



# **Erkennung von dynamischen Objekten in Stereobilddaten**

Masterarbeit

von

Daniel Jung

31.10.2018

Betreuer:	M. Sc. Martin Herrmann
1. Prüfer:	Prof. Dr.-Ing. Klaus Dietmayer
2. Prüfer:	Prof. Dr. Frank Steiper



Hiermit versichere ich, dass ich die vorliegende Arbeit mit dem Titel

**Erkennung von dynamischen Objekten in Stereobilddaten**

bis auf die offizielle Betreuung selbstständig und ohne fremde Hilfe angefertigt habe und die benutzten Quellen und Hilfsmittel vollständig angegeben sind. Aus fremden Quellen direkt oder indirekt übernommene Gedanken sind jeweils unter Angabe der Quelle als solche kenntlich gemacht.

Ich erkläre außerdem, dass die vorliegende Arbeit entsprechend den Grundsätzen guten wissenschaftlichen Arbeitens gemäß der „Satzung der Universität Ulm zur Sicherung guter wissenschaftlicher Praxis“ erstellt wurde.

Ulm, den 31.10.2018

Daniel Jung



# Inhaltsverzeichnis

<b>1</b>	<b>Bildverarbeitung</b>	<b>1</b>
1.1	Boxfilter . . . . .	1
<b>2</b>	<b>Kalibrierung</b>	<b>3</b>
2.1	Single-Kamera-Kalibrierung . . . . .	3
2.2	Stereokalibrierung . . . . .	3
2.3	Rektifizierung . . . . .	3
<b>3</b>	<b>Background Subtraction</b>	<b>5</b>
<b>4</b>	<b>Stereoberechnung</b>	<b>7</b>
4.1	Lokale Verfahren . . . . .	7
4.2	Globale Verfahren . . . . .	7
4.3	Kombinierte Verfahren . . . . .	8
4.3.1	Semi-Global-Matching . . . . .	8
4.3.2	Pyramidenverfahren . . . . .	8
<b>5</b>	<b>Segmentierung</b>	<b>9</b>
5.1	Region Growing Algorithmus . . . . .	9
5.2	DBScan Algorithmus . . . . .	9
5.3	Kombinierte Segmentierung . . . . .	9
<b>6</b>	<b>Klassifikation</b>	<b>11</b>
6.1	K-Nearest-Neighbor Klassifikation . . . . .	11
6.2	Support Vector Machine . . . . .	11
6.3	Neuronales Netz . . . . .	11
<b>7</b>	<b>Tools und Datensätze</b>	<b>13</b>
7.1	Background Subtraction Datasets . . . . .	13
7.2	Stereo Datensätze . . . . .	14
7.3	Tools . . . . .	16
<b>8</b>	<b>Evaluation Background Subtraction</b>	<b>19</b>

<b>9 Evaluation Stereoberechnung</b>	<b>21</b>
<b>10 Fahrzeugdetektion</b>	<b>23</b>
<b>Literaturverzeichnis</b>	<b>25</b>

# 1 Bildverarbeitung

## 1.1 Boxfilter

Boxfilter Faltung

$$\tilde{I}(i, j) = I * f = \sum_{\alpha=-m}^m \sum_{\beta=-n}^n I(i - \alpha, j - \beta) \cdot f(\alpha, \beta)$$

Relevant:

$$\sum_{\alpha=-m}^m \sum_{\beta=-n}^n f(\alpha, \beta) \stackrel{!}{=} 1$$

Boxfilter - Mittelwertfilter

$$f = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\tilde{I}(i, j) = \frac{1}{9} \sum_{\alpha=-1}^1 \sum_{\beta=-1}^1 I(i - \alpha, j - \beta)$$

Veranschaulicht wird dieses Vorgehen in 1.1. Im linken Abschnitt ist das Originalbild mit einem größeren Objekt und einigen Störpixeln zu sehen. Im rechten Abschnitt ist die gefilterte Version zu sehen: Die einzelnen Störpixel im oberen Abschnitt des Bildes sind deutlich abgeschwächt. Die Grenzen des Objektes in der rechten, unteren Ecke sind jedoch ebenso verschwommen. Bei mehrfacher Anwendung eines Mittelwertfilters konvergieren alle Pixelwerte gegen den Mittelwert der Pixel des gesamten Bildes. Der Mittelwertfilter ist somit ein Tiefpassfilter und sorgt anschaulich dafür, dass Bilder verschwimmen.

Gaußfilter:

$$N(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1.1)$$

Die Variable  $\sigma$  ergibt sich aus der Standardnormalverteilung mit  $\mu = 0$ . Damit ergeben sich Filterkerne wie

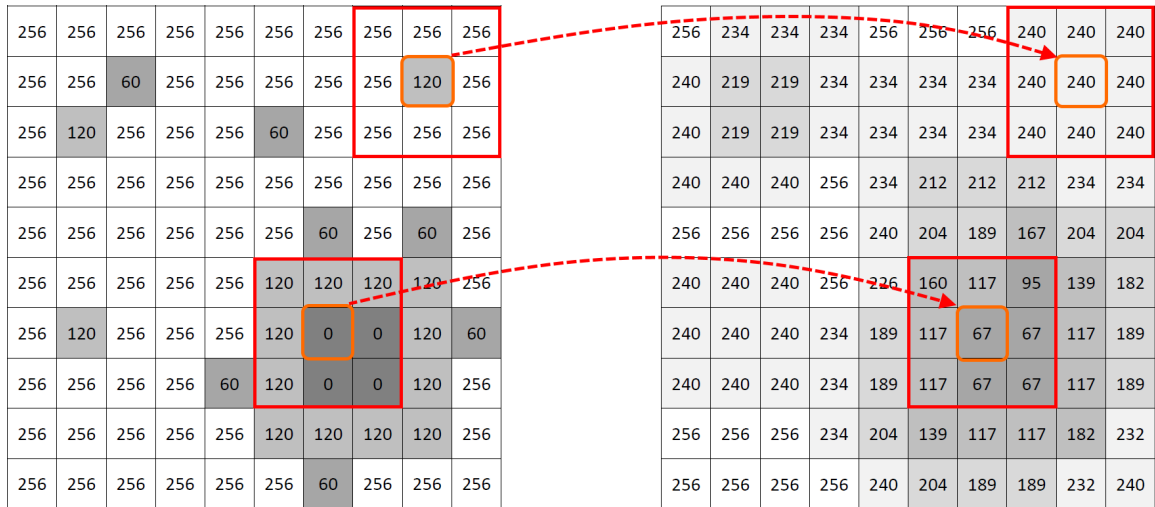


Abbildung 1.1: Beispiel für 3x3 Boxfilter

$$G_{3 \times 3} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$G_{5 \times 5} = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Sobelfilter zur Kantendetektion: Analogie zentrale Differenz:

$$f'(x) = \frac{f(x + \delta x) - f(x - \delta x)}{2\delta x}$$

$$S_x = \frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$S_u = \frac{1}{8} \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

$$S_y = \frac{1}{8} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$S_v = \frac{1}{8} \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$



## **2 Kalibrierung**

### **2.1 Single-Kamera-Kalibrierung**

### **2.2 Stereokalibrierung**

### **2.3 Rektifizierung**



# 3 Background Subtraction

Definition bzw. Problemstellung

Zwei Verfahren mit Framedifference und Tiefpassfiltern -> Problematik Rauschen



# 4 Stereoberechnung

## 4.1 Lokale Verfahren

Minimierung der Energiefunktion

$$E(d) = \sum_{(x,y)} M(x, y, d(x, y)) \quad (4.1)$$

## 4.2 Globale Verfahren

Graph Cuts, Belief Propagation, Dynamic Programming Minimierung einer globalen Energiefunktion:

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d) \quad (4.2)$$

$$E_{data}(d) = \sum_{(x,y)} M(x, y, d(x, y)) \quad (4.3)$$

$$E_{smooth} = \sum_{(x,y)} p(d(x, y) - d(x + 1, y)) + p(d(x, y) - d(x, y + 1)) \quad (4.4)$$

[SSZ18]

Sum of Absolute Differences	$\sum_{u,v}  I_1(u, v) - I_2(u + d, v) $
Sum of Squared Differences	$\sum_{u,v} (I_1(u, v) - I_2(u + d, v))^2$
Normalized Cross-Correlation	$\frac{\sum_{u,v} (I_1(u, v) - \bar{I}_1) \cdot (I_2(u + d, v) - \bar{I}_2)}{\sqrt{\sum_{u,v} (I_1(u, v) - \bar{I}_1)^2 \cdot (I_2(u + d, v) - \bar{I}_2)^2}}$
Normalized Sum of Squared Differences	$\sum_{u,v} \left( \frac{(I_1(u, v) - \bar{I}_1)}{\sqrt{\sum_{u,v} (I_1(u, v) - \bar{I}_1)^2}} - \frac{(I_2(u, v) - \bar{I}_2)}{\sqrt{\sum_{u,v} (I_2(u, v) - \bar{I}_2)^2}} \right)^2$
Rank	$\sum_{u,v} (\hat{I}_1(u, v) - \hat{I}_2(u + d, v))$ $\hat{I}_k(u, v) := \sum_{m,n} I_k(m, n) < I_k(u, v)$
Census	$\sum_{u,v} \text{HAMMING}(\hat{I}_1(u, v), \hat{I}_2(u + d, v))$ $\hat{I}_k(u, v) := \text{BITSTRING}_{m,n}(I_k(m, n) < I_k(u, v))$

[Fus18]

Boxmatching

## 4.3 Kombinierte Verfahren

### 4.3.1 Semi-Global-Matching

### 4.3.2 Pyramidenverfahren

# 5 Segmentierung

## 5.1 Region Growing Algorithmus

## 5.2 DBScan Algorithmus

## 5.3 Kombinierte Segmentierung





# **6 Klassifikation**

## **6.1 K-Nearest-Neighbor Klassifikation**

## **6.2 Support Vector Machine**

## **6.3 Neuronales Netz**



# 7 Tools und Datensätze

## 7.1 Background Subtraction Datasets

### ChangeDetection.NET

Der changedetection Datensatz ist eine im Jahr 2012 entstandene Sammlung von Videos, welche unterschiedliche Szenarien und Kameraeigenschaften darstellen. Die Motivation der Urheber ist, dass kein universeller Algorithmus existiert, welcher alle Szenarien angemessen gut auswerten kann. Im Datensatz vorhandene Szenarien sind:

Datensatz 2012:

- Dynamischer Hintergrund
- Wackelnde Kamera
- Unregelmäßige Objektbewegungen
- Schatten
- Wärmebildkameras

Datensatz 2014:

- Wettereinfluss
- Niedrige Bildrate
- Nachtaufnahmen
- Überwachungskameras
- Luftturbulenzen durch thermische Störungen

Ebenso mitgeliefert werden zu den Videos gehörige Ground-Truth-Daten, welche die tatsächliche Bewegung in den einzelnen Bildern beschreiben. Diese Daten wurden mitunter manuell erstellt. Als Vordergrund gelten unter anderem Regionen, welche nicht Teil des statischen Hintergrunds sind und bei denen es sich insbesondere um Objekte wie Personen oder Fahrzeuge handelt. Auch Objekte, die eine kurze Standzeit im Bild haben, sollen nicht als Hintergrund klassifiziert werden. Dagegen werden Störungen wie Lichtreflexion, Schatten oder Turbulenzen in der Luft nicht als Vordergrund detektiert.

## Background Model Challenge

Der Datensatz der Background Model Challenge (BMC) enthält eine Sammlung an Videos, welche ebenfalls zur Evaluation von Background-Detection-Algorithmen benutzt werden können. Hierzu ist eine Reihe von unterschiedlichen Videossequenzen vorhanden, welche verschiedene Szenen abbilden. Der Unterschied zum oben beschriebenen changedetection-Datensatz ist, dass auch synthetische Videos zur Verfügung stehen. Dies sind Videos, welche durch das Rendern einer aufgebauten 3D-Szene am Computer erstellt wurden. Der Vorteil dieses Vorgehens ist, dass die zugehörigen Ground-Truth-Daten direkt mit erzeugt werden können und keine aufwändige Nachbearbeitung nötig ist. Dafür werden jedoch in der Praxis auftretende Effekte vernachlässigt, wie beispielsweise eine sich bewegende Kamera oder Turbulenzen. Die synthetischen Videos sind daher nur bedingt nutzbar.

## 7.2 Stereo Datensätze

### Kitti Datensatz

Der Kitti Datensatz ist eine umfangreiche Sammlung von Datensätzen, welche im Zusammenhang mit autonomem Fahren verwendet werden. So sind unter anderem Datensätze vorhanden für Stereovision, Szenenfluss oder Objekt-Tracking vorhanden. Die Videodaten sowie zugehörige Ground-Truth-Daten wurden mit Hilfe eines Sensorfahrzeugs erzeugt, welches mit Lidar und GPS ausgerüstet ist. Die Sequenzen wurden sowohl innerhalb wie außerhalb von Städten aufgenommen und enthalten eine ausreichende Anzahl an Fahrzeugen und Fußgängern. Zusätzlich zu den Ground-Truth-Daten sind auch Benchmarks und Evaluationsalgorithmen vorhanden um angewendete Algorithmen vergleichbar zu machen. Die Seite bietet auch die Möglichkeit der Einreichung von Algorithmen und listet diese geordnet nach ihrer Effektivität auf.

Der Kitti Datensatz hat besondere Relevanz für diese Arbeit, da sich die Szenerie ausschließlich auf den Straßenverkehr konzentriert. Nachteilig ist, dass pro Szene nur zwei Bilder vorhanden sind und somit keine kontinuierliche Verkehrsszene abgebildet wird.

## Stereo Ego-Motion Dataset

Der Stereo Ego-Motion Datensatz enthält 494 hochauflösende Videos von vier Objekten: Eine Auto, eine Katze, ein Stuhl und ein Hund. Im Video werden diese Objekte mit der Kamera umkreist um ein Rundumbild zu erzeugen. Dazu sind ebenfalls Ground-Truth-Daten vorhanden. Mit diesem Vorgehen können komplette 3D-Meshes anhand der Stereodaten erzeugt werden. Für das einfache Benchmarking von Stereodaten ist der Datensatz auch zu verwenden, wobei die Eigenkamerabewegung nur bedingt zum Thema der Arbeit passt.

## Middlebury Stereo Dataset

Der Middlebury Datensatz enthält fünf verschiedene Datensätze mit insgesamt 71 unterschiedlichem Motiven. Enthalten sind Ground-Truth-Daten sowie ein Softwaretool zur Evaluation. Auch das Einreichen erzielter Ergebnisse ist möglich. Besonders interessant an diesem Datensatz ist, dass die Objekte unter variablen Lichtverhältnissen aufgenommen wurden. So stehen zu jedem Objekt Bilder mit vier unterschiedlichen Beleuchtungen. Dies stellt den Stereoalgorithmus zusätzlich vor Herausforderungen, da auch in der Realität und vorallem im Straßenverkehr Licht eine bedeutende Rolle spielt.

## 7.3 Tools

### OpenCV

OpenCV ist eine offene Softwarebibliothek für die Sprache C++, die auf den Bereich der maschinellen Bildverarbeitung spezialisiert ist. In OpenCV sind mehr als 2500 Algorithmen vorhanden. Am Projekt beteiligen sich laut eigenen Angaben mehr als 47000 Entwickler und die geschätzte Downloadzahl liegt bei ca. 14 Millionen [tea18]. Einen Überblick über die Kernfunktionalität der Bibliothek gibt folgende Liste [con18; Cor18]:

- Bild- und Videoverarbeitung und Darstellung
- Objekterkennung und Extraktion von Objektmerkmalen
- Kamerakalibrierung und Stereovision
- Maschinelles Lernen und Clusteringverfahren
- Eigenbewegungsschätzung
- Gesichts- und Gestenerkennung

OpenCV ist sehr effizient und wird produktiv in der Industrie eingesetzt. So setzen sowohl offene Softwareprojekte oder Startups auf OpenCV, aber auch größere Firmen wie Google, Microsoft oder Yahoo sind vertreten. Die Einsatzgebiete reichen dabei von der Verkehrsüberwachung über Roboternavigation bis hin zur Gesichtserkennung in Japan.

OpenCV kann dabei nicht nur in C++ zum Einsatz kommen: Es sind auch Schnittstellen zu Python, Java und Matlab verfügbar, wobei alle gängigen Betriebssysteme unterstützt werden.

**CUDA**

**Point Cloud Library**

**Javascript Object Notation**





## 8 Evaluation Background Subtraction



## 9 Evaluation Stereoberechnung



## 10 Fahrzeugdetektion



# Literaturverzeichnis

- [con18] contributors, Wikipedia: *Wikipedia - OpenCV*. 2018. URL: <https://de.wikipedia.org/wiki/OpenCV>.
- [Cor18] Corporation, NVIDIA: *NVidia - What is OpenCV?* 2018. URL: <https://developer.nvidia.com/opencv>.
- [Fus18] Fusiello, Andrea: *Stereo Matching: an Overview*. 2018. URL: <http://www.diegm.uniud.it/fusiello/teaching/mvg/stereo.pdf>.
- [SSZ18] Scharstein, Daniel; Szeliski, Richard und Zabih, Ramin: *A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms*. In: 2018.
- [tea18] team, OpenCV: *OpenCV - About*. 2018. URL: <https://opencv.org/about.html>.