

# Gestaltung barrierearmer mobiler Systeme für Menschen mit Sehbehinderung

Andreas Günther  
Hochschule für Technik und Wirtschaft  
Wilhelminenhofstr. 75A, 12459 Berlin  
E-Mail: [andreas.guenther@student.htw-berlin.de](mailto:andreas.guenther@student.htw-berlin.de)

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	Ziel der Arbeit . . . . .	3
<b>2</b>	<b>Grundlagen</b>	<b>4</b>
2.1	Sehbehinderung . . . . .	4
2.2	Barrierefreiheit/-armut . . . . .	5
2.3	Digitale Bildverarbeitung . . . . .	7
2.3.1	Segmentierung . . . . .	7
2.3.2	Motion tracking . . . . .	9
<b>3</b>	<b>Entwurf</b>	<b>10</b>
3.1	Anwendungsszenario . . . . .	10
3.2	Systementwurf . . . . .	11
3.3	Algorithmus . . . . .	12
<b>4</b>	<b>Implementierung</b>	<b>13</b>
4.1	Image capturing . . . . .	13
4.2	Image processing . . . . .	14
4.2.1	Segmentierung . . . . .	14
4.2.2	Motion Tracking . . . . .	14
4.3	Status Output . . . . .	15
<b>5</b>	<b>Evaluation</b>	<b>15</b>
5.1	Segmentierung . . . . .	16
5.2	Objekttracking . . . . .	17
5.3	Schlussfolgerungen . . . . .	18
<b>6</b>	<b>Schlussbetrachtungen</b>	<b>18</b>

# 1 Einleitung

## 1.1 Motivation

Das alltägliche Leben von Sehbehinderten Menschen besteht aus Herausforderungen, die für einen Menschen ohne solche Einschränkungen auf den ersten Blick trivial erscheinen. Dinge wie Surfen im Internet, die Orientierung im Straßenverkehr oder sogar ein einfacher Spaziergang werden ohne ausreichende Sehkraft zu schwierigen Aufgaben des täglichen Lebens. Die Wahrnehmung der Lebensumgebung erfolgt zu 80% über das Auge.[Deu10] Menschen, denen diese visuelle Information fehlt oder deren visuelle Wahrnehmung getrübt ist, sind stärker auf die verbleibenden Sinnesorgane angewiesen, die durch Hilfsmittel zusätzlich unterstützt werden können.

Aus [Deu10, S. 18] geht hervor, dass die Orientierung im Straßenverkehr die größte Herausforderung darstellt. Der Straßenverkehr mit Autos, Radfahrern, Fußgängern und der damit verbundenen Geräuschkulisse machen die Orientierung schwierig, wenn man sich allein auf Gehör, Gedächtnis und einem kleinen Sehrest verlassen muss. Unvorhergesehene Ereignisse oder Veränderungen auf bekannten und gewohnten Wegen stellen nicht nur Herausforderungen dar, sie sind potentielle Gefahrenquellen. Ein großes Problem bei der Orientierung im Straßenverkehr ist das Überqueren von Straßen. Die Schwierigkeit liegt darin abzuschätzen, ob die Straße frei zum Überqueren ist oder sich ein Fahrzeug nähert. Ist eine Ampel vorhanden, besteht das Problem, das Signal auf der gegenüberliegenden Straßenseite zu erkennen. Da aber der Seheindruck getrübt ist, wie in Abbildung 1 zu sehen, wird man hier auf fremde Hilfe angewiesen sein. Dabei ist es „[...] Sehbehinderten Menschen [...] sehr wichtig, ihr Leben so weit wie möglich selbstständig zu bewältigen.“ [Deu10, S. 19] Signalanlagen mit akustischen oder haptischen Signalgebern helfen hier zwar, jedoch sind diese nicht überall vorhanden.



(a) Sicht eines normal Sehenden



(b) Sicht eines sehbehinderten Menschen mit einem Restsehvermögen von 10%



(c) Sicht eines sehbehinderten Menschen mit Röhrengesichtsfeld

Abbildung 1: Seheindrücke an Straßenüberquerung [Deu10]

## 1.2 Ziel der Arbeit

Diese Arbeit soll eine Grundlage für ein mobiles System darstellen, welches sehbehinderte Menschen bei sich tragen und sie bei der Straßenüberquerung zuverlässig unterstützt. Es soll die

Einschränkung der visuellen Wahrnehmung ausgleichen und das sichere Überqueren von Straßen ermöglichen. Ziel ist es, zu ermitteln inwiefern es möglich ist mit Hilfe von visueller Bildverarbeitung eine Aussage darüber zu treffen, ob die sichere Überquerung einer Straße gewährleistet ist. Diese Einschätzung ist, wie bereits erwähnt, für Menschen mit Sehbehinderung schwer zu treffen. Hier soll das System eine möglichst zuverlässige Aussage treffen können.

Grundlegend wird hier untersucht, welche Möglichkeiten es in der visuellen Bildverarbeitung gibt, um Fahrzeuge zu erkennen und zu verfolgen. Das Tracking der Fahrzeuge soll Aufschluss darüber geben, wie weit entfernt es von dem Standpunkt des sehbehinderten Menschen ist und wie schnell es sich auf diesen zubewegt. Anhand dieser erhobenen Daten soll ermittelt werden, ob das System potentiell in der Lage wäre, eine Aussage über die Sicherheit der Straßenüberquerung treffen zu können.

Die Herausforderung liegt hier zusätzlich darin, neben der Früherkennung eines Fahrzeuges und der Abschätzung dessen Geschwindigkeit und Entfernung, dass die Aufnahme der Videosequenz durch die Freihandaufnahme mit der mobilen Kamera verwackelt ist. Somit muss eine Methode gefunden werden, diese Verzerrungen im Sequenzablauf auszugleichen.

## 2 Grundlagen

### 2.1 Sehbehinderung

Von einer Sehbehinderung spricht man dann, wenn man nicht mehr als 30% Sehvermögen gegenüber eines normal sehenden Menschen besitzt. Ab einem Sehvermögen von 5% spricht man von einer hochgradigen Sehbehinderung und bei weniger als 3% von Blindheit.

Seheinschränkungen sind sehr vielfältig geartet. Je nach Erkrankung variiert der verbleibende Seheindruck sehr stark. Die Spannweite reicht von unscharfem Sehen über Flecken im Sichtfeld bis hin zu Farbfehlsichtigkeit. Selbst bei gleicher Erkrankung kann das Restsehvermögen zweier Menschen sehr unterschiedlich sein. Die häufigsten Erkrankungen sind:

- Katarakt (Grauer Star)
- Retinitis Pigmentosa
- Altersabhängige Makula-Degeneration (AMD)
- Retinopathia diabetica
- Glaukom (Grüner Star)
- Farbfehlsichtigkeit

Die Abbildung 2 zeigt exemplarisch, wie sich die einzelnen Erkrankungen auswirken und unterscheiden.



(a) Katarakt (Grauer Star)



(b) Retinitis Pigmentosa



(c) Altersabhangige Makula-Degeneration (AMD)



(d) Retinopathia diabetica



(e) Glaukom (Gruner Star)



(f) Farbfehlsichtigkeit

Abbildung 2: Seheindrucke der am haufigsten auftretenden Seherkrankungen nach [Deu10]

Nach [Ber06], der sich auf eine Studie der WHO<sup>1</sup> von 2002 bezieht, ist die haufigste Ursache fur Sehbehinderungen in Deutschland die Altersbedingte Makula-Degeneration (siehe Abbildung 3). Da diese, wie der Name schon suggeriert, im Alter auftritt, ist bei der derzeitigen demographischen Entwicklung [Sta09] damit zu rechnen, dass es in Zukunft potentiell mehr sehbehinderte Menschen geben wird. Daher wird die Schaffung von Barrierearmen bzw. -freien Umgebungen in Zukunft noch wichtiger.

## 2.2 Barrierefreiheit/-armut

Nach dem Deutschen Gesetz zur Gleichstellung von Behinderten [Bun] definiert sich Barrierefreiheit wie folgt:

"Barrierefrei sind bauliche und sonstige Anlagen, Verkehrsmittel, technische Gebrauchsgegenstande, Systeme der Informationsverarbeitung, akustische und visuelle Informationsquellen und Kommunikationseinrichtungen sowie andere gestaltete Lebensbereiche, wenn sie fur behinderte Menschen in der allgemein ublichen Weise, ohne besondere Erschwernis und grundsatzlich ohne fremde Hilfe zuganglich und nutzbar sind." (§4, BGG)

Ziel dieses Gesetzes ist es, nach §1 BGG [Bun], die Benachteiligung von behinderten Menschen zu verhindern und zu beseitigen, bzw. ihnen die Moglichkeit bieten, gleichberechtigt am gesellschaftlichen Leben teilzunehmen und eine selbstbestimmte Lebensfuhrung zu ermoglichen.

---

<sup>1</sup>World Health Organisation

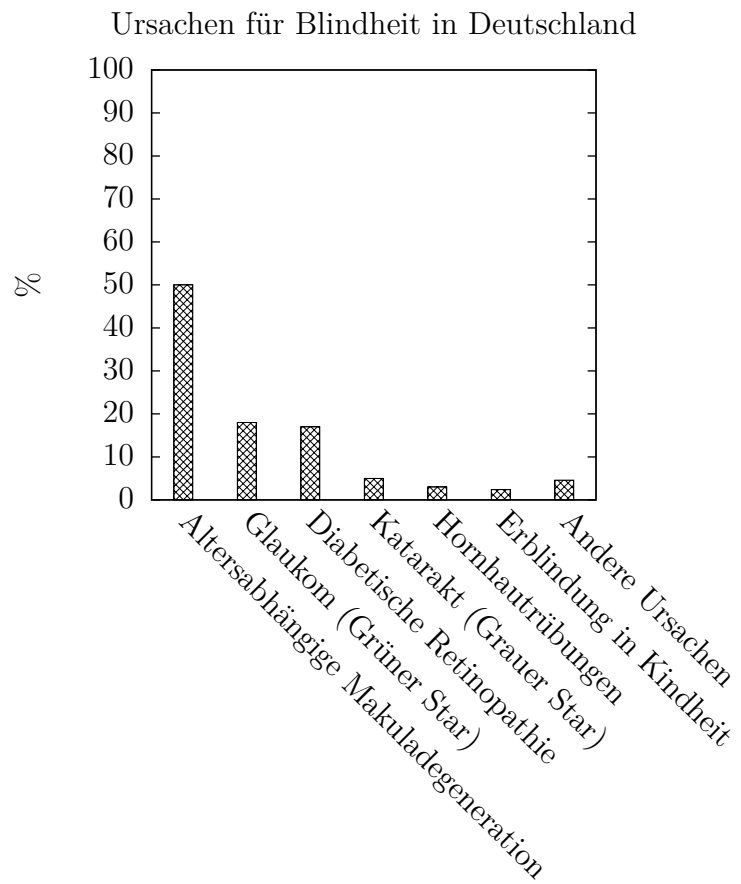


Abbildung 3: Häufigste Ursache für Blindheit: Altersbedingte Makuladegeneration nach [Ber06]

Dabei muss auf die besonderen Bedürfnisse, die behinderte Menschen im allgemeinen haben, Rücksicht genommen werden.

Im Kontext der Sehbehinderung bedeutet dies, dass Einrichtungen und Hilfsmittel angeboten werden müssen, um das eingeschränkte Sehvermögen auszugleichen bzw. qualitativ zu verbessern, beispielsweise durch Blindenleitsysteme im Straßenverkehr oder akustischen Signalen bzw. Sprachausgaben sowie haptischem Feedback, bei sonst auf visuellen Ausgaben basierenden Systemen.

Im Bereich der Anwendungsentwicklung gibt es Richtlinien, die bei der Entwicklung beachtet werden sollten, um diese Anwendungen für behinderte Menschen zugänglich zu machen. Google [dev], als auch Apple [App10], haben beispielsweise solche Richtlinien für ihre jeweilige Plattform herausgegeben. Sie beschreiben, wie Entwickler ihre Anwendungen gestalten sollten, um sie behinderten Menschen zugänglich zu machen.

## 2.3 Digitale Bildverarbeitung

Das Ziel der Arbeit ist es, ein System zu entwickeln, welches mit Hilfe digitaler Bildverarbeitung abschätzen soll, ob es sicher ist, eine Straße zu überqueren. In diesem Abschnitt soll auf grundlegende Aspekte dieses Gebietes eingegangen werden. Da die digitale Bildverarbeitung ein sehr komplexes Thema darstellt, wird sich hier nur auf Segmentierung und Motion Tracking beschränkt. Diese Teilgebiete spielen für die Arbeit eine wichtige Rolle bei der späteren Umsetzung.

### 2.3.1 Segmentierung

Eine der Kernaufgaben in der digitalen Bildverarbeitung ist die Trennung von Vordergrund und Hintergrund in einer Bildsequenz. [LHGT03] Grundlegend für die Aufteilung von Vordergrund- und Hintergrundelementen ist ein Modell des Hintergrunds. Dieses wird mit der aktuellen Bildszene verglichen, um die Vordergrundobjekte abzutrennen. Im Folgenden wird zunächst das grundlegende Verfahren bei der Vordergrund- und Hintergrundsegmentierung dargestellt und im Anschluss eine komplexere Methode, die nach [LHGT03] gute Ergebnisse bei Außenaufnahmen erzielt. Weitere Ansätze zur Segmentierung können [HHD00, SG00, Wix00] entnommen werden.

**Background subtraction** Unter der Annahme, dass das Hintergrundbild bekannt ist, kann der Vordergrund des aktuellen Bildes einer Videosequenz mit Hilfe einer einfachen Bilddifferenz ermittelt werden. Dabei werden die Pixel des aktuellen Videobildes mit dem Hintergrundbild verglichen. Alle Pixel, die sich vom Hintergrundbild unterscheiden, werden als Vordergrundpixel deklariert. Da sich der Hintergrund in einer Videosequenz auch über die Zeit verändert, z.B. durch Änderung der Lichtverhältnisse über den Tag, ist es notwendig ein dynamisches Modell des Hintergrundes zu berechnen. Dabei wird das Hintergrundbild stetig aktualisiert und mit dem aktuellen Videobild verglichen. Nach [Lag11] kann ein solches Modell mit Hilfe eines sogenannten *running average* berechnet werden. Mit Hilfe dessen kann ein Durchschnittswert eines Pixels unter Berücksichtigung des vorherigen Wertes berechnet werden. Die folgende Formel

beschreibt wie der Durchschnitt aktualisiert wird.

$$\mu_t = (1 - \alpha)\mu_{t-1} + \alpha p_t \quad [\text{Lag11}] \quad (1)$$

Dabei ist  $p_t$  ein Pixelwert zu einer bestimmten Zeit  $t$  und  $\mu_{t-1}$  der aktuelle Durchschnittswert. Der Parameter  $\alpha$  beschreibt den Einfluss des aktuellen Wertes auf den aktuell geschätzten Wert. In [Lag11] wird er als *learning rate* bezeichnet. Je höher der Wert von  $\alpha$  desto schneller passt sich der *running average* an Veränderungen an. Für die dynamische Berechnung des Hintergrundes wird der *running average*-Wert für jedes Pixel der folgenden Bildsequenzen berechnet. Durch die Berechnung der Differenz zwischen Hintergrund-Modell und aktuellem Sequenzbild können Vordergrundpixel vom Hintergrundpixel segmentiert werden. Eine einfache Implementierung kann [Lag11] entnommen werden.

Diese, im Vergleich zu anderen Algorithmen, recht einfache Methode der Segmentierung arbeitet für Videosequenzen mit relativ statischem Hintergrund recht gut. Jedoch treten in einigen Bereichen des Hintergrunds Wertschwankungen auf, die dazu führen, dass diese Bereiche als Vordergrund deklariert werden, die eigentlich zum Hintergrund gehören. Beispiele dafür sind Blätter im Wind oder Wasserspiegelungen. [Lag11]

**Mixture of Gaussian** Diese Methode ist eine Erweiterung der einfachen *Background subtraction*. Bei ihr werden mehrere Hintergrundmodelle pro Pixel gebildet. Variiert beispielsweise ein Pixel zwischen zwei Werten, werden dazu zwei *running average* berechnet und gespeichert. Erst wenn ein Pixel keines der berechneten Modelle zugeordnet werden kann, wird es als Vordergrundpixel deklariert. [Lag11] Zusätzlich wird neben dem *running average* auch zu jedem Modell die sogenannte *running variance* nach der folgenden Formel berechnet:

$$\sigma_t^2 = (1 - \alpha)\sigma_{t-1}^2 + \alpha(p_t - \mu_t)^2 \quad [\text{Lag11}] \quad (2)$$

*Running average* und *running variance* bilden ein Gauß-Modell, mit Hilfe dessen die Wahrscheinlichkeit ermittelt werden kann, mit der ein Pixel zu diesem Gauß-Modell gehört. Dies macht es einfacher, einen entsprechenden Schwellwert für die Abgrenzung zwischen Vordergrund und Hintergrund zu bestimmen, da dieser als Wahrscheinlichkeit ausgedrückt wird anstelle einer absoluten Differenz. In Hintergrundbereichen, die starken Wertschwankungen unterlegen sind, ist eine höhere Differenz nötig, um ein Vordergrundobjekt zu ermitteln. [Lag11]

Alle Gauß-Modelle, die nicht ausreichend oft getroffen wurden, werden als Teil des Hintergrundes ausgeschlossen. Für alle Pixelwerte, die keinem der aktuellen Gauß-Modelle zugeordnet werden können, wird ein neues Gauß-Modell gebildet. Wird dieses Modell zukünftig regelmäßig getroffen, wird es dem Hintergrund zugeordnet. [Lag11]

Eine detailliertere Beschreibung der „Mixture-of-Gaussian“-Methode kann [KB01] entnommen werden.



### 2.3.2 Motion tracking

Objekttracking beschreibt die Verfolgung eines sich bewegenden Objekts von Interesse in einer Videosequenz. Aus der Literatur [BK08] gehen zwei grundlegend verschiedene Verfahren hervor, auf die im Folgenden eingegangen wird.

**Optical Flow** Bewegung zwischen zwei Frames einer Videosequenz abschätzen ohne vorherige Kenntnis über dessen Inhalt. Die Bewegung selbst ist ein Indikator dafür, dass etwas von Interesse innerhalb der Videosequenz passiert. [BK08, S. 322] Durch die Verschiebung der Pixel kann so eine Distanz bzw. eine Art Geschwindigkeit für jedes Pixel ermittelt werden. Hier kann zwischen *dense optical flow* und *sparse optical flow* unterschieden werden. In Abbildung 4 ist der *Optical Flow* dargestellt.

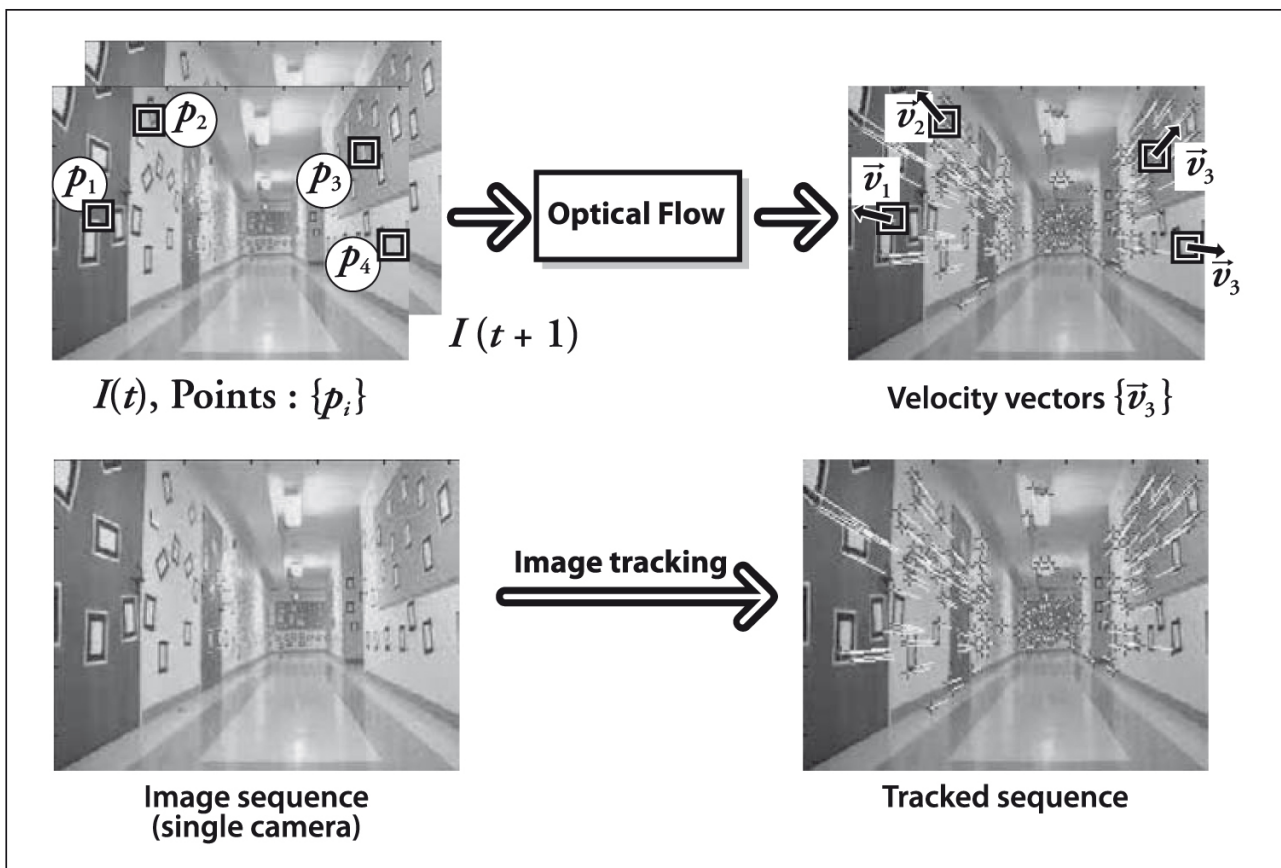


Abbildung 4: Optical Flow [BK08]

Bei *dense optical flow* Techniken wird jeder Pixelwert in einer Sequenz von Frames berücksichtigt. Die Berechnung des *dense optical flow* ist daher nicht leicht, wie [BK08, S. 323] anführt. Hier heißt es: Betrachtet man die Bewegung eines weißen Blatt Papiers, werden viele der weißen Pixel im folgenden Frame ebenso weiß bleiben. Nur die Kanten würden sich senkrecht zur Bewegungsrichtung bewegen. Daher müssen solche Methoden Verfahren anwenden, die zwischen Punkten interpolieren, die leichter zu verfolgen sind. Das resultiert in einem hohen rechnerischen Aufwand. Die *Horn-Schunck-Methode* [HS81] beschreib einen Algorithmus, der zum *dense optical flow* zu zählen ist. [BK08]

*Sparse-optical-flow*-Algorithmen arbeiten nur mit einer Teilmenge der im Frame befindlichen Pixel. Unter der Voraussetzung, dass diese Punkte Eigenschaften aufweisen, die sich in den folgenden Frames gut wiederfinden lassen, kann man relativ robuste und zuverlässige Ergebnisse erwarten. [BK08] Zudem ist der Rechenaufwand im Gegensatz zum *dense optical flow* geringer. Ein wichtiger Vertreter ist hier der *Lucas-Kanade*-Algorithmus. [LK81] Dieser basiert nach [BK08] auf folgenden Annahmen:

- *Brightness constancy*. Der Pixelwert eines sich bewegenden Objekts innerhalb einer Videosequenz verändert sich nicht bei einer Bewegung von Frame zu Frame.
- *Temporal persistence or "small movements"*. Objekte bewegen sich nur geringfügig von Frame zu Frame.
- *Spatial coherence*. Benachbarte Punkte gehören zur selben Oberfläche und bewegen sich ähnlich.

Eine detailliertere Beschreibung des Algorithmus kann [BK08, LK81] entnommen werden.

**Motion Templates** *Motion Temples* sind nach [BK08] eine effektive Methode, um Bewegungen in Videosequenzen zu ermitteln. Hierzu benötigt man eine Silhouette oder auch nur einen Teil der Silhouette eines zu trackenden Objekts. Dies kann beispielsweise durch Segmentierung erreicht werden. Verfolgt man die Silhouette von Frame zu Frame, kann so eine *Motion History* aufgebaut werden. Mit Hilfe dieser *Motion History* können anschließend Aussagen über die Bewegungsrichtung der Silhouette gemacht werden. Das *openCV* Framework bietet dazu eine Reihe von Funktionen, mit denen diese Art des Motion Trackings umgesetzt werden kann. Eine Beschreibung dessen findet sich in [BK08].

## 3 Entwurf

Wie bereits eingangs erwähnt, ist diese Arbeit darauf ausgerichtet, grundlegend zu ermitteln, ob es mithilfe der digitalen Bildverarbeitung möglich ist, die Entfernung und Geschwindigkeit eines Fahrzeuges zu ermitteln. Ziel ist es, eine Aussage darüber zu treffen, ob ein technisches System potentiell in der Lage wäre eine Aussage darüber zu treffen, ob eine sichere Straßenüberquerung gewährleistet ist. Dabei wird die Aufzeichnung der digitalen Bilddaten und deren Verarbeitung durch ein mobiles Gerät, das der Sehbehinderte bei sich trägt, vorgenommen.

### 3.1 Anwendungsszenario

Das Einsatzszenario orientiert sich am klassischen Verhalten bei Straßenüberquerungen. Möchte man eine Straße überqueren, sieht man zunächst nach links, ob die Straße frei ist. Danach schaut man nach rechts und im Anschluss daran noch einmal nach links.

Da in dieser Arbeit nur grundlegende Aspekte untersucht werden, wird das Anwendungsszenario auf Einbahnstraßen beschränkt. Hier benötigt man nur die Einschätzung ob die Straße in eine

Fahrtrichtung frei ist. Des Weiteren wird sich auf das Szenario beschränkt, dass sich nur ein potentiell Fahrzeug dem straßenüberquerenden Protagonisten nähern kann. Somit ist es nur nötig in eine Richtung zu schauen, um einzuschätzen, ob die Straße frei ist und man diese überqueren kann.

Sehbehinderte Menschen haben es trotz dieser Einschränkungen schwer einzuschätzen, ob ein sich annäherndes Fahrzeug zu schnell auf sie zukommt, als dass eine Straßenquerung sicher ist bzw. ob überhaupt ein Fahrzeug auf der Straße ist. Daher soll er mit einem mobilen Assistenzsystem ausgestattet werden, welches die visuellen Einschränkungen ausgleicht. In diesem Kontext richtet der Benutzer die Kamera des Assistenzsystems entgegen der Fahrtrichtung auf die Straße. Das System erkennt nun, ob, in welcher Entfernung und mit welcher Geschwindigkeit sich ein Fahrzeug dem sehbehinderten Menschen nähert. Anhand dieser Daten ist das System in der Lage, dem Sehbehinderten entsprechend mitzuteilen, wann es sicher ist die Straße zu überqueren oder die Gefahr eines Unfalls besteht. Das Mitteilen über die aktuelle Straßensituation kann nicht visuell erfolgen, da dies für den Sehbehinderten nicht angemessen wäre. Hier ist haptisches und akustisches Feedback angemessen.

### 3.2 Systementwurf

Das System besteht aus drei grundlegenden Komponenten, die in Abbildung 5 dargestellt sind. Die *Image capturing*-Komponente hat die Aufgabe, ein Live-Bild der aktuellen Situation auf der Straße aufzunehmen und dieses an die *Image processing*-Komponente weiterzugeben. Diese verarbeitet die digitalen Bilddaten und gibt die gewonnenen Daten an die *Status output*-Komponente weiter, die versucht, eine Aussage über die Sicherheit der Straßenüberquerung vorzunehmen und den Benutzer entsprechend über haptische und akustische Ausgaben diesen Status mitzuteilen.



Abbildung 5: Systemkomponenten

Die *Image processing*-Komponente gliedert sich weiter in zwei Verarbeitungsschritte. Zum Einen in die *segmentation* des digitalen Bildes und des Weiteren in das *motion tracking* (siehe Abbildung 6). Bei der Segmentierung soll das Fahrzeug auf der Straße vom Hintergrund abgetrennt werden und anschließend per *motion tracking*-Verfahren verfolgt werden, um daraus Informationen über die Geschwindigkeit und Entfernung zu gewinnen.

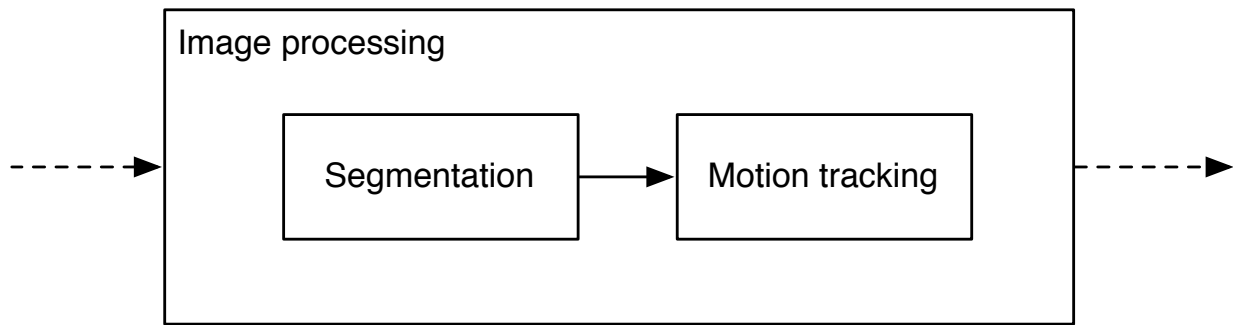
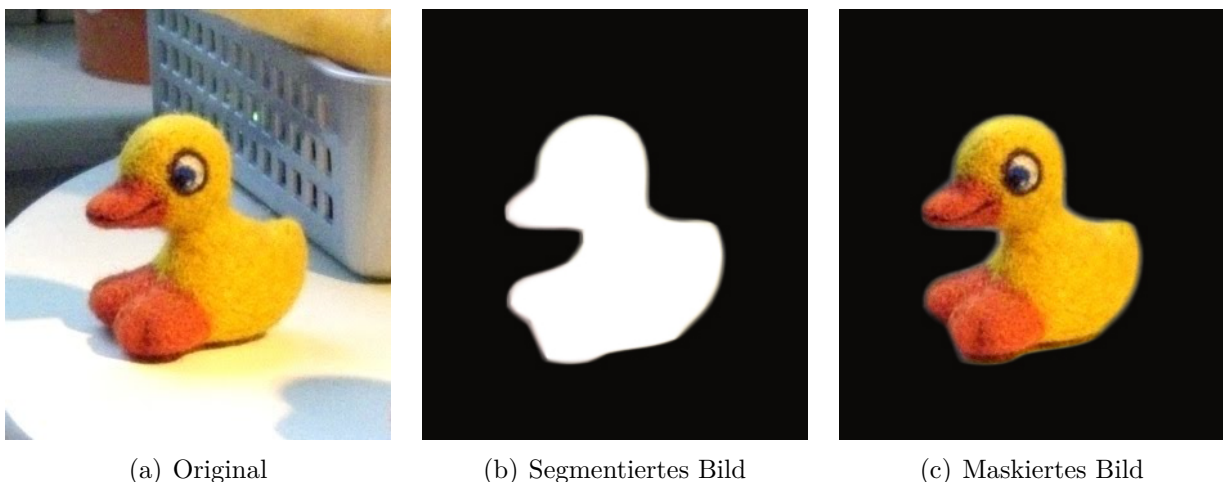


Abbildung 6: Systemkomponenten

### 3.3 Algorithmus

Der Algorithmus wird für jeden Frame einer Videosequenz durchlaufen. Zunächst wird der aktuelle Frame segmentiert, um Hintergrund- von Vordergrundobjekten zu trennen. Nach der Segmentierung werden die Konturen der Vordergrundobjekte ermittelt, um daraus ihre Boundingbox zu bestimmen. Somit erhält man Informationen darüber wo, sich das Objekt im aufgenommenen Bild befindet. Mit den Positionsdaten wird nun das Originalbild maskiert, so dass nur noch Farbinformationen zu den segmentierten Objekten im Bild enthalten sind. In Abbildung 7 ist diese Maskierung bildlich dargestellt.



(a) Original

(b) Segmentiertes Bild

(c) Maskiertes Bild

Abbildung 7: Maskierung eines Originalbildes

Im Anschluss an die Maskierung werden nach der *optical flow*-Methode *feature points* für die noch im Bild vorhandenen Pixel bestimmt. Anhand dessen sollen die Objekte im zeitlichen Verlauf der Videosequenz identifiziert werden, um so die Bewegung verfolgen zu können. Kann ein Objekt in den nachfolgenden Frames erfolgreich identifiziert werden, berechnet der Algorithmus einen Vektor, der Aufschluss über die Bewegungsrichtung des Objektes gibt. Bewegt sich das Objekt nicht auf die Kamera zu, wird es verworfen und nicht weiter beachtet. Für alle Objekte, die sich auf die Kamera zu bewegen, werden relative Geschwindigkeit und Größenänderung berechnet. Dieser Ablauf ist in Algorithmus 1 als Pseudocode noch einmal dargestellt.

---

**Algorithm 1**

---

```
for all Video-Frames do
  Segmentiere aktuellen Frame
  Ermittle Boundingboxen zu segmentierten Objekten
  if Anzahl der Boundingboxen > 0 then
    Erstelle Maske und maskiere aktuellen Frame
    Ermittle feature points
    Identifiziere Boundingboxen anhand ihrer feature points aus vorherigen Frames
    if Identifizierung erfolgreich == TRUE then
      Ermittle Richtungsvektor
      if Bewegung in Richtung Kamera == TRUE then
        Berechne Geschwindigkeit und Größenänderung des Objekts
      end if
    end if
  end if
end for
```

---

## 4 Implementierung

Die prototypische Implementierung des Systems wurde auf der iOS-Plattform von Apple vorgenommen. Somit kommen als mobile Geräte iPhone, iPad sowie iPod Touch in frage. Jedes dieser Geräte ist in der aktuellen Generation<sup>2</sup> mit einer Kamera ausgestattet und erlaubt es somit digitale Bildsequenzen zur Verarbeitung aufzunehmen. Des Weiteren wurde eine Implementierung für diese Plattform des *openCV*-Frameworks genutzt. Dieses beinhaltet eine Vielzahl von Funktionen für die digitale Bildverarbeitung.

Der Prototyp gliedert sich wie im Abschnitt 3.2 beschrieben in Bildaufnahme, Bildverarbeitung und Ausgabe der erhaltenen Daten. Der *status output* ist hier allerdings, nicht wie beschrieben bereits eine haptische und akustische Ausgabe sondern eine visuelle und textuelle Ausgabe in eine Datei zu Evaluationszwecken. Die hier ausgegeben Daten sollen zunächst grundlegend klären, ob ein technisches System mit Mitteln der digitalen Bildverarbeitung in der Lage ist eine zuverlässige Aussage über die Straßensituation zu geben.

Abbildung 8 zeigt den groben Aufbau des Systems.

### 4.1 Image capturing

Das entstandene Anwendung ist in der Lage Bilddaten aus zwei Quellen zu verarbeiten. Das System kann Daten direkt von der Kamera am iOS-Gerät aufnehmen oder diese aus einer zuvor aufgezeichneten Videodatei zur Weiterverarbeitung bereitstellen. Dazu wurde eine Schnittstelle entworfen, zu der es zwei konkrete Implementierungen gibt, die diese Funktionalität bieten.

---

<sup>2</sup>Stand März 2012

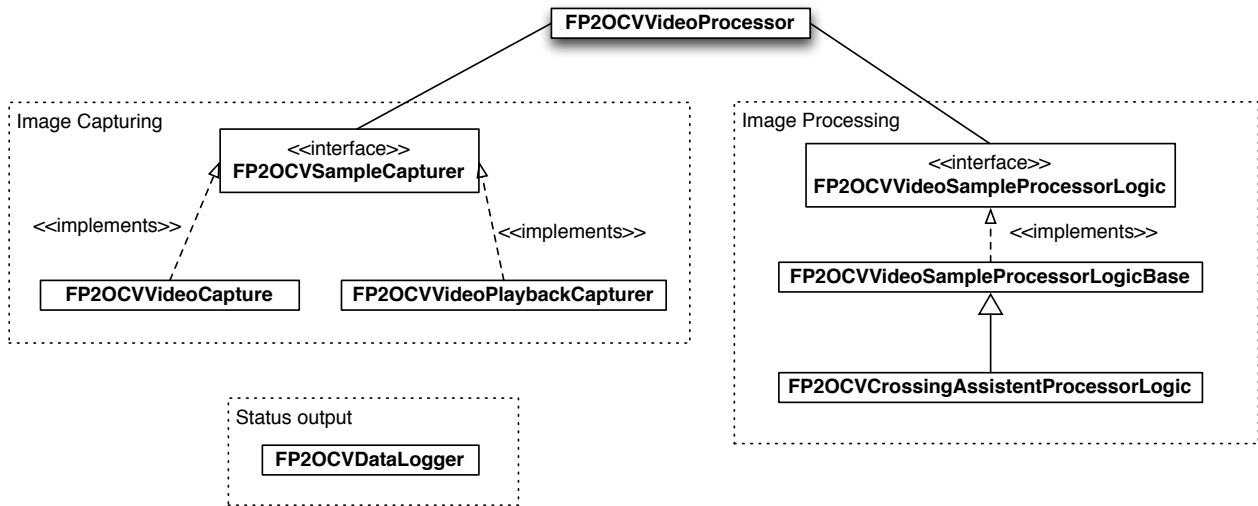


Abbildung 8: Architektur der prototypischen Anwendung

## 4.2 Image processing

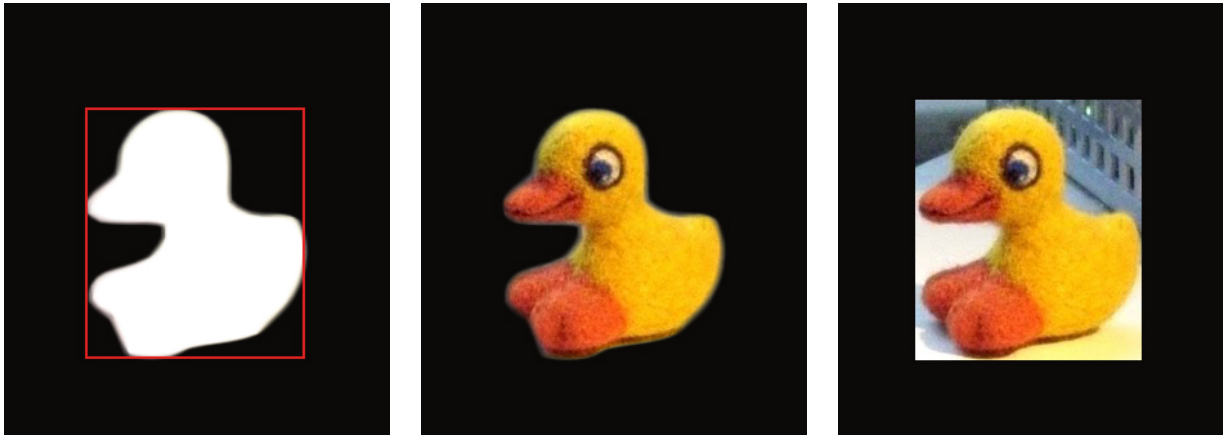
Die Verarbeitung der Bilddaten entspricht dem in Abschnitt 3.3 beschriebenen Algorithmus. Dieser ist in der Klasse *FP2OCVCrossingAssistentProcessorLogic* implementiert und nutzt Funktionen des *openCV*-Framework zur Bildverarbeitung. Die Funktionalität dieser Klasse gliedert sich in *Segmentierung* und *motion tracking*.

### 4.2.1 Segmentierung

Die Aufteilung des aktuellen Frames wird durch eine Instanz der Klasse *cv::BackgroundSubtractorMOG2* vorgenommen. Diese implementiert die *Mixtur of Gaussian*-Methode, die in Abschnitt 2.3.1 beschrieben ist. Die Segmentierung liefert ein Binärbild, in dem Vordergrundspixel weiß und Hintergrundpixel schwarz dargestellt sind. Anschließend wird dieses Bild noch von kleineren Artefakten bereinigt, so dass nur noch Objekte vorhanden sind, die eine ausreichende Größe zur Weiterverarbeitung aufweisen. Im nächsten Schritt werden die Boundingboxen der segmentierten Objekte bestimmt. Anhand der Konturen der Objekte wird die Position und Größe dieser im Bild ermittelt, die durch eine rechteckige Boundingbox dargestellt werden. Im letzten Schritt wird der aktuelle Frame mit dem segmentierten Bild maskiert. Dabei gibt es zwei Varianten. Entweder nur der segmentierte Bereich oder die gesamte Fläche der Boundingbox eines Objekts wird zur Maskierung genutzt (siehe Abbildung 9).

### 4.2.2 Motion Tracking

Das *motion tracking* wird mit Hilfe der *Lucas-Kanade*-Methode vorgenommen. Das *openCV*-Framework bietet dazu die Funktionen *cv::goodFeaturesToTrack()* und *cv::calcOpticalFlowPyrLK()*. Mit der Funktion *cv::goodFeaturesToTrack()* werden die *feature points* des aktuellen Frames ermittelt. Die Funktion *cv::calcOpticalFlowPyrLK()* ermittelt das *optical flow* dieser *feature*



(a) Segmentiertes Bild mit Boundingbox (b) Maskierung mit Hilfe des segmentierten Bereichs (c) Maskierung mit Hilfe der Fläche der Boundingbox

Abbildung 9: Maskierungsmethoden

*points* bezogen auf die vorherigen Frames. Anhand der *feature points* der Frames werden die Objekte im aktuellen Frame identifiziert und anschließend die Richtung des daraus ermittelten Bewegungsvektors berechnet. Die Richtung des Vetos wird durch den Winkel zur horizontalen Bildachse repräsentiert. Ist der Winkel kleiner als 45 Grad bewegt sich das Objekt auf die Kamera zu. Alle anderen Objekte werden nicht weiter betrachtet. Abschließend werden für die Objekte, die sich auf die Kamera zubewegen relative Geschwindigkeit und Größenänderung berechnet.

### 4.3 Status Output

Nachdem die Bildverarbeitung abgeschlossen ist, werden die ermittelten Daten als Text in eine Datei zur späteren Analyse ausgegeben. Zusätzlich wird jeder Frame als Bilddatei gespeichert. Daraus sind die erkannten Objekte mit der zugehörigen Boundingbox markiert und die Featurevektoren eingezeichnet.

## 5 Evaluation

Zu Evaluation der Anwendung wurden drei Videosequenzen aufgenommen, in denen sich ein einzelnes Fahrzeug, einer am Fahrbahnrand stehenden Person mit unterschiedlichen Geschwindigkeiten nähert. Diese Sequenzen wurden mit dem implementierten System verarbeitet und die entstandenen Daten analysiert. Untersucht wurden die Segmentierung und das Objekttracking sowie die Eignung für das eigentliche Assistenzsystem.

Da während der Entwicklung bereits festgestellt wurde, dass die Performance auf einem iPhone 4S doch sehr schlecht ist, wurde die Evaluierung mit dem iPhone Simulator auf einem MacBook durchgeführt.

## 5.1 Segmentierung

Bei den Tests zur Segmentierung wurde untersucht, wie sich die genutzte *Mixtur of Gaussian*-Methode (MOG) bei Bildsequenzen, die aus freier Hand aufgenommen wurden verhält. Speziell wurde untersucht welche *learning rate* (LR) geeignet ist, um zuverlässig Hintergrund von Vordergrund zu trennen. Dazu wurde die Segmentierung zunächst getrennt von den restlichen Verarbeitungsschritten mit verschiedenen Werten für die *learning rate* durchlaufen. In Abbildung 10 sind die Ergebnisse der Untersuchung dargestellt.

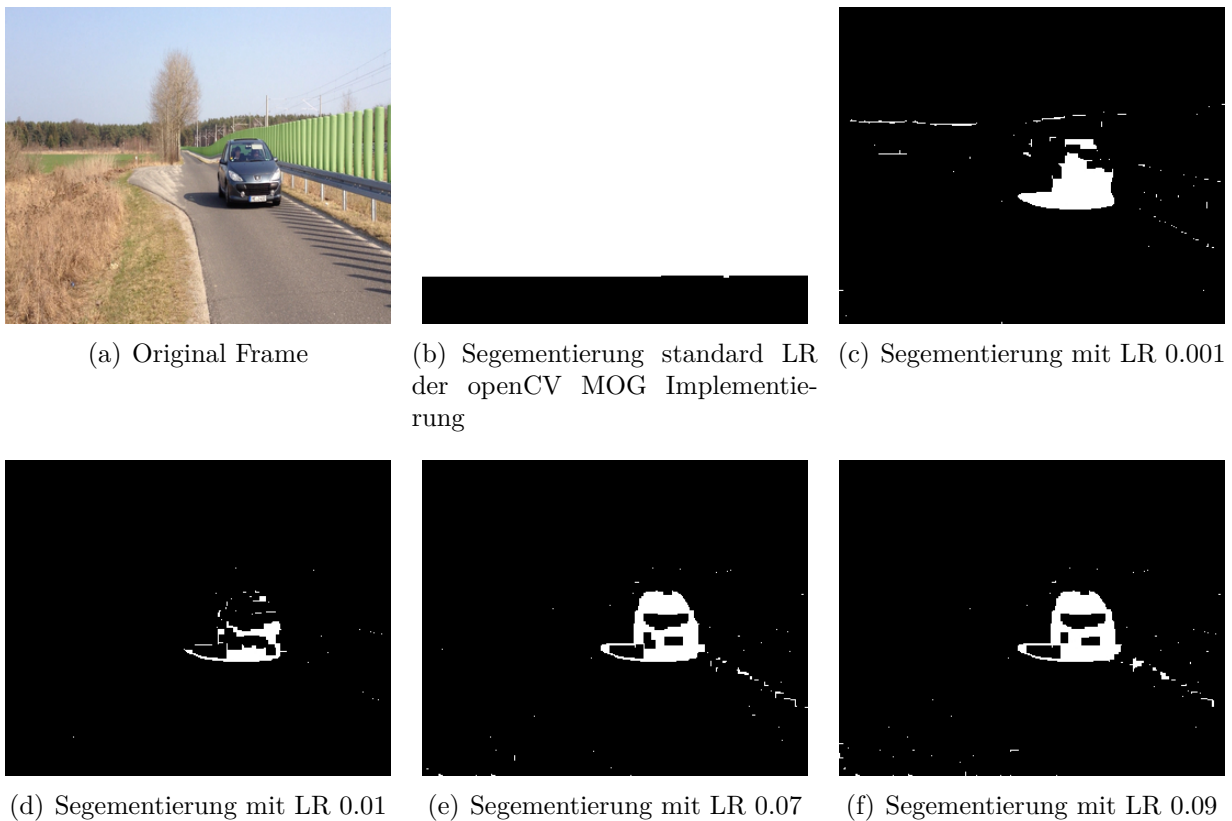


Abbildung 10: Segmentierung bei verschiedenen *learning rates*

Ergebnis dieser Untersuchung ist, dass eine Segmentierung mit einer *learning rate* zwischen 0.07 und 0.09 gut für diesen Testfall eignet. Alle weitere Untersuchungen wurden darauf hin mit einer *learning rate* von 0.07 durchgeführt.

Anschließend wurde untersucht, ob durch die Wahl des Farbraums, in dem die Segmentierung des Frames durchgeführt wird eine weitere Verbesserung der Objektextraktion erreicht werden kann. Dabei wurde festgestellt, dass dies keinen starken Einfluss auf die Segmentierung hat. Jedoch ergaben sich unterschiede in der Performance, die in *Frames per second* (fps) gemessen wurde. Die beste Performance wurde durch Graustufen erreicht, was daran liegt, dass ein Pixel nur eine Komponente für den Farbwert besitzt. Alle anderen Farbräume haben jeweils drei Komponenten, die vom MOG-Algorithmus durchlaufen werden müssen. Somit ist die Performance der Verarbeitung eines Graubildes nicht direkt mit den restlichen Farbräumen vergleichbar, bietet aber eine Orientierung. Der RGB-Farbraum ergab beim Vergleich mit den anderen dreikomponentigen Farbräumen die beste Performance. Aus diesem Grund wurde für



die weiteren Betrachtungen der RGB-Farbraum zur Segmentierung genutzt. Die Ergebnisse des Tests sind in Tabelle 1 aufgeführt.

Farbraum	Durchschnittl. Performance
Grauwerte (Orientierungswert)	71fps
RGB	55fps
HSV	28fps
HLS	27fps
YUV	47fps

Tabelle 1: Performance der Segmentierung in verschiedenen Farbräumen

## 5.2 Objekttracking

Hier wurde zunächst untersucht wie sich die Art der Maskierung des aktuellen Frames, wie in Abschnitt 4.2.1 beschrieben sich auf die Ermittlung der *feature points* des segmentierten Fahrzeugs auswirkt. Wie in Abbildung 11 zu sehen, können durch das System mehr *feature points* durch Maskierung des Frames mit der gesamten Fläche der Objekt-Boundingbox, ermittelt werden.



(a) Maskierung mit segmentiertem Bereich (b) Maskierung mit ermittelter Boundingbox

Abbildung 11: *Feature points* bei unterschiedlicher Maskierung

Wichtig für die Ermittlung der Geschwindigkeit, ist die zuverlässige Identifikation des Fahrzeugs. Da dies durch die *feature points* vorgenommen wird, wurde untersucht, wie die Qualität dieser Punkte sich auf die Identifizierung auswirkt. Es wurde die durchschnittliche Lebensdauer einer erkannten Boundingbox in der Videosequenz untersucht, um über die Zuverlässigkeit des Trackings eine Aussage treffen zu können. In Tabelle 2 sind die Ergebnisse zu sehen, aus denen hervorgeht, dass mit Abnahme der Qualität der *feature points* steigt die durchschnittliche Lebensdauer der Boundingboxen.

Abschließend wurde untersucht, ob die Geschwindigkeitsunterschiede bei den drei aufgenommenen Testvideos auch aus den gewonnenen Daten der Bildverarbeitung erkennbar sind. Das in Tabelle 3 dargestellte Ergebnis bestätigt dies. Auch ist aus den Daten eine Annäherung des Fahrzeugs zu erkennen.

<i>Feature point</i> Qualität	Durchschnittliche Lebensdauer
0,9	1,3 Frames
0,1	1,8 Frames
0,01	1,7 Frames
0,001	2,2 Frames
0,0001	2,2 frames

Tabelle 2: Durchschnittliche Lebensdauer von Boundingboxen bei unterschiedlicher Qualität der *feature points*

Fahrzeuggeschwindigkeit	Boundingbox Skalierung	Boundingbox Geschwindigkeit
30km/h	1,79	1,7 fps
50km/h	2,1	2,2 fps
70km/h	3,73	6,45 fps

Tabelle 3: Geschwindigkeit und Skalierung der Boundingboxen bei verschiedenen Fahrzeuggeschwindigkeiten

### 5.3 Schlussfolgerungen

Neben den bereits beschriebenen Ergebnissen wurde festgestellt, dass die Identifizierung nicht ausreichend Zuverlässig ist, um eine Aussage darüber treffen zu können, ob es sicher ist die Straße zu überqueren. Begründet liegt dies darin, dass ein Fahrzeug erst relativ spät erkannt wird und die Identifizierung der Boundingbox mit Hilfe der *feature points* äußerst lückenhaft ist. Es ist somit nicht möglich ein Fahrzeug durchgängig zu verfolgen. Dass sich ein Fahrzeug auf die Kamera zubewegt ist allerdings aus den Daten zu erkennen.

Die Geschwindigkeit eines Fahrzeugs lässt sich aus den gewonnenen Daten ermittelt werden, jedoch steht die ermittelte Geschwindigkeit nicht in Relation zur realen Umgebung und kann so nicht zur Einschätzung der Straßensituation hinzugezogen werden. Es fehlt hier eine Bezugsgröße. Diese zu finden ist allerdings auch im vorgegebenen Anwendungsszenario nicht zu finden, da sich die Umgebung bei jeder Anwendung verändert und zusätzlich auch die Aufnahme aus freier Hand Verzerrungen im Bild hervorruft.

## 6 Schlussbetrachtungen

Die Arbeit hat untersucht, ob es möglich ist ein System mit Mitteln der digitalen Bildverarbeitung zu schaffen, was potentiell in der Lage wäre eine Aussage über die sichere Überquerung einer Straße zu treffen. Die hier entstandene prototypische Anwendung hat anhand der Daten, die bei den Verarbeitungsschritten gewonnen wurden gezeigt, dass es sehr schwierig ist eine solche Aussage von einem Computersystem treffen zu lassen. Mit der hier vorgestellten Methode war dies nicht möglich. Hauptproblem lag hier darin, ein Fahrzeug zuverlässig über die Dauer einer Videosequenz zu identifizieren und so zu verfolgen.

Nachfolgende Untersuchungen sollten sich darauf stützen, eine zuverlässige Methode für das

*motion tracking* des annähernden Fahrzeuges zu finden. Auch muss eine Bezugsgröße in der realen Anwendungsumgebung identifiziert werden, um die gewonnenen Daten in Relation mit der realen Welt zu setzen. Dies ist nötig, damit die tatsächliche Geschwindigkeit und Entfernung des Fahrzeuges zum Benutzer abgeschätzt werden kann. Auch die frühere Erkennung des Fahrzeuges durch das System sollte Teil dieser zukünftigen Betrachtungen sein.

# Literatur

- [App10] APPLE INC.: *Accessibility Programming Guide for iOS*. [http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/iPhoneAccessibility/Accessibility\\_on\\_iPhone/Accessibility\\_on\\_iPhone.html](http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/iPhoneAccessibility/Accessibility_on_iPhone/Accessibility_on_iPhone.html). Version: 2010, Abruf: März 2012
- [Ber06] BERTRAM, Prof. B.: Blindheit und Sehbehinderung in Deutschland: Ursachen und Häufigkeit, veröffentlicht. In: *Der Augenarzt* 39. Jahrgang (2006), Dezember, Nr. 6
- [BK08] BRADSKI, Gary ; KAEHLER, Adrian ; LOUKIDES, Mike (Hrsg.): *Learning OpenCV*. O'Reilly Media, Inc., 2008
- [Bun] BUNDESMINISTERIUM DER JUSTIZ: *Gesetz zur Gleichstellung behinderter Menschen*. <http://www.gesetze-im-internet.de/bgg/>, Abruf: März 2012
- [Deu10] DEUTSCHER BLINDEN- UND SEHBEHINDERTENVERBAND E.V.: *Ich sehe so, wie du nicht siehst - Wie lebt man mit einer Sehbehinderung?* <http://www.dbsv.org/infothek/broschueren-und-mehr/>, 2010
- [dev] DEVELOPER.ANDROID.COM: *Designing for Accessibility | Android Developers*. <http://developer.android.com/guide/practices/design/accessibility.html>, Abruf: März 2012
- [HHD00] HARITAOGLU, Ismail ; HARWOOD, Davis ; DAVID, Larry S.: W4: Real-Time Surveillance of People and Their Activities. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (2000), August, Nr. 8, 809–830. <http://dx.doi.org/10.1109/34.868683>. – DOI 10.1109/34.868683. – ISSN 0162–8828
- [HS81] HORN, Berthold K. P. ; SCHUNCK, Brian G.: Determining Optical Flow. In: *ARTIFICIAL INTELLIGENCE* 17 (1981), S. 185–203
- [KB01] KAEWTRAKULPONG, P. ; BOWDEN, R.: *An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection*. 2001
- [Lag11] LAGANIÈRE, Robert: *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing Ltd., 2011
- [LHGT03] LI, Liyuan ; HUANG, Weimin ; GU, Irene Y. H. ; TIAN, Qi: Foreground object detection from videos containing complex background. In: *Proceedings of the eleventh ACM international conference on Multimedia*. New York, NY, USA : ACM, 2003 (MULTIMEDIA '03). – ISBN 1–58113–722–2, 2–10
- [LK81] LUCAS, Bruce D. ; KANADE, Takeo: An iterative image registration technique with an application to stereo vision. In: *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1981 (IJCAI'81), 674–679
- [SG00] STAUFFER, Chris ; GRIMSON, W. Eric L.: Learning Patterns of Activity Using Real-Time Tracking. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (2000), August, Nr. 8, 747–757. <http://dx.doi.org/10.1109/34.868677>. – DOI 10.1109/34.868677. – ISSN 0162–8828

- [Sta09] STATISTISCHES BUNDESAMT, WIESBADEN: *Bevölkerung Deutschlands bis 2060 - 12. koordinierte Bevölkerungsvorausberechnung*. [http://www.bmi.bund.de/SharedDocs/Downloads/DE/Themen/Politik\\_Gesellschaft/DemographEntwicklung/bev%C3%B6lkerungsentwicklung\\_2060.html;jsessionid=0885E84031E145EFCBC9D976BA1F5BC1.2\\_cid287?nn=102638](http://www.bmi.bund.de/SharedDocs/Downloads/DE/Themen/Politik_Gesellschaft/DemographEntwicklung/bev%C3%B6lkerungsentwicklung_2060.html;jsessionid=0885E84031E145EFCBC9D976BA1F5BC1.2_cid287?nn=102638).  
Version: 2009
- [Wix00] WIXSON, L.: Detecting Salient Motion by Accumulating Directionally-Consistent Flow. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), S. 774–780. <http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/34.868680>. – DOI <http://doi.ieeecomputersociety.org/10.1109/34.868680>. – ISSN 0162–8828