Hochschule Neubrandenburg
University of Applied Sciences

Hochschule Neubrandenburg

Studiengang Geodäsie und Geoinformatik

Geometrische Analyse von industriellen Schmiedestücken mit Computer Vision durch 3D Rekonstruktion

Masterarbeit

vorgelegt von: *Nico Schlegel*

Zum Erlangen des akademischen Grades

„Master of Engineering“ (M.Eng.)

Erstprüfer: Prof. Dr.-Ing. Sven Brämer

Zweitprüfer: Prof. Dr.-Ing. Thomas Willemsen

Eingereicht am: 25.09.2020

urn:nbn:de:gbv:519-thesis 2020-0142-1

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Bachelor-Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Ort, Datum Unterschrift

Inhaltsverzeichnis

1. Einführung 1
   1.1 Einsatz von Computer Vision in der heutigen Industrie 2
   1.2 Aufgabenstellung, Methodik, Abgrenzung des Themas 2
   1.3 Verwendete Soft – und Hardware 3
2. Anforderungen durch das industrielle Umfeld 5
   2.1 Beleuchtung 5
   2.2 Helligkeit und Kontrast 7
3. 3D Rekonstruktion 8
   3.1 Bestimmung der dreidimensionalen Lage durch eine einzelne Kamera 9
   3.2 Stereo Vision 11
      3.2.1 Triangulation 11
      3.2.2 Epipolare Geometrie 14
      3.2.3 Fundamentale Matrix 15
      3.2.4 Stereo Kalibrierung 16
      3.2.5 Stereoentzerrung 17
      3.2.6 Stereo Korrespondenz 18
   3.3 Time of Flight 24
      3.3.1 Prinzip einer TOF-Kamera 25
      3.3.2 Ableitung einer Punktwolke 27
   3.4 Lichtschnittverfahren 28
      3.4.1 Grundprinzip 28
      3.4.2 Aufbau und Auslegung eines Lichtschnittsensors 29
   3.5 Strukturierte Beleuchtung 30
      3.5.1 Funktionsweise 31
      3.5.2 Techniken der sequenziellen Projektion 33
         3.5.2.1 Binäre Muster und Graukodierung 33
         3.5.2.2 Graustufen-Muster 33
         3.5.2.3 Phasenverschiebung 34
   3.6 Structure from Motion 37
      3.6.1 Triangulation 37
      3.6.2 Multi view structure from motion 38
      3.6.3 Bündelausgleichung 39
4. Vermaschungsalgorithmen 40
   4.1 Delaunay 40
      4.1.1 Zweidimensional 40
      4.1.2 Dreidimensional 41

4.2 Poisson 42
5. Kalibrierung des Systems 43
5.1 Kameramodell 44
5.2 Drehungsmatrix Verschiebungsvektor 45
5.3 Kalibriertafeln 47
5.4 Homographie 49
6. Anwendung auf industrielle Schmiedestücke 51
6.1 Wahl eines geeigneten Algorithmus 51
6.2 Realisierter Lösungsansatz 57
6.2.1 Kamerakalibrierung 59
6.2.2 Bildvorverarbeitung 66
6.2.2.1 Farbfilterung 66
6.2.2.2 Maskierung 69
6.3 Möglichkeiten zur Einführung eines Maßstabes 73
6.3.1 Schachbrett 73
6.3.2 Marker 74
6.4 Berechnung eines Drahtgittermodells 74
7. Bewertung 75
7.1 Erzielte Ergebnisse 75
7.2 Kritische Diskussion und Alternativen 95
8. Zusammenfassung 96
Literaturverzeichnis

Kurzfassung

Die vorliegende Arbeit gibt einen Überblick über den Einsatz der digitalen Bildverarbeitung in der heutigen Industrie. Dabei wird speziell die Anwendung der 3D-Rekonstruktion durch Computer Vision auf die Thematik der geometrischen Analyse von industriellen Schmiedestücken untersucht. Erläutert werden dabei die Anforderungen durch das industrielle Umfeld und die Funktionsweise verschiedener Verfahren, welche eine 3D-Rekonstruktion von Objekten ermöglichen. Insbesondere wird dabei auf die benötigte Rechenzeit geachtet. Die Anwendung der 3D-Rekonstruktion wird im Rahmen eines realisierten Lösungsansatzes und eines simulierten Beispiels aufgezeigt.

Abstract

This thesis gives an overview of the use of digital image processing in today's industry. In particular, the application of 3D reconstruction by computer vision to the topic of geometric analysis of industrial forgings is examined. The requirements by the industrial environment and the function mode of different procedures, which make a 3D reconstruction of objects possible, are explained thereby. Particular attention is paid to the required computing time. The application of the 3D reconstruction is shown in the context of a realized approach and a simulated example.

1. Einführung

Computer Vision ist eine Anwendung der KI, die weit über das bloße Sehen und das Verständnis des Inhalts digitaler Bilder hinausgeht. Sie spielt eine wichtige Rolle bei der digitalen Transformation über verschiedene Industriezweige. Computer Vision erfasst, verarbeitet, analysiert und versteht digitale Bilder und Videos. Sie ermöglicht es Computern und Maschinen, gemäß den entworfenen Algorithmen, zu sehen und zu erkennen. Der Einsatz von Computer Vision richtet sich an einzigartige geschäftliche Herausforderungen und verbessert die Unternehmensleistungen durch Bildverarbeitung, Bildklassifizierung, Objekterkennung, Merkmals- und Mustererkennung, 3D Bildrekonstruktion und Videoanalyse. [Saxena, 2019]

Produktionslinien in High-Tech-Produktionseinheiten sind hochkomplex und automatisiert. Computer Vision-Technologien in Produktionseinheiten sind sehr nützlich und haben beispiellose Vorteile für das Unternehmen. Moderne Produktionseinheiten sind mit sehr teuren automatisierten Produktionslinien mit Förderanlagen und Robotern ausgestattet. Ein unerwarteter Ausfall der Produktionslinie kann für das Unternehmen schwere Folgen haben. Hier kommt die Rolle der Computer Vision ins Spiel, die jede Komponente der Produktionslinie analysiert und selbst die kleinsten Fehler im System diagnostiziert. Auf Grundlage dieser detaillierten und präzisen Untersuchung können Computer Vision-Systeme dazu beitragen im Vorfeld zu warnen und so die Ausfallzeit der Produktion zu senken.

Die Prüfung auf Fehler in der industriellen Einrichtung kann sehr riskant, kostspielig und zeitaufwendig sein. Zum Teil ist es unmöglich Fehler an den Maschinen manuell zu erkennen. Computer Vision-Technologien helfen in solchen Fällen, Risiken für die Arbeiter zu eliminieren und arbeiten präzise, um Risse, Korrosion, Lecks und andere Anomalien an den Maschinen zu erkennen. Industrien wie die Automobil-, Pharmazie-, Textil-, oder Energieindustrie verwenden Computer Vision-Technologien, um ihre Prozesse fehlerfrei ablaufen zu lassen.

Vom Fließband über den Fahrzeugherstellungsprozess bis hin zu den Fahrzeugen auf der Straße beweist Computer Vision ihre Akzeptanz und Präsenz, indem sie das Fahren jeden Tag sicherer macht. In konventionellen Produktionseinheiten ist die Qualitätsprüfung der Produkte der letzte Schritt im Produktionszyklus, der mit der Produktionskapazität, der Zeit, dem Arbeitsaufwand und den Kosten der Produktionseinrichtung Kompromisse eingeht. Mit der Einführung von Computer Vision, unterstützt durch AI- und ML-Techniken, wird die Qualität der Produkte mit ihrer Verpackung in jeder Phase der Produktion mit hoher Präzision und Genauigkeit geprüft, was zu einer minimalen Verschwendung von Produkt, Kosten und Aufwand führt. [Saxena, 2019]

1.1 Einsatz von Computer Vision in der heutigen Industrie

Ein großes Anwendungsgebiet von Computer Vision ist die Verwendung im industriellen Fertigungsprozess. Computer Vision bezieht sich auf die Prozesse und Technologien, die Maschinen dabei helfen, die Welt ähnlich wie Menschen zu ‚sehen', indem sie den Kontext interpretieren und verstehen. Der Unterschied zwischen einer Maschine und einem Menschen besteht darin, dass Algorithmen Informationen verarbeiten, indem sie diese in numerische Modelle umwandeln. Obwohl Computer Vision seinen Ursprung in den späten fünfziger Jahren hatte, gab es in den letzten Jahren einen exponentiellen Anstieg der Verwendung von Computer Vision.

Auch wenn Computer Vision die Möglichkeit bietet Geschäftsabläufe zu verändern hat es die Technologie immer noch schwer mit kleinen visuellen Variationen fertig zu werden. Jeder Schritt des Prozesses muss verbessert werden. Zum Klassifizieren von Bildern sind umfangreiche Bibliotheken mit Millionen von markierten Bildern erforderlich, die aus verschiedenen Blickwinkeln, mit Farbsättigungs- und Beleuchtungsvariationen aufgenommen wurden. Die Segmentierung erfordert eine Verfeinerung bei der Erkennung von Objektgrenzen und der Farbkodierung. Mit der allmählichen Verbesserung der Technologie wird sie jedoch in immer mehr Bereiche vordringen und den Fachleuten helfen, Entscheidungen schneller und sicherer zu treffen und den Kunden und Anwendern effizientere Alternativen zu bieten. [Brooke,2019]

1.2 Aufgabenstellung, Methodik und Abgrenzung des Themas

Ziel dieser Arbeit ist die Anwendung der 3D Rekonstruktion auf die Thematik der geometrischen Analyse von industriellen Schmiedestücken. Es soll ein Algorithmus gewählt und in einem Programm umgesetzt werden der es ermöglicht mit Hilfe von Computer Vision 3D Informationen des Objektes abzuleiten. Dazu werden verschiedene Techniken zur Gewinnung von 3D Informationen betrachtet. Untersucht wird dabei die Anwendbarkeit dieser Techniken auf industrielle Schmiedestücke.

Im Rahmen der vorliegenden Arbeit wird ein Programm entwickelt, welches mit QT als IDE geschrieben wird. Die Programmiersprache ist C++. Als Grundlage für die Bildverarbeitung wird die Open Source Library OpenCV verwendet.

Der Hintergrund ist eine Automatisierung und Optimierung des Arbeitsablaufes von Freiformschmiedemaschinen zu erreichen. Schmiedemaschinen nutzen Druckkraft und die hohe Werkstücktemperatur, um Metalle durch plastische Verformung in Form zu bringen Das Warmschmieden ist ein metallurgisches Verfahren, bei dem Metallteile geformt werden, indem heißes Metall unter Druck in Gesenke gepresst wird. Eine Art der Warmumformung, das

Warmschmieden, beinhaltet die plastische Verformung eines Metalls bei einer Temperatur und Dehnungsgeschwindigkeit, bei der Rekristallisation und Verformung gleichzeitig stattfinden. Dadurch wird eine Kälteverfestigung verhindert, die die Duktilität verringern und zu Sprödbruch und Strukturermüdung beitragen kann. [Halama, o.J.]

Der Faktor Zeit spielt bei diesen Arbeitsschritten eine entscheidende Rolle. Deshalb soll bei der Untersuchung der verschiedenen Techniken zur Gewinnung von 3D Informationen auch die zeitliche Komponente betrachtet werden. Das zeitliche Limit liegt bei drei Sekunden. Zusätzlich sollen Algorithmen entwickelt werden, um die Gewinnung von 3D Informationen weiter zu beschleunigen und die Anwendung für den Nutzer zu vereinfachen.

Da keine Genauigkeitsanforderungen von einem Hersteller vorliegen, wird, durch die Einflüsse von Schrumpfung, Fließen und Sinterbildung , welche im Nachgang durch Drehen und Fräsen beseitigt werden, eine Genauigkeitsanforderung von 1-2 mm angenommen.

1.3 Verwendete Soft- und Hardware

Als Grundlage für die Bildverarbeitungsroutinen dient die Open Source-Bibliothek OpenCV. In der vorliegenden Arbeit wird die Version 4.10 verwendet. Ins Leben gerufen wurde OpenCV im Jahre 1999 von Intel. Durch die Bereitstellung einer soliden Basis für Computer Vision, welche für jeden frei zugänglich ist, erhoffte man sich, die Entwicklung von Computer Vision und künstlicher Intelligenz zu beschleunigen. Die Bibliothek ist in den Programmiersprachen C und C++ geschrieben und läuft unter Linux, Windows und Mac OS X. Zusätzlich gibt es Schnittstellen für Java, Python, Matlab und andere Sprachen. Eine Portierung auf die Plattformen Android und iOS ist für mobile Anwendungen möglich. OpenCV wurde unter dem Gesichtspunkt Recheneffizienz und mit dem Fokus auf Echtzeitanwendungen entwickelt. Daher ist es in der Programmiersprache C++ geschrieben, welche auch im Rahmen dieser Arbeit genutzt wird. Durch die Bereitstellung einer einfach zu bedienenden Computer Vision-Struktur wie OpenCV, ist es möglich, schnell anspruchsvolle Bildverarbeitungsanwendungen zu erstellen. Seit der Alpha-Veröffentlichung 1999 wurde OpenCV in vielen Anwendungen eingesetzt. Zu diesen Anwendungen gehört das Zusammenfügen von Bildern in Satelliten- und Webkarten, Bild-Scan-Alignement, Bildrauschunterdrückung in der Medizintechnik, Objektanalyse, Sicherheitssysteme, automatische Überwachung, Fertigungsinspektionssysteme, Kamerakalibrierung und optische Sensoren für unbemannte Luft-, Boden- und Unterwasserfahrzeuge.

Bei Computer Vision geht es häufig um Datentransformationen. Helligkeitswerte einer Kamera werden durch Transformationen so aufbereitet, dass eine neue Darstellung der Ausgangsdaten entsteht. Ein Beispiel dafür ist das Transformieren eines Farbbildes in ein

Graustufenbild. Ein Bildverarbeitungssystem empfängt nur eine Zahlenmatrix von der Kamera. In Abbildung 1 wird das deutlich.

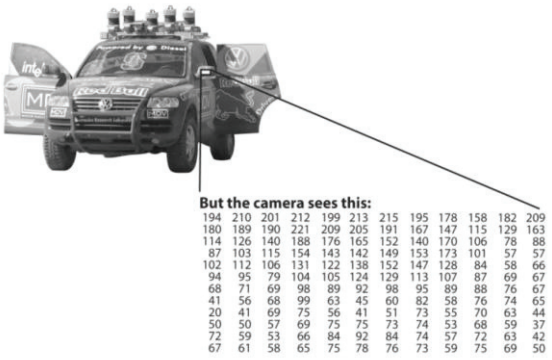

Abbildung 1: Warnehmung des Autos in Form eines Zahlenrasters [Kaehler et al., 2017]

Der Mensch erkennt ein Auto und im Detail den Seitenspiegel auf der Fahrerseite. Der Computer bekommt von der Kamera ein Zahlenraster, welches noch recht wenig Informationen bezüglich der gewünschten Aussagen enthält. Es beinhaltet lediglich Helligkeitswerte. Dem Computer muss erst ‚beigebracht' werden, dass es sich hier um einen Seitenspiegel handelt. Das ist die Aufgabe von Computer Vision. [Kaehler et al., 2017]

Wenn Bildverarbeitungsanwendungen auf Schnelligkeit oder Echtzeitdatenverarbeitung ausgelegt werden sollen, empfiehlt es sich eine Programmiersprache zu wählen, welche eine geringe Rechenzeit hat. Hier bietet sich C++ an. Die OpenCV-Bibliothek ist in dieser Sprache geschrieben. Bei der Verwendung von C++ auch für die eigentliche Hauptanwendung entfällt die Notwendigkeit von Schnittstellen zu einer anderen Programmiersprache. Das spart Prozessor-Zeit, wodurch die Bildverarbeitung schneller wird.

Eine Entwicklungsumgebung, die sowohl die Programmiersprache C++ verwendet und es ermöglicht OpenCV zu integrieren und zu nutzen, ist QT. Dabei handelt es sich um eine C++ Klassenbibliothek für die plattformübergreifende Erstellung und Programmierung von grafischen Benutzeroberflächen (GUIs). Sie bietet den Vorteil, dass man eine Anwendung auf verschiedene Plattformen übersetzen und dort ausführen kann. In der Klassenbibliothek sind sämtliche notwendige Bestandteile des Quellcodes für die Einbindung der OpenCV-Routinen vorhanden.

In dieser Arbeit wird die Version QT5.11.2 verwendet. QT verwendet einen Präprozessor (MOC – meta object compiler), womit C++ erweitert wird. Die wichtigste Erweiterung ist dabei das Konzept der Slots uns Signale. Damit bietet sich die Möglichkeit auf Ereignisse zu reagieren, wie zum Beispiel das Auslösen einer Aktion durch das Drücken eines Buttons, um ein Fenster zu schließen. Damit eine Verbindung zwischen dem Button und einer Aktion vermittelt werden kann, wird das Konzept der Signale und Slots verwendet. In Verbindung mit

der OpenCV-Bibliothek können mit QT anschauliche Bildverarbeitungsanwendungen erstellt werden. [Steyer, 2015]

2. Anforderungen durch das industrielle Umfeld

In der Industrie gibt es gewisse Anforderungen, die an die Bildverarbeitungssoft- und hardware gestellt werden. Eine dieser Anforderungen ist die zeitliche Einschränkung durch die Schmiedemaschine. Daher muss die Bildverarbeitungssoftware der Geschwindigkeitsanforderung von drei Sekunden gerecht werden und schnell genug sein, um die anfallenden Daten zu verarbeiten. Ist das System zu langsam, kann es für den Produktionsablauf nicht verwendet werden und macht es somit unbrauchbar. Die Schmiedemaschine kann nicht auf die Bildverarbeitungssoftware warten.

Eine weitere Anforderung ist die anwenderorientierte Bedienbarkeit mit einfacher und aussagekräftig visualisierter Ergebnisausgabe. Die anwenderorientierte Bedienbarkeit wird durch eine benutzerfreundliche Einrichtung des Systems gewährleistet. Die aussagekräftig visualisierte Ergebnisausgabe gestaltet sich durch die Darstellung der abgeleiteten 3D-Informationen des Schmiedestücks, statt der Ausgabe einer Liste mit Koordinaten in Textform.

Die im weiteren Verlauf der Arbeit erläuterten Verfahren zur Ableitung von dreidimensionalen Informationen eines Schmiedestücks werden teilweise durch Aufnahmen einer Kamera, in Verbindung mit einer Featuredetection, realisiert. Damit die Features bestimmt werden können, ist ein in Hinsicht auf Kontrast und Helligkeit aussagekräftiges Bild notwendig. Features werden auch Keypoints (dt.: Schlüsselpunkte) genannt. Diese können durch das Arbeitsumfeld in Form von Funkenflug, glühenden Metallen oder ungleichmäßige Beleuchtung beeinträchtigt werden. Für eine Verbesserung der Kontrast- und Helligkeitsverhältnisse auf den Kameraaufnahmen, gibt es verschiedene Möglichkeiten. Zum einen durch eine geeignete Beleuchtung des Objektes, zum anderen ist es möglich Kontrast und Helligkeit zusätzlich durch die Veränderung des Histogramms anzupassen.

2. 1 Beleuchtung

Die Beleuchtung des Objektes während des Arbeitsprozesses ist ausschlaggebend für die Qualität der Kameraaufnahmen. Je höher die Qualität der Kameraaufnahmen ist, desto besser gelingt die anschließende Bildauswertung. Mit der Bildverarbeitungssoftware wird im Grunde genommen das visuelle Abbild des Objektes überprüft und nicht das Objekt selbst. Als Grundbedingung gilt daher, dass die Beleuchtung von Arbeitsschritt zu Arbeitsschritt nicht exakt gleich sein muss, jedoch stabil und reproduzierbar genug, um ausreichend viele Features auf den Kameraaufnahmen detektieren zu können. Somit wird gewährleistet, dass

dieselben Features bei der Verwendung von mehreren Kameras, auch in mehreren Bildern ausreichend gut zu erkennen sind. Nur dann entstehen keine Diskrepanzen, was eine fehlerhafte Verknüpfung der Bilder zueinander zur Folge hätte. Ziel ist es, mit der Beleuchtung diese Features durch genügend Kontrast für die Bildverarbeitungssoftware überhaupt erst identifizierbar zu machen. Die Wahl der Beleuchtung ist in erster Linie abhängig von dem Schmiedestück. Da Eigenschaften wie Größe oder Reflexionsgrad des Objektes selten beeinflussbar sind, muss der gezielten Auswahl der Beleuchtungsmethode entsprechende Aufmerksamkeit zukommen (Abbildung 2).

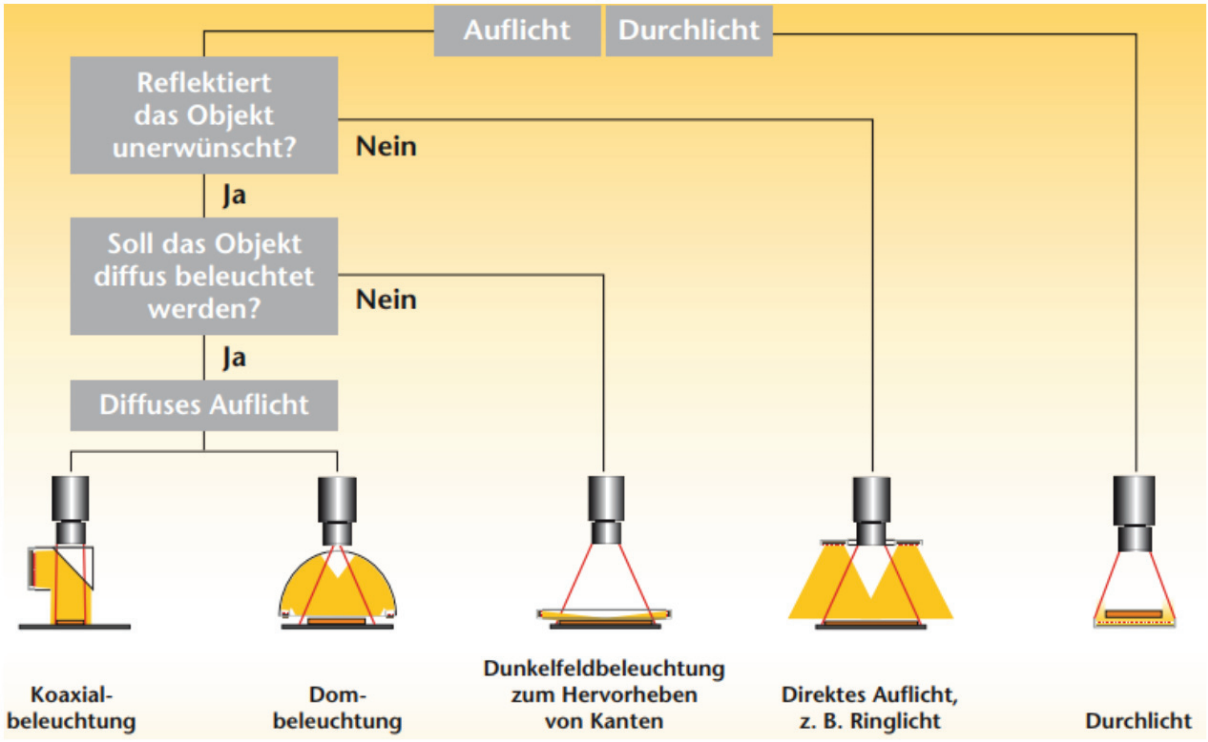

Abbildung 2: Auswahlschema für die geeignete Beleuchtung [Polytec, 2010]

Steht die Vermessung der Kontur eines Objektes im Vordergrund fällt die Wahl oft auf die Durchlichtbeleuchtung. Bedingung dabei ist, dass die Beleuchtung unterhalb des Objektes positioniert werden kann. Durch diese Art der Beleuchtung entsteht ein kontrastreiches Schattenbild des Objektes für die Kameraaufnahme von oben. Als Alternative kann auch die Dunkelfeldbeleuchtung eingesetzt werden. Durch das Hervorheben von Objektkanten bei der Dunkelfeldbeleuchtung ist es möglich, die Konturen des Objektes zu vermessen.

Bei der Auflichtbeleuchtung trifft das Licht direkt auf das Objekt. Realisiert wird die Auflichtbeleuchtung meist durch ein Ringlicht. Zum Einsatz kommt sie bei Objekten mit geringem Reflexionsgrad. Die Ringgröße der Beleuchtung wird an das zu messende Objekt angepasst. Handelt es sich um längliche oder schmale Objekte, so kommen Linienlichter zum Einsatz. Diese können parallel oder quadratisch und somit passend zum Objekt angeordnet werden.

Im Falle von unerwünschtem Schattenwurf bei räumlichen Objekten ist eine koaxiale Beleuchtung besser geeignet als ein Ringlicht oder Linienlicht. Hier wird der Schattenwurf durch die Parallelität von Bildstrahlen und Licht erheblich gemindert. Komplett beseitigen lässt er sich jedoch nicht immer.

Bei hohem Reflexionsgrad wird das Objekt mit einer Dombeleuchtung ausgeleuchtet. Dabei entsteht diffuses Licht, ähnlich wie bei einem bewölkten Himmel, und Reflexionen werden gemindert. Speziell für die Anwendung auf industrielle Schmiedestücke steht die allgemeine Ausleuchtung des Objektes im Vordergrund. Somit wird verhindert, dass gewisse Bereiche zu dunkel sind oder sich im Schatten befinden und sich so nicht rekonstruieren lassen.

Ein weiteres Kriterium, welches zur besseren Qualität der Kameraaufnahmen beitragen kann, ist die Lichtfarbe. Je nachdem, ob es sich um Farb- oder Schwarz-Weiß-Aufnahmen handelt, gibt es Unterschiede. Da viele Schwarz-Weiß-Kameras im roten Wellenlängenbereich am empfindlichsten sind, eignet sich hier rotes Licht besser als weißes Licht. Bei Farbkameras hingegen ist weißes Licht notwendig. [Polytec, 2010]

2. 2 Helligkeit und Kontrast

Helligkeit wird im Kontext der Bildverarbeitung üblicherweise als Maß für die Gesamtintensität des Bildes verstanden. Sie bestimmt, wie hell oder dunkel ein Bild dem Betrachter erscheint. Helligkeit zu verändern, ist eine additive Operation (Abbildung 3). Durch Addition eines positiven oder negativen Wertes zu den Pixelwerten kann die Bildhelligkeit angepasst werden. Eine gängige Methode, die Statistik eines Bildes zu visualisieren, ist das Histogramm. Das Histogramm zeigt die Grauwertverteilung des Bildes. Dabei handelt es sich um ein Balkendiagramm, wobei die x-Achse die Graustufenwerte und die y-Achse die Anzahl der Pixel mit diesem Grauwert ist.

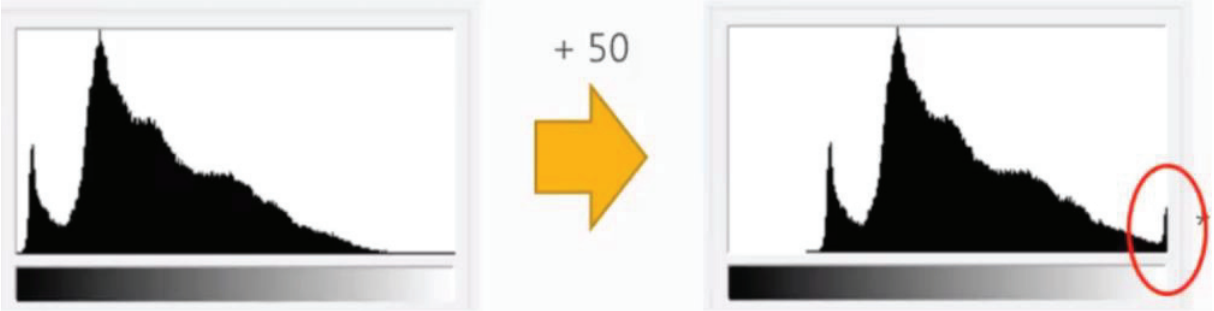

Abbildung 3: Histogramm Helligkeitsbearbeitung [Shavit, 2017]

Auf der linken Seite befindet sich das Histogramm des Originalbildes. Die Erhöhung der Helligkeit des Bildes um 50 bedeutet, dass zu jedem Pixelwert der Wert 50 dazu addiert wird. Somit verschiebt sich das gesamte Histogramm um 50 nach rechts. Die Breite der

Histogrammverteilung ändert sich nicht. Die Breite wird als dynamischer Bereich des Bildes bezeichnet. Jeder Pixelwert, der über den maximalen Grauwert verschoben wird, staut sich beim Wert 255 an, wodurch eine Überbelichtung entsteht. Dadurch kann es zum Verlust von Bildinformationen in den überbelichteten Pixeln kommen.

Kontrast hingegen gibt an, wie dunkel Bereiche des Bildes im Vergleich zu helleren Bereichen sind. Durch Erhöhung des Kontrastes des Originalbildes entsteht ein Bild, in dem die Unterschiede zwischen den dunklen und hellen Bereichen ausgeprägter dargestellt werden. Kontrastveränderung ist eine multiplikative Operation. Der Kontrast eines Bildes wird durch das Multiplizieren aller Pixelwerte mit einem Faktor verändert (Abbildung 4). Dies hat den Effekt, dass dunklere Pixel weniger von der Multiplikation betroffen sind als hellere Pixel mit größeren Werten.

Abbildung 4: Histogramm Kontrastbearbeitung [Shavit, 2017]

Durch Erhöhung des Kontrastes um den Faktor zwei wird das Histogramm nach rechts gestreckt. Durch Verringerung des Kontrastes um die Hälfte, also durch Multiplikation aller Pixelwerte um den Faktor 0.5, schrumpft das Histogramm und wird nach links verlagert. Im Vergleich zur Helligkeit verändert der Kontrast den dynamischen Bereich des Bildes. Die Gefahr der Überbelichtung besteht hier genauso wie bei der Veränderung der Helligkeit. [Shavit, 2017]

Diese Aspekte sind teilweise die Grundlage für Techniken zur Gewinnung von 3D-Informationen sowie die dreidimensionale Rekonstruktion von Objekten durch den Einsatz von Computer Vision.

3. 3D-Rekonstruktion

Zur Gewinnung von 3D-Informationen eines Objektes gibt es diverse Techniken. Das Problem der Abschätzung der Lage von Objekten im dreidimensionalen Raum kann sowohl mit einer einzigen Kamera als auch mit mehreren Kameras angegangen werden. Im Falle von mehreren Kameras werden Korrespondenzen verwendet zwischen dem, was von den einzelnen Kameras gesehen wird, um Rückschlüsse darauf zu ziehen, wo sich das Objekt befindet (z.B. durch Triangulation). Der Vorteil einer solchen Technik besteht darin, dass sie auch mit unbekannten Objekten oder Szenen funktioniert. Der Nachteil ist, dass sie mehrere Kameras erfordert. Es ist jedoch möglich die räumliche Lage eines bekannten Objektes mit nur einer

Kamera abzuleiten. Das Verständnis des Problems zur Abschätzung der dreidimensionalen Lage mit nur einer Kamera ist nicht nur an sich eine nützliche Technik, sondern gibt auch wichtige Einblicke in das Problem der Verwendung mehrerer Kameras. [Kaehler et al., 2017]

3.1 Bestimmung der dreidimensionalen Lage durch eine einzelne Kamera

Die Abschätzung der 3D-Geometrie mit einer einzelnen Kamera ist ein Spezialfall der bildbasierten 3D-Rekonstruktion aus mehreren Bildern, ist aber wesentlich schwieriger, da die Tiefe nicht aus Pixelkorrespondenzen abgeschätzt werden kann. Daher sind weitere Vorkenntnisse oder Benutzereingaben erforderlich, um Tiefeninformationen zu gewinnen oder abzuleiten. [Oswald et. al., 2017]

Mehr als die meisten anderen Probleme der Computer Vision, ist die Rekonstruktion aus einer einzigen Ansicht ein höchst ungünstig platziertes Problem. [Oswald et. al., 2012]

Wie dieses Problem gelöst wird geht aus Abbildung 5 hervor. Ein Objekt ist in dem Maße ‚bekannt', wenn eine gewisse Anzahl von Schlüsselpunkten auf dem Objekt vorhanden und identifizierbar ist. Ein Schlüsselpunkt beschreibt ein markantes Merkmal auf dem Objekt. Die Lage dieser Schlüsselpunkte im Koordinatensystem des Objektes ist ebenfalls bekannt. Wenn dasselbe Objekt aus einer anderen Richtung betrachtet wird, ist es möglich nach denselben Schlüsselpunkten zu suchen. Voraussetzung dabei ist, dass ausreichend Punkte aus einer anderen Richtung noch einsehbar sind und nicht verdeckt werden. Ziel dabei ist es, die Beziehung zwischen der Lage des Objektes und der Kamera zu bestimmen. Dabei gilt, dass jeder Schlüsselpunkt auf einem bestimmten Strahl liegen muss, der von einer Pixelposition auf dem Bildwandler der Kamera durch die Blende der Kamera hindurch geht.

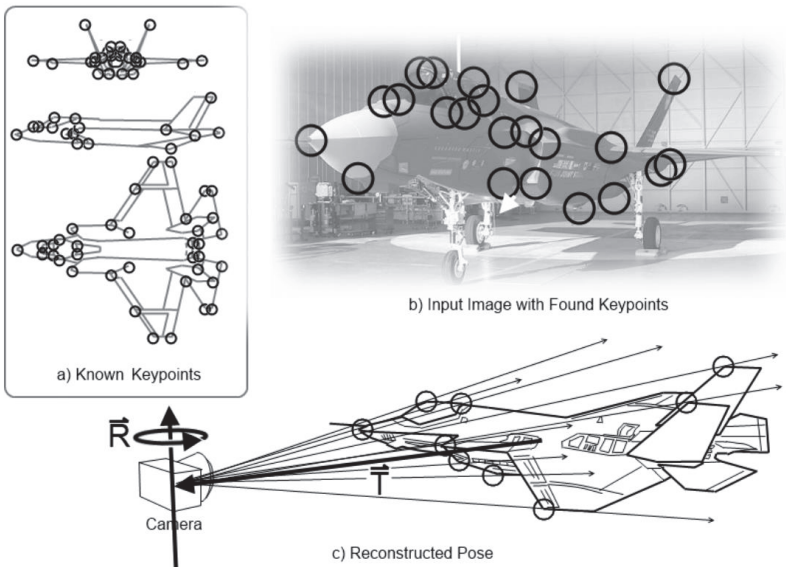

Abbildung 5: Prinzip zur Bestimmung der dreidimensionalen Lage durch eine einzelne Kamera [Kaehler et al., 2017]

Die Abbildung zeigt die schematische Abbildung eines Flugzeuges. Die Kreise zeigen Merkmale des Flugzeuges deren genaue Position (in den Objektkoordinaten des Flugzeugs) koordinatenmäßig bekannt ist. Mit diesen Informationen können Merkmale aus einem Bild extrahiert werden und somit die Pose des Objektes berechnet werden. Da jeder dieser Punkte auf einem Strahl liegen muss, der durch die Kamerablende geht und an einem bestimmten Punkt auf dem Bildwandler landet, hat dieses Problem in der Regel eine eindeutige Lösung. Auch wenn nicht bekannt ist, wo sich der Punkt auf diesem Strahl tatsächlich befindet, gibt es nur eine Möglichkeit, wie das Objekt so platziert werden kann, dass alle diese Strahlenbeschränkungen gleichzeitig erfüllt sind.

Zur Bestimmung der dreidimensionalen Lage des Objektes ist es nicht notwendig, dass alle Schlüsselpunkte aus jeder Richtung einsehbar sind. In der Praxis erkennen Schlüsselpunkt-Detektoren ein Merkmal nur aus einem kleinen Fenster von Winkeln, welche mit Hilfe von Deskriptoren beschrieben werden. Daher ist es oft hilfreich, mehrere Deskriptoren zu verwenden, um die Art und Weise zu erfassen, wie dieses Merkmal von verschiedenen Ort aus wahrgenommen wird. [Kaehler etal., 2017]

Deskriptoren sind im Allgemeinen 1D- oder 2D-Vektoren von Binär-, Ganzzahl- oder Gleitkommazahlen, die einen Schlüsselpunkt und seine Umgebung beschreiben. Ein guter Deskriptor sollte so unterscheidbar zu anderen Deskriptoren sein, dass er ein Merkmal eindeutig darstellt. [Laganiere, 2017]

Es gibt zwei Fälle, in denen dieses Verfahren keine zuverlässigen Ergebnisse liefern kann. Der erste Fall ist, wenn nicht genügend Punkte vorhanden sind. Theoretisch kann das Problem mit nur drei Punkten gelöst werden. In der Praxis kann die Lage bei so wenigen Punkten und dem natürlichen Rauschen, das sich aus der Methode ergibt, mit der sie gefunden wurden, erheblich abweichen. Der zweite Fall liegt vor, wenn das Objekt sehr weit entfernt ist. In dieser Situation werden die Strahlen, die die Position der Merkmale einschränken, parallel. Es ist die Divergenz der Strahlen, die einen eindeutigen Maßstab für das Objekt oder äquivalent eine einzigartige Lösung für die Entfernung zum Objekt liefert.

Diese monokulare Methode zur Schätzung der Lage und des Abstands eines Objekts ist in der Funktionsweise der menschlichen Augen beim Blick auf entfernte Objekte sehr ähnlich. Deshalb ist es unmöglich, die Entfernung zu einem Objekt abzuschätzen, dessen tatsächliche Größe (und damit die Lage seiner Merkmale) nicht ohne anderen Kontext bekannt ist. Dies ist auch die Grundlage für erzwungene perspektivische Illusion, z.B. das einige Gebäude in höheren Stockwerken immer kleinere Fenster haben, um das Gebäude vom Boden aus höher erscheinen zu lassen. [Kaehler et al., 2017]

3.2 Stereo Vision

Eine weitere Option 3D-Informationen und somit die räumliche Lage eines Objektes zu bestimmen ist Stereo Vision. Die Möglichkeit der Stereo-Bildgebung ist bekannt durch das Sehen mit zwei Augen. Diese Fähigkeit kann mit Hilfe von rechnergestützten Systemen nachgebildet werden. Computer erfüllen diese Aufgabe, indem sie Korrespondenzen zwischen Schlüsselpunkten finden, die von beiden Bildgebern gesehen werden. Mit solchen Korrespondenzen und einem bekannten Grundlinienabstand zwischen den Kameras kann die dreidimensionale Lage der Punkte berechnet werden. In der Praxis umfasst die Stereo-Bildgebung bei der Verwendung von zwei Kameras vier Schritte. Schritt eins ist die radiale und tangentiale Linsenverzeichnung mathematisch zu entfernen (engl. Undistortion). Das Ergebnis dieses Schritts sind unverzerrte Bilder. Schritt zwei beinhaltet das Einstellen der Winkel und Abstände zwischen den Kameras, auch Entzerrung genannt. Das Ergebnis dieses Schritts sind entzerrte und reihenweise ausgerichtete Bilder. Letzteres bedeutet, dass die beiden Bildebenen koplanar sind und die entsprechenden Bildreihen auf den beiden Bildwandlern tatsächlich kolinear zueinander sind. Der dritte Schritt umfasst das Detektieren derselben Merkmale in der linken und rechten Kameraansicht. Dieser Vorgang wird auch als Korrespondenz bezeichnet. Das Ergebnis dieses Schritts ist eine Disparitätskarte, wobei die Disparitäten die Unterschiede in den x-Koordinaten auf den Bildebenen desselben Schlüsselpunktes sind, welcher in der linken und rechten Kamera betrachtet wird. Im vierten und letzten Schritt wird die Disparitätskarte durch Triangulation in Entfernungen umgewandelt. Dieser Schritt wird als Reprojektion bezeichnet und beinhaltet eine Tiefenkarte als Ergebnis. [Kaehler et al., 2017]

3.2.1 Triangulation

Angenommen wird zunächst ein ‚ideales' Stereogerät wie in Abbildung 6 dargestellt.

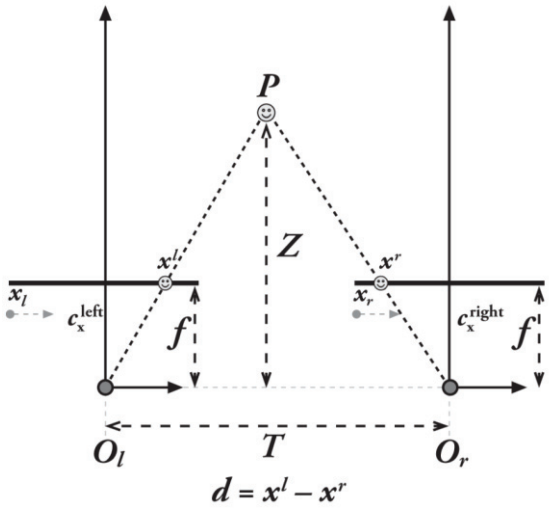

Abbildung 6: ideales Stereogerät [Kaehler et al., 2017]

Bei einem vollkommen unverzerrten, ausgerichteten Stereogerät und bekannter Übereinstimmung kann die Tiefe Z durch ähnliche Dreiecke gefunden werden. Die Hauptstrahlen der Bildwandler beginnen in den Projektionszentren O_l und O_r und erstrecken sich durch die Hauptpunkte der beiden Bildebenen bei c_l und c_r.

In diesem Fall wird von einem vollkommen unverzerrten, ausgerichteten und gemessenen System ausgegangen. Zwei Kameras, deren Bildebenen exakt koplanar zueinander sind, mit exakt parallelen optischen Achsen, die einen bekannten Abstand voneinander haben und gleiche Brennweiten besitzen. Des weiteren wird angenommen, dass die Hauptunkte c_x^{left} und c_x^{right} so kalibriert wurden, dass sie in ihren jeweiligen linken und rechten Bildern die gleichen Pixelkoordinaten haben.

Außerdem gilt für den Moment, dass die Bildaufnehmer perfekt reihenweise ausgerichtet sind, so dass jede Pixelreihe der einen Kamera exakt mit der entsprechenden Reihe in der anderen Kamera ausgerichtet ist. Diese Kameraanordnung wird frontal parallel genannt. Punkt P wird dabei in der rechten und linken Bildansicht gefunden und mit den entsprechenden horizontalen Koordinaten versehen (x_l und x_r).

In diesem vereinfachten Fall wird deutlich, dass die Tiefe umgekehrt proportional zur Disparität zwischen diesen Ansichten ist, wobei die Disparität einfach durch $d = x_l - x_r$ definiert wird. Diese Situation ist in der oberen Abbildung dargestellt, wo die Tiefe Z durch Verwendung ähnlicher Dreiecke leicht abgeleitet werden kann (Formel 1).

$$\frac{T - (x_l - x_r)}{Z - f} = \frac{T}{Z} \Rightarrow Z = \frac{f \cdot T}{x_l - x_r}$$

Formel 1 [Kaehler et al., 2017]

Da die Tiefe umgekehrt proportional zur Disparität ist, gibt es offensichtlich eine nichtlineare Beziehung zwischen diesen beiden Begriffen. Wenn die Disparität nahe Null ist, führen kleine Disparitätsunterschiede zu großen Tiefenunterschieden. Wenn die Disparität groß ist, verändern kleine Disparitätsunterschiede die Tiefe nicht wesentlich. Die Folge ist, dass Stereo Vision-Systeme eine hohe Tiefenauflösung nur für Objekte in relativer Nähe zur Kamera haben, wie in Abbildung 7 deutlich wird:

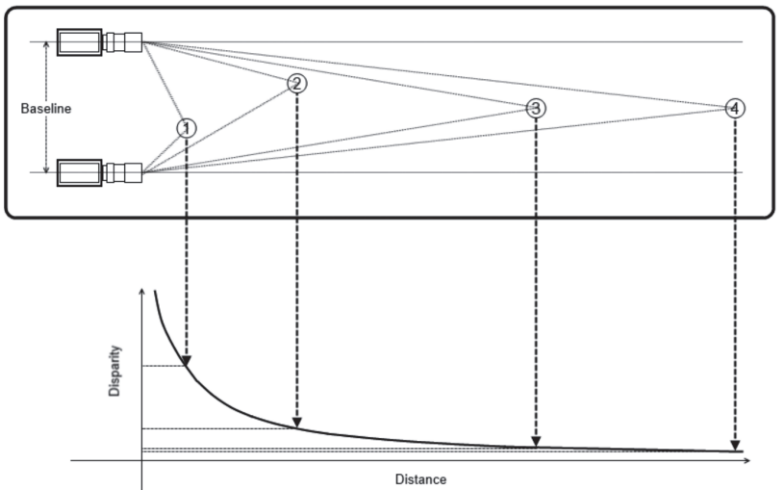

Abbildung 7: Darstellung der Beziehung von Tiefe und Disparität [Kaehler et al., 2017]

Abbildung 8 zeigt die zwei- und dreidimensionalen Koordinatensysteme, die in OpenCV für das Stereosehen verwendet werden. Zu beachten gilt, dass es sich um ein rechtshändiges Koordinatensystem handelt. Die rechten und linken Bildpixel haben ihren Bildursprung oben links im Bild und die Pixel werden durch die Koordinaten (x_l, y_l) bzw. (x_r, y_r) bezeichnet. Die Projektionszentren werden durch O_l und O_r beschrieben. Die Hauptstrahlen schneiden dabei die Bildebenen im Hauptpunkt (nicht im Zentrum) c_x und c_y. Nach der mathematischen Entzerrung werden die Kameras reihenweise ausgerichtet (koplanar und horizontal), um T gegeneinander verschoben und mit der gleichen Brennweite f versehen.

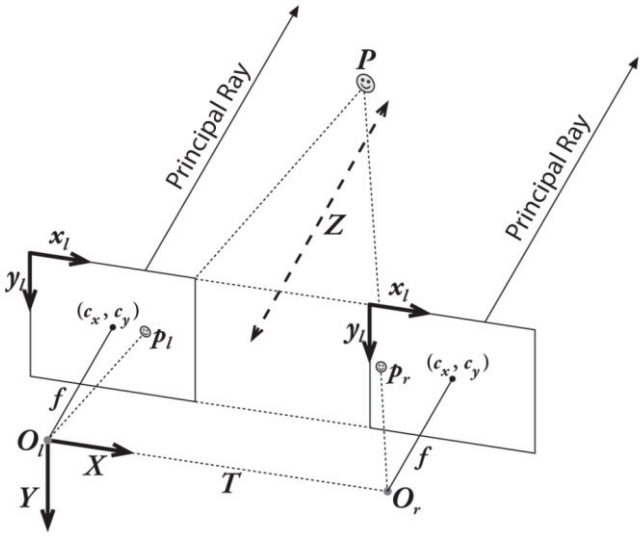

Abbildung 8: Darstellung des Stereokoordinatensystems [Kaehler et al., 2017]

Mit diesem vereinfachten Modell ist es nun möglich zu verstehen, wie eine reale Kameraanordnung in einer Geometrie abgebildet werden kann, die dieser idealen Anordnung ähnelt. In der realen Welt werden die Kameras nicht exakt in der in Abbildung 6 dargestellten frontalen Parallelkonfiguration ausgerichtet sein. Stattdessen werden die linken und rechten Bilder einer Kamera in eine frontale parallele Anordnung entzerrt. Beim Entwerfen eines Stereogeräts ist es empfehlenswert die Kameras annähernd frontal parallel und so nahe wie

möglich an der horizontalen Ausrichtung anzuordnen. Eine solche frontal parallele physische Ausrichtung macht die mathematischen Transformationen leicht durchführbar. Wenn die Kameras nicht zumindest annähernd ausgerichtet sind, kann die daraus resultierende mathematische Ausrichtung zu extremen Bildverzerrungen führen und so den Stereoüberlappungsbereich der resultierenden Bilder reduzieren oder eliminieren. Für gute Ergebnisse ist auch eine Synchronisation beider Kameras notwendig.

Abbildung 9 zeigt die reale Situation zwischen zwei Kameras und die mathematische Ausrichtung, die erreicht werden soll. [Kaehler et al., 2017]

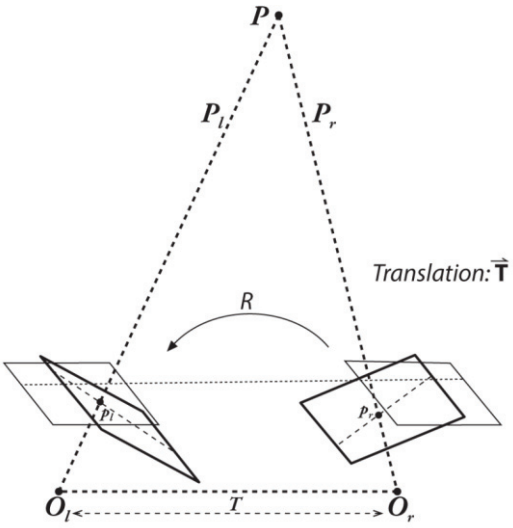

Abbildung 9: Ausrichtung zweier Kameras unter realen Bedingungen [Kaehler et al., 2017]

3.2.2 Epipolare Geometrie

Die Grundgeometrie eines Stereoabbildungssystems wird als epipolare Geometrie bezeichnet. Im Wesentlichen kombiniert diese Geometrie zwei Lochblendenmodelle (eines für jede Kamera) und einige weitere Punkte, die als Epipole bezeichnet werden (Abbildung 10).

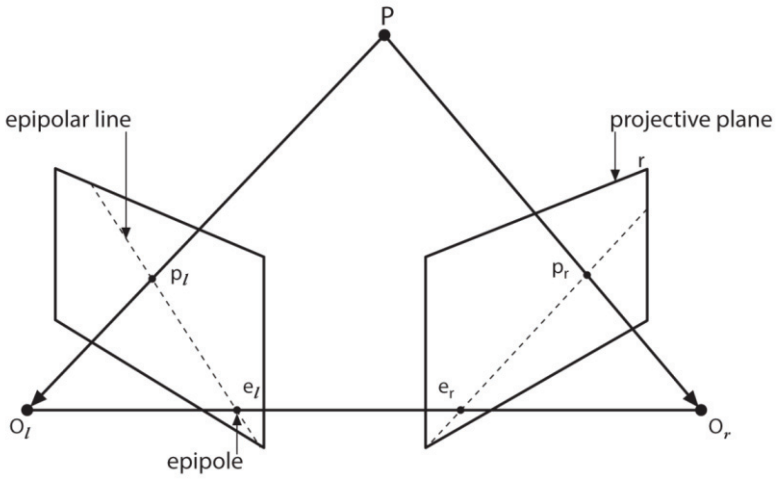

Abbildung 10: Epipolare Geometrie des Stereoabbildungssystems [Kaehler et al., 2017]

Für jede Kamera gibt es ein separates Projektionszentrum, O_l und O_r und ein Paar entsprechender Projektionsebenen, l und r. Der Betrachtungspunkt P wird in der physischen Welt Punkt R genannt. Der Punkt R besitzt eine Projektion auf jede der Projektionsebenen l und r. Diese werden mit p_l und p_r gekennzeichnet. Die Punkte von Interesse sind die Epipole. Ein Epipol e_l oder e_r auf der Projektionsebene l oder r ist definiert als das Bild des Projektionszentrums der anderen Kamera O_l oder O_r auf dieser Ebene. Die Ebene im Raum, die durch den tatsächlichen Betrachtungspunkt P und die beiden Epipole e_l und e_r (oder äquivalent durch die beiden Projektionszentren O_l und O_r) gebildet wird, wird als Epipolarebene bezeichnet und die Linien p_le_l und p_re_r (von den Projektionspunkten zu den entsprechenden Epipolen) werden als Epipolarlinien bezeichnet.

Ein Punkt in der physischen Welt, projiziert auf der linken oder rechten Bildebene, liegt auf einer Linie, gebildet durch den Strahl, welcher durch die Punkte von O_r bis p_r (oder O_l bis p_l) verläuft. Durch eine einzelne Kamera ist die Entfernung zu einem betrachteten Punkt nicht bekannt. Um dies zu verdeutlichen wird Punkt P betrachtet, wie er von der rechten Kamera gesehen wird. Da diese Kamera nur p_r (die Projektion von P auf r) sieht, muss der tatsächliche Punkt P überall auf der Linie liegen, die durch p_r und O_r definiert ist. Diese Linie enthält natürlich P, aber sie enthält auch eine Menge anderer Punkte. Interessant ist jedoch die Frage, wie diese Linie aussieht, wenn sie auf die linke Bildebene l projiziert wird. Das ist die Epipolarlinie, welche durch p_l und e_l definiert ist. Das Bild aller möglichen Orte eines Punktes, die in einem Bildwandler gesehen werden, ist die Linie, die durch den entsprechenden Punkt und den Epipol auf dem anderen Bildwandler verläuft. Für die Berechnung dieser Epipolarlinien sind noch zwei weitere Bestandteile nötig. [Kaehler et al., 2017]

3.2.3 Essenzielle und fundamentale Matrizen

Diese Bestandteile sind die essenzielle Matrix E und die fundamentale Matrix F. Die Matrix E enthält Informationen über die Translation und Rotation, die die beiden Kameras im physischen Raum in Beziehung setzen (Abbildung 11) und F enthält zusätzlich zu den Informationen über die intrinsischen Parameter beider Kameras die gleichen Informationen wie E.

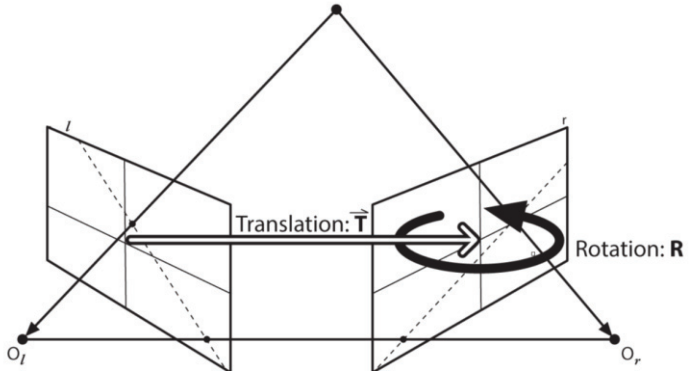

Abbildung 11: Darstellung der Rotation R und Translation T der Stereobildgebung [Kaehler et al., 2017]

Die essenzielle Matrix ist rein geometrisch und besitzt keine Informationen über den Bildgeber. Sie setzt die Lage des Punktes P, wie er von der linken Kamera gesehen wird, in physischen Koordinaten mit der Lage desselben Punktes, wie er von der rechten Kamera gesehen wird, in Beziehung (d. h. sie setzt p_l und p_r in Beziehung). Die Fundamentalmatrix F setzt die Punkte beider Bildebenen in Bildkoordinaten (Pixel) in Beziehung zueinander. [Kaehler et al., 2017]

3.2.4 Stereo-Kalibrierung

Mit der vorangegangenen Theorie ist es nun möglich eine Stereo-Kalibrierung und Stereo-Entzerrung durchzuführen. Die Stereo-Kalibrierung ist der Prozess der Berechnung der geometrischen Beziehung zwischen den beiden Kameras im Raum. Im Gegensatz dazu ist die Stereo-Entzerrung der Prozess der Korrektur der einzelnen Bilder, so dass sie so erscheinen, als wären sie von zwei Kameras mit in Reihe angeordneten Bildebenen aufgenommen worden. Bei einer solchen Entzerrung sind die optischen Achsen (oder Hauptstrahlen) der beiden Kameras parallel (schneiden sich im Unendlichen).

Der nächste Schritt ist es R und T zwischen der linken und rechten Kamera zu berechnen. Begonnen wird damit, dass für jeden beliebigen dreidimensionalen Punkt P in Objektkoordinaten separat die Ein-Kamera-Kalibrierung für die beiden Kameras verwendet wird, um P in die Kamerakoordinaten für beide Kameras zu setzen: $P_l = R_l \cdot P + T_l$ und $P_r = R_r \cdot P + T_r$. In Abbildung 11 ist ersichtlich, dass die beiden Ansichten von P (von den beiden Kameras) durch $P_l = R^T \cdot (P_r - T)$ miteinander verbunden sind, wobei R und T jeweils die Rotationsmatrix und der Translationsvektor zwischen den Kameras sind. Nimmt man diese drei Gleichungen und löst sie getrennt für die Rotation und Translation, erhält man die Beziehungen welche in Formel 2 und Formel 3 dargestellt werden.

$$R = R_r \cdot R_l^T$$

Formel 2 [Kaehler et al., 2017]

$$T = T_r - R \cdot T_l$$

Formel 3 [Kaehler et al., 2017]

Angesichts vieler gemeinsamer Ansichten von Schachbrettecken oder eines ähnlichen Kalibrierobjekts verwendet OpenCV diese Muster, um die Rotations- und Translationsparameter der Ansichten für beide Kameras separat zu lösen. Die linken und rechten Rotations- und Translationslösungen werden in die vorhergehenden Gleichungen eingesetzt, um die Rotations- und Translationsparameter zwischen den beiden Kameras zu lösen. Aufgrund von Bildrauschen und Rundungsfehlern ergeben sich für jedes Schachbrettpaar leicht unterschiedliche Werte für R und T. Als Annäherung an die wahre Lösung wird der Medianwert verwendet. Zusätzlich wird ein robuster Levenberg-Marquardt-Iterationsalgorithmus ausgeführt, um das (lokale) Minimum des Reproduktionsfehlers der Kalibrierpunkte für beide Kameraansichten zu finden und die endgültige Lösung für R und T auszugeben. Die Rotationsmatrix setzt die rechte Kamera in die gleiche Ebene wie die linke Kamera. Dadurch werden die beiden Bildebenen parallel, aber nicht reihenweise ausgerichtet.

Sobald entweder die Rotations- und Translationswerte oder die Fundamentalmatrix F bekannt sind, können diese verwendet werden, um die beiden Stereobilder zu entzerren, sodass die Epipolarlinien entlang der Bildreihen angeordnet sind und die Scanlinien auf beiden Bilder gleich sind. [Kaehler et al., 2017]

3.2.5 Stereo-Entzerrung

Die Stereodisparität lässt sich am einfachsten berechnen, wenn die beiden Bildebenen genau aufeinander ausgerichtet sind (Abbildung 6). Leider ist, wie bereits erwähnt, eine perfekt ausgerichtete Konfiguration, bei einem Stereosystem nur selten der Fall, da die beiden Kameras fast nie exakt koplanare, reihenweise ausgerichtete Bildebenen haben. Ziel ist es, die Bildebenen beider Kameras so zu projizieren, dass sie in genau derselben Ebene liegen, wobei die Bildreihen perfekt in einer frontalen Parallelkonfiguration ausgerichtet sind. Wie die Ebene gewählt wird, in der die Kameras mathematisch ausgerichtet werden, hängt von dem verwendeten Algorithmus ab.

Letztendlich sollen die Bildreihen zwischen den beiden Kameras nach der Entzerrung so ausgerichtet werden, dass die Stereo-Korrespondenz (das Auffinden desselben Punktes in den beiden verschiedenen Kameraansichten) zuverlässiger und berechenbarer wird. Insbesondere werden sowohl die Zuverlässigkeit als auch die Berechnungseffizienz dadurch verbessert, dass nur in einer Zeile nach einer Übereinstimmung mit einem Punkt im anderen Bild gesucht werden muss. Das Ergebnis der Ausrichtung horizontaler Reihen innerhalb einer gemeinsamen Bildebene, die jedes Bild enthält, ist, dass die Epipole selbst dann im Unendlichen liegen. Somit ist das Bild des Projektionszentrums in einem Bild parallel zur anderen Bildebene. Da es jedoch eine unendliche Anzahl möglicher frontaler paralleler Ebenen gibt, aus denen gewählt werden kann, müssen noch weitere Einschränkungen

hinzugefügt werden. Solche Einschränkungen basieren auf der Maximierung der Ansichtsüberlappung und/oder Minimierung der Verzerrung.

Das Ergebnis des Prozesses der Ausrichtung der beiden Bildebenen sind acht Terme, jeweils vier für die linke und die rechte Kamera. Für jede Kamera liegt ein Verzerrungsvektor, eine Rotationsmatrix und die entzerrten und unentzerrten Kameramatrizen vor.

Es gibt viele Möglichkeiten die Entzerrung zu berechnen. Eine Möglichkeit ist z.B. der Hartley-Algorithmus, der unkalibriertes Stereo allein mit Hilfe der Fundamentalmatrix liefern kann. Eine andere Möglichkeit ist der Bouget-Algorithmus, der die Rotations- und Translationsparameter von zwei kalibrierten Kameras verwendet. Der Hartley-Algorithmus kann verwendet werden, um aus der von einer einzigen Kamera aufgezeichneten Bewegung eine Struktur abzuleiten, kann aber (bei Stereokorrektur) stärker verzerrte Bilder erzeugen als der Algorithmus von Bouget.

Der Prozess der Entzerrung ist in Abbildung 12 dargestellt. Wie aus dem Ablauf in dieser Abbildung ersichtlich ist, verläuft der eigentliche Entzerrungsprozess rückwärts von (c) nach (a) in einem Prozess, der als Reverse Mapping bekannt ist. Für jedes ganzzahlige Pixel im entzerrten Bild (c) wird die zugehörige Koordinate im unverzerrten Bild (b) gefunden. Diese werden verwendet, um die tatsächlichen Fließkomma-Koordinaten im Rohbild (a) nachzuschlagen. Der Fließkomma-Koordinaten-Pixelwert wird dann aus den nahegelegenen ganzzahligen Pixelpositionen im ursprünglichen Quellbild interpoliert und dieser Wert wird zum Auffüllen der entzerrten ganzzahligen Pixelposition im Zielbild (c) verwendet. Nachdem das entzerrte Bild ausgefüllt wurde, wird es normalerweise beschnitten, um die überlappenden Bereiche zwischen dem linken und rechten Bild hervorzuheben. [Kaehler et al., 2017]

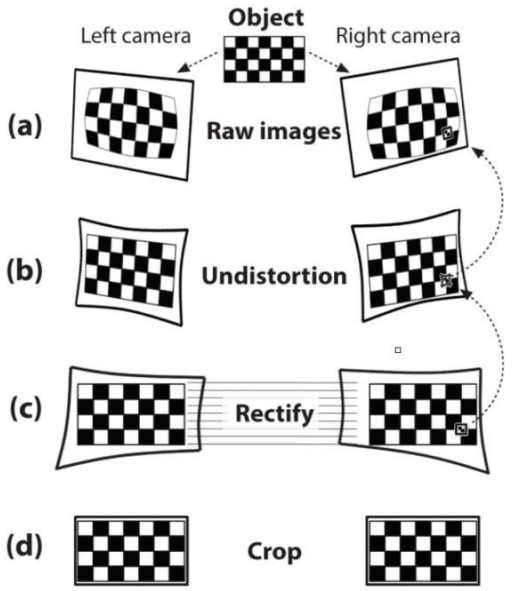

Abbildung 12: Entzerrungsprozess [Kaehler et al., 2017]

Abbildung 13 zeigt die Ergebnisse der Stereo-Unverzerrung und -Entzerrung eines Stereo-Bildpaares.

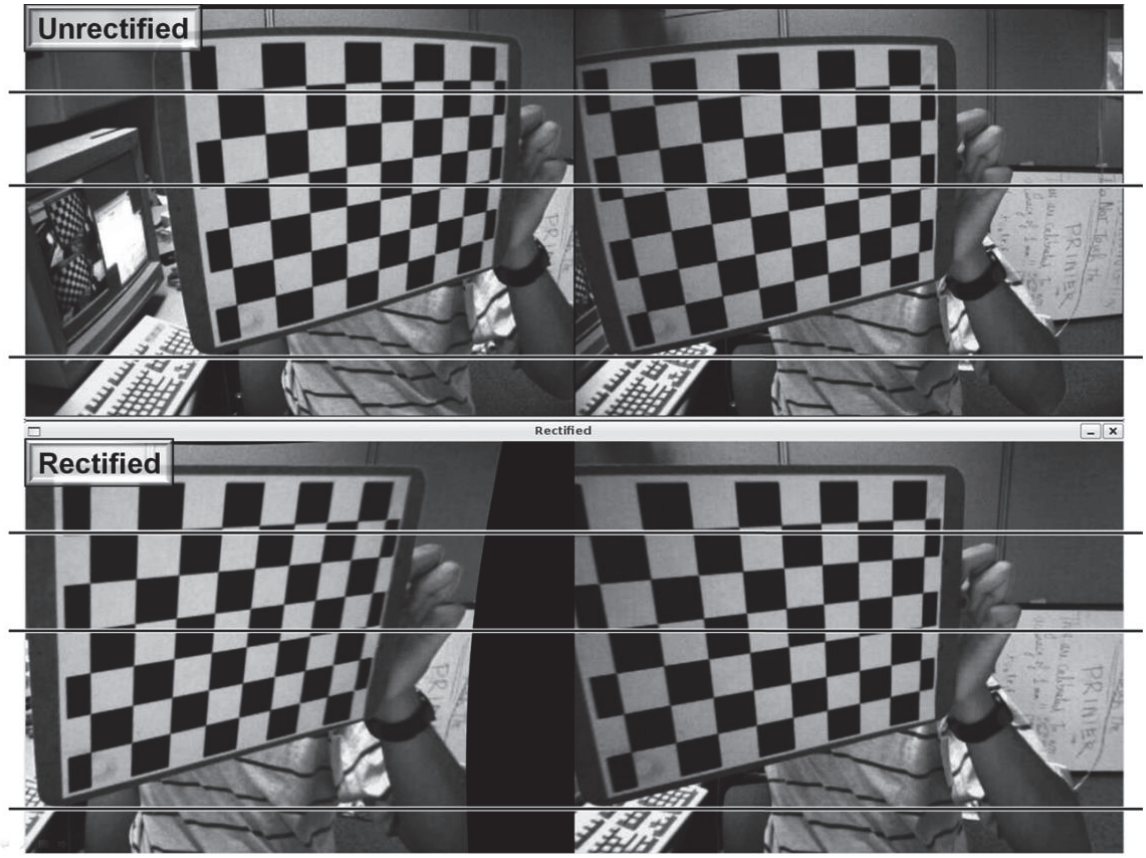

Abbildung 13: Ergebnisse der Stereo-Unverzerrung und -Entzerrung eines Stereo-Bildpaares [Kaehler et. al., 2017]

3.2.6 Stereo-Korrespondenz

Stereo-Korrespondenz ist die Übereinstimmung eines dreidimensionalen Punktes in beiden Kameraansichten. Sie kann nur über die visuellen Bereiche berechnet werden, in denen sich die Ansichten der beiden Kameras überlappen. Auch dies ist ein Grund dafür, dass tendenziell bessere Ergebnisse erzielt werden, wenn die Kameras möglichst nahezu frontal parallel angeordnet sind. Wenn die physischen Koordinaten der Kameras oder die Größe der Objekte in der Szene bekannt sind, können Tiefenmessungen aus den triangulierten Disparitätsmaßen $d = x^l - x^r$ (oder $d = x^l - x^r - (c_x^{left} - c_x^{right})$, wenn sich die Hauptstrahlen in einem endlichen Abstand schneiden) zwischen den entsprechenden Punkten in den beiden verschiedenen Kameraansichten abgeleitet werden. Ohne solche physikalischen Informationen kann die Tiefe nur bis zu einem Skalierungsfaktor berechnet werden.

OpenCV implementiert zwei verschiedene Stereo-Korrespondenz-Algorithmen, die sich eine (fast) gemeinsame Objektschnittstelle teilen. Der erste, der als Blockvergleichsalgorithmus (BM-Algorithmus) bezeichnet wird, ist ein schneller und effektiver Algorithmus, der dem von

Kurt Konolige entwickelten Algorithmus ähnlich ist. Er arbeitet mit kleinen ‚Summen der absoluten Differenz'-Fenstern (SAD-Fenster), um übereinstimmende Punkte zwischen dem linken und rechten stereogerechten Bild zu finden. Dieser Algorithmus findet nur stark übereinstimmende (hochtexturierte) Punkte zwischen den beiden Bildern. In einer stark texturierten Szene, wie sie z. B. im Freien in einem Wald vorkommen kann, könnte also jedes Pixel eine berechnete Tiefe haben. In einer Szene mit sehr geringer Textur, wie z. B. in einem Innenflur, können nur sehr wenige Punkte Tiefe registrieren. Der zweite Algorithmus wird als semi-global block matching (SGBM)-Algorithmus bezeichnet. SGBM unterscheidet sich von BM vor allem in zwei Punkten. Der erste ist, dass das Matching auf Subpixelebene unter Verwendung der Birchfield-Tomasi-Metrik erfolgt. Der zweite Unterschied besteht darin, dass SGBM versucht, eine globale Glättungsbeschränkung für die berechnete Tiefeninformation durchzusetzen, die der Algorithmus approximiert, indem er viele eindimensionale Glättungsbeschränkungen durch die Region von Interesse berücksichtigt. Diese beiden Methoden ergänzen sich in dem Sinne, dass BM zwar um einiges schneller ist, aber nicht die Zuverlässigkeit und Genauigkeit des SGBM bietet.

Der in OpenCV implementierte Block-Matching-Stereo-Algorithmus ist eine leicht modifizierte Version dessen, was mittlerweile als kanonische Technik für die Stereo-Berechnung angesehen wird. Der grundlegende Mechanismus besteht darin, die Bilder so zu entzerren und auszurichten, dass Vergleiche nur in einzelnen Zeilen durchgeführt werden müssen, um dann den Algorithmus in den Zeilen der beiden Bilder nach übereinstimmenden Pixelgruppen suchen zu lassen. Natürlich gibt es einige zusätzliche Details, die den Algorithmus auf verschiedene Weise entweder etwas präziser arbeiten oder etwas schneller laufen lassen. Das Ergebnis ist ein schneller, zuverlässiger Algorithmus, der noch immer in einer Vielzahl von Anwendungen stark genutzt wird.

Der Block-Vergleichs-Stereo-Korrespondenz-Algorithmus, der mit unverzerrten, entzerrten Stereo-Bildpaaren arbeitet, besteht aus drei Stufen:

1. Vorfilterung zur Normalisierung der Bildhelligkeit und Verbesserung der Textur.
2. Korrespondenzsuche entlang horizontaler Epipolarlinien unter Verwendung eines SAD-Fensters.
3. Nachfilterung, um schlechte Korrespondenzübereinstimmungen zu eliminieren.

Im Vorfilterungsschritt werden die Eingabebilder normalisiert, um Beleuchtungsunterschiede zu reduzieren und die Bildtextur zu verbessern. Dies geschieht mit einem Fenster der Größe 5x5, 7x7 (Standard) bis maximal 21x21. Dieses Fenster wird über das Bild laufen gelassen. Das mittlere Pixel I_c unter dem Fenster wird durch $\min(\max(I_c - I, -I_{cap}), I_{cap})$ ersetzt, wobei I der Mittelwert des Fensters und I_{cap} ein positiver numerischer Grenzwert ist, dessen Standardwert 30 beträgt. Die Alternative besteht im Wesentlichen darin, die eingehenden

Bilder in ihre x-Sobel-Ableitung umzuwandeln. Dies hat auch den Effekt, dass viele belichtungsbedingte Artefakte entfernt werden.

Als nächstes wird die Korrespondenz durch Verschiebung des SAD-Fensters berechnet. Für jedes Merkmal im linken Bild wird in der entsprechenden Zeile im rechten Bild nach der besten Übereinstimmung gesucht. Nach der Entzerrung ist jede Zeile eine epipolare Linie, so dass die übereinstimmende Stelle im rechten Bild entlang derselben Zeile (dieselbe y-Koordinate) wie im linken Bild liegen muss. Diese übereinstimmende Stelle kann gefunden werden, wenn das Merkmal genügend Textur aufweist, um erkennbar zu sein und wenn es in der Ansicht der rechten Kamera nicht verdeckt ist (Abbildung 17). Befindet sich die Pixelkoordinate des linken Merkmals bei (x_0, y_0), dann muss bei einer horizontalen frontalen parallelen Kameraanordnung die Übereinstimmung (falls vorhanden) in derselben Reihe und bei oder links von x_0 gefunden werden (Abbildung 14). Bei frontalen parallelen Kameras liegt x_0 bei null Disparität und größere Disparitäten befinden sich links davon. Bei Kameras, die im Winkel zueinander stehen, kann die Übereinstimmung bei negativen Disparitäten (rechts von x_0) auftreten. Dem Algorithmus muss die minimale Disparität mitgeteilt werden.

Die Disparitätssuche wird dann über eine vorgewählte Anzahl von Disparitäten durchgeführt, die in Pixeln gezählt werden (die Voreinstellung ist 64 Pixel). Die Disparitäten haben eine diskrete Sub-Pixel-Auflösung, die einer 4 Bit-Auflösung unterhalb der Ebene der einzelnen Pixel entspricht. Wenn es sich bei dem Ausgabebild um ein 32-Bit-Fließkomma-Bild handelt, werden nicht-ganzzahlige Disparitäten zurückgegeben. Wenn das Ausgabebild eine 16-Bit-Ganzzahl ist, wird die Disparität in 4-Bit-Festkommaform zurückgegeben (d.h. mit 16 multipliziert und auf eine Ganzzahl gerundet).

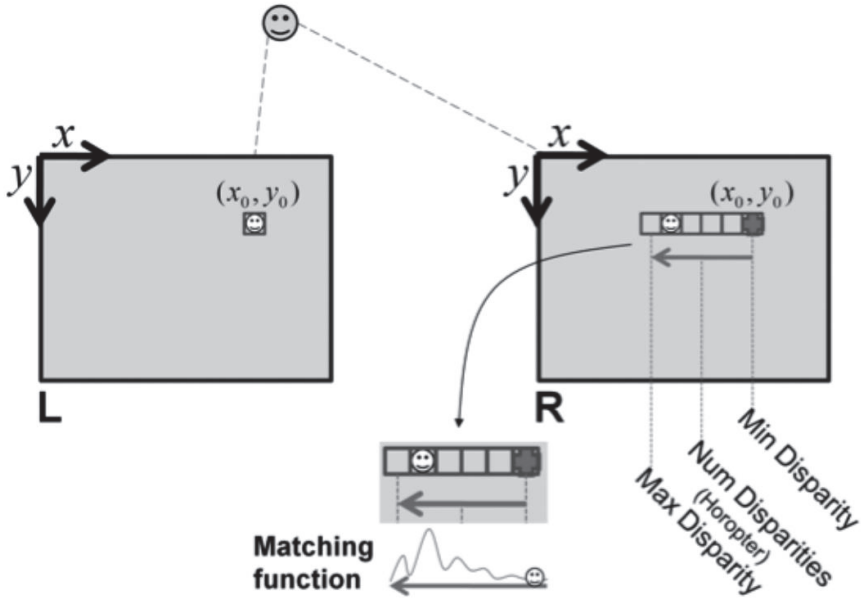

Abbildung 14: Darstellung der Disparitätsgrenzen [Kaehler et al., 2017]

Die Einstellung der Mindestdisparität und der Anzahl der zu durchsuchenden Disparitäten legt den Horopter fest. Der Horopter beschreibt die Gesamtheit aller Punkte im Außenraum, die bei binokularer Fixation eines gegebenen Objektpunktes auf korrespondierenden Netzhautstellen abgebildet werden. (Spektrum, 1999)

In diesem Fall ist es das dreidimensionale Volumen, das vom Suchbereich des Stereo-Algorithmus abgedeckt wird. Abbildung 14 zeigt Disparitäts-Suchgrenzen von fünf Pixeln, beginnend bei drei verschiedenen Disparitätsgrenzen: 20, 17 und 16. Jede Disparitätsgrenze definiert eine Ebene in einer festen Tiefe von den Kameras aus (Abbildung 15). Wie in Abbildung 14 dargestellt, legt jede Disparitätsgrenze zusammen mit der Anzahl der Disparitäten einen anderen Horopter fest, bei dem die Tiefe erkannt werden kann. Außerhalb dieses Bereichs wird die Tiefe nicht gefunden und stellt ein ‚Loch' in der Tiefenkarte dar, bei dem die Tiefe nicht bekannt ist. Der Horopter kann durch Verringerung des Basislinienabstands zwischen den Kameras, durch Verkleinerung der Brennweite, durch Vergrößerung des Suchbereichs der Stereodisparitäten oder durch das Vergrößern der Pixelbreite vergrößert werden.

Die Korrespondenz innerhalb des Horopters hat eine eingebaute Beschränkung, die so genannte Ordnungsbeschränkung, welche besagt, dass sich die Reihenfolge der Merkmale von der linken zur rechten Ansicht nicht ändern kann. Es kann sein, dass Merkmale fehlen. Wobei aufgrund von Okklusion und Rauschen einige Merkmale, die links gefunden werden, rechts nicht gefunden werden können. Die Reihenfolge der gefundenen Merkmale bleibt aber die gleiche. In ähnlicher Weise kann es auf der rechten Seite viele Merkmale geben, die auf der linken Seite nicht identifiziert wurden. Das in Abbildung 16 dargestellte Verfahren spiegelt die Ordnungsbeschränkung wider, wenn Merkmale auf einer horizontalen Abtastlinie abgeglichen werden.

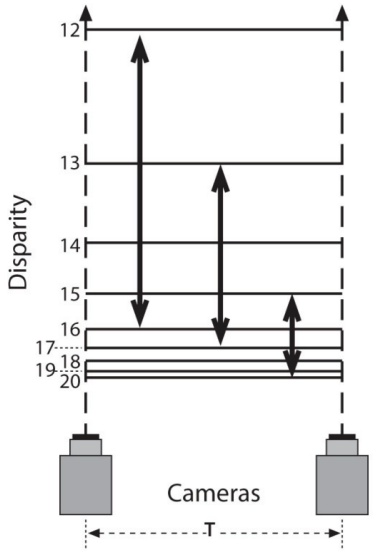

Abbildung 15: Disparitätssuche in verschiedenen Horopterbereichen [Kaehler et al., 2017]

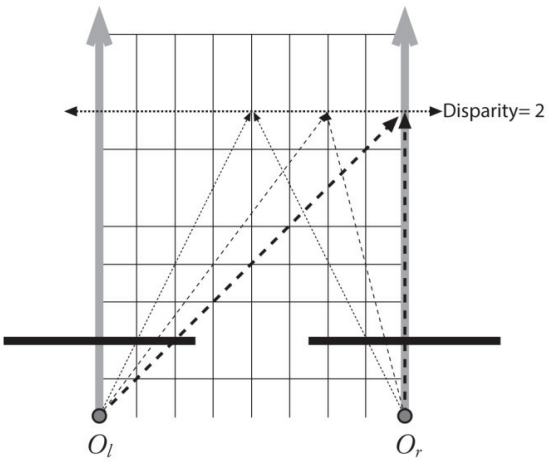

Abbildung 16: Verfahren zum Abgleich von Merkmalen auf einer horizontalen Linie [Kaehler et al., 2017]

Angesichts der kleinsten zulässigen Disparitätsstufe Δ_d ist die kleinste erreichbare Tiefenbereichsauflösung Δ_z mit Formel 4 bestimmbar.

$$\Delta_z = \frac{z^2}{f \cdot T_x} \Delta_d$$

Formel 4 [Kaehler et al., 2017]

Sobald Übereinstimmungen von Kandidatenmerkmalen zwischen den beiden Ansichten gefunden wurden, wird die Nachfilterung verwendet, um falsche Übereinstimmungen zu verhindern. OpenCV macht sich das Muster der Matching-Funktion über das Konzept des Eindeutigkeitsverhältnisses zunutze. Dieses Verhältnis erzwingt im Wesentlichen die Anforderung, dass der Übereinstimmungswert für das aktuelle Pixel größer ist als der minimale Übereinstimmungswert, der um einen gewissen Bereich beobachtet wird.

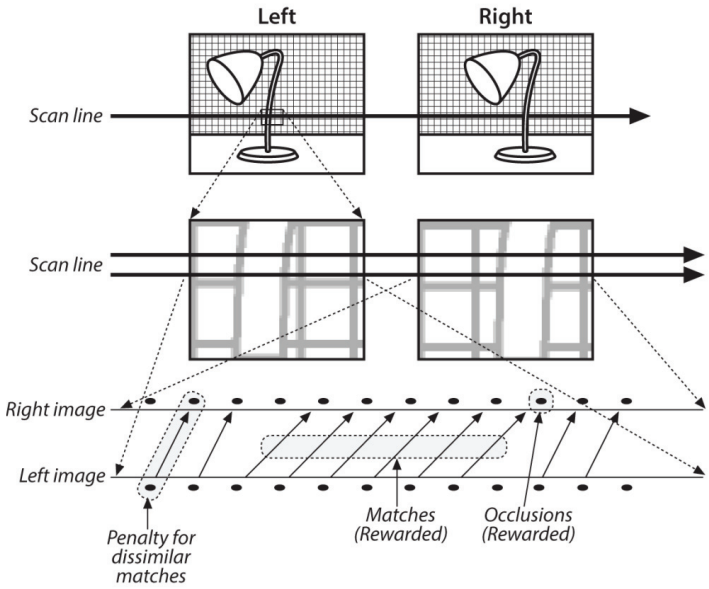

Abbildung 17: Zuweisung von Punktübereinstimmungen [Kaehler et al., 2017]

Viele Algorithmen verwenden die Disparitätskarte z. B., um zu erkennen ob sich Objekte auf einem Tisch befinden oder aus ihm herausragen. Aber für dreidimensionale Formanpassung, dreidimensionales Modelllernen, Robotergreifen usw. wird die eigentliche dreidimensionale Rekonstruktion benötigt. [Kaehler et al., 2017]

Nach Berücksichtigung und Anwendung der vorangegangenen Schritte ist das Ergebnis eine Tiefenkarte. Eine Tiefenkarte ist ein Bild, das Informationen über den Abstand zwischen der Oberfläche von Objekten von einem bestimmten Standpunkt aus enthält. Sie wird aus einem Quellbild erstellt und ist typischerweise im Graustufenformat. Wenn sie mit dem Quellbild zusammengeführt wird, entsteht ein 3D-Bild. Es erscheint, als ob das ursprüngliche Quellbild Tiefe hat. [Parikh, 2019]

Die menschlichen Augen sind horizontal im Durchschnitt um etwa 65 mm voneinander getrennt. Dadurch hat jedes Auge eine etwas andere Sicht auf die Umwelt. Durch den Vergleich dieser beiden Ansichten kann unser Gehirn nicht nur auf die Tiefe, sondern auch auf 3D-Bewegungen im Raum schließen. [Stereolabs, 2020]

OpenCV verfügt über eine eingebaute Funktion stereoSGBM zur Berechnung der Disparität aus einem Paar von Stereobildern, aber der Algorithmus hat eine sehr begrenzte Reichweite, da er eine signifikante Änderung in den beiden entzerrten Frames benötigt, um ein adäquates Ergebnis zu erzielen, das nur funktioniert, wenn sich die Objekte in unmittelbarer Nähe des Stereorigs befinden (Abbildung 18). [Giscle, 2018]

Abbildung 18: Tiefenkarte als Ergebnis unter Verwedung des SGBM Algorithmus

3.3 Time of Flight

Die Time of Flight (ToF)-Kamera ist ein Sensortyp, der dreidimensionale Bilder bei hoher Bildfrequenz und gleichzeitig Intensitätsdaten und Entfernungsinformationen für jedes Pixel liefert. Obwohl die Anzahl der Pixel in den Bildern immer noch klein ist und das Rauschen in den Tiefenwerten nach der Kalibrierung noch nicht vollständig entfernt werden kann, zeigt die

ToF-Bildgebung rasch ein großes Potenzial in zahlreichen wissenschaftlichen Bereichen. Dank des kontinuierlichen Fortschritts in der Mikroelektronik, Mikrooptik und Mikrotechnologie war die Entwicklung von ToF-Kameras in den letzten zehn Jahren möglich. [Foix, 2011]

3.3.1 Prinzip einer ToF-Kamera

Tiefenmessungen basieren auf dem ToF-Prinzip. Die Lichtlaufzeit kann entweder mit gepulster oder kontinuierlicher (CW) Wellen-Modulation gemessen werden. Eine ToF-Kamera arbeitet, indem sie die Szene mit einer modulierten Lichtquelle beleuchtet und das reflektierte Licht beobachtet. Die Phasenverschiebung zwischen der Beleuchtung und der Reflexion wird gemessen und in die Entfernung umgerechnet (Abbildung 19). Typischerweise erfolgt die Beleuchtung durch einen Festkörperlaser oder eine LED, die im nahen Infrarotbereich arbeitet und für das menschliche Auge nicht sichtbar ist. Ein bildgebender Sensor, der auf das gleiche Spektrum reagiert, empfängt das Licht und wandelt die photonische Energie in elektrischen Strom um. Um Phasenverschiebungen zwischen der Beleuchtung und der Reflexion zu erkennen, wird die Lichtquelle durch eine CW-Modulation gepulst oder moduliert. Die gepulste Modulation kann durch Integration von Photoelektronen aus dem reflektierten Licht oder durch Starten eines schnellen Zählers bei der ersten Erfassung der Reflexion erreicht werden. Dieser Zählansatz erfordert eine schnelle Elektronik, da für eine Genauigkeit von 1 Millimeter ein Impuls von 6,6 Pikosekunden Dauer erforderlich ist. [Li, 2014]

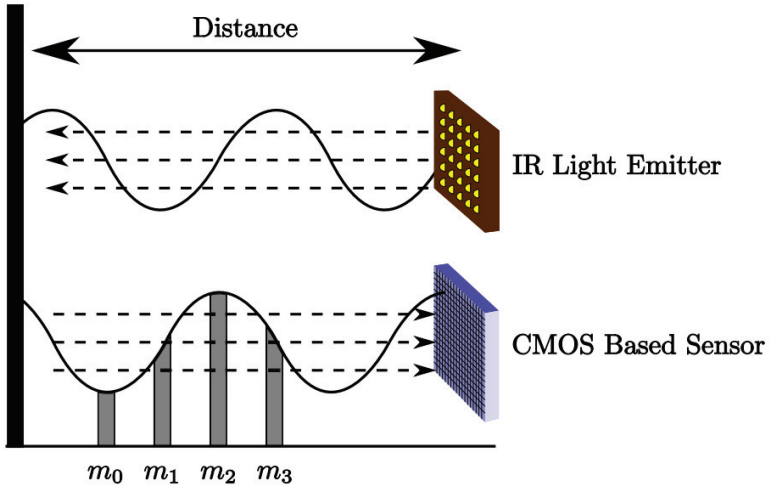

Abbildung 19: Entfernungsmessung durch Phasenoffset [Foix, 2011]

Jedes Pixel auf dem Sensor tastet die von der Szene reflektierte Lichtmenge viermal in gleichen Abständen für jede Periode ab (m0, m1, m2 und m3), was die parallele Messung seiner Phase (Formel 5), seines Offsets (Formel 6) und seiner Amplitude (Formel 7) ermöglicht.

$$\varphi = \arctan\left(\frac{m_3 - m_1}{m_0 - m_2}\right)$$

Formel 5 [Foix, 2011]

$$B = \frac{m_0 + m_1 + m_2 + m_3}{4}$$

Formel 6 [Foix, 2011]

$$A = \frac{\sqrt{[m_3 - m_1]^2 + [m_0 - m_2]^2}}{2}$$

Formel 7 [Foix, 2011]

Diese Phasendemodulationstechnik ist allgemein als „four-bucket" sampling bekannt und ermöglicht eine einfache Berechnung der Zieltiefe (Formel 8).

$$D = L\frac{\varphi}{2\pi}$$

Formel 8 [Foix, 2011]

Die Modulationsfrequenz (fm) des abgestrahlten Lichts bestimmt den mehrdeutigkeitsfreien Abstandsbereich des Sensors (Formel 9), wobei c die Lichtgeschwindigkeit im Vakuum ist. [Foix, 2011]

$$L = \frac{c}{2f_m}$$

Formel 9 [Foix, 2011]

Zur Verlängerung des messbaren Abstands, kann die Modulationsfrequenz reduziert werden, jedoch auf Kosten der reduzierten Genauigkeit. Anstatt diesen Kompromiss zu akzeptieren, setzen moderne ToF-Systeme Mehrfrequenztechniken ein, um den Abstand zu vergrößern, ohne die Modulationsfrequenz zu verringern. Jede Modulationsfrequenz hat einen anderen Mehrdeutigkeitsabstand, aber der wahre Ort ist der, an dem die verschiedenen Frequenzen übereinstimmen. Die Frequenz, bei der die beiden Modulationen übereinstimmen, die als Schwebungsfrequenz bezeichnet wird, ist normalerweise niedriger und entspricht einem viel längeren Mehrdeutigkeitsabstand. Dieses Konzept geht aus Abbildung 20 hervor. [Li, 2014]

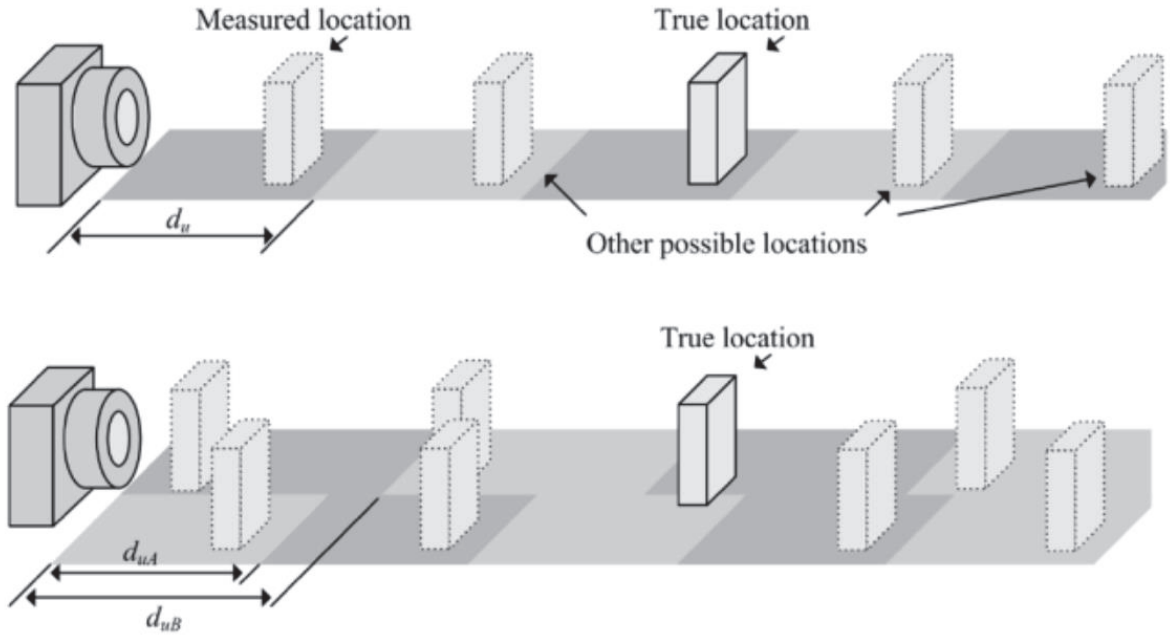

Abbildung 20: Multi-Frequenz-Konzept [Li, 2014]

3.3.2 Ableitung einer Punktwolke

Bei ToF-Sensoren wird der Abstand für jedes Pixel in einem adressierbaren 2D-Array gemessen, was zu einer Tiefenkarte führt. Als Beispiel hat ein QVGA-Sensor eine Tiefenkarte mit einer Auflösung von 320 x 240 Pixeln (Abbildung 21). [Li, 2014]

Abbildung 21: Tiefekarte las Ergebnis durch ToF-Kamera [Li, 2014]

Alternativ kann eine Tiefenkarte in einem dreidimensionalen Raum als eine Punktwolke dargestellt werden (Abbildung 22).

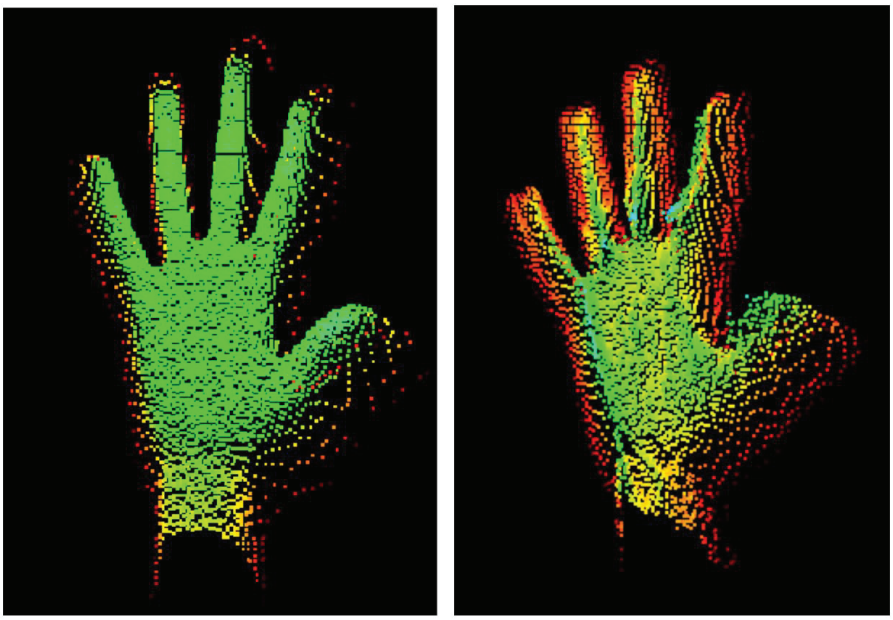

Abbildung 22: Darstellung einer Punktwolke [Foix, 2011]

3.4 Lichtschnittverfahren

Ein anderes Verfahren, welches im Bereich der Qualitätskontrolle zum Einsatz kommt, ist das Lichtschnittverfahren.

3.4.1 Grundprinzip

Ein Messsystem mit Lichtschnitt wird gebildet durch das zu messende Objekt, einen Laser und eine geeignete positionierte Kamera, welche den Schnittpunkt der Lichtebene und des Objekts beobachtet (Abbildung 23). [Fauster, 2016]

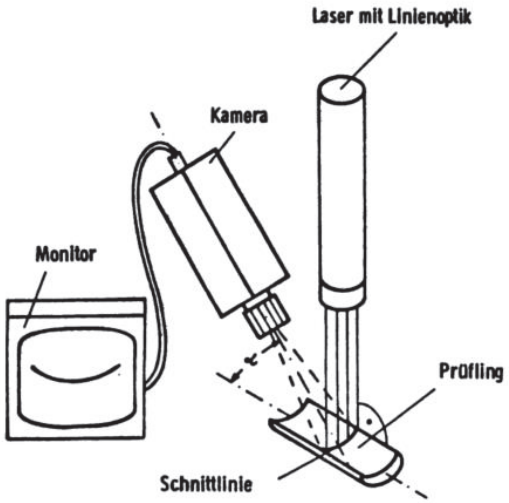

Abbildung 23: Aufbau und Prinzip des Lichtschnittverfahrens [Schreiber, 1989]

Die sich daraus ergebenden sichtbaren Schnittlinien stellen die auszuwertenden Profile dar. Diese Linie wird von der Kamera aufgezeichnet und auf einem Monitor angezeigt. Abhängig vom gewählten Aufnahmewinkel wird die Schnittlinie durch die Kamera, die die Schnittlinie aufzeichnet, deformiert. Wenn die Kameraachse parallel zur Lichtfläche verläuft, wird die aufgezeichnete Schnittlinie zu einer Geraden und liefert keine Informationen über die Kontur.

Mit der in Abbildung 23 dargestellten Anordnung von Lichtquelle und Kamera kann während des Aufnahmevorgangs nur ein Teil des Gesamtumrisses erfasst werden. Daher ist bei der Aufnahme mehrerer Profile zusätzlich zu einer Schwenkachse ein Drehtisch erforderlich, um das Profil zu drehen, so dass Schnitte auf verschiedenen Höhen platziert werden können. (Schreiber, 1989)

3.4.2 Aufbau und Auslegung eines Lichtschnittsensors

Die vom Lichtschnittsensor auf das zu prüfende Objekt projizierte Linie wird in der Regel durch eine gepunktete Laserstrahlquelle erzeugt, der ein optisches System vorgeschaltet ist. Typischerweise werden dazu Gitter- oder Powell-Linsen verwendet, die entlang der Projektionslinie eine konstante Strahlungsintensität erzeugen und damit gute Voraussetzungen für die Detektion von Bildsensoren schaffen. Die Linienlänge hängt vom Fächerwinkel der Laseroptik ab, der zwischen 1° und 120° variieren kann.

Die Breite der Laserlinie spielt eine entscheidende Rolle für die Detailauflösung, die während des Messvorgangs erreicht werden kann. Feinere Strukturen können von Lichtschnittsensoren je nach ihrer Ausrichtung nur in einem begrenzten Bereich erfasst werden. Der Querschnitt des Gauß'schen Strahls in der Laserlichtebene in Messrichtung hat eine Taille. Der Querschnitt wird mit zunehmendem Abstand von der Taille breiter. Je schmaler die Taille, desto größer die Breite. Eine Linienbreite von weniger als 20 Mikrometern ist in einem Messbereich von wenigen Millimetern technisch machbar. Häufiger sind Linienbreiten von 80 bis 300 Mikrometer, die einen Messbereich zwischen 20 und 300 mm abdecken.

Der Bildsensor erfasst das von der Zieloberfläche gestreute Laserlicht und wandelt es in zweidimensionale Bildinformationen um. Da die Ebene des Laserlichts nicht parallel zur Ebene des Bildsensors liegt, kann die begrenzte Schärfentiefe bei der Bildaufnahme den verfügbaren Messbereich einschränken. Wenn sich also Objekt-, Bild- und Objektivebene unter idealisierten Bedingungen in den meisten Fällen auf derselben Schnittlinie treffen, wird die Objektebene in den meisten Fällen scharf auf die Bildebene projiziert. Der Bildsensor wird dazu geneigt und zur optischen Achse des Objektivs montiert.

Die Größe und der Arbeitsabstand des Lichtschnittsensors stehen in engem Zusammenhang mit dem Mindestabstand zwischen der Laserstrahlquelle und dem Bildsensor aufgrund des

erforderlichen Triangulationswinkels. Mit zunehmendem Winkel nimmt die Tiefenauflösung zu und Faktoren wie die Breite der Laserlinie und die Abnahme des Anteils an diffus reflektiertem Licht an der Oberfläche haben eine begrenzte Wirkung. Triangulationswinkel zwischen 10° und 80° sind üblich und der Bereich zwischen 10° und 25° ist ein guter Kompromiss zwischen Messbereich und Messauflösung. Große Messbereiche oder große Arbeitsabstände erfordern daher entsprechend große Sensoraufbauten.

Kommerziell erhältlich sind eine Reihe kompakt konstruierter optischer Segmentsensoren, die typischerweise zwischen 20 und 300 mm messen und in Abständen zwischen 50 und 300 mm arbeiten. Dank der standardisierten Schnittstelle ist der Sensor oft einfach in Betrieb zu nehmen und liefert dank der Werkskalibrierung des Herstellers sofort Messpunkte im festen zweidimensionalen Koordinatensystem des Sensors. Alternativ kann der optische Segmentsensor aus Laserquelle und Kamera passend zur Aufgabe konfiguriert werden. Damit können enge Einbauräume oder spezielle Auflösungsanforderungen berücksichtigt werden, was jedoch ein detailliertes Verständnis der Zusammenhänge der verschiedenen Einflussfaktoren und die Kalibrierung solcher Systeme voraussetzt. [Frauenhofer Vision, 2014]

3.5 Strukturierte Beleuchtung

Eine Hauptmethode der 3D-Oberflächenabbildung basiert auf der Verwendung von strukturiertem Licht, d.h. der aktiven Beleuchtung der Szene mit speziell entworfenen räumlich variierenden 2D-Intensitätsmustern.

3.5.1 Funktionsweise

Wie in Abbildung 24 dargestellt, wird eine räumlich variierende 2D-strukturierte Beleuchtung durch einen speziellen Projektor oder eine durch einen räumlichen Lichtmodulator modulierte Lichtquelle erzeugt.

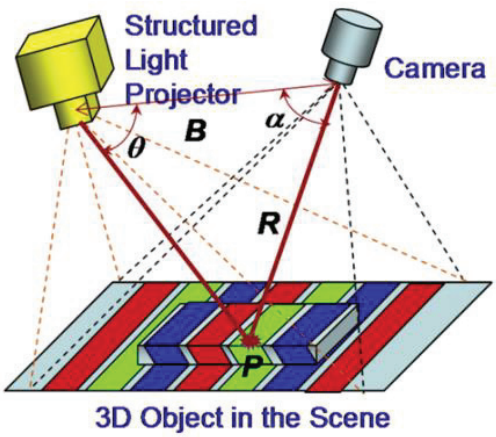

Abbildung 24: Funktionsweise von strukturiertem Licht [Geng, 2011]

Ein bildgebender Sensor (z.B. eine Videokamera) wird verwendet, um ein 2D-Bild der Szene unter der Beleuchtung mit strukturiertem Licht aufzunehmen. Wenn es sich bei der Szene um eine ebene Oberfläche ohne jegliche 3D-Oberflächenvariation handelt, entspricht das im erfassten Bild dargestellte Muster dem projizierten Strukturlichtmuster. Wenn die Oberfläche in der Szene jedoch nicht planar ist, wird das projizierte Strukturlichtmuster durch die geometrische Form der Oberfläche verzerrt dargestellt. Das Prinzip der 3D-Oberflächenabbildungstechniken mit strukturiertem Licht besteht darin, die 3D-Oberflächenform auf der Grundlage der Informationen aus der Verzerrung des projizierten Strukturlichtmusters zu extrahieren. Genaue 3D-Oberflächenprofile von Objekten in der Szene können unter Verwendung verschiedener Strukturlichtprinzipien und -algorithmen berechnet werden.

Die geometrische Beziehung zwischen einem Bildsensor, einem Strukturlichtprojektor und einem Objektoberflächenpunkt kann durch das Triangulationsprinzip ausgedrückt werden (Formel 10).

$$R = B\frac{\sin(\theta)}{\sin(\alpha + \theta)}$$

Formel 10 [Geng, 2011]

Abbildung 24 stellt ein 3D-Bildgebungssystems mit strukturiertem Licht dar. Eine beliebige 3D-Zieloberfläche wird durch ein strukturiertes Lichtprojektionsmuster beleuchtet. Ein Farbabbildungssensor erfasst das Bild der 3D-Zieloberfläche unter der Beleuchtung mit strukturiertem Licht. Basierend auf der Verzerrung des Strukturlichtmusters auf dem erfassten Bild im Vergleich zum unverzerrten Projektionsmuster kann die geometrische Form der 3D-Zieloberfläche genau berechnet werden.

Zur Oberflächenabbildung durch strukturiertes Licht stehen zahlreiche Techniken zur Verfügung (Abbildung 25).

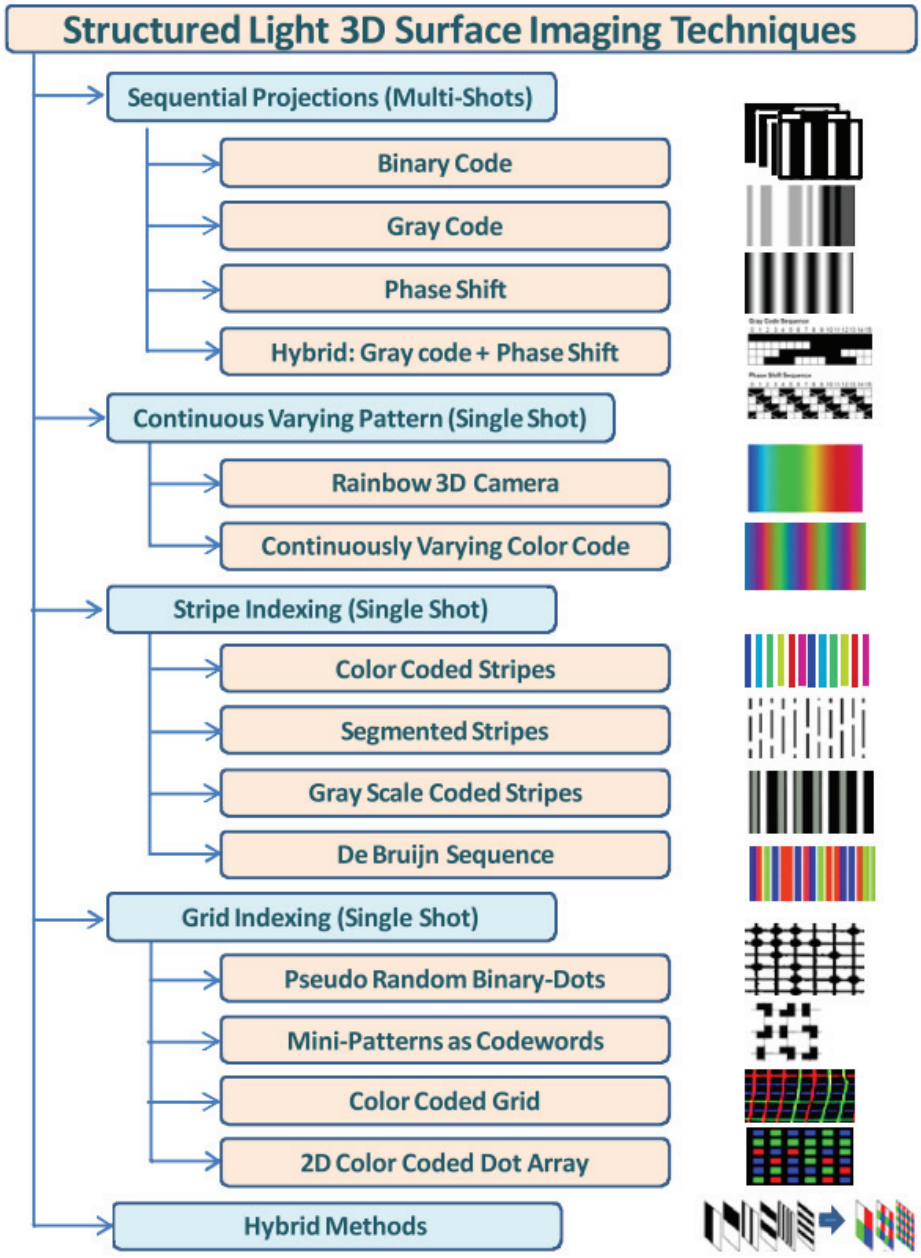

Abbildung 25: Übersicht verschiedener Techniken unter Verwendung von strukturiertem Licht [Geng, 2011]

In dieser Übersicht werden zunächst alle Techniken in sequenzielle (Multiple-Shot) oder Single-Shot-Kategorien klassifiziert. Wenn es sich bei dem 3D-Zielobjekt um ein statisches Objekt handelt und die Anwendung keine strenge Beschränkung der Erfassungszeit vorsieht, können Multiple-Shot-Techniken verwendet werden, die häufig zu zuverlässigeren und genaueren Ergebnissen führen. Wenn sich das Ziel jedoch bewegt, müssen Single-Shot-Techniken verwendet werden, um ein 3D-Oberflächenbild des 3D-Objekts zu einem bestimmten Zeitpunkt zu erfassen. [Geng, 2011]

3.5.2 Techniken der sequenziellen Projektion

Einige der Techniken aus Abbildung 25 werden im Folgenden erläutert.

3.5.2.1 Binäre Muster und Graukodierung

Die binäre Kodierung verwendet schwarze und weiße Streifen, um eine Folge von Projektionsmustern zu bilden, so dass jeder Punkt auf der Oberfläche des Objekts einen einzigartigen binären Code besitzt, der sich von allen anderen Codes verschiedener Punkte unterscheidet. Im Allgemeinen können N Muster 2^N Streifen kodieren. Abbildung 26 zeigt ein vereinfachtes 5-Bit-Projektionsmuster. Sobald diese Musterfolge auf eine statische Szene projiziert wird, gibt es 32 (=2^5) eindeutige Bereiche, die mit eindeutigen Streifen kodiert sind. Die 3D-Koordinaten (x,y,z) können (basierend auf einem Triangulationsprinzip) für alle 32 Punkte entlang jeder horizontalen Linie berechnet werden, wodurch ein Vollbild des 3D-Bildes entsteht.

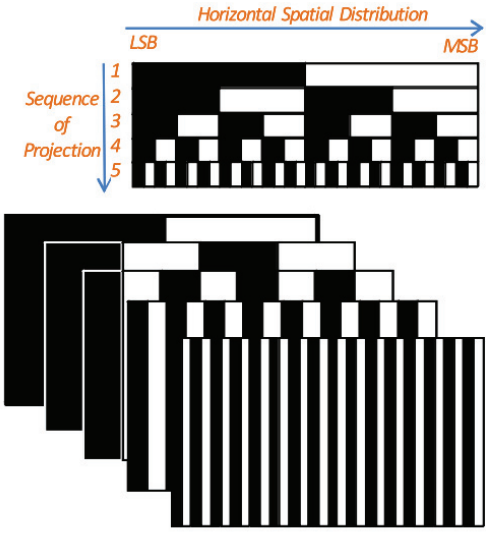

Abbildung 26: Darstellung eines Binärcode Projektionsmusters [Geng, 2011]

Die Binärkodierungstechnik ist sehr zuverlässig und weniger empfindlich gegenüber den Oberflächeneigenschaften, da in allen Pixeln nur binäre Werte existieren. Um jedoch eine hohe räumliche Auflösung zu erreichen, muss eine große Anzahl von sequenziellen Mustern projiziert werden. Alle Objekte in der Szene müssen statisch bleiben. Die gesamte Dauer der 3D-Bilderfassung kann länger sein, als es eine praktische 3D-Anwendung zulässt.

3.5.2.2 Graustufen-Muster

Um die Anzahl der Muster, die für ein hochauflösendes 3D-Bild benötigt werden, effektiv zu reduzieren, werden Graustufenmuster entwickelt. Zum Beispiel können M verschiedene Intensitätsstufen (anstelle von nur zwei im Binärcode) verwendet werden, um eine eindeutige Kodierung der Projektionsmuster zu erzeugen. In diesem Fall können N Muster M^N-Streifen codieren. Jeder Streifencode kann als ein Punkt in einem N-basierten Raum visualisiert

werden und jede Dimension hat M verschiedene Werte. Wenn z.B. N = 3 und M = 4 ist, dann beträgt die Gesamtzahl der Streifen mit einzigartigem Code 64 (4^3). Im Vergleich dazu werden für 64 Streifen mit einem Binärcode 6 Muster benötigt. Beim Entwurf der Binär- und Graukodierungsmuster gibt es eine Optimierung. Das Ziel ist es, eine Art Abstandsmaß zwischen allen eindeutigen Codewörtern zu maximieren. Für praktische 3D-Bildgebungsanwendungen ist es wichtig, benachbarte Streifen unterscheiden zu können. Abbildung 27 zeigt ein Beispiel für Graustufen-Kodierungsmuster.

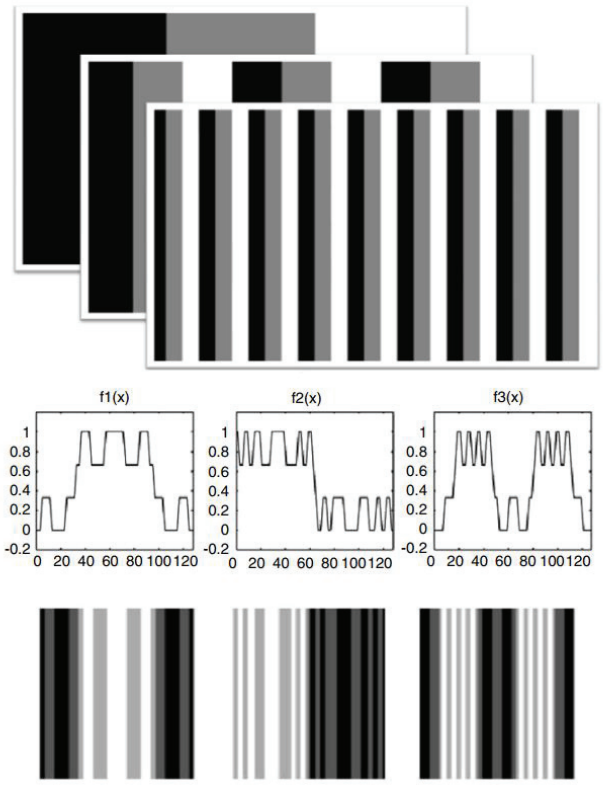

Abbildung 27: Graustufen-Koderiungsmuster [Geng, 2011]

3.5.2.3 Phasenverschiebung

Die Phasenverschiebung ist eine bekannte Streifenprojektionsmethode für die 3D-Oberflächenabbildung. Dabei wird eine Reihe von Sinusmustern auf die Objektoberfläche projiziert (Abbildung 28). Die Intensitäten für jedes Pixel (x, y) der drei projizierten Streifenmuster werden wie in Formel 11 beschrieben.

$$I_1(x,y) = I_0(x,y) + I_{mod}(x,y)\cos(\phi(x,y) - \theta),$$
$$I_2(x,y) = I_0(x,y) + I_{mod}(x,y)\cos(\phi(x,y)),$$
$$I_3(x,y) = I_0(x,y) + I_{mod}(x,y)\cos(\phi(x,y) + \theta),$$

Formel 11 [Geng, 2011]

Wobei $I_1(x, y)$, $I_2(x, y)$ und $I_3(x, y)$ die Intensitäten von drei Streifenmustern, $I_0(x, y)$ die DC-Komponente (Hintergrund), $I_{mod}(x, y)$ die Amplitude des Modulationssignals, $\varphi(x, y)$ die Phase und θ der konstante Phasenverschiebungswinkel sind.

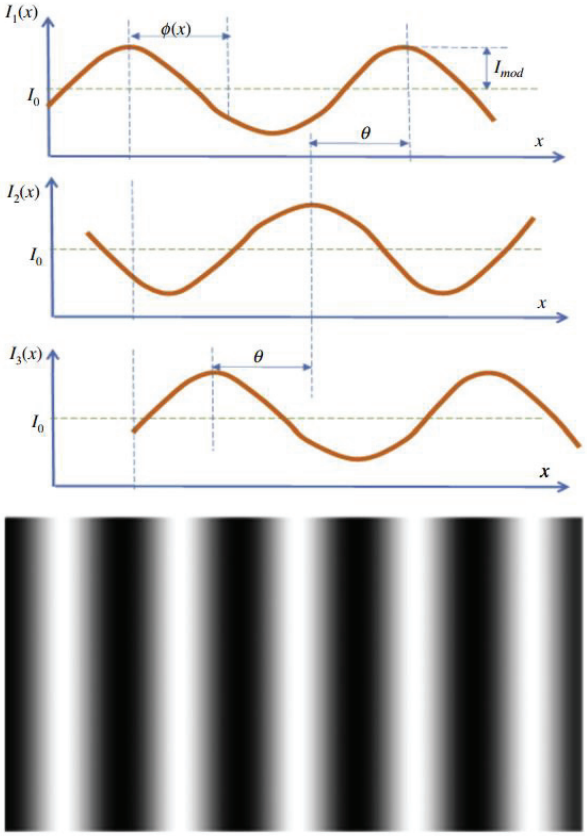

Abbildung 28: Phasenverschiebung mit drei Projektionsmustern [Geng, 2011]

Das Glätten der Phase ist der Prozess, der die ungeglättete Phase in die absolute Phase umwandelt. Die Phaseninformation $\varphi(x, y)$ kann aus den Intensitäten in den drei Randmustern abgerufen (d.h. geglättet) werden (Formel 12).

$$\phi' = \arctan\left[\sqrt{3}\frac{I_1(x,y) - I_3(x,y)}{2I_2(x,y) - I_1(x,y) - I_3(x,y)}\right]$$

Formel 12 [Geng, 2011]

Die Diskontinuität der Arcustangensfunktion bei 2π kann durch Addition oder Subtraktion des Vielfachen von 2π auf den $\varphi'(x, y)$ Wert entfernt werden (Abbildung 29, Formel 13).

$$\phi(x, y) = \phi'(x, y) + 2k\pi$$

Formel 13 [Geng, 2011]

Wobei k eine ganze Zahl ist, die den Projektionszeitraum darstellt. Die 3D-Koordinaten (x, y, z) können auf der Grundlage der Differenz zwischen der gemessenen Phase φ(x, y) und dem Phasenwert von einer Referenzebene berechnet werden. Abbildung 30 veranschaulicht einen einfachen Fall, in dem die Vereinfachung der Beziehung von Formel 14 zu Formel 15 führt.

$$\frac{Z}{L-Z} = \frac{d}{B}, \quad \text{or} \quad Z = \frac{L-Z}{B}d$$

Formel 14 [Geng, 2011]

$$Z \approx \frac{L}{B}d \propto \frac{L}{B}(\phi - \phi_0)$$

Formel 15 [Geng, 2011]

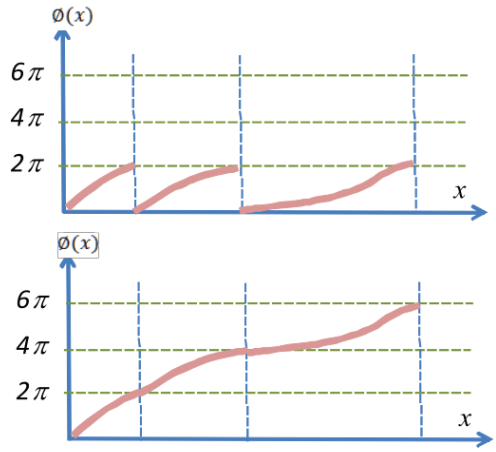

Abbildung 29: Prozess der Phasenglättung [Geng, 2011]

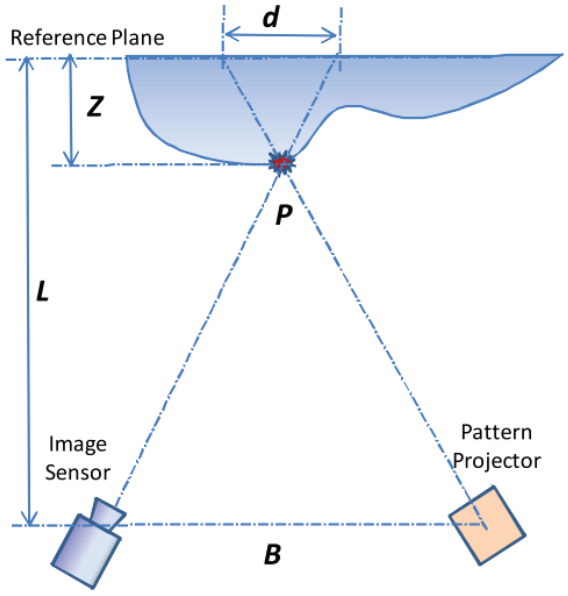

Abbildung 30: Ableitung der Tiefe Z [Geng, 2011]

3.6 Structure from Motion

Structure from Motion (SfM) ist eine photogrammetrische Methode zur Erstellung dreidimensionaler Modelle eines Objekts oder einer Topografie aus überlappenden zweidimensionalen Fotografien, die von vielen Orten und Ausrichtungen aus aufgenommen wurden, um die fotografierte Szene zu rekonstruieren. [Unavco, 2016]

Diese Aufgabe ist eine Herausforderung, da der Bildentstehungsprozess im Allgemeinen nicht umkehrbar ist. Von seiner projizierten Position in einer Kamerabildebene kann ein Szenenpunkt nur bis zu einer Ein-Parameter-Mehrdeutigkeit, die seiner Entfernung von der Kamera entspricht, wiederhergestellt werden. Daher werden zusätzliche Informationen benötigt, um das Rekonstruktionsproblem zu lösen. Eine weitere Möglichkeit ist die Verwendung korrespondierender Bildpunkte in mehreren Ansichten. Ausgehend von seinem Bild in zwei oder mehr Ansichten kann ein 3D-Punkt durch Triangulation rekonstruiert werden. Eine wichtige Voraussetzung ist die Bestimmung von Kamerakalibrierung und Pose, die durch eine Projektionsmatrix ausgedrückt werden kann. Die geometrische Theorie von Structure from Motion erlaubt die gleichzeitige Berechnung von Projektionsmatrizen und 3D-Punkten. [University of Cambridge, 2014]

3.6.1 Triangulation

Bei gegebenen Projektionsmatrizen können 3D-Punkte aus ihren gemessenen Bildpositionen in zwei oder mehr Ansichten berechnet werden. Dies ist eine Triangulation (Abbildung 31). Idealerweise sollten die 3D-Punkte am Schnittpunkt der rückprojizierten Strahlen liegen. Wegen des Messrauschens werden sich die rückprojizierten Strahlen jedoch im Allgemeinen nicht schneiden. Daher müssen 3D-Punkte so gewählt werden, dass eine geeignete Fehlermetrik minimiert wird. [University of Cambridge, 2014]

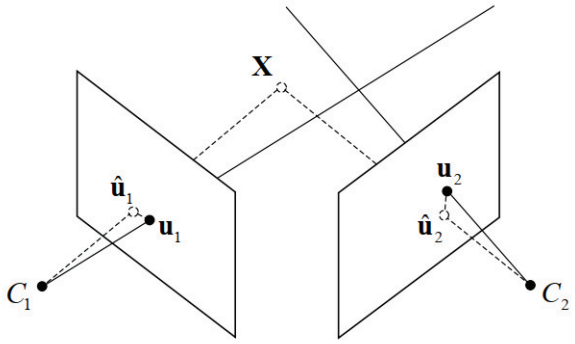

Abbildung 31: Ableitung von 3D Koordinaten aus zwei Kameraansichten [University of Cambridge, 2014]

3.6.2 Multi View Structure from Motion

Der nächste Schritt ist die Lösung des Structure and Motion-Problems für eine beliebige Anzahl von Ansichten. Die letzte Stufe ist in der Regel die Bündelausgleichung, die iterativ zur Verfeinerung der Structure and Motion-Parameter durch die Minimierung einer geeigneten Kostenfunktion verwendet wird. Die Bündelanpassung hängt jedoch entscheidend von einer geeigneten Initialisierung ab. Andernfalls kann der Algorithmus durch Konvergenz zu einer lokalen Lösung mit minimalen Kosten scheitern. In den folgenden Abschnitten werden Sequenz- und Faktorisierungsalgorithmen für die Multi View Structure from Motion diskutiert.

Sequenzielle Algorithmen arbeiten, indem sie aufeinanderfolgende Ansichten nacheinander einbeziehen (Abbildung 32). Während jede Ansicht registriert wird, wird eine Teilrekonstruktion erweitert, indem die Positionen aller 3D-Punkte, die in zwei oder mehr Ansichten sichtbar sind, mittels Triangulation berechnet werden. Eine geeignete Initialisierung erhält man typischerweise durch Zerlegung der fundamentalen Matrix, die sich auf die ersten beiden Ansichten der Sequenz bezieht.

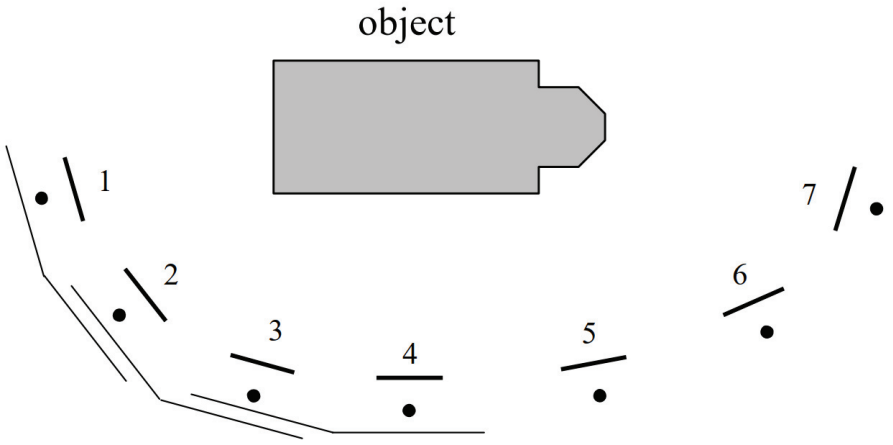

Abbildung 32: Aufnahme mehrerer Ansichten des Objekts [University of Cambridge, 2014]

Sequenzielle Registrierungssysteme besitzen einige Einschränkungen. Im Zusammenhang mit interaktiven Modellierungssystemen besteht ein Nachteil darin, dass in jeder Ansicht eine große Anzahl von korrespondierenden Punkten definiert werden muss. Da korrespondierende Punkte in der Regel in drei oder mehr Ansichten sichtbar sein müssen, bedeutet dies, dass eine erhebliche Überlappung erforderlich ist.

Im Gegensatz zu sequenziellen Methoden arbeiten Batch-Methoden durch die Berechnung der Kamerapose und der Szenengeometrie unter gleichzeitiger Verwendung aller Bildmessungen. Ein Vorteil besteht darin, dass Rekonstruktionsfehler sinnvoll auf alle Messungen verteilt werden können. Dadurch können grobe Fehler, die mit dem Sequenzabschluss verbunden sind, vermieden werden. [University of Cambridge, 2014]

3.6.3 Bündelausgleichung

Aus den Bild-Features ergibt sich eine erste Schätzung der Projektionsmatrizen und der 3D-Punkte. In der Regel wird es notwendig sein, diese Schätzung durch iterative nichtlineare Optimierung zu verfeinern, um eine geeignete Kostenfunktion zu minimieren. Dies ist die Bündelausgleichung. Die Bündelausgleichung funktioniert durch Minimierung einer Kostenfunktion, die sich auf eine gewichtete Summe quadratischer Reproduktionsfehler bezieht. Gewöhnlich wird die Gauß-Newton-Iteration für schnelle Konvergenz verwendet.

Das Ziel der Bündelausgleichung ist die Bestimmung einer optimalen Schätzung eines Satzes von Parametern θ bei einer Reihe von verrauschten Beobachtungen. Die meisten Bündelparameter können nicht direkt beobachtet werden, z.B. Projektionsmatrizen oder 3D-Punktkoordinaten. Stattdessen erlauben sie Vorhersagen von Größen, die z.B. die gemessenen Pixelkoordinaten von abgebildeten 3D-Punkten vorhersagen können.

Die Menge der Vorhersagen sei z(θ) und die Menge der entsprechenden Beobachtungen sei $\underline{z}$. Der restliche Vorhersagefehler Δz sei dann gegeben durch Formel 16.

$$\Delta \mathbf{z} = \bar{\mathbf{z}} - \mathbf{z}(\boldsymbol{\theta})$$

Formel 16 [University of Cambridge, 2014]

Im Allgemeinen kann der Beobachtungsvektor $\underline{z}$ in einen Satz von statistisch unabhängigen Messungen $\underline{z}_1 \ldots \underline{z}_N$ mit zugehörigen Vorhersagen unterteilt werden $\underline{z}_1$ (θ) . . . $\underline{z}_N$ (θ). Die Bündelanpassung erfolgt durch Minimierung einer geeigneten Kostenfunktion. Unter der Annahme eines Gauß-verteilten Messrauschens ist die geeignete Kostenfunktion eine Summe der quadrierten Fehler, die die negative Summe der logarithmischen Wahrscheinlichkeiten ist (Formel 17).

$$f(\boldsymbol{\theta}) = \frac{1}{2} \sum_i \Delta \mathbf{z}_i(\boldsymbol{\theta})^\top \mathbf{W}_i \Delta \mathbf{z}_i(\boldsymbol{\theta}) \qquad \Delta \mathbf{z}_i(\boldsymbol{\theta}) = \bar{\mathbf{z}}_i - \mathbf{z}_i(\boldsymbol{\theta})$$

Formel 17 [University of Cambridge, 2014]

Hier ist $\Delta z_i(\theta)$ der Feature-Prädiktionsfehler und W_i ist eine symmetrische Matrix mit positiv definierter Gewichtung, die zur Approximation der inversen Kovarianz des mit der Messung $\underline{z}_i$ verbundenen Messrauschens gewählt wird.

Die Bündelanpassung wird durch die Anpassung einer lokalen quadratischen Approximation an die Kostenfunktion bei jeder Iteration erreicht. In der Praxis hängt die Konvergenzrate kritisch von der Qualität dieser Approximation ab. Daher ist es wichtig, eine Parameterdarstellung so zu wählen, dass die Auswirkung der Parametervariation lokal so nah wie möglich linear auf das gewählte Kostenmodell ist. Ein interessantes Problem ist, die

Kameraausrichtungen so zu parametrisieren, dass die bekannten Singularitäten, die mit Euler-Winkeldarstellungen verbunden sind, vermieden werden. Eine gute Lösung ist, die Kameraausrichtungen R_i als 3 × 3 Rotationsmatrizen zu belassen. [University of Cambridge, 2014]

4. Vermaschungsalgorithmen

Das Ergebnis der zuvor vorgestellten Verfahren ist eine Punktwolke bestehend aus dreidimensionalen Punkten. Diese Punktwolken können vermascht werden. Dadurch entsteht ein Drahtgittermodell. Die Vermaschung kann z.B. nach Delaunay oder Poisson erfolgen.

4.1 Delaunay

Die Delaunay-Triangulation ist eine der beliebtesten und am häufigsten verwendeten Methoden bei Problemen im Zusammenhang mit der Erzeugung von Vermaschungen. Viele der optimalen Eigenschaften der Delaunay-Triangulation sind in 2D bekannt, wo sie in den letzten zwanzig Jahren intensiv studiert wurde, obwohl die Grundlagen zu Beginn des zwanzigsten Jahrhunderts formuliert wurden. Sie kann jedoch ebenso im 3D Raum verwendet werden. [Maur, 2002]

4.1.1 Zweidimensional

Die Delaunay-Triangulation ist eine geometrische Struktur, die Ingenieure seit den Anfängen der Vermaschung von Punkten verwenden. In zwei Dimensionen hat sie einen markanten Vorteil. Unter allen möglichen Triangulationen einer festen Menge von Punkten maximiert die Delaunay-Triangulation den minimalen Winkel. Sie optimiert auch mehrere andere geometrische Kriterien im Zusammenhang mit der Interpolationsgenauigkeit.

Das Wort Triangulation bezieht sich normalerweise auf einen vereinfachenden Komplex, aber es hat mehrere Bedeutungen, wenn eine Triangulation einer geometrischen Einheit diskutiert wird, die trianguliert wird. Es gibt Triangulationen von Punktmengen, Polygonen, Polyedern und vielen anderen Strukturen.

Um eine Triangulation zu konstruieren, werden die Punkte lexikographisch (d.h. nach der x-Koordinate, wobei Punkte mit derselben x-Koordinate nach ihren y-Koordinaten geordnet werden) sortiert (Abbildung 33).

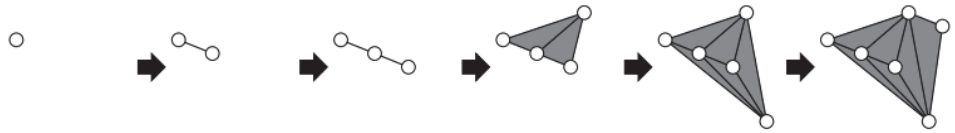

Abbildung 33: Delaunay-Vermaschung im zweidimensionalen Raum [Shewchuk, 2012]

Die von Boris Nikolajewitsch Delaunay 1934 eingeführte Delaunay-Triangulation einer Punktmenge S ist durch die Eigenschaft der leeren Umkreisscheibe gekennzeichnet. Kein Punkt in S liegt im Inneren der umschreibenden Scheibe eines Dreiecks.

Im Zusammenhang mit einer endlichen Punktmenge S handelt es sich um ein Delaunay-Dreieck, wenn seine Eckpunkte in S liegen und seine offene Umkreisscheibe keinen Punkt in S enthält. Zu beachten gilt, dass eine beliebige Anzahl von Punkten in S auf dem Umkreis eines Delaunay-Dreiecks liegen kann. Eine Dreiecksseite erfüllt die Delaunay-Anforderungen, wenn ihre Scheitelpunkte in S liegen und sie mindestens eine Umkreisscheibe besitzt. Eine Delaunay-Triangulation von S, ist eine Triangulation, in der jedes Dreieck ein Delaunay-Dreieck ist (Abbildung 34). [Berkley EECS, 2012]

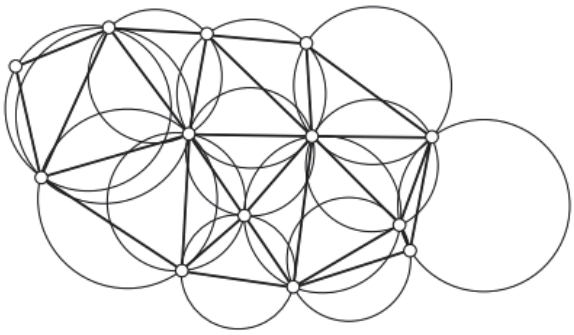

Abbildung 34: Darstellung der Umkreise im zweidimensionalen Raum [Shewchuk, 2012]

4.1.2 Dreidimensional

Im dreidimensionalen Raum verhält es sich ähnlich wie im zweidimensionalen Raum. Die Kreise werden hier jedoch rotiert, da sich die Punkte nicht alle in einer Ebene befinden (Abbildung 35). Da bei Punkten mit dreidimensionalen Koordinaten zusätzlich eine z-Koordinate vorhanden ist, wird erst nach den x-Koordinaten sortiert, anschließend nach den y-Koordinaten und zum Schluss nach den z-Koordinaten.

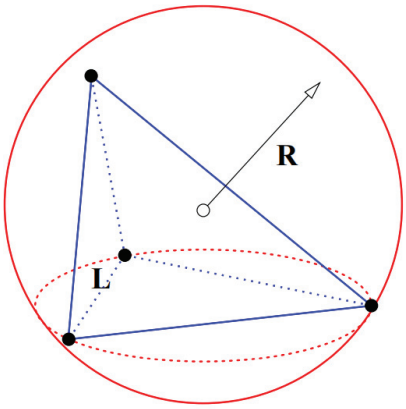

Abbildung 35: Darstellung der Umkreise im dreidimensionalen Raum [Si, 2007]

Dreidimensionale Triangulationen werden manchmal als Tetraederalisierungen bezeichnet. Delaunay-Tetraederalisierungen sind nicht ganz so effektiv wie planare Delaunay-Triangulationen, um Elemente von guter Qualität zu erzeugen, aber sie sind in der Literatur zur Netzgenerierung fast so beliebt wie ihre zweidimensionalen Vettern. Viele Eigenschaften von Delaunay-Triangulationen in der Ebene verallgemeinern sich zu höheren Dimensionen. Wie in Abbildung 35 zu sehen wird aus dem zuvor verwendeten Umkreis im 2D-Raum eine Umkreiskugel für den 3D-Raum. [Shewchuck, 2012]

4.2 Poisson

Eine weitere Möglichkeit Oberflächenrekonstruktionen durchzuführen ist der Poisson-Algorithmus. Dieser Ansatz wurde von Michael Kazhdan vorgestellt und wird unter anderem in der Netzverarbeitungssoftware MeshLab sowie in der Software Metashape verwendet.

Der Algorithmus erfordert zuvor drei Schritte als Vorbereitung. Schritt 1 beinhaltet das Finden der Tangentialebenen zur Annäherung an die lokale Oberfläche. Um dies zu erreichen, wird der Schwerpunkt jedes Scheitels als Mittelwert aller k-nächsten Nachbarn berechnet. Dann wird die entsprechende Normale durch ihren Eigenvektor berechnet. Schritt 2 ist der Aufbau des Riemannschen Graphen. Der Riemannsche Graph wird benötigt, weil er eine Struktur erzeugt, in der für jeden Scheitelpunkt die Kanten zu seinen k-nächsten Nachbarn enthalten sind, was für den nächsten Schritt benötigt wird. Schritt 3 ist die Verwendung des Minimum Spanning Tree-Algorithmus von Kruskal für den Riemannschen Graphen, um die Normalen zu gewichten. Der Algorithmus berechnet die Normalen als gewichtetes Produkt über die k-nächsten Nachbarn. Das Ergebnis ist ein sehr glattes Modell (Abbildung 36). Die rot markierte Normale wird unter Verwendung der sechs nächstgelegenen Nachbarn berechnet.

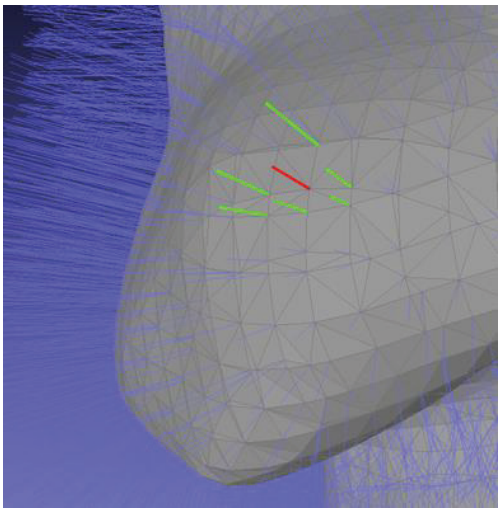

Abbildung 36: Extrapolation der Normalen [TU Freiberg, o. J.]

Unter der Annahme, dass die Punkte im 3D Raum die Oberfläche des Objekts bilden, wird die Isofläche des Modells aus dem Normalfeld approximiert, das im Verhältnis zum Gradienten der die Isofläche beschreibenden Indikatorfunktion steht. Dieses Problem wird als die Lösung eines Poisson-Problems betrachtet. Abbildung 37 veranschaulicht diesen Ansatz für ein 2D-Beispiel.

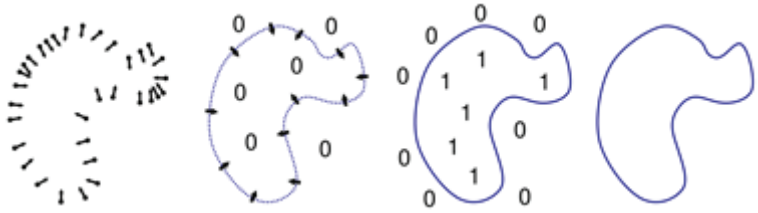

Abbildung 37: Extrahieren der Isoflächen [TU Freiberg, o. J.]

Zuerst werden die Punkte mit den orientierten Normalen genommen und die Steigung der Indikatorfunktion durch das Normalfeld approximiert. Danach wird aus diesem Gradienten die Indikatorfunktion, die überall außer in der Nähe der Oberfläche Null ist, abgeleitet und die Oberfläche des Objekts kann extrahiert werden. [TU Freiberg, o. J.]

5. Kamerakalibrierung

Die Kamerakalibrierung ist notwendig, um Kameramessungen mit Messungen in der realen Welt in Beziehung zu setzen. Dies ist erforderlich, weil Szenen nicht nur dreidimensional sind, sondern auch physische Räume mit physischen Einheiten darstellen. Daher ist die Beziehung zwischen den natürlichen Einheiten der Kamera (Pixel) und den Einheiten der physischen Welt (z. B. Meter) eine entscheidende Komponente bei jedem Versuch, eine dreidimensionale Szene zu rekonstruieren. Der Prozess der Kamerakalibrierung führt zu einem Modell der Geometrie der Kamera als auch zu einem Verzerrungsmodell des Objektivs. Diese beiden Informationsmodelle definieren die intrinsischen Parameter der Kamera. [Kaehler et al., 2017]

5.1 Kameramodell

Begonnen wird mit der Betrachtung des einfachsten Modells einer Kamera, der Lochblende. In diesem einfachen Modell wird das Licht von der Szene oder einem entfernten Objekt einfallend dargestellt, aber nur ein einziger Strahl tritt von einem bestimmten Punkt in der Szene in die Lochblende ein. In einer physikalischen Lochkamera wird dieser Punkt dann auf eine Abbildungsfläche projiziert. Folglich ist das Bild auf dieser Bildebene (auch Projektionsebene genannt) immer scharf abgebildet und die Größe des Bildes relativ zum entfernten Objekt wird durch die Brennweite der Kamera bestimmt. Für die idealisierte Lochkamera entspricht der Abstand von der Lochblende zur Abbildung genau der Brennweite. Dies ist in Abbildung 38 dargestellt, wobei f die Brennweite der Kamera, Z die Entfernung von der Kamera zum Objekt, X die Länge des Objekts und x das Bild des Objekts auf der Bildebene ist.

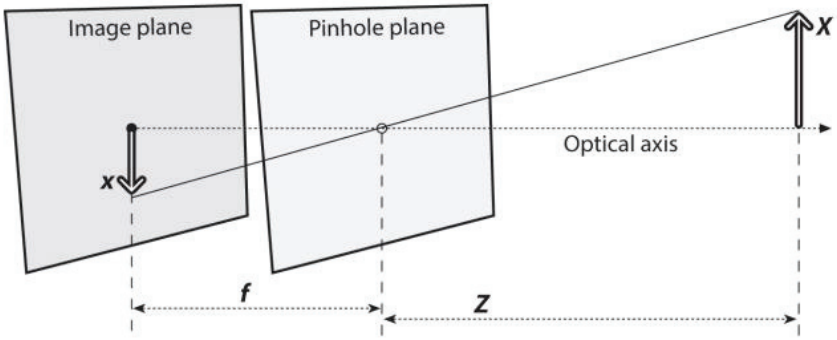

Abbildung 38: Prinzip der Lochkamera [Kaehler et al., 2017]

Das ursprüngliche Lochkameramodell wird nun in eine Form gebracht, die gleichwertig ist, aber in der die Mathematik einfacher ist. In Abbildung 39 werden Lochblende und Bildebene vertauscht. Der Hauptunterschied besteht darin, dass das Objekt jetzt mit der rechten Seite nach oben erscheint. Der Punkt in der Lochblende wird als Mittelpunkt der Projektion umgedeutet. Bei dieser Betrachtungsweise hinterlässt jeder Strahl einen Punkt auf dem entfernten Objekt und steuert das Zentrum der Projektion an. Der Punkt im Schnittpunkt der Bildebene und der optischen Achse wird als Hauptpunkt bezeichnet. Auf dieser neuen frontalen Bildebene (Abbildung 39), die der alten Projektions- oder Bildebene entspricht, hat das Bild des fernen Objekts genau die gleiche Größe wie auf der Bildebene in Abbildung 38. Das Bild wird erzeugt, indem diese Strahlen mit der Bildebene geschnitten werden, die sich genau in einem Abstand f vom Projektionszentrum befindet. Dadurch wird das ähnliche Dreiecksverhältnis $\frac{x}{f} = \frac{X}{F}$ direkter als zuvor deutlich.

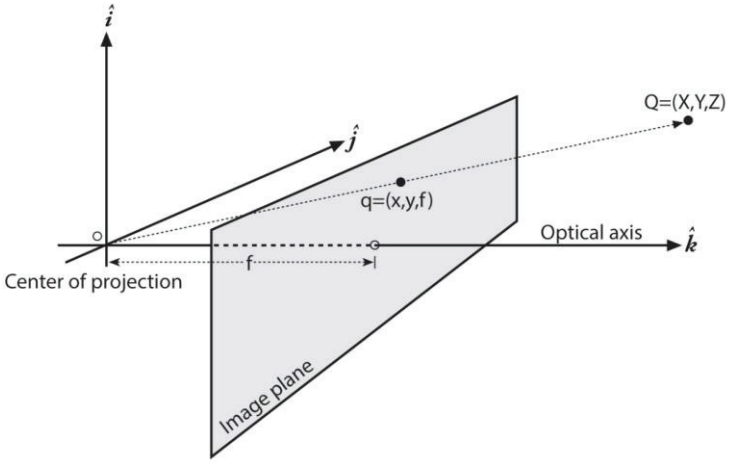

Abbildung 39: Vertauschung von Lochblende und Bildebene [Kaehler et al., 2017]

Da der Hauptpunkt nicht exakt mit der Mitte des Bildwandlers übereinstimmt, werden die zwei Parameter c_x und c_y eingeführt, um eine Verschiebung (weg von der optischen Achse) des Koordinatenzentrums auf der Projektionsfläche zu modellieren. Das Ergebnis ist ein relativ einfaches Modell, bei dem ein Punkt Q in der physikalischen Welt, dessen Koordinaten (X, Y, Z) sind, an einer durch (x_{screen}, y_{screen}) gegebenen Pixelposition gemäß Formel 18 auf den Bildwandler projiziert wird.

$$x_{screen} = f_x \cdot \frac{X}{Z} + c_x \text{ , and } y_{screen} = f_y \cdot \frac{Y}{Z} + c_y$$

Formel 18 [Kaehler et al., 2017]

Zu beachten gilt, dass zwei unterschiedliche Brennweiten eingeführt wurden. Der Grund dafür ist, dass die einzelnen Pixel auf einem typischen kostengünstigen Bildwandler rechteckig und nicht quadratisch sind. Die Brennweite f_x zum Beispiel ist eigentlich das Produkt aus der physikalischen Brennweite des Objektivs und der Größe s_x der einzelnen Bildelemente des Bildwandlers. Ähnliche Aussagen gelten natürlich auch für f_y und s_y. Es ist jedoch zu bedenken, dass s_x und s_y nicht direkt über ein Kamerakalibrierungsverfahren gemessen werden können und auch die physikalische Brennweite f ist nicht direkt messbar. Abgeleitet werden können nur die Kombinationen $f_x = F - s_x$ und $f_y = F – s_y$, ohne die Kamera tatsächlich zu zerlegen und ihre Komponenten direkt zu messen. [Kaehler et al., 2017]

5.2 Drehungsmatrix Verschiebungsvektor

Für jedes Bild, das die Kamera von einem bestimmten Objekt aufnimmt, kann die Pose des Objekts relativ zum Kamera-Koordinatensystem in Form einer Rotation und einer Translation beschrieben werden (Abbildung 40).

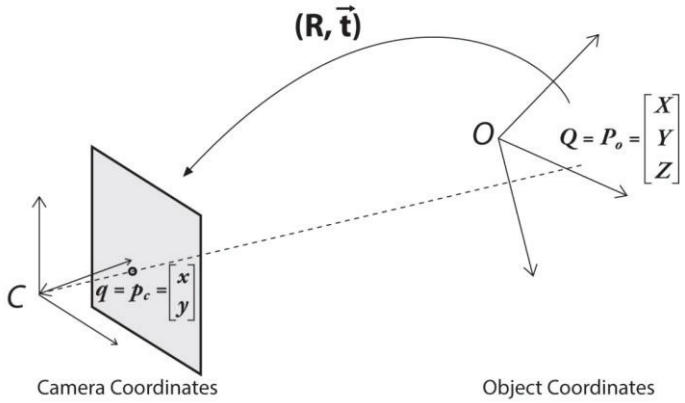

Abbildung 40: Konvertieren der Objektkoordianten ins Kamera-Koordinatensystem [Kaehler et al., 2017]

Im Allgemeinen kann eine Rotation in beliebig vielen Dimensionen durch die Multiplikation eines Koordinatenvektors mit einer quadratischen Matrix geeigneter Größe beschrieben werden. Letztendlich ist eine Drehung gleichbedeutend mit der Einführung einer neuen Beschreibung der Lage eines Punktes in einem anderen Koordinatensystem. Eine Drehung des Koordinatensystems um einen Winkel θ ist gleichbedeutend mit einer Gegendrehung des Zielpunktes um den Ursprung dieses Koordinatensystems um den gleichen Winkel θ. Die Darstellung einer zweidimensionalen Drehung als Matrixmultiplikation ist in Abbildung 41 dargestellt. Eine dreidimensionale Drehung kann in eine zweidimensionale Drehung um jede Achse zerlegt werden, bei der die Messungen der Drehachse konstant bleiben. Wenn nacheinander um die x-, y- und z-Achsen mit den entsprechenden Rotationswinkeln ψ, φ und θ rotiert wird, ist das Ergebnis eine Gesamtrotationsmatrix R, die durch das Produkt der drei Matrizen Rx(ψ), Ry(φ) und Rz(θ) gegeben ist (Formel 19).

$$R = R_x(\psi) \cdot R_y(\varphi) \cdot R_z(\theta)$$

Formel 19 [Kaehler et al., 2017]

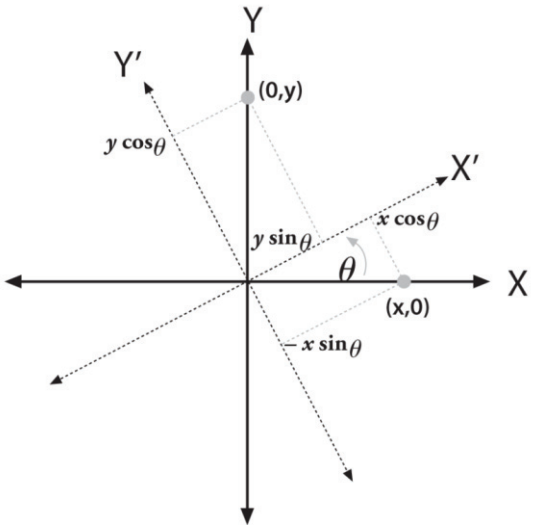

Abbildung 41: Darstellung einer zweidimensionalen Drehung [Kaehler et al., 2017]

Der Translationsvektor ist die Art und Weise, wie eine Verschiebung von einem Koordinatensystem in ein anderes System dargestellt wird, dessen Ursprung an einen anderen Ort verschoben ist. Mit anderen Worten, der Translationsvektor ist nur der Versatz vom Ursprung des ersten Koordinatensystems zum Ursprung des zweiten Koordinatensystems. Um also von einem auf ein Objekt zentrierten Koordinatensystem zu einem auf die Kamera zentrierten Koordinatensystem zu wechseln, ist der entsprechende Translationsvektor einfach $T = \text{Ursprung}_{\text{Objekt}} - \text{Ursprung}_{\text{Kamera}}$. Daraus resultiert (unter Bezugnahme auf Abbildung 40), dass ein Punkt im Objekt- (oder Welt-) Koordinatenrahmen P_o die Koordinaten P_c im Kamerakoordinatenrahmen hat (Formel 20).

$$\vec{P}_c = R \bullet (\vec{P}_o - \vec{T})$$

Formel 20 [Kaehler et al., 2017]

Die Kombination dieser Gleichung für P_c mit den kamerainternen Korrekturen bildet das grundlegende Gleichungssystem. Die Lösung dieser Gleichung liefert die Kamerakalibrierungsparameter.

Bei den in OpenCV verwendeten Kalibrierobjekten handelt es sich um flache Muster verschiedener Typen. Im folgenden wird weiter auf das Schachbrett als Kalibrierobjekt eingegangen. [Kaehler et al., 2017]

5.3 Kalibriertafeln

Prinzipiell könnte jedes entsprechend charakterisierte Objekt als Kalibrierobjekt verwendet werden. Eine praktische Wahl ist ein regelmäßiges Muster auf einer ebenen Oberfläche, wie z.B. ein Schachbrett (Abbildung 42), ein Zufallsmuster (Abbildung 43), Aruco- (Abbildung 44) oder Charuco-Muster (Abbildung 45).

Einige Kalibrierungsmethoden in der Literatur stützen sich auf dreidimensionale Objekte (z.B. eine mit Markern bedeckte Schachtel), aber flache Schachbrettmuster sind viel einfacher zu handhaben. Unter anderem ist es ziemlich schwierig, präzise dreidimensionale Kalibrierungsobjekte herzustellen. OpenCV entscheidet sich daher für die Verwendung mehrerer Ansichten eines ebenen Objekts anstelle einer einzigen Ansicht eines speziell konstruierten dreidimensionalen Objekts. Die Verwendung eines Musters aus abwechselnd schwarzen und weißen Quadraten stellt sicher, dass es bei der Messung keine Verzerrung nach der einen oder anderen Seite gibt (Abbildung 42). [Kaehler et al., 2017]

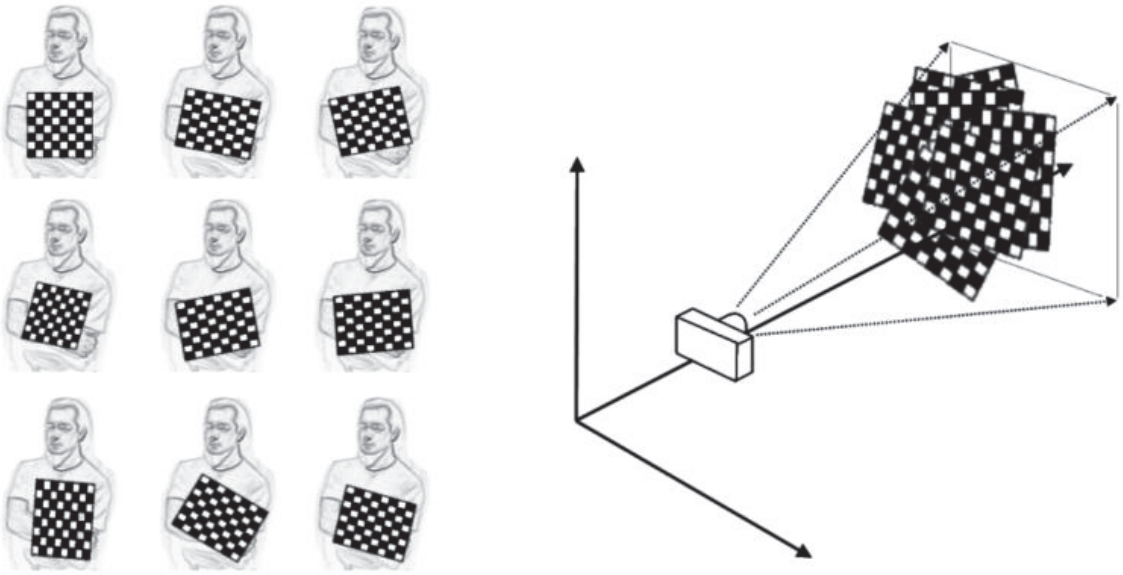

Abbildung 42: Schachbrettmuster [Kaehler et al., 2017]

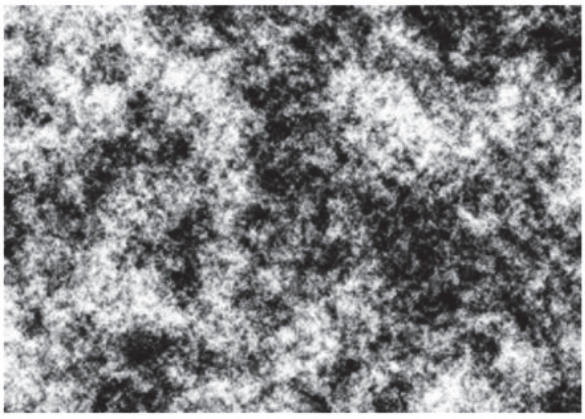

Abbildung 43: Zufallsmuster [Kaehler et al., 2017]

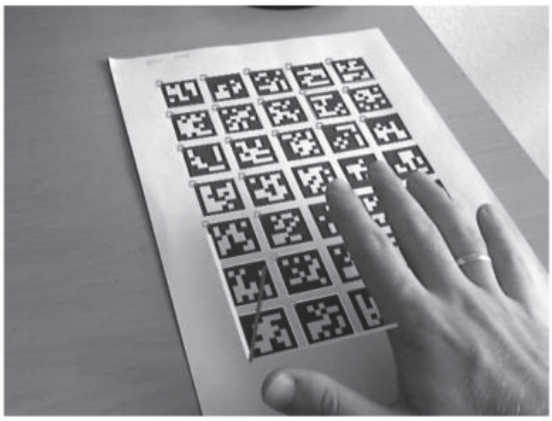

Abbildung 44: Aruco-Muster [Kaehler et al., 2017]

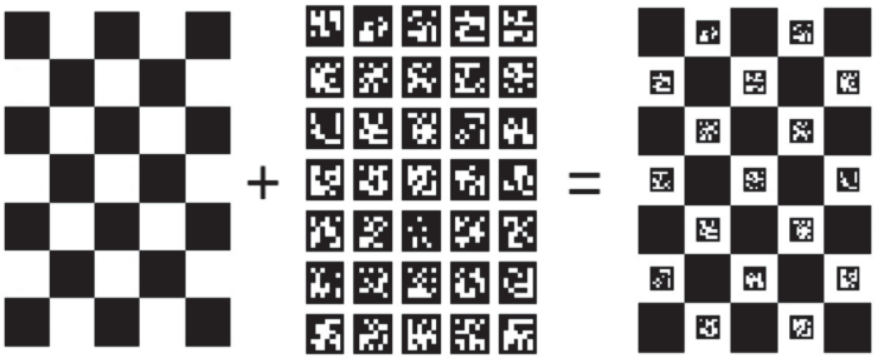

Abbildung 45: Charuco-Muster [Kaehler et al., 2017]

5.4 Homographie

In der Computer Vision wird planare Homographie als eine projektive Abbildung von einer Ebene auf eine andere definiert. Daher ist die Abbildung von Punkten auf einer zweidimensionalen planaren Oberfläche auf den Bildwandler der Kamera ein Beispiel für planare Homographie. Es ist möglich, diese Abbildung in Form einer Matrixmultiplikation auszudrücken, wenn homogene Koordinaten verwendet werden, um sowohl den betrachteten Punkt Q als auch den Punkt q auf dem Bildwandler auszudrücken, auf dem Q abgebildet ist. Wenn Q und q wie in Formel 21 definiert werden, kann die Wirkung der Homographie wie in Formel 22 ausgedrückt werden.

$$\vec{Q} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \text{and } \vec{q} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Formel 21 [Kaehler et al., 2017]

$$\vec{q} = s \bullet H \bullet \vec{Q}$$

Formel 22 [Kaehler et al., 2017]

Bei dem eingeführten Parameter s handelt es sich um einen willkürlichen Skalenfaktor. Dieser soll deutlich machen, dass die Homographie nur bis zu diesem Faktor definiert ist.

Diese Transformationsmatrix kann mithilfe von Geometrie und Matrixalgebra gelöst werden. Die wichtigste Beobachtung ist, dass H aus zwei Teilen besteht. Die physikalische Transformation, die im Wesentlichen die Objektebene, welche betrachtet wird, lokalisiert und die Projektion, die die kamerainterne Matrix einführt (Abbildung 46).

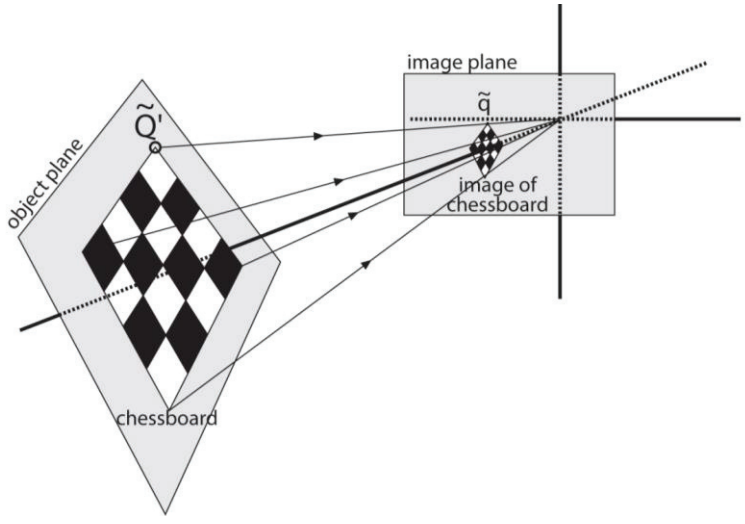

Abbildung 46: Darstellung eines planaren Objekts beschrieben durch Homographie [Kaehler et al., 2017]

Der physikalische Transformationsteil ist die Summe der Effekte einer gewissen Rotation R und einer gewissen Translation t, die die Ebene, die betrachtet wird, mit der Bildebene in Beziehung setzen. Da in homogenen Koordinaten gearbeitet wird, können diese innerhalb einer einzigen Matrix wie in Formel 23 kombiniert werden.

$$W = \left[R \ \vec{t}\right]$$

Formel 23 [Kaehler et al., 2017]

Anschließend wird die Kameramatrix M (Formel 24) mit Q multipliziert (Formel 25).

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_x & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Formel 24 [Kaehler et al., 2017]

$$\vec{q} = s \cdot M \cdot W \cdot \vec{Q}$$

Formel 25 [Kaehler et al., 2017]

Das Interesse besteht in der Praxis nicht in der Koordinate Q, die für den gesamten Raum definiert ist, sondern in der Koordinate Q‘, die nur auf der Ebene definiert ist, welche betrachtet wird. Dies führt zu einer Vereinfachung. Die Objektebene wird mit Z = 0 definiert. Der Grund dafür ist das Aufbrechen der Rotationsmatrix in drei mal eine Spalte ($R = [\,r1 \rightarrow r2 \rightarrow \ r3 \rightarrow]$) (Formel 26). Dadurch wird eine dieser Spalten nicht mehr benötigt.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = s \cdot M \cdot \left[\vec{r}_1 \ \vec{r}_2 \ \vec{r}_3 \ \vec{t}\right] \cdot \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = s \cdot M \cdot \left[\vec{r}_1 \ \vec{r}_2 \ \vec{t}\right] \cdot \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Formel 26 [Kaehler et al., 2017]

Die Homographie-Matrix H, die die Punkte eines planaren Objekts auf dem Bildgeber abbildet, wird dann vollständig beschrieben durch $H = s \cdot M \cdot [\,r1 \rightarrow \ r2 \rightarrow \ \ t \rightarrow]$.

OpenCV verwendet die vorhergehenden Gleichungen zur Berechnung der Homographie-Matrix. Verwendet werden mehrere Bilder desselben Objekts, um die einzelnen Translationen und Rotationen für jede Ansicht zu berechnen. Die Rotation wird durch drei Winkel beschrieben und die Translation durch drei Offsets. Daher gibt es sechs Unbekannte für jede

Ansicht. Das ist in Ordnung, denn ein bekanntes planares Objekt (wie das Schachbrett) ergibt acht Gleichungen, d.h. die Abbildung eines Quadrats als ein Viereck kann durch vier (x, y) Punkte beschrieben werden. Jeder neue Rahmen liefert acht Gleichungen auf Kosten von sechs neuen extrinsischen Unbekannten, so dass es bei genügend Bildern möglich ist, eine beliebige Anzahl von intrinsischen Unbekannten zu berechnen. [Kaehler et Al., 2017]

6. Anwendung auf industrielle Schmiedestücke

Auf Grundlage der vorherigen Kapitel muss nun entschieden werden, welcher Algorithmus und welches Verfahren sich am besten für die Anwendung eignet. Für diese Entscheidung dient ein Vergleich der unterschiedlichen Verfahren mit Beleuchtung der Vor- und Nachteile. Bezogen auf industrielle Schmiedestücke ist das Kriterium die Zeit. Deshalb gilt es das zeitliche Limit von drei Sekunden zu beachten.

6.1 Wahl eines geeigneten Algorithmus

Unter Punkt 3.1 wurde die Theorie zur Bestimmung der dreidimensionalen Lage durch eine einzelne Kamera erläutert. Wie der Name vermuten lässt, wird bei dieser Anwendung eine einzelne Kamera und auch nur eine einzelne Aufnahme des Objektes verwendet. Damit dieser Algorithmus erfolgreich eingesetzt werden kann, ist eine gewisse Anzahl von Schlüsselpunkten auf dem Objekt notwendig. Diese Schlüsselpunkte müssen eindeutig identifizierbar sein. Ebenso muss die Lage der Schlüsselpunkte innerhalb des Objekt-Koordinatensystems, in Form von dreidimensionalen Koordinaten, vorliegen.

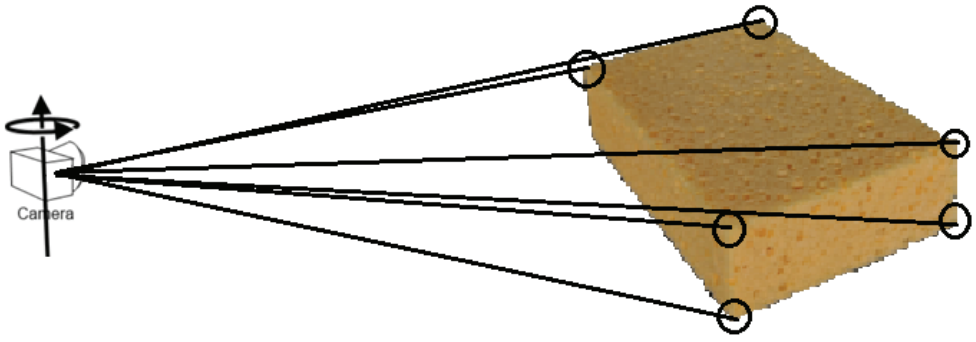

Abbildung 47: Prinzip zur Bestimmung der dreidimensionalen Lage durch eine einzelne Kamera

Diese Punkte müssen gleich bleiben und dürfen sich nicht verändern. Da sich das Schmiedestück während des Arbeitsablaufs durch Verformung stark verändert, würde sich die Position der Schlüsselpunkte auf dem Schmiedestück genauso stark verändern. Des Weiteren wäre eine Anbringung von Markern durch hohe Wärmeentwicklung schwierig. Hinzu kommt, dass mit einer einzelnen Kamera und einer einzelnen Aufnahme die Abbildung des Objektes nur aus einem Blickwinkel zulässt. Da das Schmiedestück während des Arbeitsablaufes auch

gedreht wird, ist die Betrachtung aus mehreren Blickwinkeln nötig. Die Positionierung der Kamera an verschiedenen Standpunkten wäre zu zeitintensiv.

Wie unter Punkt 3.2 deutlich wurde, ist bei Stereo Vision eine Abschätzung der Tiefeninformation ohne koordinatenmäßig bekannte Punkte auf dem Objekt möglich. Die Implementierungskosten sind gering, da handelsübliche Kameras nach Anwendung der Stereokalibrierung verwendet werden können.

Dazu werden in der Regel zwei durch einen Abstand voneinander getrennte Kameras in einer physikalischen Anordnung ähnlich den menschlichen Augen verwendet. Bei einem punktförmigen Objekt im Raum führt die Trennung der Kameras zu einer messbaren Disparität der Objektpositionen in den beiden Kamerabildern. Die große Herausforderung dabei ist die Lösung des Korrespondenzproblems. Solange die Korrespondenz nicht festgestellt werden kann, kann die Disparität und damit die Tiefe nicht genau bestimmt werden. Die Lösung des Korrespondenzproblems erfordert komplexe, rechenintensive Algorithmen zur Merkmalsextraktion und zum Abgleich. Merkmalsextraktion und Abgleich erfordern auch eine ausreichende Intensität und Farbvariation im Bild für eine robuste Korrelation. Beim Stereosehen ist der Tiefenauflösungsfehler eine quadratische Funktion der Entfernung wodurch die Entfernung zum Objekt beschränkt wird. Somit bewegt sich die Genauigkeit der Tiefeninformation im unteren Bereich, was in Abbildung 48 deutlich wird. Klare Kanten und somit eindeutige Tiefeninformationen sind nicht vorhanden. [Li, 2014]

Abbildung 48: Darstellung der Tiefenauflösung durch Verwendung von Stereo Vision [Mattoccia, 2008]

Bezogen auf die Schmiedestücke liegt hier dasselbe Problem vor wie bei der Verwendung einer einzelnen Aufnahme mit einer Kamera. Auch hier wird das Objekt nur aus einem Blickwinkel betrachtet, was nicht ausreichend ist. Verdeckte Bereiche werden nicht abgebildet, sondern nur Bereiche, welche in beiden Kamerabildern erkennbar sind.

Im Vergleich zum Stereosehen ist ein ToF-Sensor (3.3), welcher mit reflektiertem Licht arbeitet, entfernungsunempfindlicher (Abbildung 49). Anders als bei Stereo Vision arbeitet eine ToF-Kamera auch bei Oberflächen mit geringer Intensität und Farbvariation zuverlässig.

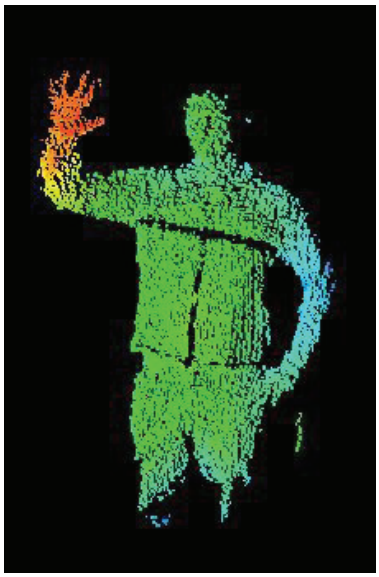

Abbildung 49: Aufnahme einer ToF-Kamera [Krone, 2010]

Die Tiefengenauigkeit liegt bei diesem Verfahren vor dem Stereosehen. Aufgrund des Einsatzes der Lichtgeschwindigkeit wird ein sehr hoher Anspruch an die genaue Zeitmessung gestellt. Als Alternative kann auch eine Phasenmessung erfolgen, wobei mit einer kontinuierlichen Sinusmodulation gearbeitet wird und der Abstand über den Phasenversatz ermittelt wird. Durch das kontinuierliche Wiederholen der Sinusmodulation kommt es jedoch zu einem eingeschränkten Eindeutigkeitsbereich, was sich negativ auf die Qualität der gewonnen 3D-Informationen auswirkt.

Bezogen auf die Anwendung bei industriellen Schmiedestücken ist die Frage, wie gut oder schlecht die ausgesandten Lichtwellen vom Objekt reflektiert werden. Außerdem besteht hier erneut das Problem der Betrachtung aus nur einem Blickwinkel.

Bei Structured Light (3.5), werden bekannte Muster auf das Objekt projiziert und die Musterverzerrungen untersucht (Abbildung 50 und Abbildung 51).

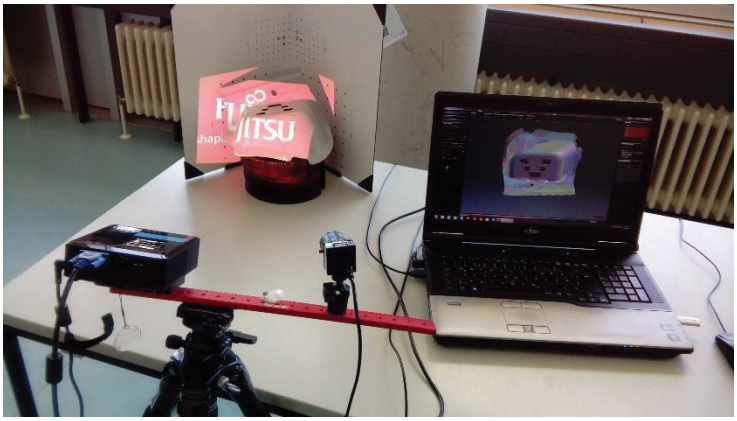

Abbildung 50: Messaufbau zur Verwendung von strukturierter Beleuchtung

Abbildung 51: Darstellung des Streifenlichtmusters auf dem zu messenden Objekt

Auch hier ist das reflektierte Muster empfindlich gegenüber optischen Interferenzen. Hinzu kommt, dass neben einer Kamera auch ein Projektor zur Projektion der Muster erforderlich ist. Da mehrere Muster aus mehreren Blickwinkeln von verschiedenen Standpunkten aufgenommen werden müssen, ist auch dieses Verfahren in Anbetracht des Zeitfaktors für diese Anwendung ungeeignet.

Die folgende Tabelle zeigt einen Vergleich der Verfahren Stereo Vision, ToF und Structured Light.

	Stereo Vision	**ToF**	**Structured Light**
Arbeitsbereich	≤ 2m	0,4 – 5m	0,2m – 3m
Genauigkeit	5% - 10% der Distanz	≤ 0,5% der Distanz	≤ 1mm
Bildauflösung	Hohe Auflösung möglich (abhängig von Kamera)	Gering (320x240 Pixel)	Hohe Auflösung möglich (abhängig von Kamera)
Einsatzbereich	Umgebung mit Feature Points	Indoor und Outdoor	Indoor
Homogene Flächen	Schlecht	Gut	Gut
Software-Verarbeitungs-anforderung	Hoch	Gering	Mittel
Automatische Beleuchtung	Nein	Ja	Ja
Performance bei schwachem Licht	Schlecht	Gut	Gut
Performance bei starkem Licht	Gut	Schlecht	Gut

Tabelle 1: Vergleich der Verfahren Stereo Vision, ToF und Structured Light (Revopoint 3D, 2019)

Durch Tabelle 1 wird deutlich, dass jedes Verfahren seine Vor- und Nachteile hat. Angefangen beim Arbeitsbereich zeigt sich, dass Stereo Vision mit 2m den geringsten Arbeitsbereich bietet. Den größten Arbeitsbereich liefert das ToF-Verfahren mit 5m. Structured Light ordnet sich mit 3m zwischen den beiden Verfahren ein. Auch in Bezug auf die Genauigkeit liefert Stereo Vision das schlechteste Ergebnis. Bei Stereo Vision ist der Tiefenauflösungsfehler eine qudratische Funktion der Entfernung. Im Vergleich dazu ist ein ToF-Sensor, welcher mit reflektiertem Licht arbeitet entfernungsunempfindlicher. Die höchste Genauigkeit dieser drei Verfahren liefert Sturctured Light. Die Bildauflösung ist bei einer ToF-Kamera mit ca. 320x240 Pixeln gering. Bei den anderen Methoden ist eine hohe Auflösung möglich. Hier ist es entscheidend, was für eine Kamera bei diesen Verfahren eingesetzt wird. Unterschiede werden auch beim Einsatzbereich deutlich. Stereo Vision ist auf Schlüsselpunkte angewiesen. Daher kann dieses Verfahren nur in einer Umgebung eingesetzt werden, welche eine ausreichend gute Merkmalsextraktion sowie den darauffolgenden Abgleich erlaubt. ToF ist sowohl für Indoor- als auch Outdoor-Anwendungen verwendbar. Structured Light hingegen nur für Indoor-Anwendungen. Der Grund dafür ist der Einsatz von projizierten Mustern. Durch eine zu hohe Sonneneinstrahlung im Outdoor-Bereich ist die Kamera nicht mehr in der Lage die Veränderung des projizierten Musters einzufangen. Dafür bietet dieses Verfahren eine gute Performance bei schlechten Lichtverhältnissen, da durch den eingesetzten Projektor eine eigene Lichtquelle gestellt wird. Aufgrund der nötigen Schlüsselpunkte bei Stereo Vision ist dieses Verfahren bei schlechten Lichtverhältnissen und homogenen Flächen ungeeignet. Sehr unterschiedlich ist auch der Anspruch an die Softwareverarbeitung. Die Lösung des Korrespondenzproblems bei Stereo Vision erfordert komplexe, rechenintensive Algorithmen zur Merkmalsextraktion und zum Abgleich. Somit ist die Anforderung an die Softwareverarbeitung hoch. Im Vergleich dazu ist die Anforderung bei ToF gering.

Eine bessere Qualität der Tiefeninformation als das Stereosehen und die Verwendung einer ToF- Kamera bietet die Ableitung der Geometrie über eine Laserlinie. Eine Laserlinie kommt beim sogenannten Lichtschnittverfahren zum Einsatz (3.4), wobei anhand der Krümmung der Laserlinie auf dem Objekt 3D Informationen abgeleitet werden können (Abbildung 52).

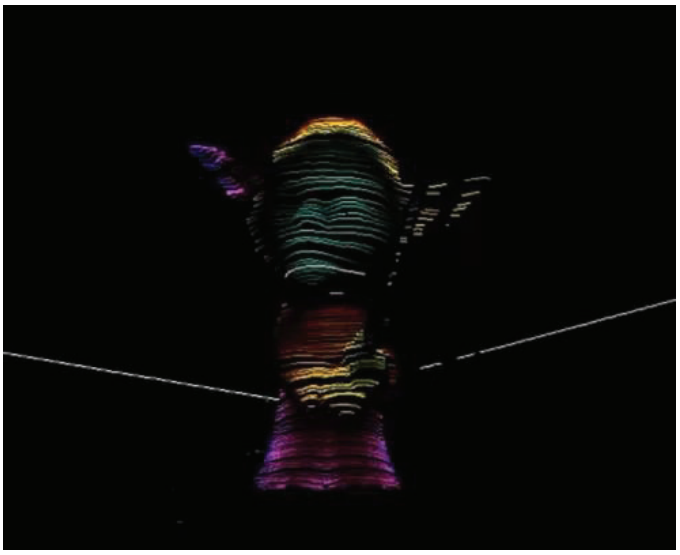

Abbildung 52: Ableitung von 3D-Informationen durch das Lichtschnittverfahren

Da die Laserlinie im Kamerabild eindeutig erkennbar sein muss, würde eine glühende Oberfläche des Schmiedestücks Probleme bereiten. Außerdem ist es nötig das gesamte Schmiedestück aus mehreren Blickwinkeln mehrfach mit der Laserlinie abzufahren. Deshalb ist dieses Verfahren für kleinere Messobjekte geeignet und nicht für großdimensionale Schmiedestücke.

Das letzte der vorgestellten Verfahren ist Structure from Motion. Bei diesem Verfahren werden mit Hilfe von 2D-Bildinformationen Rückschlüsse auf die Geometrie von 3D-Szenen geschlossen (Abbildung 53). Dabei werden mehrere Bilder um das Messobjekt herum aufgenommen. Somit ist eine Betrachtung des Schmiedestücks aus mehreren Blickwinkeln möglich. Um unnötigen Zeitverlust durch mehrere Standpunkte zu vermeiden können mehrere Kameras um das Schmiedestück herum verteilt werden. Somit ist es möglich, alle Kameras synchron auszulösen und Aufnahmen aus mehreren Blickwinkeln in kürzester Zeit zu erhalten.

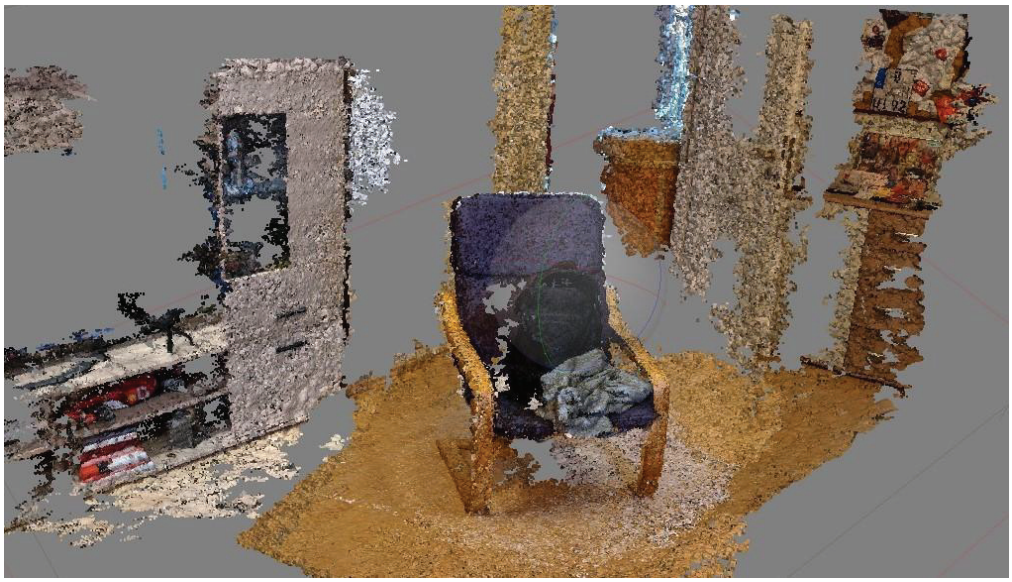

Abbildung 53: Ableitung von 3D-Informationen durch den Einsatz von Structure from Motion

Bei diesem Verfahren kommen jedoch auch komplexe Berechnungen zum Einsatz, was eine hohe Anforderung an die Softwareverarbeitung mit sich bringt. Daher gilt es im Folgenden zu untersuchen, ob auch unter Beachtung des zeitlichen Aspektes eine praktische Anwendung auf industrielle Schmiedestücke möglich ist. Bezogen auf Tabelle 1, richtet sich der Grad der Genauigkeit dieses Verfahrens nach der verwendeten Kamera und nach der Genauigkeit der bestimmten Referenzpunkte.

6.2 Realisierter Lösungsansatz

Da es im Rahmen dieser Arbeit nicht möglich ist die Anwendung unter realen Bedingungen mit industriellen Schmiedestücken zu testen, wird mit einem simulierten Bauteil gearbeitet. Simuliert wird das Bauteil durch einen Schwamm (Abbildung 54). Dieser ermöglicht es, durch einfaches Trimmen und Beschneiden, verschiedene Stadien des Bauteils zu simulieren.

Abbildung 54: Schwamm als simuliertes Schmiedestück

Begonnen wird mit Agisoft Metashape. Agisoft Metashape ist ein eigenständiges Softwareprodukt, das die photogrammetrische Verarbeitung digitaler Bilder durchführt und 3D Raumdaten erzeugt, die in GIS-Anwendungen, der Dokumentation des Kulturerbes und für indirekte Messungen von Objekten verschiedener Maßstäbe verwendet werden.

Für den ersten Durchlauf werden 27 Aufnahmen des simulierten Bauteils verwendet. Die Aufnahme des Bauteils wird durch die Verwendung eines Google Pixel 4 Smartphones realisiert. Diese Aufnahmen werden mithilfe der Software ausgerichtet. Das Ergebnis dieses Arbeitsschrittes ist eine sparse point cloud mit den dazugehörigen Kamerastandorten (Abbildung 55). Die Berechnung erfolgt unter Verwendung der Scale Invariant Feature Transform, dem sogenannten SIFT-Algorithmus.

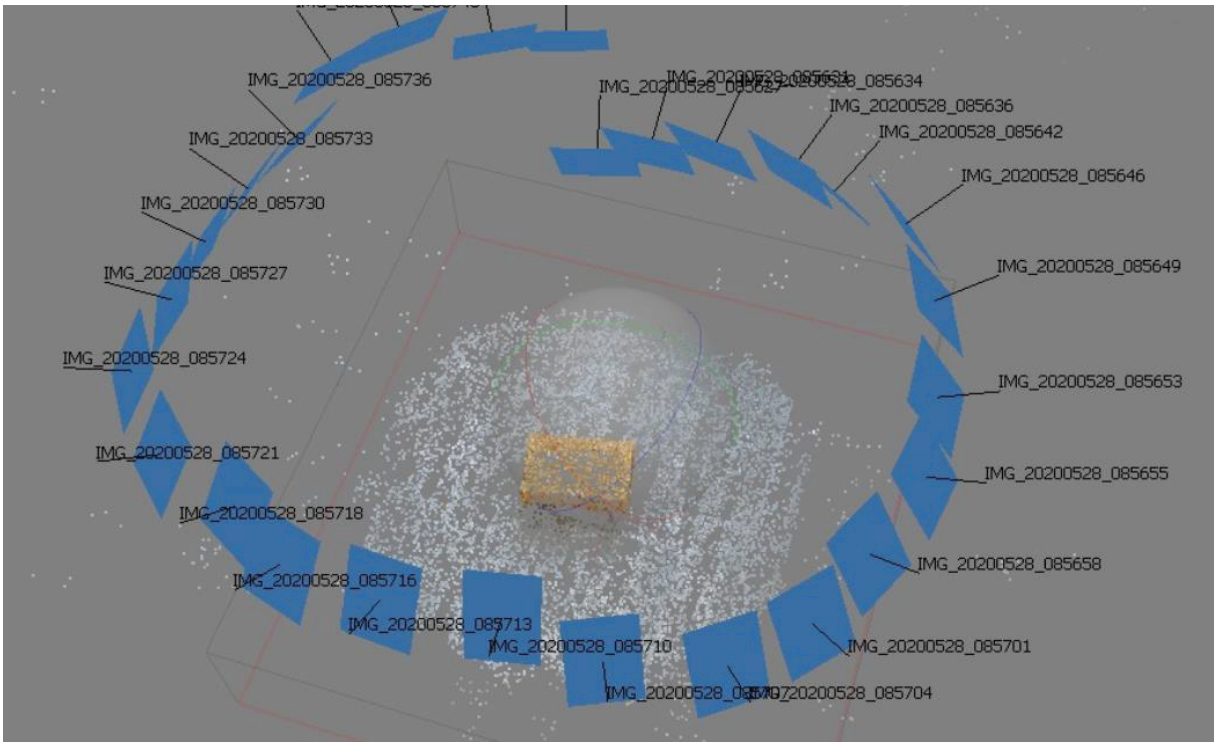

Abbildung 55: Darstellung der sparse point cloud, berechnet aus 27 Aufnahmen

Da der Fokus bei dieser Anwendung auf der Zeit liegt wird für jeden Arbeitsschritt eine Zeitmessung durchgeführt. Dieser Arbeitsschritt nimmt eine Zeit von 107 Sekunden, unter Verwendung der Standardeinstellungen, in Anspruch. Da die zeitliche Begrenzung bei drei Sekunden liegt ist dieses Ergebnis nicht akzeptabel. Die Software bietet verschiedene Möglichkeiten die Parameter der Berechnungen anzupassen. Variierbar sind zum Beispiel die Paramater Key Point Limit und Tie Point Limit. Die Anpassung dieser Paramater führt zu folgenden Ergebnissen:

Key Point Limit	Tie Point Limit	Zeit [s]
40000	2000	107,119
20000	2000	52,969
20000	1000	48,237
10000	2000	34,154
10000	1000	32,648

Tabelle 2: Zeitliche Analyse der Berechnung der sparse point cloud unter Veränderung der Parameter

Wie aus Tabelle 2 hervorgeht, ist es möglich durch Anpassung dieser Parameter die Berechnungszeit deutlich zu reduzieren. Die Berechnungszeit unter Verwendung der Standardeinstellung wird durch die Herabsetzung des Key Point und Tie Point Limits um den Faktor 3 reduziert. 32 Sekunden als Ergebnis sind jedoch weiterhin nicht ausreichend. Somit muss die Berechnungszeit noch weiter reduziert werden.

Das Verfahren Structure from Motion soll auch in Form eines eigenen Porgrammieransatzes, unter Verwendung von QT5 in Verbindung mit OpenCV, umgesetzt werden. Im Vergleich zur Verwendung der Software Agisoft Metashape, ist für den eigenen Programmieransatz zunächst eine Kamerakalibrierung notwendig. Dazu wird die unter Punkt 5 erläuterte Theorie nun in der Praxis angewandt.

6.2.1 Kamerakalibrierung

Jede Kamera bringt eine Verzerrung mit sich. Da diese Verzerrung konstant ist, kann diese mithilfe einer Kamerakalibrierung korrigiert werden. Darüber hinaus ist es möglich, mit einer Kalibrierung auch das Verhältnis zwischen den natürlichen Einheiten der Kamera (Pixel) und den Einheiten der realen Welt (z. B. Millimeter) zu bestimmen. [opencv dev team, 2014]

Umgesetzt wird die Kamerakalibrierung in einem eigenen Programmieransatz. Verwendet wird dabei Q5 in Verbindung mit OpenCV. Kalibriert wird die bereits erwähnte Smartphone-Kamera eines Google Pixel 4. Kalibriert wird die Kamera mit einem 9x6 Schachbrettmuster (Abbildung 56). Dieses Muster wird mehrmals aus verschiedenen Positionen und Richtungen aufgenommen und mithilfe der Bildverarbeitungsroutinen von OpenCV detektiert. Jedes gefundene Muster führt zu einer neuen Gleichung. Um diese Gleichungen zu lösen, ist eine gewisse Mindestanzahl von Aufnahmen notwendig, um ein gut positioniertes Gleichungssystem zu bilden. In der Theorie sind zwei Aufnahmen ausreichend. In der Praxis ist jedoch Rauschen in den Eingabebildern vorhanden, so dass für ein gutes Ergebnis mehr Aufnahmen notwendig sind. Wie viele Aufnahmen konkret für ein gutes Ergebnis erforderlich sind, wird im weiteren Verlauf der Arbeit untersucht.

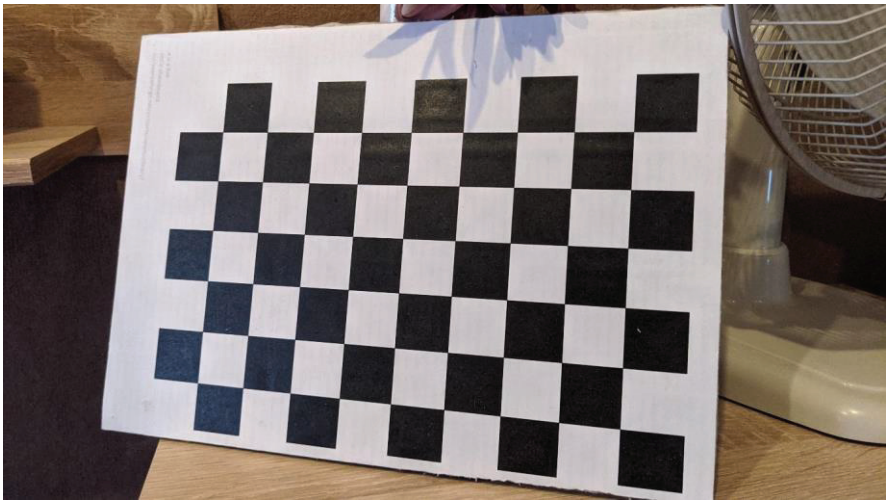

Abbildung 56: Darstellung der verwendeten Kalibriertafel

Über die digitale Bildverarbeitung werden die Ecken der Quadrate des Kalibriermusters detektiert. Die Dimension der Quadrate ist für die Kamerakalibrierung erforderlich. In diesem

Fall haben die Quadrate eine Größe von 2.55 x 2.55 cm. Hat der Algorithmus die Ecken der Quadrate erfolgreich detektiert, so werden diese farblich markiert (Abbildung 57). Die Pixelkoordinaten der Ecken werden gespeichert.

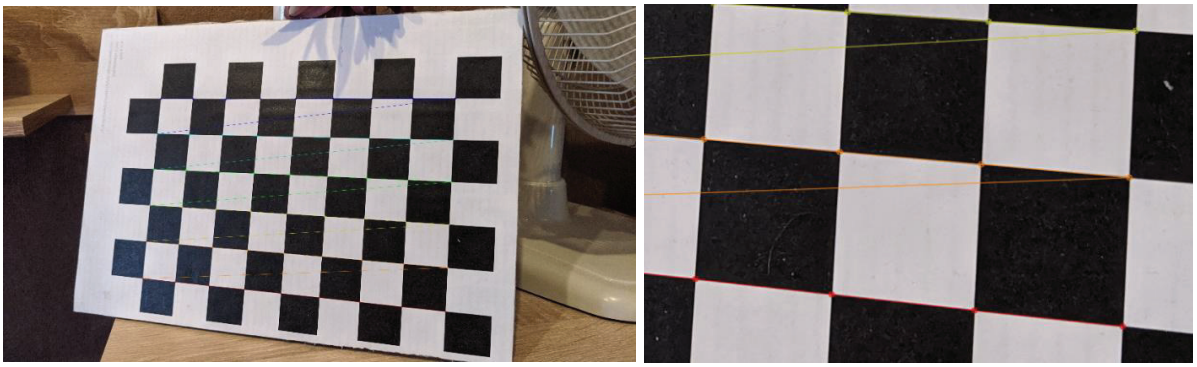

Abbildung 57: Darstellung der farblich markierten Ecken mit Detailansicht

Das Kalibriermuster wird dabei mehrmals von verschiedenen Positionen aus aufgenommen. Ebenso variiert wird dabei die Lage und Rotation des Kalibiermusters (Abbildung 58).

Abbildung 58: Darstellung der Kalibriertafel aus verschiedenen Richtungen mit unterschiedlichen Rotationen

Das Hauptaugenmerk bei der Kalibrierung liegt auf der Ermittlung der Kamerakonstante, sowie auf der Position des Bildmittelpunkts in x- und y-Richtung. Die Kalibrierung erfolgt mit unterschiedlichen Bildbänden, welche sich in ihrer Bilderanzahl unterscheiden. Die

Verwendung unterschiedlich vieler Bilder soll Aufschluss über die benötigte Anzahl von Bildern liefern, welche für eine gute Kalibrierung notwendig ist. Der Vergleich führt zu folgendem Ergebnis:

Anzahl Bilder	Kamerakonstante	Pixelposition X	Pixelposition Y
10	5924,54	4,04032	-27,1982
20	6416,16	9,134	-9,44258
30	7183,78	5,08618	-8,8176
40	8625,57	46,1894	238,89
50	3990,06	1044,98	725,975
55	3805,26	1033,77	716,324
75	3872,41	1190,75	1113,56

Tabelle 3: Analyse der Parameter der Kamerakalibrierung mit unterschiedlichen Bildbänden

Abbildung 59 bis 61 zeigt den grafischen Verlauf der Kalibrierparameter.

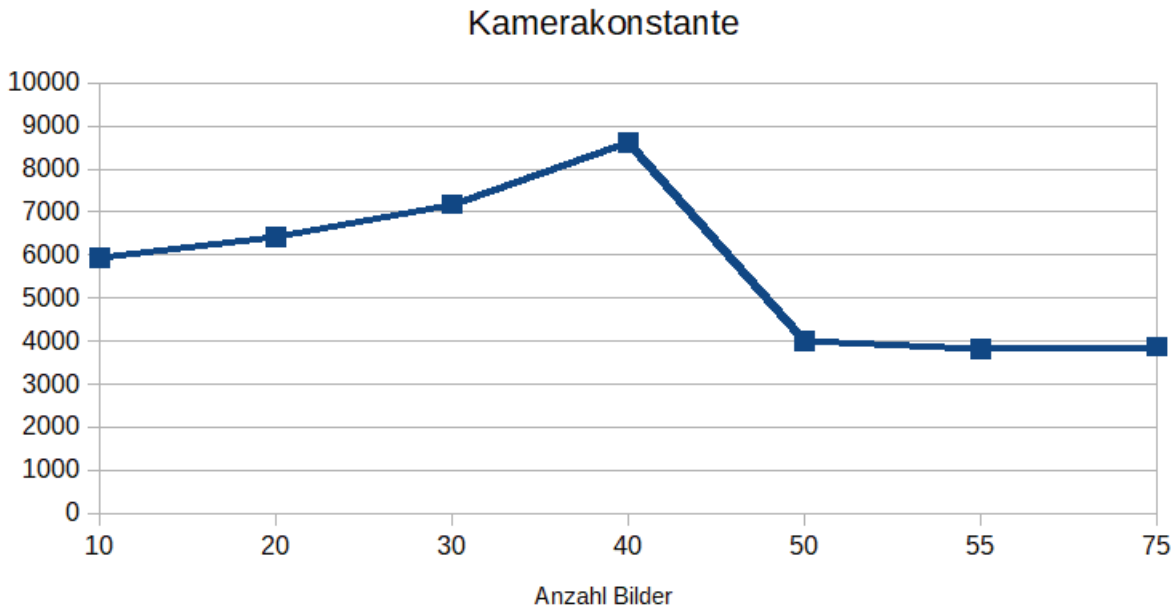

Abbildung 59: Verlauf der Kamerakonstante [Pixel]

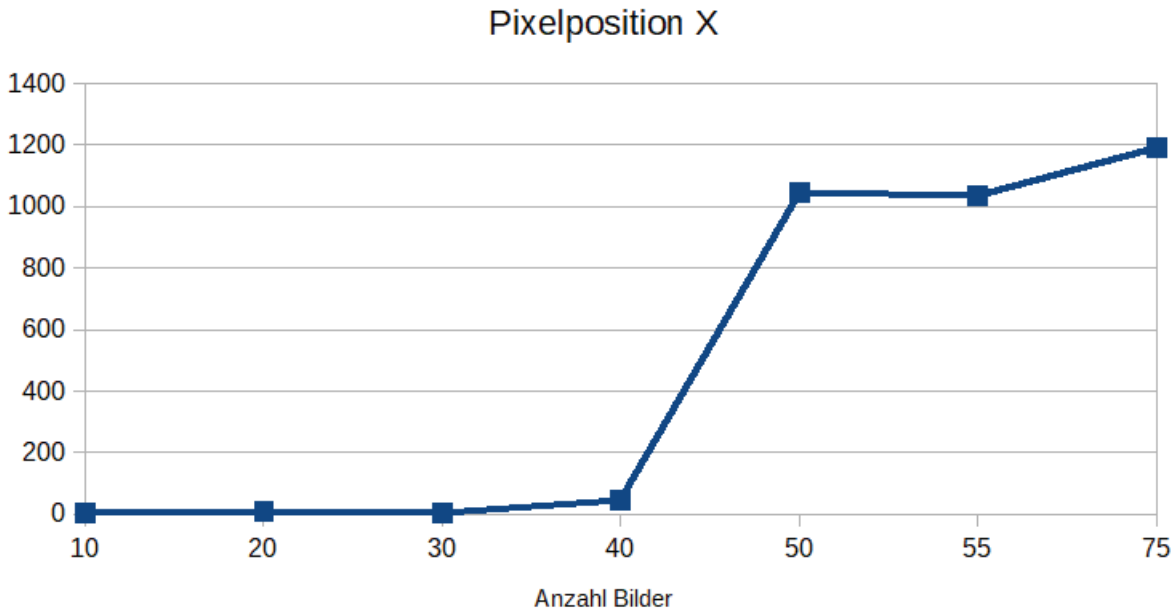

Abbildung 60: Verlauf des Bildmittelpunktes in x-Richtung [Pixel]

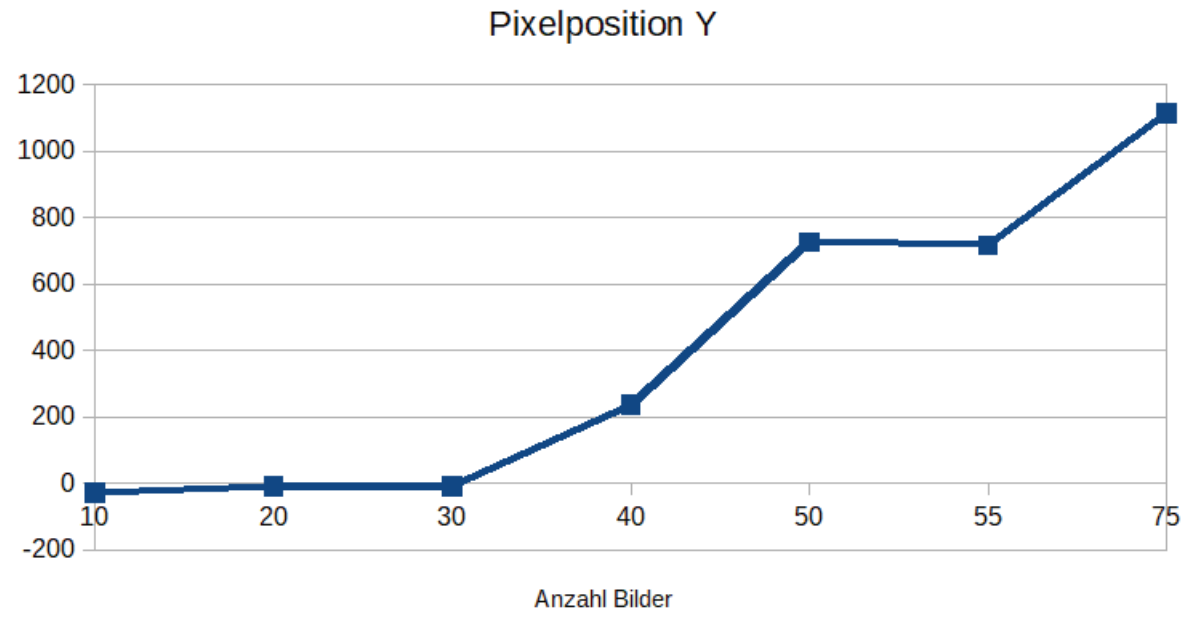

Abbildung 61: Verlauf des Bildmittelpunktes in y-Richtung [Pixel]

Der Vergleich zeigt, dass sich Theorie und Praxis sehr unterscheiden. Bei geringer Bildanzahl zwischen 10 und 50 Bildern variieren die berechneten Parameter sehr stark. Erst ab einer Bildanzahl von 55 Bildern pendeln sich die Parameter langsam ein und bleiben konstant.

Die zeitliche Komponente ist ebenfalls Bestandteil des Vergleichs. Das Ergebnis zeigt, dass die benötigte Zeit für die Kalibrierungen mit unbrauchbaren Ergebnissen ein Vielfaches geringer ist als zu den Kalibrierungen mit guten Werten. Somit dauert die Kamerakalibrierung

mit 75 Bildern bis zu einer Stunde. Hier hat jedoch die Auflösung der Kamera einen signifikanten Einfluss auf die Berechnungszeit.

Anzahl Bilder	Zeit [s]
10	3,0
20	6,4
30	11,7
40	18,5
50	55,0
55	3198,7
75	3519,2

Tabelle 4: zeitliche Analyse der Kamerakalibrierung mit unterschiedlichen Bildbänden

Für die Anwendung auf industrielle Schmiedestücke fällt die hohe Berechnungszeit der Kamerakalibrierung nicht negativ ins Gewicht. Die Kamerakalibrierung muss nur einmal zu Beginn durchgeführt werden. Anschließend kann die berechnete Kalibrierung immer wieder abgerufen und verwendet werden.

Die Kamerakalibrierung mit 75 Bildern liefert als Ergebnis eine Kamerakonstante von 3872,41 Pixeln und einen Bildmittelpunkt mit einer x-Koordinate von 1190,75 Pixeln und einer y-Koordinate von 1113,56 Pixeln. Neben diesen 3 Parametern liefert die Kalibrierung der Kamera weitere Parameter:

Radialer Verzerrungskoeffizient k1	0,0178076
Radialer Verzerrungskoeffizient k2	0,138616
Radialer Verzerrungskoeffizient k3	-0,138721
tangentialer Verzerrungskoeffizient p1	-0,0217862
tangentialer Verzerrungskoeffizient p1	-0,0494995

Tabelle 5: Übersicht zusätzlicher Kalibrierparamater

Mithilfe der berechneten Parameter aus der Kamerakalibrierung kann der eigene Programmieransatz mit Structure from Motion erfolgen.

Der erste Ansatz ist auch hier mit dem SIFT-Algorithmus, zur Bestimmung der Schlüsselpunkte, zu arbeiten. Unter Verwendung derselben Bildreihe mit 27 Bildern, wie bereits mit der Software Metashape, liefert das Ergebnis eine Berechnungszeit, welche um den Faktor 1,7 höher ist als bei der Software der Firma Agisoft. Damit beträgt die Berechnungszeit beim eigenen Programmieransatz 180 Sekunden.

Zur Reduzierung der Rechenzeit wird auf andere Algorithmen zur Berechnung der Schlüsselpunkte zurückgegriffen. Statt des SIFT-Algorithmus wird der ORB-Algorithmus verwendet. Dieser ist schneller als SIFT jedoch nicht so robust. Aufgrund der geringeren Robustheit ist der erste Versuch mit demselben Datenset fehlgeschlagen (Abbildung 62).

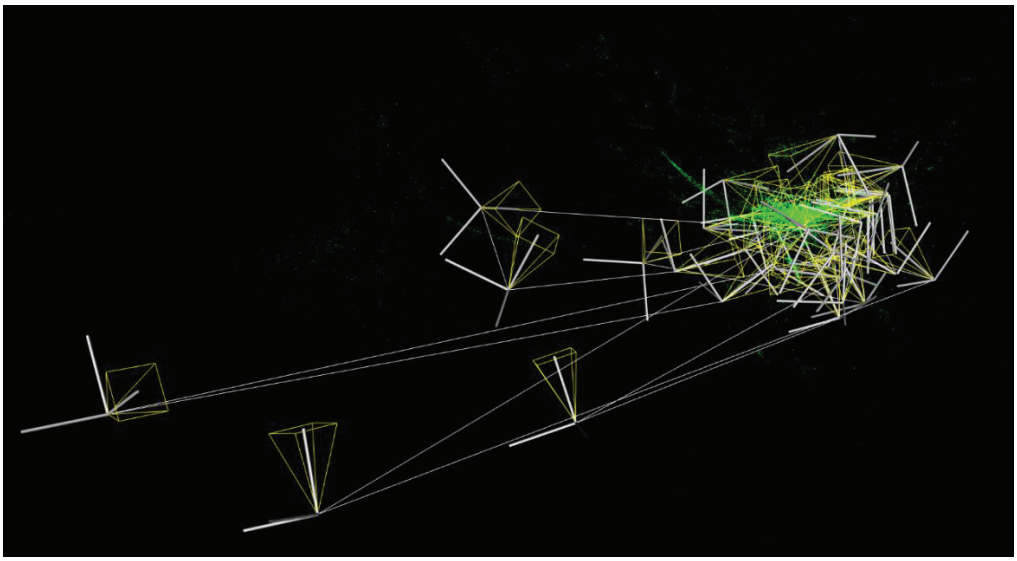

Abbildung 62: Ergebnis des eigenen Programmieransatzes unter Verwendung des ORB-Algorithmus mit 27 Aufnahmen

Der ORB-Algorithmus kann die einzelnen Schlüsselpunkte des Schwamms nicht eindeutig zuordnen. Dies führt zu einer falschen Berechnung der Kamerapositionen und der falschen Verknüpfung der Bilder untereinander.

Um den eigenen Programmieransatz dennoch mit Metashape zeitlich vergleichen zu können wird die Bildanzahl auf 5 Bilder reduziert. Die Berechnung mit 5 Bildern und dem eigenen Programmieransatz führt zu dem in Abbildung 63 dargestellten Ergebnis.

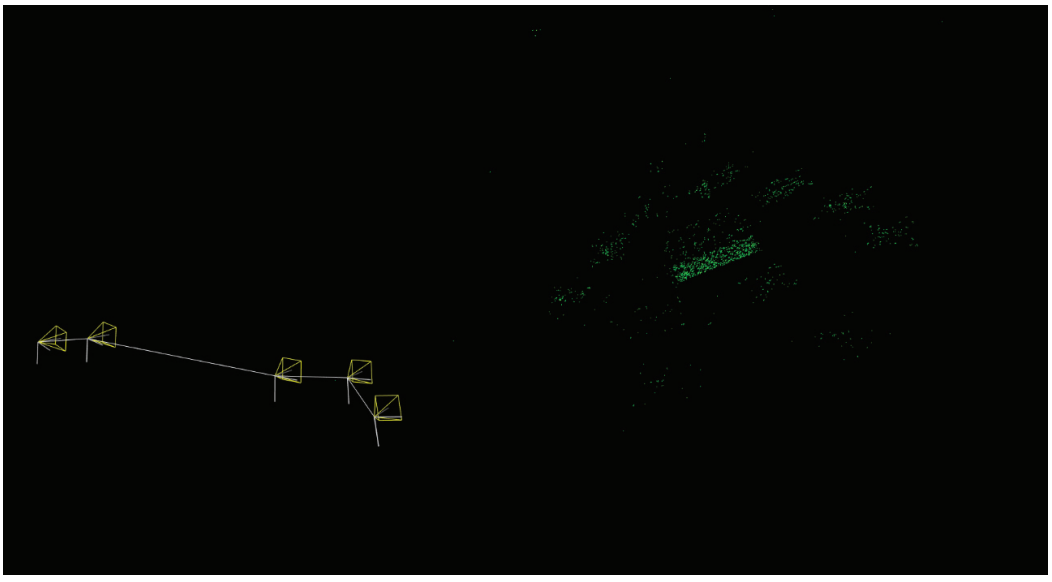

Abbildung 63: Ergebnis des eigenen Programmieransatzes unter Verwendung des ORB-Algorithmus mit 5 Aufnahmen

Das Ergebnis der Berechnung sind die korrekten Kamerapositionen sowie eine Punktwolke mit 3D-Koordinaten des theoretischen Schmiedestücks, basierend auf 5 Bildern. Die Berechnungszeit liegt dabei bei 13,45 Sekunden.

Eine weitere Berechnung auf Grundlage derselben 5 Bilder wird mit der Software Metashape durchgeführt (Abbildung 64).

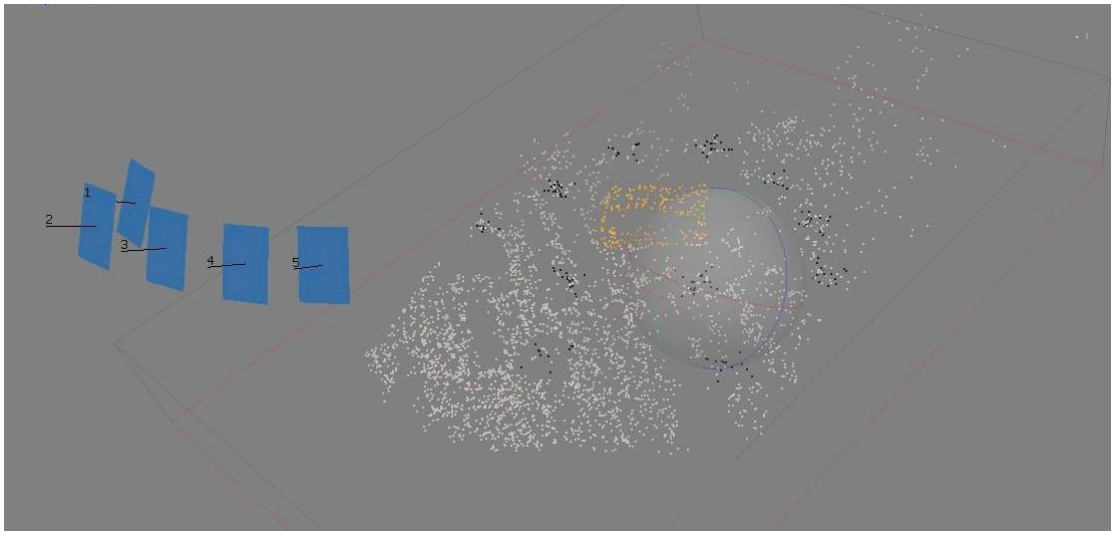

Abbildung 64: Ergebnis des Programms Metashape mit 5 Aufnahmen

Auch hier ist das Ergebnis eine Berechnung der korrekten Kamerapositionen sowie eine Punktwolke des simulierten Schmiedestücks. Im direkten Vergleich ergeben sich folgende Unterschiede:

	Eigener Ansatz	**Metashape**
Detektor	ORB	SIFT
Punktfarbe	Nein	Ja
Berechnete Punkte	2688	4012
Berechnungszeit	13,65	8,03

Tabelle 6: Gegenüberstellung der Ergebnisse zur Berechnung einer sparse point cloud

Trotz der Verwendung des SIFT Algorithmus, welcher langsamer arbeitet als der ORB-Algorithmus beim eigenen Programmieransatz, ist die Berechnung mit Metashape schneller. Verwendet wurden dabei die Standardeinstellungen. Mit 8,03 Sekunden ist die Software um den Faktor 1,7 schneller als der eigene Programmieransatz. Trotz geringerer Auflösung bei Metashape ist dort die Anzahl der berechneten Punkte höher. Die Anzahl der berechneten Punkte beträgt bei Metashape 4012. Bei dem eigenen Programmieransatz sind es 2688 Punkte. Des Weiteren wird bei der Software Metashape die Punktfarbe mit berechnet. Diese ist jedoch für die Anwendung auf industrielle Schmiedestücke uninteressant.

Der Versuch, die Auflösung der Eingangsbilder auch bei dem eigenen Programmieransatz um die Hälfte zu reduzieren, ist fehlgeschlagen. Durch die geringere Auflösung sind weniger Details in den Bildern vorhanden. Dies hat zur Folge das der ORB-Algorithmus aufgrund seiner geringeren Robustheit nicht ausreichend Schlüsselpunkte berechnen kann, um eine korrekte Verknüpfung der Bilder zu ermöglichen.

Aufgrund der besseren Robustheit und des effizienteren Algorithmus der Software Metashape im Vergleich zum eigenen Programmieransatz, wird im weiteren Verlauf der Arbeit mit der Software der Firma Agisoft weitergearbeitet. Der Versuch, die Rechenzeit erneut durch Anpassung des Key Point und Tie Point Limits zu senken, führt zu folgendem Ergebnis:

Key Point Limit	Tie Point Limit	Zeit [s]
20000	2000	5,12
20000	1000	4,89
10000	2000	3,84
10000	1000	3,79

Tabelle 7: zeitliche Analyse der Anpassung des Key Point und Tie Point Limits

Aus der Tabelle geht hervor, dass die Rechenzeit auf 3,8 Sekunden reduziert werden kann. Die Anzahl der berechneten Punkte wird dadurch jedoch stark reduziert. Außerdem ist wie in Abbildung 63 zu sehen, eine Vielzahl von Punkten vorhanden, welche das Umfeld des simulierten Schmiedestücks abbilden. Diese sind uninteressant und beanspruchen wichtige Rechenzeit. Ein Ansatz, diese unnötigen Punkte herauszufiltern, ist die sogenannte Bildvorverarbeitung.

6.2.2 Bildvorverarbeitung

Da der Zeitfaktor bei der Anwendung auf industrielle Schmiedestücke eine entscheidende Rolle spielt, gilt es zu untersuchen, durch welche Verfahren die Dauer der Berechnung von 3D-Informationen verkürzt werden kann. Eine Möglichkeit dies zu erreichen, ist die Segmentierung des Schmiedestücks. Durch eine schnelle Segmentierung im Fertigungsprozess wird die Berechnung von 3D-Modellen mit Hilfe von photogrammetrischen Methoden beschleunigt. Somit entfallen für die Berechnung und für den Anwender irrelevante Bildinformationen. Diese Segmentierung wird ebenfalls durch den Einsatz von Computer Vision realisiert.

6.2.2.1 Farbfilterung

Zur Separierung des Bauteils wird mit einer Farbfilterung des Hintergrundes gearbeitet. Die Farbfilterung wird nicht im RGB-Farbraum durchgeführt, sondern im HSV-Farbraum. Die

Vorteile des HSV-Farbraums gegenüber dem RGB-Farbraum werden in den folgenden Zeilen erläutert.

RGB wird definiert durch Auflistung der Rot-, Grün-, und Blau-Anteile eines einzelnen Farbwertes. Es wird eine additive Methode verwendet, bei der jede Farbe umso heller wird, je mehr von jeder Farbe hinzugefügt wird (Abbildung 65). Somit ist es schwierig eine Farbe mit den jeweiligen Anteilen der Primärfarben augenscheinlich zu beschreiben.

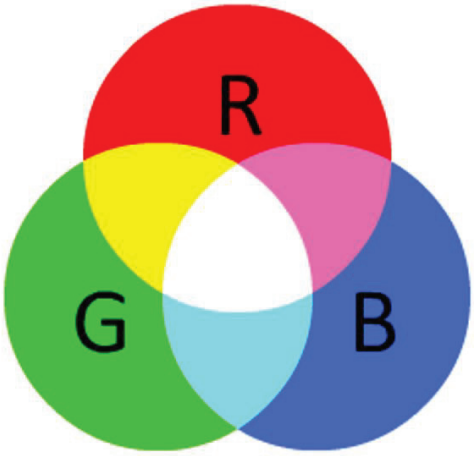

Abbildung 65: RGB-Farbraum [handmap, 2016]

Im HSV-Farbraum wird nicht mehr nach den Rot-, Grün- und Blau-Intensitäten unterschieden, sondern nach Farbwert (Hue), Farbsättigung (Saturation) und Hellwert (Value) (Abbildung 66).

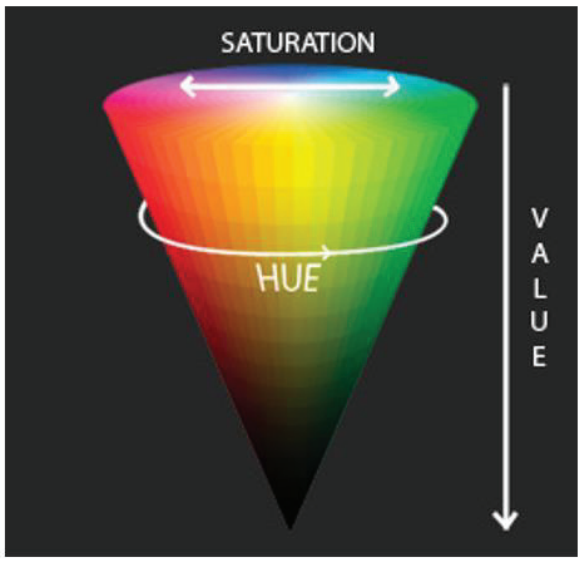

Abbildung 66: HSV-Farbraum [handmap, 2016]

Der HSV-Farbraum ähnelt der menschlichen Farbwahrnehmung. Somit ist es möglich den gewünschten Farbton zu wählen und anschließend zu entscheiden wie hell oder dunkel und wie gesättigt dieser Farbton sein soll [handmap, 2016].

Zur Analyse der konkreten Farbwerte und zur Filterung des Hintergrundes ist im Rahmen dieser Arbeit ein Tool entstanden, welches mit OpenCV und QT5 erstellt wurde. Dieses Tool ermöglicht es, über Schieberegler einen bestimmten Farbbereich aus dem Bild der Kamera herauszufiltern (Abbildung 67).

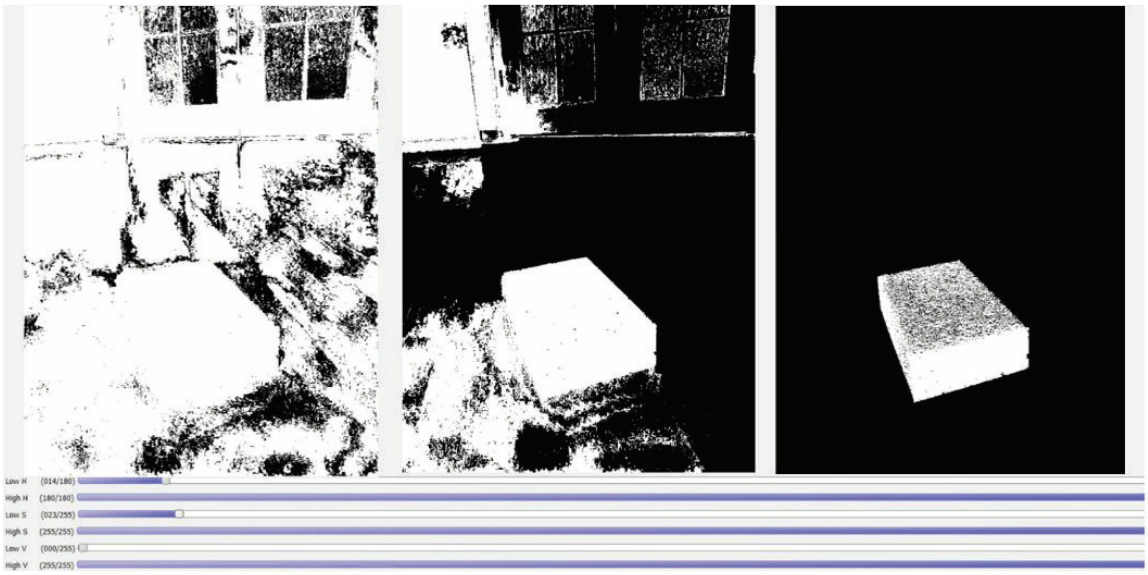

Abbildung 67: Tool zur Filterung des HSV-Farbraums mit unterschiedlichen Filterungsstufen

Mit Hilfe des HSV-Farbkreises ist es möglich, sich einen Überblick über die nötigen Farbfilter zu verschaffen (Abbildung 68). Dadurch ist es möglich, im Vorhinein abzuschätzen, in welchem Farbbereich sich das zu untersuchende Schmiedestück befindet.

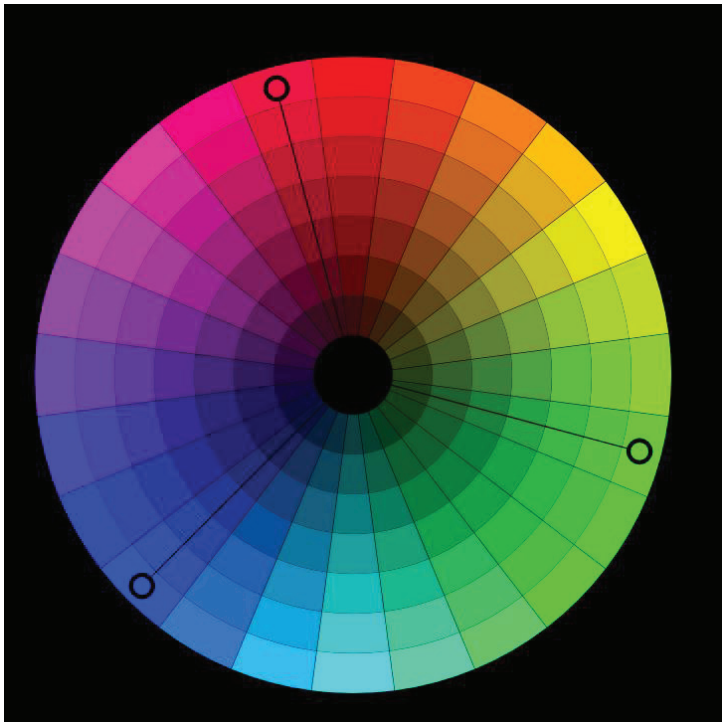

Abbildung 68: HSV-Farbkreis [CleanPNG, 2016]

6.2.2.2 Maskierung

Das Ergebnis der Filterung ist eine Maske für jedes Bild. Die Maske beschreibt den Bereich des Bauteils innerhalb des Originalbildes. Dadurch ist es möglich den Bereich des Bauteils im Originalbild zu markieren und somit vom Hintergrund zu trennen (Abbildung 69). Aufgrund der gleichbleibenden Position der Kamera und des gleichbleibenden Hintergrundes ist die Bestimmung der HSV-Werte nur einmal nötig. Die Werte können für die weiteren Aufnahmen und somit für die weiteren Arbeitsschritte im Fertigungsprozess beibehalten werden.

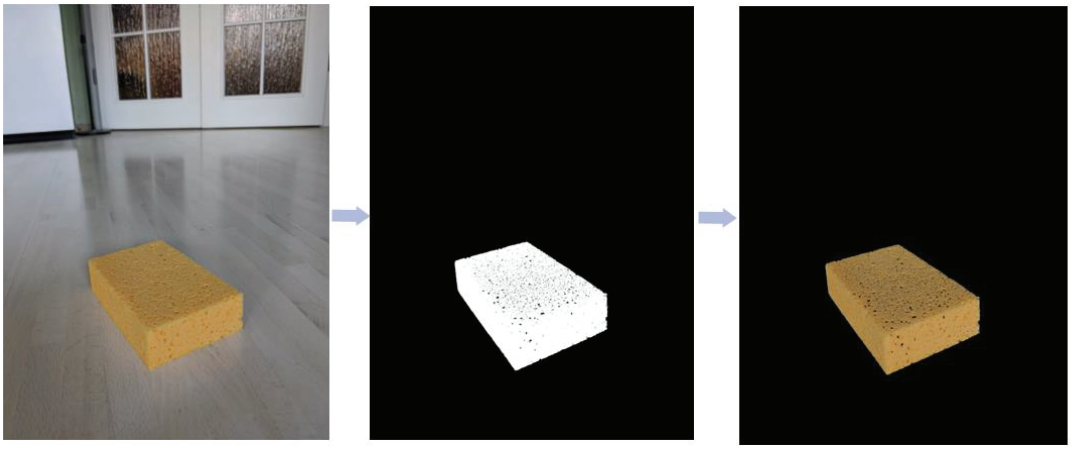

Abbildung 69: Ablauf der Separierung des Bauteils vom Hintergrund

Da das Bauteil Bereiche aufweist, welche die Farbe des zuvor herausgefilterten Hintergrundes beinhalten, sind in der Maske Lücken vorhanden. Diese Lücken müssen geschlossen werden, da das Bauteil durch die Lücken nicht vollständig abgebildet wird. Diese Lücken werden mithilfe der Dilate-Funktion geschlossen (Abbildung 70).

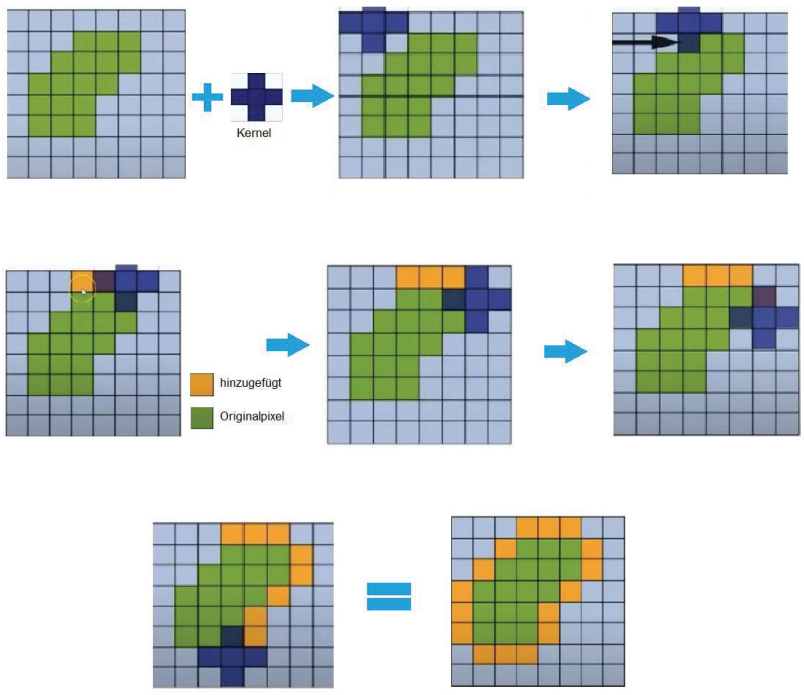

Abbildung 70: Funktionsweise der Dilate-Funktion [Kshirsagar, 2017]

Bei dieser Funktion wird das Eingabebild (die Maske) durch eine kleine Menge von Koordinatenpunkten, welche ein strukturiertes Element (auch Kernel genannt) bilden, erweitert. Das Kernel bestimmt die genaue Auswirkung der Dilate-Funktion auf das Eingabebild. Um die Dilation des Eingabebildes zu berechnen, werden nacheinander die einzelnen Hintergundpixel betrachtet. Für jedes Hintergrundpixel (Eingabepixel) wird das Kernel so über das Eingabebild gelegt, dass der Ursprung des Kernels mit der Position des Eingabepixels übereinstimmt. Wenn mindestens ein Pixel des Kernels mit einem Vordergrundpixel im darunter liegenden Bild übereinstimmt, dann wird das Eingabepixel auf den Vordergrundwert gesetzt. Sind jedoch alle entsprechenden Bildpunkte im Bildhintergrund, dann wird das Eingabepixel auf dem Hintergrundwert belassen. [Fisher et al., 2003]

Nach Anwendung der Dilate-Funktion auf die lückenhaften Masken sind diese im Anschluss geschlossen. Das Ergebnis ist ein separiertes Bauteil frei von Lücken und ohne Detailverlust (Abbildung 71).

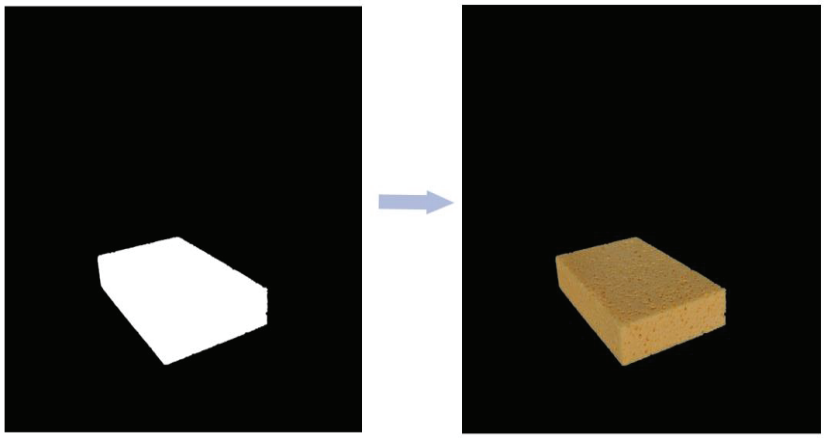

Abbildung 71: Ergebnis nach Anwendung der Dilate-Funktion

Die Anwendung der Dilate-Funktion wirkt sich ebenso auf den Rand des Schmiedestücks aus wie auch auf die Lücken im Inneren des Schmiedestücks. Die Dimensionen des Schmiedestücks verändern sich jedoch nicht, sondern nur der Bereich der Maskierung. Somit wird die Länge und Breite des Schmiedestücks nicht verändert. Es ändert sich nur der markierte Bereich, welcher den Bereich der nicht ausgeblendeten Pixel beschreibt. Eine Anwendung der Erode-Funktion ist somit nicht notwendig.

Nun gilt es zu untersuchen, inwiefern sich die Bearbeitung der Eingabebilder auf die Rechenzeit auswirkt. Durchgeführt wird eine erneute Berechnung, diesmal jedoch mit den gefilterten Fotos. Der erste Versuch erfolgt mit den Standardeinstellungen (Abbildung 72).

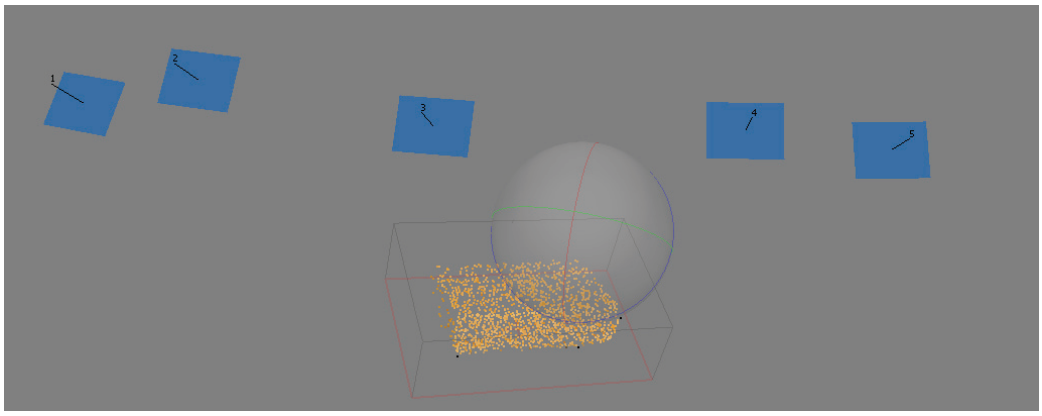

Abbildung 72: Berechnung der sparse point cloud mit gefilterten Aufnahmen

Wie zu erkennen ist, sind nun keine irrelevanten Punkte mehr vorhanden. Dies zeigt sich auch in der Anzahl der berechneten 3D-Punkte. Mit 1555 Punkten hat sich die Anzahl der Punkte um den Faktor 2,6 verringert. Ebenso reduziert wurde die Berechnungszeit. Die Berechnungszeit wurde von 8,03 auf 4,12 Sekunden, unter Verwendung der Standardeinstellungen, reduziert.

Der Versuch, die Berechnungszeit weiter über die Anpassung der Paramater zu reduzieren, führt zu folgendem Ergebnis:

Key Point Limit	Tie Point Limit	Zeit [s]
40000	2000	4,12
20000	2000	4,12
20000	1000	3,98
10000	2000	4,12
10000	1000	3,98
5000	2000	3,81
5000	1000	3,79

Tabelle 8: zeitliche Analyse mit 5 gefilterten Aufnahmen

Da durch die gefilterten Bilder weniger Pixel vorhanden sind, welche von dem SIFT-Algorithmus untersucht werden, hat die Anpassung der Parameter hier nicht so einen großen Einfluss auf die Berechnungszeit wie bei den ungefilterten Fotos. Der Algorithmus findet maximal 2500 Punkte in den Bildern, wodurch sich die Berechnungszeiten sehr ähneln. Auch bei der Festlegung des Limits auf unter 10000 Punkte gibt es keinen signifikanten Unterschied. Somit ist das Einsparpotenzial an dieser Stelle ausgereizt.

Eine weitere Option zur Reduzierung der Rechenzeit ist die weitere Reduzierung der Auflösung der Eingabebilder.

Key Point Limit	Tie Point Limit	Zeit [s]
40000	2000	1,89
20000	2000	1,89
20000	1000	1,84
10000	2000	1,90
10000	1000	1,69

Tabelle 9: zeitliche Analyse mit reduzierter Auflösung der Aufnahmen

Die Reduzierung der Auflösung um den Faktor 2 im Vergleich zu den Standardeinstellungen ermöglicht eine weitere Reduzierung der Berechnungszeit. Die Anzahl der berechneten Punkte beträgt nur noch 232 (Abbildung 73). Durch die geringere Anzahl von Punkten hat die Reduzierung des Key Point und Tie Point Limits keinen signifikanten Einfluss auf die Berechnungszeit.

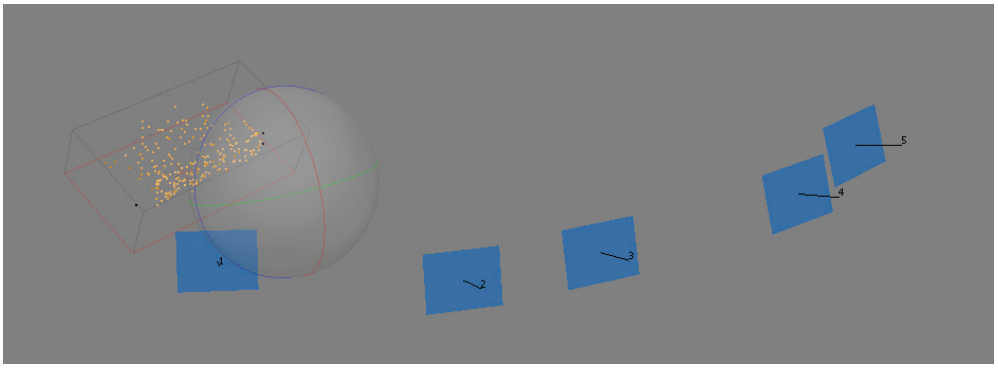

Abbildung 73: Darstellung der sparse point cloud unter Verwendung der Aufnahmen mit reduzierter Auflösung

Mit der erreichten Rechenzeit von 1,9 Sekunden wurde die Grenze von 3 Sekunden unterschritten. Ob die geringe Anzahl der berechneten Punkte jedoch ausreichend für eine aussagekräftige Beurteilung der Geometrie des simulierten Schmiedestücks ist, gilt es zu untersuchen.

Metashape bietet die Möglichkeit die Punktwolke zu verdichten (Abbildung 74). Eine dichtere Punktwolke ist für die spätere Ableitung eines Drahtgittermodells hilfreich.

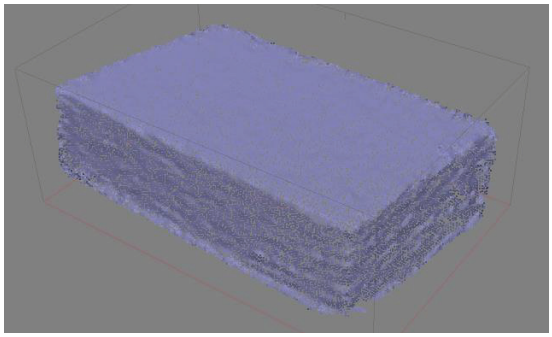

Abbildung 74: Berechnung der verdichteten Punktwolke

Auf geringen Einstellungen werden so 30000 Punkte berechnet. Dieser Prozess nimmt eine Rechenzeit von 0,87 Sekunden in Anspruch. Die in Abbildung 74 berechnete dense cloud weist jedoch durch die Verwendung der geringen Einstellungen Lücken auf. In Summe ergibt sich eine Rechenzeit von 2,77 Sekunden. Als nächstes gilt es zu untersuchen, wie genau die berechnete Punktwolke ist und inwieweit sie mit den Maßen des Originals übereinstimmen. Dazu ist die Einführung eines Maßstabes notwendig.

6.3 Möglichkeiten zur Einführung eines Maßstabes

Die Einführung eines Maßstabes ist notwendig, um die Pixel der Bilder und die daraus berechneten 3D-Punkte mit den Dimensionen der realen Umwelt in Beziehung zu setzen. Dafür muss in den Aufnahmen ein Gegenstand oder ähnliches vorhanden sein, dessen Dimensionen oder Koordinaten bekannt sind. Das kann zum Beispiel ein Schachbrettmuster mit den bekannten Maßen der Quadrate dieses Schachbretts sein, wie es auch schon bei der Kamerakalibrierung der Fall war. Oder es werden markante Punkte mit bekannten Koordinaten der realen Welt in Form von Markern verwendet.

6.3.1 Schachbrett

Mithilfe eines Schachbrettmusters ist es möglich ein Bild mit einem Maßstab zu versehen. Das Schachbrett bietet mehrere Punkte. Diese Punkte sind jedoch in einem Bereich des Bildes konzentriert. So befindet sich beispielsweise das Schmiedestück in der Bildmitte und das Schachbrettmuster in einer Ecke des Bildes (Abbildung 75). In diesem Anwendungfall ist eine Platzierung des Schachbrettmusters unter oder in der unmittelbaren Nähe des Schmiedestücks aufgrund der wirkenden Kräfte und Hitzeentwicklung nicht möglich.

Abbildung 75: Einsatz des Schachbrettmusters zur Einführung eines Maßstabes

6.3.2 Marker

Der Einsatz von Markern bietet den Vorteil, dass diese an mehreren Positionen um das Schmiedestück herum platziert werden können. Im Vergleich zum Schachbrettmuster wird hier nicht nur die Länge und Breite der Pixel georeferenziert, sondern mit 3D-Koordinaten der realen Umwelt gearbeitet. Dazu ist es nötig, die Position der Marker im realen Raum zu bestimmen. Ebenso ist die Bestimmung der Position im Pixelraum nötig. Dabei können Marker unterschiedlichster Art zum Einsatz kommen. Durch die Verwendung von beispielsweise Aruco-Markern, ist es möglich diese Marker automatisch zu detektieren (Abbildung 76).

Abbildung 76: detektierte Aruco-Marker

Das automatische Detektieren ermöglicht das Auslesen der x- und y-Koordinaten des Pixelraumes der Marker. Die im Vorfeld bestimmten 3D-Koordinaten können anschließend den Pixelkoordinaten zugeordnet werden.

Die Positionen der Kameras bleiben während der Anwendung unverändert. Ebenso unverändert bleiben die Positionen der Marker. Aus diesem Grund ist nur eine einmalige Bestimmung der Markerpositionen notwendig. Diese können gespeichert werden. Für die weiteren Arbeitsschritte können die zuvor bestimmten Positionen geladen und wiederverwendet werden.

6.4 Berechnung eines Drahtgittermodells

Metashape bietet die Möglichkeit, anhand der generierten Punktwolke ein Drahtgittermodell abzuleiten. Die Berechnung eines Drahtgittermodells ermöglicht einen späteren Abgleich der 3D-Modelle. So können die Unterschiede zwischen den einzelnen Arbeitsschritten ermittelt

und die nächsten Arbeitsschritte besser geplant werden. Berechnet werden kann das Drahtgittermodell in 2.5D und 3D (Abbildung 77).

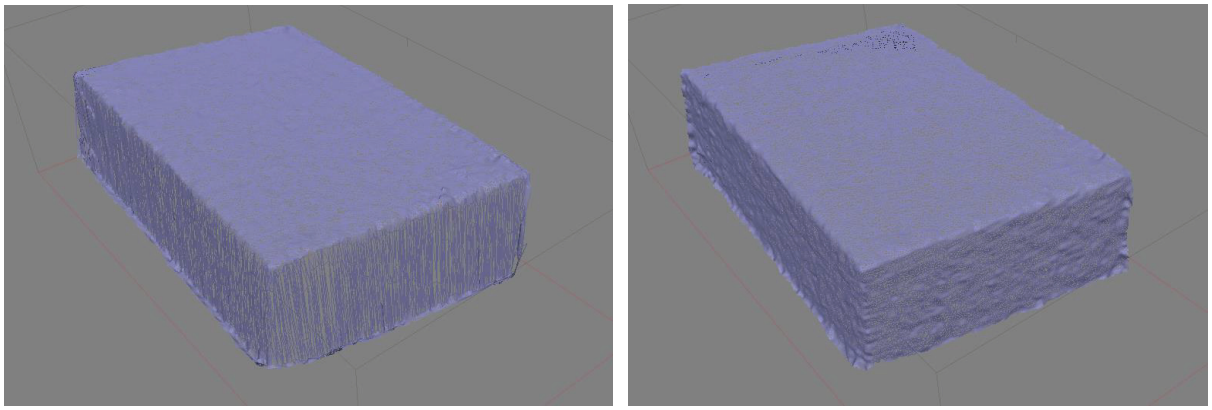

Abbildung 77: Berechnung eines Drahtgittermodells links in 2.5D und rechts in 3D

Die unterschiedlichen Berechnungsmethoden unterscheiden sich sowohl im Ergebnis als auch in der Berechnungszeit. Bei der ersten Variante ist das Ergebnis ein vereinfachtes 3D-Modell. Vergleichen lässt sich diese Methode mit einem Tuch, welches über die Punktwolke gelegt wird und sich dieser Punktwolke anschmiegt. Dieses Verfahren wird vorwiegend für Geländehöhenmodelle verwendet.

Die zweite Variante liefert ein komplexeres 3D-Modell als Ergebnis. Das eingesetzte Verfahren beruht auf der Poisson-Methode, welche in Abschnitt 4.2 beschrieben wurde. Deshalb kommt dieses Verfahren für komplexe 3D-Berechnungen zum Einsatz.

Die erste Variante benötigt eine Rechenzeit von 0,25 Sekunden, die zweite Variante 0,30 Sekunden. Wie genau die gelieferten Ergebnisse beider Verfahren in Bezug auf die Anwendungen sind, wird die Auswertung der erzielten Ergebnisse zeigen.

7. Bewertung

Anhand der erzielten Ergebnisse gilt es nun zu beurteilen, ob die geforderten Anforderungen umgesetzt werden konnten. Beurteilt wird dabei die zeitliche Komponente, die Genauigkeit und Anwendbarkeit auf die Anwendung der industriellen Schmiedestücke und wie sich das Verfahren bei Bearbeitung und Verformung des simulierten Bauteils verhält.

7.1 Erzielte Ergebnisse

Im Folgenden wird das Programm vorgestellt, welches im Rahmen dieser Arbeit entstanden ist (Abbildung 78). Das Programm dient zur Anwendung des gewählten Algorithmus. Zusätzlich bietet es Einstellungsmöglichkeiten, welche für die Anwendung nötig sind und die Anwendung einfacher gestalten.

Abbildung 78: Aufbau des eigenen Programmieransatzes

Das Programm bietet folgende Optionen: Kamerakalibrierung, Anpassung von Helligkeit und Kontrast, Detektion der Marker zur Einführung eines Maßstabes und Bestimmung der dazugehörigen Pixelkoordinaten, Filterfunktion zur Ausblendung unerwünschter Bereiche in den Bildern, Aufnahme der Bilder mit den zuvor eingestellten Parametern und eine Berechnung einer 3D-Punktwolke, auf Basis des Verfahrens Structure from Motion, mit den zuvor aufgenommenen Bildern. Für jede Option gibt es jeweils einen Button. Des Weiteren ist ein Hilfe-Button hinterlegt, welcher einen groben Überblick zur Menüführung bietet (Abbildung 79).

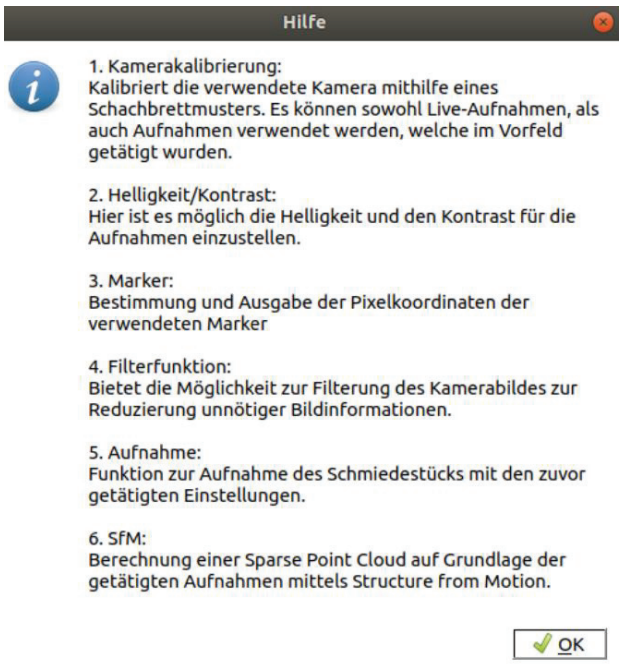

Abbildung 79: Funktion des Hilfe-Buttons

Der erste Button dient der Kalibrierung der eingesetzten Kamera. Bei Betätigung dieses Buttons öffnet sich das dazugehörige Untermenü (Abbildung 80).

Abbildung 80: Funktion Kamerakalibrierung

Zur Auswahl stehen zwei Optionen. Die Kalibrierung kann sowohl mit Liveaufnahmen (Live) als auch mit einer zuvor aufgenommenen Reihe von Bildern (Bildreihe) durchgeführt werden. Für beide Optionen wird das bereits erläuterte Kalibriermuster in Schachbrettform verwendet.

Bei Verwendung der Option Bildreihe ist es nötig den Dateipfad der aufgenommenen Bilder anzugeben. Dies geschieht ebenfalls über das Untermenü (Abbildung 81).

Abbildung 81: Untermenü der Funktion Kamerakalibrierung

Nach Auswahl des Bildpfades kann die Kalibrierung der Kamera über das Betätigen des Start-Buttons gestartet werden.

Nach dem Start wird für die Dauer der Berechnung eine Sanduhr eingeblendet (Abbildung 82).

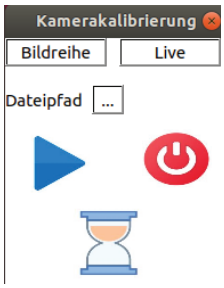

Abbildung 82: Einblendung des Sanduhr-Symbols für die Dauer der Berechnung

Über den zweiten Button kann Helligkeit und Kontrast des Bildes angepasst werden, um unterschiedliche Beleuchtungs- und Lichtverhältnisse zu kompensieren (Abbildung 83).

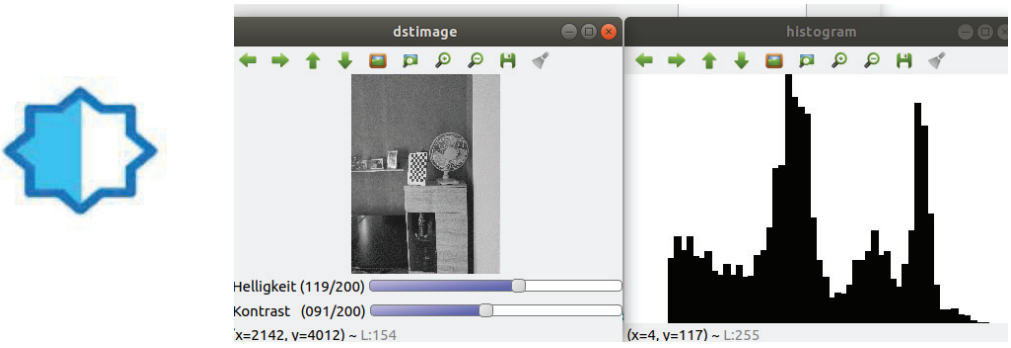

Abbildung 83: Funktion zur Anpassung von Helligkeit und Kontrast

Dieser Button öffnet das Bild der Kamera mit zusätzlichen Schiebereglern. Durch das Verschieben der beiden Regler verändert sich die Helligkeit und der Kontrast des Bildes. Der Standardwert ist auf 100 festgelegt. Zusätzlich wird das Histogramm angezeigt, über welches

man die Veränderung dieser beiden Werte optisch nachvollziehen kann. Sind die Werte wie gewünscht festgelegt, werden die Einstellungen über das Betätigen der Enter-Taste gespeichert.

Über den dritten Button erfolgt die Bestimmung der Position der Marker (Abbildung 84).

Abbildung 84: Funktion zur Detektion der Marker

Wird die Detektion der Marker gestartet, so werden die Marker, bei Verwendung von Aruco-Markern, automatisch erkannt (Abbildung 85).

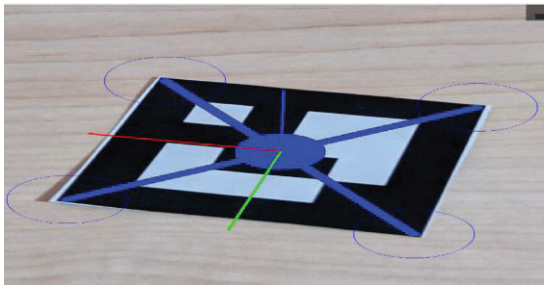

Abbildung 85: Detektion eines Aruco-Markers

Über die Eckpunkte der Marker wird das Zentrum eines jeden Markers berechnet. Die Pixelkoordinaten werden in einem separaten Fenster für jeden Marker ausgegeben (Abbildung 86).

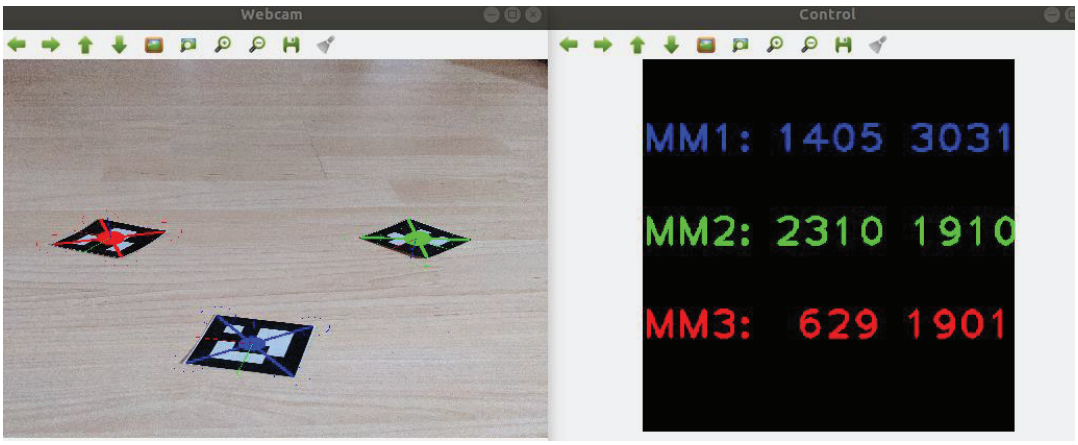

Abbildung 86: Ausgabe der Marker-Koordinaten

Eine weitere Option ist die Filterfunktion (Abbildung 87). Hier ist es möglich über eine Live Farbfilterung ungewünschte Elemente aus den Bildern zu entfernen. Das Herausfiltern von überflüssigen Elementen hat eine schnellere Berechnung der Punktwolke zur Folge.

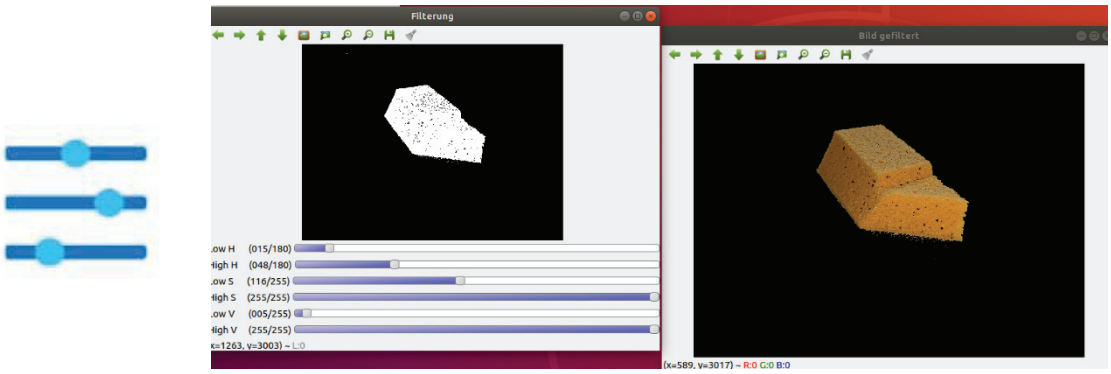

Abbildung 87: Darstellung der Filterfunktion

Im ersten Fenster wird die Filterung durchgeführt. Das zweite Fenster zeigt das gefilterte Live Bild der Kamera. Über sechs Schieberegler ist es möglich, verschiedene Farbtöne herauszufiltern. Die Farbfilterung basiert auf dem HSV-Farbraum. Die Farbe wird somit durch die drei Werte Hue, Saturation und Value beschrieben. Für jeden Wert sind zwei Schieberegler vorhanden, welche jeweils die Ober- und Untergrenze des jeweiligen Wertes beschreiben. Ist die Festlegung der Farbwerte zufriedenstellend, so können diese für den weiteren Verlauf gespeichert werden.

Die Schritte Kamerakalibrierung bis Filterfunktion müssen nur einmalig durchgeführt werden. Bei der Verwendung von mehreren Kameras an unterschiedlichen Positionen müssen diese nur einmalig für jede Kamera durchgeführt werden. Diese Schritte dienen zur Einrichtung des Systems und können für den weiteren Verlauf beibehalten werden.

Sind die zuvor erwähnten Einstellungen vorgenommen worden, so können nun Aufnahmen des Schmiedestücks getätigt werden (Abbildung 88).

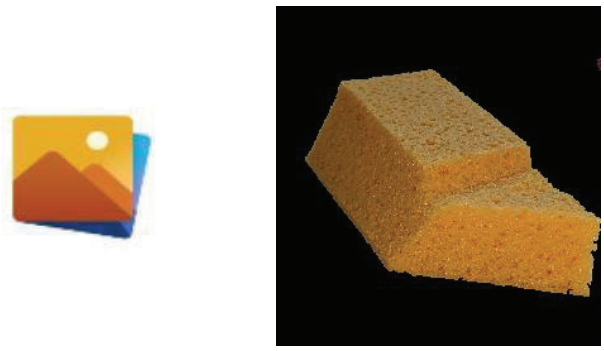

Abbildung 88: Aufnahmefunktion

Mit Hilfe der getätigten Aufnahme ist es nun möglich eine Punktwolke des Schmiedestücks zu berechnen (Abbildung 89).

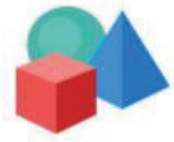

Abbildung 89: Funktion zur Berechnung einer Punktwolke nach dem Structure from Motion Verfahren

Über das Structure from Motion Verfahren werden mithilfe der Aufnahmen dreidimensionale Punkte des Schmiedestücks berechnet. Der eigene Programmieransatz ist jedoch nur in der Lage diesen Schritt mit ungefilterten Aufnahmen durchzuführen (Abbildung 90).

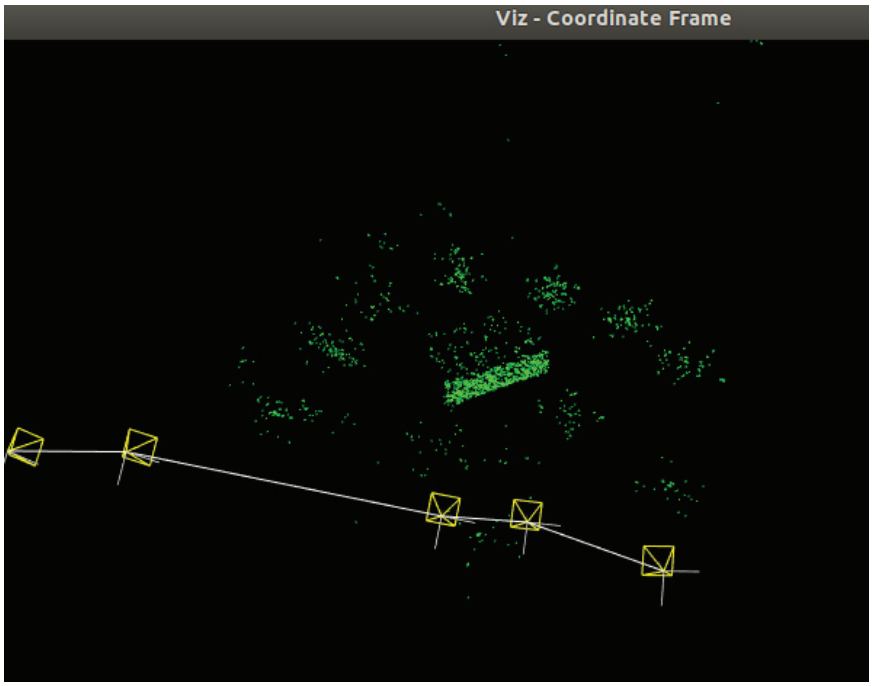

Abbildung 90: Berechnung einer Punktwolke aus ungefilterten Aufnahmen

Der eigene Programmieransatz ermöglicht eine Berechnung von 3D-Punkten auf Basis der getätigten Aufnahmen. Aufgrund des effizienteren Algorithmus der Software Metashape, bezogen auf Schnelligkeit und Robustheit, werden die Berechnungen mithilfe dieser Software durchgeführt.

Der Schwamm, welcher als simuliertes Bauteil fungiert, wird in unterschiedlichen Stadien aufgenommen. Ausgehend von der Originalform, werden die Dimensionen in drei Stufen verändert. Da sich die Rotation des Schmiedestücks im Laufe des Arbeitsprozesses verändert, wird auch der Schwamm in den verschiedenen Stufen mit unterschiedlichen Rotationen aufgenommen. Dies dient der Kontrolle der berechneten Punktwolke und der daraus resultierenden Maße des Schwamms.

Betrachtet werden die Ergebnisse nach der Berechnung eines Drahtgittermodells. Wie bereits erwähnt, bietet die Software Metashape zwei Optionen zur Ableitung eines Drahtgittermodells. Diese beiden Varianten werden im Folgenden untersucht und gegenübergestellt. Begonnen wird mit der Originalform des Schwamms (Abbildung 91).

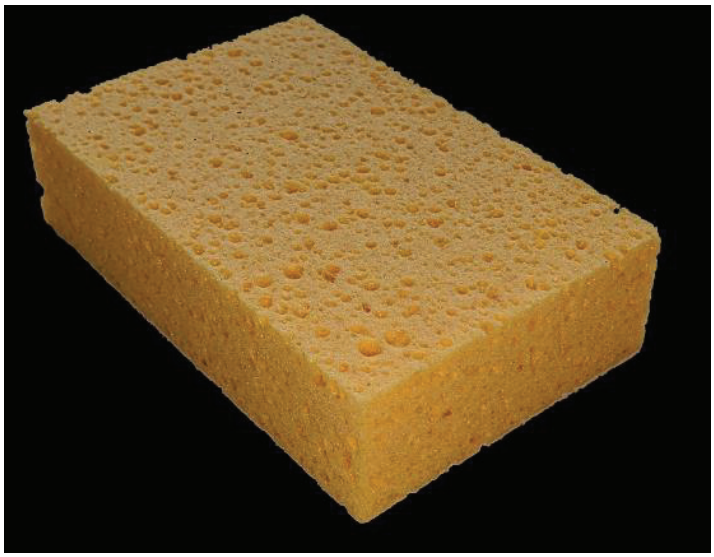

Abbildung 91: Bearbeitungsstufe 1

Abbildung 92 zeigt vier Ansichten des berechneten Modells in 2.5D.

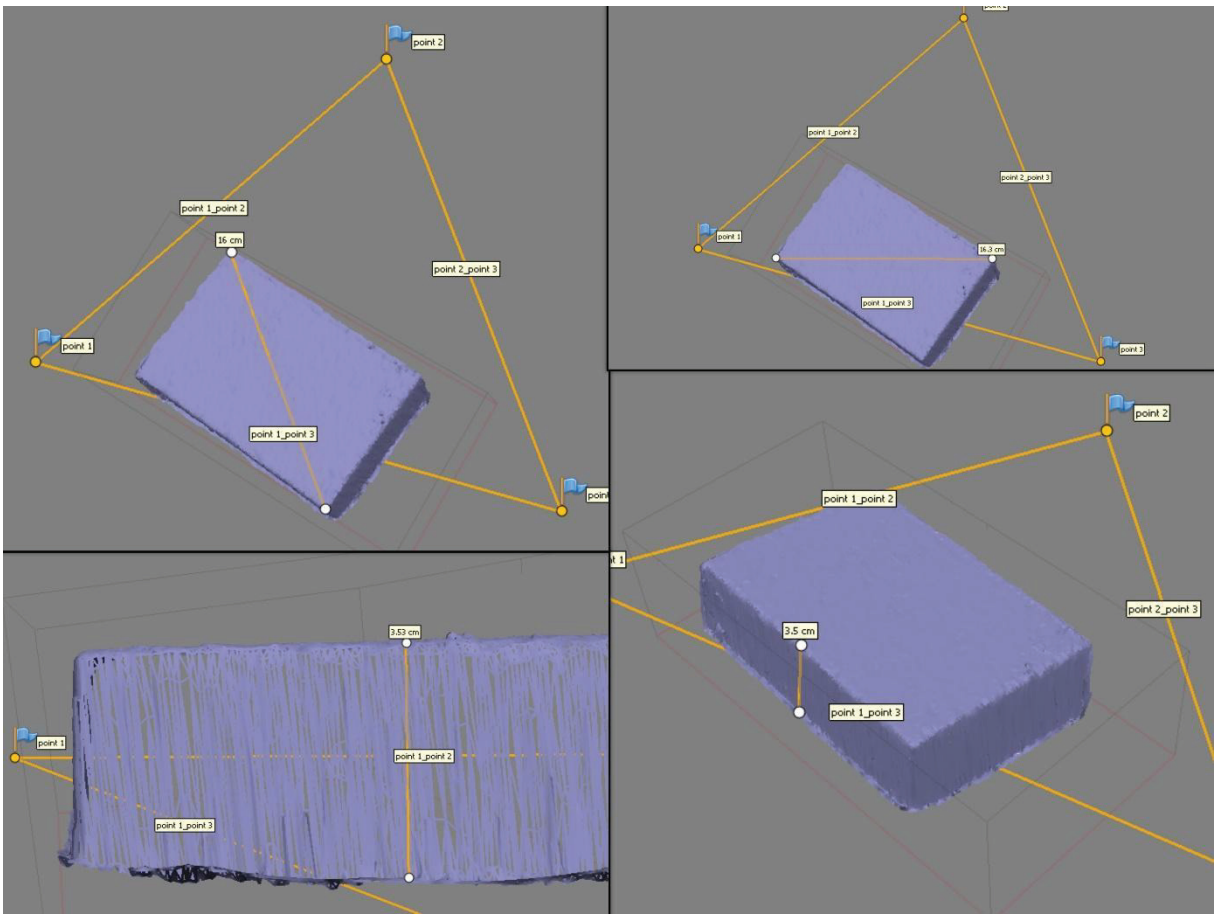

Abbildung 92: vier Ansichten des berechneten Modells in 2.5D von Bearbeitungsstufe 1

Abbildung 93 zeigt vier Ansichten des berechneten Modells in 3D.

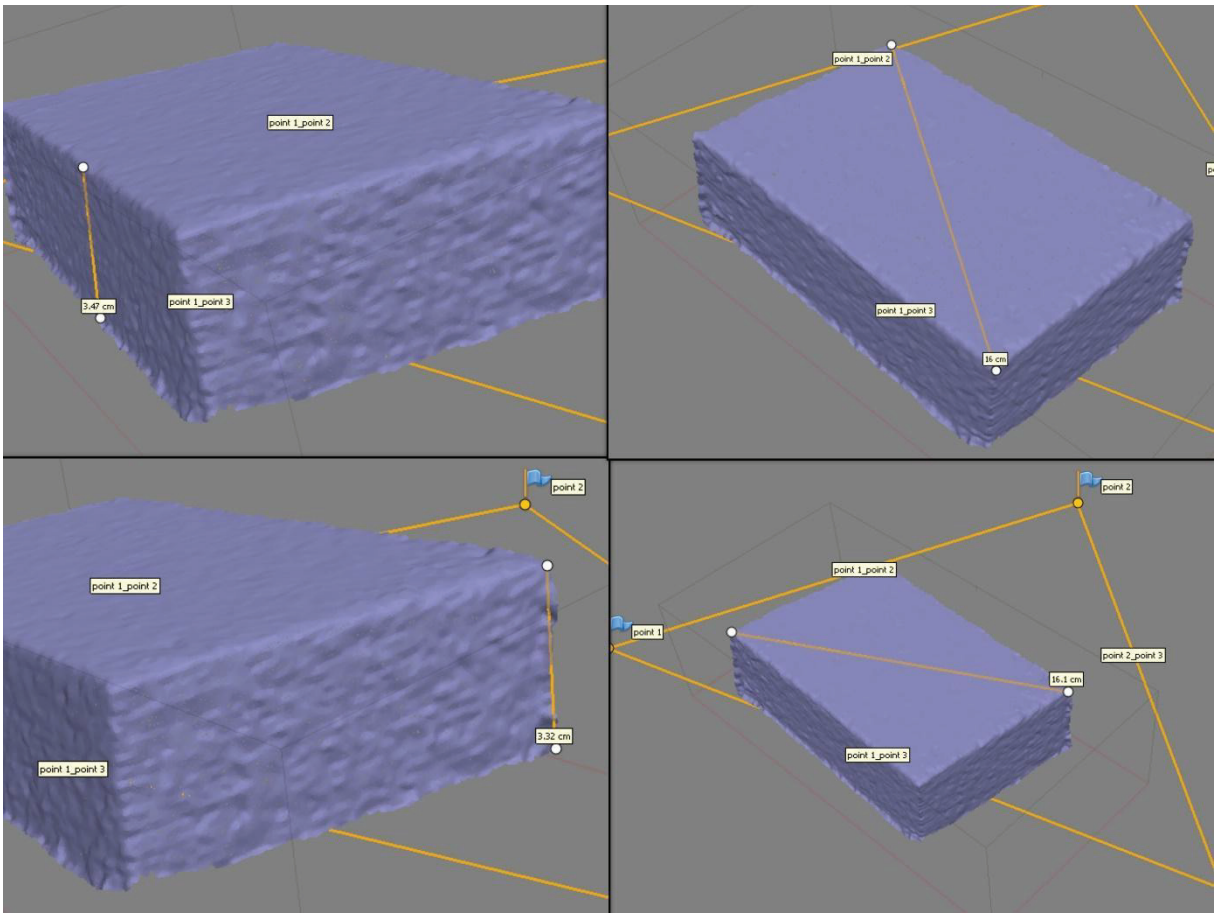

Abbildung 93: vier Ansichten des berechneten Modells in 3D von Bearbeitungsstufe 1

Die Referenzmaße des Schwamms wurden zuvor mit einem Messschieber bestimmt. Diese werden nun mit den Maßen aus den berechneten Modellen verglichen (Tabelle 10). Dafür wurden zuvor die Bilder mithilfe der detektierten Marker mit einem Maßstab versehen.

Maß	Original [cm]	2.5D [cm]	3D [cm]	Diff 2.5D [cm]	Diff 3D [cm]
1	3,50	3,50	3,32	0,00	0,18
2	3,50	3,53	3,47	-0,03	0,03
3	16,30	16,00	16,10	0,30	0,20
4	16,30	16,30	16,00	0,00	0,30
5	9,00	8,95	9,14	0,05	-0,14
6	13,50	13,40	13,50	0,10	0,00

Tabelle 10: Vergleich der Soll- und Istwerte der ersten Bearbeitungsstufe

Trotz der sehr unterschiedlichen Verfahren zur Ableitung eines Drahtgittermodells ist die Genauigkeit der Ergebnisse sehr ähnlich. Dies ist aber auch auf die unbearbeitete Rohform des simulierten Schmiedestücks zurückzuführen. Durch die Quaderform mit klaren Kanten kann auch das einfachere Verfahren gute Ergebnisse erzielen.

Als nächstes findet eine Bearbeitung des Schwamms statt. In diesem Schritt wird ein Teil des Schwamms entfernt (Abbildung 94).

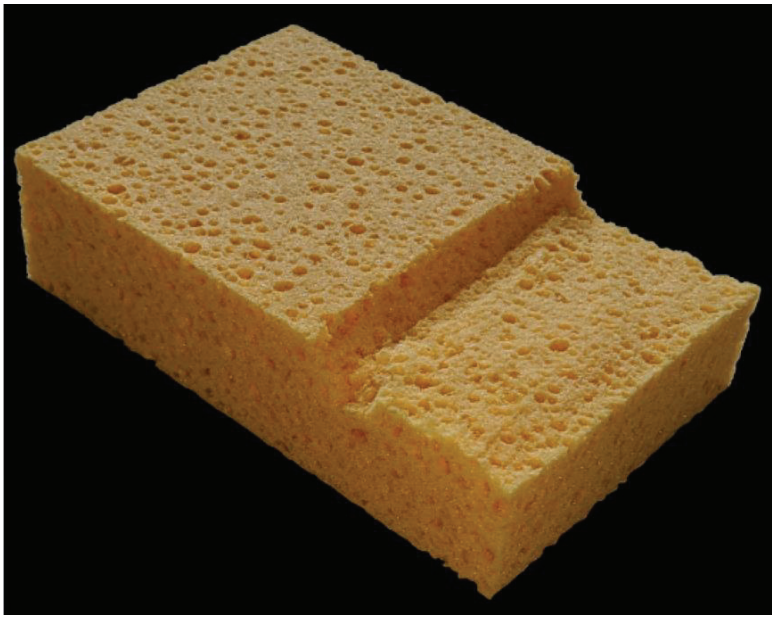

Abbildung 94: Bearbeitungsstufe 2

Abbildung 95 zeigt vier Ansichten des berechneten Modells in 2.5D.

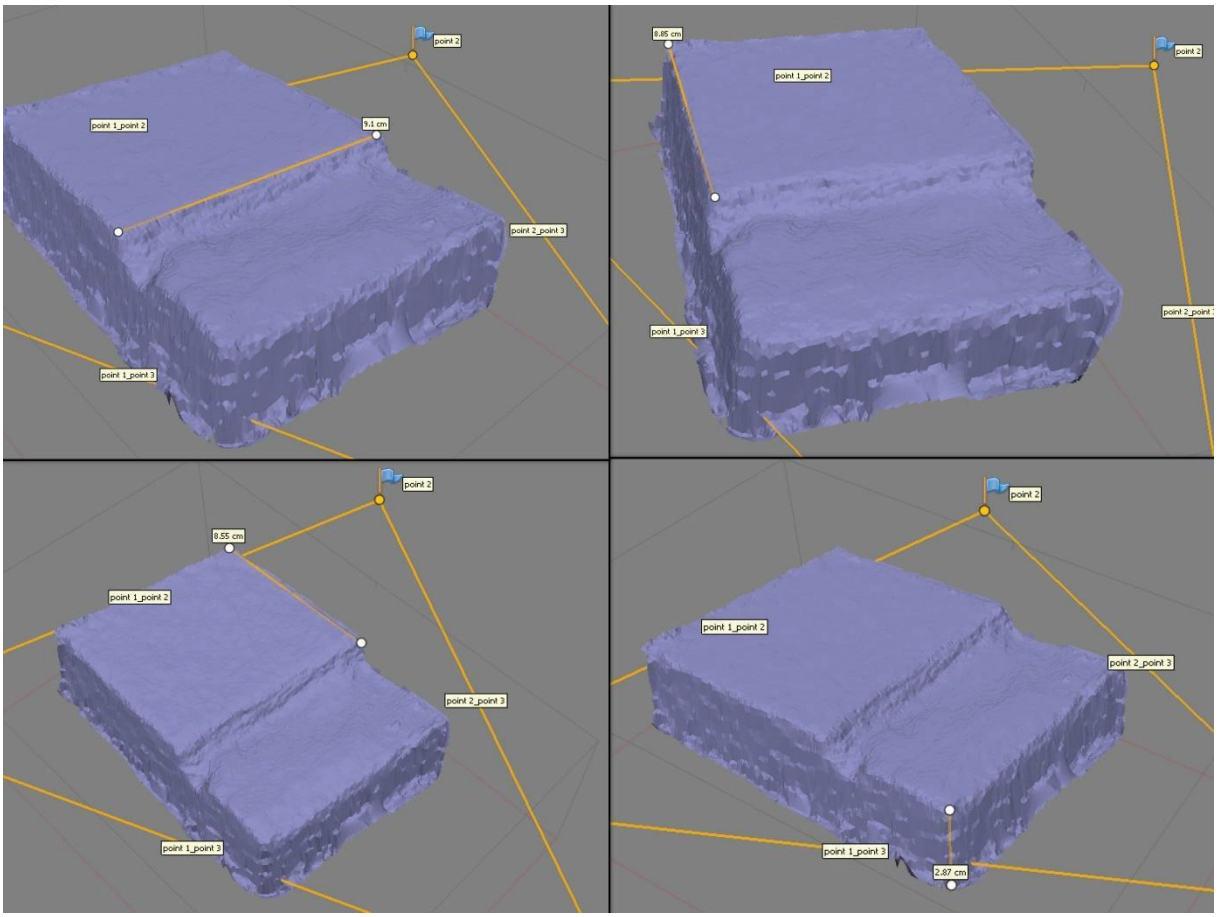

Abbildung 95: vier Ansichten des berechneten Modells in 2.5D von Bearbeitungsstufe 2

Abbildung 96 zeigt vier Ansichten des berechneten Modells in 3D.

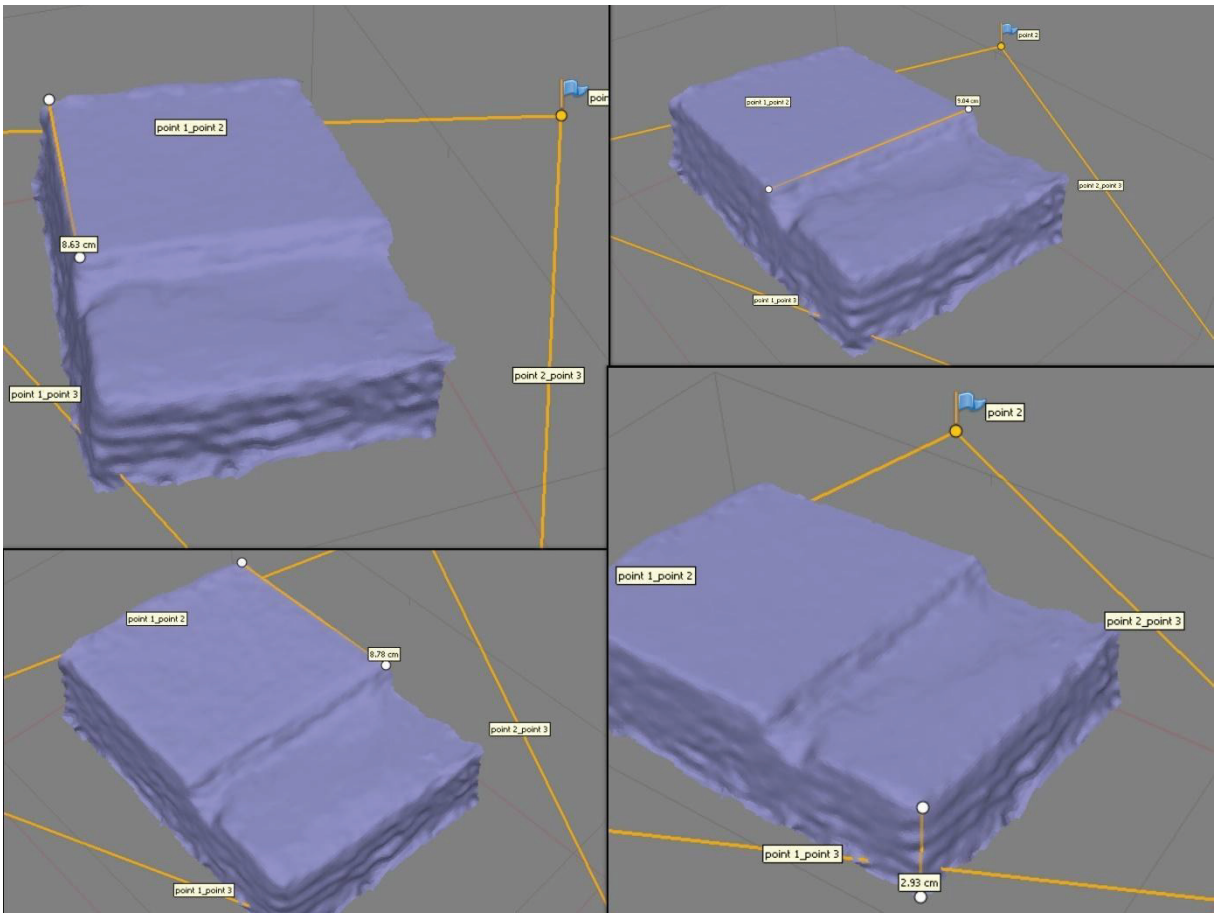

Abbildung 96: vier Ansichten des berechneten Modells in 3D von Bearbeitungsstufe 2

Tabelle 11 zeigt die Maße der berechneten Drahtgittermodelle für diese Bearbeitungsstufe.

Maß	Original [cm]	2.5D [cm]	3D [cm]	Diff 2.5D [cm]	Diff 3D [cm]
1	9,00	9,10	9,04	-0,10	-0,04
2	9,00	8,85	8,63	0,15	0,37
3	8,90	8,55	8,78	0,35	0,12
4	2,90	2,87	2,93	0,03	-0,03
5	2,80	2,61	2,63	0,19	0,17

Tabelle 11: Vergleich der Soll- und Istwerte der zweiten Bearbeitungsstufe

Im nächsten Schritt bleibt die Dimension des Schwamms unverändert. Es wird jedoch die Rotation verändert (Abbildung 97). Dies soll als Kontrolle dienen. Es wird geprüft, ob die Maße durch Veränderung der Rotation konstant bleiben.

Abbildung 97: Bearbeitungsstufe 3

Abbildung 98 zeigt vier Ansichten des berechneten Modells in 2.5D.

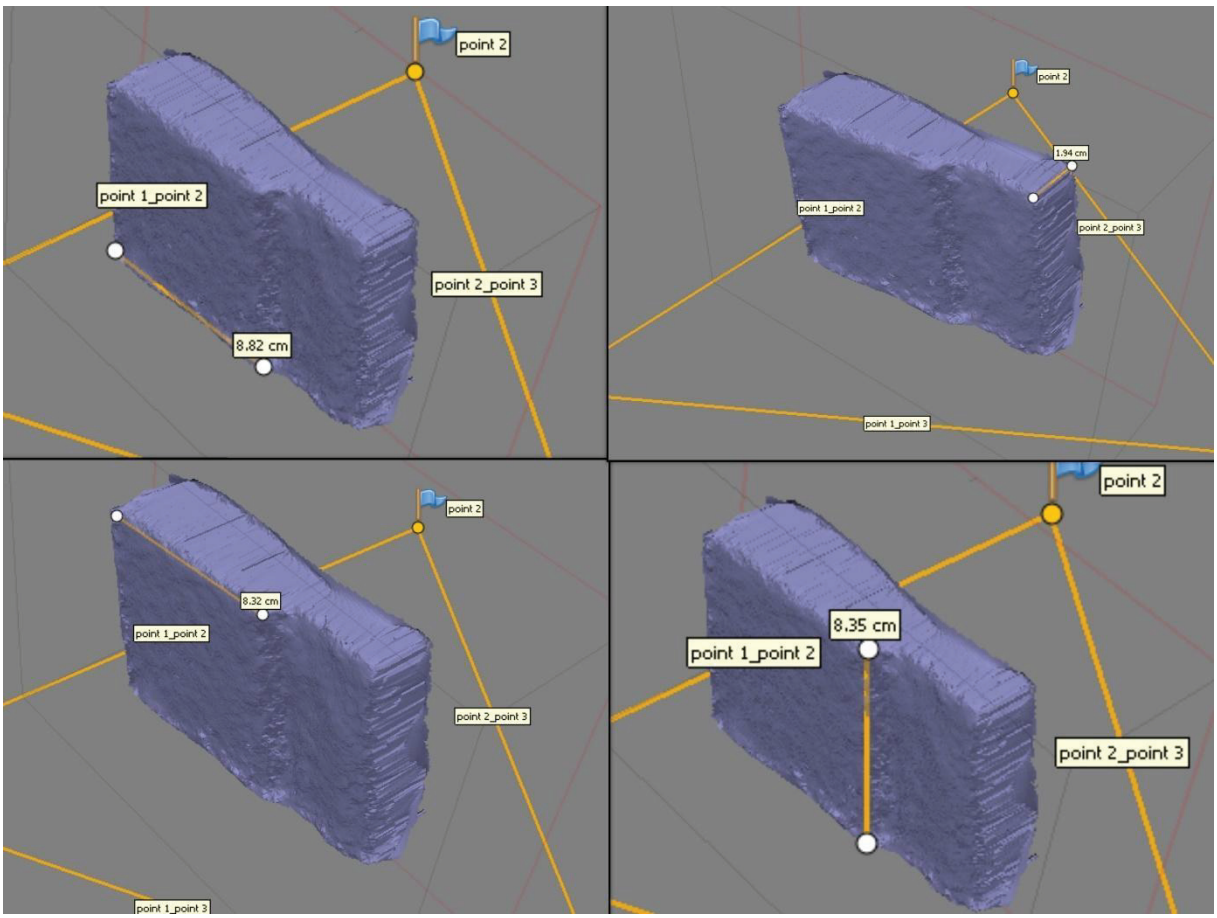

Abbildung 98: vier Ansichten des berechneten Modells in 2.5D von Bearbeitungsstufe 3

Abbildung 99 zeigt vier Ansichten des berechneten Modells in 3D.

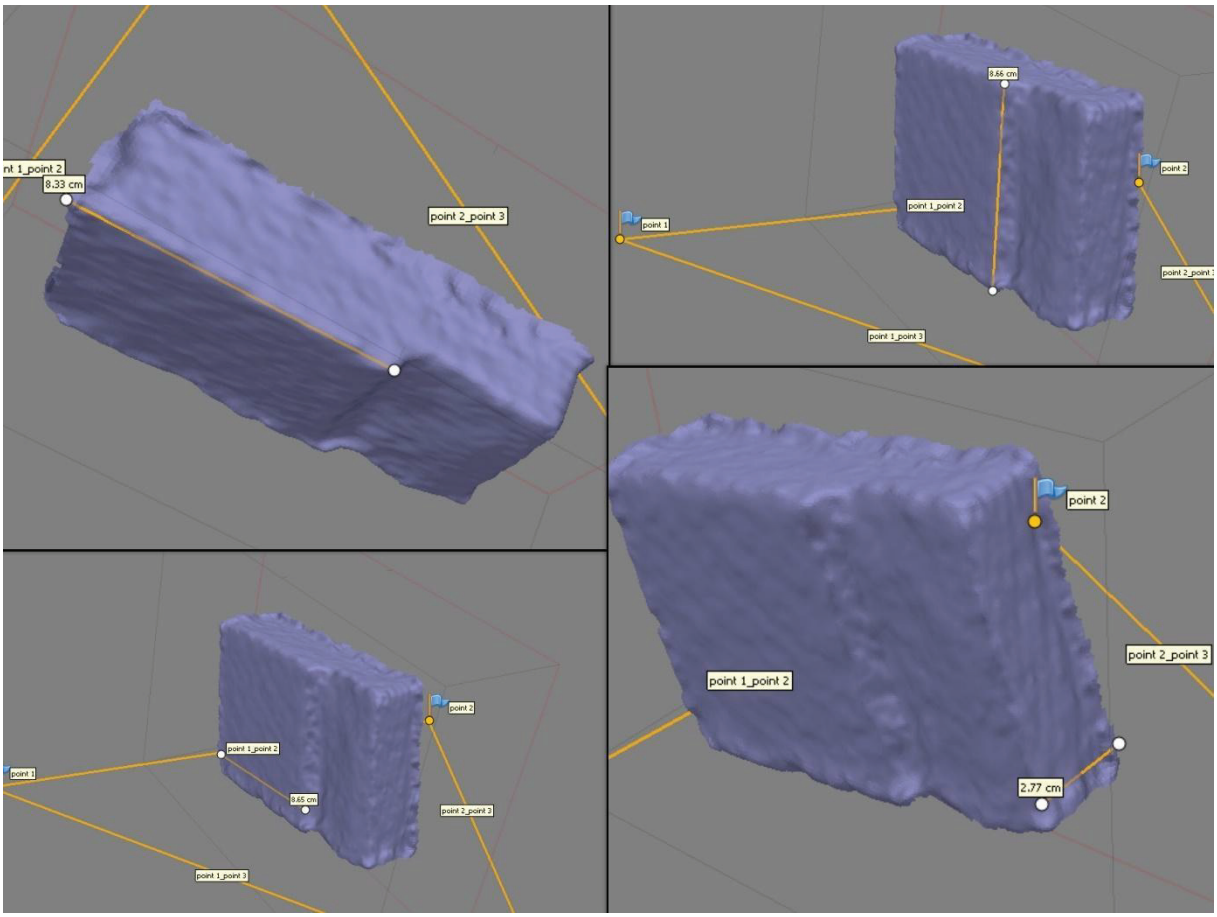

Abbildung 99: vier Ansichten des berechneten Modells in 3D von Bearbeitungsstufe 3

Tabelle 12 zeigt die Maße der berechneten Drahtgittermodelle für diese Bearbeitungsstufe.

Maß	Original [cm]	2.5D [cm]	3D [cm]	Diff 2.5D [cm]	Diff 3D [cm]
1	8,80	8,35	8,66	0,45	0,14
2	8,90	8,82	8,65	0,08	0,25
3	8,90	8,32	8,33	0,58	0,57
4	2,90	1,24	2,77	1,66	0,13
5	2,80	1,94	2,31	0,86	0,49

Tabelle 12: Vergleich der Soll- und Istwerte der dritten Bearbeitungsstufe

Aus Tabelle 12 geht hervor, dass bei dieser Bearbeitungsstufe erhöhte Abweichungen zu den Originalmaßen vorhanden sind. Diese Abweichungen sind bei beiden Modellen vorhanden. Einer der Gründe dafür ist die ungleiche Beleuchtung bei dieser Aufnahme. Bei der Betrachtung von Abbildung 97 wird deutlich, dass die Beleuchtung im oberen Bereich heller ausfällt als im unteren Bereich. Dies führt zu einer nicht optimalen Belichtung, was zu einer Verschlechterung der Qualität der Aufnahmen führt. Die Details sind dadurch zum Teil nicht mehr exakt erkennbar. Der Algorithmus ist jedoch auf diese Details in Form von

Schlüsselpunkten angewiesen und hat somit Probleme diese zu detektieren und die Bilder miteinander zu verknüpfen. Die Folge ist ein fehlerhaftes 3D Modell.

Maß 4 für das 2.5D Modell weist eine Abweichung von 1,66 cm zum Originalmaß auf. Der Grund dafür ist die fehlerhafte Berechnung des Drahtgittermodells an dieser Stelle. In diesem Bereich sind für den Algorithmus nicht ausreichend Punkte für eine adäquate Vermaschung vorhanden.

Im nächsten Schritt wird erneut die Dimension des Schwamms verändert. Geändert wird nur die Höhe. Die Rotation bleibt gleich (Abbildung 100).

Abbildung 100: Bearbeitungsstufe 4

Abbildung 101 zeigt drei Ansichten des berechneten Modells in 2.5D.

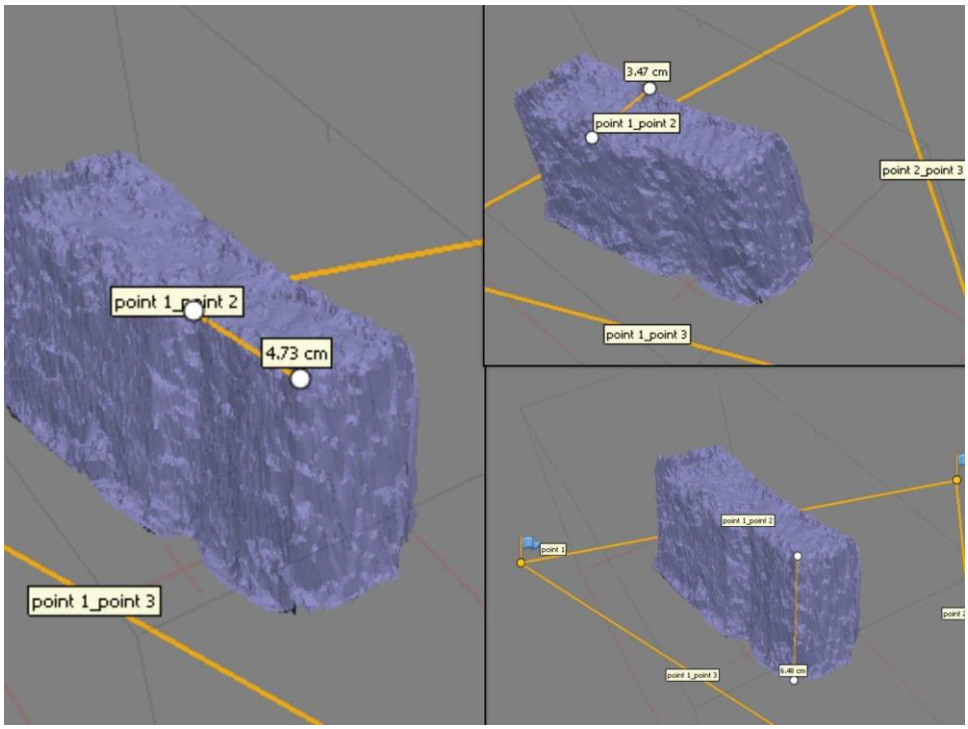

Abbildung 101: drei Ansichten des berechneten Modells in 2.5D von Bearbeitungsstufe 4

Abbildung 102 zeigt drei Ansichten des berechneten Modells in 3D.

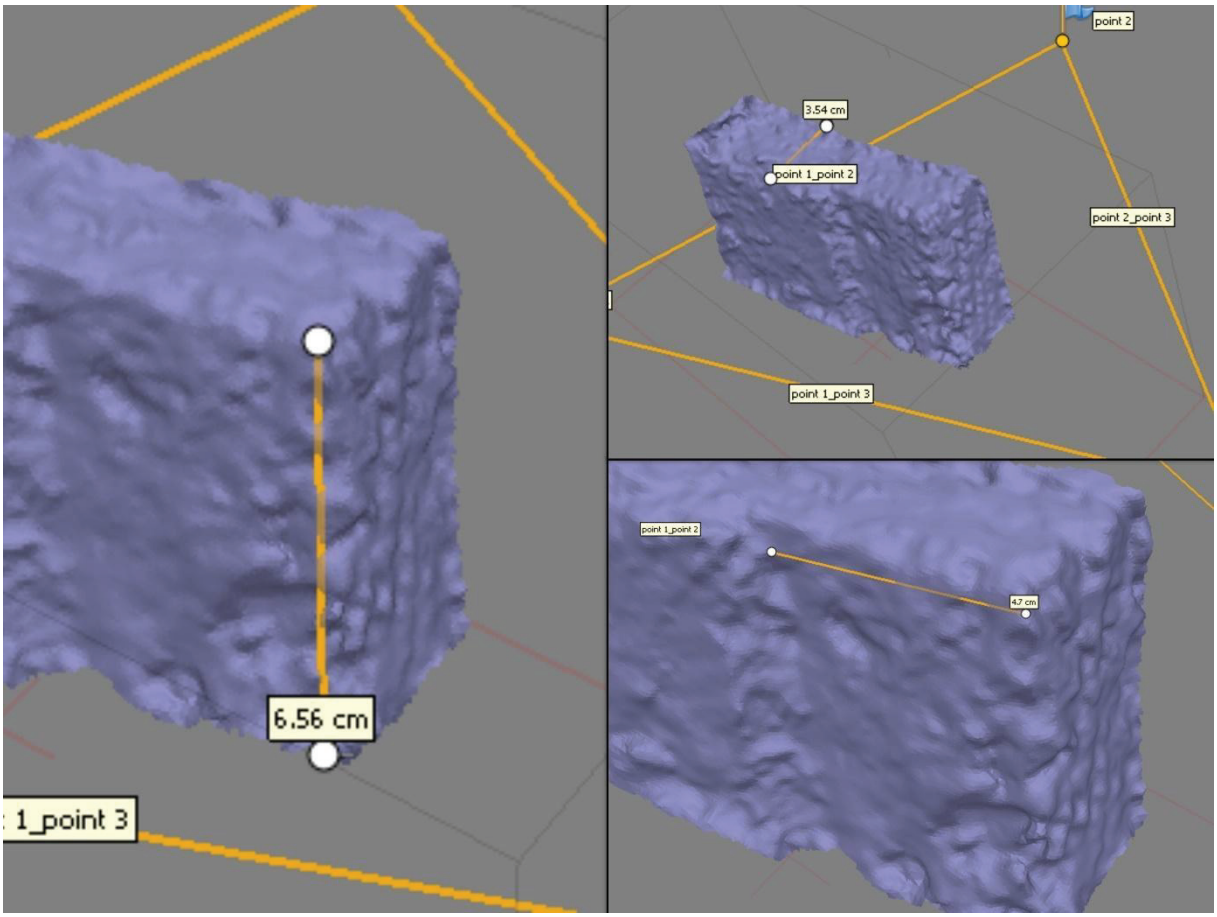

Abbildung 102: drei Ansichten des berechneten Modells in 3D von Bearbeitungsstufe 4

Tabelle 13 zeigt die Maße der berechneten Drahtgittermodelle für diese Bearbeitungsstufe.

Maß	Original [cm]	2.5D [cm]	3D [cm]	Diff 2.5D [cm]	Diff 3D [cm]
1	6,55	6,48	6,56	0,07	-0,01
2	4,70	4,73	4,70	-0,03	0,00
3	3,50	3,47	3,54	0,03	-0,04

Tabelle 13: Vergleich der Soll und Ist Werte der vierten Bearbeitungsstufe

Im Vergleich zur vorherigen Bearbeitungsstufe sind bei diesem Arbeitsschritt die Abweichungen zu den Originalmaßen geringer. Der Grund dafür ist die bessere Ausleuchtung des Objektes und somit auch die bessere Qualität der Aufnahmen.

Der nächste Arbeitsschritt beinhaltet eine erneute Rotation des Schwamms. Die Dimensionen bleiben dabei unverändert (Abbildung 103).

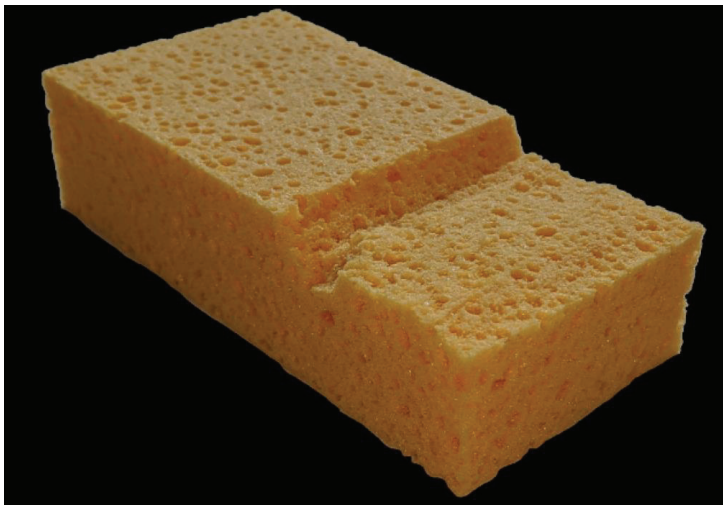

Abbildung 103: Bearbeitungsstufe 5

Abbildung 104 zeigt drei Ansichten des berechneten Modells in 2.5D.

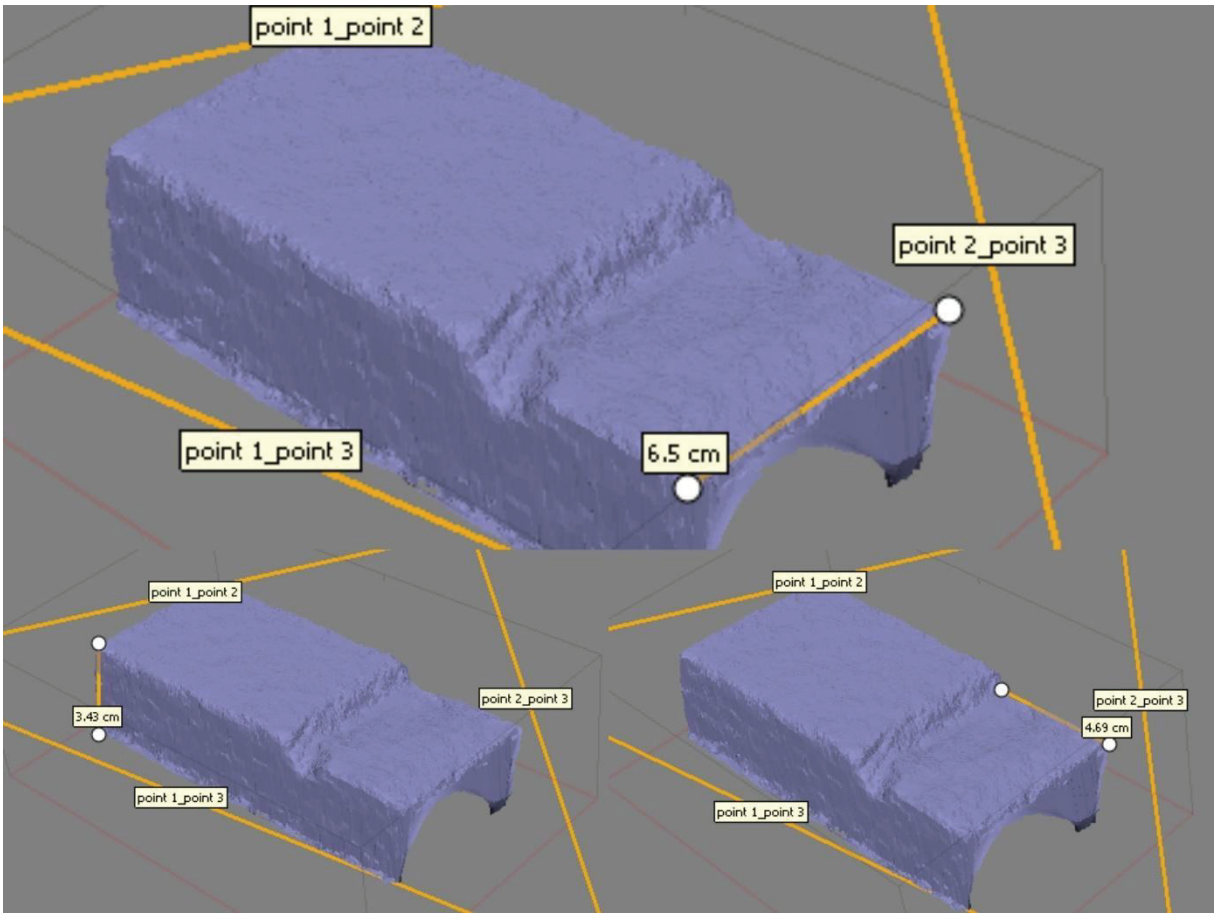

Abbildung 104: drei Ansichten des berechneten Modells in 2.5D von Bearbeitungsstufe 5

Abbildung 105 zeigt drei Ansichten des berechneten Modells in 3D.

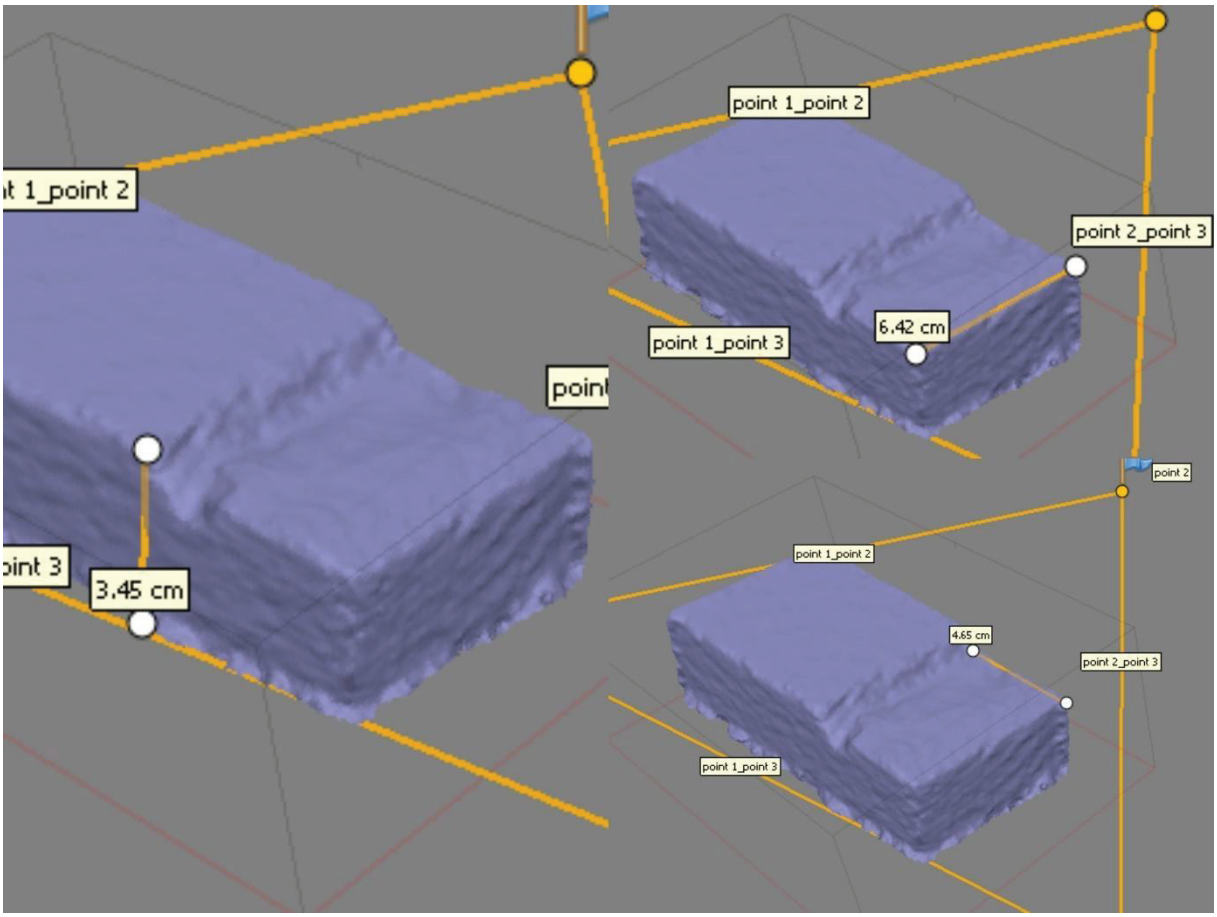

Abbildung 105: drei Ansichten des berechneten Modells in 3D von Bearbeitungsstufe 5

Tabelle 14 zeigt die Maße der berechneten Drahtgittermodelle für diese Bearbeitungsstufe.

Maß	Original [cm]	2.5D [cm]	3D [cm]	Diff 2.5D [cm]	Diff 3D [cm]
1	6,55	6,50	6,42	0,05	0,13
2	4,70	4,69	4,65	0,01	0,05
3	3,50	3,45	3,45	0,05	0,05

Tabelle 14: Vergleich der Soll und Ist Werte der fünften Bearbeitungsstufe

Die Maße sind nach der Rotation konstant geblieben. In Abbildung (Abbildung 103) wird jedoch deutlich, dass auch hier Probleme bei der Vermaschung der Puktwolke in 2.5 D aufgetreten sind.

Der nächste Bearbeitungsschritt beinhaltet die letzte Veränderung der Dimension des Schwamms (Abbildung 106).

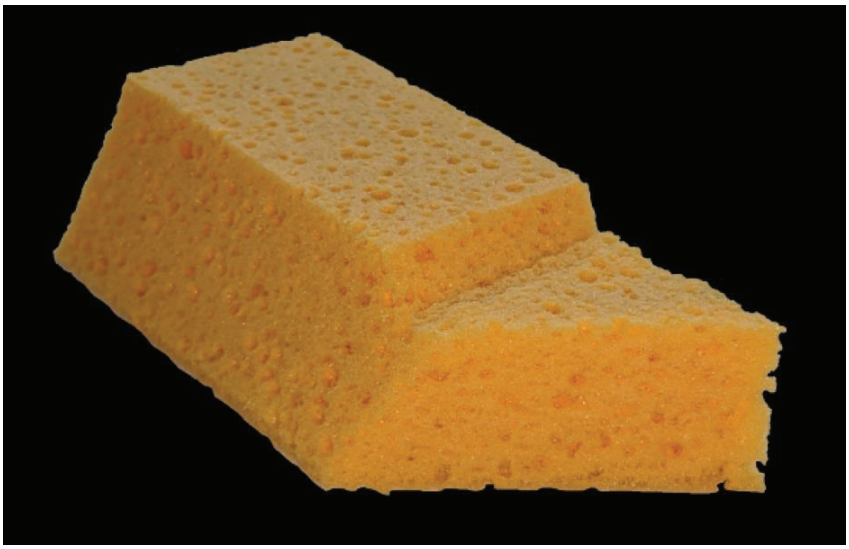

Abbildung 106: Bearbeitungsstufe 6

Abbildung 107 zeigt vier Ansichten des berechneten Modells in 2.5D.

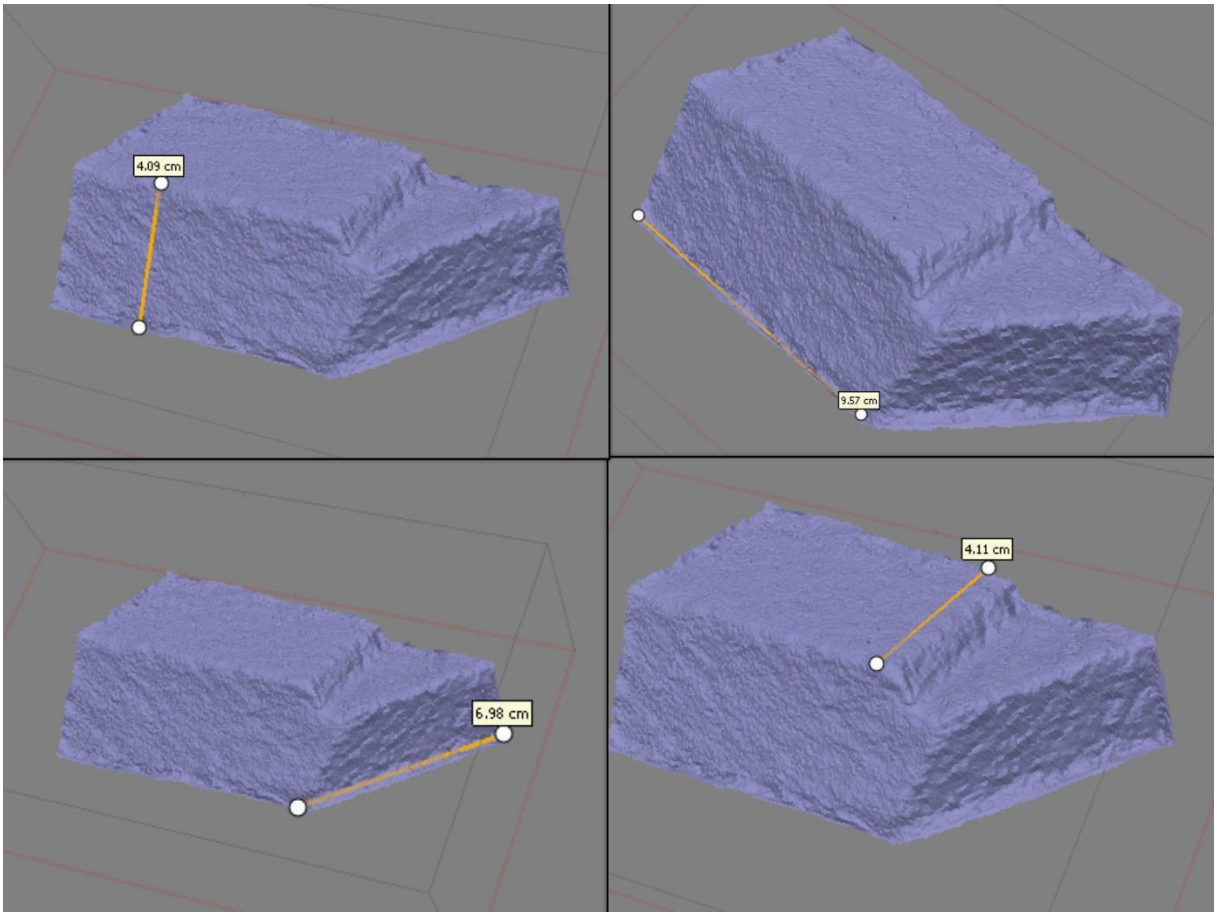

Abbildung 107: vier Ansichten des berechneten Modells in 2.5D von Bearbeitungsstufe 6

Abbildung 108 zeigt vier Ansichten des berechneten Modells in 3D.

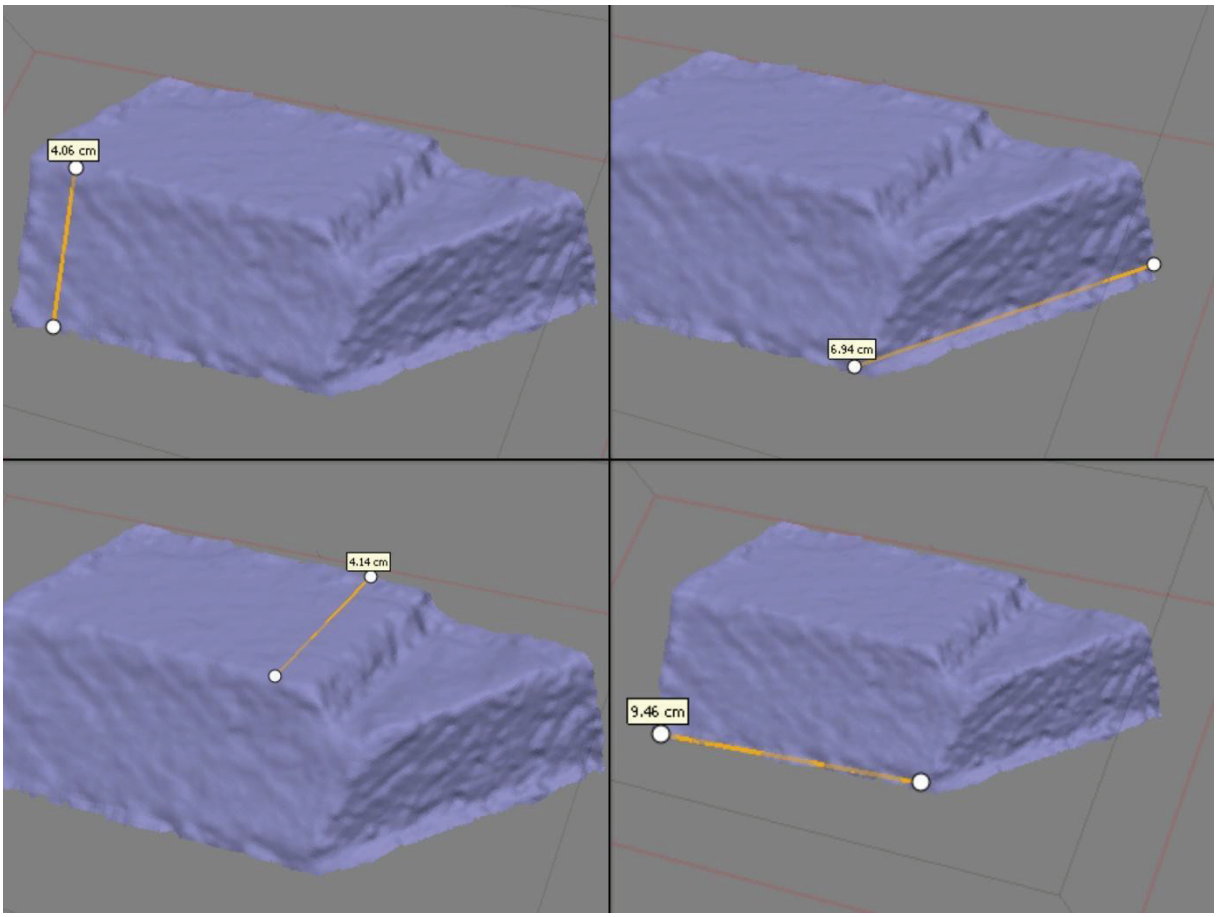

Abbildung 108: vier Ansichten des berechneten Modells in 3D von Bearbeitungsstufe 6

Tabelle 15 zeigt die Maße der berechneten Drahtgittermodelle für diese Bearbeitungsstufe.

Maß	Original [cm]	2.5D [cm]	3D [cm]	Diff 2.5D [cm]	Diff 3D [cm]
1	9,50	9,57	9,46	-0,07	0,04
2	7,10	6,98	6,94	0,12	0,16
3	8,70	8,81	8,93	-0,11	-0,23
4	4,30	4,11	4,14	0,19	0,16
5	4,05	4,09	4,06	-0,04	-0,01

Tabelle 15: Vergleich der Soll und Ist Werte der sechsten Bearbeitungsstufe

Die letzte Bearbeitungsstufe beinhaltet eine erneute Rotation des Schwamm mit gleichbleibenden Dimensionen (Abbildung 109).

Abbildung 109: Bearbeitungsstufe 7

Abbildung 110 zeigt vier Ansichten des berechneten Modells in 2.5D.

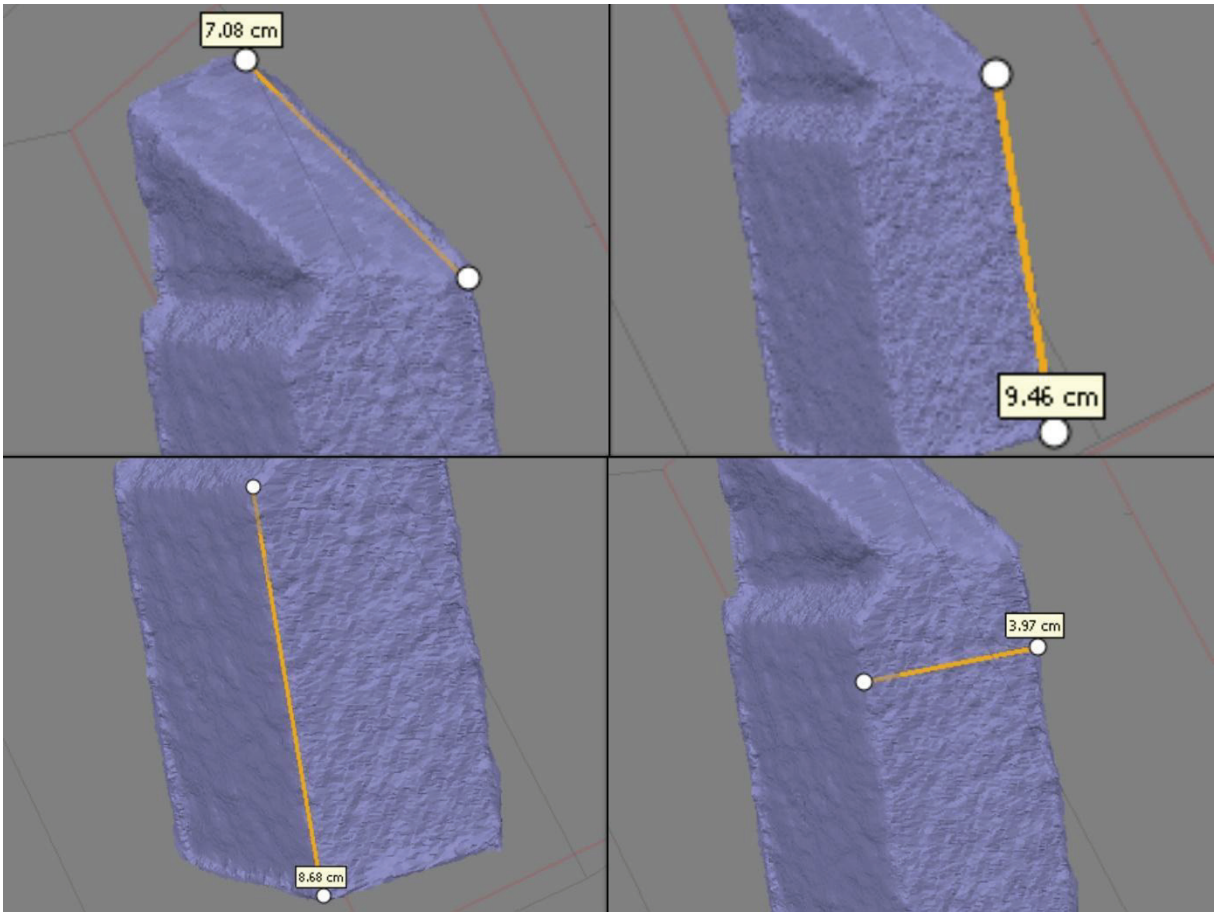

Abbildung 110: vier Ansichten des berechneten Modells in 2.5D von Bearbeitungsstufe 7

Abbildung 111 zeigt vier Ansichten des berechneten Modells in 3D.

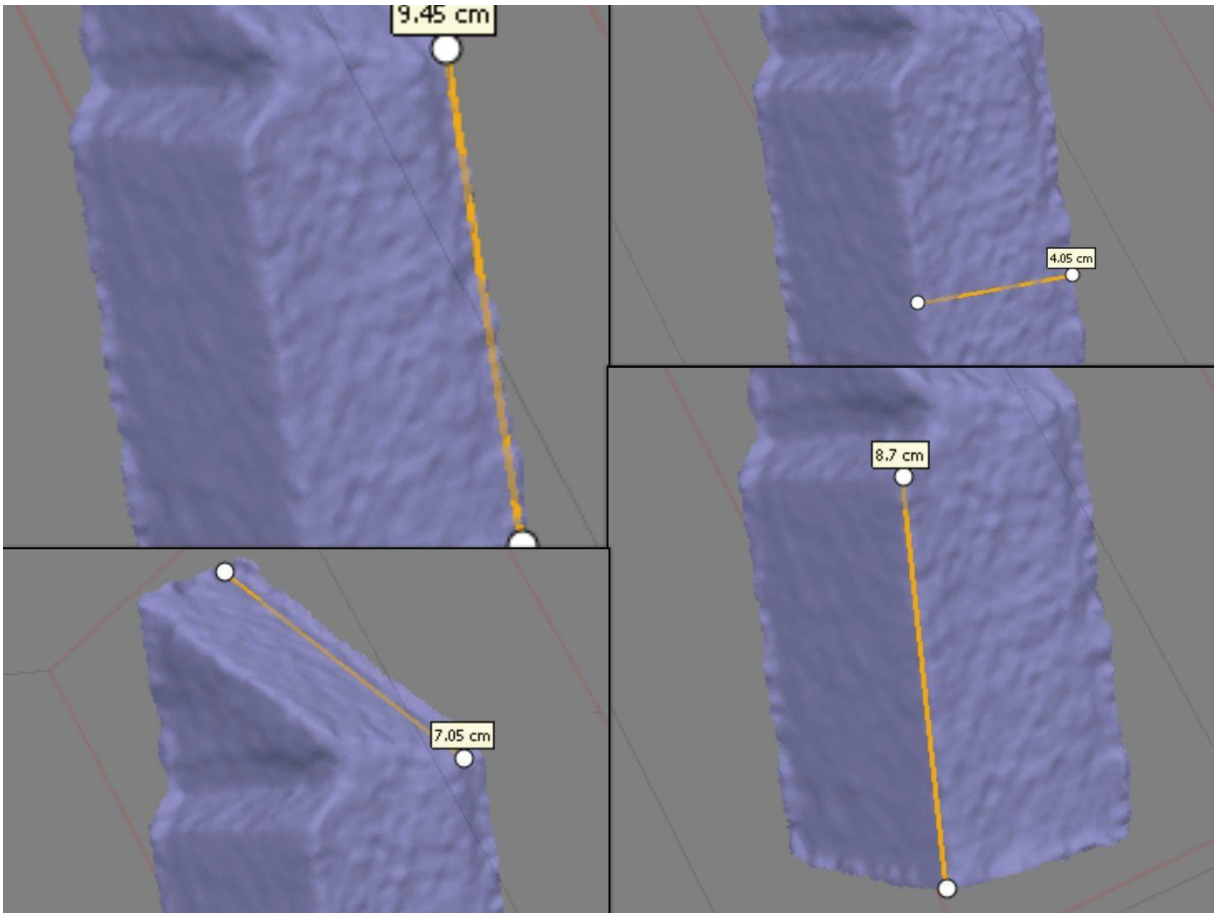

Abbildung 111: vier Ansichten des berechneten Modells in 3D von Bearbeitungsstufe 7

Tabelle 16 zeigt die Maße der berechneten Drahtgittermodelle für diese Bearbeitungsstufe.

Maß	Original [cm]	2.5D [cm]	3D [cm]	Diff 2.5D [cm]	Diff 3D [cm]
1	9,50	9,46	9,45	0,04	0,05
2	7,10	7,08	7,05	0,02	0,05
3	8,70	8,68	8,70	0,02	0,00
4	4,30	4,36	4,36	-0,06	-0,06
5	4,05	3,97	4,05	0,08	0,00

Tabelle 16: Vergleich der Soll- und Istwerte der sechsten Bearbeitungsstufe

Auch hier sind die Maße nach der Rotation stabil geblieben.

Zur Beurteilung des zeitlichen Limits von drei Sekunden wird für jede Bearbeitungsstufe eine zeitliche Analyse durchgeführt.

Tabelle 17 zeigt die Ergebnisse der zeitlichen Analyse.

	Sparse Point Cloud	Dense Point Cloud	Mesh 2.5D	Mesh 3D	Summe Point Cloud	Summe 2.5D	Summe 3D
Stufe 1	1,221	1,620	0,222	0,277	2,841	3,063	3,118
Stufe 2	1,319	1,526	0,253	0,302	2,845	3,098	3,147
Stufe 3	1,344	1,441	0,272	0,277	2,785	3,057	3,062
Stufe 4	1,340	1,490	0,283	0,292	2,83	3,113	3,122
Stufe 5	1,422	1,561	0,312	0,401	2,983	3,295	3,384
Stufe 6	1,390	1,567	0,240	0,275	2,957	3,197	3,232
Stufe 7	1,448	1,532	0,258	0,278	2,980	3,238	3,258

Tabelle 17: zeitliche Analyse der unterschiedlichen Arbeitsschritte für jede Bearbeitungsstufe [s]

Aus Tabelle 17 geht hervor, dass die Berechnung einer Punktwolke in Kombination mit der Verdichtung dieser Punktwolke das zeitliche Limit von drei Sekunden nicht überschreitet. Wird basierend auf der verdichteten Punktwolke zusätzlich eine Berechnung des Drahtgittermodells durchgeführt, wird das Limit leicht überschritten. Die Ergebnisse bewegen sich zwischen 3,06 und 3,30 Sekunden für die Modelle in 2.5D. In 3D bewegen sich die Ergebnisse zwischen 3,06 und 3,38 Sekunden.

Bei der zeitlichen Analyse wurde auch die zuvor durchgeführte Kamerakalibrierung verwendet. Durch die Einpflegung der Kamerakalibrierung in die Software Metashape kann wertvolle Rechenzeit eingespart werden, da die Software die Parameter nicht selbst berechnen muss.

7.2 Kritische Diskussion und Alternativen

Die Anwendung unter realen Bedingungen zu testen war im Rahmen dieser Arbeit nicht möglich, ebenso wie die Verwendung einer oder mehrerer professioneller Kameras. Eine gleichbleibende Beleuchtung, wie sie während des Arbeitsprozesses vorhanden ist, war im Verlauf des simulierten Beispiels nicht gegeben.

Dies hat sich zum Teil in den verschiedenen Bearbeitungsstufen des simulierten Schmiedestücks in Form der Genauigkeit widergespiegelt. Die Genauigkeitsanforderung von 1 bis 2 mm konnte jedoch bis auf vereinzelte Ausreißer erfüllt werden.

Der eigene Programmieransatz weist, in Bezug auf die Effizienz zur Berechnung einer Punktwolke, im Vergleich zur Software Metashape, Defizite auf. Die Unterfunktionen des eigenen Programmieransatzes bieten jedoch eine gute Ergänzung zu Metashape. Die

Kombination des eigenen Programmieransatzes und der Software Metashape ermöglicht es, die zeitliche Vorgabe von 3 Sekunden zu erreichen.

Da die Anwendung unter realen Bedingungen nicht getestet werden konnte, ist die Anwendung der Filterfunktion im industriellen Umfeld nicht garantiert. Eine Alternative dazu wäre die Verwendung der Funktion GrabCut. GrabCut ist ebenfalls Bestandteil der Computer Vision-Bibliothek OpenCV. Bei dieser Funktion wird im Vorfeld der Vorder- und Hintergrund durch eine Maske markiert und anschließend voneinander getrennt. Vorausgesetzt das Schmiedestück befindet sich in den einzelnen Arbeitsschritten immer an der selben Position, wäre dieses Verfahren eine Alternative zur herkömmlichen Farbfilterung.

8. Zusammenfassung

Neben der Funktionsweise der verschiedenen Methoden zur 3D-Rekonstruktion, der Anforderungen durch das industrielle Umfeld, sowie die Erläuterung der Vor- und Nachteile der vorgestellten Methoden ist deutlich geworden, dass die Wahl des Algorithmus abhängig von den Anforderungen und dem Einsatzbereich der Anwendung ist.

Bezogen auf die Thematik der geometrischen Analyse von industriellen Schmiedestücken wurde durch den realisierten Lösungsansatz gezeigt, wie die 3D-Rekonstruktion eingesetzt werden kann.

Anhand der durchgeführten zeitlichen Analyse ist deutlich geworden, dass der gewählte Lösungsansatz das Potenzial bietet für diese Anwendung umgesetzt zu werden.

Somit ist das Verfahren Structure from Motion auch für eine schnelle 3D-Rekonstruktion von Objekten geeignet und ermöglicht eine schnelle Ableitung von 3D-Informationen mit anschließender Berechnung eines Drahtgittermodells.

Literaturverzeichnis

Berkley EECS. (2012, Oktober 25). *Meshbook*. people.eecs.berkeley.edu/. https://people.eecs.berkeley.edu/~jrs/papers/meshbook/chapter2.pdf

Brooke, S. (2019, Dezember 5). *How Computer Vision Is Disrupting Different Industries*. heartbeat. https://heartbeat.fritz.ai/how-computer-vision-is-disrupting-different-industries-78eb19e341d7

CleanPNG. (2016). *Green Circle* [Abbildung]. https://www.cleanpng.com/png-color-wheel-hsl-and-hsv-colorfulness-color-scheme-1788416/download-png.html

Fauster, E., Schalk, P. & Tratnig, M. (2016, Januar). *Calibration Method for Light Sectioning Measurement Systems*. University of Leoben. https://www.researchgate.net/publication/266294158_Calibration_Method_for_Light_Sectioning_Measurement_Systems

Fisher, R., Perkins, S., Walker, A. & Wolfart, E. (2003). *Morphology - Dilation*. Image Processing Learning Resources. https://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm

Foix, S., Alenya, G. & Torras, C. (2011). Lock-in Time-of-Flight (ToF) cameras: A survey. *IEEE SENSORS JOURNAL*, *11*(3), 1. https://www.researchgate.net/publication/224211658

Frauenhofer Vision. (2014). *Leitfaden zur optischen 3D-Messtechnik*. Fraunhofer Verlag.

Geng, J. (2011, März). *Structured-light 3D surface imaging: a tutorial*. IEEE Intelligent Transportation System Society. https://www.osapublishing.org/aop/fulltext.cfm?uri=aop-3-2-128&id=211561

Giscle. (2018, August 6). *Depth Maps - Giscle*. Medium. https://medium.com/@Giscle/depth-map-depth-calculation-ce4d914c6afd

Halama, R. (o. J.). *Forging Machines (hot forming) Selection Guide | Engineering360*. globalspec. Abgerufen 29. Juli 2020, von https://www.globalspec.com/learnmore/manufacturing_process_equipment/manufacturing_equipment_components/forging_equipment

handmap. (2016, Oktober 26). *HSV vs. RGB*. Hand Map. https://handmap.github.io/hsv-vs-rgb/

Horopter. (1999). Spektrum. https://www.spektrum.de/lexikon/optik/horopter/1361

Kaehler, A. & Bradski, G. (2017). *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library* (1. Aufl.). O'Reilly Media.

Krone, T. (2010). *Analyse und Qualifizierung von laufzeitbasierten 3D-Kameras*. Hochschule Heilbronn. https://www.hs-heilbronn.de/analyse-und-qualifizierung-von-laufzeitbasierten-3d-kameras-b1b0d0ab36a0d299

Kshirsagar, P. (2017, Oktober 16). *Morphological Processing-Erosion & Dilation*. YouTube. https://www.youtube.com/watch?v=7-FZBgrW4RE&t=217s

Laganiere, R. (2017). *OpenCV 3 Computer Vision Application Programming Cookbook - Third Edition* (3rd Revised edition). Packt Publishing.

Li, L. (2014, Mai). *Time-of-Flight Camera - An Introduction*. Texas Instruments. https://www.ti.com/lit/wp/sloa190b/sloa190b.pdf?ts=1596613846829&ref_url=https%253A%252F%252Fwww.google.de%252F#:~:text=A%203D%20time%2Dof%2Dflight,measured%20and%20translated%20to%20distance.

Mattoccia, S., Tombari, F. & Di Stefano, L. (2008, März 20). *Stereo vision enabling precise border localization*. vision.deis.unibo.it. http://vision.deis.unibo.it/~smatt/stereo_precise.html

Maur, P. (2002, Februar). *Delaunay Triangulation in 3D*. https://www.kiv.zcu.cz/site/documents/verejne/vyzkum/publikace/technicke-zpravy/2002/tr-2002-02.pdf

opencv dev team. (2014). *Camera calibration With OpenCV — OpenCV 2.4.13.7 documentation*. opencv.org. https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html

Oswald, M. & Toeppe, E. (2013). *Computer Vision Group - 3D Reconstruction from a Single Image*. tum. https://vision.in.tum.de/research/single_view_3d_reconstruction

Oswald, M., Toeppe, E. & Cremers, D. (2012). *Fast and Globally Optimal Single View Reconstruction of Curved Objects*. TU Munich.

Parikh, A. (2019, Juli 24). *Depth Map: What It Is and How to Create It*. OmniVirt. https://www.omnivirt.com/blog/depth-map/

Polytec. (2010). Das optimale Licht für die Bildverarbeitung. *QZ Qualität und Zuverlässigkeit*, *12*(6), 6–7. https://www.polytec.com/fileadmin/d/Photonik/Bildverarbeitung/INFO72_Licht_fuer_die_BV.pdf

Revopoint 3D. (2019, November 7). *Comparing Three Prevalent 3D Imaging Technologies*. Revopoint 3D Technologies Inc. https://www.revopoint3d.com/comparing-three-prevalent-3d-imaging-technologies-tof-structured-light-and-binocular-stereo-vision/

Saxena, A. (2019a, Dezember 10). *Exploring the potential of computer vision across industries*. einfochips. https://www.einfochips.com/blog/exploring-the-potential-of-computer-vision-across-industries/

Saxena, A. (2019b, Dezember 11). *3 Mega Trends Driving the Future of Automotive IoT Solutions*. einfochips. https://www.einfochips.com/blog/3-mega-trends-driving-the-future-of-automotive-iot-solutions/

Schreiber, L. (1989). *Messung gekrümmter Flächen mit berührungslosen Verfahren*. Springer Publishing.

Shavit, A. (2017, Mai 31). *OpenCV 3 Projects for Photo Filtering [Video]*. packtpub. https://www.packtpub.com/application-development/opencv-3-projects-photo-filtering-video

Shewchuk, J. R. (2012, Februar 5). *Lecture Notes on Delaunay Mesh Generation*. Berkley EECS. https://people.eecs.berkeley.edu/~jrs/meshpapers/delnotes.pdf

Si, H. (2007, Juni 25). *3D Boundary Conforming Delaunay Mesh Generation*. wias-berlin. https://www.wias-berlin.de/events/insk/sommer07/si.pdf

Stereolabs. (2020). *Depth Sensing Overview | Stereolabs*. https://www.stereolabs.com/docs/depth-sensing/

Steyer, R. (2015, August 27). *Programmieren mit Qt Grundkurs*. Lynda. https://www.lynda.com/C-tutorials/Programmieren-Qt-Crashkurs/430331-2.html

TU Freiberg. (o. J.). *Poisson Surface Reconstruction*. 3D Scan 2.0. Abgerufen 6. August 2020, von http://vr.tu-freiberg.de/scivi/?page_id=365

Unavco. (2016). *Structure from Motion (SfM) Agisoft PhotoScan processing guide*. https://kb.unavco.org/kb/article/structure-from-motion-sfm-agisoft-photoscan-processing-guide-848.html

University of Cambridge. (2014). *Structure from Motion*. http://mi.eng.cam.ac.uk/. http://mi.eng.cam.ac.uk/~cipolla/publications/contributionToEditedBook/2008-SFM-chapters.pdf