



**TECHNISCHE  
UNIVERSITÄT  
DRESDEN**

Fakultät Informatik - Institut KI - Professur Bildverarbeitung

Bachelorarbeit

## Schattenerkennung

*Autor:*

Steffen Temme

*Sudiengang:*

BA INF

*Matrikelnummer:*

3763915

*Beginn:*

05.01.2015

*Betreuer:*

Holger Heidrich

*Verantwortlicher*

*Hochschullehrer:*

Prof. PhD Carsten Rother

*Datum:*

26. März 2015

Die Erkennung komplexer Szenen erfordert u.a. Objekte von Schatten zu unterscheiden. In der Bachelorarbeit soll der aktuelle Stand der Schattenerkennung dargestellt werden. Über einen kanten- und / oder regionenbasierten Ansatz sollen Vorschläge für Schattenregionen erzeugt und bewertet werden.

Aufgaben:

- Ergänzende Literatursuche / Einarbeitung
- Auswahl u. Umsetzung eines Algorithmus zur Übersegmentierung und Segmentbewertung in einer C++ / opencv Umgebung
- Evaluation u. Tests, Bestimmen von Vor- und Nachteilen sowie deren mathematischen bzw. algorithmischen Hintergrund
- Schlußfolgerungen, Erarbeitung von Vorschlägen für Verbesserungen

# Schattenerkennung

Steffen Temme

11.02.2015

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Anwendungsgebiete</b>	<b>3</b>
<b>3</b>	<b>Auswertung der Definition</b>	<b>3</b>
<b>4</b>	<b>Videos und Bilder</b>	<b>4</b>
4.1	Video . . . . .	4
4.1.1	Intensität . . . . .	4
4.1.2	Farbigkeit . . . . .	5
4.1.3	Geometrie . . . . .	6
4.1.4	Physikalische Eigenschaften . . . . .	7
4.1.5	Texturen . . . . .	8
4.1.6	Vergleich der Methoden . . . . .	9
4.2	Bilder . . . . .	13
4.3	Shadow detection in colour high-resolution satellite images . . . . .	13
4.4	Automatic Feature Learning for Robust Shadow Detection . . . . .	15
4.5	Learning to Recognize Shadows in Monochromatic Natural Images . . . . .	16
4.5.1	Schattenveränderliche Eigenschaften . . . . .	16
4.5.2	Schattenunveränderliche Eigenschaften . . . . .	17
4.5.3	Fast schwarze Objekte . . . . .	17
4.6	Intrinsic Images by Entropy Minimization . . . . .	18
4.7	Simple Shadow Removal . . . . .	19
<b>5</b>	<b>Praktischer Teil</b>	<b>20</b>
<b>6</b>	<b>Quellen</b>	<b>26</b>

# 1 Einleitung

Schatten begegnen uns überall und sind für uns meist eindeutig erkennbar, doch was macht einen Schatten aus, und wie kann ein Computer Schatten erkennen? Darum und um aktuelle Methoden zur Schattenerkennung soll es in dieser Bachelorarbeit gehen. Als erstes stellt sich die Frage, was Schatten eigentlich sind, da daraus eventuell einfache Methoden der Schattenerkennung gefunden werden können:

„Ein Schatten ist

- eine unbeleuchtete Fläche (eines Gegenstands, zum Beispiel eine Wand), beziehungsweise das mittels einer Lichtquelle auf ihr erzeugte Projektionsbild des im „Weg des Lichtes stehenden“ Gegenstands (zum Beispiel beim Schattenspiel). Deren Rand heißt Schattengrenze oder Hell-Dunkel-Grenze, in Astronomie und Meteorologie auch Terminator oder Tag-Nacht-Grenze.
- der unbeleuchtete Raum hinter einem beleuchteten Körper.

Man unterscheidet je nach Anzahl und Ausdehnung der Lichtquellen mehrere Arten von Schatten.“[1]

## 2 Anwendungsgebiete

Ob in bewegten oder unbewegten Bildern, Schatten sind häufig hinderlich, wenn es darum geht Objekte zu erkennen, da sie meist eine ähnliche Form wie das gesuchte Objekt besitzen, und zudem meist auch einen ausreichend starken Kontrast besitzen, um als mögliches Objekt in Betracht gezogen zu werden. Dies kann zum Beispiel beim Verfolgen eines Objektes in einem Video problematisch werden, da in diesem Fall der Schatten auch noch eine vom Objekt abhängige Bewegungsrichtung hat, aber bei unebener Projektionsfläche sprunghaft die Position ändern kann, was, sollte der Schatten als Objekt klassifiziert worden sein, zum Verlieren des Objektes führen kann.

Sollte man allerdings in der Lage sein Schatten zu erkennen, dann wäre es möglich diesen vorher zu entfernen, gesondert zu behandeln, oder sogar Informationen aus diesen zu beziehen. Von daher ist es erstrebenswert Schatten erkennen zu können.

## 3 Auswertung der Definition

Aus der oben erwähnten Definition kann man folgende Eigenschaften von Schatten herausfiltern, die zum Finden von Schatten geeignet sind:

- eine unbeleuchtete Fläche
- Projektionsbild des im „Weg des Lichtes stehenden“ Gegenstands
- Hell-Dunkel-Grenze
- der unbeleuchtete Raum hinter einem beleuchteten Körper

Der erste Punkt dieser Liste lässt sich in zwei mögliche Modelle aufteilen, indem man versucht die Schatten entweder anhand einer verringerten **Intensität** der Farbe, oder aber anhand einer anderen Intensität und einer konstanten **Farbigkeit** zu erkennen. Ersteres basiert meistens auf einem anderen Farbsysteme (z.B. Lab) als dem üblichen RGB, da im RGB Farbsysteme Intensität und Farbigkeit nicht voneinander zu trennen sind. Auch bei der zweiten Methode wird meist ein Modell genutzt, dass Intensität und Farbigkeit trennt (HSV oder normalisiertes RGB), wobei bei dieser hauptsächlich auf die Farbigkeit eingegangen wird und die Annahme genutzt wird, dass Schattenregionen geringere Intensität bei gleicher Farbigkeit haben.

Aus dem zweiten und dem letzten Punkt der Liste der Eigenschaften eines Schattens kann man schließen, dass es möglich sein muss bei bekannter **Geometrie** eines Objektes eine Maske auf dieses Objekt zu legen, um so den zugehörigen Schatten zu finden.

Der dritte Punkt impliziert, dass Schatten scharfe Abgrenzungen hat, was ermöglichen würde nur nach den Kanten eines Schattens zu suchen und den Innenraum der textbfKanten auch als Schatten anzunehmen.

Neben den aus der Definition schließbaren Ansätzen können auch noch andere gefunden werden.

## 4 Videos und Bilder

### 4.1 Video

Geht es in einem Video um Schattenerkennung meint man meist das finden von sich bewegenden Schatten, da die stationären Schatten besser mit Algorithmen für Bilder zu finden sind, die wiederum zu langsam für ein Video wären. Deshalb fokussiert man sich auf Möglichkeiten möglichst schnell sich bewegende Schatten zu finden und überlässt das finden der stationären Schatten einem Algorithmus für Bilder, sollten sie denn gesucht sein.

Diese Algorithmen gehen meist von einer Hintergrundreferenz und einer Vordergrundextraktion aus. Eine Hintergrundreferenz ist meist ein Bild in dem Der Hintergrund des Videos zu sehen ist, die Objekte die sich bewegen aber nicht vorhanden sind. Dieses Bild kann entweder vor dem betreten des Bildes durch die Objekte erzeugt werden oder während des Videos approximiert werden.

Eine Vordergrundextraktion bezeichnet einen Vorgang, bei dem Pixel im Video gesucht werden die sich im Bezug zum Hintergrund verändern.

#### 4.1.1 Intensität

Die Idee Schatten aufgrund ihrer Intensität zu finden beruht darauf, dass Schatten weniger helle Farben aufweisen. Es wird nach Pixeln gesucht, die eine geringere Intensität haben als ihr korrespondierender Pixel in der Hintergrundreferenz. Diese werden als Schatten gekennzeichnet.

Da die Annahme, dass Helligkeit und Intensität der Farbe korrespondieren nicht immer gilt, wird dieses Modell nicht alleinstehend verwendet.

### 4.1.2 Farbigkeit

Aufbauend auf die Idee der verringerten Intensität wird in diesem Modell noch zusätzlich der Umstand genutzt, dass die Farbigkeit eines Objektes sich nur geringfügig verändert. Pixel werden als Schatten gekennzeichnet, wenn sich die Intensität ausreichend verringert, die Farbigkeit aber gleich bleibt. Beispiele für dieses Verfahren werden in [3],[4] und [5] vorgestellt.

Für die meisten Videos funktioniert dieses verfahren. Es kommt zu Problemen, wenn ein Vordergrundobjekt eine ähnliche Farbe wie der Hintergrund hat. Weiterhin sorgen auch starke Helligkeitsänderungen und sehr dunkle Schatten für Probleme.

In [5] werden Schatten in Videos im Bezug zum Finden von bewegten Objekten betrachtet. Da die Schatten bei der Entscheidung, was ein sich Bewegendes Objekt ist und was nicht, stören sollen sie aufgespürt werden, damit sie bei der Entscheidung einbezogen werden können.

Es wird zuerst eine Hintergrundreferenz  $B^t(p)$  für den Zeitpunkt  $t$  definiert. Der Vektor  $p$  steht für einen Pixel zum Zeitpunkt  $t$  und enthält neben seiner Position auch die RGB Werte. Ist  $p$  zum Zeitpunkt  $t$  Teil des Hintergrunds, so entspricht  $B^t(p)$  den Werten von  $p$ . Befindet sich  $p$  allerdings auf einem bekannten Objekt, dann ist  $B^t(p)$  eine Annäherung an die Farbe des vom Objekt verdeckten Hintergrunds. Gilt auch dies nicht, so befindet sich  $p$  auf einem nicht bekannten Objekt und kann nur geschätzt werden. Hierfür dient die Formel:

$$B_s^{t+\Delta t}(p) = \underset{x_i, x_j \in S}{\operatorname{argmin}} \sum_{i=1, \dots, k}^k \operatorname{Distance}(x_i, x_j)$$

Wobei  $S$  eine Menge von Pixeln aus den Bildern  $I^t$  zu verschiedenen  $t$  und dem Pixel  $B^t(p)$  ist. Um genau zu sein besteht  $S$  aus den, in den Bildern  $I$ , zum Pixel korrespondierenden Pixeln  $I^t(p)$ .  $S$  ist also definiert als:

$$S = \{I^t(p), I^{t-\Delta t}, \dots, I^{t-n*\Delta t}\} \cup w_b * B^t(p)$$

Weiterhin ist  $\operatorname{Distance}$  die Tschebyschew-Norm.

$$\operatorname{Distance}(x_i, x_j) = \max(|x_i.c - x_j.c|)$$

mit

$$c = R, G, B.$$

Mit diesen Formeln ist man nun in der Lage eine Hintergrundreferenz für jeden Zeitpunkt  $t$  zu definieren.

$$B^{t+\Delta t}(p) = \begin{cases} B^t(p), & \text{wenn } p \text{ auf Hintergrund} \\ B_s^{t+\Delta t}(p), & \text{andernfalls} \end{cases}$$

Vergleicht man die so entstandene Hintergrundreferenz mit dem momentanen Bild, so erhält man eine Vordergrundextraktion, da nur der Vordergrund sich von der Referenz unterscheidet.

$$DB^t(p) = \text{Distance}(I^t(p), B^t(p))$$

Auf diesen Vordergrund wird der Algorithmus zur Schattenerkennung angewandt. Dieser arbeitet mit dem HSV Farbraum, da in diesem Helligkeit und Farbigkeit getrennt sind.

$$SP^t(p) = \begin{cases} 1, & \text{wenn } \alpha \leq \frac{I^t(p).V}{B^t(p).V} \leq \beta \wedge |I^t(p).S - B^t(p).S| \leq \tau_S \wedge D_H \leq \tau_H; \alpha, \beta \in [0, 1] \\ 0, & \text{andernfalls} \end{cases}$$

mit

$$D_H^t(p) = \min(|I^t(p).H - B^t(p).H|, 360 - |I^t(p).H - B^t(p).H|),$$

wobei 1 für Schattenregion steht.<sup>1</sup>

### 4.1.3 Geometrie

Sind geometrische Eigenschaften von Objekt und Schatten bekannt, so kann man wie in [6] diese nutzen. Dazu ist es nötig, dass das Objekt, mit dem Schatten, aus dem Bild herausgelöst wird, um die Kontur klar erkennen zu können.

Diese Methode ist auf strenge Bedingungen angewiesen, wie z.B. in [6], dass die Menschen in dem Video aufrecht laufen. Auch mehrere Schatten für ein Objekt würden in dieser Methode zu Problemen führen, da sie nur für eine einzige Lichtquelle ausgelegt ist.

Der Algorithmus in [6] geht davon aus, dass es nur ein Vordergrundobjekt gibt, welches ein Fußgänger ist. Von diesem Vordergrundobjekt nimmt der Algorithmus eine binäre Maske  $R$  entgegen, welche so genau wie möglich an das Objekt angepasst wird. Mit dieser Maske wird die Orientierung des Fußgängers und dessen Schatten wie folgt berechnet:

$$\theta_R = 0.5 \tan^{-1} \left( \frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{2,0}} \right)$$

mit

$$(\mu_{p,q})_R = \sum_{(x,y) \in R} (x - \bar{x})^p (y - \bar{y})^q.$$

Nachdem die Orientierung berechnet wurde wird nun nach einem Punkt auf der Kontur des Objektes gesucht, an dem die Gerade, die Schatten von Objekt trennen soll beginnt. Dies geschieht, indem an allen x-Koordinaten die Pixelschritte von oben an gezählt werden, die benötigt werden, um auf das Objekt zu treffen. Die Funktion die die Anzahl an Schritten für ein beliebiges  $x$  in einer Binärmaske  $R$  wiedergibt heie  $C_R(x)$ .

Der Startpunkt der Trennlinie ist:

$$x_R = \underset{x}{\operatorname{argmax}} |C_R(x) - C_R(x - 1)| \text{ und } y_R = C_R(x_R)$$

---

<sup>1</sup>Frei nach R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams.

Mithilfe dieses Punktes und der Orientierung kann eine Linie definiert werden, die grob Objekt und Schatten trennt. Alle Punkte des Objektes unter dieser Linie liegen gehören zum Schatten.

Nimmt man noch den obersten Punkt des Objektes  $P_T(x_T, y_T)$  hinzu kann man folgende Entscheidungsregel verwenden.

$$f(x, y) = (y - mx - c)(y_T - mx_T - c) < 0$$

$$m = \tan(\theta_R)$$

$$c = y_R - x_R \tan(\theta_R)$$

Die so gefundene Schattenregion wird in [6] danach noch mit Gauss-Modellen aufgebesert, die mit den bisher ermittelten Daten erstellt werden.<sup>2</sup>

#### 4.1.4 Physikalische Eigenschaften

Aufbauend auf der Idee die Farbigkeit zu nutzen, um Schatten zu finden wird in diesem Ansatz noch ein globales Modell hinzugefügt, das passive Beleuchtung, wie zum Beispiel durch einen strahlend blauen Himmel, berücksichtigt. Diese passive Beleuchtung wird meist von direkter Beleuchtung überstrahlt, fällt im Schatten dafür um so mehr auf, was es schwierig machen kann die Entscheidungsschwellen zu setzen. Auch die häufig getroffene Annahme, dass die Farbigkeit in Richtung Ursprung des Farbkoordinatensystems abfällt wird durch passive Beleuchtung widerlegt.

Um dem entgegen zu wirken, wird ein Modell erlernt, dass dieser passiven Beleuchtung entgegen wirkt.

Die von Jia-Bin Huang und Chu-Song Chen vorgestellte Methode in "Moving Cast Shadow Detection using Physics-based Features"([11]) ist ein Beispiel für physikbasierte Algorithmen zur Schattenerkennung in Videos. Sie basiert auf dem zwei-Lichtarten-zwei-Farben-Reflektionsmodell. In diesem Modell wird angenommen, dass es in einer Szene drei Arten von Licht gibt, die wahrgenommen werden können. Die erste Lichtart beinhaltet die Eigenschaften der Objekte, die beleuchtet werden. Die zweite Lichtart besteht aus den Oberflächeneigenschaften der Objekte, wie zum Beispiel Glanz oder Reflektion von Licht. Die dritte Lichtart ist ein ambientes Licht, das überall gleich vorhanden ist.

Diese Lichtarten lassen sich in zwei Bestandteile zerlegen, wobei dies nur bei den ersten beiden Lichtarten notwendig ist. Es gibt einerseits die Farbigkeit  $c_b$  (b für body, dem Körper des Objektes) und  $c_s$  (s für surface, die Oberfläche des Objektes), die nur von der Wellenlänge  $\lambda$  des Lichtes abhängen, und andererseits gibt es auch noch die Intensität  $m_b$  und  $m_s$  des zurückgeworfenen Lichts, die von den Austrittswinkeln  $\theta_e$  und  $\phi_e$  abhängen.

Da die ambiente Beleuchtung aus allen Richtungen gleichmäßig kommt ist sie nicht Richtungsabhängig, weshalb nur die Farbigkeit  $c_d$  interessant ist.

---

<sup>2</sup>Frei nach Shadow Elimination for Effective Moving Object Detection with Gaussian Models Chia-Jung Chang, Wen-Fong Hu, Jun-Wei Hsieh, and Yung-Sheng Chen Department of Electrical Engineering, Yuan Ze University, Taiwan, ROC [6]

Ist die Szene mit all ihren relevanten Eigenschaften bekannt, so kann das emittierte Licht nach Winkel  $\theta_e$  und  $\phi_e$  wie folgt berechnet werden:

$$I(\lambda) = c_b(\lambda)[m_b l_d(\lambda) + M_{ab}(\lambda)] + c_s(\lambda)[m_s l_d(\lambda) + M_{as}(\lambda)]$$

mit

$$M_{ax} = \int_{\theta_i, \phi_i} m_x(\theta_e, \phi_e, \theta_i, \phi_i) l_a(\theta_L, \phi_L) d\theta_i \phi_i,$$

wobei  $\theta_i$  und  $\phi_i$  die Eintrittswinkel sind und  $\theta_L$  und  $\phi_L$  zusammen die Richtung der Lichtquelle angeben.

Geht man nur von matten Oberflächen aus, dann kann man alle Terme, die mit dieser zu tun haben wegfällen lassen, was zusammen mit einem Lichtstärkeindikator  $\alpha$

$$I(\lambda) = \alpha m_b c_b(\lambda) l_d(\lambda) + c_b(\lambda) M_{ab}(\lambda)$$

ergibt.

Die Reaktion eines Kamerasensors an jedem Pixel des Sensors  $g_i$ ,  $i \in \{R, G, B\}$  kann mit

$$g_i = \alpha F_i m_b^i c_b^i l_d^i + F_i c_b^i M_{ab}^i, i \in \{R, G, B\}$$

berechnet werden.

Mit den Formeln aus dem vorherigen Abschnitt ist man in der Lage einen Vektor zu erzeugen, der unabhängig vom Hintergrund und unveränderlich in der Beleuchtungsstärke  $\alpha$  ist.

$$\hat{\gamma}_i = \frac{M_{ab}^i}{(1 - \alpha) m_b l_d^i} \left( \frac{1}{|\gamma|} \right)$$

$$|\gamma| = \frac{1}{(1 - \alpha) m_b} \sqrt{\left( \frac{M_{ab}^R}{l_d^R} \right)^2 + \left( \frac{M_{ab}^G}{l_d^G} \right)^2 + \left( \frac{M_{ab}^B}{l_d^B} \right)^2}$$

$$i \in \{R, G, B\}$$

Mit diesem Vektor ist es möglich ein GMM zu trainieren, dass nach dem Training in der Lage ist Schatten zu erkennen.<sup>3</sup>

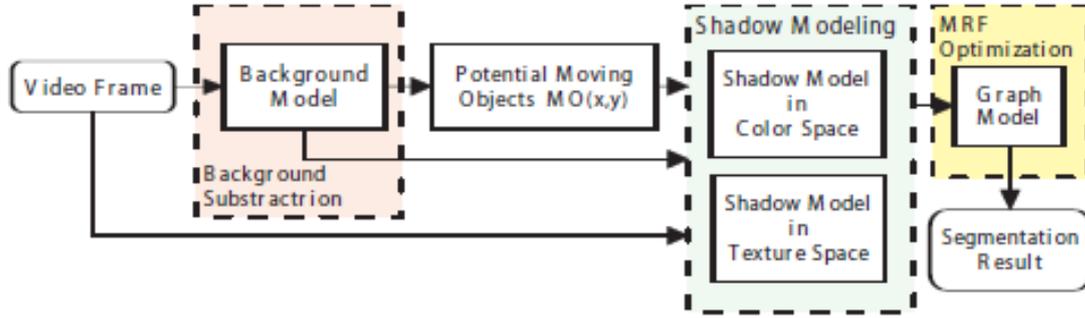
#### 4.1.5 Texturen

Da sich die Textur einer Oberfläche durch Beschattung nicht verändert, kann man Schatten anhand gleichbleibender Textur und unterschiedlicher Intensität erkennen. Je größer das für jeden Pixel betrachtete Umfeld, um so größer auch der Rechenaufwand, weshalb bei großen Flächen eine Vorauswahl getroffen werden sollte, was aber meist schon in Form einer Vordergrundextraktion geschieht.

In [7] stellen Rui Qin, Shengcai Liao, Zhen Lei und Stan Z. Li eine texturbasierte Methode vor, welche SILTP als Texturbeschreiber verwendet.

---

<sup>3</sup>frei nach [11] Absatz 2-3.1



**Bild 1** Das Flussdiagramm des vorgeschlagenen Algorithmus.

Wie zu sehen wird neben dem Texturmodell auch ein Farbmodell verwendet, um den Fall abzudecken, dass ein Vordergrundobjekt eine ähnliche Textur wie der von dem Objekt verdeckte Hintergrund hat.

Die Distanz  $D$ , der Texturen am Punkt  $(x_c, y_c)$  zum Zeitpunkt  $t$  und der Hintergrundreferenz am Punkt  $(x_r, y_r)$ , wird wie folgt berechnet:

$$Distance = |j \in \{0, \dots, 2 * N - 1\} | SILTP_{N,R}^{\tau,t}(x_c, y_c)_j \neq SILTP_{N,R}^{\tau,t}(x_r, y_r)_j |,$$

wobei  $N$  die Anzahl der Pixel ist, die gleichmäßig auf die Kante des Kreises um  $(x_c, y_c)$  mit dem Radius  $R$  verteilt sind und vom Algorithmus betrachtet werden.

$$SILTP_{N,R}^{\tau,t}(x_c, y_c) = \bigoplus_{k=0}^{N-1} s_{\tau}(I_c^t, I_k^t),$$

gibt eine Binärzahl zurück, die die Textur zum Zeitpunkt  $t$  an der Stelle  $(x_c, y_c)$  repräsentiert.  $I_c^t$  ist die Intensität an der Stelle  $(x_c, y_c)$ , während  $I_k^t$  die Intensität an der Stelle  $k$  auf dem Kreis um  $(x_c, y_c)$  angibt.

$$s_{\tau}(I_c, I_k) = \begin{cases} 01, & \text{wenn } I_k > I_c(1 + \tau) \\ 10, & \text{wenn } I_k < I_c(1 - \tau) \\ 00, & \text{andernfalls} \end{cases}$$

Wobei  $\tau$  den Grenzbereich definiert.

Mit den so erhaltenen Distanzen wird ein GMM trainiert. Dieses GMM ermöglicht zusammen mit dem Farbmodell eine erste Bewertung der Pixel, welche in ein Markov-Random-Field aufgenommen wird, um geglättet zu werden.<sup>4</sup>

#### 4.1.6 Vergleich der Methoden

In SShadow Detection: A Survey and Comparative Evaluation of Recent Methods ([2]) von Andres Sanin, Conrad Sanderson und Brian C. Lovell werden die meisten der bisher

<sup>4</sup>frei nach [7] Abschnitt 1-2.1

vorgestellten Methoden miteinander verglichen. Auch die Aufteilung in die verschiedenen Modelle ist dort wiederzufinden da sie teilweise von dort entliehen ist und sich durch die Definition von Schatten selbst anbietet.

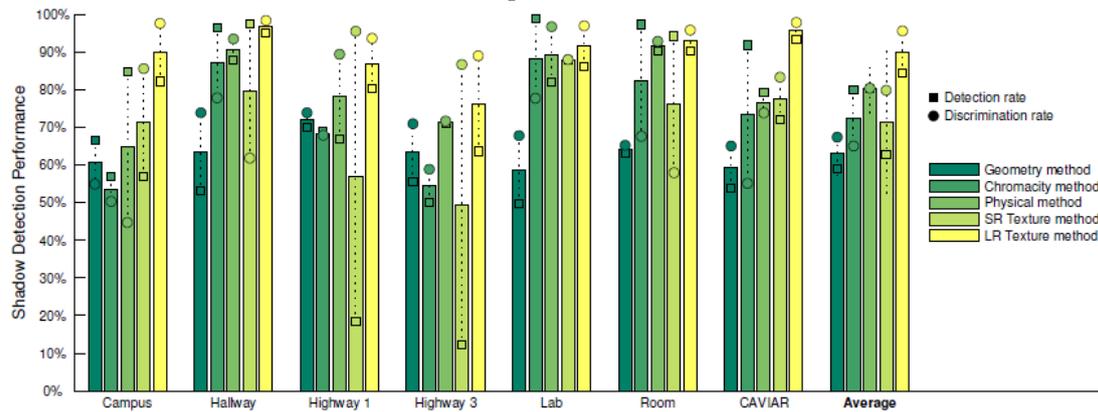
In ihrem Vergleich der Methoden werden zwei Kriterien betrachtet. Das erste Kriterium ist die Fähigkeit der Methode Schatten zu finden. Das zweite Vergleichskriterium ist, wie stark die Methoden die Leistung eines Tracking-Algorithmus verbessern.

Die Testdatensätze wurden gelabelt, indem eine Vordergrundextraktion in jedem Bild der Videos durchgeführt wurde und für die Pixel dieses Vordergrunds entschieden wurde, ob sie Objekt oder Schatten sind. Die Testdaten umfassen die Sequenzen "Campus", "Hallway", "Highway1", "Highway3", "Lab", "Room" und 25 Sequenzen aus dem CAVIAR Datensatz.<sup>5</sup> Der qualitative Vergleich wird anhand der Schattenerkennungsrate (Detection rate)  $\eta$  und der Schattenabgrenzungsrate (Discrimination rate)  $\xi$  durchgeführt. Es gilt

$$\eta = \frac{TP_S}{TP_S + FN_S}$$

$$\xi = \frac{TP_F}{TP_F + FN_F}$$

wobei  $TP$  für richtig markierte Pixel und  $FN$  für richtig nicht markierte Pixel steht.  $S$  und  $F$  stehen für Schatten und Vordergrund.



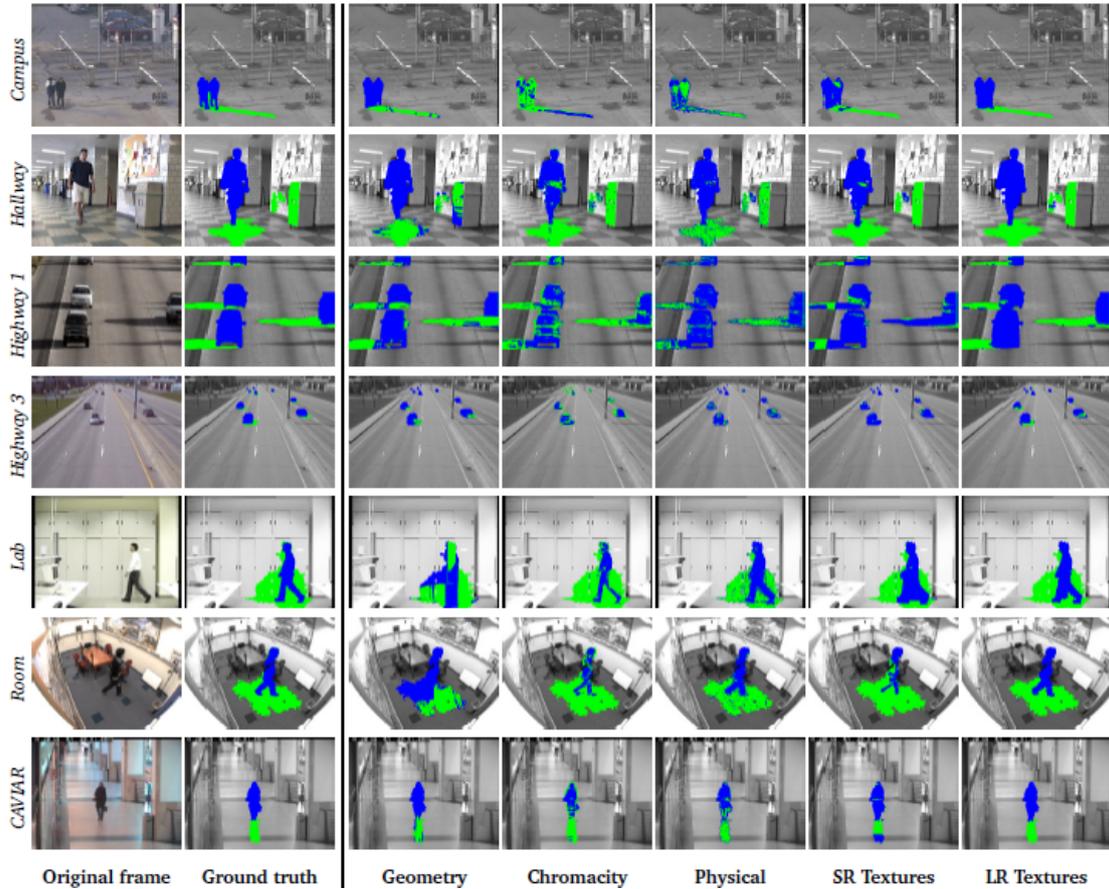
**Bild 2** zeigt Schattenerkennungs- und Schattenabgrenzungsrate der Algorithmen nebeneinander für die verschiedenen Datensätze. Die Balken stellen den Mittelwert der beiden Werte da.

Wie auf **Bild 2** zu sehen ist erzielt die texturbasierte Methode, die einen großen Texturbereich einbezieht die besten Ergebnisse, dicht gefolgt von der Methode, die physikalische Eigenschaften einbezieht. Obwohl die Methode, die kleine Texturbereiche verwendet ein ähnliches Modell wie der erfolgreichste Algorithmus in diesem Vergleich nutzt, fällt er stark zurück. Dies kann wohl darauf zurückgeführt werden, dass ein kleiner Bereich nicht immer relevante Stellen abdeckt.<sup>6</sup>

Es folgt der qualitative Vergleich mit Hilfe einer Sammlung von Bildern aus dem Paper.

<sup>5</sup>siehe [2] Table 2

<sup>6</sup>[2] Abschnitt 3.5



**Bild 3:** Beispielbilder aus jedem Datensatz der für den Vergleich verwendet wurde. Die ersten beiden Spalten enthalten das Ursprungsbild und die dazugehörigen von Hand bearbeiteten Bilder, wobei blau die Objekte markiert, während grün für Schatten steht. Die nächste Spalte zeigt die Ergebnisse des geometriebasierten Algorithmus, der wie man sieht Probleme mit breiten Schatten hat. Es folgen die Spalten für die farbigkeits- und phsikbasierten Algorithmen, die beide unter Rauscheffekten leiden und Probleme mit sehr dunklen Szenen haben, wie bei "Campus" zu sehen ist. Die letzten beiden Spalten sind den texturbasierten Algorithmen gewidmet, wobei zu erkennen ist, dass ein kleiner betrachteter Bereich zu Problemen auf texturarmen Oberflächen führt, wie zum Beispiel bei "Highway 1".<sup>7</sup>

Hier noch einen tabellarischen Überblick über die Stärken und Schwächen der Algorithmen.<sup>8</sup>

<sup>7</sup>[2] Figure 5

<sup>8</sup>siehe [2] Table 4

	Geometrie	Farbigkeit	physikalisch	kleine Texturregio- nen	große Texturregio- nen
Szenenunabhängigkeit	mittel	mittel	mittel	mittel	hoch
Objektunabhängigkeit	niedrig	hoch	hoch	hoch	hoch
Schattenunabhängigkeit	niedrig	hoch	hoch	hoch	hoch
Halbschattenerkennung	mittel	niedrig	mittel	hoch	hoch
Rauschresistenz	mittel	niedrig	mittel	hoch	hoch
Erkennungs- /Abgrenzungstauschrate	niedrig	hoch	mittel	hoch	niedrig
Rechenzeit	niedrig	niedrig	niedrig	hoch	mittel

Wobei die Bewertungskriterien wie folgt zu erstehen sind:

- Szenenunabhängigkeit: Wie unabhängig die Methode von der Szene ist, auf die sie angewandt wird. Je höher die Bewertung, desto besser.
- Objektunabhängigkeit: Wie unabhängig die Methode von dem betrachteten Vordergrundobjekt ist. Wieder gilt je höher die Bewertung, desto besser.
- Schattenunabhängigkeit: Wie unabhängig die Methode vom Schatten ist. Je höher desto besser.
- Halbschattenerkennung: Wie gut Halbschatten erkannt werden. Dieses Kriterium liefert eine gute Auskunft darüber, wie gut die Methode ist, um Schattenränder aufzuspüren. Je höher die Bewertung, desto besser.
- Rauschresistenz: Gibt die Resistenz der Methode gegenüber Rauschen in den Eingangsdaten wieder. Je höher, desto besser.
- Erkennungs-/Abgrenzungstauschrate: Ein Maß dafür, wie viel Erkennungsrate man einbüßt, wenn man die Abgrenzungsrate erhöht und umgekehrt. Je niedriger desto besser.
- Rechenzeit: Wie viel Zeit die Methoden auf dem selben Rechner brauchen, um ihre Aufgabe zu beenden. Je niedriger, desto besser.

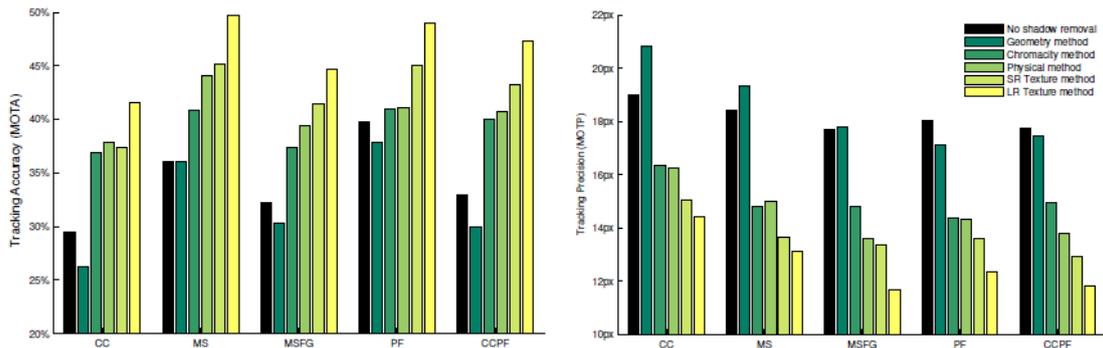
Nachdem die Fähigkeit Schatten zu finden nun verglichen wurde kommt nun das zweite Vergleichskriterium: Wie wirken sich die gefundenen Schatten auf die Leistung eines Trackingalgorithmus aus?

Diese Frage ist relevant, da nur wenige Anwendungen einen Bedarf an Pixelgenauen Schattenmasken haben, sondern meist nur den Schatten von ihren Eingabedaten entfernt haben wollen.

Die Leistung der Trackingalgorithmen wird mit Hilfe von den MOTA- und den MOTP<sup>9</sup>-

<sup>9</sup>siehe: K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the CLEAR MOT metrics. EURASIP Journal on Image Video Processing, 2008.

Werten bewertet. Der MOTA-Wert ist ein inverses Maß für die Anzahl der verlorenen Objekte. Je höher dieses Maß, desto seltener verliert der Algorithmus ein Objekt. Der MOTP-Wert zeigt dafür, wie groß die Distanz zwischen der vermuteten Position des Objektes und die tatsächliche Position ist. Je niedriger dieser Wert ist, desto genauer der Algorithmus. Diese Werte werden nur für eine Auswahl an Sequenzen aus dem CAVIAR Datenset betrachtet, da für dieses bereits Trackingwerte bekannt sind. Es werden die Tracking-Algorithmen "blob matching"(CC), zwei "mean shift tracker"(MS und MSFG), "particle filter"(PF) und "hybrid tracker"(CCPF) untersucht. Die Vielzahl von betrachteten Algorithmen rührt daher, dass die Schattendaten bei unterschiedlichen Methoden an unterschiedlichen Stellen eingebracht werden können.



**Bild 4:** Die Graphen zeigen die Leistung der verschiedenen Tracking-Algorithmen und wie diese durch Schattenerkennung beeinflusst werden. Es ist zu sehen, dass komplexere Methoden die Leistung stark erhöhen.<sup>10</sup>

Wie an **Bild 4** zu sehen lohnt es sich immer bessere Schattenerkennungsalgorithmen zu verwenden, da diese die Leistung von Tracking-Algorithmen verbessern können. Am besten schneiden die texturbasierten Algorithmen ab, wobei größere betrachtete Texturbereiche zu bevorzugen sind, da sie in allen Kategorien besser abschneiden.

## 4.2 Bilder

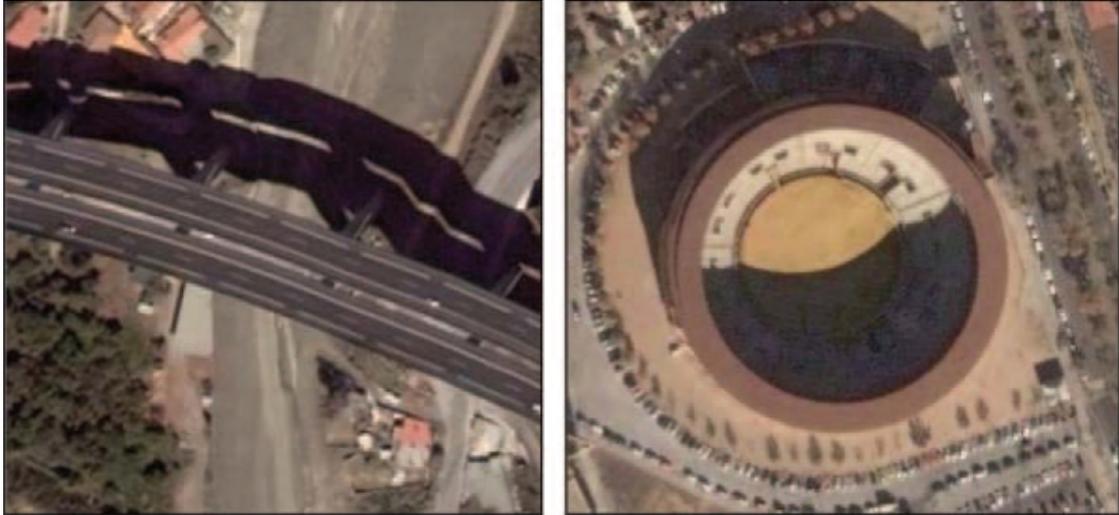
Das Auffinden von Schatten in statischen Bildern gestaltet sich schwieriger als in Videos, da generell alle Schatten gesucht werden und keine Vorauswahl getroffen wird.

Da meist mehrere Eigenschaften des Bildes verwendet werden, um die Schatten aufzuspüren, ist es nicht immer einfach die Methoden, wie bei den Videos einem Model zuzuordnen. Ich werde es anhand einiger Beispiele versuchen und auf diese Methoden näher eingehen.

## 4.3 Shadow detection in colour high-resolution satellite images

In diesem Paper von V. Are Valo, J. Gonza Lez und G. Ambrosio [13] geht es, wie im Titel schon zu erkennen, um das Erkennen von Schatten in Satellitenbildern.

<sup>10</sup>[2] Abschnitt 4.2

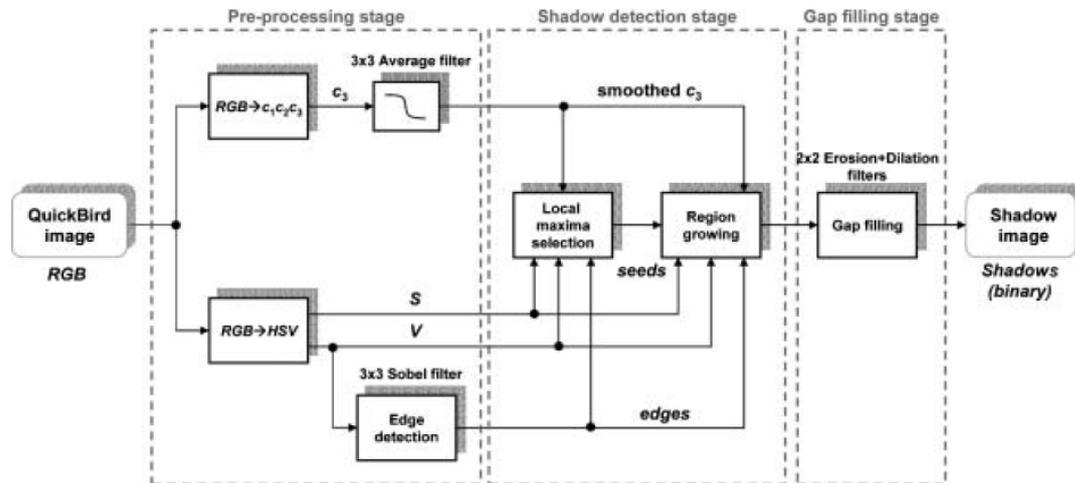


**Bild 5** zeigt zwei mögliche Szenen, in denen Schatten in Satellitenbildern zu erkennen sind.

Der vorgestellte Algorithmus verwendet einen Regiongrowing Ansatz. Im ersten Schritt wird dazu das Bild aufbereitet. Es wird ein Bild erzeugt, das nur den  $c_3$  Teil aus dem  $c_1c_2c_3$ -Farbmodell enthält, welches zusätzlich geglättet wird, da diese Komponente rauschanfällig ist. Des Weiteren wird auch eine Kopie des Bildes im HSV-Farbraum erstellt, von dem wiederum eine Kopie angelegt wird, aus der mit einem Sobel-Filter eine Karte der Gradienten erzeugt wird. Anschließend werden Saatpixel bestimmt, die ein lokales Maxima der  $c_3$  Komponente sein müssen, und innerhalb bestimmter Rahmenwerte im HSV-Raum liegen müssen, um die Schwäche der  $c_3$  Komponente in schwach gesättigten Gebieten auszugleichen.

Bei den darauffolgenden Wachstumsschritten wird eine Gauß-Verteilung der  $c_3$ -Komponenten der Schattenpixel für jeden Saatbereich mitgeführt, deren Varianz bei jedem Überprüfen von Kandidaten mit einbezogen wird. Hier wird nun auch die Gradientenkarte mit eingebracht, indem für jeden Kandidaten überprüft wird, ob dessen Gradientenwert einen bestimmten Wert überschreitet oder nicht. Überschreitet er diesen, so wird er verworfen, da er wahrscheinlich auf einer Schattengrenze liegt.

Gibt es keine Kandidaten mehr, wird noch ein Algorithmus zum Schließen möglicher Lücken in den Schattenregionen verwendet. Es folgt ein Flussdiagramm des vorgeschlagenen Algorithmus.



**Bild 6** Flussdiagramm des Algorithmus.

Wie in **Bild 6** zu sehen, verwendet diese Methode der Schattendetektion ausschließlich Farb- und Intensitätsmerkmale. Die Methode ist in das physikalische Modell einzuordnen, da die  $c_3$ -Komponente ein Maß für die bläulichkeit darstellt, welche durch die passive Beleuchtung durch die Sonne entsteht und meist nur im Schatten auffällt, was über die einfache Annahme hinaus geht, dass Farben nur dunkler werden, wie sie im Farbigkeitsmodell angenommen wird.

#### 4.4 Automatic Feature Learning for Robust Shadow Detection

Bei diesem zweiten Beispielalgorithmus von S. H. Khan, M. Bennamoun und F. Sohel, R. Togneri ist es nun definitiv nicht mehr möglich ihn in eines der oben genannten Modelle einzuordnen. Das liegt daran, dass nicht wirklich zu erkennen ist, welche Eigenschaften der Algorithmus nun genau nutzt, da er diese selber erlernt. Dies geschieht, indem zu erst Superpixel mit dem SLIC Verfahren und Grenzen mit dem gPb Verfahren extrahiert werden, von denen interessante Punkte ausgewählt werden. Mit diesen interessanten Punkten werden ConvNets, eine Spezialform eines neuronalen Netzwerkes, das an das Sehzentrum von Katzen angelehnt ist, trainiert.<sup>11</sup>

Die Nutzung von Superpixeln ermöglicht es die Anzahl der betrachteten Punkte stark zu verringern, da alle Pixel eines Superpixels ähnliche Farb- und Intensitätsmerkmale haben. Die Kanten, die durch gPb gefunden wurden, werden genutzt, um Informationen an tatsächlichen Farb- und Intensitätsübergängen zu bekommen. Denn SLIC-Superpixel haben häufig das Problem, dass auch gleiche Flächen voneinander getrennt werden, um eine einheitliche Größe der Superpixel zu gewährleisten, was dazu führt, dass ihre Kanten nicht unbedingt tatsächlichen Farbübergängen entsprechen.

Die Teilbilder für das neurale Netz werden im geometrischen Schwerpunkt der Superpixel und in regelmäßigen Abständen entlang der Kante entnommen. Diese Teilbilder werden

<sup>11</sup>siehe [8] Abschnitt 1

anschließend von den neuronalen Netzen bewertet, für die beiden Quellen von Teilbildern gibt es jeweils eines, und die durchschnittliche Wahrscheinlichkeit eines Segmentes wird für jeden Pixel des Segments eingetragen. Die Bewertungen werden zum Schluss noch in ein CRF gegeben, um diese zu glätten und konsistenter zu machen.

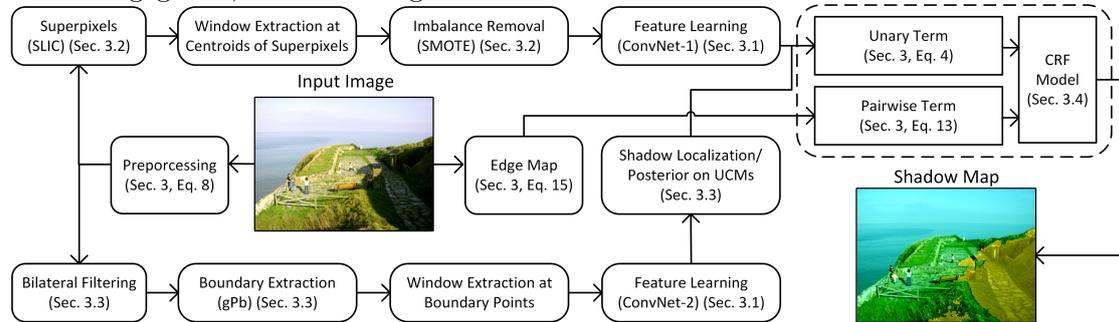


Bild 7 zeigt ein dem Paper entnommenes Flussdiagramm.

## 4.5 Learning to Recognize Shadows in Monochromatic Natural Images

Hat das Bild keine Farbinformationen, weil zum Beispiel der benutzte Sensor auf eine bestimmte Frequenz eingestellt ist, dann präsentiert dieses Werk von Jiejie Zhu, Kegan G.G. Samuel, Syed Z. Masood und Marshall F. Tappen eine Möglichkeit Schatten aufzuspüren.

Die Autoren präsentieren verschiedene Eigenschaften, die verwendet werden um Schatten aufzuspüren.

Diese Eigenschaften werden in drei Kategorien eingeteilt: Schattenveränderlich, Schattenunveränderlich und Eigenschaften, die Schatten on fast schwarzen Objekten trennen. Die Eigenschaften beziehen sich entweder auf einzelne Pixel oder aber auf homogene Segmente.

### 4.5.1 Schattenveränderliche Eigenschaften

In dieser Kategorie sind Eigenschaften eingeordnet, die durch Schatten verändert werden. darunter fallen Eigenschaften wie:

- Die Intensität, wobei sowohl die Verteilung der Werte in einzelnen Pixeln und Segmenten, als auch die Differenzen in benachbarten Pixeln und Segmenten betrachtet wird.
- Das lokale Maximum ist meist in Schattensegmenten geringer als in beleuchteten Segmenten.
- Die Ebenheit der Texturen. Texturen bleiben in Schatten zwar erhalten, verlieren aber an Intensität, was dazu führt, dass weniger von der Textur zu erkennen ist. Die Textur erscheint deswegen glatter als in einem beleuchteten Segment. Um diese Eigenschaft zu messen wird eine geglättete Version des Bildes von dem Originalbild subtrahiert und die verbleibenden Pixel werden bewertet.

- Die Symmetrie der bisher gefundenen Kurven ist, wie die Autoren festgestellt haben bei Schatten- und nicht Schattensegmenten unterschiedlich.

#### 4.5.2 Schattenunveränderliche Eigenschaften

- Gradientengleichheit: Die Verteilung der Gradienten auf einer Oberfläche wird durch den Schatten nicht verändert. Das kann erkannt werden, indem der Logarithmus der Pixelwerte gebildet wird, da dadurch die Abschwächung der Werte in den Pixeln nur noch ein additiver Offset ist. Dieser Offset verschwindet durch das Bilden der Gradienten. von diesen Gradienten kann nun für jedes Segment ein Histogramm angelegt werden, dass einen Vergleich mit anderen Segmenten erlaubt.
- Texturgleichheit: Wie bereits erwähnt werden Texturen nicht durch Schatten nicht verändert. Mithilfe einer Datenbank an erlernten Texturvorgaben können die vorgefundenen Texturen Zahlenwerten zugewiesen werden, welche dann in Form von Histogrammen verglichen werden können.

#### 4.5.3 Fast schwarze Objekte

In dieser Sektion sind Eigenschaften zu finden, die es ermöglichen sollen Schatten und schwarze beziehungsweise sehr dunkle Flächen zu unterscheiden, da diese sonst Schatten zu ähnlich sind.

- Die diskrete Entropie beschreibt, wie abwechslungsreich die Intensitäten im Umkreis eines Pixels sind. Dazu wird die Formel

$$E_i = \sum_{i \in w} -p_i * \log_2(p_i)$$

verwendet, wobei  $w$  ein 3\*3 Fenster und  $p_i$  die Wahrscheinlichkeit dafür ist, dass das entstehende Histogramm diesen Zustand bei Pixel  $i$  hat.

- Die Kantenstärke liefert eine weitere Möglichkeit um Schatten von sehr dunkelfarbigem Objekten zu unterscheiden. Dazu wird die durchschnittliche Kantenstärke in Segmenten berechnet. Ähnlich wie bei Texturen bleiben auch Kanten in Schatten erhalten, sind aber schlechter zu erkennen, ihre Kantenstärke ist also geringer. Außerdem sind auf glänzenden schwarzen Oberflächen mehr Kanten zu finden als in den Schattenflächen.



**Bild 8** enthält die vom Segmentierungsalgorithmus gefundenen Segmente rot und die Kanten weiß eingezeichnet. Es ist erkennbar, dass in dem Schatten des Autos weitaus weniger Kanten zu finden sind als auf der Autooberfläche.

All diese Eigenschaften werden in einem BCRF, mit den beiden möglichen Klassen Schatten und Nicht-Schatten verwendet.

Die einzelnen Eigenschaften können separat trainiert werden, um die Parameter für deren Bewertung zu erlernen.

#### 4.6 Intrinsic Images by Entropy Minimization

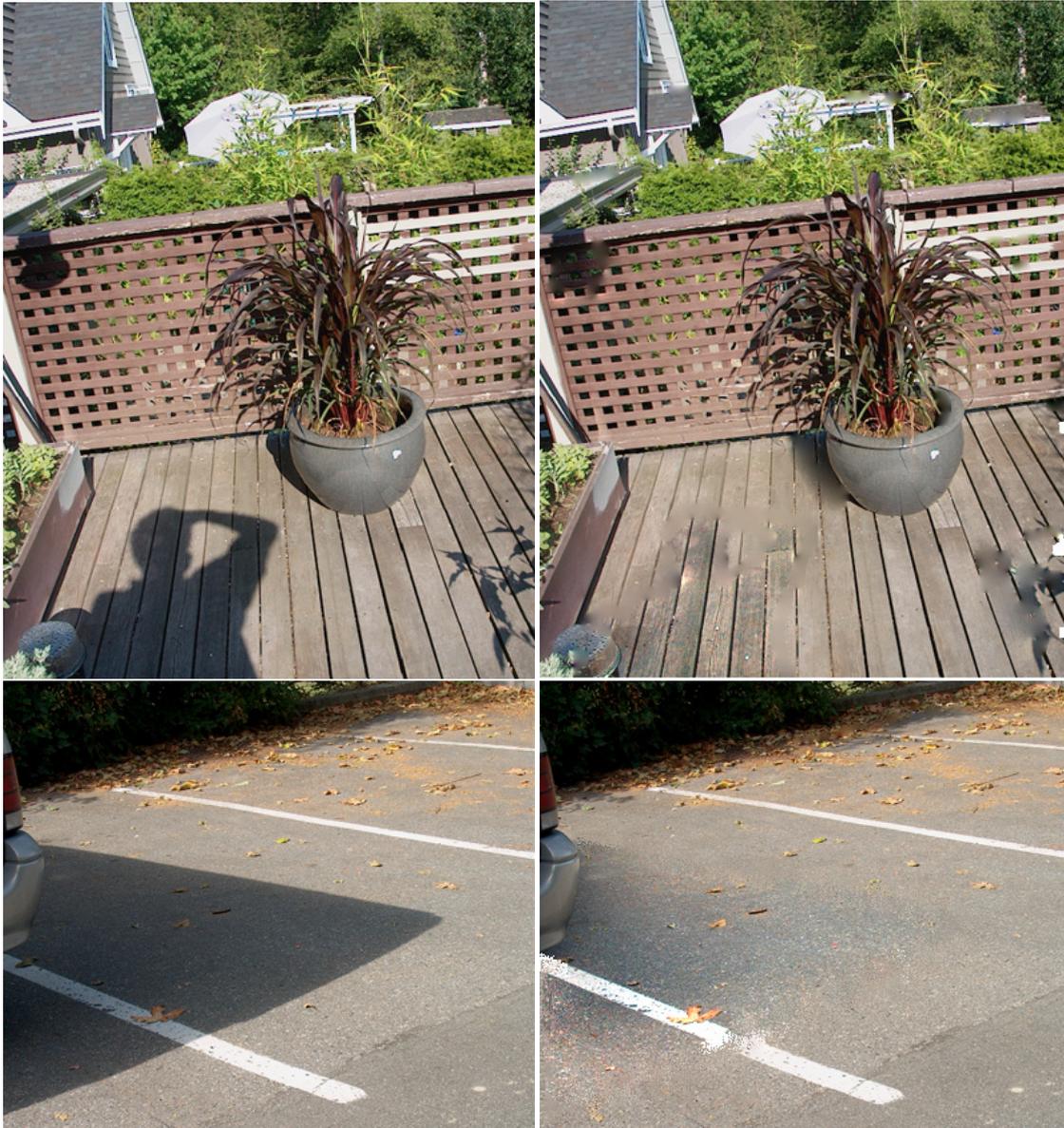
Diese von Graham D. Finlayson Mark S. Drew und Cheng Lu entwickelte Methode ermöglicht es beleuchtungsunabhängige Bilder zu erzeugen, ohne etwas über die Lichtquelle oder die Kameraeigenschaften zu wissen.

Um dies zu erreichen werden die Farbwerte des Bildes mehrfach transformiert. Im ersten Schritt werden die Werte von einem 3D-Vektor in einen 2D-Vektor umgewandelt. Dieser 2D-Vektor wird daraufhin auf eine Linie projiziert, was zu einer 1D-Darstellung führt. Die dazu benötigte Linie, beziehungsweise der Winkel dieser Linie zum Koordinatensystem bildet den Schwerpunkt der vorgestellten Methode. Im Normalfall war es so, dass dieser Winkel für jede Kamera durch eine Kalibrierung herausgefunden werden musste. Dies machte es schwierig die Methode zum Schattentfernen auf Bilder unbekannter Herkunft anzuwenden, da der Winkel nicht herauszufinden war.

Es wird jedoch eine Methode vorgestellt, mit deren Hilfe die Gerade aus den 2D-Vektoren extrahiert werden kann. Dies wird erreicht, indem alle Winkel von  $0^\circ$  bis  $180^\circ$  getestet und durch die Entropie der darauf projizierten Punkte bewertet werden. Je niedriger die beobachtete Entropie, desto besser. Gesucht wird das globale Minimum, welches auch dem Wert entspricht, welcher durch die Kalibrierung der Kamera gefunden werden würde.

Aus der 1D-Darstellung des Farbraums kann mit Hilfe der Originaldaten ein schattenfreies Bild erzeugt werden. Dazu müssen lediglich die Transformationen zum Erzeugen des invarianten Bildes umgekehrt werden. Da die Informationen über die Beleuchtungseffekte im Bild verlorengegangen sind sollten die Farbwerte des Bildes noch mit einem Wert multipliziert werden, der die Helligkeit des neuen Bildes an die des alten anpasst,

da es sonst sehr dunkel ist.



**Bild 9** Einige Beispiele von Bildern mit Schatten und ohne ihren Schatten. In dem oberen Bildpaar ist zu erkennen, dass es am Schattenrand zu einer Verwischung des Untergrundes kommt. Im unteren Bildpaar wurde der Schatten zwar größtenteils entfernt, es bleibt aber ein dunkler Fleck.

#### 4.7 Simple Shadow Removal

In diesem Paper von Clement Fredembach und Graham Finlayson werden die Probleme der Methode aus *Intrinsic Images by Entropy Minimization* angesprochen und diese Methode als Ausgangspunkt genutzt, um Schatten besser aufzuspüren und zu entfernen.

Es wird von dem Originalbild ausgegangen und von diesem ein invariantes Bild erzeugt. Von diesen beiden Bildern werden die Kanten extrahiert. Die Differenz der Kanten entspricht dann den Kanten entlang der Schatten. Diese Kanten werden dann genutzt, um den Schatten effektiver zu entfernen.

## 5 Praktischer Teil

In meinem praktischen Teil implementiere ich eine Variante des Algorithmus aus 4.3 und überprüfe seine Verwendbarkeit außerhalb von Satellitenbildern.

Der Code für diesen Algorithmus ist in der Datei *Bachelorarbeit.cpp* in der Funktion *Shadowregionfinder* zu finden. Im ersten Schritt des Algorithmus werden die Rohdaten des Bildes extrahiert, die für den Algorithmus benötigt werden. Es werden die Farbsättigungen  $S$  und die Hellwerte aus dem HSV-Farbraum erzeugt und gespeichert für weitere Verwendung. Neben diesen Werten wird auch noch der  $c3$ -Wert aus dem  $c1c2c3$ -Farbraum benötigt, der wie folgt definiert ist.

$$c1 = \arctan\left(\frac{R}{\max\{G, B\}}\right)$$

$$c2 = \arctan\left(\frac{G}{\max\{R, B\}}\right)$$

$$c3 = \arctan\left(\frac{B}{\max\{R, G\}}\right)$$

Diese Definition ist den Werken [13], wo sie für diesen Algorithmus verwendet wurde, und [12] entnommen. In [12] wurde er das erste mal definiert und für Objekterkennung in Betracht gezogen. Wie allerdings zu erkennen ist, ist diese Definition nicht für einen Computer geeignet, da es zu einer Division durch 0 kommt, wenn zwei der Farbwerte des RGB-Raums 0 sind. Unter der Annahme, dass alle RGB-Werte größer als  $-1$  sind, gehe ich davon aus, dass sich die Farbwerte von der positiven Richtung der 0 nähern. Wenn dies so ist, dann ergibt der Bruch  $+\infty$ , was wiederum bedeutet, dass der  $\arctan$  seinen Maximalwert von  $\frac{\pi}{2}$  erreicht. Die volle, von mir modifizierte Version des Farbraums, den ich hier verwende lautet also:

$$c1 = \begin{cases} \frac{\pi}{2}, & \text{wenn } \max\{G, B\} = 0 \\ \arctan\left(\frac{R}{\max\{G, B\}}\right), & \text{sonst} \end{cases}$$

$$c2 = \begin{cases} \frac{\pi}{2}, & \text{wenn } \max\{R, B\} = 0 \\ \arctan\left(\frac{G}{\max\{R, B\}}\right), & \text{sonst} \end{cases}$$

$$c3 = \begin{cases} \frac{\pi}{2}, & \text{wenn } \max\{R, G\} = 0 \\ \arctan\left(\frac{B}{\max\{R, G\}}\right), & \text{sonst} \end{cases}$$

Dieses Vorgehen ist auch im Code in der Datei *Image.cpp* zu sehen, in der alle Lese- und Schreiboperationen behandelt werden. Die Methode *C1C2C3read* beinhaltet hierbei das

auslesen der c1c2c3-Werte.



**Bild 10** Von links nach rechts und von oben nach unten: Originalbild(bereitgestellt von Christoph Sieber), Farbsättigung(S), Hellwert(V) und c3-Wert.

In **Bild 10**<sup>12</sup> sind die bisher erzeugten Bilder zu sehen. Da die c3-Komponente sehr rauschanfällig ist wird sie im nächsten Schritt des Programmes gemittelt.

Darauf folgt das extrahieren der Schattenkanten aus dem V-Bild. Dies geschieht mit einem Sobelfilter<sup>13</sup>. Ein Sobelfilter<sup>13</sup> ist ein Kantendetektor, der das Finden von Kanten in zwei Schritte aufspaltet. Im ersten Schritt werden für zwei verschiedene Richtungen auf dem Bild Kanten gefunden, deren Antwortwerte im zweiten Schritt kombiniert werden, um einen richtungsunabhängigen Wert zu erhalten. Meist werden die Matrizen  $S_x$  und  $S_y$  für diesen Filter verwendet, um  $G_x$  und  $G_y$  zu für das Originalbild  $O$  berechnen.

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ und } S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$G_x = S_x * O$$

<sup>12</sup>Die S- und V-Werte wurden mit 255 und der c3-Wert mit  $\frac{255}{2}$  multipliziert

<sup>13</sup>siehe [14] Seite 17ff.

$$G_y = S_y * O$$

Wobei  $*$  folgende Operation an allen Punkten  $(a, b)$  im Bild entspricht.

$$\begin{aligned} G_z &= O_{a-1,b-1}S_{1,1} + O_{a-1,b}S_{1,2} + O_{a-1,b+1}S_{1,3} \\ &\quad + O_{a,b-1}S_{2,1} + O_{a,b}S_{2,2} + O_{a,b+1}S_{2,3} \\ &\quad + O_{a+1,b-1}S_{3,1} + O_{a+1,b}S_{3,2} + O_{a+1,b+1}S_{3,3} \\ &\quad z \in \{x, y\} \end{aligned}$$

Die beiden Bilder  $G_x$  und  $G_y$  werden mit in dem Bild  $G$  kombiniert, wobei gilt:

$$G = \sqrt{G_x^2 + G_y^2}.$$

Dieses kombinierte Bild enthält richtungsunabhängige Informationen über die Kantestärke an jedem Punkt des Bildes.



**Bild 11** zeigt das Ergebnis des Sobelfilters. Die Werte wurden mit 10 multipliziert, um im Bild besser sichtbar zu sein.

Da nun alle Vorbedingungen erfüllt sind kann nun mit der Suche nach Saatbereichen für die Regiongrowing-Phase begonnen werden. Saatbereiche sind  $5 \times 5$  Pixel große, quadratische Bereiche im Bild, die folgende Bedingungen erfüllen:

- Das Zentrum des Bereiches ist ein lokales Minima im  $c3$ -Wert, es gibt also keinen Pixel in dem Bereich, der einen höheren  $c3$ -Wert hat.
- Alle Pixel haben einen  $c3$ -Wert, der über dem durchschnittlichen  $c3$ -Wert des Bildes liegt.
- Der V-Wert der Pixel liegt unter 0.35 (dieser Wert wurde in [13] festgelegt).
- Der S-Wert der Pixel liegt über 0.02, um die Instabilität der  $c3$ -Komponente in ungesättigten Bereichen abzufangen.
- Kein Pixel des betrachteten Bereichs ist schon Teil eines anderen Saatbereichs.

Die Saatpixel werden in der Implementierung in dem *float* Array *shadows*, das von der Größe dem Bild entspricht, mit dem Wert  $1.0f$  vermerkt. Der Wert  $1.0f$  rührt daher, dass alle Saatpixel mit 100% Wahrscheinlichkeit als Schatten angenommen werden. Die Pixel werden zusätzlich noch in die Listenstruktur *shadowlist* aufgenommen, in der jeder Saatbereich eine Liste an Pixeln zugewiesen bekommt, in der alle Pixel enthalten sind, die in diesen Saatbereich aufgenommen wurden.

Eine ähnliche zweite Liste namens *todolist* wird mit Hilfe der Saatbereiche gefüllt.

Diese Liste enthält für jeden Saatbereich eine Liste von Pixeln, die noch zur Bearbeitung ausstehen. Zur Initialisierung des Regiongrowing, das jetzt folgt wird die Liste *todolist* mit allen Pixeln gefüllt, die an einen Saatbereich anliegen.

Mit dieser initialen Füllung beginnt nun das Regiongrowing begonnen. Es wird eine Schleife begonnen, die so lange läuft, wie es in einem Durchlauf eine Veränderung an der Liste *todolist* gibt. Um diese Veränderung festzustellen wird der boolesche Wert *changed* verwendet, welcher mit *true* initialisiert wird, um die Schleife das erste Mal zu betreten und wird immer zu Beginn der Schleife auf *false* gesetzt. Der Wert wird immer dann wieder auf *true* gesetzt, wenn eine Liste in der Liste *todolist* Elemente enthält, da dieser Teil von *todolist* dann definitiv in den nächsten Schritten verändert wird.

In jedem Durchlauf der Schleife wird aus allen Elementen der Liste *todolist*, welche Listen von Pixeln sind, ein Element entnommen, sofern noch eines vorhanden ist. Für dieses Element werden folgende Bedingungen geprüft:

- Der Pixel ist nicht schon als Schatten markiert worden. Hier hilft das Array *shadows*, in dem alle Schatten mit einem Wert markiert sind.
- Der Gradient der V-Werte an dieser Position ist kleiner als 0.3. Diese Werte sind in dem Ergebnis des Sobelfilters *G* zu finden.
- Der Pixel hat einen V-Wert unter 0.35 und einen S-Wert über 0.2. Dies sind die selben Bedingungen für diese Werte, wie sie auch bei der Suche für die Saatbereiche verwendet wurden.
- Der *c3*-Wert des Pixels erfüllt folgende Bedingung:

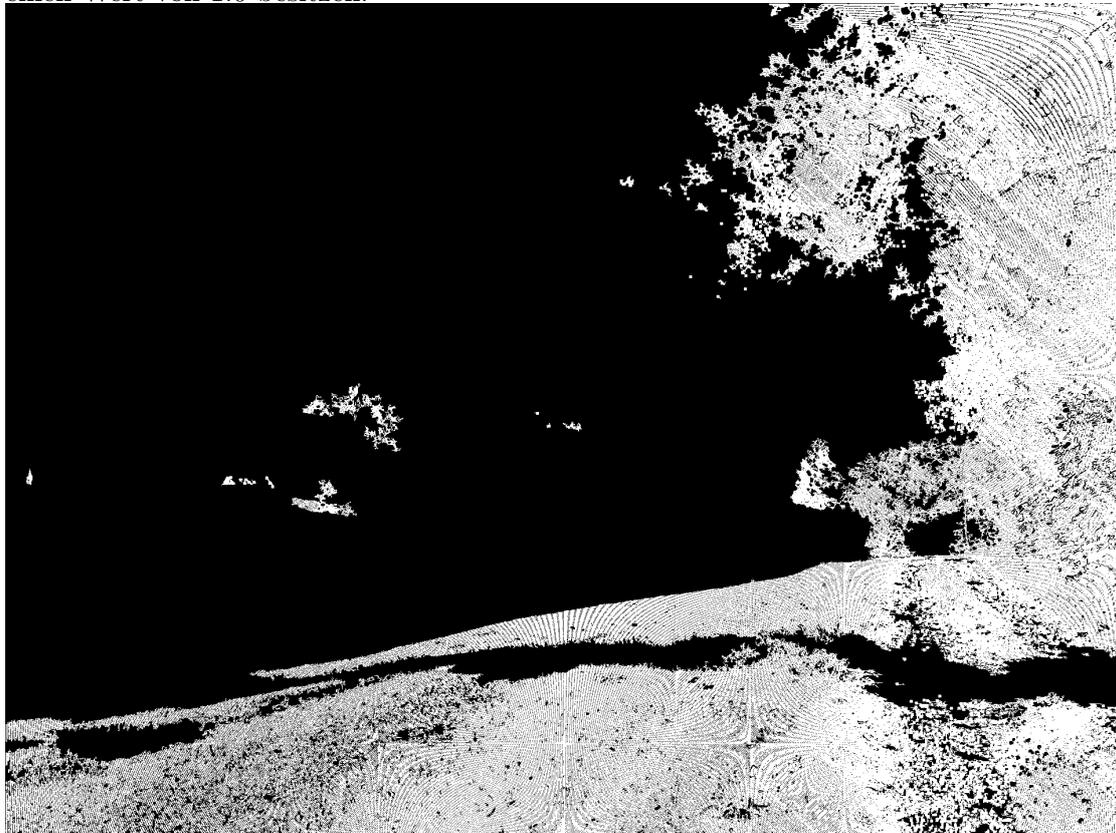
$$\frac{|c3 - \acute{c}3|}{\sigma} < 3,$$

wobei  $\sigma$  die Wurzel der Varianz der *c3*-Werte ist, die für die Region berechnet wurde.

$\acute{c}3$  ist der Mittelwert der *c3*-Werte der Pixel in dieser Region. Da diese Bedingung es erfordert, dass von allen Pixeln einer Region die *c3*-Werte gelesen werden müssen wird diese Bedingung als letzte geprüft, um dies möglichst selten zu tun.

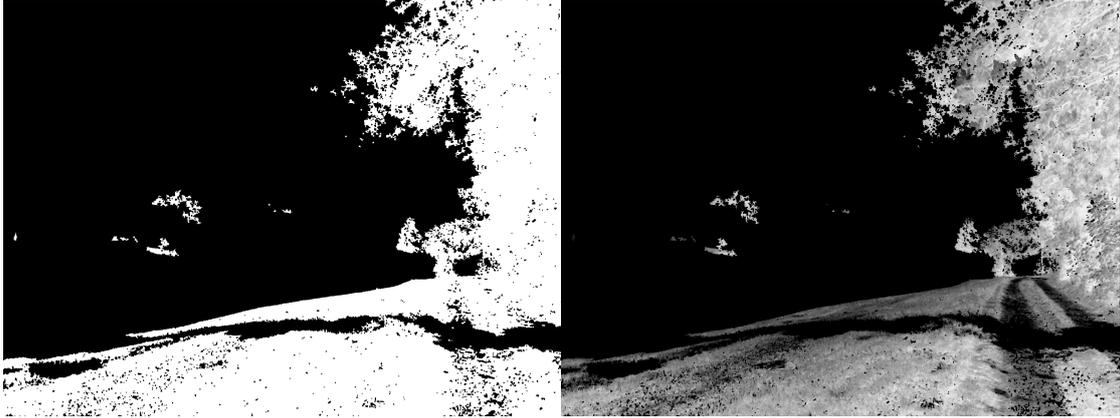
Diese Bedingungen sind auch im Code in dieser Reihenfolge zu finden. Sind sie erfüllt, dann wird ein Wert in das Array *shadow* an die entsprechende Stelle geschrieben und der Pixel wird in die dazugehörige Liste in der Liste *shadowlist* hinzugefügt. Des Weiteren werden alle benachbarten Pixel, die nicht bereits als Teil des Schattens markiert wurden in die Liste *todolist* aufgenommen.

Ist die Schleife fertig mit ihren Durchläufen, dann enthält das Array *shadow* Informationen zu allen Schatten im Bild, die der Algorithmus aufspüren konnte. Alle Nicht-Schattenpixel haben noch den initialen Wert von  $-1.0$  inne, während alle Schattenpixel einen Wert von  $1.0$  besitzen.



**Bild 11** Das Ergebnis des bisherigen Algorithmus. Alle negativen Werte werden als Schwarz, alle positiven Werte als weiß dargestellt.

Wie in **Bild 11** zu sehen ist leidet das bisherige Ergebnis an nicht richtig erkannten Einschlüssen, die meist nur ein oder zwei Pixel groß sind. Diese Einschlüsse können entfernt werden, in dem zuerst ein Dilatationsfilter und anschließend ein Erosionsfilter auf das bisherige Ergebnis angewandt wird. Für beide Filter wird ein  $2 \times 2$ -Quadrat als strukturierendes Element verwendet, so dass es ausreicht die  $3 \times 3$ -Nachbarschaft eines Pixels zu betrachten und in dieser das Maximum, beziehungsweise das Minimum der Werte zu finden, um diese Filter anzuwenden.



**Bild 12** Das Ergebnis nach dem Dilatations- und Erosionsfilter (links). Eine Variante des Programms, bei der die Graustufe vom Hellwert im Bezug zu der Schwelle für diesen Wert bestimmt wird (rechts).

Nach der Anwendung dieser Filter sind die meisten Lücken geschlossen und es verbleiben nur größere Löcher. Auch gibt der bisher für Schatten verwendete Wert von 1.0 nicht viele Informationen über die tatsächliche Schattenwahrscheinlichkeit an einer bestimmten Stelle Preis. Deswegen habe ich eine Variante programmiert, die anstelle der dieser fixen Werte einen Wert  $p$  abhängig von dem V-Wert an der Stelle des Pixels und der Schwelle für diesen Wert (0.35).

$$p = 1 - \left(\frac{V}{0.35}\right)$$

Die Werte an den Stellen, die nicht zum Schatten gehören bleiben weiterhin  $-1$ . Mit Hilfe dieser Werte ist es möglich nach Bedarf noch unwahrscheinliche Schattengebiete herauszufiltern, wie zum Beispiel das fälschlicherweise als Schatten markierte schwarze Dach im Beispielbild. Wie zu sehen hat der Algorithmus noch ein paar kleine Probleme, wie zum Beispiel schwarze und weiße Flächen, sowie zu viel Rauschen, das zu Löchern im gefundenen Schatten führt. Da der Algorithmus, indem er den c3-Wert benutzt, den Fakt ausnutzt, dass Schatten unter freiem Himmel bläulich erscheinen sind die Resultate nur bei Bildern unter freiem Himmel bei klarem Wetter gut. Da diese Bedingungen bei Satellitenbildern gegeben sind, da ansonsten auch die Schatten nicht zu sehen sind ist dieser Algorithmus hervorragend für diese geeignet. er lässt sich aber auch unter den richtigen Bedingungen hervorragend auf normale Bilder anwenden. In diesen Bedingungen spiegelt sich die Schwäche einfacher physikbasierter Algorithmen wieder, die meist nur für einen Szenentyp geeignet sind oder für jeden Szenentyp Parameter neu lernen müssen.

Ich komme also zu dem Fazit, dass diese physikbasierte Methode sich hervorragend zum finden von Schatten in Satellitenbildern eignet, wozu sie auch ursprünglich gedacht ist, aber auch für Bilder funktioniert, die vom Boden aus aufgenommen wurden, so fern bestimmte Bedingungen erfüllt sind.

## 6 Quellen

- [1] <http://de.wikipedia.org/wiki/Schatten> 26. März 2015
- [2] A. Sanin, C. Sanderson, B.C. Lovell. SShadow Detection: A Survey and Comparative Evaluation of Recent Methods. Pattern Recognition", Vol. 45, No. 4, pp. 1684–1695, 2012
- [3] E. Salvador, A. Cavallaro, and T. Ebrahimi. Cast shadow segmentation using invariant color features.
- [4] C.-T. Chen, C.-Y. Su, and W.-C. Kao. An enhanced segmentation on vision-based shadow removal for vehicle detection.
- [5] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams.
- [6] Shadow Elimination for Effective Moving Object Detection with Gaussian Models Chia-Jung Chang, Wen-Fong Hu, Jun-Wei Hsieh, and Yung-Sheng Chen Department of Electrical Engineering, Yuan Ze University, Taiwan, ROC
- [7] Moving Cast Shadow Removal Based on Local Descriptors Rui Qin, Shengcai Liao, Zhen Lei, and Stan Z. Li
- [8] Automatic Feature Learning for Robust Shadow Detection S. H. Khan, M. Bennamoun, F. Sohel, R. Togneri The University of Western Australia
- [9] Learning to Recognize Shadows in Monochromatic Natural Images Jiejie Zhu, Kegan G.G. Samuel, Syed Z. Masood, Marshall F. Tappen University of Central Florida School of Electrical Engineering and Computer Science, Orlando, FL
- [10] Intrinsic Images by Entropy Minimization, Graham D. Finlayson, Mark S. Drew, and Cheng Lu
- Bild 10** <http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/> 26. März 2015
- [11] Moving Cast Shadow Detection using Physics-based Features Jia-Bin Huang and Chu-Song Chen Institute of Information Science, Academia Sinica, Taipei, Taiwan
- [12] <http://nichol.as/papers/Gevers/Color-based%20object%20recognition.pdf> 26. März 2015
- [13] Shadow detection in colour high-resolution satellite images V. ARÉVALO, J. GONZÁLEZ and G. AMBROSIO
- [14] <https://www.inf.tu-dresden.de/content/institutes/ki/is/VORTRAG/Kantendetektoren.pdf> 26. März 2015
- Bild 15** <http://de.wikipedia.org/wiki/Datei:A-b-Farbartebene.png> 26. März 2015  
bis auf Bild 12 u.ä. wurden alle Bilder den jeweiligen Werken entnommen.