Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Christian Blank

Generierung von Tiefenbildern mittels Stereoskopie

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Christian Blank

Generierung von Tiefenbildern mittels Stereoskopie

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Birgit Wendholt
Zweitgutachter: Prof. Dr. Andreas Meisel

Eingereicht am: 13. September 2013

Christian Blank

Thema der Arbeit
Generierung von Tiefenbildern mittels Stereoskopie

Stichworte
Stereoskopie, Tiefenbild, FAST, Epipolargeometrie, Punktwolke, Korrespondenzsuche, Stereo-Vision

Kurzzusammenfassung
Gegenstand dieser Bachelorarbeit ist die Entwicklung eines Treibers zur Generierung von Tiefenbildern und Punktwolken. Das Verfahren, dass der Treiber zur Bestimmung der Tiefe nutzt, basiert auf der Stereoskopie und verwendet zur schnelleren Korrespondenzsuche Epipolargeometrie.

Christian Blank

Title of the paper
Generation of depth images through stereoscopy

Keywords
stereoscopy, depth image, FAST, epipolar geometry, depth cloud, correspondence search, StereoVision

Abstract
Subject of this thesis is the development of a driver for the generation of depth images and point clouds. The procedure that the driver uses to determine the depth, based on the stereoscopy and used for faster correspondence search epipolar geometry.

Inhaltsverzeichnis

Danksagung

An dieser Stelle möchte ich die Gelegenheit nutzen, um einigen Personen zu danken, die mich während meines gesamten Studiums und speziell bei dieser Arbeit unterstützt haben.

Mein besonderer Dank gilt Frau Prof. Dr.-Ing. Birgit Wendholt, die mich bei der Anfertigung dieser Bachelorarbeit betreut hat.

Außerdem möchte ich mich bei Herrn Prof. Dr.-Ing. Andreas Meisel für die Bereitschaft zur Erstellung des Zweitgutachtens bedanken.

Nicht zuletzt gebührt meiner Familie und meiner Freundin Dank. Sie haben mich nicht nur bei der Korrektur unterstützt, sondern standen mir auch persönlich zur Seite, wenn ich ihre Hilfe benötigt habe.

1. Einleitung

1.1. Motivation

Schon seit einigen Jahren verlagert sich die Interaktion von Menschen und Maschinen immer mehr von den alt bekannten Verfahren mit Hilfe von Maus und Tastatur zu neuen, natürlicheren Kommunikationsformen, wie etwa Gesten, Sprachsteuerung oder auch Mimik.[Bre05][AMU]

Auf Grund dieser Entwicklungen werden immer mehr Verfahren in Installationen im öffentlichen Raum eingesetzt, um diesen eine Interaktivität zu verleihen. Betrachter können sich nicht nur schöne Installationen anschauen, sie können sie sogar aktiv gestalten und mit ihnen interagieren.[O'S][Vee]

Ein beliebter Einsatzbereich von künstlerischen Installationen ist die Fassadenprojektion. [URBb][URBd] Da die Fassaden, die als eine riesige Leinwand dienen, oft zu großen und bekannten Gebäude gehören und diese Gebäude einen weitläufigen Vorplatz besitzen, sollte man die Personen, die sich vor dem Gebäude befinden, mit in die Installation einbeziehen.

Die Personen können dann durch Bewegungen oder Gesten Effekte hinzufügen, Gegenstände bewegen oder Animationen starten. Den Ideen sind dabei kaum Grenzen gesetzt.

Die Einsatzmöglichkeiten einer solchen Mensch-Maschinen-Schnittstelle sind aber nicht auf Installationen beschränkt. Andere Möglichkeiten wären die Analyse von Bewegungen oder die Steuerung von Computern.

1.2. Zielsetzung

Die Grundlage für eine Schnittstelle zur Interaktion auf diese Art und Weise sind die Tiefeninformationen einer Szene. Da ein Computer mit einer optischen Kamera nur zweidimensional sehen kann und ihm zusätzlich noch erklärt werden muss, was er eigentlich sieht, ist der erste

Schritt die Generierung einer Punktwolke, die die Punkte einer Szene im dreidimensionalen Raum beinhaltet.

Damit stehen dem Computer Daten zur Verfügung, die er für verschiedene Aufgaben verwenden kann, etwa die Erstellung eines Modells von einem Objekt oder aber die Analyse von Bewegungen, die ein Benutzer macht, um den Computer zu steuern.

Hat der Computer also die Tiefeninformationen einer Szene über eine gewisse Zeit, kann er damit Personen erkennen und Informationen - etwa Position, Bewegungsrichtung und Geschwindigkeit im Raum - über diese Personen ermitteln. Mit Hilfe dieser Informationen ist es anschließend möglich, neue Arten der Interaktion zwischen Mensch und Maschine zu entwickeln.

Im Zuge dieser Bachelorarbeit wird eine Möglichkeit gesucht, Tiefenbilder oder Punktwolken von einer Szene zu generieren und diese Daten dann zur Verfügung zu stellen, damit eine Anwendung, auf diesen Daten aufsetzend, beispielsweise die Bewegung von Usern tracken kann.

Das gesuchte System sollte sowohl in Innen- als auch in Außenbereichen ohne große Änderungen genutzt werden können. Ein wichtiges Einsatzgebiet wäre die Generierung von Tiefenbildern auf großen Plätzen, um anschließend User tracken zu können. Der zu analysierende Bereich sollte dabei kalibrierbar sein und durch Veränderung einiger Parameter von unter einem Meter bis zu 40 m reichen. Aus dem Wunsch nach Usertracking erfolgt implizit auch, dass die Bilder in einer ausreichend schnellen Folge bereitgestellt werden müssen. Hierbei werden 30 Bilder pro Sekunde als ein praktikabler Wert betrachtet. Die Bilder sollten eine VGA-ähnliche Auflösung besitzen. Somit wären sie noch ausreichend detailliert, aber immer noch akzeptabel in der Verarbeitungszeit.

1.3. Gliederung

Die Arbeit ist in sechs Kapitel gegliedert. Nach der Einleitung folgt das Kapitel Vergleichbare Arbeiten, in dem bisherige Entwicklungen vorgestellt und verglichen werden. Dabei wird zwischen Arbeiten aus der Wissenschaft und kommerziellen Entwicklungen unterschieden. Im gleichen Kapitel werden auch beispielhaft einige interaktive Installationen vorgestellt.

Im Kapitel über Grundlagen werden die wesentlichen Schritte aufgezeigt, die benötigt werden, um mittels Stereoskopie ein Tiefenbild zu generieren. Zu den beiden Hauptschritten werden jeweils mehrere Verfahren aufgezeigt.

Daran anschließend wird im Kapitel Systemkonzeption der Aufbau des entwickelten Systems erklärt und es wird aufgezeigt, welche Gründe für dieses Design sprechen. Außerdem werden Möglichkeiten gezeigt, wie das System genutzt und verändert werden kann.

In Kapitel Evaluierung werden mehrere Tests beschrieben und deren Ergebnisse präsentiert, die die Resultate des Systems qualitativ und quantitativ vergleichbar machen.

2. Vergleichbare Arbeiten

Sowohl in der Industrie als auch in wissenschaftlichen Einrichtungen ist Computer Vision ein Thema, das viel Aufmerksamkeit erhält. Gerade die Tiefeninformationen werden in vielen Anwendungsbereichen, etwa der Modellierung von Objekten, der Kartographierung der Umgebung zur Navigation oder der Abstandserkennung von Personen, benötigt. So ist es nicht verwunderlich, dass es vielfältige Lösungsansätze gibt, mit deren Hilfe man Tiefenbilder erzeugen kann.[Jac09][Teu07][Rod04][HZ00]

Für eine bessere Vergleichbarkeit der Entwicklungen werden die in der Einleitung gestellten Anforderungen (1.2 (S. 3)) verwendet. Zunächst wird eine Klassifikation erstellt, in der eine grobe Einteilung in zwei unterschiedliche Ansätze gemacht wird (2.1 (S. 5)). Anschließend werden aktuelle kommerzielle (2.2 (S. 6)) und wissenschaftliche Entwicklungen (2.3 (S. 11)) auf ihre Eignung hinsichtlich der Anforderungen untersucht und es wird gezeigt, dass die derzeitig verfügbaren Verfahren nicht die gewünschten Anforderungen erfüllen. Zudem werden einige bekannte interaktive Installationen (2.4 (S. 12)) aufgezeigt und deren Aufbau beschrieben. Die Installationen werden in Hinblick auf ihre Anforderungen untersucht und es wird geprüft, ob sich diese mit den eigenen Anforderungen überschneiden. Sollte dies der Fall sein, dann wäre die verwendete Lösung möglicherweise auch auf die vorliegende Arbeit anwendbar.

2.1. Klassifikation

Aktives Sehen Unter diesem Begriff werden Konzepte zusammengefasst, die ein irgendwie geartetes Muster auf ihre Umgebung werfen und anschließend die Differenzen zwischen Ist- und Soll-Zustand auswerten, um auf die Umgebung zu schließen.[Dyg12]

Passives Sehen Beim passiven Sehen werden Rückschlüsse auf die Umgebung allein aus den beobachteten Daten ohne Zutun von eigenen Informationen erhoben. Dabei kann man im Wesentlichen zwei Kategorien unterscheiden, Einkamera- und Mehrkamerasysteme.

2.2. Kommerzielle Entwicklungen

Im Bereich der kommerziellen Entwicklungen befindet sich bereits eine große Auswahl an Kamerasystemen, die sich zur Generierung von Tiefenbildern eignen oder sogar extra für diesen Zweck entwickelt wurden. Sie unterscheiden sich jedoch auch in vielfältiger Weise durch ihr Funktionsprinzip, aber auch durch Auflösung, Geschwindigkeit und Aufnahmebereich und den daraus resultierenden Anwendungsbereichen voneinander.

2.2.1. PointGrey Bumblebee

Die Bumblebee von PointGrey ist eine passive Stereokamera, die mithilfe einer mitgelieferten proprietären Software in der Lage ist Disparitätsbildern zu generieren. Die Software nutzt dafür den SAD-Algorithmus[1]. Es besteht auch die Möglichkeit, die Bilder aus jedem Schritt der Verarbeitung abzugreifen und somit andere Ansätze selbst zu implementieren.[Poi12]

Die Kamera ist vorkalibriert und benutzt schnelle und robuste Algorithmen, um Tiefenbilder zu erzeugen. Diese besitzen bei einer Bildrate von 48 fps eine Auflösung von 640 x 480 px. Aufgrund der festen Brennweite nimmt die Genauigkeit der Kamera mit zunehmender Entfernung ab, sodass eine sinnvolle Nutzung nur bis zu einem Abstand von 10 m gegeben ist. Das SDK der Bumblebee unterstützt offiziell nur Microsoft Windows.

Abbildung 2.1.: PointGrey Bumblebee 2 - Frontansicht

Durch die robuste und kompakte Bauweise gibt es sehr vielseitige Anwendungsgebiete für die Bumblebee. So wurden bereits Roboter mit dieser Kamera bestückt[Poib] und sie wird auch in der Medizin zur Unterstützung bei Operationen eingesetzt[Poia].

[1] Sum of Absolute Difference

Für unseren Einsatzzweck ist die Bumblebee nicht geeignet, da sie zu wenig Konfigurationsmöglichkeiten bietet. Sie besitzt beispielsweise nicht die Möglichkeit, die Optiken zu ändern und dadurch einen anderen Bildausschnitt zu erzielen. Dadurch können Entfernungen über 10 m nicht abgedeckt werden.

2.2.2. Microsoft Kinect

Aus der Kooperation zwischen Microsoft und PrimeSense entstand die Kinect. Sie wurde ursprünglich zur Körpersteuerung der Microsoft Xbox 360 entwickelt, wird aber mittlerweile in vielfältiger Weise benutzt.

Abbildung 2.2.: Microsoft Kinect für Microsoft Xbox 360

Die Kinect ist dem aktiven Sehen zuzuordnen. Sie projiziert eine Lichtpunktwolke aus Infrarotlicht auf eine Szene und berechnet mittels Triangulation die Raumkoordinate auf der der Lichtpunkt auf eine Oberfläche trifft. Dies geschieht mittels Depth From Focus und Depth From Stereo.

Die Kinect ermöglicht eine Bildfrequenz von 30 Bildern pro Sekunde bei einer Auflösung von 320 x 240 px^2. Entfernungen werden in einem Bereich von 0,8 - 5 m erkannt. Da die Kamera aktiv mit Infrarotlicht arbeitet, ist es nicht möglich, sie im direkten Sonnenlicht zu nutzen.

PrimeSense hat eine Bibliothek mit dem Namen OpenNI veröffentlicht, durch die man über die gleiche API auf verschiedene Geräte zugreifen kann, die ihrem Referenzdesigns

[2] Es gibt hierzu unterschiedliche Quellen, die diese Auflösung bestätigen, jedoch auch Quellen, die eine andere Auflösung nennen.

entsprechen. Das wären unter anderem die erste Version der Microsoft Kinect[3] und ASUS Xtion.

Der wohl wichtigste Einsatzbereich der Kinect liegt im Spielesektor. Es werden jedoch auch einige künstlerische[Vee] und wissenschaftliche Projekte[Pet13] mit ihr realisiert, da es sehr leistungsfähige Bibliotheken gibt, mit denen man in kurzer Zeit sehenswerte Ergebnisse erzielen kann. Dadurch, dass die Kinect für den Einsatz im Wohnzimmer konzipiert ist, kann man sie nur mit großen Einschränkungen in Außenbereichen einsetzen. Auch der maximale Abstand ist zu gering. Aus diesen Gründen werden zwei wichtige Anforderungen nicht erfüllt.

2.2.3. Lichtfeldkamera

Die Lichtfeldkamera oder auch plenoptische Kamera besteht aus einem hochauflösenden Sensor und einem speziellen Objektiv. Das Objektiv besitzt eine Linsenmatrix. Mit Hilfe dieser Linsen und dem Wissen über die genaue Anordnung der Linsen kann man die Länge eines Lichtstrahls berechnen. Möglich ist dies aufgrund der Tatsache, dass Lichtfeldkameras ein vierdimensionales Lichtfeld einer Szene aufnehmen. Zu der Position und der Intensität des Lichtstrahls auf dem Sensor kommt hierbei auch noch die Richtung hinzu, aus der der Lichtstrahl eingefallen ist.[Wika]

Abbildung 2.3.: Lichtfeldkamera von Raytrix mit Objektiv

Lichtfeldkameras sind aufgrund ihres vergleichsweise komplizierten Aufbaus sehr teuer. Es gibt sehr viele unterschiedliche Modelle für verschiedenste Einsatzgebiete, etwa im Sport, in der Medizin und in der Automatisierungstechnik bzw. in Qualitätskontrollen. Das ist möglich,

[3]Die zweite Version der Kinect wird von Microsoft alleine entwickelt

da die Kameras meist modular aufgebaut sind und aufgrund dessen die Brennweiten variieren können. Eine der schnellsten Lichtfeldkameras schafft eine Bildrate von 180 fps bei einer Auflösung von 2048 x 2048 px, wenn sie an einen CameraLink-Port angeschlossen ist.[Ray]

Aufgrund der hohen Anschaffungskosten und des kleinen Bildausschnitts auf größeren Entfernungen ist auch dieses System nicht für die Nutzung im Zusammenhang mit dem in der Einleitung genannten Einsatzgebiet zu verwenden.

2.2.4. TOF-Kamera

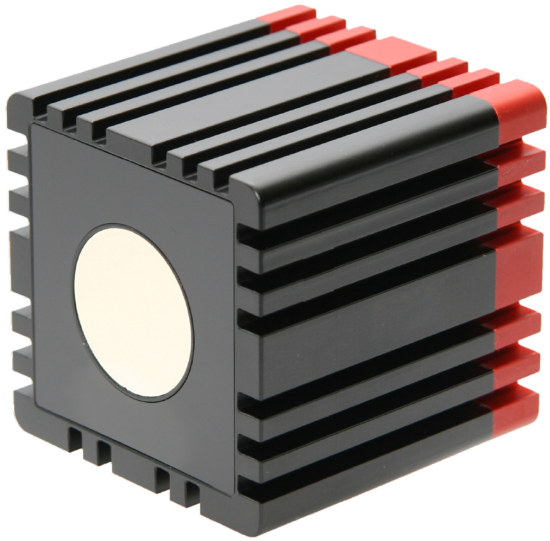

Abbildung 2.4.: MESA SR4000 - TOF-Kamera

Unter der Bezeichnung TOF-Kamera versteht man einen Kameratyp, der Distanzen mittels der Laufzeit[4] eines Lichtstrahls misst. Eine Szene wird für die Messung mit einem Lichtimpuls ausgeleuchtet und die Kamera wertet für jeden Bildpunkt aus, wie lange der Impuls von der Quelle zum Objekt und wieder zurück benötigt. Aus der Zeit kann man anschließend die Entfernung zu dem Objekt bestimmen, diese ist direkt proportional zur benötigten Zeit.[Wikb]

Kameras dieses Typs sind mit einer durchschnittlichen Bildrate von 100 fps sehr schnell und können Objekte auch noch in 40 m Entfernung vermessen. Auch bei uniformen Flächen treten

[4]Die Zeit, die der Lichtstrahl von dem Laser bis zur Oberfläche des Objektes und wieder zurück zum Lichtsensor benötigt, wird als Laufzeit bezeichnet.

keine Probleme auf, solange diese diffus sind. Im Vergleich zu Laserscannern sind sie weniger anfällig gegen Störungen, da sie keine mechanischen Bauteile besitzen.[SIC11]

Probleme können im Zusammenhang mit direktem Sonnenlicht auftreten. Da TOF-Kameras Signale im IR-Bereich senden und empfangen, kann das Sonnenlicht die Daten stark verfälschen. Zudem ist die derzeitige maximale Auflösung von 200 x 200 px recht gering und gerade Objekte in größerer Entfernung sind somit nicht mehr detailliert darstellbar.

Sowohl TOF-Kameras als auch Laserscanner werden sehr häufig in der Industrie zur Qualitätskontrolle und zur kontaktlosen Vermessung von Objekten verwendet. Die geringe Auflösung und die schlechten Ergebnisse in Außenbereichen sind jedoch zwei wichtige Gründe, diese Kameratypen nicht zu verwenden.

2.2.5. Handtracking-Devices

Zu den Handtracking-Devices zählen die NUI Group DUO und die Leap Motion Controller. Beide sind aktive 3D-Kameras, die sowohl in heller als auch in dunkler Umgebung arbeiten. Die DUO basiert auf zwei PS3 Eyes und einem Infrarotscheinwerfer. Die aufgenommen Bilder werden verarbeitet und stehen dem Nutzer anschließend zur Verfügung. Der Leap Motion Controller funktioniert auf ähnliche Weise. Die maximale Entfernung zwischen Hand und Sensor beträgt bei beiden Geräten in etwa 0,6 m und auch die Auflösung ist mit 640 x 480 px identische Geschwindigkeit von 60 fps.

Abbildung 2.5.: Gerenderte Ansicht der NUI Group DUO mit den beiden PS3 Eyes auf der Oberseite

Auch diese Geräte bieten die Möglichkeit Tiefenbilder zu generieren, jedoch eignen sie sich aufgrund der sehr geringen maximalen Entfernung zwischen Kamera und Objekt keinesfalls für den beabsichtigten Einsatzbereich.

2.2.6. Weitere Verfahren

Neben den bereits genannten Verfahren gibt es noch eine Reihe von weiteren Verfahren, die sich im Allgemeinen für die Generierung von Tiefenbildern eignen. Als zwei Vertreter wären Tiefenbildgenerierung mittels strukturiertem Licht oder Marker-basiertes Tracking zu nennen. Diese Verfahren sind jedoch meist nur stationär einsetzbar, verwenden eigene Lichtquellen, die zum Teil unverträglich für das menschliche Auge sind oder nur unter Laborbedingungen zuverlässige Ergebnisse erzielen, da sie bestimmte Anforderungen an das Umgebungslicht stellen, oder zusätzliche Hilfsmittel wie Marker.

2.2.7. Fazit

Die bisher vorgestellten Lösungen sind in der Lage Tiefenbilder zu generieren. Jedoch erreicht keine von ihnen die, in den Anforderungen gewünschte, hohe Flexibilität und Konfigurierbarkeit.

Eine Lösung basierend auf Aktivem Sehen erscheint aufgrund der Entfernung und der damit verbundenen höheren Leistung von Lichtquellen weniger praktikabel. Eine Erhöhung der Intensität könnte schädlich auf die Augen der Personen wirken, die mit dem System erfasst werden sollen.

Stereoskopie, als Vertreter der Lösungen basierend auf Passivem Sehen, hat auch einen Vorteil im Zusammenhang mit dem Sonnenlicht. Soll die Installation nur tagsüber eingesetzt werden, dann genügt das Tageslicht. Möchte man die Installation auch, aber nicht ausschließlich, nachts betreiben, dann würde sich der Einsatz von Infrarotkameras in Kombination mit einem Infrarot-Scheinwerfer eignen. Über den Tag muss dieser aufgrund des Sonnenlichts nicht betrieben werden. Man würde also nicht gegen das Sonnenlicht, sondern mit dem Sonnenlicht arbeiten.

2.3. Wissenschaftliche Arbeiten

Im wissenschaftlichen Bereich gibt es derzeit keine neuen Verfahren zur Tiefenbildgenerierung, die nicht bereits im vorangegangenen Abschnitt genannt wurden. Jedoch wurden

viele der genannten Lösungsansätze in Forschungseinrichtungen entwickelt und wurden dann, aufgrund des großen Interesses, als Produkte vermarktet. Es werden auch immer bessere Algorithmen für die einzelnen Teilaufgaben entwickelt und die bereits vorhandenen Techniken weiter verfeinert.

2.4. Interaktive Installationen

Das Ziel dieser Arbeit ist ein Kamerasystem zu finden oder zu entwickeln, welches es ermöglicht, große Installationen im Außenbereich, beispielsweise Fassadenprojektionen, interaktiv zu gestalten. In diesem Abschnitt werden vier bekannte Installationen vorgestellt, die ebenfalls eine Interaktion des Betrachters erlauben. Anschließend kann verglichen werden, inwieweit sich die Anforderungen der Installationen mit den Anforderungen aus Abschnitt 1.2 überschneiden. Es wurde keine Installation gefunden, die sich in den Anforderungen mit den eigenen überdeckt.

2.4.1. Hand from above

Diese Installation von Chris O'Shea wurde in mehreren englischen Städten, Brüssel, Tokio und Südkorea zwischen 2009 und 2010 gezeigt.[O'S]

Die Menschen auf einem Platz werden von einer Kamera aufgenommen und auf einer Großbildleinwand dargestellt. Per Zufall werden einzelne Personen ausgewählt und eine große Hand interagiert mit diesen Personen auf dem Bildschirm. So kann sie die Menschen beispielsweise stupsen, verkleinern, aufheben und wegwischen.

Abbildung 2.6.: Großbildleinwand mit interaktiver Hand

Das System basiert auf Bewegungsanalysen und Mustererkennung, wobei nur das Bild einer Kamera verwendet wird. Durch Blob-Detection werden Boundary-Boxen um die Personen gezogen und der Mittelpunkt der Boxen wird über die Zeit verfolgt. Mit diesem Aufbau ist es nicht möglich, genauere Interaktionen zu erfassen.

Abbildung 2.7.: Hand from above - Bildanalyse mit einfacher Bewegungserkennung

2.4.2. PinWall

URBANSCREEN ist bekannt für ihre Videoprojektionen unter anderem in Hamburg[URBa], Wien[URBb] und Sydney[URBd], bei denen zuvor gerenderte Sequenzen auf große Fassaden projiziert werden und so der Eindruck entsteht, dass sich die Fassade verändert.

Bei der 2007 in Bremen errichteten Installation PinWall wurde die Möglichkeit zur Interaktion des Betrachters mit der Installation hinzugefügt. Die Fassade eines Hauses wird dabei zu einem überdimensionalen Flipper. Durch die Betätigung von Tastern und Hebeln, die auf einer Bühne angebracht sind, können die Akteure die Arme des Flippers aktivieren und ihn auch schütteln.[URBc]

Im Gegensatz zur Interaktion mittels visuellen Trackingverfahren bietet die Interaktion über mechanische Komponenten den Vorteil, dass sie einfach umzusetzen und wenig störanfällig ist. Der Nachteil liegt aber in den sehr eingeschränkten Möglichkeiten, die diese Art der Interaktion bietet. Ein weiterer Nachteil ist die schlechte Erweiterbarkeit. Ist die mechanische Steuerung gebaut und installiert, dann sind Änderungen nur unter sehr großem Aufwand möglich. Möchte man im Gegensatz dazu bei einem visuellen System eine weitere Geste hinzufügen, dann kann der Wunsch recht schnell umgesetzt werden.

Abbildung 2.8.: Hausfassade mit Flipperprojektion, die von Betrachtern gesteuert werden kann

2.4.3. Stereoscope

Die 2008 von Blinken Lights entwickelte Installation in Toronto basiert auf zwei Hochhausfassaden, die sich in einem Winkel gegenüberstehen und als zwei Bildschirme fungieren. Es

werden kurze Animationen und Bilder gezeigt und die Betrachter haben die Möglichkeit, mithilfe eines Telefons Spiele auf den Bildschirmen zu spielen.[Bli08]

Hinter den 960 Fenstern der Fassaden befinden sich Diffusorfolien und LED-Lampen, die mit unterschiedlichen Intensitäten leuchten können. Aufgrund der besseren Sichtbarkeit wurde die Installation nur am Abend betrieben.

Die Animationen sind zuvor definiert und erstellt und nicht durch die Betrachter beeinflussbar. Für diese Installation wurden keine Kameras verwendet.

Abbildung 2.9.: Toronto City Hall als Fassadendisplay

2.4.4. Instant Sculpture Garden

Im Instant Sculpture Garden können Besucher bekannte Kunstfiguren nachstellen und erhalten Punkte, wenn das System sie erkennen kann. Aufgenommen werden die Akteure mithilfe einer Kinect-Kamera, die in einem Häuschen untergebracht ist, in dem sich auch der Akteur aufhält. Somit ist gewährleistet, dass kein unerwünschtes Infrarotlicht von der Kamera erfasst wird. Zudem wird der begrenzte Bereich, den die Kamera erfassen kann, gut sichtbar gemacht.[Vee]

Die Box hat aber den Nachteil, dass sie zusätzlich zur Installation bereitgestellt werden muss. Da Installationen auf freien Plätzen eines der Hauptanwendungen des Systems werden sollen, muss auch berücksichtigt werden, dass die Errichtung und Absicherung eines Objektes auf dem

Platz zusätzlichen Aufwand verursacht. Die Anforderung nach einem frei konfigurierbaren Bereich, der erfasst wird, ist ebenfalls nicht erfüllt.

Abbildung 2.10.: Instant Sculpture Garden

Die Installation konnte zwischen Mai und Juni 2013 sowohl in Wien als auch in Amsterdam genutzt werden und es bestand die Möglichkeit, gegen einen Kontrahenten aus dem jeweils anderen Land ein Spiel zu spielen.

2.4.5. Fazit

In den gezeigten Installationen kommt es zwar zu einer Interaktion zwischen Betrachter und der Installation, aber die Möglichkeiten sind sehr beschränkt. Beispielsweise kann die Installation über ein Handy gesteuert werden oder man muss Knöpfe und Hebel betätigen, um mit ihr zu interagieren. Zwei der Installationen verwenden auch ein optisches Tracking der Besucher. Hand from above nutzt jedoch nur eine einzelne Kamera und erkennt Betrachter mittels Blobdetektion, sodass nur sehr vage Annahmen über die Bewegung und die Position der Person gemacht werden können, die gerade getrackt werden soll. Instant Sculpture Garden verwendet die Kinect-Kamera für das Nutzertracking. Die Kinect ist aber, aus den in Abschnitt 2.2.2 (S. 7) genannten Gründen, nicht für eine Installation im Freien geeignet.

2.5. Zusammenfassung

Wie man sieht, gibt es viele Ansätze und Lösungen für die Generierung von Tiefenbildern, die in den vorangegangenen Abschnitten dieses Kapitels beschrieben wurden. Keine der gefundenen Lösungen erfüllt die, in der Einleitung gestellten, Anforderungen komplett.

Zum Zeitpunkt der Recherchen gab es auch keine Installation, die Anforderungen stellte, die vergleichbar mit den in Abschnitt 1.2 (S. 3) sind. Das größte Problem der bisherigen Lösungen ist die Anforderung, sowohl in geschlossenen Räumen als auch auf offenen Plätzen zu funktionieren.

Daher wird das entwickelte System wie auch die PointGrey Bumblebee und die NUI Group DUO mit Hilfe von Stereoskopie Tiefenbilder generieren. Im Gegensatz zu den beiden bereits vorgestellten Systemen wird die entwickelte Lösung aber aus separaten Kameras bestehen, um eine freie Konfiguration zu ermöglichen.

StereoVision In der vorliegenden Arbeit soll daher ein System entwickelt werden, dass den gestellten Anforderungen genügt. Die erarbeitete Lösung basiert auf einem Zwei-Kamera-System, dass mittels Stereoskopie dreidimensionale Koordinaten aus einer Szene ermittelt. Im nachfolgenden Kapitel werden die Grundlagen erläutert, auf denen die Lösung basiert. Das System wurde **StereoVision** genannt.

3. Grundlagen

In diesem Kapitel werden einige Grundlagen erläutert, die für das Verständnis dieser Arbeit erforderlich sind. Zunächst wird näher auf die Eigenheiten von Kamerasystemen und deren Kalibrierung eingegangen und anschließend werden die wichtigsten Schritte der Stereoskopie erklärt. Es werden mehrere Verfahren vorgestellt, die für diese Zwecke hilfreich sind.

3.1. Kamerakalibrierung

Im Gegensatz zu einer Lochkamera, die eine ideale Abbildung eines betrachteten Objektes liefert, kann eine Linsenkamera Objekte nicht fehlerfrei auf Bildpunkte abbilden. Diese Fehler werden radiale und tangentiale Linsenverzeichnung genannt. Da diese Fehler von Kamera zu Kamera variieren, müssen sie auch für alle verwendeten Kameras bestimmt werden.[Tet05]

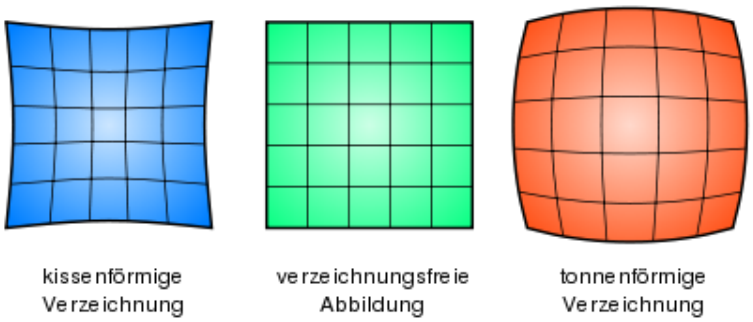

Abbildung 3.1.: Innere Kameraparameter

Zusätzlich zu den inneren Kameraparametern müssen auch die Position und die Lage der Kamera[1] im Weltkoordinatensystem ermittelt werden.

[1] Position und Lage werden als äußere Kameraparameter bezeichnet

Dies geschieht mit Hilfe eines Kalibrierkörpers, dessen markante Punkte zuvor bekannt sind. Der Vorgang wird als Kamerakalibrierung bezeichnet und ermöglicht es eine Transformationsmatrix zwischen dem 3D-Weltkoordinatensystem und dem 2D-Bildkoordinatensystem zu berechnen.

3.1.1. Rektifizierung

Oftmals sind die Kameras in der Stereoskopie so angeordnet, dass ihre optischen Achsen sich in einem Weltpunkt schneiden. Da die Kameras somit nicht parallel ausgerichtet sind, sondern zueinander gedreht, haben die Epipolarlinien der einzelnen Kameraaufnahmen einen schrägen Verlauf.[Föl12]

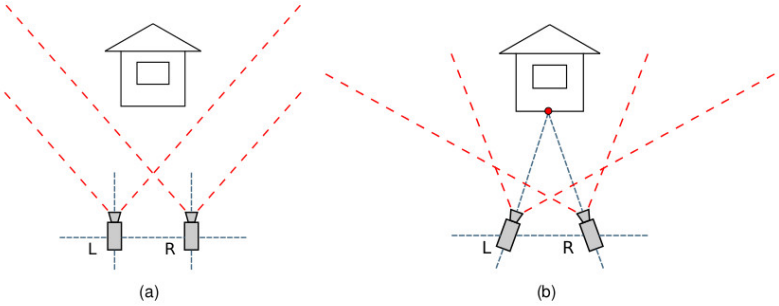

Abbildung 3.2.: Positionierung der Kameras in einem Stereokamerasystem: (a) parallel (b) zueinander gedreht

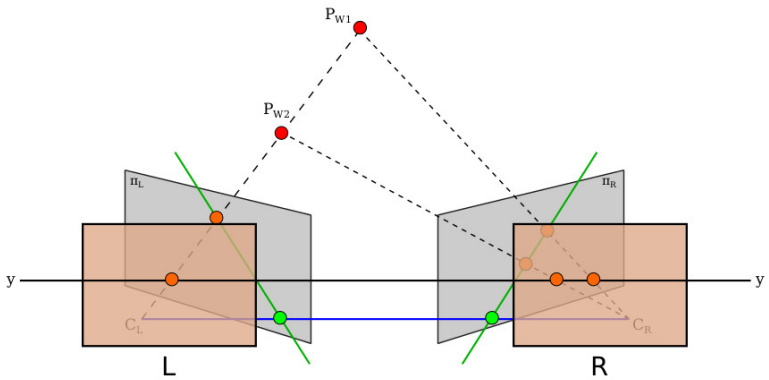

Abbildung 3.3.: Nach der Rektifizierung liegen korrespondierende Punkte in allen Bildern in der gleichen Bildzeile y.

Sind die verwendeten Kameras jedoch parallel ausgerichtet, so verlaufen die Epipolarlinien horizontal und dies vereinfacht die Suche nach Korrespondenzen erheblich. Eine exakt parallele Ausrichtung beider Kameras ist jedoch in der Praxis schwer zu erreichen und zudem wird der Bereich, in dem sich beide Kamerabilder überschneiden, verkleinert.

Durch das Wissen über die Transformationsmatrix der beiden verwendeten Kameras können die Bilder rektifiziert werden. Durch die Rektifizierung liegen alle Kameras virtuell in derselben Ebene und sind parallel ausgerichtet. Es ist eine virtuelle Drehung der Kamera. Gleichzeitig werden die zuvor genannten Linsenverzeichnungen der Kameras ausgeglichen.

3.2. Stereoskopie

Menschen können durch Stereoskopie Objekte dreidimensional wahrnehmen. Damit auch Computer ihre Umgebung räumlich erfassen, können sie sich desselben Prinzips bedienen. Was für Menschen aber ein alltäglicher Vorgang ist, über den man nicht einmal nachzudenken braucht, wird für einen Computer schnell zu einem komplizierten Vorgang, dessen Schritte in Abb 3.4 schematisch dargestellt sind.

Abbildung 3.4.: Prozess der Generierung von Punktwolken mittels Stereoskopie

Zunächst einmal wird der Input von mindestens zwei Bildquellen vorverarbeitet[2]. Die wesentlichen Punkte dazu wurden im letzten Abschnitt 3.1 (S. 18) behandelt. Im nächsten Schritt werden aus beiden Bildern Merkmale extrahiert. Welche Verfahren dort genutzt werden können und welche Gründe dieser Arbeitsschritt hat, wird im Abschnitt 3.2.1 (S. 21) geklärt.

Nachdem die Merkmale extrahiert wurden, muss innerhalb der Merkmale nach Korrespondenzen gesucht werden, beschrieben in Abschnitt 3.2.2 (S. 23). Die Korrespondenzsuche ist der rechen- und zeitintensivste Arbeitsschritt im gesamten Prozess.

[2]Der Einfachheit halber wird im weiteren Verlauf nur auf die Möglichkeiten von zwei Bildquellen eingegangen, wenn nicht anders erwähnt.

Anschließend kann nach Disparitäten[3] in den Merkmalen gesucht werden, die in beiden Bildern vorkommen. Anhand der Unterschiede kann anschließend eine Punktwolke oder eine Tiefenkarte berechnet werden 3.2.3 (S. 26).

3.2.1. Merkmalsextraktion

Am Anfang erscheint es als ein Mehraufwand, in beiden Bildern Merkmale zu suchen und dann zu vergleichen. Mit dem zusätzlichen Aufwand, der in die Merkmalsextraktion investiert wird, kann die Korrespondenzsuche jedoch erheblich vereinfacht werden.

Im wesentlichen sind Bildmerkmale, auch Keypoints genannt, Punkte in einem Bild, die sich von ihrer Umgebung unterscheiden. Die Menge aller Merkmale eines Bildes ist eine Untermenge der Menge an Bildpunkten eines Bildes und kann das Bild oder zumindest Teile des Bildes beschreiben.

Aufgrund dessen ist es möglich, mit einer viel geringeren Anzahl an Punkten zu arbeiten. Dadurch verringert sich die Rechenzeit für die Korrespondenzsuche, weil nicht mehr jeder Bildpunkt des einen Bildes mit einem Punkt des anderen Bildes verglichen werden muss.

Es gibt viele Algorithmen zur Merkmalsextraktion aus Bildern. Eine Reihe von sehr verbreiteten Algorithmen wird nachfolgend vorgestellt.[AKB08][TM08][ZN09]

SIFT

Die skaleninvariante Merkmalstransformation ist ein Algorithmus zur Extraktion lokaler Bildmerkmale aus Abbildungen. Dabei werden von einem Bild unterschiedliche Skalierungen erstellt und anschließend mehrfach mit Gaußfiltern geglättet. Die Bilder werden anschließend subtrahiert. Es bleiben markante Objektpunkte übrig. Markant sind Objektpunkte, deren Eigenschaften sich von ihrem Hintergrund unterscheiden.[RRKB11]

Extrahierte Merkmale sind unempfindlich gegenüber Translation, Rotation und Skalierung und zum Teil auch gegen Bildrauschen und projektive Abbildungen. Beschrieben und identifiziert werden die Merkmale durch einen 128-dimensionalen Vektor.

[3] Die Disparität ist der Unterschied zwischen den x-Koordinaten zweier korrespondierender Bildpunkte. Je näher sich dabei ein betrachteter Punkt im Raum an der Kamera befindet, desto größer ist die Disparität.

Der Vektor enthält dabei Gradientenwerte und -orientierungen von 16 Sektoren um den Keypoint herum. Jeder Sektor besteht dabei wiederum aus 16 Pixeln. Dabei werden die Orientierungen in acht Bereiche unterteilt und anschließend werden die Werte zu dem zugehörigen Bereich in einem Vektor addiert. Der hinzu addierte Wert ist nicht nur von dem Gradientenwert, sondern auch von der Entfernung des Sektors zum Keypoint abhängig.

SIFT findet sehr viele Merkmale in einem Bild, benötigt aber auch einen sehr hohen Rechenaufwand.

MSER

Der auf Blob-Erkennung[4] basierende Algorithmus zum Finden von Maximal-stabilen Extrema markiert Extrema mit Hilfe von Schwellenwerten. Alle Pixel in einem Bild, die unter einem Schwellenwert liegen, werden schwarz markiert, alle die gleich oder darüber liegen, weiß. Startet man mit einem Bild und markiert alle Pixel aufgrund dieses Bildes weiß und lässt anschließend eine Folge von Bildern ablaufen, dann werden schwarz markierte Bereiche entstehen und wachsen. Die Menge an verbundenen Regionen ist die Menge der Extrema.[SUKS12]

Die gefundenen Regionen sind invariant zu affinen Abbildungen, kovariant zu nachbarschaftserhaltenden Transformationen und stabil. Sie enthalten nur die Größe und den Centroid als Information.

STAR

Dieser Detektor ist eine Abwandlung des CenSurE-Detektors[5], der im Gegensatz zu CenSurE zwei Quadrate, die um 45 Grad zueinander gedreht sind, nutzt, um einen Kreis zu approximieren. Es wird ebenfalls eine zweistufige Approximation des „Laplacian of Gaussians“-Filters verwendet.

Den Entwicklern von STAR war es wichtig, einen Multiskalendetektor mit voller Raumauflösung zu erstellen. Durch die zuvor genannte Approximierung der Kreismaske ist es möglich, Integralbilder zu verwenden, die eine effiziente Berechnung ermöglichen, und gleichzeitig die Invarianz gegenüber Rotation zu erhalten.[SUKS12]

[4] Blob-Erkennung ermöglicht es, Bereiche in digitalen Bildern zu finden, die sich gegenüber ihrer Umgebung unterscheiden.
[5] Center Surrounded Extrema

Die Merkmale werden durch eine 64-dimensionalen Vektor dargestellt. Die Werte des Vektors werden durch Aufsummierung Haar-Wavelets mithilfe von Integralbildern berechnet.

ORB

Der auf der ICCV[6] 2011 von Rublee et al. vorgestellte Detektor ORB[7] ist eine Kombination aus dem Merkmalsextraktor FAST und dem Merkmalsdiskriptor BRIEF. Die von ihm produzierten Ergebnisse sind invariant gegenüber Drehung und rauschresistent, jedoch nicht skalierungsinvariant.[RRKB11]

Im Vergleich zu SIFT arbeitet ORB um zwei Größenordnungen schneller und produziert dabei vergleichbar zuverlässige Ergebnisse. Somit ist er auch für den Einsatz in Echtzeitsystemen und im mobilen Bereich geeignet.

Im Gegensatz zu SIFT und auch SURF fällt ORB unter die BSD-Lizenz und ist deshalb frei zugänglich.

FAST

Der FAST-Algorithmus (Features from Accelerated Segment Test) ist ein Corner-Detection-Algorithmus[8], der auf AST basiert. Er gehört zu den in der Berechnung effizientesten Featuredetektoren und produziert sehr stabile Features.[TH98]

Innerhalb eines zuvor definierten Radius um den Kern herum wird überprüft, ob n zusammenhängende Pixel auf der äußeren Kreisbahn alle unterhalb oder oberhalb eines Grenzwertes liegen. Sollte einer dieser beiden Fälle zutreffen, dann ist der Kern ein Feature.

FAST nutzt für den Radius nur einen Wert von 3 Pixeln, woraus sich ein Umfang von 16 Pixeln ergibt. Tests haben gezeigt, dass die besten Resultate mit n gleich 9 erzielt wurden.

3.2.2. Korrespondenzsuche

Korrespondenzsuche kann grob in zwei Kategorien eingeteilt werden, globale und lokale Ansätze. Lokale Ansätze betrachten immer nur einen kleinen Ausschnitt eines Bildes und

[6] International Conference on Computer Vision
[7] Oriented FAST und Rotated BRIEF
[8] Ein Bildausschnitt gilt dabei als Ecke, wenn man ihn nicht beliebig verschieben kann, ohne eine Änderung des Inhaltes zu erhalten.

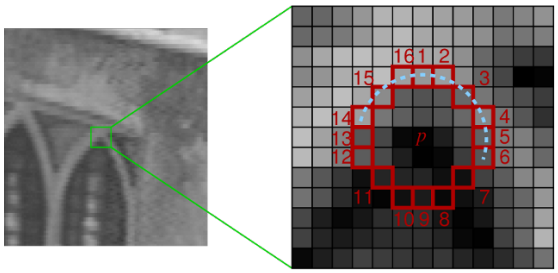

Abbildung 3.5.: Überprüfung eines Bildausschnittes mit Hilfe des FAST-Algorithmus auf das Vorhandensein eines Features

berechnen darauf das Minimum. Globale Ansätze benutzen die gesamte Punktmenge eines Bildes für die Suche.[Föl12]

In diesem Abschnitt werden zunächst zwei einfachere Matcher vorgestellt. Matcher sind Verfahren, die versuchen Korrespondenzen zu finden. Anschließend wird eine bessere Möglichkeit gezeigt, die den Suchraum reduziert.

Die Suche entspricht einer Nächster-Nachbar-Suche[9] in einem zweidimensionalen euklidischen Raum. Das Problem der NNS kann folgendermaßen definiert werden: Gegeben ist eine Menge an Punkten P = $\{p_1, ..., p_n\}$ in einem Vektorraum X. Die Menge P muss nun so vorverarbeitet werden, dass für einen neuen Punkt $q \in X$ die Suche nach seinem Nachbarn in P effizient ist.[Wik13a]

Brute-Force-Matcher

Eine aus der Sicht der Implementierung einfache Lösung ist es, jedes Merkmal des einen Bildes mit allen Merkmalen des anderen Bildes zu vergleichen. Die Komplexität dieses Verfahrens beläuft sich jedoch auf $O(n^2)$. Daraus folgt, dass der Rechenaufwand sehr groß ist. Ein Vorteil dieser Methode wäre aber der globale Ansatz. Man könnte durch Vergleiche das globale Minima berechnen und hat somit weniger Fehler bei den gefundenen Korrespondenzen[10].

FLANN-basierter Matcher

Eine weitere Möglichkeit besteht darin, den approximierten nächsten Nachbarn zu suchen. Dadurch verringert man die Suche, gerade in hochdimensionalen Suchräumen erheblich, man

[9] nearest neighbor search, NNS
[10] auch als False positives bezeichnet

geht aber auch das Risiko ein, dass ein gefundener Nachbar nicht zwingend der nächste Nachbar in der Suchmenge ist. Es wird über die Suchmenge ein kd-Baum aufgebaut und anschließend wird in diesem Baum gesucht. Die Komplexität der Suche beträgt im Durchschnitt nur O(log(n)) und dadurch liegt das gesamte Verfahren bei O(n*log(n)).

Eine Einsparung in der Rechenzeit kommt aber erst bei großen Datensätzen mit über einer Million Daten zum Tragen. Bei kleinen Mengen ist die Suche nach dem approximierten nächsten Nachbarn in den meisten Fällen gleich schnell oder sogar langsamer als Brute Force.

Für diese Art von Suche gibt es mehrere Algorithmen[KOR00][AMN+98], die jedoch schwierig zu vergleichen und jeweils für bestimmte Einsatzzwecke optimiert sind. Zudem müssen die Parameter korrekt gewählt werden, da sie einen großen Einfluss auf das spätere Ergebnis haben. Um den möglichst besten Algorithmus zu wählen, bietet sich FLANN an.

FLANN ist eine Bibliothek für die Suche von Nachbarn in mehrdimensionalen Räumen, die einem bei der Suche des besten Verfahrens unterstützt. Auf Grundlage der übergebenen Daten ermittelt sie den passenden Algorithmus und die besten Parameter. Die Auswahl eines Algorithmus ist rechenaufwändig und sollte nur einmal vor Beginn der eigentlichen Rechnungen benutzt werden. Da es das Ziel dieser Arbeit ist, ein System zu konzipieren, dass auf dynamische Änderungen reagieren kann, ist auch dieser Matcher nicht für die Verwendung geeignet.[ML09]

Epipolar-Matcher

Die vorhergehenden Matcher haben Schwierigkeiten bei der schnellen und effizienten Berechnung von Korrespondenzen. Dieser Ansatz versucht durch Verkleinerung des Suchraumes und implizite Zusammenhänge zwischen den Bildmerkmalen der Bildpaare den Aufwand zu reduzieren und deshalb schneller zu einem Ergebnis zu gelangen.

Aufgrund der Epipolargeometrie[11] muss nicht die gesamte Anzahl an Merkmalen durchsucht werden, sondern nur die Merkmale, die sich auf der Epipolarlinie befinden. Dadurch wird der Suchraum von einem zweidimensionalen auf einen eindimensionalen reduziert.[Zha98]

[11] Für ein besseres Verständnis dieses Abschnittes befindet sich im Anhang A.1 (S. 61) eine kurze Übersicht über die Epipolargeometrie.

Laut Epipolargeometrie kann die Epipolarlinie in einem Bild durchaus diagonal verlaufen. Durch die vorhergehende Rektifizierung (3.1.1 (S. 19)) der Bildquellen verlaufen alle Epipolarlinien horizontal und zudem auch in derselben Bildzeile wie auf dem Ausgangsbild.

Als nächstes kann man davon ausgehen, dass eine gewisse Reihenfolge der Merkmale eingehalten wird. Wurde eine Korrespondenz im linken Bereich einer Bildzeile gefunden und man geht nun mit der Suche einen Schritt weiter nach rechts, dann kann man davon ausgehen, dass sich das korrespondierende Merkmal ebenfalls rechts von dem zuvor gefundenen Merkmal befindet.

Ob man bei der Suche auch berücksichtigen sollte, dass bereits gefundene Merkmale nicht weiter in der Suche betrachtet werden sollten, muss zu einem späteren Zeitpunkt geklärt werden. Würde man die Merkmale aus der Suche ausschließen, dann hätte man eine weitere Einsparung im Suchaufwand, jedoch geht man das Risiko ein, einen höheren Anteil an False positives zu erhalten.

Ebenfalls ist zu bedenken, dass es Rundungsfehler bei den Berechnungen und Rauschen in den Bildquellen geben kann. Der Suchbereich sollte also nicht nur auf dieselbe Zeile begrenzt werden, sondern auch einige weitere Zeilen beinhalten.

Abschließende Anmerkungen

Bei der Suche werden auch Korrespondenzen gefunden, die in Wirklichkeit keine darstellen. Diese müssen im Nachhinein aussortiert werden. Zu diesem Zweck wird der RANSAC-Algorithmus verwendet. Dieser wird ausführlich im Anhang unter A.3 (S. 62) beschrieben.

3.2.3. Punktwolke und Tiefenbild

Es gibt zwei verbreitete Darstellungsformen für ein Ergebnis der Stereoskopie. Zum einen sind das Punktwolken und zum anderen Tiefenbilder.

Ein Tiefenbild ist im Wesentlichen die Darstellung einer Tiefe in einem zweidimensionalen Bild mittels Farbkodierung. Dabei wird zumeist eine Graustufendarstellung gewählt, aber auch eine Kodierung ähnlich einer Wärmebildkamera wird hin und wieder verwendet. Sie eignen sich besonders für die visuelle Darstellung der Ergebnisse, da sie gut zu interpretieren sind.[Tet05][Föl12]

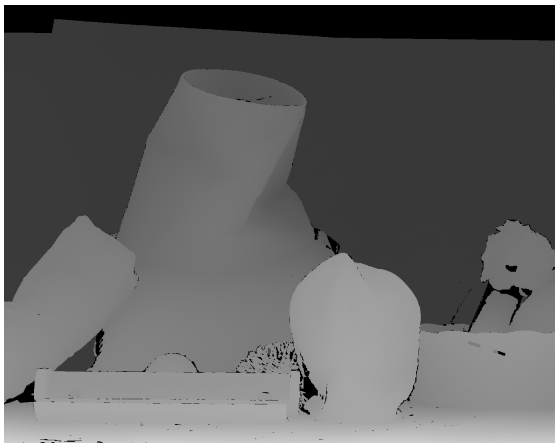

Abbildung 3.6.: Beipiel für eine Tiefenkarte in Graustufendarstellung

Im Gegensatz dazu werden Punktwolken von den meisten Systemen im Computer Vision intern verwendet und dienen als Austauschformat zwischen einzelnen Systemen. Außerdem bieten viele 3D-Programme eine Möglichkeit, Punktwolken zu importieren, um sie dann anschließend weiter zu verarbeiten.

Die Punktwolke ist eine Menge von kartesischen Koordinaten im dreidimensionalen Raum. Ein Punkt kann aber je nach Implementierung auch noch weitere Informationen, wie Farbwert, Normale etc. enthalten. Im Normalfall beschreiben sie die Oberfläche eines Objektes.

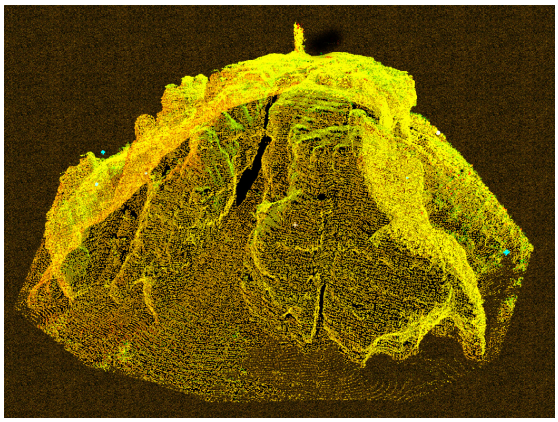

Abbildung 3.7.: Beipiel für eine Punktwolke - Darstellung einer Palastruine

Egal für welches Ausgabeformat man sich entscheidet, die Berechnungen bleiben dieselben. Die Entfernung von dem Nullpunkt des Weltkoordinatensystems zum Objektpunkt kann durch Triangulation berechnet werden. Dazu benötigt man die korrespondierenden Bildpunkte, die Basislinie und die Brennweite der verwendeten Kamerasysteme. Alle Daten können aus den vorherigen Schritten bezogen werden.

3.3. Zusammenfassung

In diesem Kapitel wurde aufgezeigt, wie aus einem vorliegenden Bildpaar eine Punktwolke generiert werden kann und welche Schritte dafür im Einzelnen notwendig sind. Unter anderem wurden dabei für die Merkmalsextraktion und die Korrespondenzsuche verschiedene Algorithmen vorgestellt, die sich sowohl in ihren Ergebnissen als auch in ihrer Ausführungszeit stark unterscheiden können.

Im Hinblick auf die Aufgabenstellung scheinen sich einige der in diesem Kapitel vorgestellten Algorithmen besser für die Verwendung in *StereoVision* zu eignen als andere. Um nicht auf einen Algorithmus beschränkt zu sein, muss *StereoVision* so entworfen werden, dass ein Wechsel zwischen verschiedenen Algorithmen mittels Konfiguration möglich ist.

4. Systemkonzeption

In diesem Kapitel wird gezeigt, wie die Algorithmen, die in dem vorherigen Kapitel vorgestellt wurden, in ein neu entwickeltes System, *StereoVision*, integriert werden können. In Kapitel 2 wurde gezeigt, dass zur Erfüllung der Anforderungen nur ein Verfahren, das auf Stereokopie basiert, verwendet werden kann. In Kapitel 1 wurden die Grundlagen für dieses Verfahren vorgestellt. Dabei hat sich gezeigt, dass für die beiden wesentlichen Schritte der Tiefenbildgenerierung mittels Stereoskopie unterschiedliche Algorithmen existieren. Daher soll in diesem Kapitel ein erweiterbares und konfigurierbares System entwickelt werden, damit in der Evaluierung eine Aussage über die Tragfähigkeit des Ansatzes unter quantitativen und qualitativen Kriterien getroffen werden kann.

In Abschnitt 4.1 werden aufbauend auf Kapitel 1 und Kapitel 2 die funktionalen und nicht-funktionalen Anforderungen ausformuliert.

In Abschnitt 3.3 wurde bereits erwähnt, dass sich nicht alle der Algorithmen dazu eignen, in dem zu entwickelnden System implementiert und verwendet zu werden. Es wurde entschieden, zum Extrahieren von Features *STAR*, *ORB* und *FAST* sowie für die Korrespondenzsuche den *Epipolar-Matcher* und den *Brute-Force-Matcher* zu verwenden. Damit zusätzlich zu den gewählten Varianten auch neue hinzugefügt werden können, wird in Abschnitt 4.2 ein Entwurf vorgestellt, der dem Austausch von Implementierungen einfach möglich macht.

Die dazu verwendeten Entwurfsmuster werden in Abschnitt 4.3 vorgestellt und es werden in Abschnitt 4.4 die Entscheidungen, die für die Implementierung getroffen werden mussten, geklärt. Der Abschnitt schließt mit einer Anleitung zum Austausch und der Erweiterung von Algorithmen.

4.1. Anforderungsanalyse

Im folgenden werden sowohl die funktionalen als auch die nicht-funktionalen Anforderungen abgesteckt. Im Abschnitt 4.2 wird dann der Systementwurf beschrieben.

Aus den gelesenen Kamerabildern sollen Daten in einem anderen Format generiert werden. Bei diesem Format kann es sich um ein Tiefenbild oder eine Punktwolke handeln. Das System selbst reicht diese Daten anschließend nach außen weiter. Es fungiert somit als ein Treiber für eine übergeordnete Anwendung.

4.1.1. Funktionale Anforderungen

Das System soll eine Tiefenbildgenerierung aus stereoskopischen Bildern ermöglichen. Der Treiber muss konfigurierbar sein und Einstellmöglichkeiten besitzen, um den zu analysierenden Bereich zu definieren. Der Bereich kann sich dabei von 1 Meter bis zu 40 Meter entfernt von dem Kamerasystem befinden. Der Einsatz ist nicht auf Innenräume beschränkt, es soll also möglich sein, das Kamerasystem im Freien zu verwenden.

Dementsprechend ist eines der wichtigsten Einsatzgebiete die Generierung von Tiefenbildern auf großen Plätzen, um anschließend User tracken zu können. Es muss möglich sein, sowohl Bilder von beliebigen Kamerasystemen als auch von der Festplatte zu verwenden.

4.1.2. Nicht-funktionale Anforderungen

Performanz Die Verarbeitung der Bilder muss möglichst schnell geschehen, damit ein fließendes Kamerabild entsteht. Ohne eine ausreichend schnelle Folge von Bildern können überliegende Anwendungen nicht Usertracking bereitstellen.

Konfigurierbar Der Austausch von Algorithmen sollte leicht zu bewerkstelligen und nicht viel Zeit in Anspruch nehmen. Schon zu Beginn sollte die Möglichkeit bestehen, dass der Treiber konfigurierbar ist, um verschiedene Implementierungen und dadurch verschiedene Algorithmen vergleichen zu können.

Qualität Die Ergebnisse des Treibers sollen weiterverarbeitet werden, um mit ihrer Hilfe Personen zu verfolgen. Sie müssen also eine ausreichend hohe Anzahl an Informationen mit einer genauen Tiefenangabe enthalten.

4.2. Systementwurf

Das System ist so entworfen, dass mögliche Änderungen von Algorithmen schnell zu bewerkstelligen sind und immer nur eine Komponente betroffen ist.

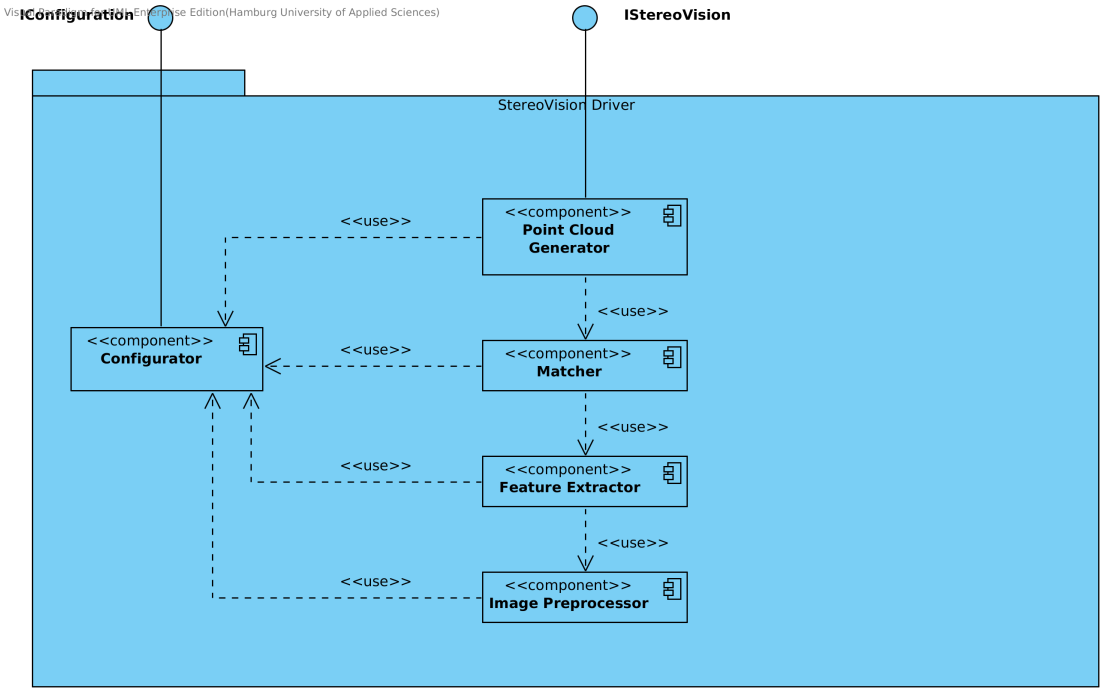

Abbildung 4.1.: Komponentendiagramm des Systementwurfs

Zunächst werden die beiden wichtigsten Klassen für die Verwendung des Treibers durch den Client beschrieben.

Die fünf Komponenten, die zusammen den Treiber bilden, werden genauer vorgestellt. Die *Configurator*-Komponente ist für alle weiteren Komponenten erreichbar und beinhaltet alle Einstellungen, die zur Initialisierung des Treibers vorliegen. Die restlichen vier Komponenten haben untereinander nur eine Verbindung zu der ihnen unterliegenden Komponente. Lässt man den *Configurator* also außen vor, dann ergibt sich ein Schichtenmodell wie es in Abbildung 4.9 zu sehen ist.

4.2.1. Schnittstellen

In diesem Abschnitt werden Interfaces bzw. Klassen vorgestellt, die für die Verwendung des Treibers durch einen Client relevant sind. Über das Interface *IStereoVision* können Punktwolken ermittelt werden, aus denen der Client anschließend die Daten zum Tracken von Usern berechnen kann.

ImagePair

ImagePair ist eine Containerklasse für zwei zueinander gehörende Bilder und die Informationen, die aus diesen Bildern generiert werden können. Dazu gehören die im *FeatureExtractor* ermittelten Keypoints und deren Matches, die in *StereoMatcher* gesucht werden. Weitere Informationen, wie ein Tiefenbild oder eine Punktwolke, werden im *PointCloudGenerator* hinzugefügt. Das Objekt wird von der untersten Schicht bis zur obersten hindurch gereicht und in jeder Schicht werden die Informationen aus vorhergehenden Schichten verarbeitet und weitere Informationen hinzugefügt.

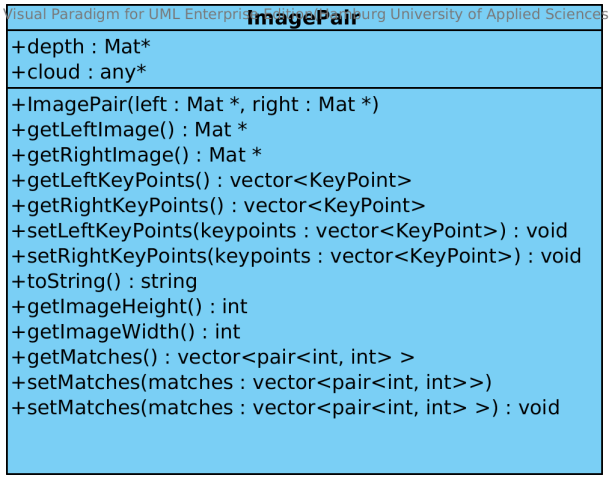

Abbildung 4.2.: Klassendiagramm des ImagePair-Containers

IStereoVision

Das Interface für die Verwendung des Treibers wird durch *IStereoVision* bereitgestellt (Abb. 4.3). Je nach Konfiguration können *getDepthImage* oder *getDepthMap* verwendet werden. Das Format des Tiefenbildes, welches mit dem Aufruf von *getDepthImage* vom Treiber zur Verfügung gestellt wird, entspricht dem Bildformat von **OpenCV** in Graustufen mit einer Farbtiefe von acht Bit. Mit *getDepthMap* erhält man die Punktwolke. Deren Format ist abhängig von der Konfiguration von StereoVision.

Die Methode *getImagePair* gibt immer das aktuelle *ImagePair* zurück. Jede der drei Aktionen führt einen kompletten Round-Trip durch und gibt einen neuen Zustand zurück. Will man mehrmals auf den gleichen Zustand zugreifen, dann muss das Ergebnis lokal in der Anwendung gehalten werden.

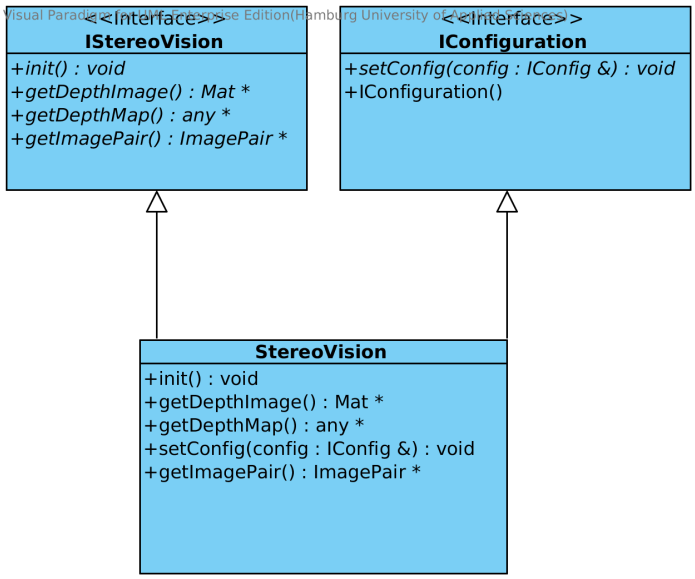

Abbildung 4.3.: Klassendiagramm des Treiber-Interfaces

Bevor eine der drei eben genannten Aktionen erfolgreich ausgeführt werden kann, muss das System mit *init* initialisiert werden. Alle bis dahin vorgenommen Konfigurationen kommen zur Anwendung und *StereoVision* ist anschließend betriebsbereit.

4.2.2. Komponenten

Configurator

Der Treiber kann mithilfe des *Configurators*, zu sehen in Abbildung 4.4, angepasst werden. Um die Konfiguration des Systems einfach zu gestalten, stehen zwei Möglichkeiten bereit, die Konfiguration zu ändern.

Der Anwendungsentwickler besitzt die Möglichkeit das Verhalten auf zwei Arten zu ändern. Zum einen über Konfigurationsdateien, die er mittels einer *FileConfig*-Instanz laden kann, und zum anderen durch eine Definition von Schlüssel-Werte-Paaren in der Software mittels *CodeConfig*. Das Format der Konfigurationsdateien muss dem JSON-Format genügen. Neue Definitionen werden in die *Configuration*-Instanz übernommen und schon vorhandene werden durch neuere Definitionen überschrieben. Sollten für eine benötigte Definition keine Werte angegeben worden sein, dann wird ebenfalls auf die Defaultwerte zurückgegriffen.

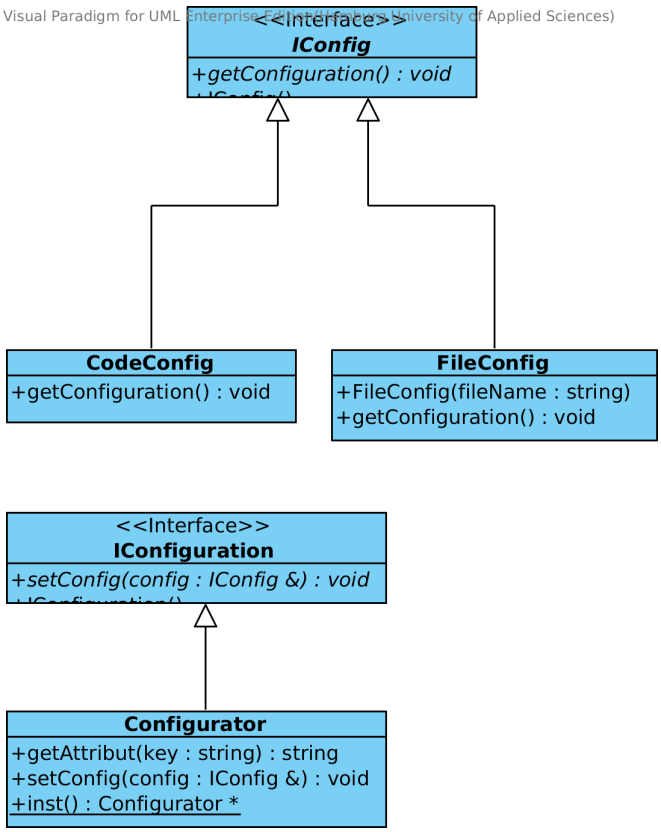

Abbildung 4.4.: Klassendiagramm der Configurator-Komponente

Wird keine der unterstützten Varianten verwendet, dann greifen die Defaultwerte, die fest in das System integriert sind.

Image Preprocessor

Die Komponente *Image Preprocessor* (Abb. 4.5) bildet die unterste Schicht und wird für die Vorverarbeitung der Bilder verwendet. Es gibt zwei Möglichkeiten, wie man ein *ImagePair* erzeugen kann. Für den Produktiveinsatz und den gedachten Einsatzzweck des Treibers wird ein *CameraImagePreprocessor*, der Bilder von zwei Kameras liest, bereitgestellt. Die Kamerabilder werden vor der Weitergabe in Graustufenbilder umgewandelt und nach Möglichkeit rektifiziert. Für die Evaluierung und zur besseren Vergleichbarkeit von verschiedenen Algorithmen wird auch die Möglichkeit geschaffen, Dateien als Input zu verwenden. Die Bilder, die über den *FileImagePreprocessor* geladen werden, werden nicht rektifiziert, sondern nur in Graustufenbilder umgewandelt.

In der Instanz des *Image Preprocessors* wird für die Weiterverarbeitung ein *ImagePair*-Objekt erstellt, auf dem alle anschließenden Operationen ausgeführt werden. In den nachfolgenden Abschnitten werden diese Operationen benannt.

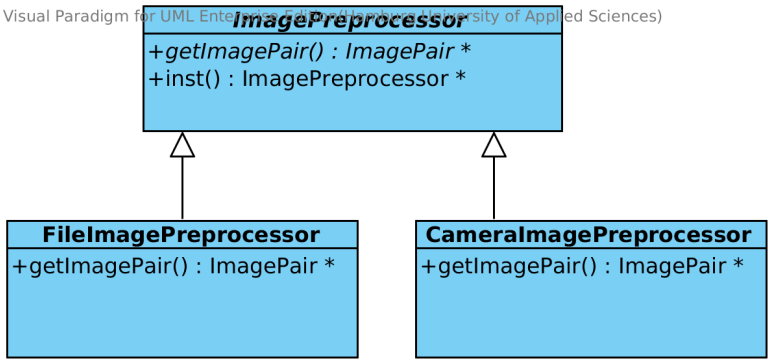

Abbildung 4.5.: Klassendiagramm des ImagePreprocessor-Layers

Feature Extractor

Nachdem die Bilder geladen wurden, werden sie in der *ImagePair*-Instanz an den *FeatureExtractor* zurückgegeben. *FeatureExtractor* ist eine abstrakte Klasse und besitzt eine öffentliche abstrakte Methode *getDescriptedImages*. Die Methode *getDescriptedImages* sucht in den beiden Bildern des Bildpaares nach Keypoints und speichert diese in dem *ImagePair* ab. Je nach Wahl des Algorithmus der die Features aus den Bildern extrahieren soll, muss sich für eine Implementierung entschieden werden. Zunächst ist vorgesehen, den *Star*-, den *ORB*- und den *FastFeatureExtractor* zur Verfügung zu stellen (Abb. 4.6), um vergleichen zu können, wie sich die Ergebnisse bei der Verwendung unterschiedlicher Algorithmen verändern.

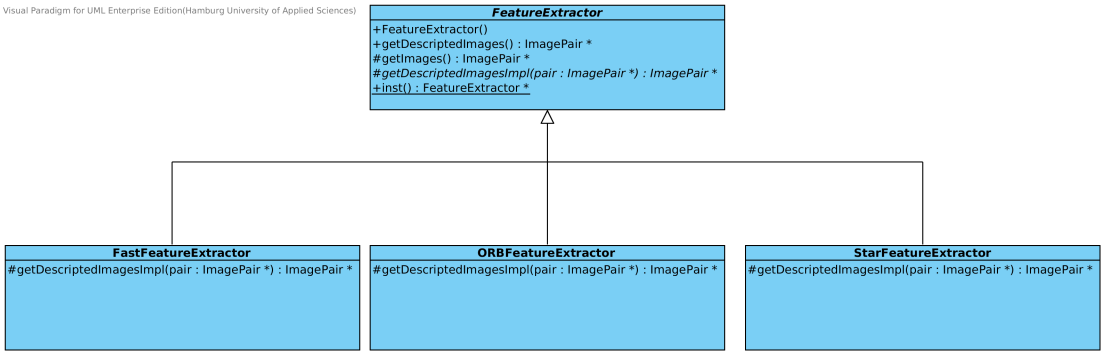

Abbildung 4.6.: Klassendiagramm des FeatureExtractor-Layers

Stereo Matcher

Wie schon in dem Grundlagenkapitel erklärt, benötigt die Korrespondenzsuche, auch Stereomatching genannt, den größten Anteil der Verarbeitungszeit. Um zu demonstrieren, wie groß der Gewinn im Hinblick auf die Performanz mit dem zuvor vorgestellten *Epipolar-Matcher*(Abschnitt 3.2.2 (S. 25)) ist, wird neben dem eigentlichen *EpipolarMatcher* auch ein *BruteForceMatcher* implementiert (Abb. 4.7).

Nach dem Erhalt des *ImagePairs* mit den gefundenen Keypoints sucht der gewählte Matcher nach Korrespondenzen in den Bildpaaren und gibt diese anschließend weiter an die nächst höhere Schicht.

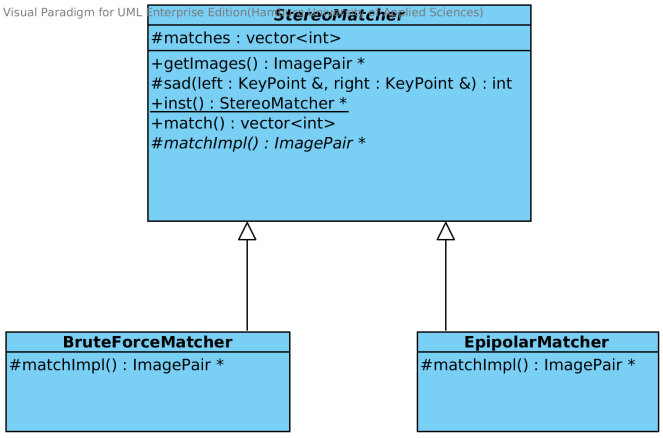

Abbildung 4.7.: Klassendiagramm des StereoMatcher-Layers

Als Maß für die Ähnlichkeit für die Keypoints wird *SAD*[1] verwendet. Dieser Algorithmus hat im Gegensatz zu dem *ZNCC*-Verfahren[2] den Vorteil, dass er wesentlich schneller zu berechnen ist. Da man davon ausgehen kann, dass die Helligkeit in beiden Bildern annähernd gleich ist, würde der *ZNCC*-Algorithmus auch keine Vorteile bei den Ergebnissen bringen. Nähere Informationen zu *SAD* und *ZNCC* befinden sich im Anhang A (S. 61).

Point Cloud Generator

Die abstrakte Klasse *PointCloudGenerator* befindet sich in der obersten Schicht des Treibers und bietet die Möglichkeit, je nach Anwendungsfall, Punktwolken oder Tiefenbilder zu gener-

[1] Sum of Absolute Difference
[2] Zero Mean Normalized Cross Correlation

ieren. Um auch an dieser Stelle einen möglichst großen Freiraum in der Konfiguration zu bieten, ist es möglich, neue Formate für Punktwolken zu implementieren und einfach in das bestehende System einzubinden.

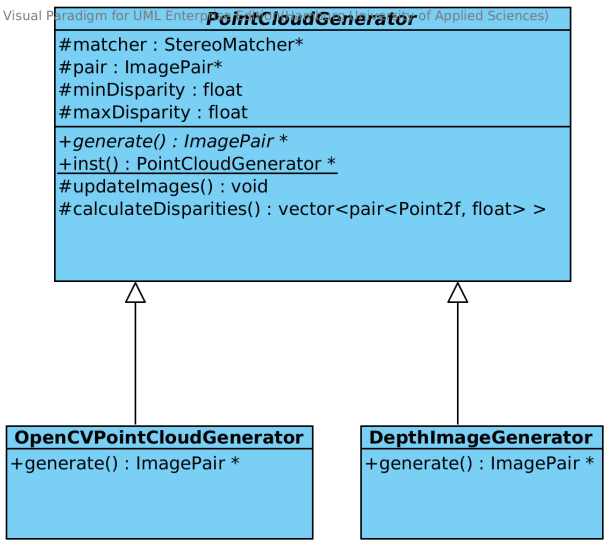

Abbildung 4.8.: Klassendiagramm des PointCloudGenerator-Layers

Mit der geschützten Methode *calculateDisparites* können die Disparitätswerte berechnet werden und es wird das Minimum und das Maximum ermittelt. Diese Werte können anschließend genutzt werden, um eine Punktwolke zu berechnen. Die Methode wird sowohl in *OpenCVPointCloudGenerator* als auch in *DepthImageGenerator* verwendet.

In Abbildung 4.8 sieht man die abstrakte, öffentliche Methode *generate* in der abstrakten Klasse *PointCloudGenerator*. Im Gegensatz zu dem *StereoMatcher* und dem *FeatureExtractor*, bei denen eine Template-Methode in der abstrakten Klasse aufgerufen wird und in dieser dann die Implementierung der abstrakten Methode durch eine Instanz einer Kindklasse vorgenommen wird, kann die Implementierung der abstrakten Methode hier direkt von außerhalb aufgerufen werden. Es werden durch die abstrakte Klasse nur Hilfsfunktionen zu Verfügung gestellt. Dieser Schritt wurde gemacht, damit man in späteren Erweiterungen flexibler arbeiten kann.

4.3. Entwurfsmuster

Im Folgenden werden die Entwurfsmuster, die für den Treiber verwendet wurden, aufgezählt und es wird beschrieben, an welcher Stelle sie eingesetzt wurden und welche Vorteile sich aus ihrer Verwendung ergeben.

4.3.1. Schichtenmodell

Der Großteil des Treibers wird als Schichtenmodell implementiert. Dadurch sind die einzelnen Komponenten besser gekapselt und eine Änderung in einer Komponente beeinflusst nicht die anderen Komponenten. Die gestrichelten Pfeile in Abbildung 4.9 veranschaulichen die Aufrufreihenfolge. Wird ein Request an den Treiber gestellt, dann beginnt der Aufruf in Point Cloud Generator. Von dort an wird er bis zum Image Preprocessor herunter gereicht. Dieser arbeitet auf der IO und gibt das geladene Bildpaar an die oberen Schichten zurück. Jede Schicht fügt zusätzliche Informationen zu den bestehenden hinzu und zuletzt wird, je nach Anfrage, eine Punktwolke oder ein Tiefenbild zurückgegeben.

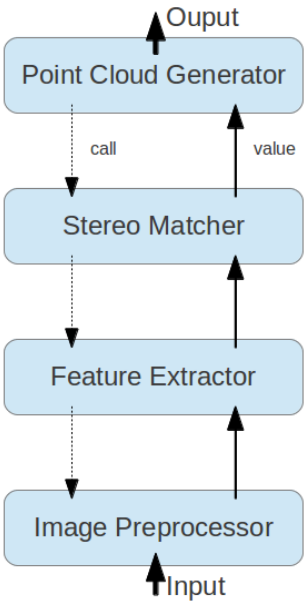

Abbildung 4.9.: Überblick über das Schichtenmodell des Treibers

4.3.2. Singleton

Für die Konfiguration wird ein Singleton-Objekt des *Configurators* verwendet. Durch die Verwendung eines *Singleton* können die einzelnen Schichten getrennt voneinander auf dieselben Einstellungen zugreifen und es muss nicht darauf geachtet werden, dass alle auf demselben Objekt arbeiten und somit auch die gleichen Daten zur Verfügung haben.

Die Verwendung einer threadsicheren Variante ist nicht notwendig, da alle Werte vor der Initialisierung vorliegen müssen und danach nur noch lesend auf das *Singleton* zugegriffen wird. Nebenläufigkeit kann erst nach der Initialisierung auftreten.

4.3.3. Fabrikmethode

Die Instanziierung der vier Komponenten erfolgt über Fabrikmethoden, die in den abstrakten Klassen implementiert sind. Dadurch kann schnell zwischen verschiedenen Implementierungen und den damit verbundenen Algorithmen gewechselt werden. Durch eine Fabrikmethode ist es möglich, erst zur Laufzeit zu entscheiden, welche Instanz einer Klasse verwendet werden soll.

Möchte man in dem Treiber einen anderen Algorithmus einsetzen, dann trägt man diesen Algorithmus in die Config-Datei ein und startet das System neu. In der Initialisierung wird dann automatisch die richtige Klasse gewählt. Deshalb entfällt die, durch eine Änderung am Code entstehende, nötige Neukompilierung.

4.4. Implementierung

4.4.1. Programmiersprache

Als Programmiersprache wurde C++ gewählt. Es wurde nicht konsequent auf die Verwendung eines C++-Standards, etwa C++0x, geachtet. Programme, die in diese Sprache geschrieben wurden, müssen vor ihrer Ausführung durch einen Compiler kompiliert und zu einer ausführbaren Datei gepackt werden.

4.4.2. Verwendete Bibliotheken

Als Grafikbibliothek wird OpenCV mit dem C++-Wrapper verwendet. Diese Bibliothek bietet sehr viele, performante Implementierungen von Algorithmen, die für die Verarbeitung von Bildmaterial wichtig sind.

Als Erweiterung der Standardbibliothek von C++ wird Boost verwendet. Boost bietet umfangreiche Erweiterungen, etwa im Bereich der Metaprogrammierung, Container-Klassen, Nebenläufigkeit und dem Parsen von Textformaten.

4.4.3. Verwendetes Entwicklungssystem

Als Entwicklungs- und Testumgebung dient für diese Arbeit Ubuntu 12.10 und der GCC g++-Compiler in Version 4.7.2. Als IDE wurde Eclipse Juno mit den Erweiterungen CDT, Ecloxy und Egit verwendet. In der Tabelle 4.1 befinden sich weitere Daten des verwendeten Systems.

CPU	Intel Core i7-3820
Kerne und Takt	4 (8 mit HT) * 3,6 GHz
Betriebssytem	Ubuntu 12.10 64bit
HDD	468,6 GB
RAM	11,7 GiB
Grafikkarte	NVIDIA Quadro 2000

Tabelle 4.1.: Die Komponenten des Entwicklungs- und Testrechners

4.4.4. Erweiterung & Modifikation

StereoVision bietet vier Möglichkeiten Algorithmen zu ersetzen, indem eine der Komponenten, Point Cloud Generator, Matcher, Feature Extractor und Image Preprocessor, neu implementiert wird.

Um einen Algorithmus durch einem anderen Algorithmus zu ersetzen, der bereits implementiert ist, muss die entsprechende Stelle in der Konfiguration angepasst werden. Die Konfiguration kann durch die Anpassung der Default-Werte, das Laden und die Übergabe einer Konfigurationsdatei mit *FileConfig* oder die Erstellung und die Übergabe eines *CodeConfig*-Objektes vor der Initialisierung des Treibers geändert werden. Das System wird dann bei der Initialisierung über Factory-Methoden die richtigen Klassen laden und zur Verfügung stellen. Sollte ein Algorithmus nicht implementiert sein, dann wird ein Fehler geworfen und das Programm beendet.

Sollte der Wunsch bestehen, einen Algorithmus zu verwenden, der bisher nicht implementiert wurde, dann muss dieser selbständig implementiert werden. Als erstes wird von der entsprechenden abstrakten Klasse geerbt und die abstrakten Methoden müssen implementiert

werden. Anschließend kann die neue Klasse in der Factory-Methode der abstrakten Klasse registriert werden. Wie genau Klassen registriert werden können, wird im Quellcode beschrieben. Nachdem die Konfiguration angepasst und der Quellcode neu kompiliert wurde, wird der neue Algorithmus vom Treiber verwendet.

4.5. Zusammenfassung

Der Treiber StereoVision wurde unter Zuhilfenahme gängiger Entwurfstechniken konzipiert. Die in Abschnitt 4.1 gestellten funktionalen Anforderungen wurden erfüllt. Es wurde ein objektorientierter Komponentenentwurf gewählt, um den Austausch und die Erweiterbarkeit von Komponenten einfach und den Aufwand gering zu halten.

Es wurden ein **Epipolar**-Matcher und ein **Brute-Force**-Matcher für das Stereomatching implementiert. Um den Zusammenhang zwischen der Qualität der Ergebnisse und den gefundenen Features zu ermitteln, wurden **ORB**, **FAST** und **STAR** als *FeatureExtractor* eingebunden. Zur Ausgabe dienen ein 8-bit-Tiefenbild und eine Punktwolke, die auf der OpenCV-Datenstruktur für Punkte im dreidimensionalen Raum basiert.

Das nachfolgende Kapitel evaluiert die Ergebnisse in Hinblick auf ihre Verlässlichkeit und Performanz, überprüft also die Einhaltung der nicht-funktionalen Anforderungen.

5. Evaluierung

Dieses Kapitel beinhaltet die Evaluierung des Treibers. In Abschnitt 5.1 wird das Testverfahren beschrieben und in Abschnitt 5.2 werden die Ergebnisse der einzelnen Tests dargestellt. Anschließend werden die Ergebnisse in Abschnitt 5.3 ausgewertet und es wird ein Fazit gegeben.

5.1. Testverfahren

Quantitativ Das Testverfahren besteht aus zwei Teilen. Im ersten Teil werden Messungen mit verschiedenen Bildpaaren gemacht und die Ergebnisse der einzelnen Algorithmen im Hinblick auf benötigte Zeit und gefundene Korrespondenzen verglichen.

Qualitativ In einem weiteren Test werden die Ergebnisse visuell mit dem optimalen Ergebnis verglichen, um abzuschätzen, wie gut die Qualität der erzeugten Bilder ist. Das optimale Ergebnis eines Tiefenbildes ist ein **Ground-Truth**. Diese Bilder werden zumeist mit Hilfe von strukturiertem Licht erzeugt, indem ein Muster auf eine Szene projiziert wird.

Die Bilddaten und das Ground-Truth-Bild stammen aus der Stereo-Bilddatenbank des Middlebury Colleges[1]. Für die Vergleiche werden folgende Algorithmen verwendet:

Feature Extractor

- STAR
- FAST
- ORB

Matcher

[1] URL: http://vision.middlebury.edu/stereo/data/

- Brute-Force-Matcher
- Epipolar-Matcher

Es werden vier Bildpaare mit jeder Extractor/Matcher-Kombination getestet, sodass insgesamt 24 Tests durchgeführt werden. Jeder Test wird zehn Mal ausgeführt und die Ergebnisse anschließend gemittelt.

5.2. Testergebnisse

5.2.1. Verwendete Bilder

Nachfolgend wird jeweils das linke Bild der vier Bildpaare, die in den Tests verwendet werden, dargestellt, um eine Vergleichsmöglichkeit mit den Ergebnissen zu erhalten.

Tabelle 5.1.: Links oben befindet sich Testbild A, rechts daneben Testbild B in der unteren Zeile links Testbild C und daneben Testbild D

5.2.2. Ergebnisse aus Messungen

In diesem Abschnitt werden die Testergebnisse aufgelistet. In Tabelle 5.2 befinden sich die Ergebnisse in Hinblick auf die benötigte Gesamtzeit in Millisekunden und in Tabelle 5.3 die Anzahl der gefunden Korrespondenzen. In Tabelle 5.4 wird dann die Zeit pro Korrespondenz in Millisekunden angegeben.

5.2.3. Generierte Tiefenbilder

Für die qualitative Prüfung der Ergebnisse werden in diesem Abschnitt einige der generierten Tiefenbilder dargestellt. Dabei wird jeweils das Graustufenbild des linken Bildes, das Ground-Truth und zwei durch den Treiber generierte Tiefenbilder dargestellt. Bei den beiden generierten Bildern werden unterschiedliche Kombinationen aus *Matcher* und *FeatureExtractor* verwendet.

5.3. Schlussfolgerung

Bei dem Vergleich der Ergebnisse des ersten Teils der Tests wird ersichtlich, dass die Zeit, die die Generierung von Tiefenbildern benötigt, stark von der Wahl des Algorithmus zur Merkmalsextraktion abhängig ist. *FAST* und *ORB* finden eine Anzahl von Features in derselben Größenordnung. *STAR* hingegen extrahiert nur einen kleinen Teil der Features, braucht dementsprechend aber auch weniger Zeit für einen gesamten Durchlauf. Da die Anzahl der Merkmale, die mit *STAR* gefunden werden, sehr gering ist, ist dieser FeatureExtractor nicht für eine Verwendung in StereoVision geeignet.

Den größten Einfluss auf die Durchlaufzeiten hat die Wahl des Matchingverfahrens, wenn die Anzahl der gefundenen Features groß ist.

Betrachtet man die Ergebnisse des zweiten Testteils, dann stellt sich heraus, dass der Epipolar-Matcher wesentlich bessere Ergebnisse erzielt. Diese Ergebnisse sind jedoch auch stark abhängig von den genutzten Extrahierungsalgorithmus.

Die Ergebnisse zeigen, dass die Kombination von Epipolar-Matcher und FAST-FeatureExtractor die besten Ergebnisse, sowohl qualitativ als auch quantitativ, erzielt.

Alle FeatureExtractoren haben große Schwierigkeiten, auf monotonen Flächen, Features zu extrahieren. Somit stehen diese Bereiche nicht für eine Tiefenbildgenerierung zur Verfügung.

		Bilder			
Matcher	Feature Extractor	A	B	C	D
Brute Force	FAST	1054	5168	184369	98783
	STAR	87	51	45	119
	ORB	329	1210	60299	20179
Epipolar	FAST	63	204	3112	1683
	STAR	84	47	42	44
	ORB	63	74	877	307

Tabelle 5.2.: Die Gesamtzeit in Millisekunden, die die einzelenen Kombinationen für die Testbilder benötigen.

		Bilder			
Matcher	Feature Extractor	A	B	C	D
Brute Force	FAST	1277	3136	18188	12825
	STAR	23	38	54	108
	ORB	685	1430	10000	5780
Epipolar	FAST	1037	2670	15328	10922
	STAR	17	21	39	79
	ORB	465	1051	6250	3694

Tabelle 5.3.: Die Anzahl der gefunden Korrespondenzen der Matcher/FeatureExtractor-Kombinationen.

		Bilder			
Matcher	Feature Extractor	A	B	C	D
Brute Force	FAST	0,83	1,65	10,14	7,70
	STAR	3,78	1,34	0,83	1,10
	ORB	0,48	0,85	6,03	3,49
Epipolar	FAST	0,06	0,08	0,20	0,15
	STAR	4,94	2,24	1,08	0,56
	ORB	0,14	0,07	0,14	0,08

Tabelle 5.4.: Die Zeit in Millisekunden, die die Kombinationen pro Korrespondenz benötigen.

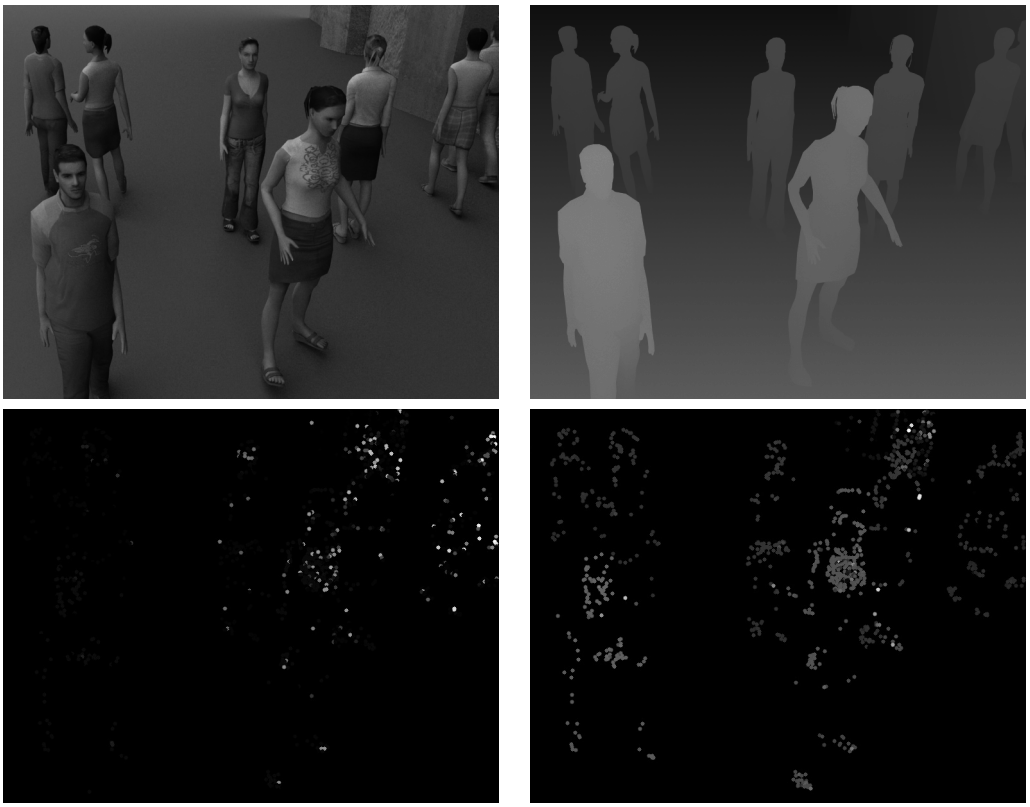

Tabelle 5.5.: Graustufenbild von Testbild A, Ground-Truth, Ergebnis der Kombination von Brute-Force-Matcher mit FAST-FeatureExtractor(l.) und Epipolar-Matcher mit FAST-FeatureExtractor(r.)

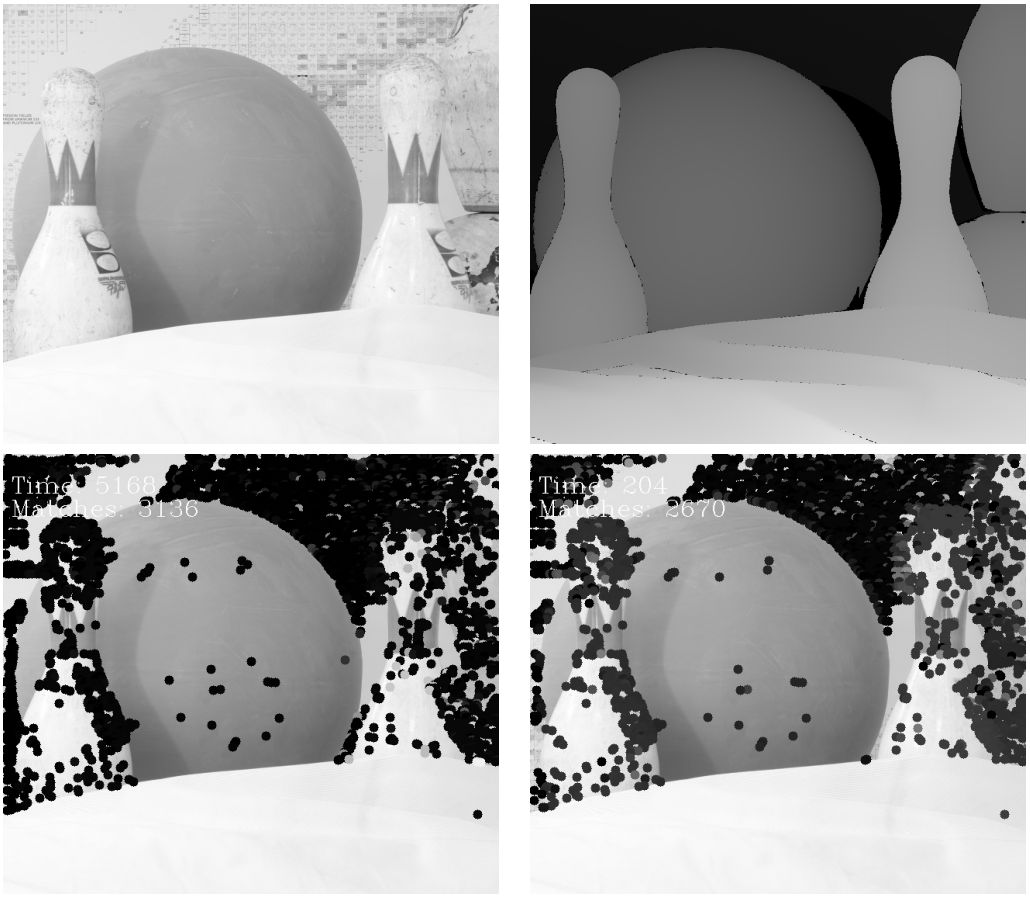

Tabelle 5.6.: Graustufenbild von Testbild B, Ground-Truth, Ergebnis der Kombination von Brute-Force-Matcher mit FAST-FeatureExtractor(l.) und Epipolar-Matcher mit FAST-FeatureExtractor(r.)

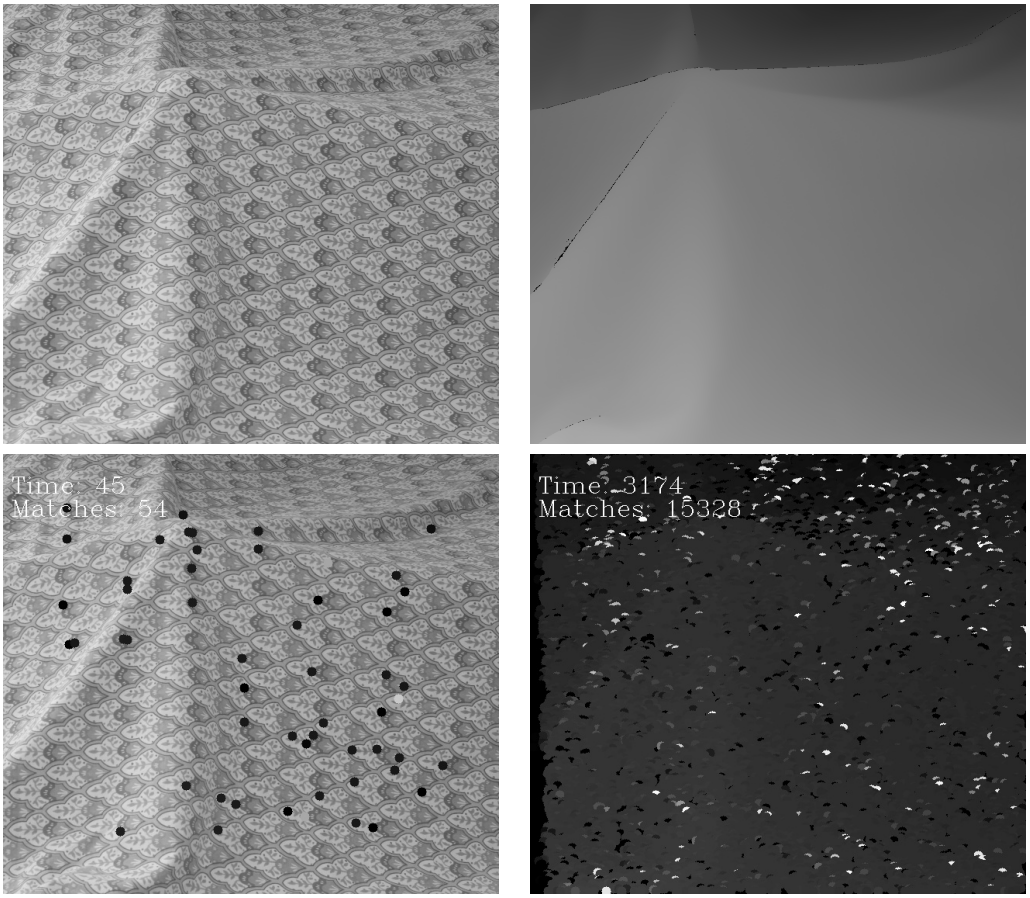

Tabelle 5.7.: Graustufenbild von Testbild C, Ground-Truth, Ergebnis der Kombination von BruteForce-Matcher mit STAR-FeatureExtractor(l.) und Epipolar-Matcher mit FAST-FeatureExtractor(r.)

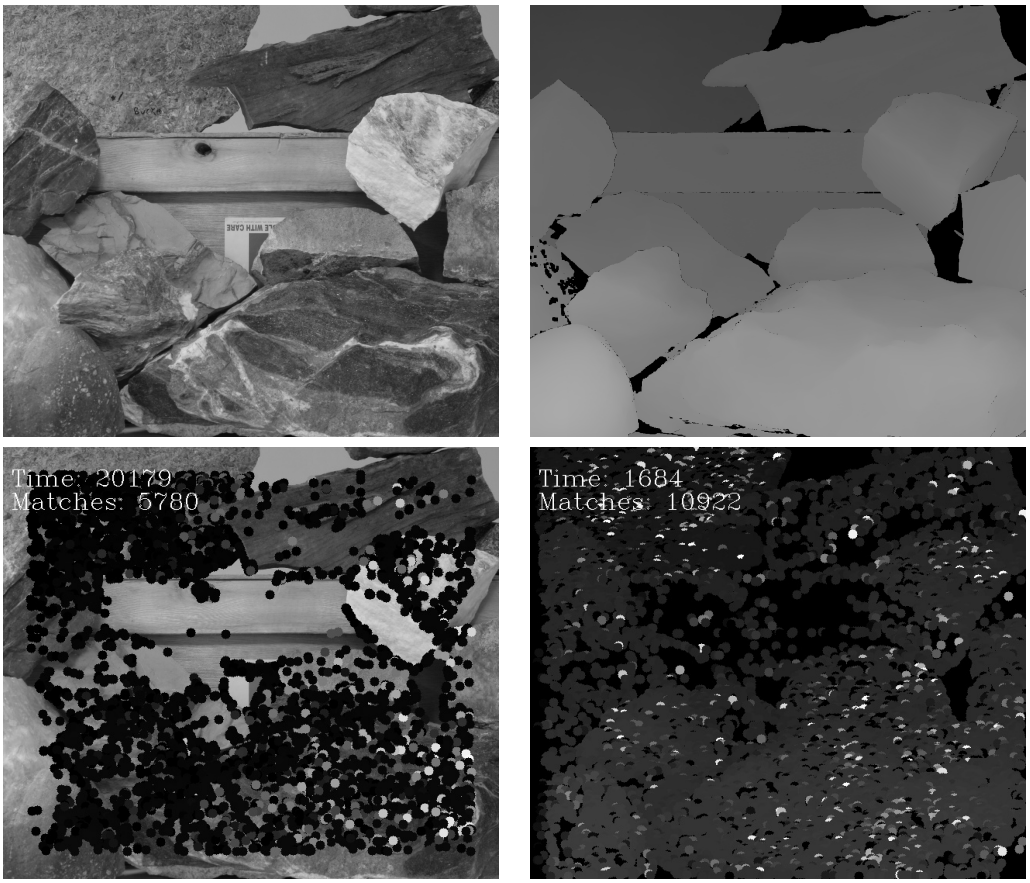

Tabelle 5.8.: Graustufenbild von Testbild D, Ground-Truth, Ergebnis der Kombination von BruteForce-Matcher mit ORB-FeatureExtractor(l.) und Epipolar-Matcher mit FAST-FeatureExtractor(r.)

6. Zusammenfassung

In den vorangegangenen Kapiteln wurde ein Treiber für die Generierung von Tiefenbildern vorgestellt und evaluiert. Dieses Kapitel fasst die Ergebnisse, die erzielt wurden zusammen und zeigt Schwachpunkte der Lösung auf. Am Ende wird ein Ausblick auf eine mögliche Weiterentwicklung und Verbesserung der bisherigen Ergebnisse gegeben.

Der Ausgangspunkt dieser Arbeit war die Suche nach einer Möglichkeit, Personen für eine Interaktion zu tracken und dabei nicht den Einschränkungen bisheriger Verfahren unterliegen.

Die Verfahren, die derzeit zur Verfügung stehen, wurden in Kapitel 2 vorgestellt. In Kapitel 3 wurde dann auf die Grundlagen eingegangen, die benötigt werden, um eine Lösung auf Basis von Stereoskopie zu erstellen.

Das Konzept des Treibers wurde in Kapitel 4 vorgestellt. Dabei wurde besonderes Augenmerk auf die Austauschbarkeit und die Erweiterbarkeit von **StereoVision** gelegt. Außerdem wurde genauer auf die Nutzung der angebotenen Schnittstellen eingegangen.

Kapitel 5 stellt anhand einiger Beispielbildpaare die quantitativen und qualitativen Eigenschaften der Lösung dar.

6.1. Ergebnis

Die Ergebnisse, die in Kapitel zur Evaluierung präsentiert wurden, zeigen, dass mit der Kombination aus Epipolar-Matcher und Fast-FeatureExtractor qualitativ gute Ergebnisse erzielt werden können, die die gleichen Charakteristika wie die Ground-Thruth-Bilder aufweisen. Sowohl die Formen als auch die Helligkeitsabstufungen sind in den Bildern sehr ähnlich.

Im Vergleich zu dem Brute-Force-Matcher bietet der Epipolar-Matcher einen signifikanten Geschwindigkeitsvorteil, der in sich in einer 30 bis 60 mal schnelleren Berechnungszeit widerspiegelt. Auch im Vergleich zu vorangegangenen Arbeiten können Verbesserungen, sowohl

in der Geschwindigkeit als auch in der Anzahl der gefunden Korrespondenzen beobachtet werden.[Föl12][Tet05]

Vergleicht man die einzelnen Ergebnisse miteinander, dann wird klar, dass die Resultate stark von der dargestellten Szene abhängig sind. So kann es vorkommen, dass viele Features im Hintergrund eines Bildes erkannt werden, aber der Vordergrund und somit das eigentliche Objekt, dass untersucht werden sollte, kaum enthält Features. Überträgt man diese Beobachtung auf das beabsichtigte Anwendungsgebiet, dann muss sichergestellt werden, dass Personen, die mit dem System interagieren wollen, sich deutlich vom Hintergrund abheben und Kleidung tragen, die gemustert ist. Zudem sollte der Hintergrund möglichst einfach sein, damit kaum Features erkannt werden.

Je mehr Features gefunden werden, desto besser ist das Ergebnis des Tiefenbildes. Doch mit zunehmender Anzahl steigt auch die Dauer der Verarbeitung. In den Tests wurden Zeiten von etwa 3,5 Sekunden gemessen. Aufgrund dessen können zwischen zwei Ergebnissen fast vier Sekunden vergehen. Die Bewegung eines Nutzers zu verfolgen, ist mit dieser großen Zeitspanne nicht möglich.

Es ist nach derzeitigem Wissen technisch nicht möglich, eine Lösung auf Basis von Stereoskopie zu schaffen, mit deren Hilfe man Benutzer auf öffentlichen Plätzen dreidimensional tracken kann und dabei ähnliche Ergebnisse erzielt, wie sie etwa von einer Kinect-Kamera geliefert werden.

6.2. Ausblick

Eine erster Ansatz der Verbesserung wäre die Parallelisierung des Epipolar-Matchers und des FeatureExtractors mithilfe eines Threadpools. Die wichtigsten Schritte dafür wurden bereits bearbeitet. Durch die Parallelisierung können die rechenintensiven Schritte um ein Vielfaches verkürzt werden.

Da die Verarbeitungszeit stark von der Anzahl der Features abhängt, sollte man diese Anzahl auf die n besten begrenzen. Durch eine Entfernung des Hintergrunds vor der eigentlichen Verarbeitung können mehr Details im Vordergrund gefunden werden und das Ergebnis erhält eine höhere Genauigkeit.

Auch wenn die Ergebnisse schon sehr nahe an die Ground-Thruth-Bilder herankommen, fällt ein Rauschen in den Tiefenbildern auf, dass von falsch zugeordneten Korrespondenzen stammt. Dieses Rauschen kann entfernt werden, indem man die falschen Zuordnungen ausfiltert. Ein Beispiel für einen solchen Filter wäre die Verwendung des RANSAC-Algorithmus.

Werden die Verbesserungen angewendet und beachtet man die Einschränkungen mit den Texturen, die jedes passive Stereoskopie-System besitzt, dann können qualitativ hochwertige Tiefenbilder und Punktwolken in annehmbarer Zeit generiert werden.

Ebenfalls wäre es möglich, andere Ansätze als den in StereoVision verwendeten merkmalbasierten Ansatz zu verwenden.

Auch wenn der Treiber nicht die Leistung für den gedachten Einsatzzweck besitzt, kann er mit einigen Verbesserungen für andere Zwecke verwendet werden. Vorstellbar wären die Erstellung einer groben Übersicht über einen Platz durch einen Roboter oder die Abstanderkennung für zeitunkritische Systeme.

Literaturverzeichnis

[AKB08] Agrawal, Motilal ; Konolige, Kurt ; Blas, Morten R.: Censure: Center surround extremas for realtime feature detection and matching. In: *Computer Vision–ECCV 2008*. Springer, 2008, S. 102–115

[AMN+98] Arya, Sunil ; Mount, David M. ; Netanyahu, Nathan S. ; Silverman, Ruth ; Wu, Angela Y.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. In: *J. ACM* 45 (1998), November, Nr. 6, 891 - 923. http://doi.acm.org/10.1145/293347.293348. – ISSN 0004–5411

[AMU] AMUSEMENT.NET: *Spandex and a Kinect make for a great interactive installation |AMUSEMENT.NET*. http://tinyurl.com/ot9fuwd

[AOV12] Alahi, Alexandre ; Ortiz, Raphaël ; Vandergheynst, Pierre: FREAK: Fast Retina Keypoint. Rhode Island, Providence, USA : IEEE Conference on Computer Vision and Pattern Recognition, Juni 2012 (IEEE Conference on Computer Vision and Pattern Recognition New York: Ieee, 2012)

[Bli08] Blinkenlights: *Stereoscope*. Website. http://blinkenlights.net/stereoscope. Version: 2008

[Bre05] Breuer, Pia: *Entwicklung einer prototypischen Gestenerkennung in Echtzeit unter Verwendung einer IR-Tiefenkamera*, Universität Koblenz-Landau, Diplomarbeit, 2005

[Dyg12] Dygoduk, Aleksej: *Entwicklung einer Android App für 3D Rekonstruktion*. Hamburg, HAW Hamburg, Diplomarbeit, 2012

[Esc06] Eschenburg, Jonas: *Optisches Kameratracking anhand natürlicher Merkmale*. Augsburg, Universität Augsburg, Diplomarbeit, 2006

[FB81] Fischler, Martin A. ; Bolles, Robert C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications of the ACM* 24 (1981), Nr. 6, S. 381–395

[Föl12] Föll, Gregory: *Robuste Tiefenbildgewinnung aus Stereobildern.* Hamburg, HAW Hamburg, Diplomarbeit, 2012

[HZ00] Hartley, R. I. ; Zisserman, A.: *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521623049, 2000

[HZW+10] Humenberger, Martin ; Zinner, Christian ; Weber, Michael ; Kubinger, Wilfried ; Vincze, Markus: A fast stereo matching algorithm suitable for embedded real-time systems. In: *Computer Vision and Image Understanding* 114 (2010), Nr. 11, 1180 - 1202. http://dx.doi.org/http://dx.doi.org/10.1016/j.cviu.2010.03.012. – DOI http://dx.doi.org/10.1016/j.cviu.2010.03.012. – ISSN 1077–3142. – Special issue on Embedded Vision

[Jac09] Jacobi, Dirk: *Identifikation und räumliche Lokalisierung skalierungsinvarianter Merkmale für die visuelle Navigation.* Hamburg, HAW Hamburg, Diplomarbeit, 2009

[KOR00] Kushilevitz, Eyal ; Ostrovsky, Rafail ; Rabani, Yuval: Efficient search for approximate nearest neighbor in high dimensional spaces. In: *SIAM Journal on Computing* 30 (2000), Nr. 2, S. 457–474

[MCUP04] Matas, Jiri ; Chum, Ondrej ; Urban, Martin ; Pajdla, Tomás: Robust wide-baseline stereo from maximally stable extremal regions. In: *Image and vision computing* 22 (2004), Nr. 10, S. 761–767

[ML09] Muja, Marius ; Lowe, David G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: *In VISAPP International Conference on Computer Vision Theory and Applications*, 2009

[NFH11] Nischwitz, Alfred ; Fischer, Max ; Haberäcker, Peter: *Computergrafik und Bildverarbeitung.* 3. Wiesbaden : Vieweg, 2011. – ISBN 978–3–8348–0186–9

[O'S] O'Shea, Chris: *Hand from Above.* Website, PDF, Datasheet. http://www.chrisoshea.org/hand-from-above

[Pet13] Petersen, Iwer: *Using object tracking for dynamic video projection mapping.* Hamburg, HAW Hamburg, Diplomarbeit, 2013

[Poia] Point Grey Research Inc.: *Point Grey Research Inc. - News: Bumblebee in Surgery.* http://www.ptgrey.com/news/casestudies/details.asp?articleID=276

[Poib] Point Grey Research Inc.: *Point Grey Research Inc. - News: Bumblebee Rover*. http://www.ptgrey.com/news/casestudies/details.asp?articleID=379

[Poi12] Point Grey Research Inc.: *Bumblebee - Stereo Vision Camera System*. Website, PDF, Datasheet. http://www.ptgrey.com/products/bumblebee2/bumblebee2_xb3_datasheet.pdf. Version: June 2012

[Ray] Raytrix GmbH: *R5 High Speed Video 3D Light Field Cameras*. http://www.raytrix.de/tl_files/downloads/R5.pdf

[RC98] Roy, Sébastien ; Cox, Ingemar J.: A maximum-flow formulation of the n-camera stereo correspondence problem. In: *Computer Vision, 1998. Sixth International Conference on* IEEE, 1998, S. 492–499

[Rod04] Rodehorst, Volker: *Photogrammetrische 3D-Rekonstruktion im Nahbereich durch Auto-Kalibrierung mit projektiver Geometrie*. Berlin : Wvb, Wiss. Verl., 2004. – ISBN 3936846839 9783936846836

[RRKB11] Rublee, Ethan ; Rabaud, Vincent ; Konolige, Kurt ; Bradski, Gary: ORB: an efficient alternative to SIFT or SURF. (2011). http://www.vision.cs.chubu.ac.jp/CV-R/pdf/Rublee_iccv2011.pdf

[SIC11] SICK: *Detection and Ranging Solutions - Laser measurement technology*. https://www.mysick.com/saqqara/im0044207.pdf. Version: 2011

[SS02] Scharstein, Daniel ; Szeliski, Richard: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: *International journal of computer vision* 47 (2002), Nr. 1-3, S. 7–42

[SUKS12] Senst, Tobias ; Unger, Brigitte ; Keller, Ivo ; Sikora, Thomas: Performance Evaluation of Feature Detection for Local Optical Flow Tracking. In: *ICPRAM (2)*, 2012, S. 303–309

[Sun01] Sun, Changming: Fast Algorithm for Local Statistics Calculation for N-Dimensional Images. In: *Real-Time Imaging* 7 (2001), Nr. 6, S. 519–527

[Sun02] Sun, Changming: Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques. In: *International Journal of Computer Vision* 47 (2002), Nr. 1-3, S. 99–117

[SZS+08] Szeliski, Richard ; Zabih, Ramin ; Scharstein, Daniel ; Veksler, Olga ; Kolmogorov, Vladimir ; Agarwala, Aseem ; Tappen, Marshall ; Rother, Carsten: A comparative study of energy minimization methods for markov random fields with smoothness-based priors. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30 (2008), Nr. 6, S. 1068–1080

[Tet05] Tetzlaff, Olaf: *Tiefenbilder aus Stereo-Bildpaaren mit Hilfe der dynamischen Programmierung*. Hamburg, HAW Hamburg, Diplomarbeit, 2005

[Teu07] Teutsch, Christian: *Model-based analysis and evaluation of point sets from optical 3Dlaser scanners*. Aachen, Shaker, Diss., 2007

[TH98] Trajković, Miroslav ; Hedley, Mark: Fast corner detection. In: *Image and Vision Computing* 16 (1998), Nr. 2, S. 75–87

[TM08] Tuytelaars, Tinne ; Mikolajczyk, Krystian: Local invariant feature detectors: a survey. In: *Foundations and Trends® in Computer Graphics and Vision* 3 (2008), Nr. 3, S. 177–280

[URBa] URBANSCREEN: *URBANSCREEN 555 KUBIK*. `http://www.urbanscreen.com/usc/41`

[URBb] URBANSCREEN: *URBANSCREEN MQ10*. Website. `http://www.urbanscreen.com/usc/1044`

[URBc] URBANSCREEN: *URBANSCREEN PINWALL*. Website. `http://www.urbanscreen.com/usc/31`

[URBd] URBANSCREEN: *URBANSCREEN Sydney*. Website. `http://www.urbanscreen.com/usc/1124`

[Vee] Veenhof, Sander: *INSTANT SCULPTURE GARDEN*. `http://www.sndrv.nl/instantsculpturegarden/`

[Wika] Wikipedia: *Plenoptische Kamera*. `http://de.wikipedia.org/wiki/Plenoptische_Kamera`

[Wikb] Wikipedia: *TOF-Kamera*. `http://de.wikipedia.org/wiki/TOF-Kamera`

[Wik13a] Wikipedia: *Nearest neighbor search*. `http://en.wikipedia.org/w/index.php?title=Nearest_neighbor_search&oldid=558338819`. Version: 2013. – Page Version ID: 558338819

[Wik13b] Wikipedia: *RANSAC-Algorithmus*. `http://de.wikipedia.org/w/index.php?title=RANSAC-Algorithmus&oldid=117296352`. Version: 2013. – Page Version ID: 117296352

[Zha98] Zhang, Zhengyou: Determining the epipolar geometry and its uncertainty: A review. In: *International Journal of Computer Vision* 27 (1998), Nr. 2, S. 161–195

[ZN09] Zhao, Wan-Lei ; Ngo, Chong-Wah: Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection. In: *Image Processing, IEEE Transactions on* 18 (2009), Nr. 2, S. 412–423

Abbildungsverzeichnis

Tabellenverzeichnis

A. Anhang

A.1. Epipolargeometrie

Abb. A.1 zeigt die Epipolargeometrie für die Aufnahme mit zwei Kameras. Die Projektionszentren, O_L und O_R, der beiden Kameras und ein Weltpunkt X spannen die Epipolarebene auf. Die Punkte e_L und e_R, in denen die Basislinie die Bildebenen durchstößt, werden als Epipole bezeichnet. Geraden, die durch die Schnittpunkte der Epipolarebene mit der jeweiligen Bildebene verlaufen, bezeichnet man als Epipolarlinien. Jeder Punkt innerhalb der Epipolarebene, wird auf dieselben Epipolarlinien in den Bildebenen projiziert.[Zha98]

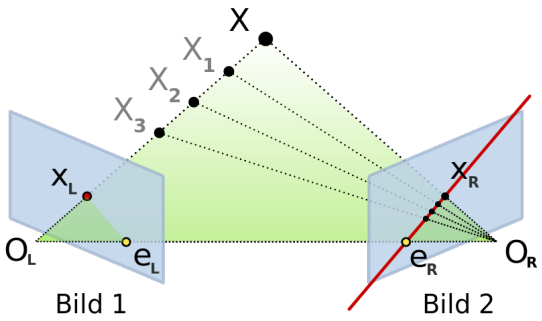

Abbildung A.1.: Darstellung der Epipolarebene

Mittels Transition und Rotation kann das Koordinatensystem der einen Kamera in das Koordinatensystem der anderen Kamera überführt werden. Dabei entspricht der Translationsvektor der Basislinie, die durch O_L und O_R verläuft. Die Beziehung zwischen den beiden Kameras ist die Epipolargeometrie.

A.2. Trifokalgeometrie

Nutzt man mehr als zwei Kameras, so kann man den Trifokaltensor verwenden. Der durch die Trifokalgeometrie, eine Erweiterung der Epipolargeometrie auf drei Bilder, definierte Tri-

fokaltensor beschreibt die Beziehung zwischen drei Bildern eines Objektes aus verschiedenen Ansichten. Ist also die Position eines Objektpunktes in zwei Bildern gegeben, so ist seine Position im dritten Bild der „Schnittpunkt der beiden Epipolarlinien in diesem Bild“.

Somit existiert ein eindeutiges Ergebnis, wenn der Punkt nicht in der Trifokalebene liegt[1].

A.3. RANSAC-Algorithmus

Durch Bildrauschen oder sehr ähnliche Merkmalsvektoren kann es bei der Korrespondenzsuche zu Fehlzuordnungen kommen. Mithilfe eines robusten Ausgleichsverfahrens können falsche Zuordnungen eliminiert werden. Der RANSAC-Algorithmus[2] ist solch ein robustes Ausgleichsverfahren. Die Idee in ihm besteht darin, dass einige wenige falsche Werte die Ausgleichsgerade[3] stark beeinflussen können und diese Werte aussortiert werden müssen, wie in Abb A.2 gezeigt.[Wik13b]

Der Algorithmus wählt zufällig zwei Punkte aus der Menge aller Messwerte aus und berechnet aus ihnen die Ausgleichsgerade. Zunächst wird davon ausgegangen, dass die gewählten Punkte valide sind. Anschließend wird die Gerade mit allen übrigen Punkten verglichen und solange der Abstand einen Grenzwert nicht überschreitet, gehören sie in die aktuelle Teilmenge. Nun werden diese Schritte einige Male wiederholt und die Teilmenge mit der größten Anzahl an Punkten wird für die weiteren Berechnungen verwendet.

Der Algorithmus greift nicht, wenn es zu viele fehlerhafte Messwerte gibt, da dann nicht mehr zwischen gewünschten und nicht gewünschten Werten unterschieden werden kann.

A.4. SAD - Sum of Absolute Difference

Sum of Absolute Difference ist ein Maß zur Ähnlichkeit von Bildern oder Bildregionen. Es werden dabei die Farbwerte von jeweils zwei Punkten verglichen und der Betrag der Differenz wird aufsummiert. Je kleiner die Summe ist, desto ähnlicher sind sich die verglichenen Bilder oder Regionen.[Föl12]

[1] Die Ebene, die aus den drei Projektionszentren gebildet wird.
[2] Random Sample Consensus
[3] In diesem Beispiel wird eine Ausgleichsgerade zur Vereinfachung gewählt.

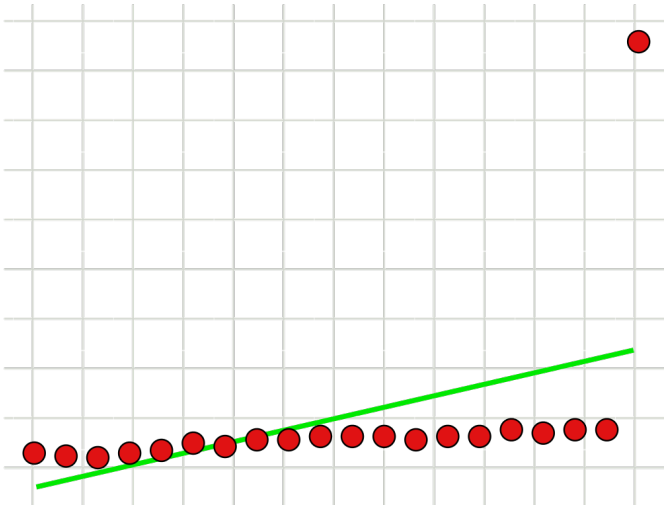

Abbildung A.2.: Anwendungsmöglichkeit des RANSAC-Algorithmus - Ein fehlerhafter Messwert beeinflusst die Ausgleichsgerade zu stark und muss aussortiert werden

$$SAD(Img_1, Img_2, u_1, v_1, u_2, v_2, n) := \sum_{i=-n}^{n} \sum_{j=-n}^{n} \mid Img_1(u_1{+}i, v_1{+}j) {-} Img_2(u_2{+}i, v_2{+}j) \mid$$

Img_1 und Img_2 zwei Bilder oder Bildausschnitte repräsentieren. u_1 und v_1 sowie u_2 und v_2 sind die Koordinaten des Pixels im Zentrum der Bildregion. n ist gleich dem Radius der Region.

A.5. ZNCC - Zero Mean Normalized Cross-Correlation

Auch ZNCC ist ein Algorithmus, mit dessen Hilfe man zwei Bilder oder Bildregionen vergleichen kann, um ihre Ähnlichkeit zu bestimmen.

Im ersten Schritt wird der durchschnittliche Grauwert des Bildes und die Standardabweichung ermittelt.

$$\overline{Img}(u, v, n) := \frac{1}{(2n+1)^2} \sum_{i=-n}^{n} \sum_{j=-n}^{n} Img(u+i, v+j)$$

$$\sigma(u,v,n) := \sqrt{\frac{1}{(2n+1)^2}\left(\sum_{i=-n}^{n}\sum_{j=-n}^{n}(Img(u+i,v+j)-\overline{Img}(u,v,n))^2\right)}$$

Anschließend kann der ZNCC wie folgt berechnet werden:

$$ZNCC(Img_1, Img_2, u_1, v_1, u_2, v_2, n) :=$$

$$\frac{\frac{1}{(2n+1)^2}\sum_{i=-n}^{n}\sum_{j=-n}^{n}\prod_{t=1}^{2}\left(Img_t(u_t+i,v_t+j)-\overline{Img}(u_t,v_t,n)\right)}{\sigma_1(u_1,v_1,n)\cdot\sigma_2(u_2,v_2,n)}$$

Img_1 und Img_2 zwei Bilder oder Bildausschnitte repräsentieren. u_1 und v_1 sowie u_2 und v_2 sind die Koordinaten des Pixels im Zentrum der Bildregion. n ist gleich dem Radius der Region.[Sun01][Sun02]

Je höher der Wert ist, desto ähnlicher sind sich die Bilder oder Regionen. ZNCC ist im Gegensatz zu SAD wesentlich komplexer in der Berechnung, aber dafür auch zuverlässiger bei dem Vergleich von Bildern mit unterschiedlicher Helligkeit.