraktion. (b) Prädiktion der Partikel zu neuen Positionen (blau). (c) Berechnung der Partikelgewichte (Partikel repräsentieren hier die Mittelpunkte der Templates). Ansteigendes Partikelgewicht von blau nach rot. (d) Schätzung der aktuellen Objektposition aus der A-Posteriori Verteilung. (e) Neuer, reduzierter Partikelsatz (rot) nach dem Resampling aller Partikel (gelb) anhand ihrer Gewichte.

von den Parametern der Merkmalsextraktion (siehe Kapitel 4 für nähere Details).

**Erzeugung der Partikel** Ist die Position des Objektes für den ersten Zeitpunkt  $x_0$  definiert und wurden die Bildmerkmale für diese Region extrahiert, können die Partikel generiert werden. Hierfür werden  $N$  Partikel  $\{x_k^i\}_{i=1}^N$  erzeugt, die sich normalverteilt um  $x_0$  aufhalten. Die Streuung der Partikel wird durch die Wahl des Prozessrauschens definiert. Man erhält dadurch die A-Priori Verteilung  $p(x_0)$  zum Zeitpunkt  $k = 0$ .

Damit wurden alle Vorkehrungen zur Initialisierung des Partikel Filters getroffen und die sequentielle Filterung zur Objektverfolgung des detektierten Objekts kann begonnen werden. Die nachfolgenden Abschnitte bilden dabei einen Iterationsschritt der Filterung, der wie in Kapitel 3.3 aus *Prädiktion*, *Korrektur* und *Resampling* besteht.

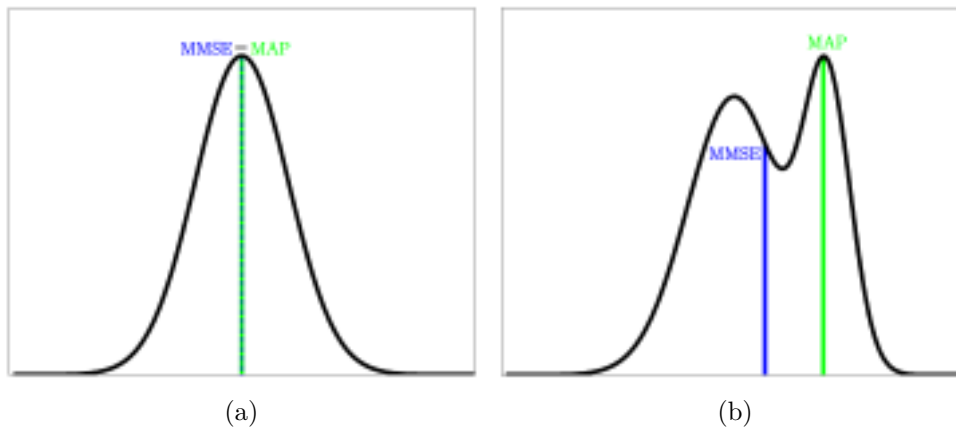
**Prädiktion** Beginnend mit dem Erhalt des neuen Frames zum Zeitpunkt  $k$  soll das Objekt nun in diesem Bild wiedergefunden werden. Demnach soll die neue Position, Geschwindigkeit in der Bildebene und Objektgröße, zusammengefasst im Zustandsvektor  $x_k$ , geschätzt werden. Hierfür werden die generierten Partikel mit Hilfe eines Bewegungsmodells zu neuen Zuständen bewegt, von denen man sich erhofft, dass die neue Objektposition durch diese neuen Zustände  $x_k^i$  abgedeckt wird (Abbildung (5.3b)). Damit die Partikel in die korrekte Richtung bewegt werden, ist ein Modell notwendig das die Bewegung des Objekts anhand der letzten geschätzten Positionen

möglichst präzise vorhersagen, aber auch eventuelle, abrupte Bewegungsänderungen abfangen kann. Dieser Schritt entspricht der *Prädiktion* eines rekursiven Bayes-Filters  $p(x_k^i | x_{k-1}^i)$ . In Kapitel 5.3 werden zwei Bewegungsmodelle detailliert vorgestellt. Jedes Partikel entspricht dabei einem Template an unterschiedlicher Position im Bild und mit unterschiedlicher Größe, ausgedrückt durch die Objektskalierung.

**Merkmalsextraktion und Berechnung der Partikelgewichte** Anschließend werden, basierend auf diesen prädizierten Zuständen, Partikelgewichte berechnet, die (bis auf einer Normalisierung) der Darstellung der approximierten, diskreten, A-Posteriori Wahrscheinlichkeitsverteilung entsprechen (siehe Abbildung (5.3c)). Hierfür müssen zunächst für jedes Partikel  $x_k^i$  Bildmerkmale extrahiert werden, und zwar dieselben (dies gilt auch für die Parameterwahl) wie die der Referenzmerkmale des Objekts  $q$ . Man erhält für jedes Partikel einen Merkmalsvektor  $h_i$ . Im nächsten Schritt werden Ähnlichkeiten zwischen den Referenzmerkmalen und jeder Objekthypothese, ausgedrückt durch die Bildmerkmale eines jeden Partikels, berechnet. Je höher die Ähnlichkeit der Merkmalsvektoren ist, desto wahrscheinlicher ist, dass dieser Zustand zum gesuchten Objekt passt. Hierfür wird ein geeignetes Ähnlichkeitsmaß nötig, auf dessen Basis die Likelihood-Wahrscheinlichkeiten  $p(y_k | x_k^i)$  für jeden Zustand berechnet werden können. Im Sinne eines *Bootstrap Filters* erhält man die Partikelgewichte durch anschließende Normierung der Likelihood-Wahrscheinlichkeiten. Das Resultat liefert die A-Posteriori Wahrscheinlichkeitsverteilung  $p(x_k | y_k)$ , aus der die aktuelle Objektposition geschätzt werden kann.

**A-Posteriori Zustandsschätzung** Anhand von  $p(x_k | y_k)$  kann nun die aktuelle Objektposition geschätzt werden. Dies entspricht dem Auswerten der Wahrscheinlichkeitsverteilung und verfolgt im Tracking generell das Ziel, denjenigen Zustandsvektor zu extrahieren, der das Maximum der Verteilung darstellt (Abbildung (5.3d)). Hierfür können unterschiedliche Verfahren herangezogen werden. Die gängigsten dabei sind *MMSE* (*Minimum-Mean-Square-Error*) und *MAP* (*Maximum-A-Posteriori*). Während ersteres dem gewichteten Mittel aller Partikelzustände mit ihren normierten Partikelgewichten entspricht, wird bei der MAP-Schätzung der Zustand verwendet, der das größte Partikelgewicht enthält. Beide Varianten haben ihre Vor- und Nachteile, und haben dabei großen Einfluss auf die geschätzte Objektlokalisierung. In dieser Arbeit wird die Schätzung in Form von MMSE herangezogen, da diese in der Regel eine weichere Trajektorie gewährleistet. Jedoch liefert sie schlechtere Genauigkeiten bei multimodalen Verteilungen (siehe auch Abbildung (5.4)). In dieser Arbeit soll jedoch die Entstehung multimodaler Verteilungen möglichst verhindert werden. Die Wahl des Schätzers hat keinerlei Auswirkung auf den weiteren Verlauf des Partikel Filters.

**Resampling** Im letzten Schritt erfolgt, wie bereits in Kapitel 3.3 beschrieben, eine Fokussierung auf die wesentlichen Anteile der Wahrscheinlichkeitsverteilung, damit einem Degenerieren der Partikel in den nächsten Iterationsschritten entgegengewirkt werden kann. Hierfür werden mit Hilfe eines *Resampling*-Verfahrens (hier: *Systematic Resampling*) Partikel mit niedrigen Gewichten (und damit schlechten Schätzungen) eliminiert und stattdessen Partikel mit ho-



**Abbildung 5.4:** Auswertung einer Wahrscheinlichkeitsverteilung (schwarz) mit den Schätzverfahren MMSE (blau) und MAP (grün). (a) Gaußverteilung, (b) Bimodale Gaußverteilung.

hen Gewichten (und damit sehr guten Schätzungen) dupliziert (siehe Abbildung (5.3e)). Durch das Prozessrauschen wird u.a. auch gewährleistet, dass sich identische Partikel (die durch das Resampling öfters gezogen wurden) um ihren gemeinsamen Zustand streuen und damit keine Redundanz auftritt.

Hiermit ist ein Iterationsschritt des Partikel Filters beendet und es folgt dieselbe Prozedur für das folgende Bild zum Zeitpunkt  $k+1$ , beginnend mit der Prädiktion der geresampelten Partikel.

### 5.3 Prädiktion mittels Bewegungsmodellen

Folgendes Kapitel beschäftigt sich mit der Wahl des Dynamikmodells für das Tracking von Fahrzeugen in Videosequenzen. Ziel des Dynamikmodells ist es, die Position des verfolgten Objekts anhand seiner bekannten vergangenen Positionen möglichst genau für den nächsten Zeitpunkt vorherzusagen. Im Partikel Filter kommt das Dynamikmodell im Prädiktions-Schritt zum Einsatz und repräsentiert dabei  $p(x_k|x_{k-1})$  und im Rahmen eines *bootstrap Filters* gleichzeitig die *importance sampling distribution*. Das Bewegungsmodell stellt damit einen zentralen Baustein des Partikel Filters dar. Auf Grundlage der prädizierten Partikel mit diesem Bewegungsmodell wird die A-Posteriori Wahrscheinlichkeitsverteilung berechnet. Werden die Partikel durch das Bewegungsmodell in die falsche Richtung prädiziert, und decken die zunächst unbekannte Position des Objekts nicht ab, kann diese schließlich auch nicht mehr gefunden werden. Zusätzlich dient das Dynamikmodell für den Fall einer Verdeckung des Objekts. Hier kann die Bewegung des Objekts, die es in den vorherigen Bildern vollzogen hat, weiter fortgesetzt werden und damit die Wahrscheinlichkeit erhöhen das Objekt bei erneutem Erscheinen auch wiederfinden zu können. Wie bereits in Kapitel 5.1 angedeutet, setzt sich die parametrisierte Form des Zustandsvektors aus Position, Geschwindigkeit in Pixelkoordinaten und eventuell einer Objektskalierung zusammen. Diese Darstellung eignet sich vor allem für Sequenzen, in denen sich der Blickwinkel auf ein Fahrzeug im Laufe der Szene nicht ändert. Klassische Autobahnüberwachungsaufnahmen wären ein Beispiel hierfür. In diesem Fall können achsparallele Templates angenommen werden.

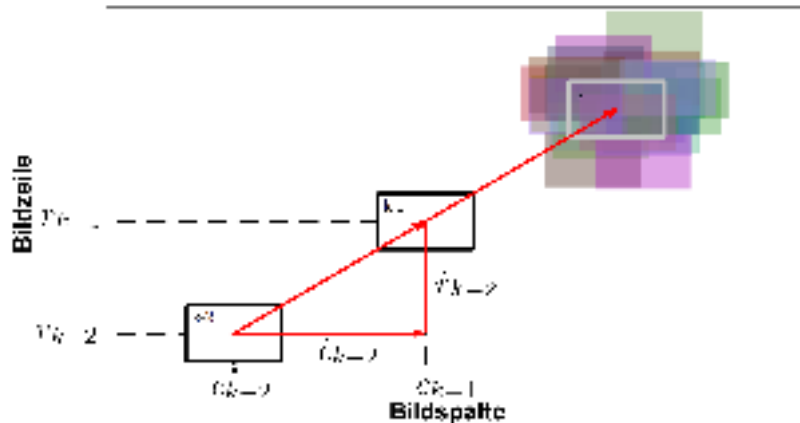
Diese Anwendung kommt in Form eines konstanten Geschwindigkeits Modells in Kapitel 5.3.1 zum Einsatz. Sollte das Objekt eine Rotation in Bezug auf die Achsen wahrnehmen können, wie zum Beispiel ein abbiegendes Fahrzeug aus Sicht der Vogelperspektive, können achsparallele Templates das Fahrzeug nicht präzise verfolgen. Hierfür wird ein weiteres Dynamikmodell in Kapitel 5.3.2 dargestellt, das eine zusätzliche Rotation des Objekts berücksichtigt.

### 5.3.1 Konstantes Geschwindigkeitsmodell

Ein Bewegungsmodell in Form eines konstanten Geschwindigkeitsmodells (engl. *constant velocity model*) (CV) modelliert die Bewegung eines Objekts aus der Position und einer als konstant angenommenen Geschwindigkeit zum letzten Zeitpunkt. Modellunsicherheiten, sowie Beschleunigungen werden durch die Überlagerung eines normalverteilten Rauschens (Prozessrauschen) modelliert. Somit setzt sich die Position eines Objektes zum Zeitpunkt  $k$  aus der Position und Geschwindigkeit zum Zeitpunkt  $k-1$  und dem Prozessrauschen zusammen. Die Objektskalierung wird im Modell als konstant angenommen. Veränderungen können ausgehend von der Objektskalierung zum letzten Zeitpunkt ebenfalls durch ein normalverteiltes Rauschen modelliert werden. Folgende Formel drückt diesen Sachverhalt aus:

$$x_k^i = \begin{pmatrix} r_k^i \\ c_k^i \\ \dot{r}_k^i \\ \dot{c}_k^i \\ s_k^i \end{pmatrix} = \begin{pmatrix} 1 & 0 & \tau & 0 & 0 \\ 0 & 1 & 0 & \tau & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} x_{k-1}^i + \begin{pmatrix} \sigma_r \\ \sigma_c \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \sigma_{\dot{r}} \\ \sigma_{\dot{c}} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \sigma_s \end{pmatrix} \quad (5.1)$$

$\tau$  stellt dabei einen Zeitschritt dar und nimmt bei Verwendung der vollen Bildfrequenz den Wert 1 an. Der erste Term beschreibt den theoretischen Zustandsübergang mit Hilfe des zugrunde liegenden Modells, die weiteren modellieren das Prozessrauschen für die unterschiedlichen Parameter. Durch die Verwendung einer stationären Kamera mit einer hohen Bildfrequenz treten in der Regel geringe Geschwindigkeiten und kaum Beschleunigungen auf. Daher könnte man eine niedrige Varianz für die Geschwindigkeit (dritter Term) ansetzen; dies könnte jedoch zu Problemen bei der Initialisierung der Objekte führen, die eine höhere Geschwindigkeit als die Geschwindigkeitsvarianz aufweisen. Um einen schnellen Anschluss an das Objekt zu finden, sollte entweder die Objektgeschwindigkeit während der Initialisierung berücksichtigt werden, indem gute Näherungswerte gewählt werden (beispielsweise durch Berechnung des optischen Flusses während der Detektion), oder es wird eine höhere Positionsvariation verwendet (zweiter Term). Letzteres erhöht zudem den Suchbereich um den aktuellen Objektzustand und kann dabei helfen, abrupte Bewegungen zu verfolgen die durch die einfache Geschwindigkeitsschätzung nicht vorhersehbar gewesen wären. Je höher allerdings die Positionsvariation gewählt wird und je niedriger die Bewegung eines Objektes zwischen zwei Bildern ist, desto ungenauer kann die Objektgeschwindigkeit geschätzt werden, da die aktuelle Objektposition nicht nur durch die Geschwindigkeit sondern auch durch die Positionsvariation erreicht werden kann. Für geringe Objektgeschwindigkeiten reicht in der Regel die Positionsstreuung aus um die Bewegung des Objekts zu modellieren.



**Abbildung 5.5:** Zustandsprädiktion mit Hilfe eines CV-Modells. Lineare Fortsetzung der Objektposition für den Zeitpunkt  $k$  (grau) anhand der Objektposition und -geschwindigkeit zum Zeitpunkt  $k - 1$ . Zusätzliche Streuung der prädizierten Objektposition durch Überlagerung mit dem Prozessrauschen (transparent).

Sollten jedoch hohe Geschwindigkeiten auftreten, bzw. eine niedrige Bildfrequenz verwendet werden, muss auf die Schätzung einer Objektgeschwindigkeit zurückgegriffen werden, um die schnellen Objektbewegungen abfangen zu können. Die Wahl dieser Parameter hängt stark von der verwendeten Szene ab, und sollte individuell an diese angepasst werden, da diese eine große Auswirkung auf die Trackinggenauigkeit hat. In Kapitel 6.1 wird dieser Einfluss untersucht.

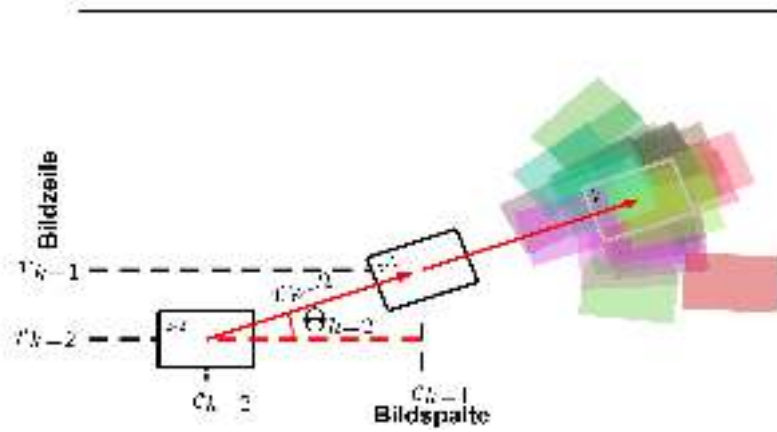
Abbildung (5.5) stellt die Prädiktion eines Partikels für zwei Zeitpunkte mit Hilfe eines CV-Modells, sowie die Überlagerung mit dem Prozessrauschen dar. Ausgehend von der letzten geschätzten Objektposition und Geschwindigkeit zum Zeitpunkt  $k - 1$  (schwarzes Rechteck) kann der neue Zustandsvektor für den Zeitpunkt  $k$  linear fortgesetzt werden (graues Rechteck). Um Unsicherheiten abfangen zu können, wird dieser Zustand mit dem Prozessrauschen überlagert, woraus eine normalverteilte Streuung um den prädizierten Zustand entsteht (transparent). Die prädizierte Objektposition kann also jeglichen Zustand, der durch die Normalverteilung um den prädizierten Modellzustand entsteht, annehmen. Zusätzlich werden durch das Streuen der Objektskalierung unterschiedliche Größen der Templates entstehen.

### 5.3.2 Einbeziehung einer Rotation

In Szenen, in denen Objekte eine Rotation vollziehen können, wie beispielsweise abbiegende Autos an einer Kreuzung bei hoch montierter Kamera, kann das eben vorgestellte CV-Modell das Objekt aufgrund fehlender Rotationsmodellierung nicht zufriedenstellend verfolgen. Aus diesem Grund wird auf Basis eines konstanten Geschwindigkeitsmodells eine zusätzliche Rotation eingeführt.

Der Zustandsvektor des CVR (*constant velocity and rotation*) Modells umfasst weiterhin die Parameter für Objektposition  $r$  und  $c$  und eventuell einen Objektskalierungsfaktor des Templates  $s$ , jedoch wird von der Darstellung der Geschwindigkeiten in Zeilen- und Spaltenrichtung abgesehen und zu einer Absolutgeschwindigkeit  $v$  und einer Rotation  $\theta$  verändert. Die Position





**Abbildung 5.6:** Zustandsprädiktion mit Hilfe eines CVR-Modells. Lineare Fortsetzung der Objektposition und Rotation für den Zeitpunkt  $k$  (grau) anhand der Objektposition, Geschwindigkeit und Rotation zum Zeitpunkt  $k-1$ . Zusätzliche Streuung durch Überlagerung mit dem Prozessrauschen (transparent).

des neuen Zustandes zum Zeitpunkt  $k$  kann damit ausgedrückt werden durch die Position auf einem Kreissegment mit dem Radius  $v$  und der Rotation  $\Theta$  um den Mittelpunkt der zuletzt geschätzten Objektposition. Daraus folgt für den Zustandsvektor zum Zeitpunkt  $k$ :

$$x_k^i = \begin{pmatrix} r_k^i \\ c_k^i \\ v_k^i \\ \Theta_k^i \\ s_k^i \end{pmatrix} = \begin{pmatrix} 1 & 0 & \tau \sin(\Theta_{k-1}^i + \sigma_\Theta) & 0 & 0 \\ 0 & 1 & \tau \cos(\Theta_{k-1}^i + \sigma_\Theta) & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} x_{k-1}^i + \begin{pmatrix} \sigma_r \\ \sigma_c \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \sigma_v \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_\Theta \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \sigma_s \end{pmatrix} \quad (5.2)$$

Dieses Modell entspricht dem Bewegungsverhalten von Vehikeln, die nahezu in Nadirrichtung betrachtet werden. Mit diesem Modell können Objekte eine Rotation vornehmen, deren maximaler Kurvenradius durch  $\sigma_\Theta$  vorgegeben wird. Durch  $\sigma_v$  werden dagegen, wie im herkömmlichen CV-Modell, Beschleunigungen berücksichtigt. Kennzeichnend für dieses Modell ist außerdem, dass Bewegungen nur in Fahrtrichtung erfolgen können und nicht senkrecht dazu. Durch  $\sigma_r$  und  $\sigma_c$  können allerdings Diskrepanzen zwischen wahrer Objektposition und Schätzung ausgeglichen werden, die beispielsweise entstehen, wenn das Objekt zum Stillstand kommt und die geschätzte Position zwar die korrekte Richtung angibt, jedoch kleine Positionsfehler parallel zu den Objektachsen aufweist. Durch diese Parameter (die in der Regel klein gewählt werden sollten) können demnach auch kleine Bewegungen quer zur Fahrtrichtung zugelassen werden um diese Positionsfehler auszugleichen. Auf die Schätzung einer Skalierung  $s$  kann verzichtet werden, wenn die Aufnahmegeometrie exakt senkrecht zum Boden erfolgt.

Abbildung (5.6) stellt die Prädiktion eines Partikels mit Hilfe eines CVR-Modells dar. Die Geschwindigkeitsschätzung zu  $k-1$  wird zum Zeitpunkt  $k$  konstant fortgesetzt. Das Gleiche gilt für die Rotation. Daraus erhält man den prädizierten Zustand (graues Rechteck). Durch  $\sigma_v$  werden

Beschleunigungen modelliert und durch  $\sigma_\Theta$  werden weitere eventuelle Rotationen des Objekts berücksichtigt. Das Partikel kann damit einen um den prädierten Zustand normalverteilten Zustand annehmen (transparent).

## 5.4 Korrektur mittels Beobachtungsmodellen

Durch die Prädiktion der Partikel mit Hilfe eines der in Kapitel 5.3 erläuterten Bewegungsmodelle ergeben sich die prädierten Zustände, auf dessen Grundlage der Korrektur-Schritt des Partikel Filters durchgeführt wird. Jeder Zustand  $x_k^i$  repräsentiert damit eine Bildregion  $S_{x_k^i}$  (vgl. Abbildung 5.7a). Eine Merkmalsextraktion für jedes dieser Templates wird nun durchgeführt und eine Wahrscheinlichkeit berechnet, mit der diese vom gesuchten Objekt stammen. Die Merkmalsextraktion für Farbe und Kanten wurde in Kapitel 4 vorgestellt und wird im Folgenden als bekannt vorausgesetzt.

### 5.4.1 Farbe

Partikel Filter für die Objektverfolgung die ausschließlich Farbe in Form von Farbhistogrammen als Bildmerkmale verwenden finden sich u.a. in [50], [43], [44]. Die Merkmalsextraktion der Farbe erfolgt durch die Histogrammberechnung innerhalb der Templates  $S_{x_k^i}$ . Als Ergebnis erhält man pro Partikel  $i$  und für jeden Farbkanal  $c$  einen Merkmalsvektor  $h_i^c$  mit der Länge der Anzahl der Bins  $B$ . Um diese mit dem Merkmalsvektor  $q^c$  des Referenzobjektes aus  $k = 0$  vergleichen zu können, müssen alle Vektoren normiert sein, da nur die relative Häufigkeit der Farbwerte von Interesse ist (die absolute Anzahl kann sich durch Skalierungsänderung des Objekts verändern). Um zwei Histogramme miteinander vergleichen zu können, muss man ein Ähnlichkeitsmaß heranziehen. Hierfür eignen sich eine Vielzahl von verschiedenen Maßen. Ein häufig verwendetes Maß ist die *Bhattacharyya Distanz*  $d(\cdot)$ , die sich für einen Vergleich zweier diskreter Dichten, wie auch Histogrammen eignet. Für den Vergleich zweier normierter Histogramme gleicher Länge kann diese folgendermaßen berechnet werden:

$$d(h_i^c, q^c) = \sqrt{1 - \rho(h_i^c, q^c)}, \quad (5.3)$$

wobei  $\rho(h_i^c, q^c)$  als *Bhattacharyya Koeffizient* bezeichnet wird, und sich aus dem Skalarprodukt der Häufigkeiten pro Bin zusammensetzt:

$$\rho(h_i^c, q^c) = \sum_{j=1}^B \sqrt{h_i^{j,c} \cdot q^{j,c}} \quad (5.4)$$

Durch die Normierung der Histogramme kann  $\rho$  als auch  $d$  Werte zwischen  $[0, 1]$  annehmen. Je ähnlicher sich zwei Histogramme sind, desto höher wird  $\rho$  und kleiner  $d$ . Für zwei identische Histogramme erhält man demnach  $\rho = 1$  und  $d = 0$ .

Im Falle von Farbbildern, kann für jeden Farbkanal die *Bhattacharyya Distanz* berechnet und

anschließend gemittelt werden:

$$d(h_i, q) = \frac{1}{3} \sum_{c \in R, G, B} d(h_i^c, q^c) \quad (5.5)$$

Werden Multi-Part Histogramme verwendet, muss dieser Histogrammvergleich für jede Subregion des Templates mit der des Referenztemplates auf gleiche Art geschehen. Die Darstellung des Histogramms für die Region  $t = 1, \dots, T$  ist  $h_i^{c,j,t}$  für den Objektkandidaten und  $q^{c,j,t}$  für das Referenzobjekt. Zusätzlich können die einzelnen Regionen unterschiedlich gewichtet werden, woraus eine gezielte Behandlung von Verdeckungen möglich wird. Verdeckte Regionen können bei den Histogrammvergleichen je nach Verdeckungsgrad schwächer oder sogar ganz vernachlässigt werden damit sich die Suche auf die noch sichtbaren Regionen beschränken kann. Hierfür wird ein Vektor  $\alpha$  mit Gewichtungsparmetern für jede Region eingeführt. Für die Unterteilung in vier Subregionen erhält man demnach

$$\alpha = [\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4] \quad (5.6)$$

Die Nummerierung der Gewichtungsfaktoren entspricht dabei der Unterteilung in Abbildung (4.3a) aus Kapitel 4.1.3. Werden Multi-Part Histogramme verwendet, ändert sich Gleichung (5.4) zu

$$\rho(h_i^c, q^c) = C_T \cdot \sum_{t=1}^T \alpha_t \cdot \sum_{j=1}^B \sqrt{h_i^{j,c,t} \cdot q^{j,c,t}} \quad (5.7)$$

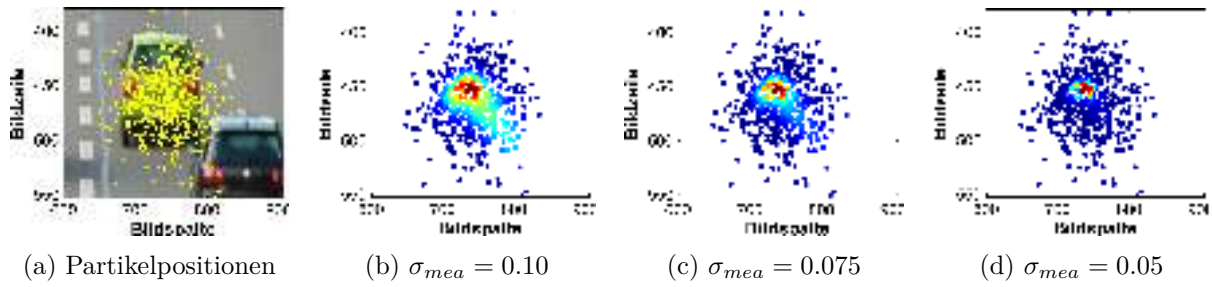
wobei  $C_T = \left(\sum_{t=1}^T \alpha_t\right)^{-1}$  als Normalisierungskonstante angesehen werden kann damit der Gesamtkoeffizient wiederum Werte zwischen  $[0, 1]$  annehmen kann.

Basierend auf diesem Ähnlichkeitsmaß kann man die Likelihood-Wahrscheinlichkeit jedes Partikels berechnen, die angibt mit welcher Wahrscheinlichkeit der jeweilige Merkmalsvektor vom gesuchten Objekt stammt:

$$p(y_k | x_k^i) \propto \frac{1}{\sqrt{(2\pi\sigma_{mea})}} \exp\left(-\frac{d(h_i, q)^2}{2\sigma_{mea}^2}\right), \quad (5.8)$$

wobei  $\sigma_{mea}$  das Messrauschen beschreibt. Zur Steigerung der Effizienz kann die Berechnung von  $p(y_k | x_k^i)$  vereinfacht werden. Zum Einen muss  $d(h_i, q)$  nicht ausdrücklich berechnet werden, da durch das Quadrieren dieses Terms der *Bhattacharyya Koeffizient*  $(1 - \rho(h_i, q))$  übrig bleibt. Zum Anderen dient der erste Term der Funktion einer Normierung. Da in unserem Fall eines *SIR-Filters* die Likelihoodwahrscheinlichkeiten durch anschließende Normierung sogleich als Partikelgewichte verwendet werden können, kann auf die Normierungskonstante der Likelihoodfunktion verzichtet werden und es ergibt sich schließlich

$$p(y_k | x_k^i) \propto \exp\left(-\frac{(1 - \rho(h_i, q))}{2\sigma_{mea}^2}\right) \quad (5.9)$$

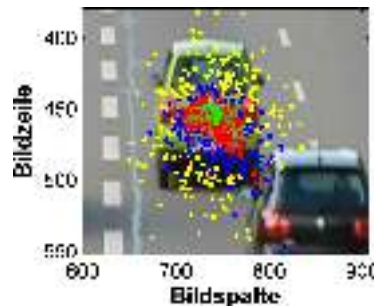


**Abbildung 5.7:** Einfluss unterschiedlichem Messrauschens (b-d) auf die Wahrscheinlichkeitsverteilung  $p(x_k|y_k)$  der prädizierten Partikel (a) für Verwendung von Farbmerkmalen mit  $c = 3$  Farbkanäle und  $B = 16$  Bins für das Modell des grünen Fahrzeuges. Darstellung der Mittelpunkte der Templates durch die Partikel. Steigendes Partikelgewicht von blau nach rot.

Somit hängt die Likelihood-Wahrscheinlichkeit nur vom *Bhattacharyya Koeffizient*  $\rho$  und dem Messrauschen  $\sigma_{mea}$  ab. Hohe Werte für  $\rho$  resultieren dadurch in hohen Likelihood-Wahrscheinlichkeiten und damit potentiellen Kandidaten, dass dieses Template vom gesuchten Objekt stammt. Im SIR-Filter kann Gleichung (5.9) direkt für die Berechnung der Gewichte verwendet werden. Da die Gewichte ebenfalls eine Wahrscheinlichkeit darstellen sollen, wird  $p(y_k|x_k^i)$  nach der Berechnung normiert, sodass  $\sum_{i=1}^N w_k^i = 1$  gilt:

$$w_k^i = \frac{p(y_k|x_k^i)}{\sum_{i=1}^N p(y_k|x_k^i)} \quad (5.10)$$

Das Messrauschen  $\sigma_{mea}$  in Gleichung (5.9) beeinflusst wie spitz die A-Posteriori Wahrscheinlichkeitsdichte  $p(x_k|y_k)$  ist und hat damit einen erheblichen Einfluss auf die Genauigkeit des Trackings. Durch Angabe eines hohen Messrauschens geht man von einer hohen Messunsicherheit aus, dass in einer flacheren Wahrscheinlichkeitsverteilung resultiert, bei dem auch höhere Distanzmaßen und damit schlechtere Histogrammübereinstimmungen in höhere Gewichte resultieren (siehe Abbildung (5.7b)). Hier kann es durchaus vorkommen, dass Störobjekte in der Nähe des gesuchten Objekts zu hohen Gewichten beteiligter Partikel führen (hier: helle Reflexionen des dunklen Autos). Die Verteilung kann somit eine Krümmung oder sogar multimodal werden, woraus eine Auswertung durch einen MMSE-Schätzer nicht mehr präzise erfolgen kann. Wird ein



**Abbildung 5.8:** Einfluss des Messrauschens auf die Anzahl gezogener Partikel für das Resampling eines Partikelsatzes (gelb). Gezogene Partikel für  $\sigma_{mea} = 0.05$  (grün),  $\sigma_{mea} = 0.075$  (grün+rot) und  $\sigma_{mea} = 0.1$  (grün+rot+blau)

geringeres Messrauschen verwendet (Abbildung (5.7c)), fällt diese Störung nicht mehr so stark ins Gewicht und die Verteilung wird allgemein spitzer. Für sehr geringes Messrauschen (Abbildung (5.7d)) hat das Störobjekt zwar keinen Einfluss mehr auf die Verteilung, jedoch haben die meisten Partikel ein vernachlässigbar kleines Gewicht und wurden somit umsonst berechnet. Eine zu spitze Verteilung ist ebenfalls sehr sensitiv, wenn sich das Erscheinungsbild im Laufe der Sequenz durch äußere Einflüsse geringfügig verändert. Dies führt schnell dazu, dass nur noch sehr wenige Partikel ein hohes Gewicht erhalten. Dementsprechend kann eine Auswertung dieser sehr spitzen Verteilung nicht mehr präzise durchgeführt werden.

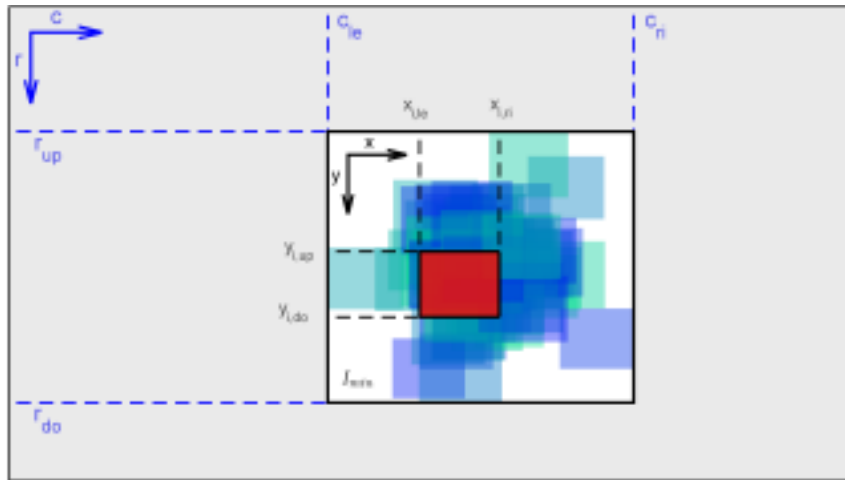
Die Wahl des Messrauschens sollte pro Szene individuell gestaltet werden, da es ebenfalls stark von der Parameterwahl (Anzahl der Bins, Farbraum, Block-Anzahl) abhängt. Da krumme oder multimodale Verteilungen vor allem im Fall von Verdeckungen entstehen die bei Verkehrsaufnahmen sehr häufig vorkommen, sollte dies auf jeden Fall berücksichtigt werden. Entweder sollte die Auswertung dieser Verteilungen durch bessere Schätzverfahren durchgeführt werden, die den richtigen Mode der Verteilung extrahieren, oder die Entstehung solcher Verteilungen sollte möglichst verhindert werden. Letzteres wird durch die explizite Behandlung von Verdeckungen (siehe Kapitel 5.5) erreicht.

Zusätzlich hat die Wahl des Messrauschens auch direkten Einfluss auf das Resampling. Erhalten viele Partikel ein hohes Gewicht, werden auch zunehmend flächendeckender Partikel durch das Resampling gezogen. Dies kann zu steigender Divergenz der Partikel in den nächsten Frames führen. Zu geringes Messrauschen resultiert oft darin, dass nur noch wenige oder nur ein Partikel gezogen werden, und somit die Streuung der Partikel durch das Prozessrauschen hoch gewählt werden muss damit diese sich im nächsten Prädiktionsschritt ausreichend streuen. (vgl. Abbildung (5.8)).

### 5.4.2 Kanten

Partikel Filter, die Kantenrichtungshistogramme als Bildmerkmale verwenden, finden sich in [7], [41], [54]. Werden Kantenrichtungshistogramme als Bildmerkmale verwendet, erfolgt die Merkmalsextraktion für jedes Partikel nach dem in Kapitel 4.2 dargestellten Verfahren. Die Erstellung der Gradientenamplituden- und orientierungsbilder pro Partikel nimmt die größte Rechenzeit in Anspruch, kann aber aufgrund der Redundanz überlappender Bildregionen reduziert werden. Es kann ein minimaler Bildausschnitt  $I_{min}$  berechnet werden, der die Templates aller Partikel abdeckt (vergleiche Abbildung (5.9)). Die Erstellung des Amplituden- und orientierungsbildes muss somit nur einmal für diesen Bildausschnitt berechnet werden. Benötigt wird der Ursprung des Bildausschnittes im Originalbild, wobei in  $I_{min}$  die Achse in Spaltenrichtung als  $x$  und in Zeilenrichtung als  $y$  bezeichnet wird. Der Ursprung eines Templates  $i$  (rot) im Originalbild  $(r_i, c_i)$  findet sich in  $I_{min}$  bei  $y_{i,up} = (r_i - r_{up})$  und  $x_{i,le} = (c_i - c_{le})$ .

Als Ergebnis erhält man pro Partikel  $i$  einen normierten Merkmalsvektor  $h_i^{j,t}$  der Länge der Bins  $j = 1, \dots, B$  pro Block  $t = 1, \dots, T$ . Die Berechnung der Partikelgewichte erfolgt wiederum durch das *Bhattacharrya*-Ähnlichkeitsmaß im gleichen Schema wie in Kapitel 5.4.1 nach den



**Abbildung 5.9:** Verringern des Rechenaufwandes bei Verwendung von Kantenmerkmalen. Berechnung der Gradientenbilder nur für den minimalen Bildbereich  $I_{min}$ , der die Fläche aller Templates (transparent) abdeckt.

Gleichungen (5.3) und (5.4). Eine dynamische Gewichtung der einzelnen Blöcke kann bei Kantenmerkmalen nicht angewendet werden. Bereits geringe Störungen in einem Teil des Blocks kann zu komplett unterschiedlichen Kantenrichtungshistogrammen führen.

## 5.5 Behandlung von Verdeckungen

Verdeckungen stellen eine der größten Herausforderungen in der Objektverfolgung dar. Besonders in Verkehrssituationen kann es dabei oft zu Teilverdeckungen, aber auch zu vollständigen Verdeckungen der Fahrzeuge kommen. Grundsätzlich unterscheidet man Verdeckungen in drei Kategorien, **Selbstverdeckung**, **gegenseitige Objektverdeckung** und **Verdeckung mit dem Hintergrund** [56].

Unter **Selbstverdeckung** versteht man die Eigenschaft eines Objektes, dass sich Teile des Objektes hinter anderen Teilen verdecken. Beispiel hierfür wäre ein mit dem Rücken zur Kamera gerichteter Mensch, dessen Hand bei Anwinkeln des Armes nicht mehr sichtbar ist. Solche Teilverdeckungen sollten durch die verwendeten Objektmodelle berücksichtigt werden. In Verkehrsszenarien handelt es sich allerdings um starre Körper, sodass solche Verdeckungen unberücksichtigt bleiben können.

**Gegenseitige Objektverdeckungen** treten bei zwei (oder mehreren) bewegten Objekten auf. Dabei können Teilverdeckungen aber auch daraus resultierend Kompletterdeckungen entstehen. Da mit einem Trackingsystem möglichst alle Objekte in der Szene verfolgt werden sollen, können durch die bekannten Objektpositionen Verdeckungsart und Dauer jedoch vorhergesagt und damit berücksichtigt werden (siehe auch Abbildung (5.10b)).

Des Weiteren kann es zu **Verdeckungen mit dem Hintergrund** kommen. Je nach Szene können Teile von Objekten durch die Umwelt verdeckt werden (Störobjekte im Vordergrund wie



**Abbildung 5.10:** Auftreten unterschiedlicher Verdeckungsarten in Verkehrsaufnahmen. (a) Teilverdeckung mit statischem Hintergrund. (b) Gegenseitige Verdeckung zweier bewegter Objekte

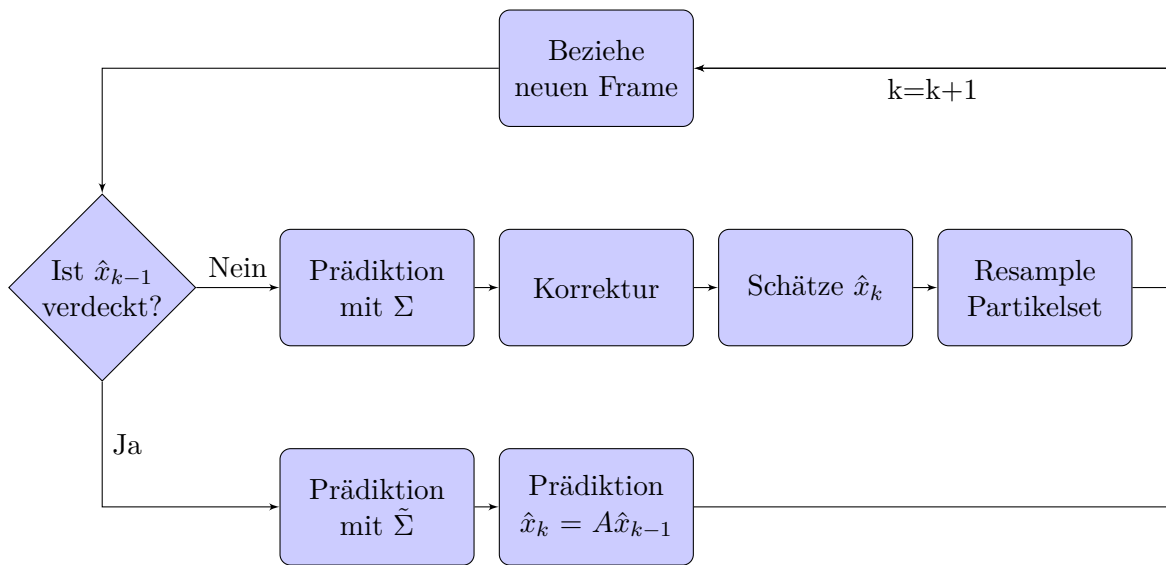
in Abbildung (5.10a)) sowie Objekte für einen längeren Zeitraum komplett verdeckt sein (z.B. Tunnel). Die Schwierigkeit besteht hier in einer automatischen Detektion einer Verdeckung. Während Teilverdeckungen mit dem Partikel Filter und geeigneten Beobachtungsmodellen bewältigt werden können, fällt die Detektion einer Verdeckungen hinter einem Objekt mit ähnlichem Erscheinungsbild (z.B. gleiche Farbe) schwer. Da es sich bei den Verkehrsszenen allerdings um Aufnahmen mit einer stationären Kamera und einer wohlbekannten Szene handelt, können verdeckte Regionen in der Szene vorab definiert werden und das Eintreten eines Objekts in jene Regionen vorhergesagt werden. Kapitel 5.5.1 beschäftigt sich mit der Behandlung von Verdeckungen mit dem Hintergrund.

### 5.5.1 Verdeckung mit dem Bildhintergrund

Wie bereits beschrieben handelt es sich hierbei um Verdeckungen des Objekts mit starren Körpern der Szene, die abhängig von ihrer Größe, kurzzeitige Teilverdeckungen, aber auch vollständige Verdeckungen für einen längeren Zeitraum verursachen.

**Teilverdeckungen** können meist ohne zusätzliche Behandlung gemeistert werden. Grund hierfür liegt in der Stärke des Partikel Filters, durch das mehrere Hypothesen simultan verfolgt werden, die durch das Prozessrauschen eine größere Suchumgebung um das Objekt abdecken. So kann selbst nach kurzzeitigem Verlust des Objekts dieses in dessen näheren Umgebung wiedergefunden werden. Selbst wenn zunächst nur wenige Partikel das Objekt nach Verlust wieder detektiert und damit hohe Gewichte erhalten haben, werden diese nach dem Resampling dupliziert und zum nächsten Zeitpunkt können bereits mehrere Partikel das Objekt lokalisieren. Dies stellt einen wesentlichen Vorteil gegenüber dem Kalman Filter dar, mit dem nur eine Hypothese verfolgt wird. Diese eine Hypothese wird in der Regel nicht ausreichen um das Objekt nach einer Störung, und damit Verlust des Objekts, wiederfinden zu können. In Kapitel 6.2.1 wird die Fähigkeit des Partikel Filters mit Teilverdeckungen umgehen zu können untersucht.

**Vollverdeckungen über einen längeren Zeitraum** können selbst mit dem Partikel Filter nicht ohne weiteres behandelt werden. Verschwindet das Objekt vollständig für einen längeren



**Abbildung 5.11:** Ablauf des Partikel Filters unter Berücksichtigung von Vollverdeckungen. Ausschluss des Korrektur- und Resampling-Schrittes im Falle von Vollverdeckungen, sowie Prädiktion der Partikel mit gedämpften Prozessrauschen.

Zeitraum, wird der Tracker das Objekt verlieren und bei Wiedererscheinen in der Regel nicht wiederfinden. Je nach Umgebung können zwei Verhalten der Partikel entstehen. Sollte in der näheren Umgebung des verlorenen Objekts ähnliche Bildmerkmale (Farbe oder Kanten) extrahiert werden, werden sich die Partikel in den nächsten Zeitpunkten vermehrt um diese Region fokussieren und dort für die restliche Sequenz verbleiben. Das Partikel Filter hängt an dieser Region fest und kann sich nicht mehr von ihr lösen. Das Wiederfinden des Objekts wird in diesem Fall unmöglich. Sollte selbst die Umgebung keinerlei ähnliche Bildmerkmale aufweisen, werden alle Partikel eine niedrige Likelihood-Wahrscheinlichkeit erhalten. Dadurch wird die Varianz der Partikelgewichte sinken und durch das Resampling zufällige Partikel gezogen, die sich nicht um eine bestimmte Region konzentrieren. Geschieht dies über mehrere Zeitpunkte, kann eine starke Streuung und ein zufälliges „Wandern“ der Partikel beobachtet werden. Dies wird solange anhalten, bis zufällig ein ähnliches Bildmerkmal extrahiert wird und sich dadurch mehrere Partikel wieder um diese Region fokussieren, oder die Partikel den Bildrand erreichen.

Um ein Wiederfinden des Objekts zu ermöglichen, müssen also die Partikel in die Richtung bewegt werden, in der das Objekt wieder in der Szene erscheinen wird. Hierfür eignet sich das Konzept der Trennung von *Prädiktion* und *Korrektur* eines rekursiven Bayes-Filters. Im Folgenden wird angenommen, dass die Bildregion in der die Verdeckung mit dem Objekt stattfindet, als bekannt vorausgesetzt wird. Durch die Verwendung einer stationären Kamera an bekannten Verkehrsszenen ist dies eine legitime Voraussetzung. Alternativ kann eine verdeckte Region mit Hilfe von weiteren Verfahren detektiert werden. So kann beispielsweise vor dem eigentlichen Tracking der optische Fluss der Szene berechnet werden. Sollten sich zwischen stark dynamischen Regionen statische Regionen befinden, kann von einer Verdeckung dieses Gebiets ausgegangen werden. Erreicht das verfolgte Objekt diese Bildregion zum Zeitpunkt  $k$ , kann mit Hilfe



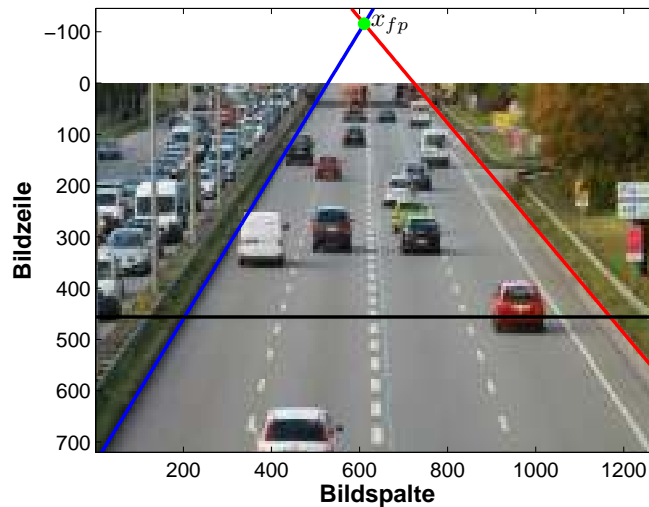
des Bewegungsmodells eine Vorhersage der Objektbewegung während der Verdeckung bestimmt werden. Durch die rekursive Schätzung des Zustandsvektors enthält der Zustandsvektor  $x_{k-1}$  neben der Position auch die Geschwindigkeit, mit der sich das Objekt in den letzten Zeitpunkten bewegt hat. Während der Verdeckung können nun die Partikel und auch die Objektposition mit Hilfe dieser Geschwindigkeit zur neuen Position  $x_{k+1}$  prädiziert werden, die durch die Verwendung eines konstanten Geschwindigkeitsmodells aus Kapitel 5.3 als konstant angenommen wird. Eine Korrektur, und damit die Extraktion der Bildmerkmale für diese prädizierten Partikel wird während der Verdeckung ausgesetzt. Abbildung (5.11) stellt den Ablauf des Partikel Filters unter Berücksichtigung einer Verdeckung schematisch dar. Zu jedem Zeitpunkt  $k$  wird untersucht ob das Objekt unter einer Verdeckung leidet. Falls dem nicht so ist, wird der herkömmliche *Prädiktion-Korrektur-Resample* Ablauf angewendet, der in Kapitel 5.2 dargestellt wurde. Falls eine Verdeckung vorliegt, wird lediglich die Prädiktion der Partikel angewendet und anstatt den neuen Zustand aus der approximierten Wahrscheinlichkeitsverteilung zu schätzen, wird dieser ebenfalls mit Hilfe des Bewegungsmodells fortgesetzt. Dies geschieht solange, bis die prädizierte Objektposition keiner Verdeckung mehr unterliegt. Anschließend wird die Korrektur erneut angewendet und das Objekt in der neuen Umgebung gesucht. Dies setzt voraus, dass die prädizierte Geschwindigkeit des Objekts während der Verdeckung möglichst akkurat geschätzt wurde und sich das Objekt ebenfalls mit konstanter Geschwindigkeit fortbewegt. Während der Verdeckung sollte jedoch eine Modifikation des Prozessrauschens vorgenommen werden.

Das Prozessrauschen, zusammengefasst im Vektor  $\Sigma = [\sigma_r \ \sigma_c \ \sigma_{\dot{r}} \ \sigma_{\dot{c}} \ \sigma_s]^T$ , formuliert die Streuung der Partikel (analog bei Verwendung des CVR-Modells). Durch Verwenden des Resamplings wird einem additiven Streuen der Partikel im Laufe der Zeit entgegengewirkt. Da dieser Schritt während der Verdeckung nicht angewendet wird, werden die Partikel immer weiter gestreut, sodass schließlich ein viel zu großer Suchraum entsteht. Eine Vergrößerung dieses Suchraums ist einerseits von Vorteil, da Ungenauigkeiten während der Prädiktion abgefangen werden können, jedoch ist die Gefahr von Fehldetektionen bei Wiederscheinen des Objekts groß. Daher wird während der Verdeckung ein gedämpftes Messrauschen  $\tilde{\Sigma}$  verwendet, um die Streuung der Partikel zu kompensieren. Dabei kann ebenfalls die Richtung der Streuung beachtet werden, in dem beispielsweise bei einer horizontalen Verdeckung eine größere Streuung entlang der Bildspalte und eine kleine Streuung in der Bildzeile berücksichtigt wird.

### 5.5.2 Verdeckung zweier Objekte

Beim simultanen Verfolgen mehrerer Objekte stehen zu jedem Zeitpunkt die Positionen sowie Ausmaßen aller Objekte zur Verfügung. Sollten sich zwei Objekte verdecken, kann dies durch die Schnittmenge beider Objektregionen bestimmt werden. Bei Unterteilung des Objektmodells in mehrere Blöcke kann zu jedem Zeitpunkt für jeden Block  $t = 1, \dots, T$  der Verdeckungsgrad  $O^t \in [0, 1]$  berechnet werden. Dieser nimmt den Wert 0 an, wenn keine Verdeckung vorliegt und 1 bei maximaler Verdeckung. Die Identifikation des verdeckten Objektes kann in der Regel durch die bekannte Bildgeometrie bestimmt werden.

Kommt es zu einer Verdeckung eines Objekts, kann anhand des berechneten Verdeckungsgra-



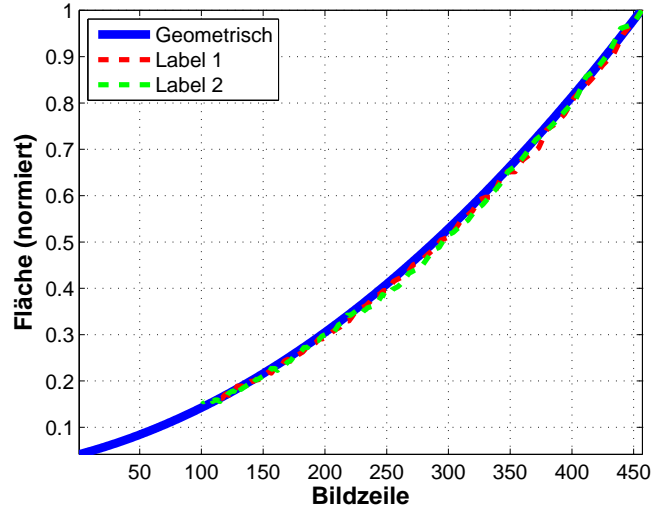
**Abbildung 5.12:** Bestimmung des Fluchtpunktes  $x_{fp}$  einer Szene als Schnittpunkt zweier in Weltkoordinaten paralleler Geradenstücke (blau, grün). Initialisierungslinie (schwarz) dient zur Berechnung der Objektgröße entlang der Bildzeile.

des jedes Blockes eine dynamische Gewichtung  $\alpha_t = (1 - O^t)$  erfolgen, die in die Berechnung des *Bhattacharyya Koeffizienten* in Gleichung (5.7) eingeht. Je höher die Verdeckung für diesen Block ist, desto geringer geht das Modell für diesen Block beim Histogrammvergleich ein. Wird im Extremfall ein Block komplett verdeckt, erfolgen die Histogrammvergleiche und damit die Suche nach dem Objekt anhand der restlichen drei sichtbaren Blöcke.

Dies ermöglicht ein robusteres, längeres Tracking verdeckter Objekte. Sollte die Gesamtverdeckung  $\sum_{t=1}^t O^t$  jedoch einen Schwellwert  $S_1$  überschreiten, wird von einem unzureichenden Modell ausgegangen in dem kein robustes Tracking mehr erzielt werden kann. Von da an wird von einer Vollverdeckung ausgegangen, die nach dem in Abbildung (5.11) dargestellten Verfahren behandelt wird. Die Objektposition wird nun so lange präzidiert, bis die theoretische Gesamtverdeckung unter einem Schwellwert  $S_2$  liegt.  $S_2$  sollte dabei deutlich niedriger als  $S_1$  gewählt werden, damit zum einen eventuelle Prädiktionsfehler ausgeglichen werden können und zum anderen die Gefahr verringert wird, dass das Objekt nach wenigen Frames wieder in die Vollverdeckung gerät. Anhand von Tests ergaben sich die besten Ergebnisse für  $T = 4$  Blöcke bei  $S_1 = 3$  und  $S_2 = 2$ .

## 5.6 Externe Schätzung der Skalierung

Die Schätzung der Skalierungsänderung eines Objekts innerhalb einer Szene stellt eine kritische Größe dar, die durch die verwendeten Bildmerkmale nicht immer optimal gelöst werden kann (siehe auch Kapitel 6.2.2). Wird die Skalierungsänderung durch die bekannte Bildgeometrie extern berechnet, können wesentlich robustere Ergebnisse erzielt werden. Hier sind insbesondere Szenen betroffen, in denen eine Neigung der Bildebene zur Straßenebene vorliegt, wie beispiels-



**Abbildung 5.13:** Verlauf der Objektfläche entlang der Bildzeile für eine Initialisierungsgröße von 1 bei  $r_{init} = 456$  (entspricht Initialisierungslinie in Abbildung (5.12)). (blau) Verlauf bei geometrischer Berechnung. (rot,grün) Verlauf zweier manuell gelabelter Fahrzeuge.

weise eine auf Brücken montierte Überwachungskamera. Ist der Straßenverlauf geradlinig und liegt in einer Ebene (planar) kann die Skalierungsänderung extern berechnet werden und aus der Schätzung im Partikel Filter entnommen werden.

Abbildung (5.12) stellt eine solche Szene dar. Wie bereits erwähnt, werden die Proportionen der Breite und Höhe des Templates als konstant angenommen und eine Größenänderung des Objekts kann somit als Skalierungsfaktor relativ zur Initialisierungsgröße bestimmt werden. Somit haben Objekte, die in gleicher Entfernung zur Kamera (in diesem Fall in der selben Bildzeile  $r_{init}$ , schwarze Linie) initialisiert werden, den gleichen Verlauf der Skalierungsänderung abhängig von ihrer Position in der Bildzeile. Für automatische Trackingsysteme (wie in Kapitel 7) kann diese Initialisierungslinie durch die Verwendung von Detektions- und Trackingzonen erreicht werden. Hier werden alle Objekte mit der Skalierung 1 in Bezug auf ihrer jeweilige individuelle Objektgröße initialisiert. Die Objektfläche nimmt quadratisch entlang der Bildzeile ab (bzw. zu wenn sich das Objekt auf die Kamera zubewegt) und hat theoretisch den Wert 0 wenn es den Horizont erreicht.

Die Bestimmung dieses Verlaufs kann durch die Kenntnis des Bildhorizonts und Anwendung des Stahlsatzes berechnet werden.

Der Horizont kann durch die Berechnung des Fluchtpunktes anhand zweier in der Weltebene paralleler Geraden bestimmt werden. Im Fall von Verkehrsanwendungen kann somit immer die Kenntnis von parallelen Straßenrändern herangezogen werden. In [25] wird ein Ansatz zur automatischen Horizontbestimmung in Bildern vorgestellt. Ein manuelles Eingreifen eines Benutzers wäre dadurch nicht nötig, wird jedoch in dieser Arbeit nicht angewendet.

Für eine robuste Bestimmung des Fluchtpunktes sollten mehrere Punkte auf jedem Straßenrand angegeben werden. Anhand zweier ausgleichender Geraden (blaue und rote Linie) kann

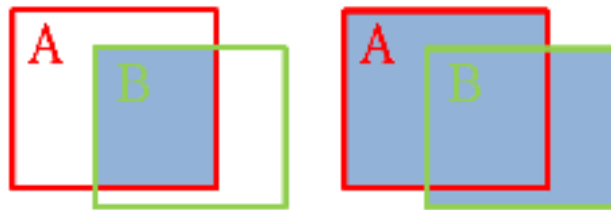
der Fluchtpunkt  $x_{fp}$  durch den Schnittpunkt dieser Geraden bestimmt werden. Durch Anwendung des Strahlensatzes kann nun für jede Bildzeile  $r_i$  die normierte Fläche  $A_i$  mit Hilfe der Initialisierungszeile  $r_{init}$  (schwarze Linie) und der theoretischen Bildzeile des Fluchtpunktes  $r_{fp}$  folgendermaßen bestimmt werden:

$$A_i = \left( \frac{r_i - r_{fp}}{r_{init} - r_{fp}} \right)^2 \quad (5.11)$$

Abbildung (5.13) zeigt den Verlauf der normierten Fläche entlang der Bildzeile für die geometrische Berechnung (blau) im Vergleich zu zwei manuell erstellen Ground-Truth Datensätzen dieser Szene. Die Objekte wurden für  $r_{init} = 456$  mit einem Skalierungsfaktor  $s = 1$  initialisiert. Die quadratische Abnahme der Objektfläche in Fahrtrichtung wird ersichtlich und eine gute Übereinstimmung des Verlaufs im Vergleich zu den Ground-Truth Datensätzen kann festgestellt werden. Durch die Annahme einer konstanten Proportion zwischen Objekthöhe- und Breite erfolgt eine lineare Abnahme der Objektskalierung und kann durch  $s_i = \sqrt{A_i}$  berechnet werden.

In diesem Kapitel soll das in Kapitel 5 erläuterte Trackingverfahren anhand realer Verkehrsaufnahmen untersucht werden. Dabei sollen neben dem Einfluss verschiedener Parameter auf die Robustheit des Verfahrens auch unterschiedliche Szenen aus dem Straßenverkehr verwendet werden, um die Fähigkeiten, sowie Stärken und Schwächen des Verfahrens aufzuzeigen. Schließlich sollen externe Einflüsse, wie Bildfrequenz, Videoauflösung und Anzahl der Farbkanäle auf die Ergebnisse des Trackings untersucht werden. Die aufgeführten Laufzeiten beziehen sich auf eine in *HALCON*-eingebundene Implementierung in C für einen Desktop PC mit einem 3.3 GHz Intel Core i5-3550 Prozessor und 8 GB Arbeitsspeicher.

Für einige Szenen wurden Ground-Truth Datensätze erstellt um ein Maß der Genauigkeit und Wiederholbarkeit für verschiedene Einstellungen zu erhalten. Hierfür wird ein anerkanntes Fehlermaß der Objektdetektion und des Trackings herangezogen. Unter anderem findet es in der Datenbank des „*Pattern Analysis, Statistical Modelling and Computational Learning*“ (*PASCAL*) Anwendung [21]. Für jedes Bild stehen manuell gelabelte Rechtecke um das Objekt **A** und die geschätzte Position aus dem Trackingalgorithmus **B** zur Verfügung. Eine Genauigkeitsangabe  $S(A, B)$ , die neben Positionsgenauigkeit zusätzlich die geschätzte Skalierung bewertet,



**Abbildung 6.1:** Evaluierung des Trackingergebnisses mit Ground-Truth Daten als Verhältnis der Schnittmenge (links) zur Vereinigungsmenge (rechts) der Ground-Truth Daten (A) und Trackingergebnisse (B)

kann durch folgende Formel ausgedrückt werden (siehe auch Abbildung (6.1)).

$$S(A, B) = \frac{\text{Area}(A \cap B)}{\text{Area}(A \cup B)}, \quad (6.1)$$

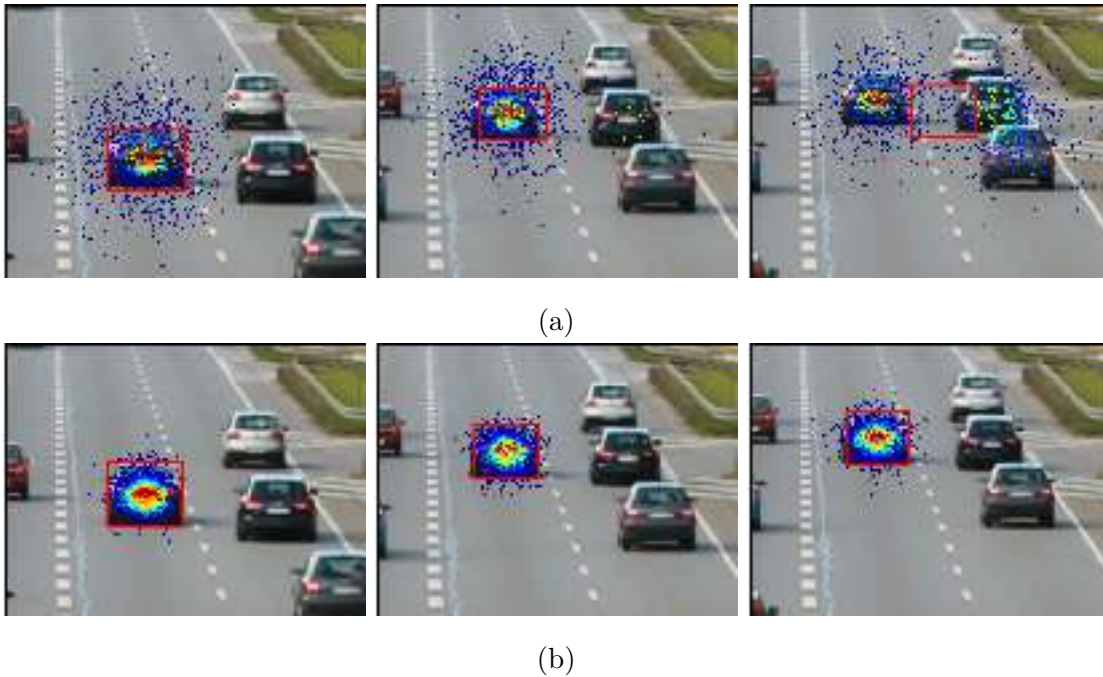
Das Bewertungsmaß, im folgenden *Konfidenz* genannt, kann dabei Werte zwischen  $[0, 1]$  annehmen, wobei 1 eine perfekte Übereinstimmung in Position und Größe des Templates und 0 ein Verlust des Objekts bedeutet (Rechtecke schneiden sich nicht).

## 6.1 Einfluss der Parameter des Partikel Filters auf Laufzeit und Robustheit

Zunächst wird untersucht, inwiefern verschiedene Parameter des Partikel Filters das Tracking-ergebnis beeinflussen und welchen Einfluss sie auf die Laufzeit des Programms haben. Die wichtigsten Parameter sind **Wahl des Prozessrauschens**, **Parameterwahl der Bildmerkmale** und **Partikelanzahl**.

### 6.1.1 Wahl des Prozessrauschens

Das Prozessrauschen gibt die Unsicherheit des Bewegungsmodells an und hat damit direkten Einfluss auf die Größe des Suchraums. Hohes Prozessrauschen resultiert darin, dass die Streuung der Partikel zunimmt. Dies ermöglicht, dass ein verlorenes Objekt wiedergefunden werden

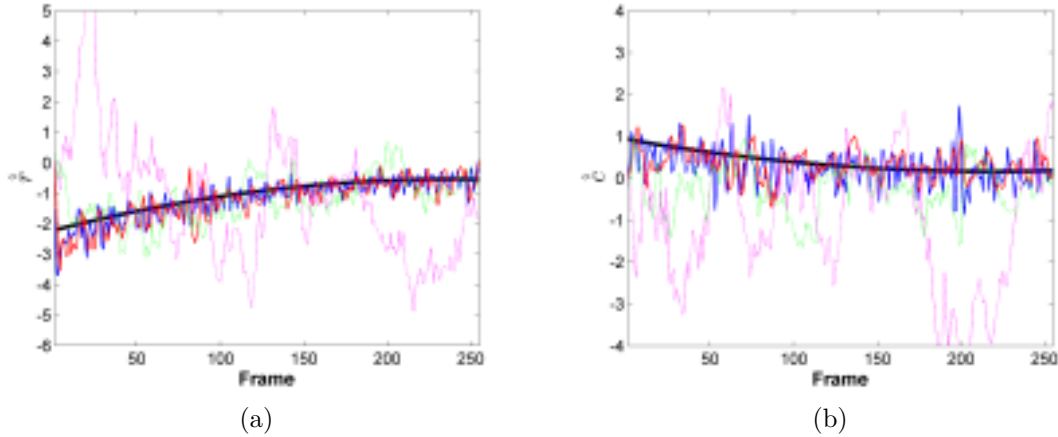


**Abbildung 6.2:** Auswirkungen unterschiedlichen Prozessrauschens auf die Trackinggenauigkeit in der Nähe von Störobjekten für (a) hohe Streuung und (b) niedrige Streuung. Steigendes Partikelgewicht von blau nach rot.

kann, aber auch, dass der Tracker zu benachbarten, ähnlichen Objekten springen kann. Dagegen kann mit einer kleinen Streuung ein verlorenes Objekt nur schwer wiedergefunden werden, wobei Störungen benachbarter Objekte den Tracker nicht beeinflussen, da sie nicht in den Suchbereich des Trackers fallen. Die Wahl der Streuung hängt stark von der Objektgröße und der Szene ab. Wird ein Objekt verfolgt, dessen Bewegung sehr konstant ist und von anderen Objekten isoliert ist, kann die Streuung sehr gering gehalten werden, da die lineare Bewegung gut mit den Bewegungsmodell modelliert werden kann. Sollte die Bewegung komplexer sein oder eine geringere Bildfrequenz vorliegen, sollte von einer höheren Unsicherheit ausgegangen werden. Abbildung (6.2) zeigt eine Szene in der ein dunkles Auto verfolgt werden soll. Auf der benachbarten Spur befindet sich ein ähnliches, dunkles Auto. Wird ein hohes Prozessrauschen gewählt, wird von einem großen Suchbereich ausgegangen (a). Im Laufe der Sequenz können zufällig Partikel auf das benachbarte Auto fallen. Da sich dieses Auto im Merkmalsraum, sowohl anhand der Farbverteilung, als auch in den Kanten dem Referenzobjekt ähnelt, können hohe Partikelgewichte ebenfalls in der Nähe des zweiten Auto entstehen. Diese zunächst wenigen Partikel (Mitte) werden aufgrund des Resamplings dupliziert, woraus sich vermehrt Partikel mit hohem Gewicht am falschen Auto aufhalten (rechts). Es entstehen zwei Peaks in der Wahrscheinlichkeitsverteilung. Die Schätzung mit MMSE liegt in der Mitte der beiden Peaks, bis sich schließlich eine Hypothese durchsetzt. Wird ein MAP-Schätzer verwendet, könnte das richtige Auto zwar länger korrekt verfolgt werden, jedoch wird der Tracker, sobald der zweite Peak das globale Maximum erreicht, direkt auf das falsche Auto springen.

Bei Verwendung eines geringen Prozessrauschens (b) wird im Laufe der Sequenz der Suchbereich des Trackers nicht auf das benachbarte Auto fallen, somit können auch keine Fehldetektionen entstehen. Für eine automatische Wahl der Positionsstreuung kann diese abhängig von der Objektgröße gewählt werden ( $\sigma_r = 1/10 \cdot h_0$  und  $\sigma_c = 1/10 \cdot w_0$ ). Für die Wahl des Prozessrauschens der Objektskalierung ergaben sich bei  $\sigma_s = 0.03$  für alle Szenen zufriedenstellende Ergebnisse. Damit können Abweichungen von 3% der zum letzten Frame geschätzten Objektskalierung berücksichtigt werden.

Weiterhin hat die Wahl des Prozessrauschens einen Einfluss auf die Genauigkeit der Geschwindigkeitsschätzung (innerhalb der Bildebene). Sollte sich die Position des Objekts zwischen zwei Bildern nur gering verändern (langsame Bewegung, und/oder hohe Bildfrequenz), kann die Positionsstreuung ( $\sigma_r, \sigma_c$ ) größer ausfallen als die tatsächliche Objektgeschwindigkeit. Dadurch kann keine signifikante Geschwindigkeit geschätzt werden. Abbildung (6.3) zeigt die geschätzten Geschwindigkeiten für unterschiedliche Positionsstreuungen. Je geringer die Positionsstreuung gewählt wird, desto genauer können die Geschwindigkeiten geschätzt werden. Jedoch verkleinert sich dadurch ebenfalls der Suchraum, woraus ein robustes Tracking bei erschwerten Bedingungen nicht immer gewährleistet werden kann. Im Falle einer Verdeckung muss eine gute Kenntnis über die bisherige Geschwindigkeit vorliegen, damit eine Vorhersage über die Lokalisierung während der Verdeckung getroffen werden kann, aber zugleich sollte der Suchraum groß genug sein, um



**Abbildung 6.3:** Auswirkung des Prozessrauschens auf die Geschwindigkeitsschätzung entlang der Bildzeile (a) und der Bildspalte (b). (schwarz) Geschwindigkeiten aus Ground-Truth Daten, (rot) aus der Differenz der Positionsschätzungen, (blau) für  $\sigma_{r,c} = 0$ , (grün) für  $\sigma_{r,c} = 5$  und (lila) für  $\sigma_{r,c} = 10$ .

Fehlprädiktionen während der Verdeckung ausgleichen zu können. Letzteres kann ausschließlich durch die Wahl einer hohen Positionsstreuung erreicht werden. Die Geschwindigkeit des Objekts kann jedoch ebenfalls aus der Differenz der geschätzten Positionen zweier aufeinanderfolgenden Frames ermittelt werden (rot). Durch die Differenzbildung kann also, sofern das Objekt präzise verfolgt wird, mit hohem Positionsrauschen eine ähnliche Geschwindigkeitsschätzung erzielt werden, wie bei gering gewählter Positionsstreuung.

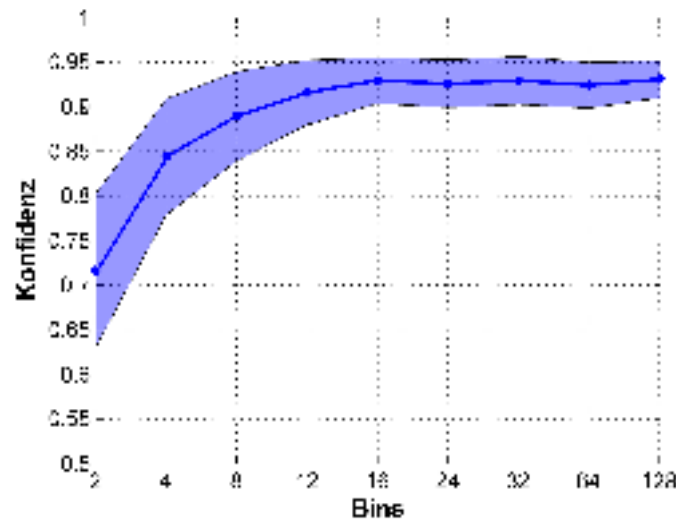
Zudem sollte bei geeigneten Szenen, in denen sich die Größe der Objekte stark verändern kann, eine skalierungsabhängige Positionsstreuung berücksichtigt werden. Je weiter sich die Objekte von der Kamera entfernen, desto kleiner erscheinen sie und desto näher ist der Abstand zwischen zwei Fahrspuren. Bleibt die Positionsstreuung über die gesamte Sequenz konstant, kann sie für weit entfernte Objekte einen zu großen Suchraum über mehrere Spuren abdecken und erhöht damit die Gefahr von Fehldetektionen (vergleiche Abbildung (6.2)). Daher sollte die Positionsstreuung abhängig von der aktuellen Objektskalierung gewählt werden ( $\sigma_r = s_k \cdot 1/10 \cdot h_o$ ,  $\sigma_c = s_k \cdot 1/10 \cdot w_o$ ).

Auf die Laufzeit hat die Wahl des Prozessrauschens nur Einfluss bei Verwendung der Kantenrichtungshistogramme. Die Erstellung des Kantenbildausschnittes ist, wie in Kapitel 5.4.2 gezeigt, abhängig von der Streuung der Partikel. Bei einer geringeren Streuung fallen die zu berechnenden Gradientenbilder kleiner aus und damit ergibt sich ein geringerer Rechenaufwand.

### 6.1.2 Wahl der Merkmalsparameter

Die Extraktion der Bildmerkmale ist ausschlaggebend für die Genauigkeit, aber zugleich der rechen-intensivste Teil der Filterung. Im Folgenden werden die wichtigsten Parameter und deren Auswirkungen auf die Ergebnisse, sowie Laufzeit erläutert.

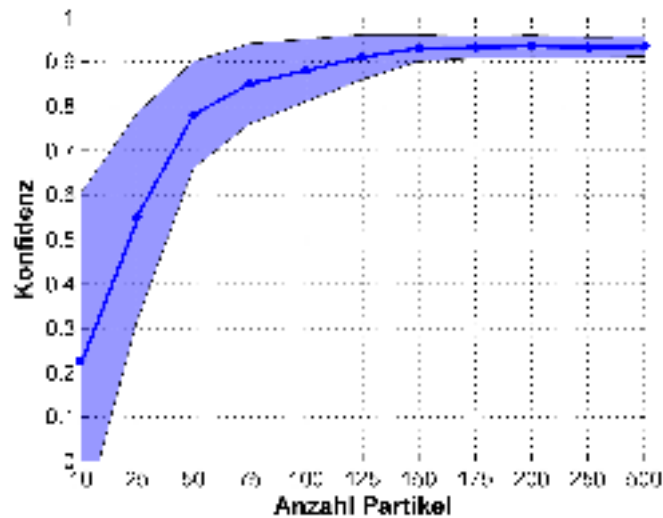




**Abbildung 6.4:** Einfluss der Bin-Anzahl auf die Robustheit des Farbtrackings anhand Szene 3. Mittelwert und Standardabweichung für 20 Simulationen mit 150 Partikel bei Verwendung von einem Block.

**Die Anzahl der Farbkanäle** hat Einfluss auf die Robustheit, sowie auf die Laufzeit des Trackings. Grundsätzlich gilt, je mehr Kanäle verwendet werden, desto mehr Information steht zur Verfügung und desto robuster kann das Objekt verfolgt werden. Jedoch geht die Anzahl der Kanäle linear in die Laufzeit des Trackings ein, da bei Verwendung von Farbmerkmalen für jeden Kanal Histogramme berechnet und miteinander verglichen werden müssen. Für die Verwendung von Kanten erhöht sich die Laufzeit ausschließlich bei der Erstellung der Kanten- und Orientierungsbilder. Für die Wahl der stärksten Kanten werden die Gradientenamplituden- und orientierungsbilder für jeden Kanal erstellt und schließlich für jedes Pixel der Farbkanal mit der höchsten Gradientenamplitude verwendet. Tests für Tracking in Grauwertbildern haben gezeigt, dass trotz einkanaligen Videos in der Regel ein robustes Tracking möglich ist. Für die Kantenrichtungshistogramme konnten keinerlei Unterschiede festgestellt werden. Lediglich bei Verwendung von Farbverteilungen sollte auf eine größere Anzahl von Bins zurückgegriffen werden, damit ein detailliertes Modell erstellt werden kann. Es kann allerdings eine größere Störung ähnlicher Objekte festgestellt werden. Für ein robustes Tracking sollten daher, wenn möglich, drei-kanalige Videos verwendet werden, jedoch können (bei nicht zu hoher Verkehrsdichte) auch bei Grauwertvideos gute Ergebnisse erzielt werden.

**Die Anzahl der Bins** stellt den Detaillierungsgrad des Modells dar. Werden zu wenige Bins verwendet, kommt es häufig zu Störungen benachbarter Objekte oder des Hintergrunds. Zu viele Bins ergeben ein sehr starres Modell, bei dem kleine Veränderungen im Laufe der Sequenz zu starken Unterschieden der Histogramme führen. Für die Laufzeit sind die Histogrammvergleiche ausschlaggebend. Je mehr Bins verwendet werden, desto größer werden die Dimensionen der Merkmalsvektoren und desto mehr Rechenoperationen sind pro Histogrammvergleich notwendig. Jedoch fällt diese Erhöhung der Laufzeit im Vergleich zur Merkmalsextraktion nur sehr gering

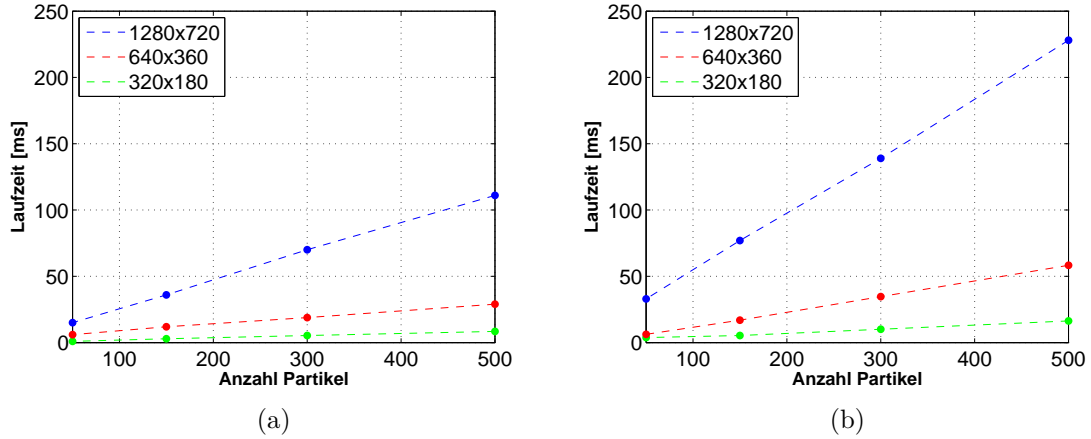


**Abbildung 6.5:** Einfluss der Partikelanzahl auf die Robustheit des Farbtrackings anhand Szene 3. Mittelwert und Standardabweichung für 20 Simulationen bei Verwendung von 16 Bins bei Verwendung von einem Block.

aus. In mehreren Tests anhand einer Szene wurden für ein Farbtracking alle Parameter, bis auf die Bin-Anzahl fixiert. Bei einer Verwendung von 4 Bins und 256 Bins konnte eine Erhöhung der Laufzeit um 8% festgestellt werden. Zudem wurden Untersuchung über den Einfluss der Bin-Anzahl auf die Robustheit des Trackings durchgeführt (vergleiche Abbildung (6.4)). Für die hier verwendete Szene (Szene 3 aus Abbildung (6.11a)) kann ab der Verwendung von 16 Bins keine weitere Genauigkeitssteigerung mehr erreicht werden. Dies ist jedoch abhängig von der Farbe des Objekts und der Umgebung. Für dunkle Autos auf hellem Hintergrund kann auch mit zwei Bins robust getrackt werden. Sollten sich Objekt und Hintergrund ähneln, müssen mehrere Bins verwendet werden. Jedoch zeigte sich in diversen Tests, dass 16x16x16 Bins bei dreikanaligen Bildern für ein robustes Tracking unter verschiedensten Bedingungen ausreicht und rechen-effizient durchführbar ist.

**Die Anzahl der Blöcke** haben einen großen Einfluss auf die Genauigkeit des Trackings. Dies wird explizit in Kapitel 6.2.2 untersucht. Besonders für die Schätzung der Objektskalierung sollten mehrere Blöcke verwendet werden. Ähnlich der Bin-Anzahl steigt die Laufzeit des Trackings mit Zunahme der Block-Anzahl. Für jeden Block wird ein gleichartiges Modell aufgestellt, das abhängig von den oben genannten Parametern ist. Somit steigt auch hier die Anzahl der Histogrammvergleiche linear mit der Block-Anzahl.

**Die Wahl des Messrauschens** hat zwar keinen Einfluss auf die Laufzeit des Trackings, ist dafür allerdings ausschlaggebend für die Genauigkeit des Trackings. Dieser Wert kann nicht global angegeben werden, da er stark abhängig von der Parameterwahl der verwendeten Bildmerkmale ist (unterschiedliche Sensitivität der Bildmerkmale, Bin-Anzahl, Block-Anzahl, Anzahl der Farbkanäle). Die Auswirkungen unterschiedlichen Messrauschens wurden in Kapitel 5.4.1 dar-



**Abbildung 6.6:** Abhängigkeit der Partikelanzahl auf die Laufzeit bei Verwendung von (a) Farbhistogrammen und (b) Kantenrichtungshistogrammen bei unterschiedlichen Bildauflösungen. Initialisierungsgröße des Objekts von 120x120 Pixel bei voller Auflösung.

gestellt. Grundsätzlich sollte es so gewählt werden, dass die Wahrscheinlichkeitsverteilung weder zu breit (hohes Messrauschen erhöht Anfälligkeit gegenüber Störobjekten), aber auch nicht zu schmal (sehr geringes Messrauschen bedeutet ein starres Vertrauen auf das Modell und damit eine geringe Fähigkeit mit Modellveränderungen umgehen zu können; zudem werden die meisten Partikel ein vernachlässigbar kleines Gewicht erhalten, was die Auswertung der Verteilung erschwert) ausfällt.  $\sigma_{col} = 0.04$  ergab für die Verwendung von vier Blöcken und 16 Bins bei Farbhistogrammen und  $\sigma_{edge} = 0.05$  für neun Bins und neun Blöcken bei Kantenrichtungshistogrammen in den meisten Fällen zufriedenstellende Ergebnisse.

### 6.1.3 Anzahl der Partikel

Die Partikelanzahl stellt eine kritische Größe dar. Sie ist verantwortlich für die Dichte der approximierten Wahrscheinlichkeitsverteilung und damit zugleich für die Genauigkeit. Die Schätzung des A-Posteriori Zustands kann mit steigender Partikelanzahl genauer vollzogen werden, jedoch erhöht sich auch der Rechenaufwand linear mit der Anzahl der Partikel. Da für jeden Partikel eine Merkmalsextraktion durchgeführt werden muss, sollte die Anzahl der Partikel nicht zu hoch gewählt werden, um ein möglichst echtzeitfähiges System zu gewährleisten (vergleiche die Abhängigkeiten auf die Laufzeit im vorigen Kapitel). Mit einer sehr geringen Partikelanzahl kann die Wahrscheinlichkeitsverteilung nicht mehr repräsentativ approximiert werden, wodurch eine schlechte Schätzung folgt. Oft kann zwar das Objekt auch mit einer sehr geringen Anzahl von Partikeln verfolgt werden, jedoch reicht bei leichter Veränderung der Objekterscheinung oder kurzzeitiger Verdeckung die Anzahl der Partikel nicht mehr aus um das Objekt wiederzufinden. Um dies zu testen, wurden erneut für Szene 3 sämtliche Parameter fixiert und lediglich die Partikelanzahl variiert. Es wurde ein Fartracking mit einem Block, 16 Bins und  $\sigma_{col} = 0.04$  durchgeführt. Abbildung (6.5) zeigt die mittlere Genauigkeit, sowie Standardabweichung für den Vergleich mit einem Ground-Truth Datensatz entlang der gesamten Szene für jeweils 20 Simu-

lationen. Es lässt sich eine Genauigkeitssteigerung bei zunehmender Partikelanzahl feststellen. Zugleich verringert sich die Standardabweichung, die als Maß für die Robustheit des Trackers zu verstehen ist und darstellt, inwiefern eine Wiederholbarkeit des Ergebnisses gegeben ist. Die Genauigkeit steigt bis zu etwa 150 Partikel an, erhöht sich von da an allerdings nicht mehr signifikant. Ein ähnliches Verhalten kann bei Verwendung von Kantenrichtungshistogrammen festgestellt werden. In Abbildung (6.6) werden Laufzeit für Farb- und Kantenmerkmale abhängig von der Partikelanzahl aufgetragen (blau). Der lineare Verlauf und die höhere Berechnungszeit der Kantenmerkmale werden ersichtlich. Die Laufzeiten beziehen sich dabei auf die Auswertung eines Frames für ein Objekt mit der Initialgröße von 120x120 Pixel. Die Abnahme der Objektgröße im Laufe der Sequenz führt dabei zu einer Beschleunigung des Trackings. Die Verwendung von 150 Partikeln bei voller Bildauflösung kann zur Echtzeitfähigkeit des Farbtracking, jedoch nicht für das Kantentracking verwendet werden. Allerdings wird in Kapitel 6.3.1 und 6.3.2, ob diese Bildauflösung und 30 Hz überhaupt für ein robustes Tracking notwendig ist.

## 6.2 Testen unterschiedlicher Szenen

Anhand realer Szenen soll im Folgenden die vielfältige Einsatzfähigkeit des Trackingsystems getestet werden. Ziel soll es sein, ein möglichst robustes System für verschiedene Verkehrssituationen zu realisieren. Aufnahmen von Verkehrsüberwachungskameras können sich dabei anhand ihrer Aufnahmeperspektive unterscheiden und stellen den Tracker dabei vor unterschiedliche Herausforderungen. Die nachfolgenden Szenen unterscheiden sich vor allem in ihrer Aufnahmeperspektive, beginnend mit Aufnahmen vom Straßenrand. Typische Autobahnaufnahmen, gefilmt von einer Brücke stellen große Herausforderungen in variierenden Objektskalierungen, sowie Verdeckungen dar. Schließlich soll anhand einer nahezu vertikalen Aufnahme von einer Kreuzung die Fähigkeit des CVR-Bewegungsmodells getestet werden. Nachfolgende Tabelle enthält eine Übersicht über die vorhandenen Szenen.

Aufnahmeperspektive	Szene	Beschreibung
Horizontal	Szene 1	Kurzzeitige Teil- und Vollverdeckung
	Szene 2	Langanhaltende Vollverdeckung
Schräg	Szene 3	Ändernde Objektskalierung
	Szene 4	Beleuchtungsänderung
	Szene 5	Gegenseitige Teilverdeckung
	Szene 6	Gegenseitige Vollverdeckung
Vertikal	Szene 7	Abbiegende Autos, Passanten

Für die untersuchten Szenen werden die in der nachfolgenden Tabelle aufgelisteten Parameter verwendet, falls nicht anders angegeben.

	Parameter	Wert
Partikelanzahl	N	150
Prozessrauschen	$\sigma_r$	$s_k \cdot \mathcal{N}(0, 1/10 \cdot h_0)$
	$\sigma_c$	$s_k \cdot \mathcal{N}(0, 1/10 \cdot w_0)$
	$\sigma_{\dot{r}}, \sigma_{\dot{c}}$	2
	$\sigma_s$	0.03 oder <i>extern</i>
Messrauschen	$\sigma_{col}$	0.04
	$\sigma_{edge}$	0.05
Merkmalsparameter	Bins <sub>col</sub>	16
	Bins <sub>edge</sub>	9
	Blockanzahl	<i>variabel</i>
	Farbkanäle	$R, G, B$

### 6.2.1 Aufnahmen vom Straßenrand

Aufnahmen vom Straßenrand und quer zur Fahrtrichtung sind dadurch charakterisiert, dass sich Objekte linear und in Richtung der Bildzeile bewegen. Problematisch sind hierbei allerdings Verdeckungen, die entweder durch entgegenkommende Fahrzeuge oder auch mit dem Bildhintergrund entstehen können. Abbildung (6.7) zeigt zwei Szenen, die diese eben genannten Probleme aufzeigen. Im Folgenden werden die Ergebnisse des Trackingsystems anhand dieser Sequenzen dargelegt.

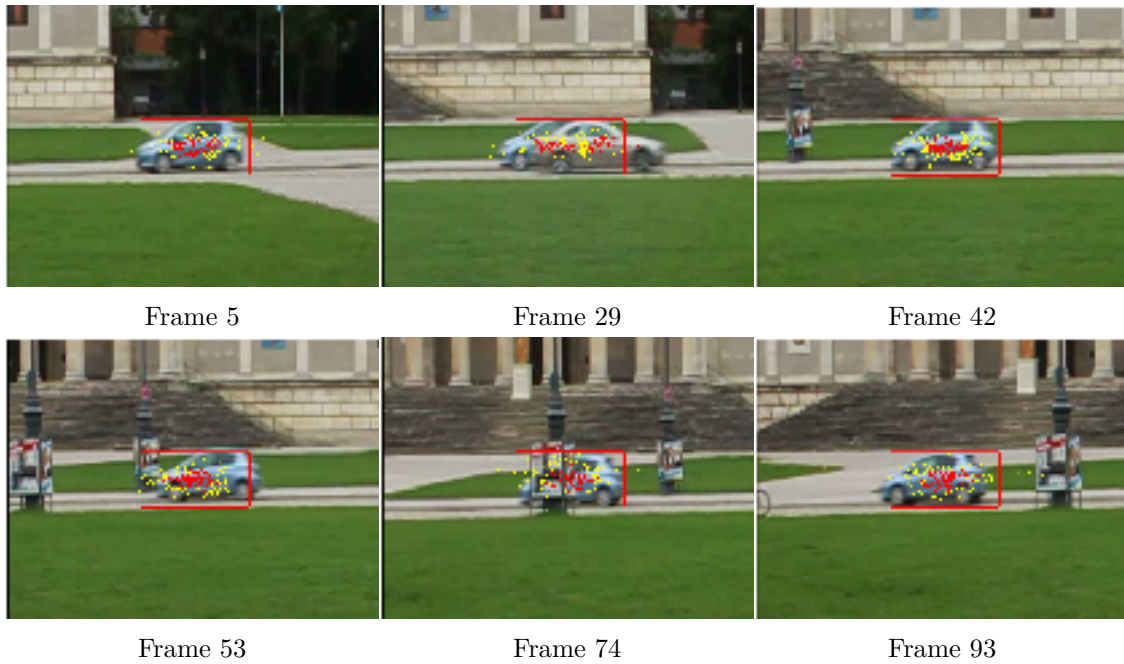


(a) Szene 1



(b) Szene 2

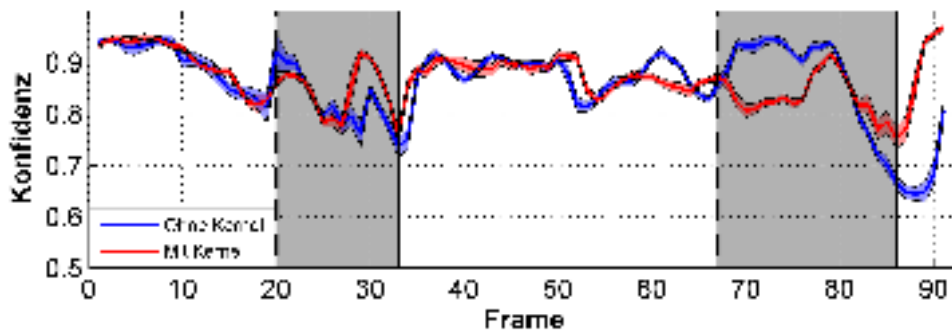
**Abbildung 6.7:** Testsequenzen von Straßenrandaufnahmen. (a) Kurzzeitige Komplettverdeckung mit Gegenverkehr und Teilverdeckung mit Bildhintergrund. (b) Länger anhaltende Komplettverdeckung mit Bildhintergrund



**Abbildung 6.8:** Trackingergebnisse für Szene 1 bei Verwendung von Farbmerkmalen mit einem Block.

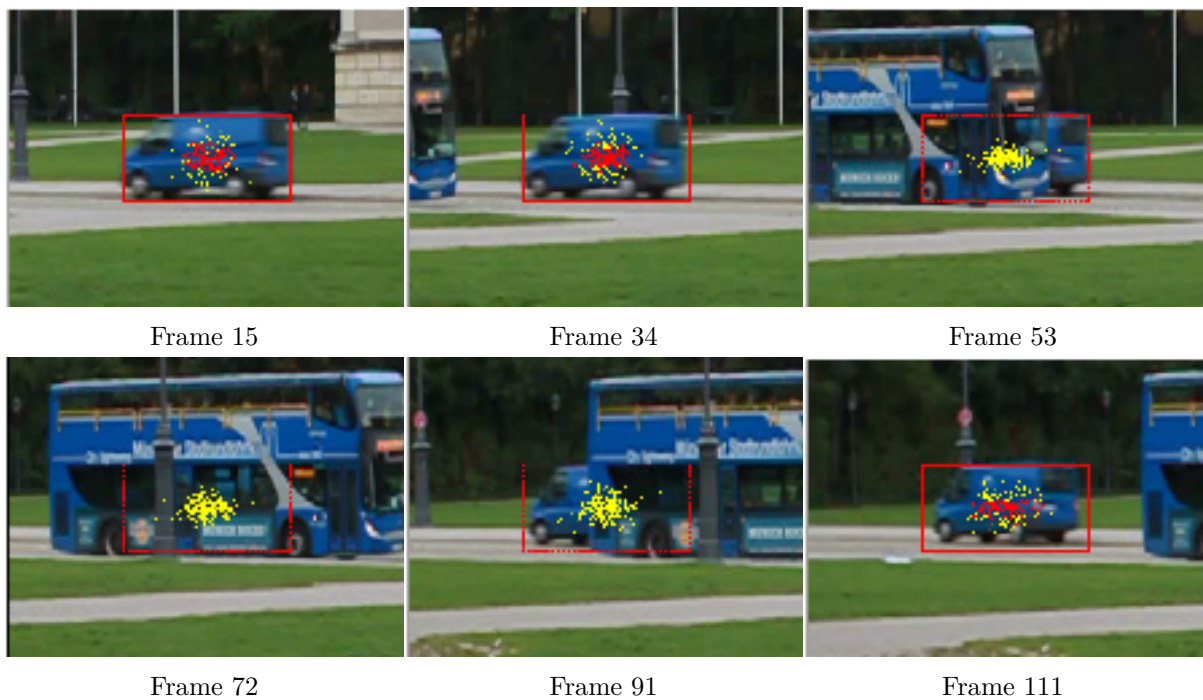
**Szene 1** In der ersten Sequenz soll ein hellblaues Auto verfolgt werden, dass sich von rechts nach links in der Bildebene bewegt. Währenddessen wird es zeitweise von einem entgegenkommenden Auto komplett verdeckt, erscheint wieder und erleidet einer Teilverdeckung hinter einer an einer Straßenlaterne befestigten Werbetafel. Siehe auch Abbildung (6.7a).

Das Modell eines rechteckigen Templates eignet sich für die Seitenansicht eines dieser Autos nur bedingt, da viel Hintergrundinformation über der Motorhaube des Autos mit eingeht. Daher wird hier zusätzlich eine Gewichtungsfunktion in Form eines *Epanechnikov Kernels*, wie in Kapitel 4.1.2 und Gleichung (4.2) und (4.3) verwendet, um den Einfluss des Hintergrunds abzuschwächen. Durch das Vorwissen, dass sich Objekte in dieser Szene entlang der Bildzeile



**Abbildung 6.9:** Genauigkeit und Standardabweichung des Trackings für Szene 1 ohne Verwendung einer Gewichtungsfunktion (blau) und mit Verwendung eines Epanechnikov Kernels als Gewichtungsfunktion (rot). (grau) Bereiche der Verdeckungen.





**Abbildung 6.10:** Trackingergebnisse für Szene 2 bei Verwendung von Farbmerkmalen mit einem Block.

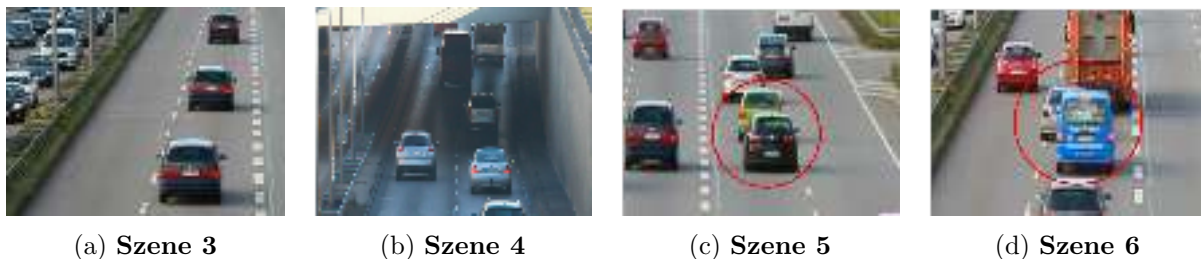
bewegen, kann die Streuung der Partikel durch geeignete Wahl des Prozessrauschens in Richtung der Bildzeile verkleinert werden. Ebenso werden Objekte anhand der Perspektive keine Größenänderung widerfahren. Deswegen kann auf die Schätzung eines Skalierungsfaktors in diesem Fall verzichtet werden. Als Bildmerkmale wurde das Farbhistogramm ohne Unterteilung in mehrere Blöcke gewählt.

In den Bereichen ohne Verdeckung kann das Objekt problemlos verfolgt werden. Abbildung (6.8) stellt Ausschnitte aus dem Trackingprozess dar. Die dunklen Bereiche in Abbildung (6.9) zeigen die Verdeckungszeiträume. Der linke Bereich beinhaltet die Teil- und Vollverdeckung mit dem Gegenverkehr, der rechte die Teilverdeckung mit der Werbetafel. In Frame 29, kurz nach der Vollverdeckung, kann man anhand der geresampten Partikel (rot) erkennen, dass die Wahrscheinlichkeitsverteilung zwei Modi und damit zwei Haupthypothesen enthält. Eine gilt für das korrekte Objekt, die andere für den Gegenverkehr. Hier zeigt sich der Vorteil des Partikel Filters. Innerhalb der nächsten Frames werden beide Hypothesen verfolgt, aber schließlich das korrekte Objekt weiterverfolgt. Die Teilverdeckung mit der Werbetafel kann ebenfalls bewältigt werden. Es besteht zwar eine Störung, die zum einen mit der ähnlichen Farbverteilung der Werbetafel (Frame 74), aber auch dem teilweise grauen Hintergrund (Frame 93) zusammenhängt. Dennoch können sich sowohl der Tracker ohne Kernel als auch derjenige mit Kernel vom Störobjekt lösen und das Auto weiterverfolgen. Aufgrund der ähnlichen Farbverteilung der Werbetafel in den mittleren Bereichen Autos wird der Tracker mit dem Kernel stärker beeinflusst und gestört, kann aber nach Wiedererscheinen des Objekts einen schnelleren Anschluss an das Auto finden (siehe Abbildung (6.9)).

**Szene 2** In der zweiten Sequenz soll ein blauer Van verfolgt werden, der für mehrere Bilder hinter einem ebenfalls blauen Bus verschwindet und anschließend wieder erscheint (Abbildung (6.7b)). Es werden die selben Einstellungen wie in Szene 1 verwendet. Aufgrund der ähnlichen Farbe fällt eine automatische Detektion der Verdeckung schwer, da Bildmerkmale des Busses dem Van sehr ähnlich sind. Die Verdeckung ist zudem zu lang, um das Wiedererscheinen des Vans erfolgreich zu detektieren. Ohne explizite Berücksichtigung einer Vollverdeckung kann das Objekt nicht vollständig verfolgt werden. Eine erfolgreiche Verfolgung bis zum Verschwinden des Vans kann realisiert werden, danach bleibt der Tracker jedoch während der Verdeckung in einer Region des Busses hängen, in der die Farbverteilung am ehesten der des Vans entspricht. Daher wird eine Prädiktion des Objektes während der Verdeckung verwendet, wie in Kapitel 5.5.1 vorgestellt. Durch die lineare Objektbewegung, tritt bei der Prädiktion kein Problem auf und das Objekt kann selbst bei nicht exakter Vorhersage problemlos wiedergefunden werden (siehe Abbildung (6.10)). Trotz ähnlicher Farbe von Van und Bus wurde bei mehrfachen Simulationen nach der Vorhersage richtigerweise immer der Van weiterverfolgt, nie der Bus.

### 6.2.2 Autobahnsequenzen

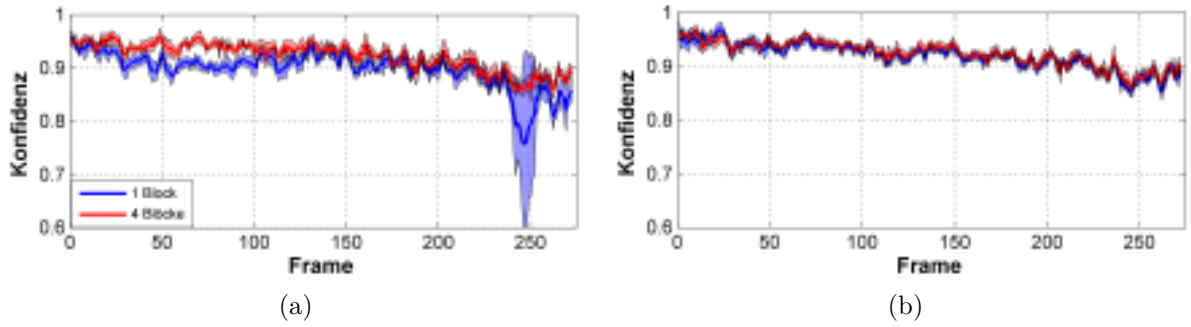
Aufnahmen von Autobahnbrücken stellen größere Herausforderungen dar. Bedingt durch die Aufnahmegeometrie ändern die Objekte je nach Entfernung zur Kamera ihre Größe in der Bildebene. Dies muss durch eine zusätzliche Schätzung der Skalierung berücksichtigt werden. Des weiteren können sich die Objekte gegenseitig verdecken, was ein robustes Tracking erschwert. Beleuchtungsunterschiede in der Szene erschweren das Tracking mit Farbe. Im Folgenden werden vier Sequenzen untersucht, die in Abbildung (6.11) dargestellt sind. Szene 3 (a) stellt ein von weiteren Objekten isoliertes Auto dar, anhand dessen verschiedene Trackingparameter untersucht werden. In Szene 4 (b) werden Beleuchtungsunterschiede behandelt, die durch Autos in einem unbeleuchteten Tunnel entstehen. Szene 5 (c) stellt das Problem von gegenseitigen Teilverdeckungen und Szene 6 (d) schließlich kurzzeitige Vollverdeckungen zweier Objekte dar.



**Abbildung 6.11:** Testsequenzen von Aufnahmen von Autobahnbrücken. (a) Schätzung der Objektskalierung. (b) Testen bei Beleuchtungsänderungen. (c) Gegenseitige Teilverdeckung. (d) Gegenseitige Vollverdeckung

**Szene 3** In dieser Szene soll ein dunkelblaues Auto verfolgt werden das sich von der Kamera wegbewegt und dadurch im Laufe der Sequenz kleiner wird. Es hebt sich klar vom Hintergrund ab, und wird nicht durch andere Objekte in der Nähe gestört. Anhand dieser Szene kann die





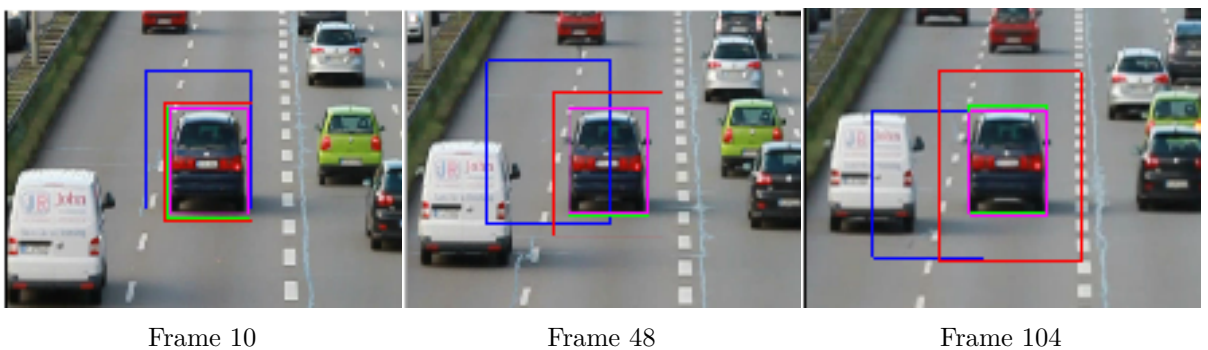
**Abbildung 6.12:** Genauigkeiten und Standardabweichungen für Szene 3 bei Verwendung von Farbmerkmalen für eine unterschiedliche Anzahl an Blöcken. (a) Schätzung der Objektskalierung mit dem Partikel Filter und (b) externe Vorgabe der Objektskalierung.

Leistungsfähigkeit beider Bildmerkmale getestet werden.

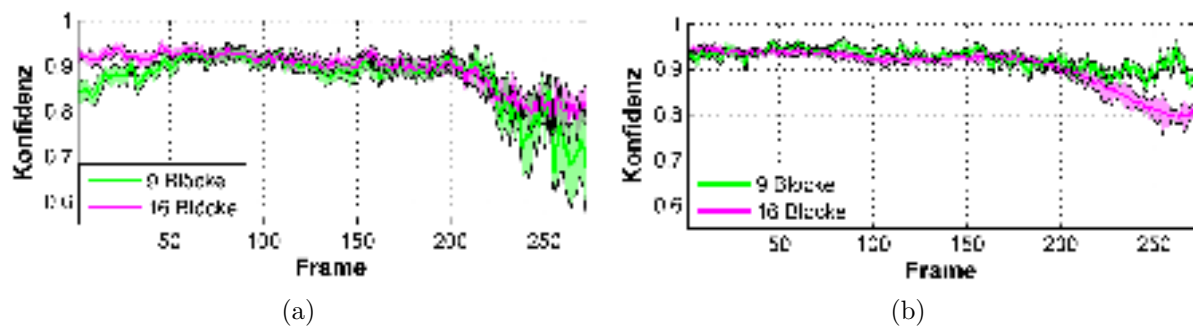
Die Modelldarstellung mit einem Rechteck eignet sich für diese Aufnahmen sehr gut, ohne dass Hintergrundinformation in das Modell eingeht, daher kann von der Einführung einer Gewichtungsfunction abgesehen werden. Rotationen des Objekts können ebenfalls nicht auftreten, daher eignet sich die Verwendung des CV-Modells. Eine Schätzung der Skalierung ist allerdings durch die Aufnahmegeometrie notwendig um die Größenänderung abhängig von der Objektentfernung zur Kamera zu berücksichtigen.

Zunächst soll auf eine externe Schätzung der Skalierung verzichtet werden um die Leistungsfähigkeit des Trackers für die verschiedenen Bildmerkmale zu testen.

Die Verwendung von Farbhistogrammen ohne Unterteilung in Blöcke liefern gute Ergebnisse (siehe Abbildung (6.12a) blau). Positionsgenauigkeit und auch die Schätzung der Objektskalierung kann präzise bestimmt werden. Jedoch kann eine erhöhte Standardabweichung zum Ende der Sequenz festgestellt werden. Diese resultiert darin, dass in manchen Simulationen die Objektskalierung zu klein geschätzt wurde. Dieser „Schrumpfeffekt“ kann bei farblich homogenen



**Abbildung 6.13:** Probleme der Objektskalierungsschätzung bei Verwendung von Kantenmerkmalen für (blau) 1 Block und (rot) 4 Blöcken. Gute Ergebnisse für (grün) 9 Blöcke und (lila) 16 Blöcke.



**Abbildung 6.14:** Genauigkeiten und Standardabweichungen für Szene 3 bei Verwendung von Kantenmerkmalen für eine unterschiedliche Anzahl an Blöcken. (a) Schätzung der Objektskalierung mit dem Partikel Filter und (b) externe Vorgabe der Objektskalierung.

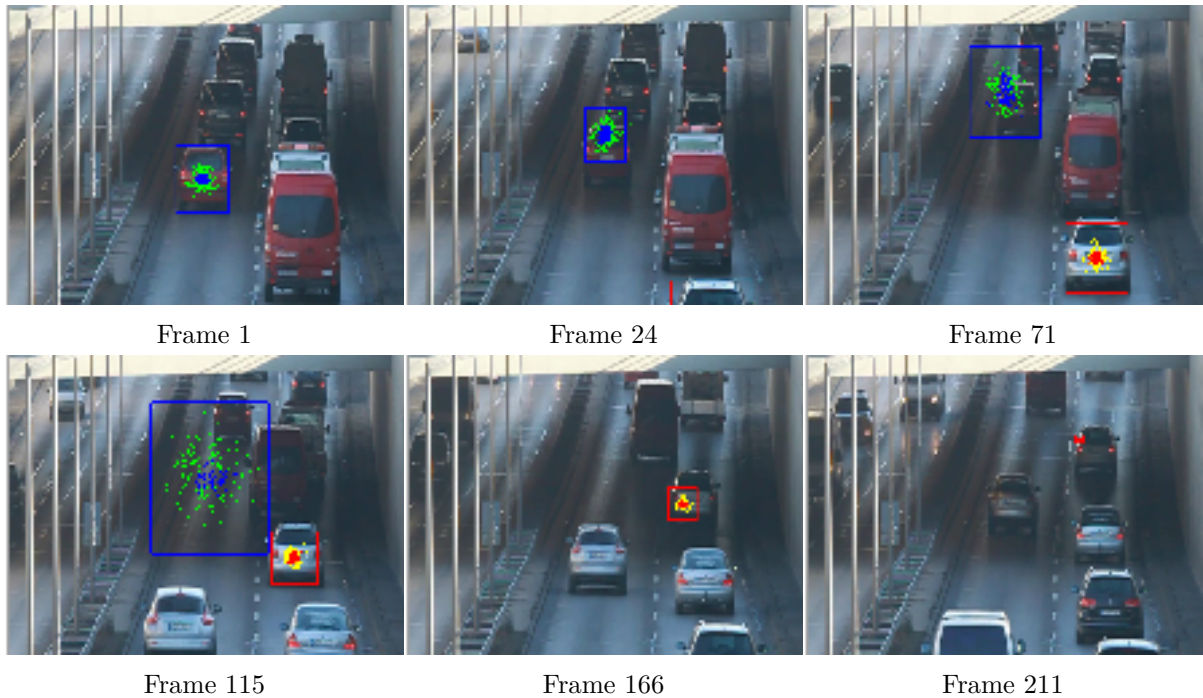
Objekten vorkommen, da die Farbverteilung in kleinen Bereichen des Objekts der Farbverteilung des gesamten Objekts ähnelt.

Durch die Verwendung von Multi-Part Regionen kann dieser Effekt abgeschwächt werden (rot). Die Standardabweichung erscheint ebenfalls etwas kleiner. Es kann also eine erhöhte Robustheit der Objektskalierungsschätzung festgestellt werden.

Wird die Skalierung anhand der Bildgeometrie vorgegeben (vergleiche Kapitel 5.6) und aus der Schätzung herausgenommen, liefern sowohl Gesamtregion, als auch Mehrfachregionen sehr genaue und robuste Ergebnisse (Abbildung (6.12b)). Es kann eine sehr geringe Standardabweichung und damit hohe Wiederholbarkeit festgestellt werden.

Bei isolierten und sich vom Hintergrund abgrenzenden Objekten können also anhand der Farbmerkmale sehr gute Ergebnisse erzielt werden. Die Unterteilung in mehrere Blöcke kann die Schätzung der Objektskalierung verbessern, wobei bei externer Vorgabe der Objektskalierung auf eine solche Unterteilung verzichtet werden kann.

Werden Kantenrichtungshistogramme als Bildmerkmale verwendet, können abhängig von der Anzahl der Blöcke ebenfalls sehr gute Ergebnisse erzielt werden. Abbildung (6.13) zeigt einige Ausschnitte aus der Szene und die Resultate für eine unterschiedliche Blockanzahl. Hier fällt zunächst deutlich auf, dass die Verwendung von einem (blau) und vier (rot) Blöcken bereits nach wenigen Frames zu schlechten Ergebnissen führt. Dies betrifft weniger die Positionsgenauigkeit als die Schätzung der Objektskalierung. Je weniger Blöcke für die Modellbildung verwendet werden, desto weniger wird eine räumliche Anordnung der Gradientenorientierungen berücksichtigt. Da die Straße als Objekthintergrund eine sehr homogene Fläche aufweist in denen keine starken Kanten vorhanden sind, haben diese keinen Einfluss auf das Histogramm. Somit entstehen bei zu groß gewählter Objektskalierung ähnliche Kantenrichtungshistogramme und damit hohe Gewichte der Partikel. Dadurch wird die Skalierung im Laufe der Sequenz einen Drift aufweisen und die Größe des Objekts immer weiter zunehmen. Durch die Verwendung von vier Blöcken wird dieser Drift abgeschwächt, jedoch nicht aufgehoben, woraus ein langsamerer Drift entsteht.



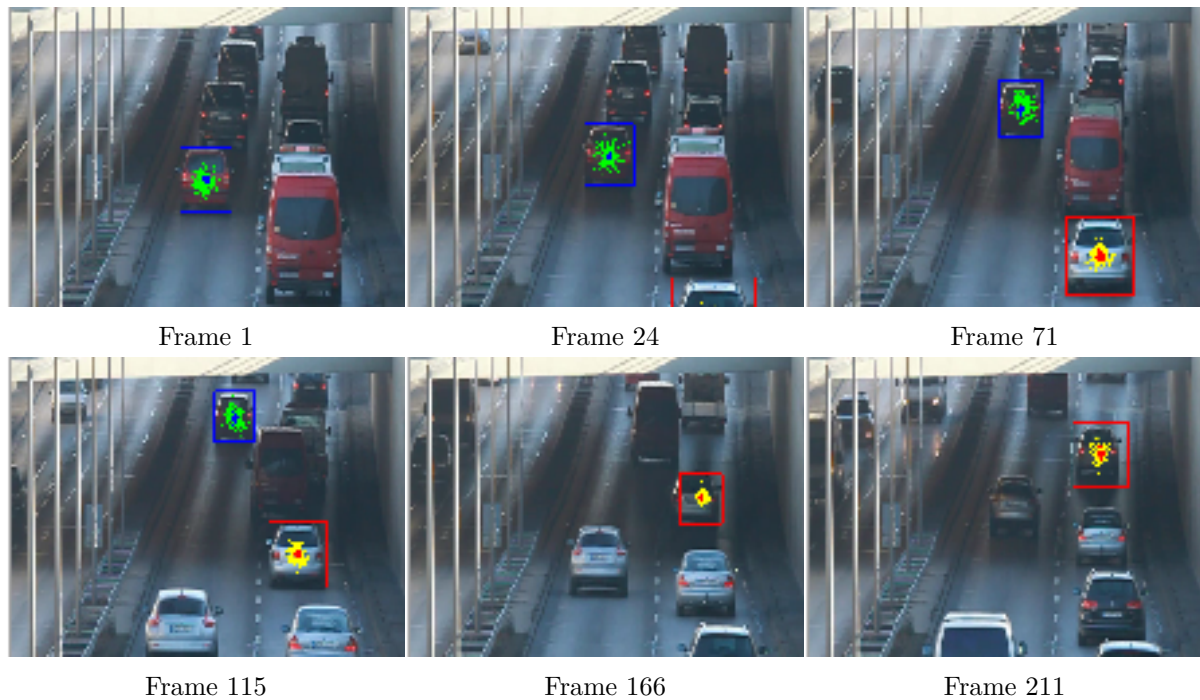
**Abbildung 6.15:** Trackingergebnisse für Szene 4 bei Verwendung von Farbmerkmalen mit 4 Blöcken.

Erst ab neun (grün) Blöcken kann ebenfalls die Objektskalierung robust geschätzt werden. Abbildung (6.14a) stellt die Genauigkeit und Robustheit für neun und 16 Blöcke dar, bei denen bis zu drei Viertel der Sequenz vergleichbare Ergebnisse wie bei der Verwendung von Farbmerkmalen erzielt werden können. Anschließend nimmt die Genauigkeit stark ab.

Wird die Objektskalierung extern vorgegeben (Abbildung 6.14b), kann eine zusätzliche Erhöhung der Genauigkeit und Robustheit festgestellt werden. Dennoch ergibt sich ein Genauigkeitsabfall bei Verwendung von 16 Blöcken. Dieser ist darauf zurückzuführen, dass aufgrund des kleiner werdenden Objekts zum Ende der Sequenz nur noch sehr wenige Pixel pro Block (hier: 12x14 Pixel bei Frame 254) anfallen, woraus keine signifikanten Kanten mehr detektiert werden können. Die Anzahl der Blöcke sollte daher auch abhängig von der Objektgröße gewählt werden.

**Szene 4** Diese Szene stellt die Problematik von Beleuchtungsänderungen während einer Sequenz dar, die in diesem Fall durch einen Tunnel hervorgerufen wird. Getestet werden Farb- und Kantenmerkmale für ein dunkles und ein helles Auto. Zudem kann eine geometrische Berechnung der Objektskalierung für diese Szene nicht angewendet werden, da die Straße eine Neigung aufweist. Somit muss die Schätzung der Objektskalierung innerhalb des Trackings erfolgen.

Abbildung (6.15) stellt die Ergebnisse für die Verwendung von Farbhistogrammen mit vier Blöcken dar. Durch die Tunnelfahrt ändern sich die Modelle der Farbhistogramme zu stark, um die Autos anhand ihrer Farbe während der Tunnelfahrt verfolgen zu können. Sobald der Schatten auf ein Auto fällt entstehen große Diskrepanzen zwischen Referenzmodell und den Objekthypo-



**Abbildung 6.16:** Trackingergebnisse für Szene 4 bei Verwendung von Kantenmerkmalen mit 16 Blöcken.

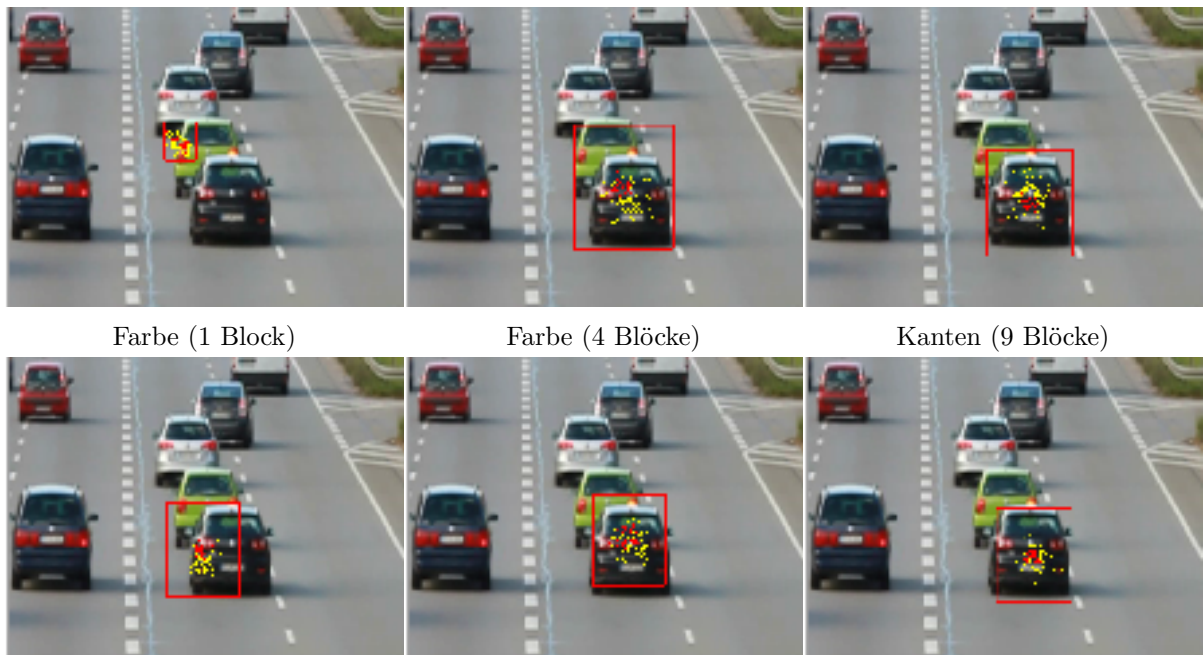
thesen der Partikel. Je nach Farbe des Autos und Umgebung entstehen willkürliche Bewegungen des Trackers. Ein Tracking der Objekte kann also durch die Verwendung der Farbe nicht gewährleistet werden. Um dieses Problem zu lösen könnte eine adaptive Modellkorrektur, wie in [42] vorgestellt, verwendet werden; allerdings ist der Übergang zu schnell um auf die starken Modellveränderungen reagieren zu können. Aus diesem Grund müssen alternative Bildmerkmale für ein erfolgreiches Tracking verwendet werden.

Kantenrichtungshistogramme sind beleuchtungsinvariant und liefern weitaus bessere Ergebnisse. Abbildung (6.16) stellt die Ergebnisse für die Verwendung von 16 Blöcken dar. Dabei können beide Objekte sehr robust über die gesamte Szene verfolgt werden und es kann ebenfalls eine ausreichend genaue Objektskalierung geschätzt werden. Dies gilt sowohl für das helle Auto als auch für das dunkle, bei dem die Kanten bei Schattenwurf weniger stark ausgeprägt sind als beim hellen Auto.

**Szene 5** Nachfolgende Szene stellt ein typisches Problem beim Tracken in Aufnahmen von Autobahnbrücken dar, bei dem ein sich bewegendes Auto von einem anderen über einen gewissen Zeitraum teilverdeckt wird. Ein Tracking wird hier erschwert, da Teile des Objekts zeitweise nicht mehr sichtbar sind und sich das Modell des Objekts dabei in der Bildebene verändert. In dieser Szene wird der untere Teil eines grünen Autos durch einen Spurwechsel von einem nachfolgenden dunklen Auto verdeckt. Anhand dieser Sequenz soll der Vorteil der Unterteilung in mehrere Blöcke verdeutlicht werden.

Bei Verwendung von Farbhistogrammen mit einem Block können keine robusten Ergebnisse er-

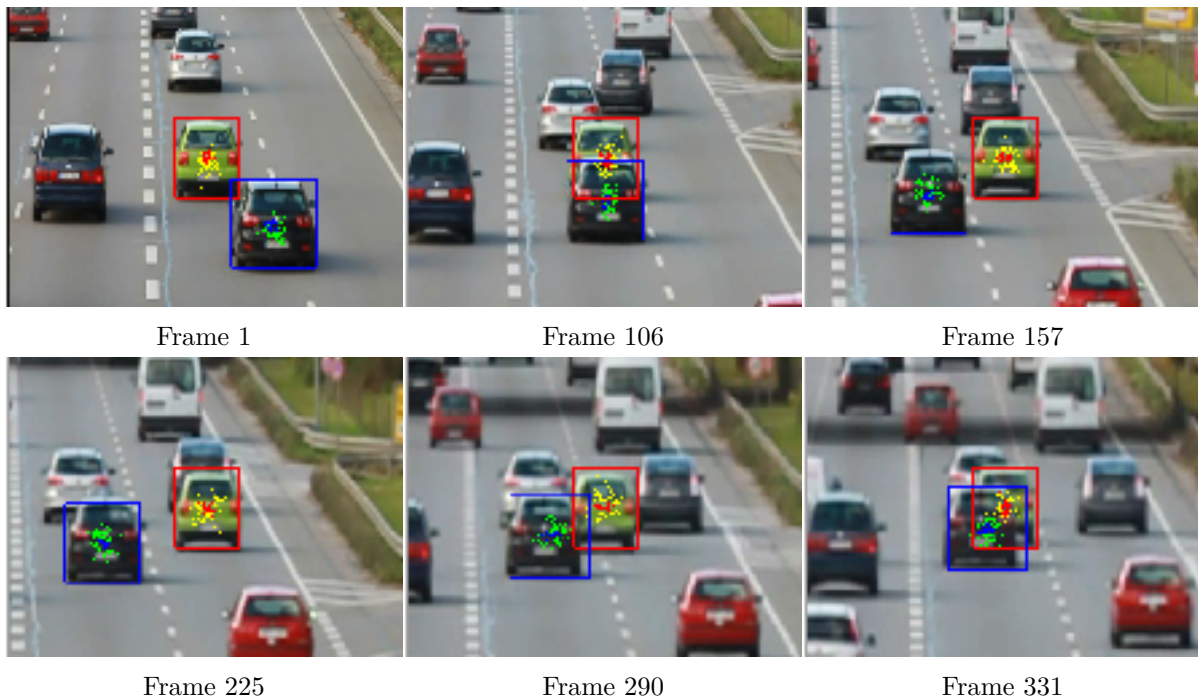




**Abbildung 6.17:** Trackingergebnisse für Szene 5 (Frame 87) ohne Berücksichtigung der Verdeckung für unterschiedliche Merkmalswahl. Oben: Schätzung der Objektskalierung mit dem Partikel Filter. Unten: Externe Vorgabe der Objektskalierung.

zielt werden, da sich das Modell durch die Verdeckung zu stark ändert. Als Folge kann oft eine zu klein geschätzte Objektskalierung beobachtet werden (siehe Ausschnitt aus der Szene in Abbildung (6.17) links oben). Bei der Verwendung von vier Blöcken (Mitte oben) kann das Objekt während der Verdeckung länger verfolgt werden, da zunächst drei Regionen von der Verdeckung nicht tangiert werden. Jedoch wird im Laufe der Verdeckung der komplette untere Bereich von der Verdeckung betroffen. Hier kommt es oft zu einem Drift der Objektskalierung der immer größer wird und schließlich beide Autos umfasst. So wird die höchste Histogrammähnlichkeit durch den dunklen Bereich des Unterbodens des Referenzmodells durch das nachfolgende dunkle Auto abgedeckt und der hauptsächlich grüne Bereich des korrekten Autos durch die oberen Regionen. Erscheint das grüne Auto wieder vollständig in der Szene, kann es zu einem Sprung auf das richtige Auto kommen. In anderen Simulationen wurde aber auch teilweise ein kompletter Verlust und das Weiterverfolgen des falschen Autos beobachtet. Ein robustes Tracken kann also trotz Mutli-Part Regionen nicht erzielt werden. Zudem wird bei zuvor viel zu großer Objektskalierung eine Rückkehr auf das korrekte Auto erschwert.

Die Verwendung von Kantenmerkmalen liefern bei Verdeckungen sehr schlechte Ergebnisse (oben rechts). Das Auto kann zwar zunächst sehr lange trotz Verdeckung verfolgt werden; werden jedoch zu viele Teile verdeckt, kommt es zu einem Sprung auf das nachfolgende Auto, da sich die Kantenmerkmale unterschiedlicher Autos aus dieser Perspektive sehr ähneln. Da zudem keine Unterscheidung zwischen der Farbe der Objekte getroffen wird, bleibt der Tracker während der Szene auf dem falschen Auto. Ob das korrekte Auto wiedergefunden werden kann hängt zudem

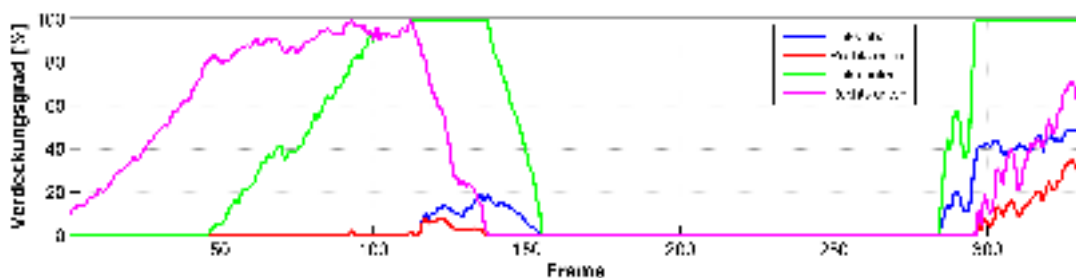


**Abbildung 6.18:** Trackingergebnisse für Szene 5 mit Berücksichtigung der Verdeckung durch zusätzliches Tracken des zweiten Objekts bei Verwendung von Farbmerkmalen mit 4 Blöcken.

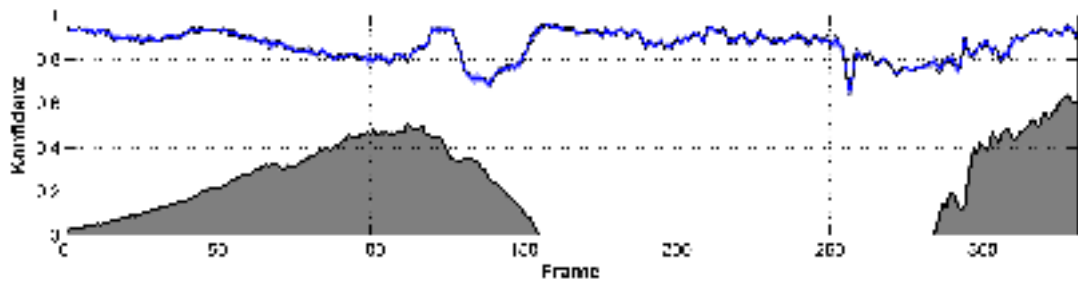
vom gewählten Prozessrauschen ab (vgl. Kapitel 6.1.2).

Selbst die Berücksichtigung einer extern vorgegebenen Objektskalierung kann kein robustes Tracking ermöglichen (untere Zeile). Ohne Berücksichtigung einer Verdeckung durch ein weiteres Auto kann ein robustes Tracken in der Regel nicht gewährleistet werden.

Um bessere Ergebnisse mit Teilverdeckungen zu erreichen, müssen diese also explizit behandelt werden. Im Sinne eines Trackingsystems sollen möglichst alle bewegten Objekte verfolgt werden. Somit steht zu jedem Zeitpunkt für jedes Objekt dessen Position, sowie Ausmaße zur Verfügung, mit deren Hilfe eine Verdeckung zweier oder mehrerer Objekte berechnet werden kann. Ist die Neigungsrichtung der Kamera und die Fahrtrichtung der Objekte bekannt, kann anhand der Positionen ebenfalls bestimmt werden, welches der Objekte verdeckt wird. In Szene



**Abbildung 6.19:** Verdeckungsgrad aller Blöcke für das Trackingergebnis aus Abbildung (6.18).



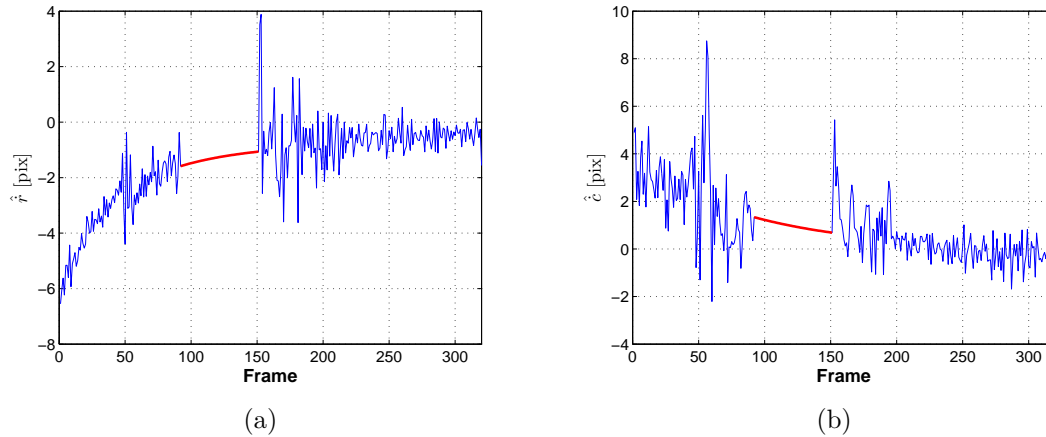
**Abbildung 6.20:** Genauigkeit und Standardabweichung für Szene 5 bei Verwendung von Farbmerkmalen mit 4 Blöcken und Berücksichtigung der Verdeckung für 20 Simulationen. (grau) Gesamtverdeckungsgrad.

5 kann das verdeckte Fahrzeug anhand der geschätzten geringeren Bildzeile bestimmt werden, welches auf das grüne Auto zutrifft. Durch die Verwendung der Mehrfachregionen kann somit für jeden Zeitpunkt ein Verdeckungsgrad aller Regionen berechnet werden, welcher zugleich wie in Kapitel 5.5.2 dargestellt, als Gewichtungsfaktor für die jeweilige Region gilt.

In Abbildung (6.18) und Abbildung (6.19) werden Ausschnitte aus der Szene und Verdeckungsgrade der einzelnen Regionen entlang der Szene dargestellt. Zunächst wird die untere rechte Region verdeckt. Mit Zunahme dieser Verdeckung erfolgt eine immer geringere Gewichtung dieser Region. Ab hier werden vermehrt die oberen Regionen des Objekts verfolgt, die ebenfalls unter geringen Verdeckungen leiden. Erscheint das Auto wieder (Frame 157), können alle Regionen für das Tracking verwendet werden, bis es schließlich bei einem weiteren Spurwechsel erneut von dem selben Objekt verdeckt wird und somit wieder eine Gewichtung der Regionen erfolgt. Abbildung (6.18) zeigt die Ergebnisse für die Verwendung dieser Trackingstrategie. Neben den Templates des gesuchten Autos werden zusätzlich die Trackingergebnisse des zweiten Autos dargestellt, welches mit Hilfe von Mehrfachregionen und Farbe als Bildmerkmal erfolgreich verfolgt werden kann.

Ein Vergleich mit den manuell gelabelten Ergebnissen kann Abbildung (6.20) entnommen werden. Zusätzlich wird hier der Gesamtverdeckungsgrad dargestellt (grau). Es lässt sich feststellen, dass sich das Objekt entlang der gesamten Sequenz sehr genau verfolgen lässt, die Standardabweichung bei 20 Simulationen sehr gering ist und sich damit eine hohe Wiederholbarkeit erzielen lässt.

**Szene 6** Eine gegenseitige Verdeckung mehrerer Objekte kann nicht nur, wie in Szene 5 dargestellt partial, sondern auch für einen gewissen Zeitraum vollständig erfolgen. Da das Objekt zu diesem Zeitpunkt nicht sichtbar ist, kann ähnlich der Szene 2 kein robustes Tracking erfolgen. Die Gefahr, dass der Tracker während der Verdeckung von einem anderen Objekt abgelenkt wird ist groß und damit die Chance geringer das richtige Objekt bei Wiedererscheinen zu finden. Somit sollte die Bewegung während der Verdeckung vorhergesagt werden und bei theoretischem Erscheinen wieder mit der Suche begonnen werden. In den Untersuchungen stellte sich eine Prädiktion ab einem Gesamtverdeckungsgrad von 75% als gute Wahl heraus. Es erfolgt das gleiche



**Abbildung 6.21:** Geschätzte Geschwindigkeiten während dem Tracking in Szene 6 entlang der (a) Bildzeile und (b) Bildspalte. (rot) Prädiktionsparameter während der Vollverdeckung.

Prinzip wie bei der Prädiktion des Objekts in Szene 2, mit dem Unterschied, dass die Bewegung während der Verdeckung aufgrund der Kameraneigung nicht mehr linear in Richtung der Bildzeile ist und das Objekt eine Skalierungsänderung vornimmt. Wird dies nicht berücksichtigt und eine lineare Fortsetzung der Zustandsparameter gewählt, erfolgt die Bewegung in Richtung der Bildzeile zu schnell. Wird die Größenänderung des Objekts aufgrund der Bildgeometrie nicht berücksichtigt, erscheint dieses schließlich zu groß. Ein Wiederfinden des Objekts wird erschwert und die Gefahr einer Fehldetektion stark erhöht.

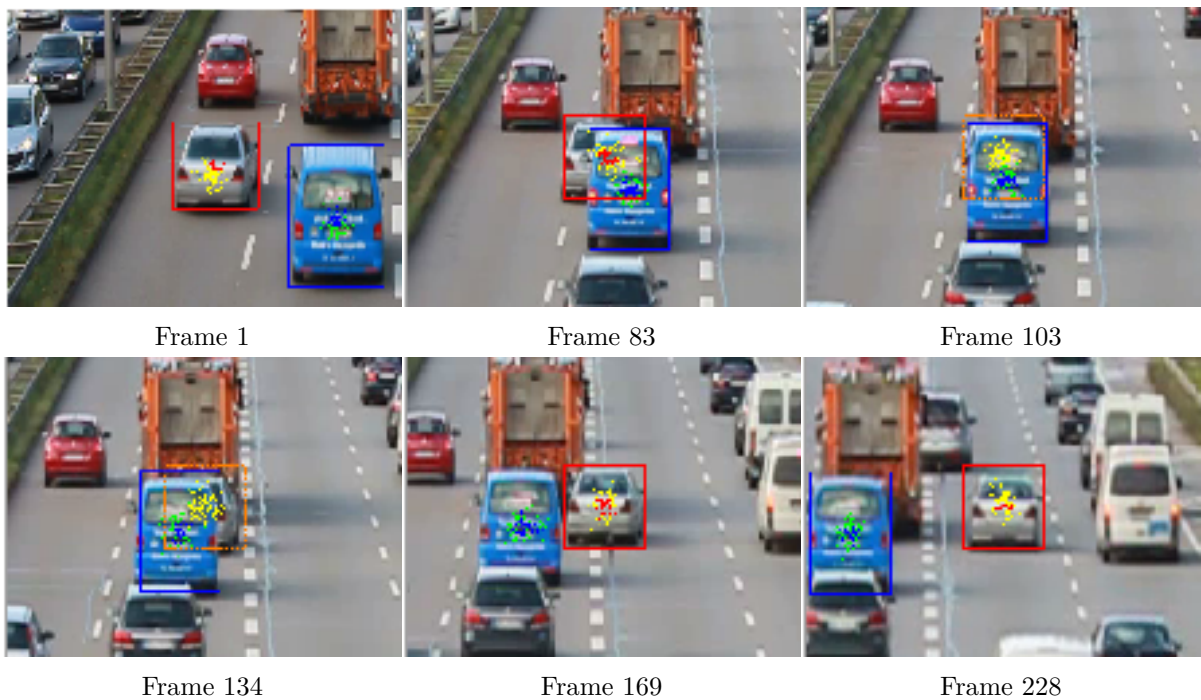
Eine robustere Bewegungsvorhersage kann erfolgen, indem die Kameraneigung miteinbezogen wird. Für die einzelnen Parameter heißt dies:

- **Geschwindigkeit entlang der Bildspalte** kann als linear angenommen werden. Aufgrund stabilen Trackings bis zu einem großen Verdeckungsgrad kann die Bewegung bei einem Spurwechsel gut präzisiert und als konstant angenommen werden.
- **Geschwindigkeit entlang der Bildzeile** ist abhängig von der Neigung der Bildebene zur Straßenebene. Ähnlich der Schätzung der Objektskalierung nimmt eine konstante Geschwindigkeit bei Abnahme der Bildzeile quadratisch ab.
- **Objektskalierung** ist ebenfalls abhängig von der Kameraneigung. Im besten Fall kann sie, wie in Szene 3 und 5, extern vorgegeben werden. Ist dies nicht möglich, sollte sie ebenfalls quadratisch fortgesetzt werden.

Abbildung (6.21) stellt die Schätzungen der Geschwindigkeiten für Bildzeile und Bildspalte entlang der Szene dar. Die Prädiktion der Geschwindigkeit während der Verdeckung (rot) wird anhand der bisherigen Geschwindigkeiten und den oben aufgeführten Zusammenhängen ausgeglichen.

Während der Vollverdeckung sollte auch hier ein geringeres Prozessrauschen verwendet werden, damit die Streuung der Partikel nicht zu groß wird. Ein erneutes Suchen des Objekts wird ab 50% theoretischem Verdeckungsgrad fortgesetzt. Dies erhöht zum einen die Chance das Objekt

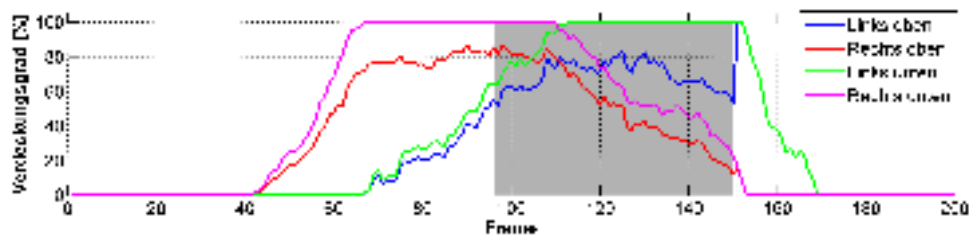




**Abbildung 6.22:** Trackingergebnisse für Szene 6 bei Verwendung von Farbmerkmalen mit 4 Blöcken und Berücksichtigung der Verdeckung. (orange) Prädizierte Objektposition während Vollverdeckung.

wiederzufinden falls die Prädiktion ungenau erfolgt, und zum anderen verringert es die Gefahr wenige Frames nach Erscheinen wieder in Vollverdeckung zu geraten.

Abbildung (6.22) stellt Ausschnitte der Szene und die Trackingergebnisse des verdeckten Autos (rot), sowie des sichtbaren Autos (blau) dar. Ein oranges Rechteck deutet die Vollverdeckung des Trackers an, dessen Bewegung nur aufgrund der Vorhersage beruht. Zusätzlich wird in Abbildung (6.23) der Verdeckungsgrad aller Regionen, sowie der Bereich der Vollverdeckung (grau) dargestellt. Bis zum Frame 96 erfolgt das Tracking mit Hilfe der gewichteten Mehrfachregionen. Anschließend wird die Objektposition zum Frame 151 prädiziert. Ab hier wird das Objekt erneut gesucht. Aufgrund der Restverdeckung im linken Bereich wird daher vorwiegend der rechte Teil des Autos gesucht. Nach wenigen Frames kann das Objekt wiedergefunden und weiterhin präzise



**Abbildung 6.23:** Verdeckungsgrad aller Blöcke für das Trackingergebnis aus Abbildung (6.22). (grau) Bereich der Vollverdeckung.

verfolgt werden.

### 6.2.3 Kreuzungsaufnahmen

Häufig werden Überwachungskameras an hochgelegenen Stellen von Straßenkreuzungen montiert um eine bestmögliche Übersicht der Szene zu erhalten. Charakteristisch ist dabei oft die nahezu vertikale Aufnahmeperspektive, woraus sich Rotationen der Objekte ergeben die soeben abbiegen. Mit Hilfe des in den anderen Szenen verwendeten CV-Modells ist es nun nicht mehr möglich die Objekte exakt zu verfolgen, da diese Rotation nicht berücksichtigt wird. Durch Verwenden des CVR-Modells kann diese Rotation nun zusätzlich geschätzt und damit ein robusteres Tracking ermöglicht werden. Zusätzlich wird an dieser Szene ein Tracking von Fußgängern und Fahrradfahrern untersucht, die aus dieser Perspektive nur noch eine sehr geringe Größe im Bild einnehmen und damit wenig Information beinhalten.

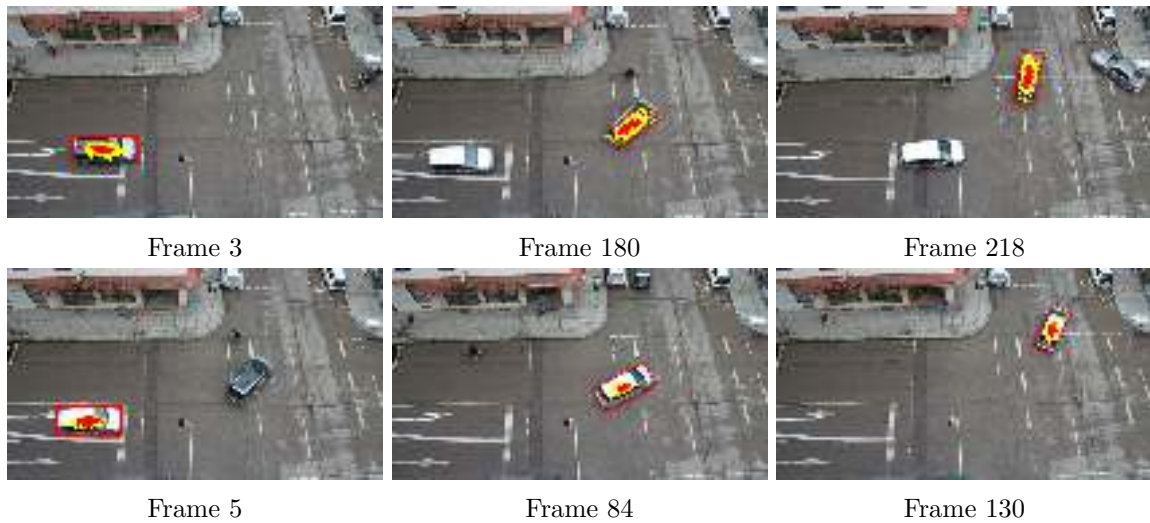
**Szene 7** Zunächst soll ein weißes und ein dunkelgraues Auto verfolgt werden. Als Bildmerkmale wurden Farbhistogramme mit einem Block gewählt. Kantenrichtungshistogramme können bei Rotationen nicht mehr ohne weiteres verwendet werden, da sich durch die Drehung des Objekts ebenfalls die Richtungen der Gradienten ändern. Sollten dennoch Kanten verwendet werden, müssen die berechneten Gradientenorientierungen um den Drehwinkel reduziert werden, bevor sie mit dem Referenzmodell verglichen werden.

Abbildung (6.24) zeigt Ausschnitte aus der Szene für beide Autos. Die Drehungen des Objekts wurden in beiden Fällen ausreichend gut geschätzt und die Objekte können bis zum Ende der Szene verfolgt werden. Jedoch sollten mehrere Partikel für die Schätzung mit einem CVR-Modell verwendet werden (hier: 300 Partikel), um eine zusätzliche Rotation robust schätzen zu können.

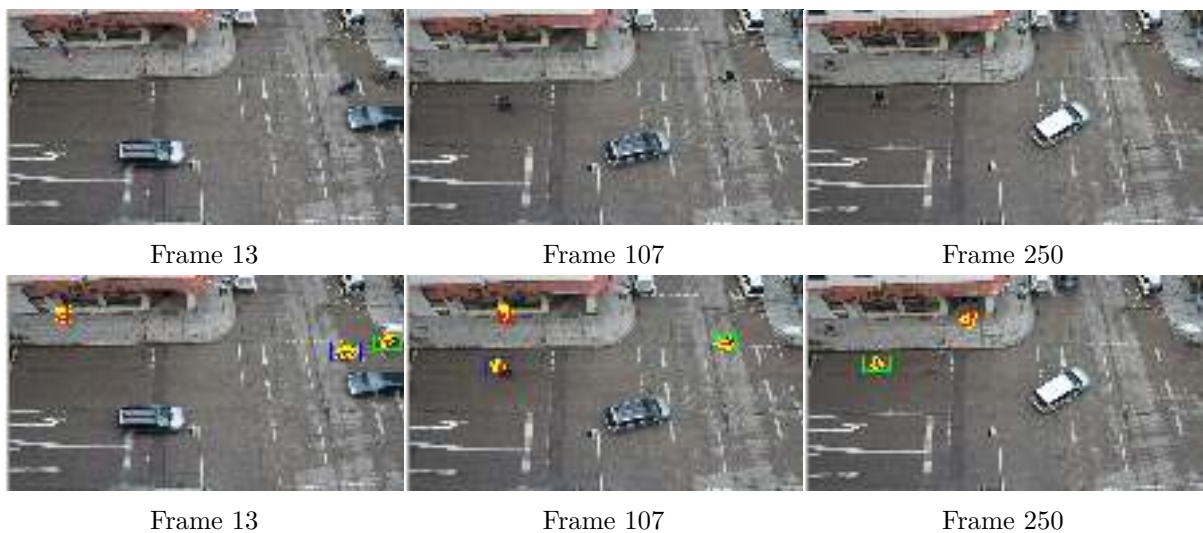
Schließlich wird versucht, die in der Szene sichtbaren Passanten zu tracken. Hierfür kann wieder das CV-Modell verwendet werden. Aufgrund der vielen Hintergrundinformation, die in das Modell einfließt, wird eine Gewichtungsfunktion eines Epanechnikov Kernels eingeführt. Die Schätzung einer Objektskalierung kann hier aufgrund der nahezu vertikalen Perspektive vernachlässigt werden und führt sogar zu schlechten Ergebnissen. Abbildung (6.25) stellt in der oberen Reihe Ausschnitte der Rohsequenz dar, in der unteren Bildzeile die getrackten Ergebnisse von zwei Fahrradfahrern und einem Fußgänger. Alle Objekte können erfolgreich verfolgt werden. Jedoch ist die Verwendung eines Partikel Filters für die sehr langsamen Bewegungen eines Fußgängers aufgrund der relative hohen Laufzeit fragwürdig und gleichwertige Ergebnisse könnten auch mit anderen Verfahren (Mean-Shift, Kalman-Filter) erzielt werden.

## 6.3 Einfluss externer Faktoren

Häufig kann aufgrund von begrenzten Speicherkapazitäten nicht auf hochauflösende, echtzeitfähige Videos zurückgegriffen werden. Hierfür wurde untersucht welchen Einfluss Bildauflösung



**Abbildung 6.24:** Trackingergebnisse für Szene 7 unter Verwendung des CVR-Modells und Farbmerkmalen mit 1 Block. Oben: Ergebnisse für graues Auto, unten: Ergebnisse für weißes Auto.



**Abbildung 6.25:** Trackingergebnisse für Szene 7 beim Verfolgen von Passanten. Oben: Bildausschnitte, unten: Trackingergebnisse.

sowie die Bildfrequenz auf die Trackingergebnisse haben.

### 6.3.1 Einfluss der Bildauflösung

Die Bildauflösung entscheidet über die Größe des Objekts und darüber wie viel Information über das Objekt zur Verfügung steht. Anhand Szene 3 wurde untersucht welchen Einfluss eine verringerte Auflösung auf das Ergebnis hat. In der Originalszene hat das Objekt bei Initialisierung die Größe von 120x120 Pixel bei einer Bildauflösung von 1280x720 Pixel. Am Ende der Szene hat das Objekt nur noch die Größe von ca. 50x50 Pixel.

Es wurde ein Tracking mittels Farb- und Kantenrichtungshistogramme ebenfalls für eine Bildauflösung von 640x360 Pixel und 320x180 Pixel durchgeführt. Somit hat das Objekt nur noch ein Viertel bzw. ein Sechzehntel der Größe wie im Originalbild. Für eine Auflösung von 640x360 Pixel kann trotzdem mit beiden Merkmalen gut getrackt werden. Farbe verzeichnet keinerlei Abzüge der Genauigkeit. Für die Verwendung von Kantenmerkmalen sollten max. 9 Blöcke verwendet werden, damit noch ausreichend viele Pixel in den Blöcken vorhanden sind. Bei einer Auflösung von 320x180 Pixel kann zunächst auch noch robust getrackt werden, allerdings entstehen ab etwa der Hälfte der Sequenz große Diskrepanzen und ein Driften des Trackers kann festgestellt werden.

Eine Verwendung von 640x360 Pixel kann also zum Tracken für die hier vorgestellten Szenen verwendet werden und zusätzlich die Laufzeit des Verfahrens beschleunigen. Abbildung (6.6) zeigt ebenfalls die Laufzeiten für unterschiedliche Bildauflösungen. Die quadratische Abhängigkeit der Laufzeit mit der Auflösung wird auch hier ersichtlich. Die Laufzeit für das Verfolgen eines Objektes mit der Initialisierungsgröße von 60x60 Pixel nimmt für einen Frame für Verwendung der Farbmerkmale etwa 12 ms und für Kantenmerkmale etwa 17 ms in Anspruch.

### 6.3.2 Einfluss der Bildfrequenz

Bisher wurden Szenen mit einer Bildfrequenz von 30 Hz verwendet. Wird nicht die volle Bildfrequenz für das Tracking verwendet oder enthält das Video eine geringere Bildfrequenz, erscheinen schnellere und abruptere Bewegungen. Eine Schätzung der Objektgeschwindigkeit durch das Bewegungsmodell wird hier unerlässlich. Besonders bei der Initialisierung sollten für die Geschwindigkeiten gute Näherungswerte verwendet werden damit der Tracker das Objekt nicht direkt zu Beginn verliert. Für Szene 3 konnte dennoch das Objekt ohne Vorgabe einer Näherungsgeschwindigkeit bis zu 3 Hz verfolgt werden ohne es zu verlieren. Es dauert allerdings einige Frames bis die Geschwindigkeit präzise geschätzt werden kann.

Wird eine Auflösung von 640x360 Pixel verwendet und lediglich jedes dritte Frame (wobei keinerlei Unterschiede für die Ergebnisse im Vergleich zu jedem Frame festgestellt werden konnte) verwendet, können zeitgleich ca. 10 Fahrzeuge der Größe 60x60 Pixel bei Verwendung der Farbmerkmale und 7 Fahrzeuge bei Verwendung der Kantenmerkmale verfolgt werden.

---

## Anwendung eines automatischen Verkehrsüberwachungssystems

---

Das in dieser Arbeit entwickelte Trackingverfahren kann nun dazu verwendet werden, um aus einer Verkehrssequenz automatisch Statistiken über den Verkehrsfluss der Szene zu extrahieren.

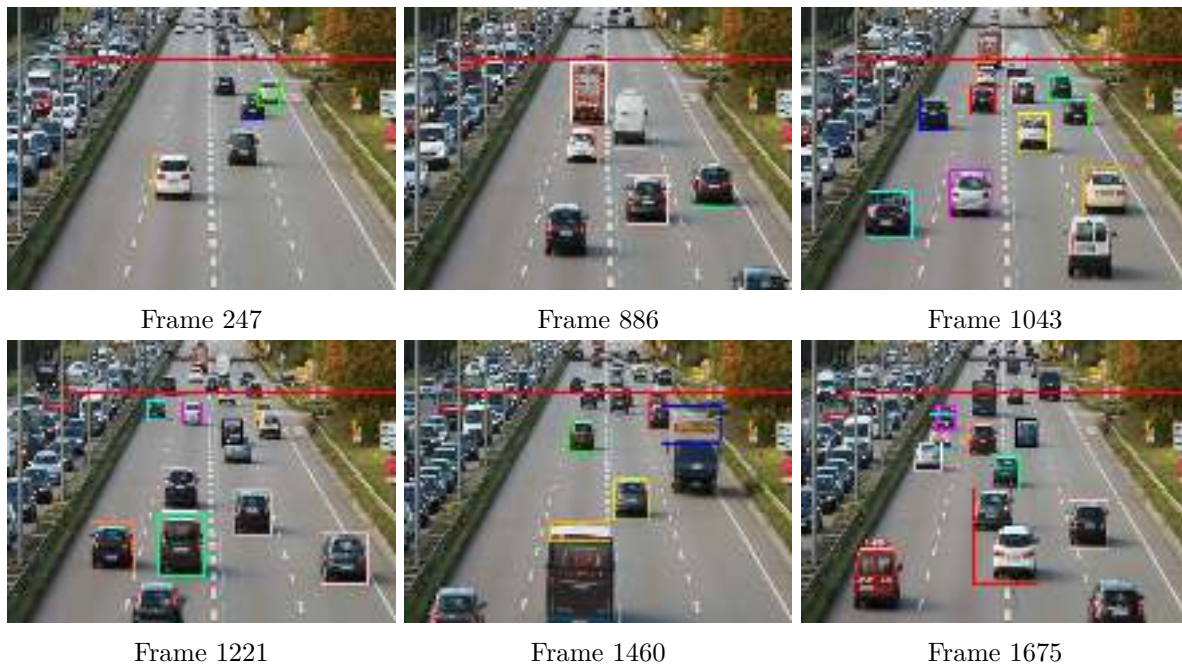
### 7.1 Ziele

Ziel soll es dabei sein, ein vollautomatisches Verkehrsüberwachungssystem zu entwickeln, das in der Lage sein soll ohne manuelle Eingriffe Statistiken über den Verkehrsfluss einer Szene zu extrahieren, die für anschließende Verkehrsanalysen verwendet werden können. Hierfür wird das in dieser Arbeit entwickelte Trackingverfahren mit einem Objektdetektor kombiniert, der neu in der Szene erscheinende Objekte detektiert und diese automatisch für den Tracker initialisiert. Die Szene wird dabei in eine Detektions- und Trackingzone unterteilt (vgl. Abbildung



**Abbildung 7.1:** Hintergrundbild der Szene für das automatische Verkehrsüberwachungssystem unterteilt in Detektionszone (grün) und Trackingzone (rot).





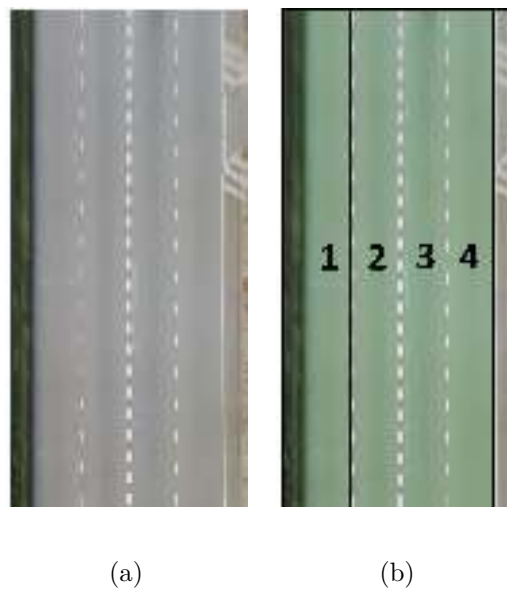
**Abbildung 7.2:** Ausschnitte aus dem automatischem Trackingsystem

(7.1)). Zusätzlich wird ein metrisches Maß eingeführt, damit verkehrsrelevante Analysen über die Geschwindigkeit der Objekte durchgeführt werden können. Dies ermöglicht schließlich die Extraktion mehrerer Verkehrsstatistiken, darunter Fahrzeuganzahl, Spurdichte, Spurwechsel, reale Geschwindigkeiten und liefert Trajektorien der einzelnen Fahrzeuge.

## 7.2 Detektion

Die Detektion neu erscheinender Objekte kann durch die Verwendung einer stationären Kamera anhand eines Hintergrundschätzers erfolgen. Hierfür wird während der Szene in einer vorab definierten Detektionszone (Abbildung 7.1 (grün)) ein dynamisches Hintergrundbild erzeugt und Veränderungen in diesem Bild durch erscheinende Objekte als Vordergrund klassifiziert. Mit Hilfe von morphologischen Operatoren können diese Vordergrundpixel in zusammengehörige Regionen gruppiert werden und anschließend mit einer Objektform (hier: Rechteck) repräsentiert werden. Nähere Informationen des hier verwendeten *ViBe* Detektionsalgorithmus finden sich in [4].

Erreicht das detektierte Objekt die Trackingzone (Abbildung 7.1 (rot)), wird ein Partikel Filter initialisiert. Konkret bedeutet dies, dass für diesen Bildausschnitt die Bildmerkmale berechnet werden, das Bewegungsmodell an dieser Position aufgestellt wird und die Partikel für dieses Objekt generiert werden. Die Initialisierung der Geschwindigkeit erfolgt durch die zusätzliche Berechnung des optischen Flusses innerhalb der Detektionszone. Dies ermöglicht, dass selbst schnelle Objekte nach der Initialisierung robust verfolgt werden können.



**Abbildung 7.3:** Darstellung der entzerrten Trackingzone aus Abbildung (7.1). (a) Entzerrte Trackingzone und (b) Definition der Fahrspuren.

### 7.3 Tracking

Nach der Initialisierung eines Objektes wird dieses mit Hilfe des Partikel Filters bis zum Ende der Trackingzone verfolgt. Es werden Kantenrichtungshistogramme mit neun Blöcken als Bildmerkmal verwendet und die Skalierung extern über die Bildgeometrie ermittelt. Für jedes Objekt wird ein individuelles Partikel Filter initialisiert.

Abbildung (7.2) zeigt einige Ausschnitte aus der Trackingsequenz für unterschiedliche Zeitpunkte. Während die meisten Fahrzeuge sehr genau über die gesamte Sequenz verfolgt, sowie robuste Ergebnisse bei Teilverdeckungen (Frame 1221 rechter oberer Bildausschnitt) erzielt werden können, gibt es noch mögliche Fehlerquellen des Systems. Zwar werden die allermeisten Objekte korrekt detektiert, es kann jedoch bei sich verdeckenden Objekten während der Detektion keine Trennung zwischen den Objekten erfolgen; somit werden sie als ein einziges Objekt initialisiert (siehe Frame 1675 unten). Bei größeren, farblich inhomogenen Objekten (Frame 1460 LKW im unteren und rechten Bildausschnitt) fällt ebenfalls eine korrekte Detektion schwer.

Aus schlecht initialisierten Objekten kann demnach kein robustes Tracking erfolgen. Allerdings können Fehldetektionen und daraus schlecht getrackte Objekte in der Regel innerhalb der anschließenden Analyse identifiziert und eliminiert werden.

### 7.4 Ergebnisanalyse

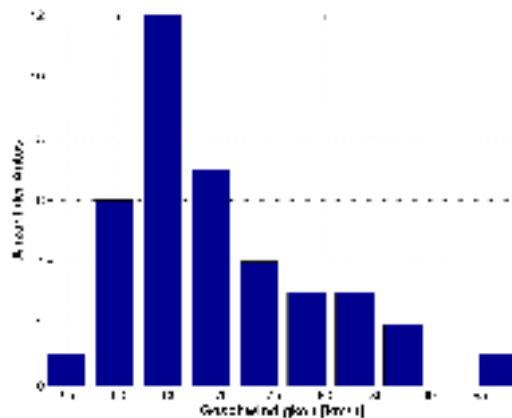
Um reale Geschwindigkeiten zu schätzen muss zusätzlich ein metrisches Maß eingeführt werden. Hierzu wird die Trackingzone mit Hilfe einer *direkten linearen Transformation* (DLT) rektifiziert, um die Neigung der Bildebene zur Straßenebene zu eliminieren. Nähere Informationen über die DLT-Methode finden sich u.a. in [24].



**Abbildung 7.4:** Trajektorien der getrackten Fahrzeuge im entzerrten Bild

Abbildung (7.3a) stellt die entzerrte Trackingzone dar, in der die Straße nun parallel erscheint und Fahrstreifen den gleichen Abstand zueinander haben. Durch die Einführung eines metrischen Maßes für beide Achsen (hier: Abstand zweier Mittelstreifen: 6m, Straßenbreite: 13m ) können nun Messungen im entzerrten Bild durchgeführt werden. Ebenfalls wird die Einteilung der einzelnen Fahrspuren im entzerrten Bild vereinfacht (siehe Abbildung (7.3b)), die für eine anschließende Analyse verwendet werden können.

Während dem Tracking wird der mittlere, bodennächste Punkt des Objekts mit Hilfe der berechneten Homographie in das entzerrte Bild transformiert. Damit können die einzelnen Trajektorien der Fahrzeuge übersichtlich dargestellt werden (siehe Abbildung (7.4)) und eine Zuweisung jedes



**Abbildung 7.5:** Histogramm der absoluten Geschwindigkeiten aller Fahrzeuge in der Szene.



Objektes zu einer Fahrspur erfolgen.

Anhand der bekannten Abtastfrequenz der Kamera (hier: 30 Hz) und der Positionsänderungen des Objekte im rektifizierten Bild können zusätzlich reale Geschwindigkeiten ermittelt werden.

Folgende Tabelle enthält einige Statistiken über den Verkehrsfluss der beobachteten Szene.

<b>Parameter</b>	<b>Spur 1</b>	<b>Spur 2</b>	<b>Spur 3</b>	<b>Spur 4</b>
Anzahl Vehikel	10	14	14	10
Spurwechsel in	0	1	2	1
Spurwechsel aus	1	2	1	0
Mittlere Geschwindigkeit [ $km/h$ ]	73.02	73.79	67.52	64.28
Maximale Geschwindigkeit [ $km/h$ ]	86.46	99.20	86.78	77.09
Minimale Geschwindigkeit [ $km/h$ ]	64.28	59.01	60.10	52.60

Neben der Überwachung der Geschwindigkeitsbegrenzungen (hier: 60  $km/h$ ) können ebenso Geschwindigkeitsverteilungen aller Fahrzeuge ermittelt werden (siehe Abbildung (7.5)). Daraus kann ein Überblick des aktuellen Verkehrsflusses abgeleitet werden.



#### 8.1 Zusammenfassung

Im Rahmen der vorliegenden Arbeit wurde ein Partikel Filter für die Objektverfolgung in Verkehrsaufnahmen entwickelt. Verschiedene Bewegungsmodelle und histogrammbasierte Bildmerkmale von Farbe und Kanten wurden für unterschiedliche Szenen getestet und miteinander verglichen. Durch die Unterteilung von Objektmodellen in mehrere Blöcke können wesentlich robustere Ergebnisse erzielt werden, die besonders bei Veränderung der Objektgröße von Vorteil sind. Größenänderungen der Objekte können entweder innerhalb des Partikel Filters geschätzt werden oder bei ebener Fahrbahn extern berechnet werden. Letzteres liefert in der Regel robustere Ergebnisse, kann jedoch nicht in jeder Szene verwendet werden.

Während Farbe als Bildmerkmal in den meisten Fällen, besonders in Kombination mit Mehrfachregionen, sehr gute Ergebnisse erzielt, muss bei dynamischen Lichtverhältnissen auf Kantenrichtungshistogramme zurückgegriffen werden.

Großer Bestandteil der Untersuchungen lag in der Behandlungen von Verdeckungen. Während kurzzeitige Teil- oder Komplettverdeckungen meist ohne Probleme anhand des Multihypothesen Trackings des Partikel Filters und durch die Unterteilung des Objektmodells in mehrere Regionen gemeistert werden können, muss bei längerer Verdeckung auf eine Vorhersage der Objektposition zurückgegriffen werden. Eine Parameterschätzung der Prädiktionsparameter anhand von Ausgleichungen der einzelnen Parameter in Bezug auf die Aufnahmegeometrie muss für präzise Vorhersagen verwendet werden. Beim simultanen Tracking mehrerer Objekte, die sich gegenseitig verdecken, können verdeckte Objektregionen durch eine dynamische Gewichtung vom Trackingprozess ausgeschlossen werden und das Tracking auf Basis der unverdeckten Objektregionen fortgesetzt werden.

## 8.2 Ausblick

Selbst unter verschiedenen Aufnahmebedingungen erweist sich das hier vorgestellte Trackingsystem bei richtiger Parameterwahl als robust. Solange verdeckte Objekte eine konstante Geschwindigkeit und Richtung beibehalten, kann das verdeckte Objekt in der Regel wiedergefunden werden. Probleme ergeben sich allerdings, wenn sich die Geschwindigkeit und/oder Richtung während der Verdeckung ändert oder die Verdeckung zu lange anhält. Ein Wiederfinden des Objekts wird in diesen Situationen erschwert und kann nicht garantiert werden.

Zudem ergeben sich häufig Probleme bei Teilverdeckungen unter Verwendung von Kanten als Bildmerkmale, sodass oft das falsche Objekt verfolgt wird, da sich die Kanten unterschiedlicher Fahrzeuge im Merkmalsraum sehr ähneln. Im Partikel Filter können dabei kurzzeitig zwei stark ausgeprägte Objekthypothesen für jeweils ein Fahrzeug ersichtlich werden. Eine verbesserte Auswertung und Berücksichtigung dieser Multimodalitäten stellen noch zu lösende Probleme dar. Clustering-Methoden könnten Aufschluss über die Anzahl der Objekthypothesen geben, sodass jede Hypothese solange getrennt weiterverfolgt werden könnte, bis die Störung des Objekts vorüber ist. Ein Ansatz hierfür wird beispielsweise in [13] vorgestellt.

Die hier verwendeten Bildmerkmale eignen sich für Verkehrsaufnahmen bei Tag, liefern jedoch schlechte Ergebnisse bei Nachtaufnahmen. Hier gibt es weder ausprägende Farbmerkmale der Objekte, noch starke Konturen und durch die Beleuchtung anderer Fahrzeuge ebenfalls stark dynamische Lichtverhältnisse. Chan et al. liefern in [12] einen Ansatz für das Verfolgen von Verkehrsobjekten bei Nachtaufnahmen durch Verwendung weiterer Merkmale, die sich u.a. auf das Tracken markanter Punkte in Form von Scheinwerfern und Rücklichtern konzentrieren.

Eine Fusion beider hier verwendeter Merkmale kann in den meisten Fällen keine sichtbare Verbesserung der Ergebnisse erzielen. Zudem erhöht sich die Laufzeit des Trackers und eine Echtzeitfähigkeit kann nicht mehr gewährleistet werden. Eine Kombination beider Merkmale kann jedoch zu einer Verbesserung des Systems führen, wenn diese je nach Situation dynamisch gewichtet werden. Ein robuster Ansatz zur dynamischen Gewichtung, der auch global für unterschiedliche Szenen verwendet werden kann, konnte in dieser Arbeit nicht erarbeitet werden. Unterschiedliche Methoden zur Fusion mehrerer Bildmerkmale in Partikel Filter können in [40], [7] und [47] gefunden werden.

Die Laufzeit des Trackingverfahrens ist stark abhängig von den Ausgangsbedingungen (Anzahl der Farbkanäle, Bildgröße, Objektgröße, Merkmalswahl, Parameterwahl), kann aber Echtzeitfähigkeit für die hier dargestellten Szenen selbst in hoher Auflösung gewährleisten. Im Falle von Multi-Objektverfolgung kann diese aber abhängig von der Anzahl der zu verfolgenden Objekte nicht mehr garantiert werden. Die Merkmalsextraktion bedarf der meisten Rechenzeit, die linear mit der Partikelanzahl zunimmt. Eine Reduzierung der Partikelanzahl wird angestrebt und könnte beispielsweise durch Verwendung eines Hintergrundschätzers erreicht werden. Wenn es möglich ist, zu jedem Zeitpunkt ein robustes Vordergrundbild zu erstellen, könnte ein Tracking auch nur in diesem erfolgen. Ein Verlust des Objekts kann dadurch ausgeschlossen werden und die Partikelanzahl eventuell reduziert werden. Eine zusätzliche Beschleunigung der Laufzeit könnte bei der Berechnung der Farbhistogramme erzielt werden. Da es sich bei den Fahrzeugen in der

Regel um farblich homogene Objekte handelt, könnte die Verwendung jedes zweiten oder vierten Pixels innerhalb des Modells für die Histogrammerstellung ausreichen. Dies führt zur Reduktion der Rechenzeit, speziell wenn es sich um sehr große Objekte, wie beispielsweise Busse oder LKW's handelt. In [41] wurden Untersuchungen hierzu gestartet.

Weiterhin kann versucht werden, sich von rechteckigen Templates zu lösen und die wahre Objektform durch Segmentierung zu berücksichtigen. Dies verhindert die Einbeziehung von Hintergrundinformation in das Objektmodell. Ebenso könnte mit einem *Condensation Algorithmus* ein Trackingverfahren in Bezug auf die Kontur des Objekts durchgeführt werden [26]. Das Tracking von Objektkonturen könnte bei Verdeckungen weniger sensitiv als die Verwendung von Kantenrichtungshistogrammen sein.



---

## Abbildungsverzeichnis

---

2.1	Unterschiedliche Trackingverfahren für Verkehrsüberwachungsanwendungen . . .	6
2.2	Vergleich regionenbasierter und merkmalsbasierter Trackingverfahren bei Verdeckungen . . . . .	7
2.3	Verschiedene Darstellungsmöglichkeiten von Objektformen . . . . .	9
2.4	Unterteilung der Trackingverfahren in SHL- und MHL-Ansätze . . . . .	11
3.1	Darstellung eines Hidden Markov Modells . . . . .	17
3.2	Prinzip des Importance Samplings . . . . .	23
3.3	Ablauf eines Iterationsschrittes eines SIR-Filters . . . . .	27
4.1	Merkmalsextraktion mittels Farbhistogrammen . . . . .	30
4.2	Gewichtungsfunktion für Histogrammberechnung mittels Epanechnikov-Kernels .	31
4.3	Farbhistogramme für Mehrfachregionen . . . . .	32
4.4	Merkmalsextraktion mittels Kantenrichtungshistogrammen . . . . .	33
4.5	Faltung einer Bildregion mit Sobeloperatoren . . . . .	34
4.6	Gradientenamplituden- und orientierungsbild . . . . .	35
4.7	Einteilung der Gradientenorientierungen . . . . .	35
4.8	Kantenrichtungshistogramm für 1 Block . . . . .	36
4.9	Kantenrichtungshistogramme für unterschiedliche Blockanzahl . . . . .	36
5.1	Darstellung des Objektmodells im Zustandsraum . . . . .	40
5.2	Ablauf des Partikel Filters für Trackinganwendungen . . . . .	41
5.3	Visualisierung eines Iterationsschrittes des Partikel Filters . . . . .	42
5.4	Auswertung einer Wahrscheinlichkeitsverteilung mit den Schätzverfahren MMSE und MAP . . . . .	44
5.5	Zustandsprädiktion mit einem CV-Modell . . . . .	46
5.6	Zustandsprädiktion mit einem CVR-Modell . . . . .	47
5.7	Einfluss des Messrauschens auf die Wahrscheinlichkeitsverteilung . . . . .	50

5.8	Einfluss des Messrauschens auf das Resampling . . . . .	50
5.9	Effiziente Gradientenberechnung . . . . .	52
5.10	Auftreten unterschiedlicher Verdeckungsarten in Verkehrsaufnahmen . . . . .	53
5.11	Ablauf des Partikel Filters unter Berücksichtigung von Vollverdeckungen . . . . .	54
5.12	Bestimmung des Fluchtpunktes einer Szene . . . . .	56
5.13	Berechnung der geometrischen Objektfläche . . . . .	57
6.1	Genauigkeitsmaß der Evaluierung . . . . .	59
6.2	Auswirkungen des Prozessrauschens auf die Trackinggenauigkeit . . . . .	60
6.3	Auswirkungen des Prozessrauschens auf die Geschwindigkeitsschätzung . . . . .	62
6.4	Einfluss der Bin-Anzahl auf die Robustheit des Verfahrens . . . . .	63
6.5	Einfluss der Partikelanzahl auf die Robustheit des Verfahrens . . . . .	64
6.6	Einfluss der Partikelanzahl auf die Laufzeit des Verfahrens . . . . .	65
6.7	Aufgenommene Szenen vom Straßenrand . . . . .	67
6.8	Trackingergebnisse für Szene 1 . . . . .	68
6.9	Trackinggenauigkeit für Szene 1 . . . . .	68
6.10	Trackingergebnisse für Szene 2 . . . . .	69
6.11	Aufgenommene Szenen von Autobahnbrücken . . . . .	70
6.12	Trackinggenauigkeit für Szene 3 bei Verwendung von Farbmerkmalen . . . . .	71
6.13	Probleme der Objektskalierungsschätzung bei Verwendung von Kantenmerkmalen . . . . .	71
6.14	Trackinggenauigkeit für Szene 3 bei Verwendung von Kantenmerkmalen . . . . .	72
6.15	Trackingergebnisse für Szene 4 bei Verwendung von Farbmerkmalen . . . . .	73
6.16	Trackingergebnisse für Szene 4 bei Verwendung von Kantenmerkmalen . . . . .	74
6.17	Trackingergebnisse für Szene 5 ohne Berücksichtigung der Verdeckung . . . . .	75
6.18	Trackingergebnisse für Szene 5 mit Berücksichtigung der Verdeckung . . . . .	76
6.19	Verdeckungsgrad aller Blöcke für Szene 5 . . . . .	76
6.20	Genauigkeit und Standardabweichung für Szene 5 bei Verwendung von Farbmerkmalen mit 4 Blöcken und Berücksichtigung der Verdeckung . . . . .	77
6.21	Geschätzte Geschwindigkeiten während dem Tracking in Szene 6 . . . . .	78
6.22	Trackingergebnisse für Szene 6 . . . . .	79
6.23	Verdeckungsgrad aller Blöcke für Szene 6 . . . . .	79
6.24	Trackingergebnisse für Szene 7: Fahrzeuge . . . . .	81
6.25	Trackingergebnisse für Szene 7: Passanten . . . . .	81
7.1	Hintergrundbild der Szene für das automatische Verkehrsüberwachungssystem . . . . .	83
7.2	Ausschnitte aus dem automatischem Trackingsystem . . . . .	84
7.3	Darstellung der entzerrten Trackingzone . . . . .	85
7.4	Trajektorien der getrackten Fahrzeuge im entzerrten Bild . . . . .	86
7.5	Histogramm der absoluten Geschwindigkeiten aller Fahrzeuge in der Szene . . . . .	86



---

## Literaturverzeichnis

---

- [1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 798–805, 2006.
- [2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, February 2002.
- [3] K. D. Baker and G. D. Sullivan. Performance assessment of model-based tracking. In *IEEE Workshop on Applications of Computer Vision, Proceedings*, pages 28–35, 1992.
- [4] O. Barnich and M. Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, 2011.
- [5] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3457–3464, 2011.
- [6] P. L. M. Bouttefroy, A. Bouzerdoun, S. L. Phung, and A. Beghdadi. Vehicle tracking by non-drifting mean-shift using projective kalman filter. In *IEEE Conference on Intelligent Transportation Systems*, pages 61–66. IEEE, 2008.
- [7] P. Brasnett, L. Mihaylova, D. Bull, and N. Canagarajah. Sequential Monte Carlo tracking by fusing multiple cues in video sequences. *Image and Vision Computing*, 25(8):1217–1227, 2007.
- [8] P. Brasnett, L. Mihaylova, N. Canagarajah, and D. Bull. Particle filtering with multiple cues for object tracking in video sequences. pages 430–441. Electronic Imaging International Society for Optics and Photonics, 2005.

- [9] S. Brutzer, B. Hoferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1937–1944, 2011.
- [10] N. Buch, S. A. Velastin, and J. Orwell. A review of computer vision techniques for the analysis of urban traffic. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):920–939, 2011.
- [11] A. Cavallaro, O. Steiger, and T. Ebrahimi. Tracking video objects in cluttered background. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(4):575–584, 2005.
- [12] Y.-M. Chan, S.-S. Huang, L.-C. Fu, P.-Y. Hsiao, and M.-F. Lo. Vehicle detection and tracking under various lighting conditions using a particle filter. *IET Intelligent Transport Systems*, 6(1):1–8, 2012.
- [13] C. Chang, R. Ansari, and A. Khokhar. Multiple object tracking with kernel particle filter. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 566–573, 2005.
- [14] Z. Chen. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.
- [15] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C: Emerging Technologies*, 6(4):271–288, 1998.
- [16] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.
- [17] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.
- [18] R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pages 64–69. IEEE, 2005.
- [19] A. Doucet, S. J. Godsill, and C. Andrieu. *On sequential simulation-based methods for Bayesian filtering*. Department of Engineering, University of Cambridge UK, 1998.
- [20] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009.
- [21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

- [22] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *IEEE Pervasive Computing*, 2(3):24–33, 2003.
- [23] M.S Grewal and A. P. Andrews. *Kalman filtering: theory and practice using MATLAB*. Wiley. com, 2011.
- [24] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*, volume 2. Cambridge University Press, 2000.
- [25] C. Herdtweck and C. Wallraven. Horizon estimation: perceptual and computational experiments. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, pages 49–56. ACM, 2010.
- [26] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
- [27] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, 2003.
- [28] Y.-K. Jung and Y.-S. Ho. Traffic parameter extraction using video-based vehicle tracking. In *IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, pages 764–769, 1999.
- [29] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):108–118, 2000.
- [30] J. B. Kim and H. J. Kim. Efficient region-based motion segmentation for a video monitoring system. *Pattern Recognition Letters*, 24(1):113–128, 2003.
- [31] D. Koller, K. Daniilidis, and H.-H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, 1993.
- [32] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, volume 1, pages 126–131. IEEE, 1994.
- [33] A. Kong, J. S. Liu, and W. H. Wong. Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288, 1994.
- [34] P. Kumar, S. Ranganath, H. Weimin, and K. Sengupta. Framework for real-time behavior interpretation from traffic video. *IEEE Transactions on Intelligent Transportation Systems*, 6(1):43–53, 2005.

- [35] J.-C. Lai, S.-S. Huang, and C.-C. Tseng. Image-based vehicle tracking and classification on the highway. In *International Conference on Green Circuits and Systems*, pages 666–670. IEEE, 2010.
- [36] V. Lepetit and P. Fua. *Monocular Model-based 3D Tracking of Rigid Objects: A survey*. Now Publishers Inc, 2005.
- [37] Z. Li, Q. L. Tang, and N. Sang. Improved mean shift algorithm for occlusion pedestrian tracking. *Electronics Letters*, 44(10):622–623, 2008.
- [38] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.
- [39] E. Maggio and A. Cavallaro. *Video tracking: theory and practice*. 2011.
- [40] E. Maggio, F. Smerladi, and A. Cavallaro. Adaptive multifeature tracking in a particle filtering framework. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(10):1348–1359, 2007.
- [41] K. K. Ng and E. J. Delp. New models for real-time tracking using particle filtering. In *IS&T/SPIE Electronic Imaging*, pages 72570B–72570B–12. International Society for Optics and Photonics, 2009.
- [42] K. Nummiaro, E. Koller-Meier, and L. Van Gool. Object tracking with an adaptive color-based particle filter. In *Pattern Recognition*, pages 353–360. Springer, 2002.
- [43] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110, 2003.
- [44] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Computer Vision-ECCV*, pages 661–675. Springer, 2002.
- [45] B. Ristic, S. Arulampalam, and N. J. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.
- [46] N. Saunier and T. Sayed. A feature-based tracking algorithm for vehicles in intersections. In *The 3rd Canadian Conference on Computer and Robot Vision*, pages 59–59. IEEE, 2006.
- [47] C. Shen, A. Van den Hengel, and A. Dick. Probabilistic multiple cue integration for particle filter based tracking. In *Proceedings of the VIIth International Conference on Digital Image Computing: Techniques and Applications*, pages 309–408, 2003.
- [48] S. N. Sinha, J. M.l Frahm, M. Pollefeys, and Y. Genc. GPU-based video feature tracking and matching. In *EDGE, Workshop on Edge Computing Using New Commodity Architectures*, volume 278, page 4321, 2006.
- [49] X. Song and R. Nevatia. A model-based vehicle segmentation method for tracking. In *Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1124–1131, 2005.

- [50] B. Sugandi, H. Kim, J. K. Tan, and S. Ishikawa. A color-based particle filter for multiple object tracking in an outdoor environment. *Artificial Life and Robotics*, 15(1):41–47, 2010.
- [51] E. Trucco and K. Plakas. Video tracking: a concise survey. *IEEE Journal of Oceanic Engineering*, 31(2):520–529, 2006.
- [52] Y. Wu. Visual tracking. <http://www.ece.northwestern.edu/~yingwu/teaching/EECS432/Notes/tracking.pdf>, March 2013. Accessed December 15, 2013.
- [53] C. Yang, R. Duraiswami, and L. Davis. Efficient mean-shift tracking via a new similarity measure. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 176–183, 2005.
- [54] C. Yang, R. Duraiswami, and L. Davis. Fast multiple object tracking via a hierarchical particle filter. In *IEEE International Conference on Computer Vision*, volume 1, pages 212–219, 2005.
- [55] A. Yilmaz. Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.
- [56] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):13, 2006.