



UNIVERSITÄT ZU LÜBECK

Automatisierte Echtzeit-3D-Rekonstruktion auf mobilen Geräten

*Automated Real-time 3D Reconstruction on Mobile Devices*

**Masterarbeit**

verfasst am

**Institut für Robotik und Kognitive Systeme**

im Rahmen des Studiengangs

**Informatik**

der Universität zu Lübeck

vorgelegt von

**Viktor Daibert**

ausgegeben und betreut von

**Prof. Dr. Elmar Rückert**

Lübeck, den 5. November 2019

### Eidesstattliche Erklärung

*Ich erkläre hiermit an Eides statt, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.*

---

Viktor Daibert

## Zusammenfassung

Im Rahmen dieser Masterarbeit wird mithilfe existierender Frameworks ein Prototyp entwickelt, der ein Smartphone in einen 3D-Scanner verwandelt. Der entstandene Prototyp setzt sich zusammen aus einer Smartphone-, Server-Anwendung und einer dazwischen befindlichen Datenbank. Auf der Server-Anwendung werden in Echtzeit Punktwolken aus Fotos erstellt. Die Punktwolke und weitere Unterstützungen sollen dem Anwender bei der Fotoerstellung und Rekonstruktion von Fotos nach 3D-Modellen assistieren. Für diesen Zweck werden nach einer einleitenden Betrachtung des theoretischen Hintergrunds gängige Algorithmen der 3D-Rekonstruktion auf ihre Vor- und Nachteile mit Hinblick auf die Rechenzeit und Effizienz mit der Zielplattform untersucht. Für die Umsetzung wird eine Kombination aus MeshLab, OpenMVG, OpenMVE und SMVS zum Konstruieren der 3D-Modelle verwendet. Mit Hinblick auf die Fehlertoleranz bei unsortierten Bildern haben sich AKAZE und DAISY im Kontext des Prototyps hervorgetan. Weshalb im Rahmen dieser Arbeit die Anwendung OpenMVG um die Verfahren AKAZE und DAISY erweitert wurden. Weiter beantwortet diese Arbeit die Forschungsfrage nach dem Wie einer solchen technologischen Realisierung.

## Abstract

In this master thesis, a prototype is being developed that uses existing frameworks to make a Smartphone to a pocket 3D scanner. The resulting prototype consists of a smartphone-, server-application, and a database in between. On the server application, point clouds from photos are created in real time. The point clouds and user support, should be used to assist during the photo creation and reconstruction of 3D-models from photos. After an introductory consideration of the theoretical background, common algorithms of 3D reconstruction are examined for their advantages and disadvantages with regard to the computing time and efficiency with the target platform. Based on the results, a combination of MeshLab, OpenMVG, OpenMVE, and SMVS is used to construct the 3D models. On the basis of fault tolerance with unsorted images, AKAZE and DAISY have been selected, which is why they are being supplemented in the OpenMVG application in this work. This work answers the question of How a technological can be implemented.

# Inhaltsverzeichnis

Glossar	1	
1	Einleitung	2
1.1	Motivation	2
1.2	Ziele der Arbeit	2
1.3	Einführung	3
1.4	Beitrag dieser Arbeit zur 3D-Rekonstruktion	5
1.5	Stand der Technik	5
1.6	Aufbau dieser Arbeit	9
2	Grundlagen	10
2.1	Sammeln der Bilder	11
2.2	Kamerakalibrierung	13
2.3	Erstellen einer Punktwolke	14
2.4	Texturieren der Punktwolke	19
2.5	Verfeinern des 3D-Modells	21
2.6	Smartphone Sensoren	21
2.7	Problem-, Kontext- und Aufgabenanalyse	22
3	Konzeption	26
3.1	Auswertung der Korrespondenzverfahren	26
3.2	Auswertung der Verfahren zur Modellerstellung	34
3.3	Vorteile der Client-Server-Architektur	39
3.4	Systemstruktur	40
4	Realisierung	47
4.1	Implementierung	47
4.2	Hardwareanforderung	53
5	Technische Evaluation und Validierung	54
5.1	Qualität der erstellten 3D-Modelle	55
5.2	Bewertung der Benutzbarkeit	56
5.3	Echtzeitfähigkeit	57
5.4	Ergebnis der Evaluation	58

6	Zusammenfassung	59
6.1	Diskussion über die 3D-Rekonstruktion des entwickelten Prototyps	60
6.2	Ausblick und offene Punkte	60
Literatur		62
A	Anhang A: Analyse der Korrespondenzverfahren anhand wissenschaftlicher Veröffentlichungen	66
B	Anhang B: Analyse der Korrespondenzverfahren anhand von Invarianten	70
C	Anhang C: Grafische Benutzeroberfläche der Smartphone-Anwendung	75
D	Anhang D: Belastungsanalyse	80



## Glossar

**AR3DROS** Abkürzung für *Automatisierte Echtzeit-3D-Rekonstruktion auf mobilen Geräten* ist die in dieser Masterarbeit implementierte Anwendung.

**FBM** Abkürzung für *Feature Based Matching* gehört zu den klassischen Image Matching Verfahren, bei dem Merkmale wie Punkte, Linien oder Regionen aus den Bildern verwendet werden, Gemeinsamkeiten zwischen den Bildern zu finden [6].

**FLANN** Abkürzung für *Fast Library for Approximate Nearest Neighbors* ist eine Ansammlung an Algorithmen, die eine hohe Menge an Merkmalen mit einer schnellen Bearbeitungsdauer miteinander vergleichen.

**Merkmalsbeschreibung** (englisch: *Feature Description*) ist ein Verfahren zum Beschreiben von Merkmalen, sodass diese selbst bei Veränderung des Blickwinkels oder Bildeigenschaften wie z. B. der Helligkeit eindeutig wiedergefunden werden können.

**Merkmalerkennung** (englisch: *Feature Detection*) dient zum Abstrahieren von Bildinformationen und Finden von lokalen Merkmalen in den Bildern.

**Merkmalsvergleich** (englisch: *Feature Matching*) ist eine Methode zum Vergleichen von Merkmalen in Bildern, welche durch die Merkmalsbeschreibung mit Informationen angereichert wurden.

**MeshLab** ist eine freie Programmbibliothek, welche für eine hohe Anzahl an Algorithmen zur Verarbeitung von Punktwolken und 3D-Modellen genutzt.

**MVS** entnimmt die Informationen, welche von SfM bereitgestellt werden und erstellt daraus eine verdichtete Punktwolke.

**OpenCV** Bei OpenCV handelt es sich um eine freie Programmbibliothek mit Algorithmen, die für die Bildbearbeitung und das maschinelle Sehen verwendet wird.

**Punktwolke** besteht aus einer Menge von Punkten eines Vektorraums, welche eine unorganisierte räumliche Struktur aufweisen. Der Vektorraum kann sich aus den Raumkoordinaten, Georeferenzierung, Farbwerten und Messgenauigkeit zusammensetzen [39].

**RANSAC** wird vornehmlich im Bereich des maschinellen Sehens eingesetzt, um ein Modell innerhalb einer Reihe von Messwerten mit Ausreißern und groben Fehlern zu schätzen.

**SfM** Abkürzung für *Structure from motion* ist eine Methode zum Schätzen einer dreidimensionalen Punktwolke aus zweidimensionalen Bildsequenzen.

# 1

## Einleitung

### 1.1 Motivation

Mobile Applikationen erleichtern den Nutzern in vielen Situationen den Alltag. Moderne Methoden der Computer Vision (CV), vor allem im Bereich der Fotogrammetrie, ermöglichen es dem Anwender, sein sichtbares Umfeld sowohl zu digitalisieren. Als auch durch dreidimensionale Modelle digital anzureichern. Dabei werden in sogenannten 3D-Rekonstruktionsprozessen durch Fotos oder Videos erfasste Daten dazu verwendet, 3D-Modelle aus dem physischen Raum digital zu rekonstruieren. Diese Technologie kann auf vielfältige Weise zum Einsatz kommen. Im Bereich des maschinellen Sehens in der Robotik können 3D-Rekonstruktionsprozesse dazu verwendet werden die exakte Positionierung und Charakteristik von Objekten im Raum zu berechnen. Ein praktisches Einsatzgebiet ist beispielsweise die Denkmalpflege. 3D-Modelle können dazu genutzt werden interessante Funde mit einem 3D-Scanner für die Zukunft virtuell zu sichern [52]. Dies erfordert jedoch die Anschaffung dedizierter und zumeist teurer Scanner-Hardware. Zusätzlich den Transport und Aufbau dieser Hardware vor Ort, was durch eine mobile Applikation erleichtert werden könnte [11, 12].

### 1.2 Ziele der Arbeit

Ziel dieser Masterarbeit ist es, eine Anwendung zu implementieren, welche automatisiert aus Fotos 3D-Objekte unter Erfassung von Echtzeitinformationen erstellt. Dabei soll aus den Fotos in Echtzeit eine Punktwolke hergestellt und dem Nutzer während der Fotoaufnahme präsentiert werden. Echtzeitfähigkeit bedeutet in diesem Kontext, innerhalb einer festgelegten minimalen Zeitspanne auf ein Ereignis zu reagieren. Da das hier entwickelte System für die Berechnungen eine Eingabe vom Anwender erfordert, wird die zeitliche Wahrnehmung eines Menschen als maximale Zeitspanne für die Echtzeitfähigkeit herangezogen. Die zeitliche Wahrnehmung eines Menschen beträgt im Durchschnitt etwa 3 Sekunden [26], in welchen aufeinanderfolgende Ereignisse als gegenwärtig wahrgenommen werden. Die Forschungsfrage dieser Arbeit richtet sich nach dem Wie der technologischen Realisierung von geeigneten Algorithmen zur Erreichung des Zieles.

Ein erster Ansatz zum Erreichen des Ziels ist es, eine Pipeline zu implementieren, welche automatisiert die Verarbeitung von Fotos bis hin zu einem 3D-Objekt ausführt. Das Systemkonzept setzt sich dabei aus drei Komponenten zusammen. Zum einen einer Anwendung für mobile Geräte (AR3DROS), mit dem Fotos erstellt und selektiert werden. Die zweite Komponente besteht aus einem Serverdienst, der den Smartphonedienst unterstützt und aus einem Satz an Fotos ein entsprechendes 3D-Modell konstruiert. Die dritte Komponente ist eine zwischen dem Smartphone und der Serveranwendung befindliche Datenbank. Diese erfüllt Anforderungen an Speicherung, Verarbeitung und Verwendung.

Weiter soll es Personen ohne jegliche Erfahrung in der 3D-Rekonstruktion, das Überführen von Fotos in 3D-Modelle vereinfachen. Um dies zu erreichen, werden mögliche Schwierigkeiten, die die Zielgruppe haben könnte, erfasst und analysiert. Auf Basis dieser Daten werden Hilfestellungen unter Einbeziehung der verwendeten Rekonstruktionsprogramme implementiert. Die so hergestellten Verfahren erweitern die bestehenden 3D-Rekonstruktionsdienste [13]. Aus dem Grund, dass bereits während des Erfassungszeitpunkts der Fotos sichergestellt wird, dass der Satz an Fotos die erforderliche Qualität und Vollständigkeit zum Erstellen eines 3D-Modells besitzt.

### 1.3 Einführung

Der Prozess zum Überführen von Fotos in 3D-Modelle wird *Bildabgleich* (englisch: *Image Matching*) genannt. Der Vorgang kann in flächenbezogenes Matching (englisch: *Area Based Matching* (ABM)) und eigenschaftsbasiertes Matching (englisch: *Feature Based Matching* (FBM)) unterteilt werden. Beim ABM werden die Bilder in Ausschnitte unterteilt und anhand ihrer Intensität miteinander verglichen. Bereits kleine Unterschiede im untersuchten Ausschnitt führen dazu, dass keine Gemeinsamkeiten gefunden werden können. FBM hingegen verwendet Merkmale wie Punkte, Linien oder Regionen. Dabei ist die Idee bei FBM, unverwechselbare und robuste Eigenschaften innerhalb der Bilder zu finden und die Ergebnisse weniger anfällig gegen Störungen zu machen [6]. Aufgrund der hohen Robustheit und Effizienz wird FBM in dieser Ausarbeitung näher untersucht.

Beim FBM-Verfahren gibt es verschiedene Modi, die gewählt werden können. Exhausting Matching vergleicht jedes Bildes miteinander. Sequential Matching nutzt Bilder, die in einer sequenziellen Reihenfolge vorliegen. Spartial Matching orientiert sich an der räumlichen Orientierung. Transitive Matching nutzt die transitive Beziehung der Bilder. In dieser Arbeit wird die Exhaustive Matching Methode genutzt, da keine Beziehung der Bilder vorausgesetzt werden kann.

Zum Erstellen eines 3D-Modells müssen beim FBM mit der Exhausting Matching Methode mehrere Prozesse durchlaufen werden. Zu Beginn werden mit der Merkmalerkennung die markanten Merkmale in den Bildern gesucht (siehe Abbildung 1.1 links), welche sich bei einer Veränderung der Helligkeit, Rotation, Skalierung oder weiterer Eigen-

schaften wie des Blickwinkels wiederfinden lassen. Dann werden die Merkmale durch eine Merkmalerkennung aus den Bildern ermittelt, welche durch die Merkmalsbeschreibung mit Informationen angereichert werden. Anschließend wird ein Merkmalsvergleich genutzt, um die gefundenen Merkmale miteinander zu vergleichen (siehe Abbildung 1.1 rechts). Gefundene zueinanderpassende Merkmale enthalten viele falsche Matches, weshalb Reduktionens-Algorithmen durchlaufen werden müssen. Zu den bekanntesten Vertretern der Reduktion gehört der Lowe Ratio-Test (siehe Abbildung 1.2 links). Nach dem Einsatz des Lowe Ratio-Tests bleiben außer den zueinanderpassenden Matches noch Ausreißer und grobe Fehler, welche mit einer Schätzung aus der Epipolargeometrie reduziert werden können. Bei der Epipolargeometrie wird versucht, die Bewegung zwischen zwei Frames zu schätzen. Hierzu werden mindestens 3 korrekte rekonstruierbare 3D-Punkte benötigt [22]. Der *random sample consensus* (RANSAC)-Algorithmus ist der erste Algorithmus, mit dem große Mengen an Ausreißern entfernt und zuverlässige Ergebnisse erzielt werden können [19]. Nach dem Einsatz von RANSAC kann begonnen werden die Homographie zwischen den Bildern zu suchen (siehe Abbildung 1.2 rechts). Bei der Homographie wird aus den Beschreibungen der zueinanderpassenden Bilderpaaren eine Matrix erstellt, welche die Transformation zwischen den Bilderpaaren beschreibt. Diese Informationen werden mithilfe der *Structure from Motion* (SfM) Methode in eine Punktwolke überführt, welche durch das *Multi-View Stereo* (MVS) Verfahren verdichtet wird (siehe Abbildung 3.1 links). Zum Abschluss werden über die erstellte Punktwolke mehrere Polygonnetze gespannt und mit Zuhilfenahme der aus den Bildern entnommenen Bildabschnitten texturiert (siehe Abbildung 3.1 rechts).

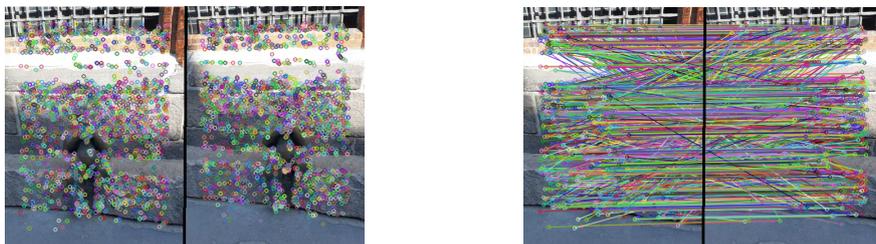


Abbildung 1.1: Links: Ergebnis der Merkmalerkennung durch den SIFT-Algorithmus. Rechts: Ergebnis des Merkmalsvergleichs durch den SIFT-Algorithmus.

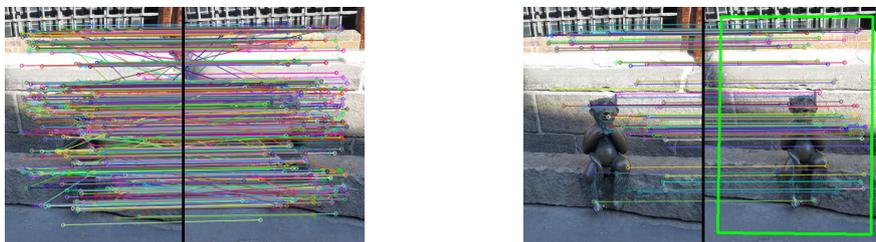


Abbildung 1.2: Links: Ergebnis des Lowe Ratio-Test zum Entfernen von Fehlern aus der Menge der gefundenen Merkmale. Rechts: Ergebnis von RANSAC zum Entfernen von Ausreißern aus dem Ergebnis des Lowe-Ratio-Test.

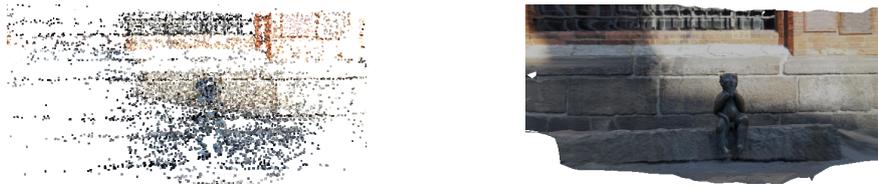


Abbildung 1.3: Links: Erstellte Punktwolke nach dem Einsatz des Structure from Motion und Multi-View Stereo Verfahrens aus 10 Bildern. Rechts: Durch das *Poisson Surface Reconstruction* Verfahren erstellte 3D-Modell.

### 1.4 Beitrag dieser Arbeit zur 3D-Rekonstruktion

In dieser Ausarbeitung wird überprüft, inwieweit eine mobile Anwendung zum Überführen von Fotos in ein 3D-Modell aus bestehenden Rekonstruktionsverfahren entwickelt werden kann. Unter der Verwendung des Frameworks *OpenCV* werden die Berechnungsvorgänge Merkmalerkennung, -extraktion, -beschreibung, -vergleich, Lowe Ratio-Test und RANSAC in einer Smartphone-Applikation implementiert. Im Hinblick darauf, wie diese dem Nutzer eine Hilfestellung bei der Fotoaufnahme geben könnten. Die in dieser Arbeit umgesetzten Hilfestellungen setzen sich zusammen aus einer Punktwolke, welche mit den bis dahin gemachten Fotos erstellt wird. Einem Verfahren, das überprüft, ob sich ein Objekt umwandeln lässt und wie das Objekt aufgenommen werden muss, um sich optimal umwandeln zu lassen. Sobald Fotos in ausreichender Stückzahl auf den Server hochgeladen wurden, sollen diese in ein 3D-Modell umgewandelt und dem Nutzer präsentiert werden. Hierfür werden die bekanntesten SfM und MVE Verfahren auf Basis ihrer Umsetzbarkeit der Rekonstruktionsverfahren für das Anwendungsziel miteinander verglichen. Die Analyse der Rekonstruktionsverfahren ergibt, dass AKAZE und Daisy am wenigsten fehleranfällig bei Bildveränderungen sind und eine hohe Anzahl anpassenden Matches finden. Weshalb die für diese Arbeit passendste Anwendung OpenMVG durch die Merkmalerkennung AKAZE und den Merkmalsdetektor DAISY aus dem Framework *OpenCV* erweitert wird.

### 1.5 Stand der Technik

Im Bereich der Echtzeitrekonstruktion auf mobilen Geräten sind in den vergangenen Jahren verschiedene Prototypen entstanden. In diesem Unterabschnitt soll der wissenschaftliche Stand näher untersucht werden. Dabei wird der Fokus auf Anwendungen gelegt, welche eine Echtzeitrekonstruktion von Fotos zu 3D-Modellen oder Punktwolken auf Smartphones ermöglichen. Zu jeder Arbeit wird untersucht, wie die Sensoren, Merkmalsextraktion und Vergleich der Merkmale eingesetzt werden. Ebenso wie eine Echtzeitfähigkeit erreicht wird. Anhand der Ergebnisse wird in Abschnitt 1.5 betrachtet, welche Ansätze in dieser Masterarbeit verwendet und abgegrenzt werden können. Alle vorgestellten Prototypen und deren Quellcode sind nicht öffentlich zugänglich. Daher werden die Informationen nur aus den wissenschaftlichen Publikationen erhoben.

### MonoFusion

MonoFusion nutzt die Videoeingabe einer Kamera um daraus ein 3D-Modell (siehe Abbildung 1.4) herzustellen. Bevor die Anwendung ausgeführt werden kann, muss der Nutzer seine Kamera anhand eines Schachbretts erfolgreich kalibrieren lassen und die sechs Freiheitsgrad (6DoF) des Objekts ermitteln. Sobald die Kamerakalibrierung abgeschlossen wurde, kann die Rekonstruktion begonnen werden. Hierzu wird mit der Kamera ein Video erstellt. Dem Video wird von jedem Frame ein Bild extrahiert, auf dem der FAST Algorithmus (siehe Abschnitt 2.3) eingesetzt wird, um die Merkmale zu bestimmen. Gleichzeitig wird zu jedem Bild eine Tiefenkarte erstellt. Weitere Informationen zum Objekt werden mittels Sensordaten erhoben, um die Bewegung der Kamera nachzuvollziehen. Zu jedem aufgenommenen Bild wird nach dem Einsatz des FAST Algorithmus ein dazu passendes Bild mittels RANSAC [19] aus der Bildermenge bestimmt. Gefundene Bilderpaare werden anhand der Merkmale in ein Modell übertragen. Anhand ihrer Orientierung und der Tiefenkarte mit den bereits erzeugten Modellen verschmolzen, wodurch ein verdichtetes Modell entsteht. In der Publikation wurde das Verfahren mit einer RGB Kamera und einer NVidia GTX580 Grafikkarte mit 30Hz getestet. Eine Besonderheit an MonoFusion ist, dass Lücken in der Punktwolke mit einer optimierten Variante von PatchMatch Stereo [7] ausgefüllt werden. Ein bedeuteter Nachteil an MonoFusion ist, dass viele Einstellungen vor der Erstellung des Modells gemacht werden müssen, ohne die eine Umwandlung nicht möglich ist. Es ist zwar denkbar die Anwendung direkt von einem Smartphone auszuführen. Jedoch ist die Smartphone-Hardware unzureichend, da eine leistungsstarke GPU benötigt wird, welche derzeit nur von einem Laptop oder Desktop PC zur Verfügung gestellt wird. Als möglicher Ansatz MonoFusion auf Smartphones auszuführen wird von den Autoren empfohlen die Videos auf einen Server zu übertragen, auf dem alle Berechnungen ausgeführt werden, und anschließend das fertiggestellte Modell auf dem Smartphone darzustellen [41].



Abbildung 1.4: Erstelltes 3D-Modell von MonoFusion der Entwickler. Zu finden in der Publikation: [41].

### MobileFusion

MobileFusion wurde in der Publikation nur an einem Apple iPhone 6 ausgeführt. Es nutzt sowohl die GPU als auch die CPU des Smartphones, um 3D-Modelle (siehe Abbildung 1.5) in Echtzeit aus einem Video mit 25Hz zu rekonstruieren. Dem Smartphone werden 25

Frames pro Sekunde extrahiert, zu jedem Frame die Kamera-Parameter berechnet und ähnliche Frames miteinander gruppiert. Das Besondere an MobileFusion ist, dass in den Bildern keine Merkmale gesucht werden, sondern ein ABM-Ansatz verfolgt wird, indem zwei zueinanderpassende Frames in eine Tiefenkarte überführt werden. Aus der Tiefenkarte wird ein 2,5D<sup>1</sup> Modell erstellt, welches anhand der Farbe, Tiefeninformationen und Orientierung mit anderen 2,5D Modellen zu einem 3D-Modell verschmolzen wird. Die Kamera Orientierung geschieht hierbei anhand des Volumenmodells, welches sich aus dem gerenderten Modell und den Kamerabildern zusammensetzt. Das führt dazu, dass keine Möglichkeit existiert, die Orientierung der Kamera zu ermitteln, sollte diese einmal verloren gehen. Ein denkbarer Nachteil an MobileFusion könnte sein, dass die Bilder für die Umwandlung auf eine Auflösung von 320x240 Pixel reduziert werden, was zu einem hohen Kontrastverlust führt. Des Weiteren werden bei den Verfahren keine Sensoren verwendet. Da MobileFusion einen ABM-Ansatz verfolgt, sollten bereits kleine Unterschiede zwischen den Frames ausreichend sein, dass keine Gemeinsamkeiten zwischen den Bildern gefunden werden [38].



Abbildung 1.5: Ergebnis einer 3D-Rekonstruktion aus einer Videoaufnahme mit einer Länge von 20 Sekunden durch MobileFusion. Zu finden in der Publikation: [38].

### Live metric 3d reconstruction on mobile phones

In der Publikation wird Live metric 3d reconstruction on mobile phones auf einem Samsung Galaxy SIII I9300GT mit 15-30Hz betrachtet. Innerhalb 2-3 Sekunden werden aus einer Videoaufnahme bei einer Auflösung von 640x480 Pixel eine farbige Punktwolke erstellt. Hierbei kommt sowohl der Accelerometer, als auch das Gyroskop zum Einsatz. Diese Sensoren starten die Berechnungen, sollte sich das Smartphone bewegen. Andere Sensoren wie der Magnetometer oder GPS werden nicht verwendet. Da bei der Videoaufnahme keine großen Strecken zurückgelegt werden, weshalb diese nicht optimal genutzt werden können. ORB wird für die Merkmalerkennung und Merkmalsbeschreibung ausgeführt. Die Merkmale werden anschließend mit RANSAC weiterverarbeitet, um einen Überblick der Umgebung zu erhalten. Zum Erstellen einer dichten Punktwolke werden die Umgebungsinformationen aus der bereits erstellten Punktwolke extrahiert und weitere Merkmale mit dem FAST Algorithmus gesucht.

---

<sup>1</sup> Der Begriff 2,5D Modell bezieht sich hierbei auf 3D-Modelle, bei denen die dritte Dimension nicht vollständig modelliert wird. Hierzu werden 2D-Objekte mit Zusatzinformationen z.B. Verschiebevektor oder Rotationsachse versehen. Über diese Angaben können die 2D-Objekte im 3D-Raum bewegt oder rotiert werden.

Anschließend erfolgen Berechnung zum Erstellen einer Tiefenkarte, welches zum Texturieren der verdichteten Punktwolke gebraucht werden. Nachteilig ist, dass die Punktwolke große Lücken aufweisen (siehe Abbildung 1.6) [49].



Abbildung 1.6: Ergebnis einer Umwandlung der Anwendung Live metric 3d reconstruction on mobile phones der Entwickler. Zu finden in der Publikation: [49].

### Abgrenzung gegenüber bestehender Systeme Dritter

Eine Abgrenzung dieser Arbeit von bestehenden Systemen wird anhand der Funktionen vorgenommen. Die Systeme versuchen alle Berechnungen komplett auf einem System auszuführen. Das führt dazu, dass die Fotos aufgrund begrenzter Systemreserven von mobilen Geräten (Smartphones) erheblich herunterskaliert werden müssen, was häufig die Erstellung kontrastarmer 3D-Modelle zur Folge hat. Da die Systeme meist nur auf einem Referenz-System zentriert entwickelt wurden, gibt es keine Hinweise darauf, ob die Verfahren auf anderen Geräten ausgeführt werden können. Darüber hinaus den Anwender bei der Fotoerstellung nicht unterstützen, sondern aus den Fotos direkt das 3D-Modell konstruieren. Damit dienen solche Systeme vielmehr der Bewertung der unterschiedlichen Rekonstruktionsansätze. Weiter ist ungewiss, wie sich die Systeme verhalten, wenn sie nicht ordnungsgemäß verwendet werden.

Aus dem Vergleich von Systemen verwandter Arbeiten kann entnommen werden, dass unter Einsatz großer Rechenleistung eines Smartphones, die Punktwolken oder 3D-Modelle erstellt werden können. Diese enthalten häufig Lücken und sind undetailliert. Eine höhere Qualität wird erreicht, wenn die Berechnungen auf einen Laptop oder Server ausgelagert werden. Die Gemeinsamkeit ist, dass das Nutzen von Sensoren von besonderer Bedeutung ist, um weitere Informationen zum Objekt zu erhalten. Weiterhin zeigt sich, dass die Kamera vor Beginn der Anwendungen kalibriert werden muss.

Diese Masterarbeit hingegen wird eine Anwendung implementiert, welche auf unterschiedlichen Smartphones ausgeführt werden kann und den Anwendern im Hinblick auf die Bedienbarkeit unterstützt. Dabei wird der Forschungsfrage nach dem Wie der Umsetzung und Erweiterung von bestehenden Algorithmen nachgegangen. Um in Echtzeit unter Auslagerung rechenintensiver Prozesse auf leistungsfähige Systeme eine Punktwolke zu konstruieren.

## 1.6 Aufbau dieser Arbeit

Der weitere Aufbau dieser Arbeit gestaltet sich folgendermaßen: In Kapitel 2 die unterschiedlichen Möglichkeiten analysiert, um ein 3D-Modell aus einer Reihe an Bildern herzustellen. In Kapitel 3 werden die Algorithmen für eine Rekonstruktion untersucht und eine konzeptionelle Realisierung erstellt. Weiter wird eine Konzeption des Prototypen *Automated Realtime 3D Reconstruction on Mobile Devices* (AR3DROS) dargestellt. Kapitel 4 erstellt auf Basis der Konzeption nach dem Feature Driven Development Ansatz eine konzeptionelle Realisierung. Kapitel 5 evaluiert den Prototypen hinsichtlich der Zielumsetzung und ein Belastungstest. Zum Abschluss werden in Kapitel 6 verschiedene Möglichkeiten dargestellt, wie die Anwendung in Zukunft erweitert werden kann.

# 2

## Grundlagen

Zum Erstellen eines 3D-Modells kann auf mehrere Algorithmen zurückgegriffen werden. Jeder Rekonstruktionsalgorithmus hat hierbei seine Limitierungen und muss hinsichtlich auf das Anwendungsszenario ausgewählt werden. Da in der Praxis in den Bildern eine sehr begrenzte Menge an Informationen oder Falschinformationen zu verarbeiten sind. Die Verfahren lassen sich grob in die folgenden Kategorien einteilen.

Das Erstellen der 3D-Modelle über die Abbildung der Bildunterschiede. Hierzu wird die Kamera kalibriert, die Bilder ausgerichtet und anhand der in den Bildern abgebildeten Unterschiede auf Gemeinsamkeiten verglichen (häufig geschieht dies durch das Erstellen einer Differenz-Karte der Bilder oder StereoBM/S-GBM). Der Vorteil an dieser Methode sind die hochwertigen Ergebnisse. Nachteil ist die große Menge an Algorithmen, welche auf die Bilder angepasste Parameter oder manuelle Eingriffe erforderlich machen. Ebenfalls der hohe Aufwand beim Kalibrieren der Kamera.

Die zweite Methode nutzt den optischen Fluss (z.B. Farneback-Verfahren) einer Bildsequenz. Bei diesem werden auf der Bildebene die Bewegungen von Oberflächen, Objekten und Kanten zwischen dem Betrachter und der Szenerie verwendet. Der Vorteil an dieser Methode sind die wenigen Einzelschritte. Nachteilig sind kaum verwertbare Endergebnisse.

Die dritte Technik, welche in dieser Arbeit näher untersucht wird, extrahiert eine Reihe an unsortierten Bildern. Untersucht die Bilder auf identische Kanten und überführt die Bilder anhand ihrer Gemeinsamkeiten in eine Punktwolke und anschließend in ein 3D-Modell.

Dieser Abschnitt befasst sich mit den Grundlagen zur Überführung von Fotos in ein 3D-Modell. Bei Structure from Motion (SfM) handelt es sich um ein Verfahren, bei dem es darum geht Bilder zu strukturieren, in eine Punktwolke zu überführen und anschließend mittels Multi-View Stereo (MVS) zu verdichten. Die erstellte Punktwolke kann abschließend texturiert und vernetzt werden, um ein dichtes 3D-Modell zu erhalten. Es gibt keine genaue Eingrenzung der Verfahren SfM und MVS, wodurch diese häufig synonym verwendet oder von den Autoren neu definiert werden. Der im Folgenden vorgestellte Ablauf wurde [20] abgeleitet. Hierbei werden die Verfahren einzeln in voneinander ge-

trennten Unterkapiteln behandelt. Zusätzlich findet in Abschnitt 2.6 eine Analyse der Smartphone-Sensoren statt und Abschnitt 2.7 geht näher auf mögliche Problematiken während der Fotoaufnahme ein.

### 2.1 Sammeln der Bilder

Bevor die Algorithmen näher untersucht werden, wird auf die Aspekte während der Fotoaufnahme eingegangen, die für eine optimale Überführung von Fotos in ein 3D-Modell eine zentrale Rolle spielen [34]:

- Aufnahmemenge
- Aufnahmewinkel
- Belichtung
- Kontrast
- Objekt- und Bildgröße
- Kontext

#### Aufnahmemenge

Die Qualität der Punktwolke und des 3D-Modells steigt mit der Fotomenge. Die Mindestmenge der benötigten Fotos für ein optimales Ergebnis ist dabei abhängig vom Detailgrad des Objektes und dessen Umgebung, sowie der Auflösung des Aufnahme gerätes. Fotokameras mit wenig Einstellmöglichkeiten, geringen Bildsensoren oder Objektiv (z. B. Smartphones) erstellen oftmals keine hochauflösenden Fotos. Weswegen für eine 3D-Rekonstruktion eine hohe Menge an Fotos des Objektes vorausgesetzt werden.

#### Aufnahmewinkel

Ein für die 3D-Rekonstruktion sehr wichtiger Faktor beim Fotografieren ist der Aufnahmewinkel während der Aufnahme der Fotos. Es gibt drei mögliche Arten, nach denen ein Foto erstellt werden kann [18].

1. Konvergierend: Bei der konvergierenden Aufnahme variiert stetig die Position beim Fotografieren minimal. Empfohlen wird eine Bildüberlappung von 60-80% mit einer maximalen Kameraneigung von  $20^\circ$  rund um das zu fotografierende Objekt. Diese Methode stellt die erfolversprechendsten Fotos für eine Rekonstruktion zur Verfügung [1].
2. Divergent: Bei der divergenten Aufnahmemethode werden die Fotos von einem Standpunkt, mit möglichst wenig oder gar keinen Überlappungen erstellt. Diese Fotos sind für die meisten Rekonstruktionsverfahren nicht verwertbar und stellen von allen Methoden die schlechteste Wahl dar.
3. Parallel: Bei der parallelen Aufnahmemethode wird das Objekt aus unterschiedlichen Standorten mit derselben Blickrichtung erstellt. Die daraus ergebenden Fotos eignen sich nur dazu, zweidimensionale Endergebnisse zu erstellen und neigen dazu, das Endresultat signifikant zu verzerren.

### Beleuchtung

Es ist darauf zu achten, Objekte möglichst kontrastreich darzustellen, da die Rekonstruktionsverfahren bei kontrastarmen Bildern wenig bis gar keine Merkmale finden können. Gut ausgeleuchtete Objekte haben den Vorteil, dass ihnen ein ausgeprägte Farbmuster entnommen und als Textur über das fertig rekonstruierte Modell gelegt werden kann.

### Kontrast

Der Kontrast der Fotos ist ein relevanter Faktor bei der Rekonstruktion. So sollte nicht nur das Objekt, sondern auch der Hintergrund möglichst kontrastreich gehalten sein. Es sollte vermieden werden, Objekte vor einem einfarbigen oder homogenen Hintergrund zu fotografieren, da dieser als Bestandteil des Objektes angesehen wird und zur Erstellung des Modells beiträgt.

### Objekt- und Bildgröße

Bei der Objekt- und Bildergröße soll eine hohe Auflösung priorisiert werden. Dies führt zu umfangreichen Punktwolken und zu Modellen mit einem hohem Detailgrad, da den Anwendungen bei Bildern mit einer hohen Auflösung mehr Pixel zu Verfügung stehen. Was zwar zu längeren Berechnungszeiten, aber auch zu 3D-Modellen mit einer hohen Qualität führt. Zusätzlich wurde in Abschnitt 3.1 festgestellt, dass die Bilder nicht nachträglich verändert werden sollen, da dies zu hohen Konturverlusten führt.

### Kontext

Anhand der eigenen Analyseergebnisse lässt sich herleiten, dass die zu fotografierenden Objekte sich auf unterschiedlichste Weise aufnehmen lassen, wodurch folgende Szenarien bei der Fotoerstellung zu unterscheiden sind:

1. Einfach zuzuordnende Objekte: Auf den abfotografierten Objekten wird stets dasselbe Objekt, mit demselben unbeweglichen Hintergrund dargestellt. Beides ist auf jedem Bild gut sichtbar, mit einem hohen Kontrast. Typisch für diese Kategorie sind Skulpturen und fest verankerte Objekte.
2. Schwer zuzuordnende Objekte: Sind identisch zu den einfach zuzuordnen Objekten, mit der Ausnahme, dass auf den Bildern mehrere Objekte fotografiert werden, sodass das Objekt nicht eindeutig erkennbar ist. Beispiel hierfür sind Gebäude und nah beieinander befindliche Sehenswürdigkeiten.
3. Gar nicht zuzuordnende Objekte: Dieser Fall ist gegeben, wenn sich die Objekte bewegen oder stetig unterschiedliche Objekte fotografiert werden, sodass die Bilder nicht einander zugeordnet werden können. Mögliche Beispiele sind bewegende Fahrzeuge oder Lebewesen.

Die Fotos sollten möglichst dem Best-Case entsprechen, also aus einfach zuzuordnenden Objekten bestehen. Mit Techniken wie dem Auslesen der GPS-Geolokalisationsdaten können nicht zuzuordnende Objekte weitestgehend eingeschränkt werden. Lediglich schwer zuzuordnenden Objekte stellen eine große Schwierigkeit für die Auswertung der Bilder dar. Aus diesem Grund werden in dieser Arbeit nur einfache und gar nicht zuzuordnende Objekte behandeln.

## 2.2 Kamerakalibrierung

Für die Erstellung einer Punktwolke wird zwischen den zwei Verfahren SfM und *Visual Simultaneous Localization and Mapping* (VSLAM) unterschieden. Während SfM bei unsortierten Bildern zum Einsatz kommt, ist VSLAM spezialisiert auf die Berechnung der Parameter aus Videos in Echtzeit. In dieser Ausarbeitung wird nur auf SfM eingegangen, da es sich bei der Eingangsmenge ausschließlich um Bilder handelt.

SfM benötigt für seine Berechnungen Kamerainformationen während der Fotoaufnahme, welche beschreiben, wie sich die 2D-Bildpunkte aus den 3D-Punkten der fotografierten Umgebung zusammensetzen. Das am häufigsten verwendete Model beim SfM Verfahren ist das 'pinhole camera model', welches aus einer 3x4 projektionsmatrix besteht und sich in eine intrinsische 3x3 Matrix  $K$  (siehe (2.2)) und einer 3x4 Matrix bestehend aus extrinsischen Kameraparametern  $[R \mid T]$  aufteilen lässt (siehe (2.1)).  $K$  beschreibt die intrinsischen Kameraparameter und setzt sich aus der vertikalen und horizontalen Fokallänge  $(f_x, f_y)$ , Hauptpunkt  $(c_x, c_y)$  und Abweichung  $s$  zusammen.

$$[R \mid T] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & | & t_x \\ r_{21} & r_{22} & r_{23} & | & t_y \\ r_{31} & r_{32} & r_{33} & | & t_z \end{bmatrix} \quad (2.1)$$

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Das Zentrum der Kamera (optische Zentrum) liegt im Ursprung eines 3-dimensionalen Koordinatensystems (siehe Abbildung 2.1 links), wobei der Abstand der Bildweite (Fokallänge) die Entfernung zwischen dem optischen Zentrum und der Bildebene beschreibt. Auf der Bildebene wird die 3-dimensionalen Welt 2-dimensionalen abgebildet. Die Bildebene wird dabei stets parallel zur  $xy$ -Ebene ausgerichtet. Das Zentrum der Kamera, auf dem der Strahl die Bildebene durchstößt, liegt auf der Position  $(c_x, c_y, 1)^T$  (siehe Abbildung 2.1 rechts) auf der Ebene. Die Projektion  $c$  eines beliebigen 3D-Punktes auf der Bildebene ergibt sich als Schnittpunkt des Richtungstrahls von der Kamera zu  $C$ . Weiter wird zur Berechnung der Abweichung der Achsen der Bildebene noch der Parameter  $s$  (Skew) verwendet. Insofern das Bild nicht zugeschnitten wurde, beträgt der Wert für skew 0.

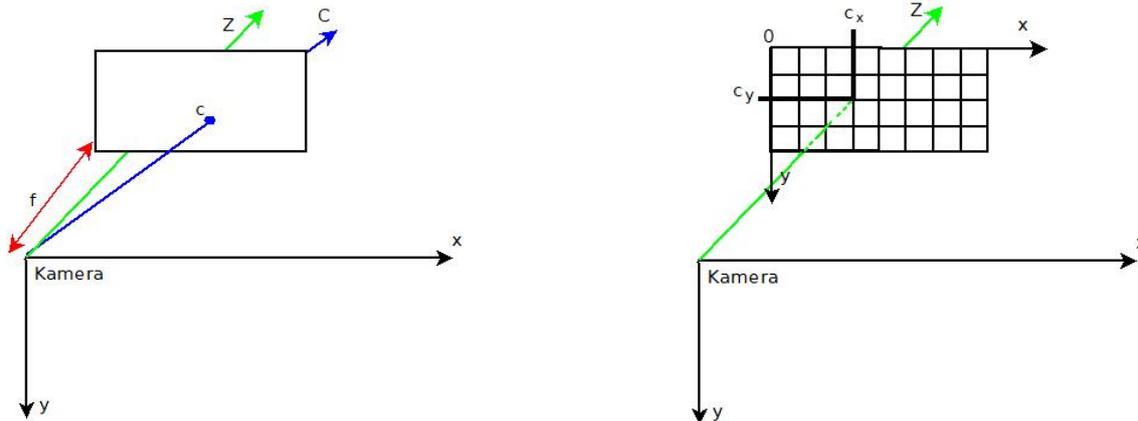


Abbildung 2.1: Links: Visualisierung des Kameramodells, Rechts: Koordinatensystem der Bildebene. Beide Abbildungen wurden in Anlehnung an [24] erstellt.

Beim bisherigen Modell wird nicht berücksichtigt, dass die Koordinatenachsen der Bildebene in der Regel unterschiedlich und unabhängig vom dreidimensionalen Koordinatensystem der Kamera skaliert sind. Um dies zu korrigieren, müssen die Rotations- ( $r$ ) und Skalierungsfaktoren ( $t_x$ ,  $t_y$  und  $t_z$ ) auf der x, y und z Achse berücksichtigt werden. Die Skalierungsfaktoren werden aus dem errechneten CCD-Array aus der Bildaufnahme entnommen und setzen sich aus der Anzahl der CCD-Elemente pro Einheitslänge im 3-dimensionalen Koordinatensystem der Kamera zusammen [24].

## 2.3 Erstellen einer Punktwolke

SfM erhält eine Menge an Bildern und Kamerainformationen, welche dazu genutzt werden, eindeutige Merkmale in den Bildern zu suchen und in 3D-Punkte umzuformen. Der weitverbreitetste Vorgang des SfM Algorithmus lässt sich in folgende Teilschritte zusammenfassen:

1. Bestimmen und beschreiben der Merkmale.
2. Vergleichen der Merkmale.
3. Erstellen einer Punktwolke aus den verglichenen Merkmalen.
4. Verfeinern der Punktwolke.

Features, lokale Merkmale oder Keypoints sind Eigenschaften eines Bildes und helfen dabei, Objekte unabhängig von ihrem Blickwinkel, Form, Helligkeit, Auflösung oder anderen Faktoren eindeutig wiederzufinden. Es gibt 5 Eigenschaften, die beim Arbeiten mit Merkmalen zu beachten sind [25]:

1. Robustheit: Je mehr veränderbare Bildeigenschaften abgedeckt werden können, desto höher ist die Robustheit der Merkmale.

2. Wiederholbarkeit: Dieser Wert sagt aus, ob sich die Ergebnisse replizieren lassen.
3. Generalität: Misst die Verwendbarkeit der gefundenen Merkmale.
4. Geschwindigkeit: Ist die Bearbeitungsdauer und der Ressourcenverbrauch.
5. Qualität: Gebrauchbarkeit der Merkmale.

### Merkmalerkennung

Bei der Verarbeitung der Bilder ist es notwendig, möglichst genaue und unterscheidbare Repräsentationen zu finden. Eine mögliche Repräsentation ist die Beschreibung der Merkmale anhand ihrer globalen oder lokalen Merkmale. Über die globalen Merkmale wird das Bild mithilfe eines Histogramms oder durch Informationen aus den Filterfunktionen beschrieben. Dies hat den Vorteil, dass die Merkmale schnell verarbeitet werden können. Die Merkmale sind jedoch uneindeutig und häufig unbrauchbar. Eine Beschreibung über die lokalen Merkmale geschieht hierbei über die im Bild befindlichen Strukturen, bei denen die Merkmale eindeutig von ihren benachbarten Merkmalen abgegrenzt werden. Dies benötigt zwar eine längere Bearbeitungszeit der Prozesse, führt jedoch zu eindeutigeren Merkmalen. Aufgrund des Qualitätsvorteils werden in dieser Arbeit ausschließlich lokale Merkmale betrachtet.

Merkmalerkennung (englisch: Feature Detection) ist eine Technik, bei der Bilder nach markanten Merkmalen untersucht werden. Merkmale können hierbei aus Ecken oder Blobs bestehen. Bei der Ecken-Suche werden Überschneidungen eines oder mehrere Kanten gesucht. Als Beispiel kann Abbildung 2.2 herangezogen werden. Der blaue Rahmen kann beliebig verschoben werden, wodurch dieser nicht eindeutig wiedergefunden werden kann. Der schwarz umrandete Rahmen ist nicht wiederzufinden, sollte dieser vertikal verschoben werden. Nur die Ecken, also der rote Rahmen, lässt sich eindeutig wiederfinden, solange das Bild nicht rotiert wird. Beim *Blob-Verfahren* wird jedes Bildmuster bezeichnet, bei dem andere Faktoren als die Ecken eine Rolle spielen. Dabei können die Eigenschaften der Nachbarn, die eigene Intensität oder Textur genutzt werden. Der Unterschied der beiden Verfahren ist, dass über der Ecken-Suche eine wesentlich höhere Lokalisationsgenauigkeit der Artefakte erreicht wird, wohingegen das Blob Verfahren besser darin ist, die gefundenen Artefakte unverwechselbar wiederzufinden [20, 36].

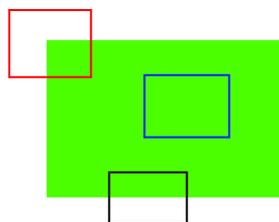


Abbildung 2.2: Mögliche Merkmale eines Objektes beim Einsatz der Ecken-Suche. Quelle des Bildes: [36].

## Merkmalsbeschreibung

Merkmalsbeschreibung (englisch: Feature Description) dient dazu, die Länge und Orientierung der Gradienten aus der Umgebung der gefundenen Merkmale eindeutig in einer Vektorstruktur zu speichern. Nachdem diese mittels Merkmalerkennung gefunden und durch die Merkmalsextraktion aus den Bildern entnommen wurden. Informationen werden entweder in reellen Werten (Fließkommazahlen) oder zwei Werten (Bitstrings) gespeichert. Aufgrund der Einschränkung beim Bitstring die Merkmale in zwei Werten zu speichern, besitzen diese beim Matching zwar eine schnellere Berechnungszeit, können jedoch schwerer voneinander unterschieden werden.

## Merkmalsvergleich

Beim Matching werden die Merkmale unterschiedlicher Bilder anhand ihrer Beschreibungen einander zugewiesen. Für das Verfahren kann auf zwei Ansätze zurückgegriffen werden:

Das *feature-tracking*, bei dem die Überschneidungen der Merkmale aus einem Bild mit einem entsprechend großen Fenster auf weiteren Bildern gesucht werden. Der Nachteil an dieser Methode ist, dass sie nicht gut mit veränderbaren Merkmalen verfährt. Weshalb diese üblicherweise nur dann verwendet wird, wenn die Bildunterschiede sehr gering sind.

Zweite *Matching-Methode* ist der Merkmalsvergleich (englisch: Feature Matching), bei dem die Merkmale aus den Bildern extrahiert und miteinander verglichen werden. In dieser Ausarbeitung wird der Merkmalsvergleich näher untersucht, da größere Bildunterschiede bei der manuellen Bildaufnahme nicht umgangen werden können. Beim Merkmalsvergleich wird der Abstand der Merkmale zueinander im n-dimensionalen Raum berechnet. Der geringste Abstand, den ein Merkmal zu einem anderen Merkmal besitzt, entspricht dem *Match*. Vektoren bestehend aus Fließkommazahlen werden mit dem euklidischen Abstand  $d_{euklid}$  (siehe (2.3)) und Vektoren bestehend aus Bitstrings werden anhand ihrer Hammingdistanz  $d_{hamming}$  (siehe (2.4)) miteinander verglichen [14].

$$d_{euklid}(x, y) = \|x - y\|_2 = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.3)$$

$$d_{hamming}(x, y) = \sum_{i=1}^n (a_i \oplus b_i) \quad (2.4)$$

OpenCV stellt zwei Merkmalvergleichs-Verfahren zur Verfügung [16]:

*BruteForce*, bei dem jede Kombination der Merkmale anhand ihres Abstandes miteinander verglichen wird. Das zweite Verfahren ist der *Fast Library for Approximate Nearest Neighbors* (FLANN). FLANN enthält eine große Ansammlung an Algorithmen, die das Ziel haben, hohe Mengen an Merkmalen in sehr kurzer Zeit miteinander zu vergleichen.

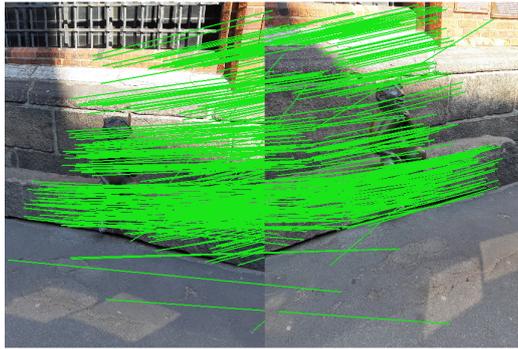


Abbildung 2.3: Gefundene Matching beim BruteForce-Verfahren.

Die von den Matching-Mechanismen generierten Gemeinsamkeiten (in Abbildung 2.3 gekennzeichnet durch die grün gefärbten Kanten) können für Genauigkeitstests in die vier Arten True Positive, False Positive, True Negative und False Negative klassifiziert werden [48].

- True Positive: Richtige Matches, die vom Verfahren als richtig erkannt werden.
- False Positive: Falsche Matches, die vom Verfahren als richtig erkannt werden.
- True Negative: Falsche Matches, die vom Verfahren als falsch erkannt werden.
- False Negative: Richtige Matches, die vom Verfahren als falsch erkannt werden.

Hierbei entsteht das Problem, dass sich die Matches aus den True Positive und False Positive Matches zusammensetzen können. Um die Qualität der gefundenen Matches zu verbessern, kann das K-Nearest-Neighbour (KNN)-Verfahren angewendet werden. Anstatt die Punkte mit der kleinsten Abstandsdistanz zusammenzusetzen, werden beim KNN-Verfahren zu jedem gefundenen Merkmal die  $k$  besten Matches aus der resultierenden Menge aller Matches zurückgeliefert. Auf die Reduktionsalgorithmen angewendet werden können. Um die False-Positive Matches aus der Menge der Matches herauszufiltern, wird beim KNN Verfahren der Ratio-Test nach Lowe verwendet [33]. Der Ratio-Test untersucht das Distanzverhältnis der  $k$  gefundenen Merkmalpaare zum Merkmal  $n$  und entnimmt das Match  $n$ , falls gilt:

$$n.distance < r * k.distance \quad (2.5)$$

Für den Wert  $r$  schlägt Lowe einen Wert höher oder gleich 0.8 vor. Ab diesem Wert werden im Schnitt 90% aller False Positive Matches und weniger als 5% der True-Positive Matches entfernt. In der Anwendung wird  $r=0.8$  verwendet.

Nach dem der Ratio-Test ausgeführt wurde, kann auf eine Auswahl an True Positive Matches zurückgegriffen werden. In dieser Arbeit wird zusätzlich der *random sample consensus* (RANSAC)-Algorithmus angewendet, um Ausreißer und grobe Fehler zu entfernen. Um also eine korrekte Abschätzung zu erstellen, wird aus einem Ausreißer behaftetem Modell durch ein iteratives Verfahren kontinuierlich neue Modelle aus einem mini-

malen gewählten Datensatz erzeugt. Der Hauptkritikpunkt an RANSAC ist die hohe Anzahl an Iterationen  $I$ , die benötigt werden, um mit einer Wahrscheinlichkeit  $p$  ein korrektes Modell zu finden. Die Anzahl der Iterationen hängt ebenfalls vom Anteil der korrekten Bildpunkte  $a$  und der Gesamtgröße  $s$  ab. Aus den gegebenen Größen lässt sich folgende Laufzeit herleiten:

$$(1 - a^s)^I = 1 - p \quad (2.6)$$

Zum Vergleich zweier Bilder können die gegebenen Matches über eine Transformationsmatrix  $H$  zweier Bereiche errechnen werden, sodass für alle Punkte  $x^{-1}$  aus dem Teilbild und den Punkten  $\overrightarrow{x^{-1}}$  aus dem Ursprungsbild gilt:

$$\overrightarrow{x^{-1}} = H * x^{-1} \quad (2.7)$$

In Abbildung 2.4 ist die Transformierung des Ursprungsbildes auf die vier Eckpunkte des Teilbildes auf der rechten Szene sichtbar. Die gefundene Region wird hierbei mit einem grünen Rahmen kenntlich gemacht [19].

Sobald die Matches der Bilder gefunden wurden, kann damit begonnen werden, einen Konnektivitätsgraphen zu konstruieren, indem Bildpaare gruppiert gespeichert werden. Hierbei erhält jedes Bild einen Knoten und Bilder zu denen eine ausreichend große Menge an Matches besteht, werden mit einer Kante miteinander verbunden.



Abbildung 2.4: Ergebnis vom RANSAC-Algorithmus zum Entfernen von Ausreißern und die daraus abgeleitete Transformationsmatrix.

### Erstellen einer Punktwolke aus den verglichenen Merkmalen

Sowie der Konnektivitätsgraph erstellt wurde, werden die Kamerapositionen und korrespondierende Merkmale einer Bildsequenz im dreidimensionalen Raum bestimmt. Dieses Verfahren ist bekannt als das *SfM Problem*, bei dem mittels Triangulation eine Konfiguration der Kameras und 3D-Punkten gesucht wird, sodass die Projektion jedes beliebigen Punktes aus dem 3D-Raum in ein von der konstruierten Kamera aufgenommenen Bildes möglichst nah beieinander liegt [47].

## 2 Grundlagen

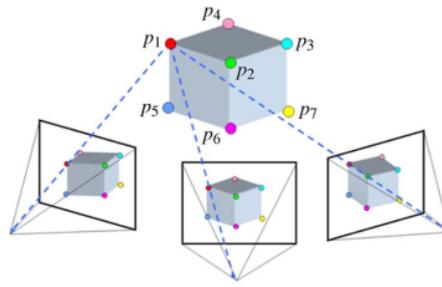


Abbildung 2.5: Grundidee des Structure from Motion Verfahrens. Quelle des Bildes: [47].

### Bundle adjustment

Nachdem die Punktwolke konstruiert wurde, kann optional das Bundle adjustment eingesetzt werden, welches die Kamera und Punktabbildungen nutzt, um die Punktwolke zu optimieren, sodass verbleibende Fehler (z. B. aus Bildverzerrungen) minimiert werden. In der Anwendung wird das Bundle adjustment in den Verfahren Meshroom und OpenMVG (siehe Abschnitt 4.1) ausgeführt.

### Verdichten der Punktwolke

Das MVS Verfahren erhöht die Anzahl der gefundenen Bildpunkte und verdichtet die Punktwolke. Hierzu werden die Kameraparameter und Bildpunktzuweisungen vom SfM-Verfahren übernommen. Dadurch kann ein Verfahren angewendet werden, bei dem es darum geht von jedem Pixel eines Bildes einen dreidimensionalen optischen Strahl zum gegenüberliegenden Bild zu projizieren. Anhand des Projektionsstrahls werden die benachbarten Pixel der gegenüberliegenden Bilder auf weitere Gemeinsamkeiten untersucht, welche als 3D-Punkte in die Punktwolke übertragen werden [20].

## 2.4 Texturieren der Punktwolke

Nach der Fertigstellung der Punktwolke, kann damit begonnen werden, diese zu vernetzen. Laut [27, 29, 53] eignet sich das *Poisson Surface Reconstruction* am besten (siehe Abbildung 2.6 für eine Übersicht dieses Vorgangs), wenn es darum geht, eine dichte und glatte Vernetzung der Punktwolke herzustellen. Weiter ist es resistent gegenüber einer geringen Menge an Ausreißern. Die Ausgangspunktwolke besteht aus nach innen zeigenden Normalen (Oriented points), welche die Fläche der darzustellenden Figur begrenzen. Mittels der berechneten Normalfelder wird der *Gradient* (Indicator gradient) für die *Indikatorfunktion* (Indicator function) berechnet. Durch die Indikatorfunktion werden die Punkte innerhalb des Gradienten der Wert 1 und außerhalb der Wert 0 zugewiesen. Alle Flächen mit einem Wert von 0 werden entfernt und die Oberfläche (surface) entnommen [27].

Auf der Oberfläche wird ein Marching-Cube Algorithmus ausgeführt (siehe Abbildung 2.7) und die erstellte Oberfläche in eine Octree-Struktur<sup>2</sup> umgewandelt. Hierzu wird die Punktwolke in Voxelgitter zerlegt und entschieden, welche Punkte für die Modell-Isofläche des Objektes verwendet werden kann. Die Schnittflächen des Voxelgitters und der Modell-Isofläche bilden hierbei die Triangulierung des 3D-Modells [28]. Auf den aufgespannten Polygonen wird anschließend entweder ein Farbpartikelnetz oder eine Textur gelegt. Beim Farbpartikelnetz werden die Farben des Polygons aus den zugeordneten Bildern abgeleitet. Wird eine Textur benötigt, so wird die Lokalisierung des Polygons auf den Bildern gesucht. Sollten mehrere Bilder dem Polygon zugeordnet werden können, so wird der Bildabschnitt mit dem geringsten Blickwinkel entnommen. Nach diesem Prozess ist das Modell als zusammengesetztes 3D-Modell erkennbar.

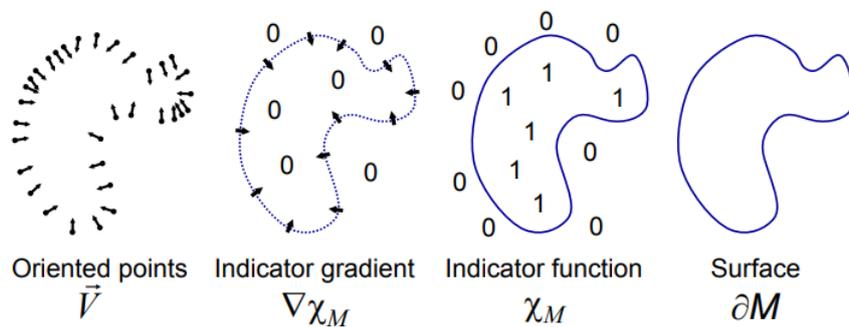


Abbildung 2.6: Übersicht über den Ablauf des Poisson Surface Reconstruction Verfahren. Quelle des Bildes: [27].

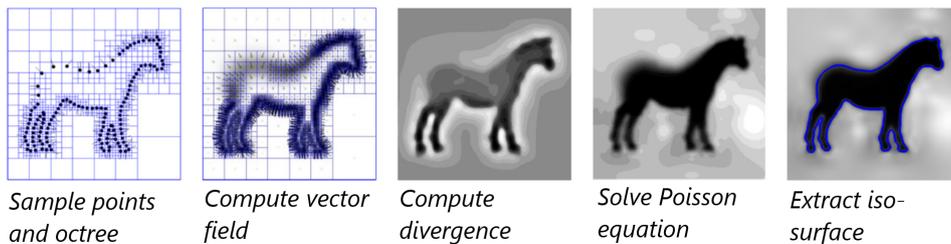


Abbildung 2.7: Übersicht des Ablaufs des Marching-Cube Algorithmus. Quelle des Bildes: [30].

<sup>2</sup> Beim Octree handelt es sich um Bäume, deren Knoten entweder 8 Nachfolger oder keine Nachfolger besitzen

## 2.5 Verfeinern des 3D-Modells

Zum Verfeinern des 3D-Modells stehen eine Vielzahl an Anwendungen zur Verfügung. *MeshLab* ist eine frei erhältliche und ständig weiterentwickelte Software zum Betrachten, Konvertieren und Bearbeiten von unstrukturierten 3D-Meshes. Dem System steht eine große Sammlung an Algorithmen zum Bearbeiten von Modellen zur Verfügung. Innerhalb von MeshLab können diverse Ausführungsschritte in Form von Skripten zum Bearbeiten der Punktwolke und 3D-Modellen ausgeführt werden.

## 2.6 Smartphone Sensoren

In dieser Arbeit werden Smartphones verwendet, welche auf unterschiedliche Sensoren zurückgreifen können. Nachfolgend werden die für diese Arbeit relevanten Sensoren betrachtet, welche in marktüblichen Smartphones eingebaut sind.

1. **Barometer:** Wird dazu genutzt den Luftdruck der Umgebung zu messen, um z. B. die Höhe des Smartphones zu bestimmen. In Kombination mit den Mobilfunkantennen, den WLAN-Netzen und GPS-Satelliten kann der exakte Standort des Smartphones ermittelt werden. Die Höhenangaben des Barometers sind unpräzise [45], welches sich während der eigenen Analyse bestätigt hat. Weshalb der Barometer in dieser Arbeit nicht verwendet wird.
2. **Beschleunigungssensor:** Dienen dazu die Beschleunigung des Smartphones in allen Richtungen zu messen. Der Beschleunigungssensor wird in dieser Arbeit dazu genutzt, Berechnungen auf dem Smartphone nur auszuführen, sollte sich der Anwender bewegen. Dadurch wird das Smartphone entlastet und sorgt für eine flüssige Verwendung.
3. **Globales Positionsbestimmungssystem (GPS):** Nahezu jedes Smartphone besitzt heute eine GPS-Funktion, mit der die Positionierung des Gerätes auf bis zu 5 Meter genau ortbar ist [5]. In dieser Arbeit wird, das GPS Verfahren dazu verwendet zu überprüfen, ob sich der Nutzer weiter als 8 Meter vom Objekt entfernt hat. Unter anderem hat dies den Vorteil, dass der Vorgang zur Fotoerstellung abgebrochen werden kann, falls die Entfernung zwischen Objekt und Anwender zu groß ist. Es kann vorkommen, dass das Smartphone trotz aktivierter GPS-Funktionalität die Positionierung des Smartphones nicht bestimmen kann. Dies hätte den Nachteil, dass das zu modellierende Objekt trotz GPS Unterstützung aus einer größeren Entfernung als 8 Meter fotografiert wird.
4. **Gyroskop:** Mit einer Kombination aus der Corioliskraft und dem Stimmgabelprinzip erkennt das Smartphone, wie seine vier Ecken sich im Verhältnis zueinander im Raum befinden. In dieser Ausarbeitung wird dies dazu genutzt, um zu prüfen, ob ein Bild im Hochformat fotografiert wurde und verwirft Fotos im Querformat. Grund dafür ist, dass unterschiedliche Darstellungsformate einen negativen Einfluss auf die Reduktion besitzen.
5. **Magnetfeldsensor :** Kann ermitteln, wo die vier Himmelsrichtungen liegen. Damit kann ermittelt werden, in welcher Richtung ein Smartphone bewegt oder gehalten

wird. In dieser Arbeit wird es dazu verwendet, den Anwender darauf hinzuweisen wo noch Fotos für ein lückenloses 3D-Modell fehlen.

6. **Umgebungslichtsensor:** Über den Umgebungslichtsensor wird die Helligkeit der aktuellen Umgebung gemessen und anhand der Messung die Display-Helligkeit des Bildschirms angepasst. Über diesen Wert könnte bestimmt werden, ob ein Foto in einer hellen Umgebung erstellt wurde und somit ob es für eine Umwandlung in Frage kommt. In dieser Ausarbeitung wurde keine Möglichkeit gefunden, den Umgebungslichtsensor optimal in die Anwendung einzubauen. So hat der Sensor häufig falsche Werte ausgegeben, weshalb auf eine Anbindung verzichtet wurde.
7. **3D-Kamera:** Smartphones mit einer 3D-Kamera strahlen ein Infrarotlicht aus, um den Abstand zwischen Smartphone und Objekt zu messen. Mittels 3D-Kamera wären weitere Analysen des Objektes und eine darauf aufbauende Umwandlung möglich. Aufgrund dessen, dass diese Hardware noch nicht in den marktüblichen Smartphones verbaut ist. Dies grundlegende andere, bislang nicht erprobte Rekonstruktionsverfahren benötigen würde, welche nicht im Fokus dieser Arbeit liegen, wird der Sensor in dieser Arbeit nicht weiter behandelt.

### 2.7 Problem-, Kontext- und Aufgabenanalyse

In dieser Arbeit soll eine Anwendung entwickelt werden, welche in Echtzeit aus Fotos eine Punktwolke herstellt, automatisiert ein 3D-Modell konstruieren kann und dem Benutzer Hilfestellung während der Fotoaufnahme liefert. Hierzu werden die verschiedenen Komplikationen und Eingabemöglichkeiten der potenziellen Benutzer während der Fotoaufnahme und beim Überführen der Fotos in ein 3D-Modell untersucht. Primär wird sich in dieser Arbeit auf die technische Umsetzung konzentriert. Die Benutzungsschnittstelle dient hierbei lediglich zur Betrachtung einer möglichen Benutzung und der Sicherstellung ausreichender qualitativer Parameter für die 3D-Rekonstruktion. Zur Veranschaulichung des Einsatzzwecks des Prototyps dient das nachstehende Szenario.

#### Szenario

Sarah Albert ist 27 Jahre alt und arbeitet in einem Museum. So einige Dinge im Museum bereiten ihr Sorgen. Darunter zählen u.a. antike Denkmäler, die im Laufe der Zeit beschädigt wurden und bei denen eine Restauration nur mit hohem Kostenaufwand möglich ist. Nach kurzer Recherche hat Sarah eine Lösung zu ihrem Problem gefunden. Eine Applikation, die am Smartphone gestartet werden kann und selbst Benutzern wie ihr ohne weitergehende Erfahrung im Bereich Computer Vision, die Möglichkeit gibt 3D Modell zu konstruieren. Kurz nachdem Sarah das Programm gestartet hat, wird sie gebeten, das Objekt zu fotografieren. Sie fotografiert das Objekt und als sie versucht ein zweites Foto von dem Objekt zu erstellen, werden Markierungen auf ihrem Gerät, an denen sie erkennen kann, wie sie die weiteren Fotos zu erstellen hat, um ein qualitatives 3D-Modell im Anschluss zu erhalten. Nachdem Sie etwa fünf Fotos auf diese Weise erstellt hat, wird ihr eine Punktwolke eingeblendet, an der sie die Struktur des resultierenden Modells erahnen kann. Aus der Einführung der Anwendung weiß Sarah, dass es sich hierbei nicht um das Endresultat handelt, sondern lediglich um ein vorübergehendes Modell, welches sie bei der Fotoaufnahme unterstützt.

Weiterhin kann sie auf dem Bildschirm erkennen, an welcher Position sie zu wenig Fotos gemacht hat. Nachdem Sarah die fehlenden Fotos erstellt hat und zufrieden mit der dichten Punktwolke ist, betätigt sie die Eingabe über einen Button. Nachdem der Button betätigt wurde, wird am Server ein 3D-Modell aus den übertragenen Fotos erstellt und mit dem Account von Sarah verknüpft. Nachdem der Server das Modell vollständig konstruiert hat, kann Sarah das Modell betrachten. Statt nun das echte Objekt in die Ausstellung zu bringen, kann Sarah dieses ihren Besuchern virtuell und über das Smartphone zeigen.

### Benutzeranalyse

Der Fokus dieser Arbeit liegt in der technischen Umsetzung und dem Vergleich von Rekonstruktions-Algorithmen. Weshalb die potenziellen Benutzer nur am Rande betrachtet werden. Sozusagen als Mittel zum Zweck, weil es zum Bedienen der Smartphone-Anwendung einer grafischen Benutzungsschnittstelle bedarf. Für die Entwicklung der Smartphone-Anwendung werden einige der technischen Bedürfnisse der zu erwartenden Benutzer untersucht. Daraus ergibt sich wie die technischen Entscheidungen bei der Systemkonzeption und die Benutzerschnittstelle auszusehen hat, um eine Benutzbarkeit im Alltag zu gewährleisten. Als Benutzerklasse für den Prototypen werden Menschen im Industriellen Bereich betrachtet, weil diese im Kontext des Endproduktes am größten vertreten sind. Aufgrund dessen dass die 3D-Modelle nachträglich weiterverarbeitet werden, kann davon ausgegangen werden dass die Anwender eine Affinität zu technischen Systemen besitzen. Deshalb wird im Folgenden davon ausgegangen, dass die Anwender problemlos ein Smartphone verwenden können oder den Willen besitzen die Verwendung zu erlernen. Um den Anwendern einen einfachen Einstieg zu ermöglichen, sollten komplexe Begriffe vermieden werden, da nur wenige Benutzer sich direkt mit den 3D-Rekonstruktion auseinandergesetzt haben sollten. Ebenso sollten einfache und verständliche Bedienelemente verwendet werden, um den Einstieg in die Smartphone-Anwendung möglichst einfach zu gestalten. Sollten selbst Personen ohne Affinität zu technischen Systemen als potenziellen Benutzer des Systems angestrebt werden, so muss die Anwendung selbsterklärend sein und nur minimale Kenntnisse mit Smartphones voraussetzen.

### Aufgabenanalyse

Aufgabenanalyse dient dazu zu untersuchen, welche Tätigkeiten die Benutzer mit dem System ausführen können. Benutzer der Smartphone-Anwendung können ein Objekt in ein 3D-Modell überführen. Hierzu wird von den Benutzern erwartet, dass sie ein Objekt mit einer möglichst hohen Anzahl an Merkmalen aus allen Richtungen fotografieren. Hierbei gibt es einige wichtige Funktionen, welche die Anwendung unterstützen muss. Ein Benutzer muss in der Lage sein zu erkennen, ob ein Objekt die Anforderungen aus Abschnitt 2.1 erfüllt und sich somit für eine Umwandlung eignet. Es muss ihn bei der Fototerstellung unterstützen und fehlerhafte Fotos (z. B. verschwommene oder unscharfe) aus der Fotomenge entfernen, sodass diese die Endqualität des 3D-Modells nicht verschlechtern. So sollen auch Anwender ohne viel Erfahrung mit Smartphones oder der 3D-Rekonstruktion die Verwendung ermöglicht werden. Hierfür wird ein Tutorial benötigt, in welchem die Funktionalität der Anwendung erläutert wird.

Um Anwender aktiv während der Verwendung zu unterstützen werden Information während der Bedienung benötigt, wie z. B. der noch fehlenden Menge an Fotos oder Fehlermeldungen. Durch die Rückmeldungen erhalten unerfahrene Benutzer einen überschaubaren und einfachen Einstieg in das System. Ein weiterer Anspruch, dass das System erfüllen muss ist, dass es selbst bei Unterbrechungen der Internetverbindung funktionieren soll. Dies ist notwendig, da eine konstante Internetverbindung auf mobilen Geräten nicht gewährleistet werden kann.

### Kontextanalyse

In diesem Abschnitt wird untersucht, in welchem räumlichen und zeitlichem Kontext die Anwendung genutzt werden kann. Bei der Anwendung handelt es sich um eine Mobile App, womit diese nur auf Systemen mit einem mobilen Betriebssystem ausführbar ist. Aufgrund dessen das in dieser Masterarbeit Fotos von einem Objekt erstellt werden müssen, wird eine an das System angeschlossene und von der Anwendung ansprechbare Kamera benötigt. Selbst in öffentlichen Orten wie im beschriebenen Szenario kann die Anwendung auf mobilen Geräten leicht getragen und die Touch-Eingabe eines Smartphones komplikationslos verwendet werden. Eine Ausführung der Anwendung kann sowohl in Gebäuden als auch außerhalb geschehen. Dabei kann eine räumliche Bewegungseinschränkung vorliegen, sodass die Anwender in ihrer Bewegungsfreiheit eingeschränkt sind. Hinzu kommt die Möglichkeit, dass Störfaktoren in Form von Bewegungen im Hintergrund, Lärm oder Ablenkungen durch Dritte auftreten können. Weshalb eine Selektierung der Fotos vor der Überführung dieser in ein 3D-Modell ausgeführt werden muss. An beiden Orten kann es möglich sein eine Netzwerk- und Internetverbindung aufzubauen, wobei gelegentliche Verbindungsabbrüche auftreten können. Da es sich bei der Zielplattform um Smartphones handelt, kann das Datenvolumen auf Basis der Verträge beschränkt sein. Weshalb die Menge und Größe, der unterwegs zu ladenden Informationen komprimiert werden muss, um keine zusätzlichen Kosten für die Anwender zu generieren.

### Anforderungsanalyse

Aus der Aufgabenanalyse und Kontextanalyse können die folgenden Anforderungen an das System abgeleitet werden.

1. **Einführung:** Vor der Verwendung soll der Benutzer eine Einweisung in den Umgang erhalten.
2. **Bildüberlappung:** Für eine erfolgreiche Rekonstruktion ist eine hohe Anzahl zueinanderpassende Matches notwendig. Von Relevanz dafür ist, dass zwischen den Bildern eine Bildüberlappung von 60-80% mit einer maximalen Kameraneigung von 20° rund um das zu fotografierende Objekt besteht [1].
3. **Datenformate:** Bei einer Erweiterung des Systems können unter Umständen Formate verwendet werden, die nicht dem von der Anwendung unterstütztem Format entsprechen. Dazu müsste eine Konvertierung innerhalb der Anwendung möglich sein.
4. **Datenschutz:** Für die Überführung in ein 3D-Modell muss die Speicherung und Verwaltung der Daten möglich sein. Dabei darf die Datenspeicherung nur unter Beach-

tung der datenschutzrechtlichen Bestimmung erfolgen, sodass der Urheber eines Mediums der Rechteinhaber bleibt und personenbezogene Daten geschützt werden.

5. **Fehlerbeseitigung:** Bei der Fotoaufnahme können fehlerhafte Bilder (z. B. verschwommene oder unscharfe) gemacht werden, welche das Ergebnis signifikant verschlechtern. Diese müssen erkannt und gefiltert werden.
6. **Komprimierung der Daten:** Um das Datenvolumen der Smartphones möglichst gering zu halten, sollen die zu ladenden Informationen komprimiert werden.
7. **Menge der Fotos:** Bei der 3D-Modellierung von Objekten ist es nicht ersichtlich, wie viele Fotos gemacht werden müssen. Sollte der Anwender zu wenig Fotos erstellen, so ist das 3D-Modell undetailliert oder es wird keines erstellt. Falls die Fotomenge zu hoch sein sollte, so können unerwünschte Modelle aufgrund von Bildinformationen aus qualitativ minderwertigen Fotos konstruiert werden. Ein visuelles Feedback wird folglich benötigt.
8. **Modellierbarkeit eines Objektes:** Ohne Kenntnisse des Rekonstruktionsprozesses ist es kaum ersichtlich ob sich ein Objekt für eine Umwandlung eignet. Hierzu müssten die Voraussetzungen aus Abschnitt 2.1 erfüllt sein.
9. **Technische Anforderungen:** Die Smartphone-Anwendung soll plattformunabhängig mit minimalem technologischem Aufwand verwendbar sein.
10. **Verbindungsabbrüche:** Während der Verwendung können gelegentliche Verbindungsabbrüche zustande kommen. Deshalb muss die Verwendung der Anwendung selbst unter entfallen einer Netzwerk- und Internetverbindung möglich sein.

### Restriktionen

Durch die Beschaffenheit von Smartphones gibt es Restriktionen, die berücksichtigt werden müssen. Eine Restriktion ist z. B. die Geschwindigkeit des Prozessors sowie des vergleichsweise geringen Speichers der Smartphones, sodass rechenaufwendige Operationen wie die Auswertung der Bildinformationen oder Darstellung von 3D-Modellen optimiert oder ausgelagert werden müssen, sodass eine flüssige Darstellung möglich ist. Darüber hinaus können Smartphones nur über Touch Gesten auf einem Touch-Screen gesteuert werden und haben keinen CUDA-Grafikchip.

# 3

## Konzeption

In Abschnitt 3.1 werden Verfahren für die Konzeption herangezogen, welche für eine Rekonstruktion von Fotos in ein 3D-Modell benötigt werden. Dabei wird das für diese Arbeit passendste Verfahren verwendet und angepasst. In Abschnitt 3.4 wird ein Konzept beschrieben, welches beschreibt, wie das System aufgebaut ist und die Kommunikation in diesem funktioniert.

### 3.1 Auswertung der Korrespondenzverfahren

Im Folgenden werden Testauswertungen an einer Reihe an Fotos von der Teufelsfigur von Rolf Goerler auf dem Teufelsstein vor der Marienkirche in der Hansestadt Lübeck ausgeführt. Für die Auswertung der Merkmalserkennung wurde ein Foto (siehe Abbildung 3.1 links) verwendet, aus dem alle repräsentativen Merkmale extrahiert werden. Bei der Merkmalsbeschreibung und dem Merkmalsvergleich werden beide Fotos verwendet. Für die Testauswertung gelten folgende Spezifikationen:

Die Abmessung der Fotos beträgt 1290 x 1720 Pixel.

Verwendete System ist ein Acer Aspire A315-53G mit einem Intel Core i5-7200U Prozessor, einer GeForce MX130 Grafikkarte mit einer Arbeitsspeichergröße von 2GB und einer Prozessorgeschwindigkeit von 2,50 GHz.



Abbildung 3.1: Während der Testverfahren verwendete Fotos der Teufelsfigur von Rolf Goerler aus dem Datensatz.

Algorithmus	Detektor	Deskriptor	Matching	Speicherart
SURF [8]	x	x		Fließkommazahlen
SIFT [33]	x	x		Fließkommazahlen
ORB [44]	x	x		Bitstrings
BRISK [32]	x	x		Bitstrings
AKAZE [3]	x	x		Bitstrings
FAST [43]	x			Fließkommazahlen
MSER [42]	x			Fließkommazahlen
FREAK [2]		x		
DAISY [51]		x		
BRIEF [15]		x		
Harris Corner [23]	x			Bitstrings
Shi-Thomasi Method [46]	x			Bitstrings
FLANN [37]			x	
Brute Force [15]			x	

Tabelle 3.2: Getestete Merkmalerkennungs-, Merkmalsbeschreibungs- und Merkmalsvergleichs-Verfahren und ihre Anwendungsgebiete. Verfahren mit der Speicherart Bitstrings besitzen eine schnellere Bearbeitungszeit als Fließkommazahlen, die ergebnen Merkmale sind jedoch schwerer voneinander zu unterscheiden.

In den Testverfahren werden die Algorithmen auf ihre bestmöglichen Einstellungen hin überprüft, mit besonderer Berücksichtigung des Verhältnisses zwischen der benötigten Dauer und Qualität der gefundenen Punkte. Hierzu wurden die empfohlenen Einstellungen der Entwickler in Kombination mit den Ergebnissen der eigenen Testverfahren verwendet. Dabei werden nicht alle existierenden Verfahren geprüft, da dies den Rahmen dieser Arbeit sprengen würde. Weshalb nur Algorithmen getestet werden, die zu diesem Zeitpunkt fehlerfrei von der aktuellsten OpenCV Version 4.1.0 ausgeführt werden können (siehe Tabelle 3.2 für eine Auflistung). Tabelle 3.4 bis Tabelle 3.6 werden die Merkmalerkennungs- und Merkmalsbeschreibungs-Verfahren anhand der Metriken Genauigkeit und Wiederholpräzision mit dem BruteForce- und FLANN-Matcher getestet.

In den Testverfahren (siehe Anhang A) werden die Korrespondenzverfahren anhand existierenden wissenschaftlicher Arbeiten evaluiert. Hierbei wird der SURF-Algorithmus für die Merkmalerkennung verwendet, da sich dieser am besten zum Vergleichen mit bereits existierenden wissenschaftlicher Arbeiten eignet. Gewichtet werden die Algorithmen nach der in Tabelle 3.3 gesetzten Gewichtung, dabei ist ein hoher Wert in der Spalte Gewichtung nützlicher für diese Arbeit, als ein geringer Wert. Die Ergebnisse aus Anhang A werden in Tabelle 3.7 und Tabelle 3.9 zusammengefasst. Die Berechnung der Auswertung setzt sich zusammen aus der Multiplikation der gesetzten Gewichtung und des Schnittes der jeweiligen Tabelle. Da für jedes Verfahren unterschiedliche Implementierungen existieren und selbst das Framework OpenCV die Methoden sich bei den Versionsnummern unterscheiden, werden weitere Testverfahren benötigt, um den passendsten Algorithmus für diese Arbeit auszuwählen.

Im zweiten Testverfahren (siehe Anhang B) bleibt ein Bild unangetastet, während das zu vergleichende Bild anhand der Faktoren Helligkeit, Kontrast, Rotation oder Skalierung verändert wird. Nachfolgend werden die veränderbaren Faktoren als Invarianz betitelt. Tabelle 3.8 und Tabelle 3.10 fassen die Ergebnisse aus Anhang B zusammen. Alle im zweiten Testverfahren getesteten Invarianten werden hierbei gleich gewichtet. Ein roter Kreis (Markierung) in den Ergebnissen deutet darauf hin, dass eines oder mehrere der Testverfahren keine verwertbaren Ergebnisse geliefert haben.

<b>Merkmale</b>	<b>Gewichtung für diese Arbeit</b>
Effizienz	6
Genauigkeit	5
Geschwindigkeit	4
Wiederholpräzision	3
Anzahl der gefundenen Merkmale	2

Tabelle 3.3: Gewichtung der für diese Arbeit notwendigen Merkmalerkennungs-, Merkmalsbeschreibungs- und Merkmalsvergleichseigenschaften.

### **Genauigkeit**

Genauigkeit ist die Eigenschaft richtige Matches zu finden. Je höher dieser Wert ist, desto besser ist die Merkmalerkennung und der Merkmalsvergleich. Errechnen lässt sich diese wie folgt:

$$\text{Genauigkeit} = \frac{\|\text{Richtige Matches aus der Homographie Matrix}\|}{\|\text{Vom Merkmalsvergleich gefundene Merkmale}\|} \quad (3.1)$$

### **Wiederholpräzision**

Wiederholpräzision repräsentiert die Eigenschaft dasselbe Merkmal einer Szenerie unter Veränderung der Invarianten wiederzufinden. Desto höher der Wert ist, desto besser sind die von der Merkmalsbeschreibung gefundenen Merkmale und wird folgendermaßen berechnet:

$$\text{Wiederholpräzision} = \frac{\|\text{True Positive Matches aus der Homographie Matrix}\|}{\|\text{Vom Merkmalsvergleich gefundene Merkmale}\|} \quad (3.2)$$

### **Effizienz**

Effizienz misst den gesamten Bildverarbeitungsprozess und errechnet die Leistungsfähigkeit der Merkmalerkennung, -beschreibung und -vergleich der Verfahren. Je höher dieser Wert ist, desto besser ist der Gesamtprozess. Berechnen lässt sich dieser folgendermaßen:

$$\text{Effizienz} = \text{Genauigkeit} * \text{Wiederholpräzision} \quad (3.3)$$

## Auswertung der Effizienz der Korrespondenzverfahren

In diesem Unterabschnitt werden die Merkmalerkennungs- und -beschreibungs-Verfahren anhand der Bilder der Teufelsfigur getestet. Dabei wird die Wiederholpräzision und Genauigkeit notiert. Hierfür wird die Methode 'evaluateFeatureDetector' von OpenCV genutzt. Für die Berechnung der Wiederholpräzision wird das zu untersuchende Bild mehrmals durch den gewünschten Merkmalerkennung überprüf und berechnet, wie oft dieselben Merkmale gefunden werden. Zum Berechnen der Genauigkeit wird der Methode 'evaluateFeatureDetector' eine von RANSAC berechnete Homographie-Matrix und die gefundenen Matches übergeben. Daraus wird berechnet, wie viele der übergebenen Matches in der Homographie Matrix wiederzufinden sind. Die Ergebnisse wurden in Tabelle 3.4 bis Tabelle 3.6 notiert, dabei bilden die Reihen, die Merkmalerkennungs-Algorithmen, welche mit Erkennen abgekürzt und die Spalten die Merkmalsbeschreibungs-Algorithmen, die mit Beschreiben abgekürzt werden.

Erkennen	Beschreiben								
	SURF	SIFT	ORB	BRISK	BRIEF	FREAK	DAISY	AKAZE	
SURF	0,833	0,833					0,833	0,833	
SIFT	0,668	0,668					0,668	0,668	
ORB	0,806	0,806	0,806		0,806		0,806	0,806	
BRISK	0,720	0,720		0,720			0,720	0,720	
<b>AKAZE</b>	<b>0,844</b>								
FAST	0,903	0,903					0,903	0,903	
MSER	0,537	0,537	0,537	0,537	0,537	0,537	0,537	0,537	
KAZE	0,890	0,890					0,890	0,890	
Harris-Corner							0,918		
Shi-Tomasi							0,918		

Tabelle 3.4: Bewertung der Wiederholpräzision. Horizontal sind die Merkmalerkennungs- und Vertikal die Merkmalsbeschreibungs-Algorithmen. Je näher der Wert bei eins ist, desto präziser sind die Ergebnisse. Der Gelb markierte Algorithmus ist der vielversprechendste für diese Arbeit.

Erkennen	Beschreiben							
	SURF	SIFT	ORB	BRISK	BRIEF	FREAK	DAISY	AKAZE
SURF	0,05	0,273					0,156	0,155
SIFT	0,05	0,420					0,223	0,176
ORB	0,092	0,078	0,007		0,005		0,13	0,085
BRISK	0,059	0,134		0,038			0,059	0,094
<b>AKAZE</b>	<b>0,195</b>	<b>0,163</b>	<b>0,182</b>	<b>0,253</b>	<b>0,192</b>	<b>0,142</b>	<b>0,307</b>	<b>0,277</b>
FAST	0,015	0,115					0,027	0,054
MSER	0,036	0,205		0,053	0,048	0,262	0,086	0,107
KAZE	0,013	0,126	0,182	0,253	0,192	0,142	0,207	0,175
Harris-Corner							0,25	
Shi-Tomasi							0,25	

Tabelle 3.5: Bewertung der Genauigkeit mit BruteForce als Merkmalsvergleich. Horizontal sind die Merkmalerkennungs- und Vertikal die Merkmalsbeschreibungs-Algorithmen. Je näher der Wert bei eins ist, desto besser sind die gefundenen Matches. Der Gelb markierte Algorithmus ist der vielversprechendste für diese Arbeit.

Erkennen	Beschreiben							
	SURF	SIFT	ORB	BRISK	BRIEF	FREAK	DAISY	AKAZE
SURF	0,048	0,352					0,129	0,730
SIFT	0,050	0,519					0,223	0,661
ORB	0,131	0,114	0,125		0,123		0,106	0,104
BRISK	0,120	0,124		0,018	0,004		0,123	0,104
<b>AKAZE</b>	<b>0,198</b>	<b>0,163</b>	<b>0,135</b>	<b>0,255</b>		<b>0,158</b>	<b>0,222</b>	<b>0,308</b>
FAST	0,015	0,115					0,030	0,08
MSER	0,016	0,205		0,043	0,037	0,016	0,186	0,205
KAZE	0,013	0,126					0,047	0,276
Harris-Corner							0,250	
Shi-Tomasi							0,250	

Tabelle 3.6: Bewertung der Genauigkeit mit FLANN als Merkmalsvergleich. Horizontal sind die Merkmalerkennungs- und Vertikal die Merkmalsbeschreibungs-Algorithmen. Je näher der Wert bei eins ist, desto Präziser sind die Ergebnisse. Der Gelb markierte Algorithmus ist der vielversprechendste für diese Arbeit.

## Auswertung der Merkmalerkennungsverfahren

**Erste Testauswertung:**

Merkmals- erkennungs- verfahren	Gewichtung der Wiederhol- präzision	Gewichtung der Anzahl der gefundenen Merkmale	Gewichtung der Lauf- zeit	Auswertung
FAST	9	4	12	25
MSER	9	6	10	25
<b>AKAZE</b>	<b>7,8</b>	<b>6</b>	<b>8</b>	<b>21,8</b>
BRISK	7,8	3	9,6	20,4
SURF	7,8	4	8	19,8
ORB	6,75	2	9,2	17,05
SIFT	6,9	6	4	16,9
Harris Corner	6	2	8	16
Shi-Thomasi	6	2	8	16

Tabelle 3.7: Beim Ergebnis wird die Gewichtung und der Durchschnitt aller Ergebnisse aus Anhang A multipliziert. Rote Markierungen deutet auf fehlerhafte Ergebnisse hin. Je höher die Werte sind, desto besser sind Algorithmen in der untersuchten Kategorie. Der Gelb markierte Algorithmus ist der vielversprechendste für diese Arbeit.

**Zweite Testauswertung:**

Merkmals- erkennungs- verfahren	Wiederhol- präzision	Genauigkeit	Gewichtung der Laufzeit	Auswertung
<b>AKAZE</b>	<b>9</b>	<b>13,75</b>	<b>12</b>	<b>34,75</b>
FAST	9	12	10,4	● 31,4
ORB	9	12,5	9,33	● 30,83
SURF	8,63	11,88	10	30,51
MSER	6,75	13,13	10,5	30,38
BRISK	6	10,71	10,29	● 27
Harris-corner	8,25	5	12	25,25
Shi-Thomasi	8,25	5	12	25,25
SIFT	7,13	11,88	5	24,01
KAZE	6	6,67	4	● 16,67

Tabelle 3.8: Beim Ergebnis wird die Gewichtung und der Durchschnitt aller Ergebnisse aus Anhang B multipliziert. Rote Markierungen deutet auf fehlerhafte Ergebnisse hin. Je höher die Werte sind, desto besser sind Algorithmen in der untersuchten Kategorie. Der Gelb markierte Algorithmus ist der vielversprechendste für diese Arbeit.

### Ergebnis der Auswertung der Merkmalerkennungs-Verfahren

SIFT benötigt wesentlich mehr Zeit, liefert aber nicht entscheidend bessere Merkmale, als die anderen Verfahren. FAST, MSER und SURF sind zwar schnell, erzeugen praktisch keine besseren Resultate als AKAZE. Nach der gesetzten Gewichtung hat AKAZE sowohl bei der ersten, als auch bei der zweiten Testauswertung sehr gut abgeschnitten.

### Auswertung der Merkmalsbeschreibungsverfahren

#### Erste Testauswertung

Merkmals- erkennungs-Verfahren	Gewichtung der Genauigkeit	Gewichtung der Laufzeit	Auswertung
FREAK	15	12	27
AKAZE	15	9,2	24,2
BRISK	13	9,2	21,2
DAISY	15	4	19
ORB	5	12	17
BRIEF	8	8	16
SURF	10	5	15
SIFT	10	4	14

Tabelle 3.9: Beim Ergebnis wird die Gewichtung und der Durchschnitt aller Ergebnisse aus Anhang A multipliziert. In Auswertung werden die Ergebnisse addiert. Rote Markierungen deutet auf fehlerhafte Ergebnisse hin. Je höher die Werte sind, desto besser sind Algorithmen in der untersuchten Kategorie. Der Gelb markierte Algorithmus ist der vielversprechendste für diese Arbeit.

#### Zweite Testauswertung

Merkmals- erkennungs-Verfahren	Genauigkeit	Dauer	Auswertung
AKAZE	11,88	12	23,88
DAISY	15	7	22
FREAK	8,75	12	● 20,75
BRISK	10	10,4	● 20,4
SURF	10	8,67	18,67
SIFT	13,75	4	17,75
BRIEF	Keine verwertbaren Ergebnisse		
ORB	Keine verwertbaren Ergebnisse		

Tabelle 3.10: Beim Ergebnis wird die Gewichtung und der Durchschnitt aller Ergebnisse aus Anhang B multipliziert. Rote Markierungen deutet auf fehlerhafte Ergebnisse hin. Je höher die Werte sind, desto besser sind Algorithmen in der untersuchten Kategorie. Der Gelb markierte Algorithmus ist der vielversprechendste für diese Arbeit.

### Ergebnis der Auswertung der Merkmalsbeschreibungs-Verfahren

Welcher Algorithmus sich am besten für die Merkmalerkennung und Merkmalsbeschreibung eignet, hängt von den Bildern ab. In den Testverfahren wurde AKAZE für die Merkmalerkennung genutzt, wodurch einige Verfahren wie BRIEF oder ORB nicht optimal verwendet werden konnten. Aufgrund der schnellen Bearbeitungszeit und Qualität wird in dieser Arbeit AKAZE für die Merkmalerkennung und Merkmalsbeschreibung, während der Fotoerstellung verwendet. Um ein 3D-Modell zu konstruieren wird auf AKAZE und DAISY zurückgegriffen, da diese die qualitativ besten Ergebnisse liefern und am besten mit Invarianten auskommen.

### Merkmalsvergleichsverfahren

**Erste Testauswertung:** Es konnte keine ausreichende Menge an wissenschaftlichen Evaluationen der Merkmalsvergleichs-Verfahren gefunden, damit ein Vergleich der Ergebnisse ausgeführt werden kann. In [10] wurden mehrere Verfahren verglichen. FLANN hat dabei von allen von OpenCV zur Verfügung gestellten Merkmalsvergleichs-Verfahren besten Ergebnisse geliefert.

### Zweite Testauswertung:

Merkmalsvergleichsverfahren	Durchschnittliche Wiederholpräzision	Durchschnittliche Genauigkeit	Durchschnittliche Dauer	Auswertung
BruteForce	8,03	10,34	9,02	27,39
FLANN	7,89	10,96	9,68	28,53

Tabelle 3.11: Beim Ergebnis wird die Gewichtung und der Durchschnitt aller Ergebnisse aus Anhang B multipliziert. Je höher die Werte sind, desto besser sind Algorithmen in der untersuchten Kategorie. Der Gelb markierte Algorithmus ist der vielversprechendste für diese Arbeit.

FLANN hat in den Versuchen bessere Ergebnisse als das BruteForce-Verfahren erstellt. FLANN-Algorithmus besitzt viele Einstellungsmöglichkeiten, die gezielt an die verwendeten Deskriptor-Verfahren angepasst werden müssen. Es wurde in den Versuchen bestmögliche Einstellungen verwendet. Trotzdem ist es nicht ausgeschlossen, dass sich mit FLANN bessere Resultate, durch weitere Testverfahren erreichen lassen. Unter anderem hat das BruteForce Verfahren in [10] besser als das FLANN-Verfahren abgeschnitten. Ein weiterer Vorteil an Flann ist, dass es eine Menge an Algorithmen bereitstellt, die einen größeren Spielraum bei der Suche von Matches ermöglichen.

### 3.2 Auswertung der Verfahren zur Modellerstellung

Eine eigene Implementierung für das Erstellen, Verdichten und Texturieren einer Punktwolke würde den zeitlichen Rahmen dieser Arbeit sprengen, weshalb auf Anwendungen zurückgegriffen wird, in denen diese bereits vollständig implementiert sind. Nachfolgend werden alle bekannten und nicht kommerziellen Anwendungen zum Erstellen eines 3D-Modells betrachtet. Zusätzlich werden nur Anwendungen getestet, welche nicht webbasiert sind. Da bei webbasierten Anwendungen eine automatisierte Rekonstruktion und die Datensicherheit der Anwender nicht garantiert werden kann (siehe Abbildung 3.12 für eine Übersicht der getesteten Applikationen).

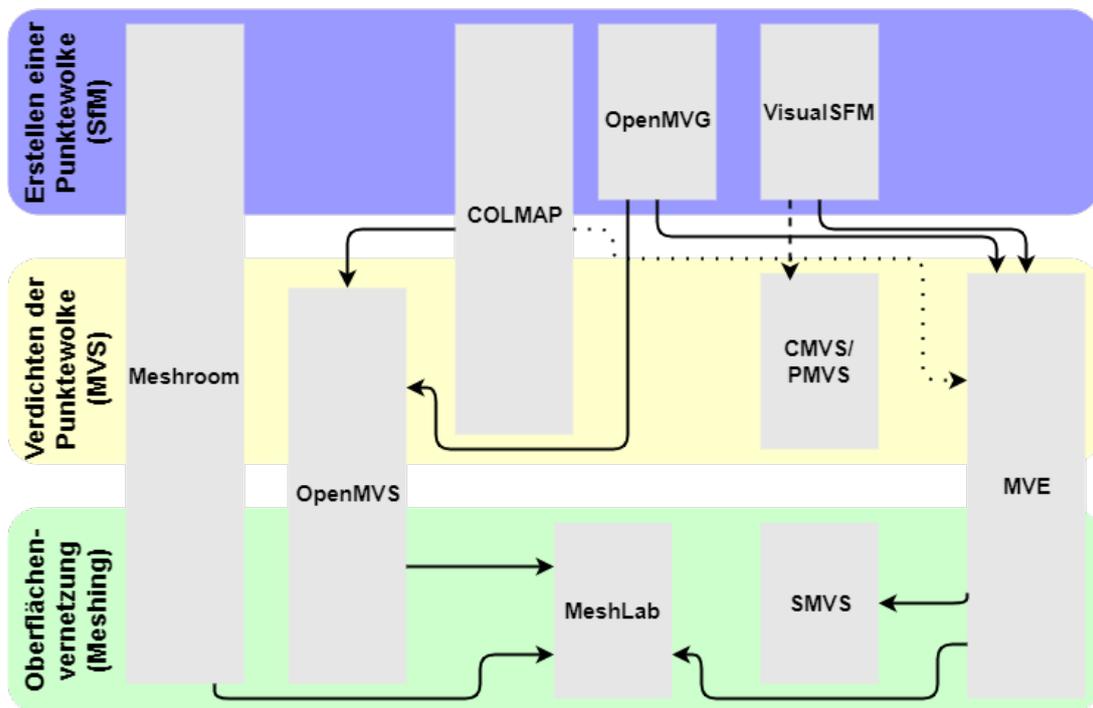


Abbildung 3.12: Übersicht der getesteten Rekonstruktionsverfahren und ihrer Beziehungen zueinander.

Die Applikationen wandeln eine Menge an Bildern anhand ihrer 3D-Punkte mittels der Verfahren SfM und MVS in eine Punktwolke um. In den Verfahren werden die Applikationen mit den bestmöglichen Einstellungen ausgeführt. Die Einstellungen wurden hierzu aus einer Kombination zwischen den von den Entwicklern empfohlenen Werten und eigenen Testverfahren ermittelt. In Tabelle 3.14 werden die ergebnen Punktwolken und in Tabelle 3.15 die 3D-Modelle visuell dargestellt. Bei den Abbildungen wurden die 3D-Modelle aus einer Perspektive aufgenommen, in der die Qualität der Umwandlung am deutlichsten zu erkennen ist. Tabelle 3.13 stellt die Dauer der Umwandlung, die Anzahl der gefundenen Merkmale und Anzahl der erstellten Polygone dar. Für die Darstellung einer dichten Punktwolke ist eine hohe Anzahl an gefundenen Merkmalen notwendig.

Hierbei hat sich in der Analyse gezeigt, dass aus dichteren Punktwolken in den meisten Fällen detaillierte und lückenlose 3D-Modelle konstruiert werden. Ebenso hat sich gezeigt, dass bei 3D-Modellen eine hohe Anzahl der erstellten Polygone von Relevanz sind. Ein hoher Wert an Polygonen ist häufig ein Indikator wie detailliert das 3D-Modell ist. Sowohl bei der Erstellung der Punktwolke, als auch des 3D-Modells ist eine geringe Dauer der Umwandlung von Interesse. Als Eingabemenge werden die folgenden Bildermengen verwendet:

1. 1 Bildermenge: 22 Bilder mit einer Dimension von 1290 x 1720 Pixel der Tefelsfigur auf dem Teufelsstein vor der Marienkirche in der Hansestadt Lübeck.
2. 2 Bildermenge: 25 Bilder mit einer Dimension von 2124 x 2843 Pixel der Statue Johannes Brahms auf der Wallstraße in der Hansestadt Lübeck.
3. 3 Bildermenge: 90 Bilder mit einer Dimension von 1836 x 3264 Pixel der Statue Hygieia auf der Sana-Klinik in der Hansestadt Lübeck.

### 3 Konzeption

Rekonstruktions-Verfahren	Bilderreihe	Dauer der Umwandlung in Minuten	Anzahl der gefundenen Merkmalen	Anzahl der erstellten Polygone
COLMAP	1	2,59	2.308.612	
	2	5,27	2.322.354	
	3	28,48	14.515.020	
COLMAP MVE	1	14,12	112.342	221.705
	2	16,17	139.429	271.739
	3	127,17	14.515.020	939.522
COLMAP MVE SMVS	1	36,44	528.186	1.035.145
	2	90,35	1.114.407	2.156.608
	3	127,17	21.707	
VisualSFM	1	3,18	13.361	
	2	5,42	11.609	
	3	14,30	21.707	
VisualSFM CMVS/PMVS	1	4,52	134.917	
	2	81,12	266.212	
	3	10,04	266.212	
VisualSFM MVE	1	16,42	75.482	124.114
	2	51.51	148.640	290.159
	3	122.40	484.494	947.061
VisualSFM MVE SMVS	1	19,24	115.750	228.539
	2	95,05	901.282	1.732.295
	3	303,42	2.777.817	5.418.026
Meshroom	1	9,37	136.277	272.324
	2	19,25	604.191	1.208.179
	3		Kein Ergebnis	
OpenMVG	1	4,22	2.351.042	
	2	12,22	2.916.919	
	3	36,03	14.912.847	
OpenMVG MVE	1	7,35	99.083	195.188
	2	88,23	159.613	309.163
	3	135,45	517.378	1.009.311
OpenMVG MVE SMVS	1	35,21	933.821	1.858.599
	2	136.51	1.896.139	3.763.842
	3	344,05	5.435.850	10.796.025

Tabelle 3.13: Vergleich der Rekonstruktions-Verfahren. Die gelb markierten Algorithmen stellen die vielversprechendsten Verfahren für diese Arbeit dar.

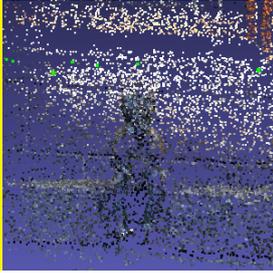
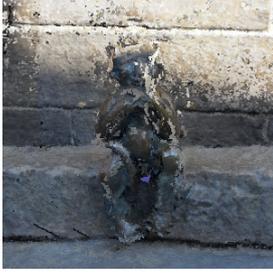
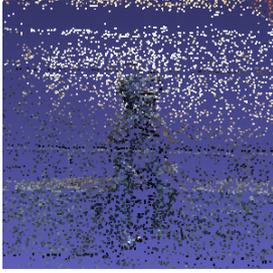
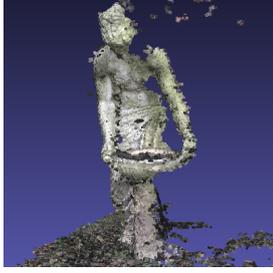
Rekonstruktions- Verfahren	Ergebnis der ersten Bildreihe	Ergebnis der zweiten Bildreihe	Ergebnis der dritten Bildreihe
COLMAP			
Meshroom			Kein Ergebnis
OpenMVG			
VisualSFM			
VisualSFM CMVS/PMVS			

Tabelle 3.14: Vergleich der Punktwolken, die mit einem Rekonstruktions-Verfahren hergestellt wurden. Der gelb markierte Algorithmus stellt das vielversprechendste Verfahren für diese Arbeit dar.

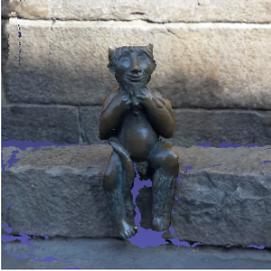
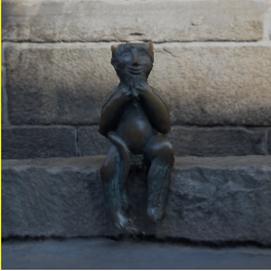
Verfahren	Ergebnis der ersten Bildreihe	Ergebnis der zweiten Bildreihe	Ergebnis der dritten Bildreihe
COLMAP MVE			
COLMAP MVE SMVS			
Meshroom			Kein Ergebnis
OpenMVG MVE			
OpenMVG MVE SMVS			

Tabelle 3.15: Teil 1 des Vergleichs der 3D-Modelle, die mit einem Rekonstruktions-Verfahren hergestellt wurden. Die gelb markierten Algorithmen stellen die vielversprechendsten Verfahren für diese Arbeit dar.

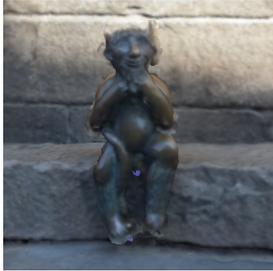
Verfahren	Ergebnis der ersten Bildreihe	Ergebnis der zweiten Bildreihe	Ergebnis der dritten Bildreihe
VisualSFM MVE			
VisualSFM MVE SMVS			

Tabelle 3.16: Teil 2 des Vergleichs der 3D-Modelle, die mit einem Rekonstruktions-Verfahren hergestellt wurden. Die rot markierten Algorithmen stellen die vielversprechendsten Verfahren für diese Arbeit dar.

### Bewertung der Verfahren zur Modellherstellung

In der Testauswertung hat sich herausgestellt, dass Meshroom bzw. OpenMVG und MVE in Kombination mit SMVS die qualitativ höchsten 3D-Modelle erstellen. Anzumerken sei, dass SMVS die höchste Laufzeit besitzt und das OpenMVG eines der wenigen Verfahren ist, dessen Quellcode öffentlich zugänglich ist und somit zugunsten dieser Arbeit verändert werden kann. Weiter zeigt sich, dass das Erstellen eines 3D-Modells unter den Versuchsbedingungen und auf den Versuchssystemen dieser Arbeit ungefähr doppelt so lange dauert, wie das Erstellen einer Punktwolke. Aufgrund dessen wird zur Laufzeit aus den Fotos eine Punktwolke erstellt, da diese für die Konstruktion eines Modells unumgänglich ist und zukünftig um das Rekonstruieren eines 3D-Modells während der Fotoaufnahme erweitert werden kann.

### 3.3 Vorteile der Client-Server-Architektur

Während der Analyse wurde ersichtlich, dass der Prozess zum Konstruieren eines 3D-Modells viele rechenintensive Verfahren durchlaufen muss. Sie alle von einem Smartphone auszuführen würde das Smartphone überlasten, was zu zeitverzögerten Ergebnissen und einem hohen Energiebedarf führt. Weiter ist aus den bestehenden Systemen aus Abschnitt 1.5 ersichtlich, dass die Fotos dafür herunterskaliert werden müssten, was

qualitativ schlechtere Modelle ergibt. Da es ebenfalls das Ziel dieser Arbeit ist, die Anwendung für eine möglichst große Menge an Smartphone bereitzustellen, wird ein Ansatz verfolgt, bei dem die rechenintensiven Prozesse auf einen Server ausgelagert werden, sodass das Smartphone nur für die Fotoerstellung, Vorselektierung der Fotos und Veranschaulichung der Modelle verwendet wird.

#### 3.4 Systemstruktur

Die unternommene Auswertung aus Kapitel 3 zeigt, dass sich der Prototyp konzeptuell in drei Komponenten gliedern lässt. Einer Smartphone-Anwendung, welche eine Benutzeroberfläche für den Anwender während der Erstellung der Fotos zur Verfügung stellt und eine Vorselektierung der Fotos vornimmt.

Die von der *Smartphone-Anwendung* akzeptierenden Fotos werden über das Internet an eine *Server-Anwendung* gesendet, um daraus ein 3D-Modell zu konstruieren. Darüber hinaus werden Zusatzinformationen während der Aufnahme erhoben, welche den Modellerstellungsprozess auf der Serverseite unterstützen.

Eine *Datenbank* verwaltet die Kommunikation, welche zwischen der Smartphone- und der Server-Anwendung realisiert wird. Die Einteilung der Prototypen ist sinnvoll gewählt, da somit rechenaufwendige Operationen auf den Server verschoben werden können, ohne die Smartphone Anwendung zu überlasten.

In diesem Abschnitt werden die wichtigsten Abläufe erläutert. Hierzu werden alle zwischen der Smartphone- und der Server-Anwendung interagierenden Prozesse beschrieben und mit einem dazugehörigem Ablaufdiagramm visualisiert. Weitere Informationen zu den einzelnen Implementierungen werden in Abschnitt 4.1 behandelt.

#### Verbindungsanwendung

Der Smartphone-Client stellt die Hauptkomponente der Logik dar. Dieser dient einerseits als Schnittstelle für die Interaktion, über welchen die Bilder und Modelle verwaltet werden und andererseits steuert dieser den Programmablauf der Server-Komponente (siehe Abbildung 3.17). Nachdem der Smartphone-Client gestartet und ein neuer Auftrag erstellt wurde, propagiert er dem Server seine Anwesenheit. Am Server wird zum Auftrag eine eindeutige Auftrags-ID, und wenn noch nicht vorhanden eine Nutzer-ID, erstellt. Die Kommunikation zwischen dem Server- und dem Smartphone-Client geschieht komplett mittels JSON-Objekten.

Anschließend wird dem Anwender eine Maske angezeigt, über die er Fotos von einem Objekt erstellen kann. Erstellt der Nutzer ein Foto, so wird geprüft, ob es verwendet werden kann. Kann es nicht verwendet werden, so wird der Nutzer darüber informiert, andernfalls wird das Foto an den Server gesendet. Am Server angekommen, werden alle zum Auftrag erstellten Fotos in eine Punktwolke umgewandelt, welche bei Anfrage des Smartphone Clients zurückgesendet werden, wo diese dem Nutzer präsentiert wird. Möchte der Benutzer den Auftrag beenden, so wird auf dem Server ein 3D-Modell aus

den Fotos erstellt. Das Modell kann vom Smartphone aufgerufen werden, welches das Modell speichert und dem Nutzer präsentiert.

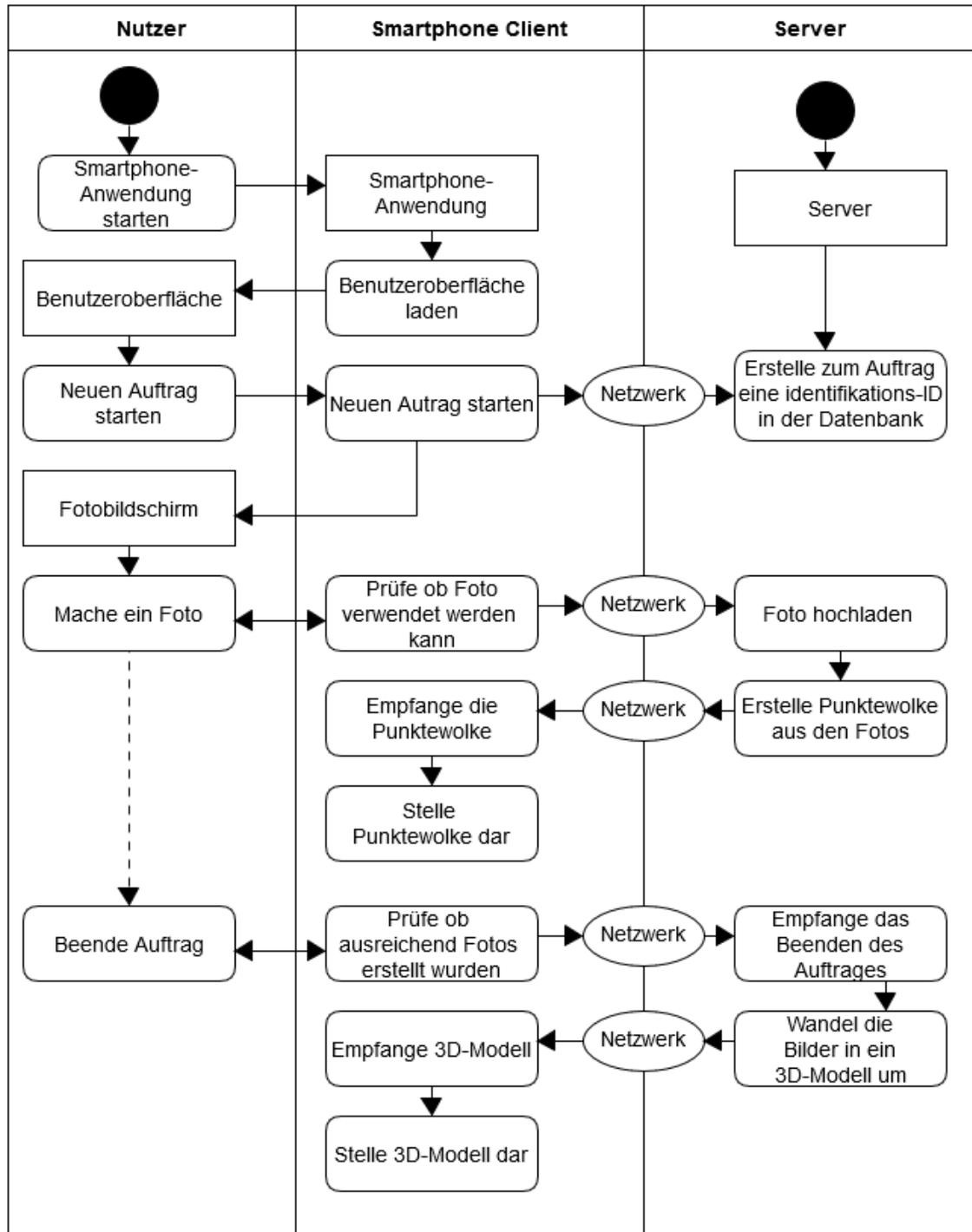


Abbildung 3.17: Schematische Darstellung des Ablaufs zum Erstellen eines 3D-Modells, der Nutzerinteraktion, Smartphone- und Server-Anwendung.

#### Smartphoneanwendung

Die Smartphoneanwendung besitzt mehrere Fenster (siehe Abschnitt 4.1). In diesem Unterabschnitt wird nur die Logik beim Fotografieren behandelt (siehe Abbildung 3.18), da dies das Hauptaugenmerk dieser Arbeit ist. Der Nutzer kann ein Verfahren ausführen, welches untersucht, ob ein erwünschtes Objekt sich rekonstruieren lässt. Hat sich der Anwender für ein Objekt entschieden, kann dieser ein Foto vom Objekt erstellen. Anhand des erstellten Fotos besteht die Möglichkeit, sich bei den Folgebildern anzeigen zu lassen, ob diese ausreichend viele zueinander passende Merkmale besitzen. Diese Methode unterstützt den Nutzer während der Fotoaufnahme und informiert, welche Fotos sich für eine Rekonstruktion eignen. Berechnungen der Merkmale werden nur dann ausgeführt, wenn sich das Smartphone minimal bewegt hat oder länger als zwei Sekunden keine Berechnung mehr erfolgt ist. Andernfalls werden die zuletzt berechneten Matches angezeigt. Wenn ein Foto vom Objekt erstellt wurde, wird überprüft, ob sich die durchschnittlichen RGB-Werte des Bildes zwischen 80 und 220 befinden, das Foto frei von Verzerrungen und Unschärfe aufgenommen wurde und ausreichend viele Matches zwischen dem aktuellem und dem zuletzt aufgenommenen Foto vorhanden sind. Sollte dies der Fall sein, so wird das Foto auf den Server hochgeladen und dort in eine Punktwolke umgewandelt und dem Anwender präsentiert. Sobald der Anwender mit der Punktwolke zufrieden ist, so kann er dem Server ein Signal geben. Durch das Signal wird dem Server mitgeteilt, die Fotos in ein 3D-Modell umzuwandeln, welches dem Anwender bei Fertigstellung präsentiert wird.

#### Serveranwendung

Beim Start der Serveranwendung wird überprüft, ob ein Flag auf dem Server gesetzt wurde. Wurde ein solches Flag gesetzt, werden die zum Flag gehörenden Fotos in eine Punktwolke oder einem 3D-Modell überführt

Soll eine Punktwolke erstellt werden, so werden mittels Meshroom alle Merkmale gesucht, beschrieben und miteinander verglichen. Die daraus resultierenden Punktabbildungen werden in einer Textdatei gespeichert und dazu genutzt, die Fotomenge zu selektieren. Es werden alle Fotos entfernt, welche keine beziehungsweise wenig Gemeinsamkeiten zu ihren 5 vorherigen und nachfolgenden Nachbarn besitzen. Hierbei wurde ein Wert von 5 gewählt, da sich während der Analyse gezeigt hat, dass bei einem kleineren Wert verwertbare Fotos entfernt werden. Bei einem größeren Wert hingegen werden Matches durch Bewegungen Dritter im Hintergrund gefunden, welche zu Fehlinterpretationen führen. Bei diesen Fotos kann man davon ausgehen, dass sie nicht zur Fotoreihe gehören. Das Entfernen dieser Fotos reduziert die Berechnungszeit und verhindert, dass u. U. unerwünschte Modelle erstellt werden. Aus den übrig gebliebenen Fotos werden die ermittelten Kameraparameter an die Anwendung Meshroom übergeben, welches aus den Fotos eine Punktwolke erstellt. Als letzten Ausführungsschritt reduziert MeshLab die Anzahl der Punkte in der Punktwolke und überführt diese in ein für die Smartphone-Anwendung geeignetes Format.

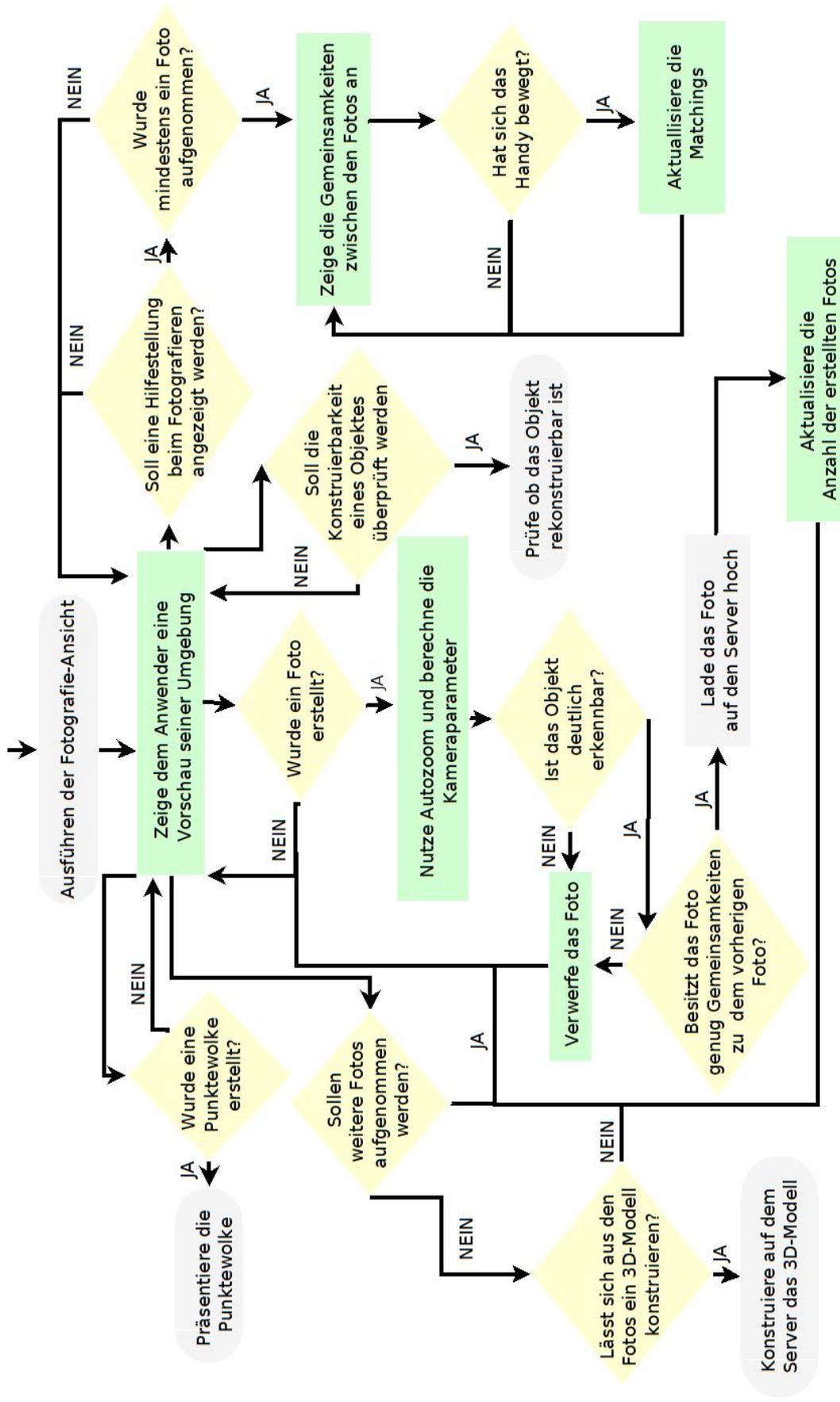


Abbildung 3.18: Schematische Darstellung des Ablaufs, der Komponenten und Verbindungen auf der Smartphone-Ansicht. Grün und Gelb gefärbte Felder stellen Methoden, Grau die Übergänge, Entscheidungen werden durch ein Rauten-Symbol, Tätigkeiten durch ein Rechteck und Terminatoren durch ein Rechteck mit gerundeten Ecken gekennzeichnet.

Die Punktwolke wird abschließend auf dem Server gespeichert und bei Anfrage des Smartphones unter Zuhilfenahme der Auftrags-ID an dieses übergeben.

Falls der Anwender ausreichend Fotos aus allen Richtungen gesammelt hat, kann die Umwandlung in ein 3D-Modell begonnen werden. Zuerst hat der Nutzer die Auswahl, welche Qualität das Modell besitzen soll. Bei der Qualitätsstufe 'normal' wird Meshroom ausgeführt, welches qualitativ mindere 3D-Modelle herstellt, aber dafür eine kurze Bearbeitungszeit besitzt. Dabei wird in Meshroom sowohl bei der Erstellung des 3D-Modells als auch bei der Punktwolke, AKAZE für die Merkmalerkennung und Merkmalsbeschreibung verwendet. Wählt der Anwender die Qualitätsstufe 'hoch' aus, so werden die Verfahren AKAZE und DAISY aus OpenCV dazu genutzt, die Merkmale zu suchen. Die Ergebnisse werden an OpenMVG übergeben, um daraus eine Punktwolke zu erstellen, welches mittels OpenMVE verdichtet wird. Die Punktwolke wird dann an die Anwendung SMVS übergeben, welche daraus ein 3D-Modell konstruiert. Dieses Verfahren dauert wesentlich länger als das von Meshroom, erstellt jedoch in den meisten Fällen ein qualitativeres 3D-Modell. Nachdem das 3D-Modell erstellt wurde, wird MeshLab ausgeführt, um das 3D-Modell ruckelfrei auf der Smartphone-Anwendung darstellbar zu machen.

Im ersten Schritt kommt der *Poisson Surface Reconstruction* Algorithmus zum Einsatz, um einen Abdruck ähnlich dem eines Gipsabdrucks vom Modell zu erstellen und die Textur zwischenspeichern. Der erstellte Abdruck wird mittels der Funktion *Quadric Edge Collapse Decimation* auf eine Polygonmenge von 150.000 reduziert, da sich diese Menge problemlos auf gängige Smartphones darstellen lässt. Anschließend wird die Funktion *Vertex Attribute Transfer* dazu verwendet alle Details des Originalmodells auf den erstellten Abdruck zu übertragen. Das daraus entstandene Resultat besitzt dieselben Details wie das Original. Anschließend wird das Modell durch die Ausführung vom *Laplacian Smooth* geglättet und vorhandene Lücken im Modell verdichtet. Einziger Nachteil des Vorgangs ist, dass die Textur nicht auf den Abdruck mit übertragbar wird. Hierzu kommt die Funktion *Transfer Vertex to Texture* zum Einsatz, welche anhand des Originalmodells die Positionierung der Textur berechnet und anschließend die gespeicherte Textur auf den Abdruck überträgt. *Transfer Vertex to Texture* texturiert ebenfalls Stellen, welche in den vorherigen Schritten hinzugekommen sind. Das gespeicherte Modell kann bei Bedarf von der Smartphone-Anwendung heruntergeladen und präsentiert werden.

### 3 Konzeption

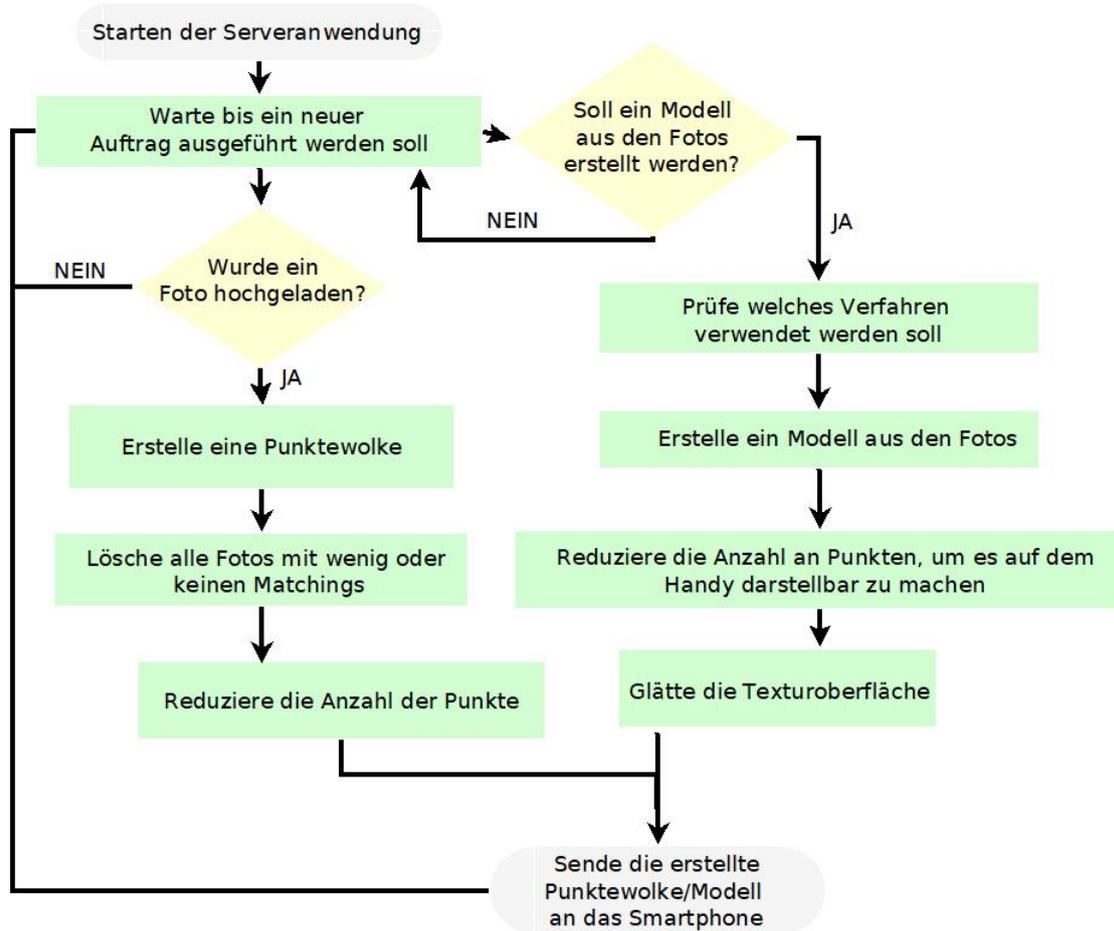


Abbildung 3.19: Schematische Darstellung des Ablaufs auf der Serveransicht. Grün und Gelb gefärbte Felder stellen Methoden, Grau die Übergänge, Entscheidungen werden durch ein Rauten-Symbol, Tätigkeiten durch ein Rechteck und Terminatoren durch einen Rechteck mit gerundeten Ecken gekennzeichnet.

### 3 Konzeption

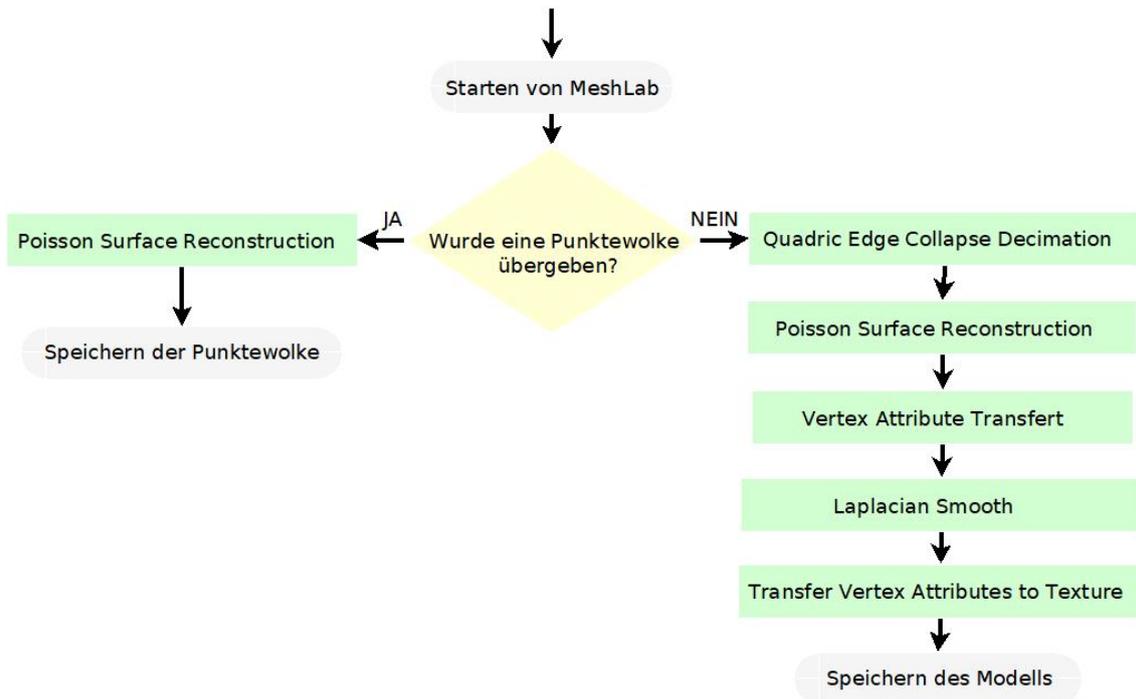


Abbildung 3.20: Schematische Darstellung des Ablaufs MeshLab-Skripte. Grün und Gelb gefärbte Felder stellen Methoden, Grau die Übergänge, Entscheidungen werden durch ein Rauten-Symbol, Tätigkeiten durch ein Rechteck und Terminatoren durch ein Rechteck mit gerundeten Ecken gekennzeichnet.

# 4

## Realisierung

Dieser Abschnitt gibt in Abschnitt 4.1 einen Einblick in die relevantesten Aspekte der Realisierung. Abschnitt 4.2 beschäftigt sich mit den Hardwareanforderungen, die benötigt werden, um die Anwendung komplikationsfrei auszuführen.

### 4.1 Implementierung

In diesem Unterabschnitt werden die eigen für diese Arbeit implementierten Methoden erläutert. Die erstellten Verfahren werden auf Basis der untersuchten Sensoren aus Abschnitt 2.6, der Problem-, Kontext- und Aufgabenanalyse aus Abschnitt 2.7 und der Konzeption aus Abschnitt 3.4 erstellt.

#### Kameraparameter

Für die Ausführung der SfM-Verfahren von OpenMVG und Meshroom wird als Übergabeparameter die Fokallänge in Pixel benötigt. Bei Smartphones lassen sich die Parameter der projektiven Abbildung nicht aus den EXIF Informationen auslesen. Dies liegt wohl an der Kompaktheit von Smartphones begründet, da diese nur einen kleinen Sensor und Blende besitzen. Dies führt dazu, dass eine große Schärfentiefe erreicht wird, die es erlaubt, große Flächen fokussiert aufzunehmen und sich das Smartphone somit nicht wie eine DSLR-Kamera<sup>3</sup> kontrolliert lässt.

Eine Möglichkeit, die Fokallänge zu erhalten, ist eine Onlinedatenbank zu verwenden, die ständig aktualisiert wird. Leider konnte bei der Recherche keine solche Datenbank mit Bezug zu Smartphones gefunden werden. Eine weitere Möglichkeit ist es, die Kamerakalibrierung mithilfe von sogenannten Kalibrierungsmustern, vor Beginn des Einsatzes der Kamera auszuführen. Dabei sind für verschiedene Einstellungen der Kamera voneinander unabhängige Kalibrierungen erforderlich. Solch eine Vorgehensweise ist sehr aufwendig, beschränkt die Flexibilität beim Einsatz der Anwendung und erzwingt die Bereitstellung eines möglichst exakten Kalibriermusters. Durch Bibliotheken wie

---

<sup>3</sup> Eine DSLR ist eine Digitalkamera, welches einen digitalen Bildsensor und die Mechanismen einer Spiegelreflexkamera besitzt.

'camera2' in Android Studio lassen sich diese zwar zurückgeben, so sind die Werte häufig verkehrt oder führen zu einer Fehlermeldung. Die sicherste Vorgehensweise, um die Fokallänge in Pixel zu erhalten, wäre es diese zu berechnen. Dazu wird in dieser Arbeit die folgende Formel verwendet:

$$\begin{aligned} \text{Fokallänge (pixel)} &= \frac{\text{Fokallänge (mm)} * \text{Bildbreite (pixel)}}{\text{Sensorbreite (mm)}} & (4.1) \\ \equiv \text{Fokallänge (mm)} &= \frac{\text{Fokallänge (pixel)} * \text{Sensorbreite (mm)}}{\text{Bildbreite (pixel)}} \end{aligned}$$

Aus Android Studio können die Werte für die horizontale oder vertikale Bildwinkel, die Bildbreite und die Fokallänge ermittelt werden. Somit fehlt noch die physikalische Sensorbreite. Hierzu wird die folgende Formel verwendet, um die fehlenden Werte zu erhalten:

$$\begin{aligned} \text{Bildwinkel (Bogenmaß)} &= 2 * \arctan\left(\frac{\text{Physikalische Sensorbreite (mm)}}{2 * \text{Fokallänge (mm)}}\right) & (4.2) \\ \equiv \text{Physikalische Sensorbreite (mm)} &= \tan\left(\frac{\text{Bildwinkel (Bogenmaß)}}{2}\right) * 2 * \text{Fokallänge (mm)} \end{aligned}$$

### Anwendung zum Erstellen des 3D-Modells

Bei der Rekonstruktion des 3D-Modells werden die Anwendungen Meshroom oder OpenMVG in Kombination mit OpenMVE und SMVS für die Überführung in ein 3D-Modell verwendet. Die Analyse der Auswertung der Korrespondenzverfahren in Abschnitt 3.1 zeigte das AKAZE in Kombination mit DAISY, die qualitativ höchste Anzahl an Merkmalen findet. Da OpenMVG derzeit nur die Verfahren AKAZE und SIFT unterstützt, wurde das Programm um das Beschreiben der Merkmale mittels Daisy aus der Programmbibliothek OpenCV im Rahmen dieser Arbeit erweitert. Bei der Erstellung der Punktwolke werden stets die Ergebnisse der vorherigen Ergebnisse weiterverwendet und jedes Foto wird mit den 5 vorherigen Bildern verglichen. Es werden nur die 5 vorherigen bzw. nachfolgenden Bildern verglichen, weil ein geringerer Wert viele Bilder bei einer Änderung des Blickwinkels entfernt und bei einem höherem Wert Matches in Hintergrundbewegungen gefunden werden. 3D-Modelle, welche durch Matches aus den Hintergrundbewegungen der Bilder und nicht dem zu rekonstruierbaren Objekten resultieren, neigen dazu das Endresultat signifikant zu verzerren.

### Übertragung der Medien

Für die Kommunikation und die Datenverwaltung, zwischen der Smartphone- und der Server-Anwendung wurde, eine SQL-Datenbank entwickelt. Smartphone- und Server-Anwendung kommunizieren über eine TCP-Verbindung mit Nachrichten im JSON-Format. Diese Form der Nachrichtenübermittlung wurde gegenüber den anderen Möglichkeiten, wie dem Austausch von HTTP-Nachrichten, gewählt, da damit die stark

limitierten Ressourcen der Smartphones nur sehr wenig belastet werden. Falls die Berechnungen auf dem Smartphone schneller als das Hochladen der Fotos geschehen, werden die Fotos und dazugehörigen Informationen auf dem Smartphone zwischengespeichert. Eine Zwischenspeicherung geschieht ebenfalls, falls ein vorübergehender Verbindungsabbruch zum Server zustande kommt. Eine Umwandlung kann hingegen nur bei Vorhandensein einer Internetverbindung gestartet werden.

Um die Übertragung weiterer Daten, wie den Bildern von der Smartphone Anwendung auf den Server möglichst schnell zu übertragen gibt es folgende Ansätze:

- Bildskalierung ändern: Die Bilder werden auf eine Maximalgröße von 650x650 Pixel herunterskaliert, um eine schnelle Bearbeitung auf dem Smartphone und hochladen der Bilder zu erreichen. Es hat sich gezeigt, dass kleinere Bilder nahezu ungeeignet für eine Rekonstruktion sind. Weiter zeigte die Analyse aus Kapitel 3, dass ein nachträgliches Hochskalieren der Fotos zu keinen Verbesserungen führt.
- Änderung des Bildformates: Eine weitere Möglichkeit wäre das Datenformat oder die Bildqualität zu reduzieren. Es wurde entschieden, JPEG-Bilder ohne Reduzierung der Qualität zu verwenden, da andere Bildformate oder eine Reduktion der Bildqualität das entstehende 3D-Modell gravierend verschlechtert.
- Netzwerkoperationen: Zum Übertragen von Fotos wird die Netzbibliothek Volley im ArrayBuffer Format verwendet, welches zu den schnellsten Verfahren für Android Studio gehört.
- Komprimierung: Die 3D-Modelle und Punktwolken werden vor dem Herunterladen auf der Server-Anwendung in ein ZIP-Dateiformat komprimiert. Durch die Komprimierung ist ein schnelleres Herunterladen möglich. Die Fotos werden nicht vor dem Hochladen komprimiert, da es sich bei JPEG um ein komprimiertes Verfahren handelt und durch weitere Kompressionsverfahren keine Reduzierung zustande kommt.

### Modellierbarkeit des Objektes

Es gibt wenig Möglichkeiten, um zu überprüfen ob sich ein Objekt in ein 3D-Modell umwandeln lässt. Da jeglicher Vorgang, der nicht denselben Merkmalerkennungs-Algorithmus verwendet, nicht in Frage kommt. Dies begründet sich darin, dass die Verfahren eine grobe Einschätzung, aber keine genaue Antwort auf die Umwandelbarkeit des Objektes bieten. Daher wird AKAZE für die Merkmalerkennung auf das dem Nutzer präsentierte Foto angewendet. Nun besteht die Möglichkeit, die Merkmale dem Anwender zu präsentieren, welche jedoch vom Nutzer falsch interpretiert werden könnten. Um das Deuten der gefundenen Merkmale zu vereinfachen, werden nahbeieinanderliegende Merkmale mit einer Linie miteinander verbunden. Merkmale mit wenig oder keinen Verbindungen werden entfernt, da diese eine Rekonstruktion erschweren. Anschließend wird versucht, eine Kontur zu den markierten Merkmalen zu finden, einfarbig gefärbt, und anschließend dem Anwender präsentiert. Je größer die angezeichnete Fläche ist, desto mehr Merkmale wurden darin gefunden. An diesen kann sich der Anwender orientieren, ob das erwünschte Objekt umwandelbar ist und wie weit dieser sich beim Fotografieren vom Objekt entfernt befinden muss.

Wenn ein Problem bei der Fotoerstellung auftritt oder ein Foto nicht hochgeladen werden kann, so wird der Benutzer über den Fehler über die Smartphone-App informiert.

### Selektierung der Fotos auf dem Smartphone

Um nicht alle Fotos auf den Server hochzuladen, werden diese nach der Fotoaufnahme selektiert. Weiter wird überprüft ob bei der Fotoaufnahme, die Smartphone-App im Hoch- oder Querformat befinden. Fotos im Querformat werden nicht hochgeladen, da die Algorithmen Probleme damit haben Fotos mit unterschiedlichen Formaten zu bearbeiten. Um nur Fotos im Hochformat auf die Server-Anwendung hochzuladen wird mittels des Gyroskop-Sensors bestimmt, in welcher Position sich das Smartphone befindet. Sollte sich das Smartphone weniger als  $-90^\circ$  oder mehr als  $+90^\circ$  Grad geneigt werden, so werden die Fotos nicht verarbeitet.

Um zu gewährleisten, dass sich der Anwender nicht zu weit vom Objekt entfernt, wird über GPS die Position des Nutzers bestimmt. Es wird darauf geachtet, dass sich der Anwender nicht weiter als 8 Meter vom Objekt entfernt und es erfolgt eine Mitteilung, falls dies auftreten sollte. Die GPS-Funktionalität kommt nur zum Einsatz, falls es auf dem Smartphone aktiviert wurde und es möglich ist, den Standpunkt des Nutzers zu bestimmen. Zwar wäre es ratsamer, darauf zu achten, dass sich der Nutzer nicht weiter als 3 Meter vom Objekt entfernt, was wiederum aufgrund der Genauigkeit der GPS-Funktion von 5 Meter nicht möglich ist [5]. Ein Nachteil wäre ebenfalls, falls der Nutzer das Objekt in einer Umgebung fotografiert, wo es nicht möglich ist, das Smartphone zu orten. So können trotz aktivierter GPS Ortung Objekte aus einer größeren Distanz als 8 Meter erstellt werden. Wird ein Foto aufgenommen, so wird im ersten Schritt überprüft, ob sich der durchschnittliche RGB-Wert des Fotos zwischen 80 und 220 befindet. Wenn dies nicht der Fall sein sollte, so sind die Fotos häufig über- bzw. unterbelichtet und eignen sich nicht für eine Rekonstruktion.

Im zweiten Selektionsprozess wird überprüft, ob zum dargestellten Foto zwei vorherigen Fotos mindestens eine Matchmenge von 10 Matches gefunden wurde. Ist dies der Fall so, wird das Foto zum Server hochgeladen. Andernfalls falls es kontrastreich und nicht verschwommen ist, aber keine ausreichende Menge an Matches zu den vorherigen Bildern besitzt, wird es auf dem Smartphone zwischengespeichert, um es mit den nachfolgenden Fotos zu vergleichen. Um nahezu identische Aufnahmen zu umgehen, muss zwischen zwei Aufnahmen mindestens 500ms gewartet werden. Dies liegt ebenfalls darin begründet, dass die Kamera für eine Fotoaufnahme fokussiert werden muss. Durch das Fokussieren wird die Aufnahme von unscharfen Bildern verhindert. Fotos welche aufgenommen werden, obwohl die Selektierung des vorherigen Fotos auf der Smartphone-Anwendung noch nicht abgeschlossen ist, werden bis zu ihrer Selektierung im Cache gehalten.

### Selektierung der Fotos auf dem Server

Da dem Server weitere Möglichkeiten für die Selektierung der Fotos zur Verfügung stehen, werden diese in der Serverkomponente dieser Arbeit vorgenommen. Zum einen wird über *ImageMagick* die Helligkeit der Fotos überprüft und zu dunkle oder helle Fotos entfernt. Diese können zwar Matches zu den anderen Bildern besitzen, können jedoch zu Kontrastverlusten bei der Textur des erzeugten Modells führen. Ebenfalls werden nahezu identische Bilder entfernt. Zwar lassen sich viele Matches in diesen finden, werden jedoch durch SfM verworfen. Hierdurch wird die Performance erhöht. Mit den restlichen Fotos wird die Punktwolke oder das 3D-Modell konstruiert. Nach der Konstruktion der Punktwolke wird der Konnektivitätsgraph aller Matches mit den hochgeladenen Fotos verglichen. Dabei werden bis auf die letzten 5 hochgeladenen Fotos alle Fotos entfernt, welche zu ihren 5 vorherigen oder nachfolgenden Fotos keine ausreichende Menge an Matches besitzen. Ab einer Fotomenge von 50 Stück wird der Selektionsalgorithmus auf der Server-Anwendung verändert, indem eine höhere Beziehung der Fotos zu ihren vorherigen oder nachfolgenden Fotos erwartet wird. Dies wird unternommen, weil sich während der Analyse herausgestellt hat, dass die besten 3D-Modelle mit einer Fotomenge zwischen 10 und 50 Stück entstanden sind.

### Bildüberlappung

Für eine Rekonstruktion müssen die Bilder gewisse Voraussetzungen erfüllen (siehe Abschnitt 2.1), wovon eines die Bildüberlappung ist. Um dies während der Fotoaufnahme zu gewährleisten, wird ein Merkmalsvergleich zwischen dem neu aufgenommenen und den vorherigen beiden Bildern ausgeführt. Die gefundenen Merkmale werden mit dem Lowe Ratio-Test und RANSAC selektiert. Übrig gebliebene Merkmale bestehen zum größten Teil aus True Positive Matches. Merkmale mit einem niedrigem Distanzwert werden in der Smartphone-App mittels eines grünen Kreises und Merkmale, mit einem hohem Distanzwert durch einen gelben Kreis kenntlich gemacht. An der Menge der grünen Kreise kann sich der Anwender während der Fotoaufnahme orientieren.

### Menge der Fotos

Es ist notwendig dem Nutzer mitzuteilen, ob ausreichend Fotos erstellt wurden, um daraus ein 3D-Modell herzustellen. Zum einen wurde während der Ausarbeitung ersichtlich, dass mindestens 10 Fotos benötigt werden, damit ein 3D-Modell überhaupt erstellt werden kann. Deshalb wird dem Anwender angezeigt, wie viele Fotos zu den 10 Fotos fehlen. Bei Bedarf ist es dem Anwender frei bis zu 100 zueinanderpassender Fotos für die 3D-Rekonstruktion zu erstellen. Derzeit beträgt die obere Fotomenge 100 Stück, da mehr Fotos häufig das 3D-Modell durch Bewegungen im Hintergrund verschlechtern. Andererseits kann der Anwender über eine Darstellung der Punktwolke auf der Smartphone-Anwendung erahnen wie das 3D-Modell mit den hochgeladenen Bildern ausschauen könnte. Zusätzlich zur Punktwolke wird angezeigt, wo Fotos rund um der Punktwolke fehlen. Hierzu wird in zu jedem Blickwinkel rund um die Punktwolke ein farbiger Punkt angezeigt. Ein roter Punkt deutet darauf hin, dass aus der der Blickrichtung keine Fotos, ein gelber Punkt das exakt ein und ein grüner Punkt mehr als ein Foto vorliegt. An diesen Punkten erkennt der Anwender, wo noch Fotos erstellt werden müssten, um

eine dichtere Punktwolke und somit ein detaillierteres 3D-Modell zu erstellen. Aus welchem Blickwinkel die Fotos gemacht wurden, werden aus den Informationen während der Erstellung der Punktwolke herangezogen. Die Rekonstruktion des 3D-Modells wird dem Nutzer erst ermöglicht, falls mindestens 10 zueinanderpassende Fotos hochgeladen wurden, eine Punktwolke aus den Fotos hergestellt werden kann und das Objekt aus mindestens 3 Blickwinkel fotografiert wurde.

### Restriktionen während der Smartphone Nutzung

Über den Return-Button des Smartphones gelangt der Nutzer der Smartphone-App von jedem Bildschirm aus direkt ins Hauptmenü und der Einsatz der Anwendung ist nur im Hochformat möglich. Selbst wenn das Smartphone geneigt wird, so verbleibt es im Hochformat. Durch diese Maßnahmen werden mögliche Komplikationen vermieden. Weiter wurde die Licht-, Zoom- und Filterfunktionalität bei der Fotoerstellung deaktiviert, da diese Funktionen den Rekonstruktionsprozess durch Beeinträchtigung der Fotoqualität häufig behindern.

### Benutzungsschnittstelle

Beim Start der Smartphone-Anwendung wird das Hauptmenü (siehe Abbildung C.1 links) geöffnet, über welches sich der Anwender durch das gesamte Programm navigieren kann. Die einzelnen Fenster werden im Anhang C beschrieben. In diesem Unterabschnitt wird näher auf das Fotoerstellungsfenster eingegangen, welche die Benutzeroberfläche zum Verwenden der Funktionalitäten bereitstellt. Durch eine Betätigung des Burger Symbols wird eine Symbolleiste geöffnet, welches einen Button für jede in diesem Abschnitt beschriebene Funktionalität bereitstellt. Es können nicht alle Modis gleichzeitig ausgeführt werden, da dies die Rechenleistung der heutigen Smartphones nicht zulässt. Die grafische Benutzeroberfläche für die Smartphone Anwendung wurde in der Programmiersprache Java implementiert, wobei bei rechenintensiven Operationen auf die Programmiersprache C++ und der Bibliothek OpenCV zurückgegriffen wird. Zum Darstellen der Modelle und Punktwolken wird die quelloffene und frei zur Verfügung gestellte Software vom Nutzer Andres Oviedo<sup>4</sup> verwendet. Es wurde für diese Anwendung entschieden, da diese robust ist und kontinuierlich weiterentwickelt wird. Andere betrachtete Viewer (z.B. VES/Kiwi<sup>5</sup>, PCL<sup>6</sup> oder Rajawali<sup>7</sup>) sind veraltet, fehlte eine gradle Anbindung oder benötigen sehr viel Rechenleistung, sodass eine ruckelfreie Verwendung nicht möglich sei.

<sup>4</sup> <https://github.com/andresoviedo/android-3D-model-viewer>

<sup>5</sup> <https://vtk.org/gitweb>

<sup>6</sup> <https://github.com/PointCloudLibrary/pcl>

<sup>7</sup> <https://github.com/Rajawali/Rajawali>

## 4 Realisierung

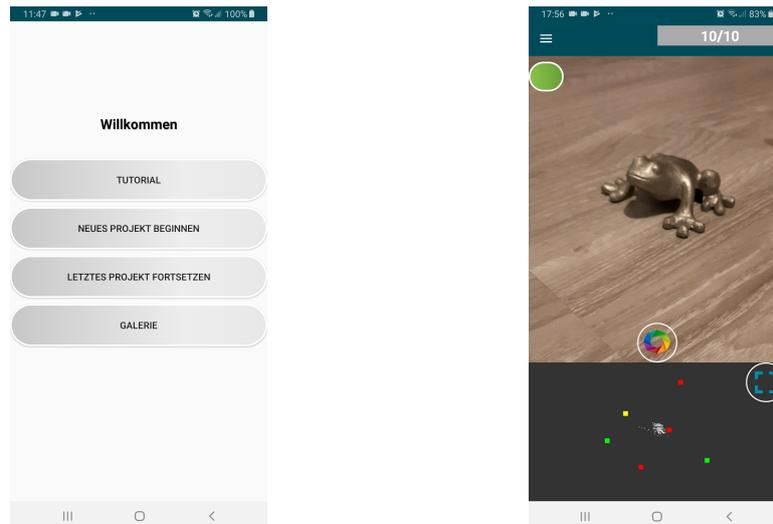


Abbildung 4.1: Links: Hauptmenü auf der Smartphone Anwendung. Rechts: Normaler Aufnahmemodus auf der Smartphone Anwendung.

### 4.2 Hardwareanforderung

Um die Anforderungen der Smartphone- und Server-Anwendung zum Ausführen der entwickelten Anwendungen zu bestimmen, wird die Belastbarkeitsanalyse aus Abschnitt 5.2 und Evaluierung der Echtzeitfähigkeit aus Abschnitt 5.3 verwendet. Falls nicht alle 0,5 Sekunden ein Foto erstellt wird, sollte bereits ein Smartphone mit 75% der Leistung des getesteten Smartphones mit 2,2 GHz/1,6GHz Octa-Core ausreichen, da andernfalls keine ruckelfreie Verwendung garantiert werden kann. Aufgrund dessen, dass viele Prozesse in einzelnen Threads ablaufen, wird ein Prozessor mit einer Aufteilung der Kerne in virtuelle Kerne empfohlen.

Die Hardwareanforderung des Servers orientiert sich daran, ob eine schnelle Rekonstruktion erwünscht ist. Dabei sollte die Prozessorleistung möglichst hoch sein, da diese bei der Belastbarkeitsanalyse nahezu komplett ausgelastet wird. Somit ist ein Prozessor auf dem Server empfehlenswert, welcher mindestens einen Intel Core i7-3770 CPU (3,40 GHz) besitzt. Zu beachten ist außerdem, dass für die Verwendung von Meshroom eine CUDA-fähige mit einem Toolkit Support 2.0 [17] Grafikkarte benötigt wird. Weiter empfehlen die Entwickler von Meshroom mindestens eine Nvidia GeForce 1070 Grafikkarte, weil ein Teil der Berechnungen auf der GPU ausgeführt wird. Eine Auslagerung der Berechnungen auf leistungsschwachen Grafikkarten z. B. auf zwei GeForce 760 Grafikkarten wird von den Algorithmen unterstützt [4].

# 5

## Technische Evaluation und Validierung

Zunächst wird in diesem Abschnitt auf die Ziele der Arbeit in eingegangen, welche durch eine Bewertung der Benutzbarkeit abgerundet wird. Welche mit der wissenschaftlichen Frage der Arbeit nach dem Wie der technischen Realisierung abgerundet wird. Im Unterabschnitt Echtzeitfähigkeit und Belastbarkeit werden folgende Testsysteme verwendet:

System 1:	Windows Server 2016	System 2:	Samsung Galaxy A7 (2018)
	Intel Core i7-3770 CPU (3,40 GHz)		2,2 GHz/1,6 GHz Octa-Core
	32 GB RAM		4 GB RAM
	nVidia GeForce GTX 1660 Ti		
	GPU-Taktrate 1500MHz		
	Arbeitsspeicher: 6GB		

### Ziele der Arbeit

Zu folgenden Zielen aus Abschnitt 1.2 soll die Evaluierung Ergebnisse liefern:

- Bewertung der Benutzbarkeit der Anwendung.
- Der Prototyp stellt unter Erfassung von Echtzeitinformationen einen automatischen Ablauf, zum Rekonstruieren von 3D-Objekte aus Fotos her.
- Ist die Anwendung Echtzeitfähig.
- Eine Bewertung über die Qualität der rekonstruierten 3D-Modelle und Punktwolken.

Die Ziele der Arbeit werden in den folgenden Unterabschnitten diskutiert.

## 5.1 Qualität der erstellten 3D-Modelle

Um die Qualität der von der Anwendung konstruierten 3D-Modelle zu überprüfen, werden 45 Bilder einer Froschfigur mit der Smartphone-Anwendung aufgenommen, welche mit den Rekonstruktionsverfahren Meshroom und OpenMVG, OpenMVE und SMVS auf der Server-Anwendung automatisiert in ein 3D-Modell überführt werden. Das linke Bild aus der Abbildung 5.1 stellt das 3D-Modell nach der Ausführung von Meshroom dar. Für die Darstellung auf den Smartphones wird die Polygonmenge von 174.264 auf 97.778 Stück reduziert und die Oberflächentextur mit MeshLab geglättet. In Abbildung 5.1 rechts ist deutlich zu erkennen, dass sich die Textur des 3D-Modells trotz der Glättung und Reduzierung der Polygonmenge nicht merklich verschlechtert wurde.

Abbildung 5.2 Links ist das Endergebnis von OpenMVG, OpenMVE und SMVS konstruierte 3D-Modell mit SIFT für die Merkmalerkennung und Merkmalsbeschreibung. Abbildung 5.2 Rechts stellt das 3D-Modell dar. Bei dem anstelle des SIFT-Verfahrens, der in dieser Arbeit in OpenMVG erweiterte Implementierung mit AKAZE für die Merkmalerkennung und DAISY für die Merkmalsbeschreibung gemacht wurde. Weiter wurde die Polygonmenge beider 3D-Modelle auf unter 100.000 Stück reduziert und es fand eine Glättung der Modelloberfläche statt. Die 3D-Modelle in Abbildung 5.2 sehen nahezu identisch aus, weshalb an diesem Beispiel keine signifikante Qualitätsverbesserung oder Verschlechterung festgestellt werden konnte.

Das Konstruieren des 3D-Modells von Abbildung 5.1 hat 46,31 Minuten, das SIFT-Verfahren in Abbildung 5.2 hat 148,42 Minuten und durch AKAZE und DAISY hat 109,53 Minuten gedauert. Somit ist die in dieser Masterarbeit implementierte Methode mit AKAZE und DAISY schneller als das von OpenMVG bereitgestellte Verfahren mit SIFT. Außerdem konnten alle von der Server-Anwendung zurückgegebenen Modelle erfolgreich auf der Smartphone-Anwendung betrachtet werden.



Abbildung 5.1: Links: Unverarbeitetes 3D-Modell nach der durch Meshroom ausgeführten 3D-Rekonstruktion. Rechts: 3D Modell nach der Bearbeitung durch die Anwendung MeshLab.



Abbildung 5.2: Links: 3D-Modell nach der Ausführung von OpenMVG, OpenMVE und SMVS mit SIFT für die Merkmalerkennung und Merkmalsbeschreibung. Rechts: 3D-Modell nach der Ausführung von OpenMVG, OpenMVE und SMVS mit AKAZE für die Merkmalerkennung und DAISY für die Merkmalsbeschreibung.

### 5.2 Bewertung der Benutzbarkeit

In diesem Unterabschnitt wird näher auf die gesetzten Ziele aus Abschnitt 1.2 eingegangen, wobei Anmerkungen zu den Funktionen gegeben werden.

#### Automatische Rekonstruktion von 3D-Modellen

Eine automatische Rekonstruktion der Fotos in ein 3D-Modell ist mit den auf dem Server, im Rahmen dieser Arbeit implementierten Funktionen möglich. Dies wird unterstützt durch die aus den Smartphones gewonnene Echtzeitinformationen, Kamerakalibrierung und Selektierung der Fotos, wodurch die Funktionsfähigkeit des Gesamtprozesses gewährleistet wird und mögliche Fehlerfälle umgangen werden.

#### Bedienbarkeit der Anwendung

Für die Verwendung der Anwendung diese Arbeit wurde eine Benutzeroberfläche realisiert, welche eine Einführung über den Menüeintrag 'Tutorial' zur Verfügung stellt. Nicht Gegenstand der Arbeit war eine Benutzeranalyse oder Benutzerevaluierung, wodurch die Bedienbarkeit der Anwendung nicht an Probanden getestet werden konnte.

#### Sicherheit und Authentifikation

Die Analyse zeigt die Notwendigkeit der Implementierung von Sicherheitsfunktionen beim Zugriff auf die persönlichen Daten der Nutzer. Für Anfragen der Dienste wird der *Transport Security Layer* (TLS) für die Kommunikation zwischen der Smartphone- und Server Anwendung verwendet. Diese wird mit HTTPS eingesetzt, welche eine verschlüsselte und abhörsichere Übertragung gewährleistet. Dabei haben die Anwender keinen Zugriff auf die erstellten Fotos. Mit dem Server sind nur Interaktionen mit einer Benutzer-ID und Task-ID im *Universally Unique Identifier* (UUID) Format möglich. Selbst bei einer zweckentfremdeten Verwendung der ID Dritter wird kein Zugriff auf die hochgeladenen Fotos gewährt, da damit nur Fotos hochgeladen und Punktwolken/3D-Modelle betrachtet werden können.

### Konvertierung von Medien

Für eine zukünftige Arbeit wäre eine Konvertierung von Mediendateien notwendig. Zum aktuellen Zeitpunkt werden hochgeladene Fotos nur im JPEG Format akzeptiert und Formate im PNG oder MP4 nach JPEG umgewandelt. Hierzu werden aus einem Video im MP4 Format alle 0,5 Sekunden ein Frame extrahiert, welche im JPEG Format abgespeichert wird. Es wird alle 0,5 Sekunden ein Frame extrahiert, da bei einem geringeren Wert viele identische Frames zustande kommen, welche nicht verwertet werden können.

### Belastbarkeit

Für den Test werden die unterschiedlichen Funktionen ausgeführt und die Prozessorlast, Belastung des Arbeitsspeichers und Netzwerklast sowohl auf dem Smartphone- als auch der Server-Anwendung notiert (siehe Anhang D für die Ergebnisse der Belastbarkeitsanalyse). Betrachten wir zunächst die Testläufe der Smartphoneanwendung. Hinsichtlich der CPU Auslastung bleibt diese selbst unter maximaler Beanspruchung bei knapp 50% oder darunter. Dies sorgt stets für eine ruckelfreie Benutzung der Anwendung. Die Speicherlast ist eine gerade Kurve (mit Ausnahme des Startes), was dem Optimum entspricht. Es zeigt, dass Objekte entfernt werden, sollten diese nicht mehr gebraucht werden und deutet auf eine stabile Speicherlast hin. Die Netzwerklast besitzt einige Unregelmäßigkeiten, was jedoch mit dem häufigen Anfragen des Servers nach Punktwolken oder Nutzerinformationen zu tun hat und zeigt wie der Speicherverbrauch ein zu erwartendes Verhalten. Bis auf die Prozessorlast scheint ebenfalls auf dem Server nichts außergewöhnlich vorzuliegen. Die Prozessorlast ist bei den rechenaufwendigen Prozessen nahezu komplett ausgelastet, was wiederum notwendig ist, um die Fotos möglichst schnell in eine Punktwolke oder 3D-Modell umzuwandeln. Zusammenfassend lässt sich feststellen, dass sich die Anwendung im Testbetrieb stabil zeigt. Für einen Produktivbetrieb sei allerdings zu bedenken, dass die Laborbedingungen keine Grundlast durch nebenläufige Berechnungen oder Netzwerkverkehr besitzt.

### 5.3 Echtzeitfähigkeit

Zum Test der Echtzeitfähigkeit werden 20 zueinanderpassende Fotos mit einer Größe von 135KB auf den Server hochgeladen. Die Fotos werden einzeln hochgeladen, sobald alle vorherigen Berechnung abgeschlossen wurden. Dabei wird die Zeit vom Beginn des Hochladens des jeweiligen Fotos bis hin zum Herunterladen der Punktwolke gemessen. Die Ergebnisse (siehe Tabelle 5.3) zeigen, dass die Berechnung eines Fotos im Schnitt zwischen 1-2 Sekunden benötigt. Selbst das Erhöhen der Testfälle sollte nicht das Resultat ändern, weil stets die Ergebnisse der vorherigen Berechnung weiterverwendet werden, die Bildergröße unveränderbar ist und der Merkmalvergleich jedes Fotos nur mit den 5 zuvor hochgeladenen Fotos ausgeführt wird.

Das Ergebnis der Echtzeitfähigkeit zeigt, dass das System keine Echtzeitfähigkeit im Sinne der zeitlichen Wahrnehmung eines Menschen von 3 Sekunden erfüllt. Dies liegt jedoch weniger an der Smartphone- oder Server-Anwendung begründet, sondern vielmehr am Übertragungsweg der Daten. Da bis auf wenige Ausnahmen die Berechnungen

auf der Server-Anwendung innerhalb der zielgesetzten Echtzeit von 3 Sekunden erfolgt. Somit sollte es möglich sein, eine Echtzeitfähigkeit der entwickelten Anwendung durch Veränderung der Übertragungswege herzustellen (z. B. mittels Umstellung des mobilen Internets auf 5G).

Anzahl der Fotos	Gesamtdauer in Minuten	Dauer der Berechnung auf dem Server in Minuten
2	01,646	00,101
3	03,966	00,671
4	06,978	00,576
5	04,352	01,069
6	04,537	01,095
7	06,248	01,535
8	06,072	01,336
9	04,034	00,910
10	06,407	03,626
11	06,054	01,724
12	06,430	02,398
13	07,143	00,892
14	09,518	02,017
15	07,009	01,406
16	07,569	01,629
17	05,889	02,284
18	10,453	01,590
19	06,643	01,434
20	05,301	01,251

Tabelle 5.3: Auswertung der Bearbeitungsdauer der sequenziell hochgeladenen Fotos. Die Fotos sind durchschnittlich 135KB und die herunterzuladene Punktwolke maximal 15KB groß.

## 5.4 Ergebnis der Evaluation

Die Evaluation zeigt, dass sich der Prototyp technisch als stabil und robust erweist. Einzige Ausnahme bildet hierbei die Echtzeitfähigkeit, welche durch eine Veränderung der Übertragungswege erfüllt werden kann. Weiter hat sich gezeigt, dass es möglich ist, die in der Smartphone-Anwendung erstellten Fotos automatisiert in ein 3D-Modell zu überführen und problemlos auf der Smartphone-Anwendung darstellbar zu machen. Ungeachtet der technischen Sicht ist der Prototyp noch an verschiedenen Stellen (siehe Abschnitt 6.2) erweiterungsfähig und es bedarf einer Nutzer- und Langzeitevaluation.

# 6

## Zusammenfassung

Im Rahmen dieser Masterarbeit wurde ein System entwickelt, welches sich aus drei Softwarekomponenten zusammensetzt.

Einer Smartphone Anwendung, welche als Benutzungsschnittstelle dient und eine personalisierte Nutzung bei der Fotoerstellung ermöglicht.

Rechenaufwendige Operationen werden auf einen Server ausgelagert, wo Merkmale der Bilder dazu verwendet werden, Bildinhalte eindeutig zu beschreiben, miteinander zu vergleichen, in eine Punktwolke zu überführen und abschließend aus der Punktwolke ein 3D-Modell herzustellen.

Für den Datenaustausch zwischen der Smartphone- und der Serveranwendung wird eine SQL-Datenbank verwendet. Der Informationsaustausch geschieht über JSON Dateien über HTTPS.

Um die bestmöglichen Ergebnisse zu erhalten, wird eine ausführliche Analyse betrachtet, die Verfahren anhand von wissenschaftlichen Arbeiten und eigenen Testverfahren untersucht. Als Ergebnis kam heraus, dass Meshroom schnelle und guten Punktwolken oder 3D-Modelle erstellt. Wenn eine Rekonstruktion mit Meshroom nicht möglich ist, hat sich die Kombination von OpenMVG, OpenMVE in SMVS als Alternative bestätigt. Unter den Algorithmen haben sich AKAZE und DAISY als äußerst positiv hervorgetan, weswegen diese in die Anwendung OpenMVG mithilfe von OpenCV im Rahmen dieser Arbeit integriert wurden. Nach der Fertigstellung der Software wurden diese abschließend durch eine Evaluation bewertet, um technische und gestalterische Mängel zu testen.

Ausschlaggebend für die Entwicklung der Applikation ist die Frage, inwiefern eine Echtzeitanwendung gemäß vorhandener Augment-Reality-Applikationen basierend auf den Matching-Prozessen möglich ist. Dies sind bereits aufgrund des entstehenden Datenverkehrs durch häufiges Hochladen der Bilder und der Latenzen bei einer ggf. schlechten Internetverbindung nicht denkbar. Die in dieser Arbeit eingesetzten Techniken sind zukunftsorientiert, weil die Anwendungen eine große daran arbeitende Gemeinschaft besitzen und aufgrund des offenen für jeden zugänglichen Quellcodes diese jederzeit erweitert werden können. Aufgrund der losen Kopplung des Systems ist eine dynamische Lastverteilung auf mehrere Serversysteme denkbar. Abschließend sei gesagt, dass die Smartphone- und die Serveranwendung prototypisch entwickelt wurden.

Dieser Abschnitt stellt eine Diskussion zu der wissenschaftlichen Frage dieser Masterarbeit her. Zusätzlich werden offene Punkte und Verbesserungen genannt, die an den Prototypen in einer zukünftigen Arbeit realisiert werden könnten.

### 6.1 Diskussion über die 3D-Rekonstruktion des entwickelten Prototyps

Die zentrale Frage dieser Arbeit ist die Frage nach dem Wie der technologischen Realisierung. Aktuell gibt es mehrere Anwendungen (z.B. Meshroom), die eine Benutzerschnittstelle realisiert haben und dem Anwender bei der Erzeugung von 3D-Modellen unterstützen. Diese Anwendungen besitzen jedoch viele Einstellungsmöglichkeiten, welche an das anzuwendende Szenario angepasst werden müssen, wodurch der Nutzer Grundkenntnisse in der Computer Vision benötigt. Weiter benötigen die Anwendungen eine lange Berechnungszeit, da die Bilder hierbei nicht vorselektiert werden. Wohingegen Anwendungen, welche eine Echtzeitrekonstruktion ermöglichen nur für einen Smartphone typ entwickelt werden. Das Ziel dieses Prototyps liegt darin, eine Forschungsfrage zu beantworten, weshalb diese nicht öffentlich zugänglich gemacht werden, was eine Beurteilung der implementierten Algorithmen nahezu nicht möglich macht.

In dieser Arbeit wird auf Technologien gesetzt, die sich in der Analyse für einen prototypischen Betrieb als geeignet erwiesen haben. Die realisierte Anwendung lagert Rechenaufwendige Operationen aus und schafft es 3D-Modelle in kürzester Zeit zu erstellen. Dabei wird der Anwender während der Fotoaufnahme mittels Hilfestellungen an die Hand genommen und durch Echtzeitinformationen erfolgt eine Selektierung von nicht erwünschten Bildern. Das Wie beantwortet sich folglich durch den prozessualen Ablauf der 3D-Rekonstruktion und die Parametrisierung für den mobilen Nutzungskontext sowie die Verteilung der Systeme. Aus der Evaluierung lässt sich das Erreichen der Zielsetzung mit Ausnahme der Echtzeitfähigkeit aus Abschnitt 1.2 feststellen.

### 6.2 Ausblick und offene Punkte

Aufgrund des aktuellen Standes der Anwendung ergeben sich zahlreiche Möglichkeiten, diese in einer zukünftigen Arbeit zu erweitern. Im nachstehenden Abschnitt werden einige Verbesserungen betrachtet.

#### Weitere Analysen

In der vorliegenden Arbeit wurden die Verfahren anhand eines Testbildes der Teufelsstatue in Lübeck ausgeführt. Dabei gibt es Objekte, die anderen Objekteigenschaften aufweisen, welche es noch zu testen gibt. Somit ist es denkbar, dass das hier genutzte Verfahren zwar die bestmöglichen Ergebnisse auf Objekten mit einer rauen Oberfläche erzielen, aber es sein kann, dass es ein besseres Verfahren für glatte oder Transparente Oberfläche gibt.

## SLAM und Visual Odometry

Visual Odometry ist der Prozess zum Bestimmen der Position und Orientierung eines Roboters auf Basis der Veränderungen auf Videos. Bei der Visual Odometry handelt es sich um eine Teilmenge der SfM Verfahren, in der nicht über die Merkmale in den Bildern, sondern aus allen Informationen aus einem Video direkt eine Rekonstruktion der Umgebung vorgenommen wird. Dabei werden die Prozesse vom SfM Verfahren um die Algorithmen robust minimal-set-solver (z.B. RANSAC), point triangulation, camera-pose-optimisation (z.B. Powells method, Levenberg Marquardt) und Loop Closure erweitert. Während SfM ein Objekt anhand einer Ansammlung an unsortierten Bildern behandelt, deckt die Visual Odometry einen gezielten Bereich mittels Videos ab. Visual Odometry behandelt die Lokale Konsistent (nicht geschlossener Bereich), SLAM hingegen die globale Konsistent (geschlossener Bereich) einer Umgebung.

Mit SLAM oder Visual Odometry könnte die Umgebung um das Objekt eingefangen werden. Aus den Umgebungsinformationen eines Objektes lässt sich das hier betrachtete Kamera- in das Weltkoordinatensystem überführen, welche für die Herstellung von detailreichereren Modellen genutzt werden könnte. Außerdem ließe sich Visual Odometry oder SLAM in Kombination mit dem hier entwickelten Verfahren zum Rekonstruieren von 3D-Modellen ebenfalls in anderen Bereichen Nutzen. So wäre es denkbar einem Roboterarm Objekte im dreidimensionalen Raum besser greifbar zu machen. Indem dem Roboterarm das Objekt dreidimensional im Weltkoordinatensystem unter Zuhilfenahme der Umgebungsinformationen übergeben wird.

## Literatur

- [1] Agisoft *Useful Tips on Image Capture: How to Get an Image Dataset that Meets Metashape Requirements?* 2014. URL: [https://www.agisoft.com/pdf/tips\\_and\\_tricks/Image%20Capture%20Tips%20-%20Equipment%20and%20Shooting%20Scenarios.pdf](https://www.agisoft.com/pdf/tips_and_tricks/Image%20Capture%20Tips%20-%20Equipment%20and%20Shooting%20Scenarios.pdf) (besucht am 30. 09. 2019).
- [2] Alahi, A., Ortiz, R. und Vanderghenst, P. Freak: Fast retina keypoint. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee. 2012, S. 510–517.
- [3] Alcantarilla, P. F. und Solutions, T. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In: *IEEE Trans. Patt. Anal. Mach. Intell* 34(7):1281–1298, 2011.
- [4] AliceVision *GPU usage*. 2019. URL: <https://github.com/alicevision/meshroom/wiki/GPU-usage> (besucht am 18. 10. 2019).
- [5] Annex, A. *Global Positioning System Standard Positioning Service Signal Specification*. 2008.
- [6] Babbar, G., Bajaj, P., Chawla, A. und Gogna, M. Comparative study of image matching algorithms. In: *International Journal of Information Technology and Knowledge Management* 2(2):337–339, 2010.
- [7] Barnes, C., Shechtman, E., Finkelstein, A. und Goldman, D. B. PatchMatch: A randomized correspondence algorithm for structural image editing. In: *ACM Transactions on Graphics (ToG)*. Bd. 28. 3. ACM. 2009, S. 24.
- [8] Bay, H., Ess, A., Tuytelaars, T. und Van Gool, L. Speeded-up robust features (SURF). In: *Computer vision and image understanding* 110(3):346–359, 2008.
- [9] Bayraktar, E. und Boyraz, P. Analysis of feature detector and descriptor combinations with a localization experiment for various performance metrics. In: *arXiv preprint arXiv:1710.06232*, 2017.
- [10] Bian, J., Yang, R., Liu, Y., Zhang, L., Cheng, M.-M., Reid, I. und Wu, W. *Match-Bench: An Evaluation of Feature Matchers*. Aug. 2018.
- [11] Bouck-Standen, D., Ohlei, A., Daibert, V., Winkler, T. und Herczeg, M. NEMO Converter 3D: Reconstruction of 3D Objects from Photo and Video Footage for Ambient Learning Spaces. In: *AMBIENT*:6–12, 2017.
- [12] Bouck-Standen, D., Ohlei, A., Höffler, S., Daibert, V., Winkler, T. und Herczeg, M. Reconstruction and Web-based Editing of 3D Objects from Photo and Video Footage for Ambient Learning Spaces. In: *International Journal On Advances in Intelligent Systems* 11, 2018.
- [13] Bouck-Standen, D., Ohlei, A., Höffler, S., Daibert, V., Winkler, T. und Herczeg, M. Reconstruction and Web-based Editing of 3D Objects from Photo and Video Footage for Ambient Learning Spaces. In: *International Journal On Advances in Intelligent Systems* 11, 2018.
- [14] Burger, W. und Burge, M. J. *Digitale Bildverarbeitung: Eine algorithmische Einführung mit Java*. 3. Aufl. Berlin: Springer Vieweg, 2015. ISBN: 978-3-642-04603-2.

- [15] Calonder, M., Lepetit, V., Strecha, C. und Fua, P. Brief: Binary robust independent elementary features. In: *European conference on computer vision*. Springer, 2010, S. 778–792.
- [16] Corporation, I., Garage, W. und Itseez *Descriptor Matchers*. 2015. URL: [https://docs.opencv.org/master/db/d39/classcv\\_\\_1\\_\\_1DescriptorMatcher.html](https://docs.opencv.org/master/db/d39/classcv__1__1DescriptorMatcher.html) (besucht am 18. 10. 2019).
- [17] Corporation, N. *CUDA Toolkit Archive*. 2019. URL: <https://developer.nvidia.com/cuda-toolkit-archive> (besucht am 18. 10. 2019).
- [18] Dietrich, J. Applications of structure-from-motion photogrammetry to fluvial geomorphology. In: 2015.
- [19] Fischler, M. A. und Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: *Communications of the ACM* 24(6):381–395, 1981.
- [20] Furukawa, Y. und Hernández, C. Multi-View Stereo: A Tutorial. In: *Found. Trends. Comput. Graph. Vis.* 9(1-2):1–148, Juni 2015. ISSN: 1572-2740. DOI: 10.1561/06000000052. URL: <http://dx.doi.org/10.1561/06000000052>.
- [21] Ghosh, P., Pandey, A. und Pati, U. C. Comparison of different feature detection techniques for image mosaicing. In: *ACCENTS Transactions on Image Processing and Computer Vision* 1(1):1–7, 2015.
- [22] Haralick, R. M., Lee, C.-N., Ottenburg, K. und Nölle, M. Analysis and solutions of the three point perspective pose estimation problem. In: *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*:592–598, 1991.
- [23] Harris, C. G., Stephens, M. u. a. A combined corner and edge detector. In: *Alvey vision conference*. Bd. 15. 50. Citeseer, 1988, S. 10–5244.
- [24] Hartley, R. und Zisserman, A. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [25] Hassaballah, M., Abdelmgeid, A. A. und Alshazly, H. A. Image features detection, description and matching. In: *Image Feature Detectors and Descriptors*. Springer, 2016, S. 11–45.
- [26] James, W., Burkhardt, F., Bowers, F. und Skrupskelis, I. K. *The principles of psychology*. Bd. 1. 2. Macmillan London, 1890.
- [27] Kazhdan, M., Bolitho, M. und Hoppe, H. Poisson surface reconstruction. In: *Proceedings of the fourth Eurographics symposium on Geometry processing*. Bd. 7. 2006.
- [28] Kazhdan, M., Klein, A., Dalal, K. und Hoppe, H. Unconstrained isosurface extraction on arbitrary octrees. In: *Symposium on Geometry Processing*. Bd. 7. 2. 2007.
- [29] Khatamian, A. und Arabnia, H. R. Survey on 3D Surface Reconstruction. In: *Journal of Information Processing Systems* 12(3), 2016.
- [30] Kim, S., Rhee, S. und Kim, T. Digital Surface Model Interpolation Based on 3D Mesh Models. In: *Remote Sensing* 11(1):24, 2019.
- [31] Komorowski, J., Czarnota, K., Trzcinski, T., Dabala, L. und Lynen, S. Interest point detectors stability evaluation on ApolloScape dataset. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, S. 0–0.
- [32] Leutenegger, S., Chli, M. und Siegwart, R. BRISK: Binary robust invariant scalable keypoints. In: *2011 IEEE international conference on computer vision (ICCV)*. Ieee, 2011, S. 2548–2555.

- [33] Lowe, D. G. Distinctive image features from scale-invariant keypoints. In: *International journal of computer vision* 60(2):91–110, 2004.
- [34] Lukas, F. „Structure from Motion“ in der Praxis 3D-Visualisierung mittels Digitalfotos. In: *Netzpublikationen zur Grabungstechnik*, 2015.
- [35] Miksik, O. und Mikolajczyk, K. Evaluation of local detectors and descriptors for fast feature matching. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE. 2012, S. 2681–2684.
- [36] Mordvintsev, A. und Abid, K. *Useful Tips on Image Capture: How to Get an Image Dataset that Meets Metashape Requirements?* 2014. URL: [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_feature2d/py\\_features\\_meaning/py\\_features\\_meaning.html#features-meaning](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_features_meaning/py_features_meaning.html#features-meaning) (besucht am 09. 09. 2019).
- [37] Muja, M. und Lowe, D. G. Scalable Nearest Neighbor Algorithms for High Dimensional Data. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36, 2014.
- [38] Ondrůška, P., Kohli, P. und Izadi, S. Mobilefusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones. In: *IEEE transactions on visualization and computer graphics* 21(11):1251–1258, 2015.
- [39] Otepka, J., Ghuffar, S., Waldhauser, C., Hochreiter, R. und Pfeifer, N. Georeferenced point clouds: A survey of features and point cloud management. In: *ISPRS International Journal of Geo-Information* 2(4):1038–1065, 2013.
- [40] Peng, Z. Efficient matching of robust features for embedded SLAM. Magisterarb. 2012.
- [41] Pradeep, V., Rhemann, C., Izadi, S., Zach, C., Bleyer, M. und Bathiche, S. Mono-Fusion: Real-time 3D reconstruction of small scenes with a single web camera. In: Okt. 2013, S. 83–88. DOI: 10.1109/ISMAR.2013.6671767.
- [42] Rosten, E. und Drummond, T. Machine learning for high-speed corner detection. In: *European conference on computer vision*. Springer. 2006, S. 430–443.
- [43] Rosten, E., Porter, R. B. und Drummond, T. Faster and Better: A Machine Learning Approach to Corner Detection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32:105–119, 2010.
- [44] Rublee, E., Rabaud, V., Konolige, K. und Bradski, G. R. ORB: An efficient alternative to SIFT or SURF. In: *ICCV*. Bd. 11. 1. Citeseer. 2011, S. 2.
- [45] Sankaran, K., Zhu, M., Guo, X. F., Ananda, A. L., Chan, M. C. und Peh, L.-S. Using mobile phone barometer for low-power transportation context detection. In: *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. ACM. 2014, S. 191–205.
- [46] Shi, J. und Tomasi, C. Good Features to Track. In: *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 600, März 2000. DOI: 10.1109/CVPR.1994.323794.
- [47] Snavely, N., Simon, I., Goesele, M., Szeliski, R. und Seitz, S. M. Scene reconstruction and visualization from community photo collections. In: *Proceedings of the IEEE* 98(8):1370–1390, 2010.
- [48] Szeliski, R. Computer Vision: Algorithms and Applications. In: *Instructor* 201901, 2019.

- [49] Tanskanen, P., Kolev, K., Meier, L., Camposeco, F., Saurer, O. und Pollefeys, M. Live metric 3d reconstruction on mobile phones. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, S. 65–72.
- [50] Tareen, S. A. K. und Saleem, Z. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In: *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. IEEE. 2018, S. 1–10.
- [51] Tola, E., Lepetit, V. und Fua, P. Daisy: An efficient dense descriptor applied to wide-baseline stereo. In: *IEEE transactions on pattern analysis and machine intelligence* 32(5):815–830, 2009.
- [52] Vinken, G. und Franz, B. *Das Digitale und die Denkmalpflege : Bestandserfassung-Denkmalvermittlung-Datenarchivierung-Rekonstruktion verlorener Objekte : Chancen und Grenzen im Einsatz digitaler Technologien : Veränderungen in der Praxis von Denkmalpflege und Kulturgutsicherung*. Holzminden, 2017. URL: <http://books.ub.uni-heidelberg.de/arthistoricum/catalog/book/263>.
- [53] Wiemann, T., Annuth, H., Lingemann, K. und Hertzberg, J. An evaluation of open source surface reconstruction software for robotic applications. In: *2013 16th International Conference on Advanced Robotics (ICAR)*. IEEE. 2013, S. 1–7.

# A

## Anhang A: Analyse der Korrespondenzverfahren anhand wissenschaftlicher Veröffentlichungen

In diesem Anhang werden die Ergebnisse der Testverfahren für die Merkmalerkennung, Merkmalsbeschreibung und Merkmalsvergleichsverfahren anhand von wissenschaftlichen Arbeiten untersucht. Die Werte für die Sparte 'Gewichtung der eigenen Analyse' können Tabelle 3.3 entnommen werden. Das Speichern der Informationen kann hierbei mittels Vektoren von Fließkommazahlen oder Bitstrings geschehen. Die Art der Speicherung der Informationen ist hierbei von besonderer Relevanz, da sich dadurch die Algorithmen nicht beliebig mischen lassen. Die Art der Speicherung wird in Tabelle 3.2 notiert. Tabelle A.1 bis Tabelle A.3 untersuchen die Merkmalerkennungs-Verfahren. Tabelle A.4 und Tabelle A.5 die Merkmalsbeschreibungs-Verfahren, wobei einheitlich Surf für die Merkmalerkennung verwendet wird. Um die Menge der Testverfahren repräsentativ zu halten, werden die Ergebnisse anhand einer Skalierung von 1 bis 3 gewichtet. Ein Wert von 1 wird bei auffallend negativen Ergebnissen und 3 bei positiv hervorzuhebenden vergeben. Sollte kein Wert angegeben sein, so wird der Algorithmus in dem Bereich entweder nicht getestet oder dies ist aus Kompatibilitätsgründen nicht möglich.

## Bewertung der Merkmalerkennungsverfahren:

Merkmals- erkennungs- verfahren	Gewichtung der eigene Analyse	Gewichtung nach [31]	Gewichtung nach [50]	Gewichtung nach [40]	Durchschnitt
SURF	3		3	2	2,6
SIFT	2		3	2	2,3
ORB	3	2	1	3	2,25
BRISK	3		2	3	2.6
AKAZE	3	2	3		2.6
FAST	3	3		3	3
MSER	2				2
Harris Corner	2				2
Shi-Thomasi	2				2

Tabelle A.1: Subjektive Bewertung der Wiederholpräzision der Merkmalerkennungsverfahren mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen und 3 bei positiven vergeben. Wenn ein Verfahren nicht getestet wurde, wird kein Wert vergeben.

Merkmals- erkennungs- verfahren	Gewichtung der eigene Analyse	Gewichtung nach [35]	Durchschnitt
SURF	2	2	2
SIFT	3		3
ORB	1	1	1
BRISK	1	2	1,5
AKAZE	3		3
FAST	1	3	2
MSER	3	3	3
Harris Corner	1		1
Shi-Thomasi	1		1

Tabelle A.2: Subjektive Bewertung der Anzahl an gefundenen Merkmalen der Merkmalerkennungsverfahren mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen und 3 bei positiven vergeben. Wenn ein Verfahren nicht getestet wurde, wird kein Wert vergeben.

A Anhang A: Analyse der Korrespondenzverfahren anhand wissenschaftlicher Veröffentlichungen

Merkmals- erkennung- verfahren	Gewichtung der eigene Analyse	Gewichtung nach [50]	Gewichtung nach [35]	Durchschnitt
SURF	2	2		2
SIFT	1	1	1	1
ORB	1	3	3	2,3
BRISK	2	3	3	2,6
AKAZE	3	1		2
FAST	3		3	3
MSER	3		2	2,5
Harris Corner	2			2
Shi-Thomasi	2			2

Tabelle A.3: Subjektive Bewertung der Laufzeit der Merkmalerkennungsverfahren mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen und 3 bei positiven vergeben. Wenn ein Verfahren nicht getestet wurde, wird kein Wert vergeben.

### Testen der Merkmalsbeschreibungsverfahren:

Merkmals- beschreibungs- verfahren	Gewichtung der eigene Analyse	Gewichtung nach [35]	Gewichtung nach [9]	Durchschnitt
SURF	1	2	3	2
SIFT	2	2	2	2
ORB		1	1	1
BRISK		3	2	2,5
FREAK			3	3
BRIEF		2	1	1,5
DAISY	3			3
AKAZE	3			3

Tabelle A.4: Subjektive Bewertung der Genauigkeit der Merkmalsbeschreibungsverfahren mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen und 3 bei positiven vergeben. Wenn ein Verfahren nicht getestet wurde, wird kein Wert vergeben.

A Anhang A: Analyse der Korrespondenzverfahren anhand wissenschaftlicher  
Veröffentlichungen

Merkmals- beschreibungs- verfahren	Gewichtung eigene Analyse	Gewichtung nach [21]	Gewichtung nach [9]	Gewichtung nach [40]	Durchschnitt
SURF	2	1	1	2	1,25
SIFT	1		1	1	1
ORB	3		3	3	3
BRISK	2		2	3	2,3
FREAK	3	3			3
BRIEF	3		3	1	2
DAISY	1				1
AKAZE	3		2	2	2,3

Tabelle A.5: Subjektive Bewertung der Laufzeit der Merkmalsbeschreibungsverfahren mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen und 3 bei positiven vergeben. Wenn ein Verfahren nicht getestet wurde, wird kein Wert vergeben.

# B

## Anhang B: Analyse der Korrespondenzverfahren anhand von Invarianten

Im Folgenden werden die Invarianzen Dauer, Wiederholpräzision und Genauigkeit der Rekonstruktionsprozesse unter Veränderung der Skalierung, Kontrast, Rotation oder Helligkeit betrachtet. Hierzu werden zwei Bilder verwendet, wovon eines anhand der Invarianten verändert wird. Bei der Analyse der Merkmalsbeschreibung wird einheitlich AKAZE für die Merkmalserkennung verwendet, da die Untersuchung der Merkmalsbeschreibung-Verfahren (siehe Anhang A) zeigt, dass es die höchste Menge an zueinanderpassenden Matches aufweist. Bei jeder Auswertung werden die Ergebnisse der Verfahren Brute-Force und FLANN notiert.

Zu jeder Invarianz wurden mehrere Einstellungen angewendet, um die bestmögliche Ergebnisse zu erhalten. Am Beispiel der Rotation wurde das Testbild 12 mal um  $30^\circ$  im Uhrzeigersinn gedreht und die Ergebnisse jeder Berechnung niedergeschrieben. Um eine bessere Übersicht der gemachten Testverfahren zu erhalten wurden die Ergebnisse zusammengefasst und mit den Werten 1-3 Gewichtet. Ein hoher Wert wird bei einer äußerst positiven und ein niedriger Wert bei einem auffallend negativen Ergebnis vergeben.

## Bewertung der Detektions-Verfahren

Merkmals- vergleichs- verfahren	BruteForce			FLANN		
	Wiederhol- präzision	Genauigkeit	Dauer	Wiederhol- präzision	Genauigkeit	Dauer
SURF	3	2	2	3	3	3
SIFT	2	3	1	2	2	2
ORB	3	1	1	3	3	3
BRISK	2	1	3			
AKAZE	3	3	3	3	3	3
FAST						
MSER	2	2	3	2	2	2
KAZE	3	1	1	2	3	1
Harris-corner	2	1	3	3	1	3
Shi-Thomasi	2	1	3	3	1	3

Tabelle B.1: Subjektive Bewertung der Merkmalerkennungs-Verfahren durch Veränderung der Helligkeit, mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen, 3 bei positiven und kein Wert bei fehlenden Ergebnissen vergeben. Spalten 2 bis 4 werden mit BruteForce und 5 bis 7 mit FLANN als Merkmalsvergleich ausgeführt.

Merkmals- vergleichs- verfahren	BruteForce			FLANN		
	Wiederhol- präzision	Genauigkeit	Dauer	Wiederhol- präzision	Genauigkeit	Dauer
SURF	3	2	3	3	3	2
SIFT	3	2	1	2	3	1
ORB	3	3	2	3	2	3
BRISK	3	3	1	3	3	3
AKAZE	3	3	3	3	2	3
FAST	3	3	1	3	3	3
MSER	2	3	3	2	3	2
KAZE	3	1	1	2	1	1
Harris-corner	2	1	3	3	1	3
Shi-Thomasi	2	1	3	3	1	3

Tabelle B.2: Subjektive Bewertung der Merkmalerkennungs-Verfahren durch Veränderung des Kontrastes, mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen, 3 bei positiven und kein Wert bei fehlenden Ergebnissen vergeben. Spalten 2 bis 4 werden mit BruteForce und 5 bis 7 mit FLANN als Merkmalsvergleich ausgeführt.

B Anhang B: Analyse der Korrespondenzverfahren anhand von Invarianten

Merkmalsvergleichsverfahren	BruteForce			FLANN		
Merkmalserkennungsverfahren	Wiederholpräzision	Genauigkeit	Dauer	Wiederholpräzision	Genauigkeit	Dauer
SURF	3	2	3	3	2	3
SIFT	2	2	2	2	2	1
ORB	3	3	2	3	3	3
BRISK	3	3	3	3	3	3
AKAZE	3	2	3	3	3	3
FAST	3	3	3	3	2	3
MSER	2	3	3	3	3	3
KAZE	1	1	1	1	1	1
Harris-corner	3	1	3	3	1	3
Shi-Thomasi	3	1	3	3	1	3

Tabelle B.3: Subjektive Bewertung der Merkmalserkennungs-Verfahren durch Veränderung der Skalierung, mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen, 3 bei positiven und kein Wert bei fehlenden Ergebnissen vergeben. Spalten 2 bis 4 werden mit BruteForce und 5 bis 7 mit FLANN als Merkmalsvergleich ausgeführt.

Merkmalsvergleichsverfahren	BruteForce			FLANN		
Merkmalserkennungsverfahren	Wiederholpräzision	Genauigkeit	Dauer	Wiederholpräzision	Genauigkeit	Dauer
SURF	3	3	2	2	2	2
SIFT	3	2	1	3	3	1
ORB						
BRISK	3	1	2	1	1	3
AKAZE	3	3	3	3	3	3
FAST	3	1	3			
MSER	3	3	3	2	2	2
KAZE						
Harris-corner	3	1	3	3	1	3
Shi-Thomasi	3	1	3	3	1	3

Tabelle B.4: Subjektive Bewertung der Merkmalserkennungs-Verfahren durch Veränderung der Ausrichtung, mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen, 3 bei positiven und kein Wert bei fehlenden Ergebnissen vergeben. Spalten 2 bis 4 werden mit BruteForce und 5 bis 7 mit FLANN als Merkmalsvergleich ausgeführt.

## Bewertung der Merkmal Beschreibungs-Algorithmen

Merkmals- vergleichs- verfahren	BruteForce		FLANN	
	Genauigkeit	Dauer	Genauigkeit	Dauer
SURF	2	2	1	2
SIFT	3	1	2	1
ORB				
BRISK	2	2	3	2
BRIEF				
FREAK	2	3	3	3
Akaze	3	3	3	3
DAISY	3	1	3	2

Tabelle B.5: Subjektive Bewertung der Merkmalsbeschreibungs-Verfahren durch Veränderung der Helligkeit, mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen, 3 bei positiven und kein Wert bei fehlenden Ergebnissen vergeben. Spalten 2 bis 4 werden mit BruteForce und 5 bis 7 mit FLANN als Merkmalsvergleich ausgeführt.

Merkmals- vergleichs- verfahren	BruteForce		FLANN	
	Genauigkeit	Dauer	Genauigkeit	Dauer
SURF	2	2	3	2
SIFT	2	1	3	1
ORB				
BRISK	3	3	1	3
BRIEF				
FREAK	1	3	1	3
Akaze	3	3	3	3
DAISY	3	1	3	2

Tabelle B.6: Subjektive Bewertung der Merkmalsbeschreibungs-Verfahren durch Veränderung des Kontrastes, mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen, 3 bei positiven und kein Wert bei fehlenden Ergebnissen vergeben. Spalten 2 bis 4 werden mit BruteForce und 5 bis 7 mit FLANN als Merkmalsvergleich ausgeführt.

B Anhang B: Analyse der Korrespondenzverfahren anhand von Invarianten

Merkmals- vergleichs- verfahren	BruteForce		FLANN	
	Genauigkeit	Dauer	Genauigkeit	Dauer
SURF				
SIFT	3	1	3	1
ORB				
BRISK	1	3		
BRIEF				
FREAK				
Akaze	1	3	2	3
DAISY	3	1	3	3

Tabelle B.7: Subjektive Bewertung der Merkmalsbeschreibungs-Verfahren durch Veränderung der Skalierung, mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen, 3 bei positiven und kein Wert bei fehlenden Ergebnissen vergeben. Spalten 2 bis 4 werden mit BruteForce und 5 bis 7 mit FLANN als Merkmalsvergleich ausgeführt.

Merkmals- vergleichs- verfahren	BruteForce		FLANN	
	Genauigkeit	Dauer	Genauigkeit	Dauer
SURF	1	2	3	3
SIFT	3	1	3	1
ORB				
BRISK	2	3	2	3
BRIEF	1	3	1	3
FREAK				
Akaze	2	3	2	3
DAISY	3	1	3	3

Tabelle B.8: Subjektive Bewertung der Merkmalsbeschreibungs-Verfahren durch Veränderung der Ausrichtung, mit einem Wert von 1 bis 3. Ein Wert von 1 wurde bei auffallend negativen, 3 bei positiven und kein Wert bei fehlenden Ergebnissen vergeben. Spalten 2 bis 4 werden mit BruteForce und 5 bis 7 mit FLANN als Merkmalsvergleich ausgeführt.

# C

## Anhang C: Grafische Benutzeroberfläche der Smartphone-Anwendung

Beim Start der Smartphone-Anwendung wird das Hauptmenü (siehe Abbildung C.1 links) geöffnet, über welches sich der Anwender durch das gesamte Programm navigieren kann. Im Hauptmenü befinden sich die vier Buttons 'Tutorial', 'Neues Projekt beginnen', 'Letztes Projekt fortsetzen' und 'Galerie'. Der Button Tutorial leitet direkt in ein Fenster (siehe Abbildung C.6), in welchem dem Anwender die Verwendung der Anwendung über Bilder und in Textformat beschrieben wird. Durch das Betätigen des Buttons 'Neues Projekt beginnen' wird ein Fenster geöffnet (siehe Abbildung C.1 rechts), auf dem ein Name für das Projekt und die Art der Fotoaufnahme ausgewählt wird. Bei der Art der Fotoaufnahme wird zwischen Foto und Fotoreihen unterschieden. Der Aufnahmetyp Foto erfordert für jede Fotoaufnahme eine Betätigung des Aufnahmeknopfes. Fotoreihe hingegen wird über den Aufnahmeknopf gestartet und erstellt bei einer Bewegung des Smartphones oder alle 0,5 Sekunden ein Foto. Es wird alle 0,5 Sekunden ein Foto erstellt, da damit die Aufnahme von identischen Fotos vermieden wird. Mit dem Modus Fotoreihe wird das Hochladen großer Fotomengen in kürzester Zeit ermöglicht. Sobald der Aufnahmetyp gewählt wurde und der Name des Projektes eine Zeichenlänge zwischen 3 und 20 besitzt gelangt der Anwender in das Fenster zur Fotoerstellung (siehe Abbildung C.2 links). Das Fenster zur Fotoerstellung setzt sich aus 5 Elementen zusammen.

Auf der oberen linken Ecke befindet sich ein Menü-Symbol, welches das Interaktionsmenü (siehe Abbildung C.2 rechts öffnet). In diesem können die Modi 'Normaler Modus', 'Merkmalerkennung', 'Matcher' und 'Umwandlung' gestartet werden. Die Modi 'Merkmalerkennung' und 'Matcher' sind nur für den Aufnahmetyp 'Foto' verfügbar und werden erst angezeigt, sobald mindestens ein Foto gemacht wurde. Der 'Normale Modus' stellt das aufzunehmende Bild unverändert dar, 'Merkmalerkennung' analysiert ob das Objekt umwandelbar ist (siehe Abbildung C.3 links), 'Matcher' zeigt die zum letzten Foto übereinstimmenden Merkmale (siehe Abbildung C.3 rechts) und 'Umwandlung' leitet in ein Fenster weiter, wo die Umwandlung in ein 3D-Modell gestartet werden kann. Der Button 'Umwandlung' kann erst gestartet werden, wenn alle unter den Interaktionsbutton befindlichen Voraussetzung für eine erfolgreiche Umwandlung erfüllt wurden. Unter dem Menü-Symbol ist ein farbiges Feld, welches anzeigt, ob sich das Smartphone im Hochformat befindet. Die Farbe ist grün, falls sich die Position des Smartphones zwi-

schen  $-45^\circ$  und  $+45^\circ$  Grad befindet, Gelb wird diese bei einer Neigung von  $-90^\circ$  bis  $-45^\circ$  und  $+45^\circ$  bis  $+90^\circ$  und Rot bei unter  $-90^\circ$  oder über  $+90^\circ$  Grad. Wenn das Feld Rot ist, wird der Anwender darauf hingewiesen und erstellte Fotos nicht hochgeladen. Das dritte Element im Fotoaufnahmefenster ist eine Anzeige, welches die Menge aller derzeit akzeptierten Fotos und die Mindestmenge für eine erfolgreiche Rekonstruktion anzeigt. In der Mitte des Fensters ist der Fotoaufnahmeknopf und auf der unteren Seite wird die Punktwolke, mit den Positionen der erstellten Fotos angezeigt. Das untere Fenster lässt sich durch einen Button auf dem unteren Fenster in den Vollbildschirm (siehe Abbildung C.4 links) und wieder in den Normalbildschirmmodus verändern. Wenn der Button 'Umwandlung' betätigt wurde, gelangt der Anwender in das Modellerstellungsfenster (siehe Abbildung C.4 rechts), welches für die Rekonstruktion des 3D-Modell zuständig ist.

Dieses Fenster setzt sich aus einem Informationsfeld auf der oberen Bildschirmseite und 4 Buttons zusammen. Zum einen kann der Nutzer über die Buttons 'Normal' und 'Hoch' auswählen, welche Qualität das hergestellte 3D-Modell besitzen soll. Wird eines dieser Buttons betätigt, so wird der Umwandlungsprozess gestartet und der Nutzer über den Bearbeitungsfortschritt durch das Informationsfeld informiert. Sobald die Umwandlung abgeschlossen wurde, kann das Modell über den Button 'Modell betrachten' betrachtet werden (siehe Abbildung C.5 links). Der Button Hauptmenü leitet den Anwender zurück ins Hauptmenü. Der zweite Button im Hauptmenü 'Letztes Projekt fortsetzen' lässt den Nutzer die zuletzt gestartete Umwandlung fortsetzen. Über den Button Galerie im Hauptmenü öffnet sich ein Fenster (siehe Abbildung C.5 rechts), auf dem alle vom Nutzer erstellten Modelle geöffnet und betrachtet werden können. Die Realisierung der beschriebenen Funktionalität wird in Abschnitt 4.1 beschrieben.

## C Anhang C: Grafische Benutzeroberfläche der Smartphone-Anwendung

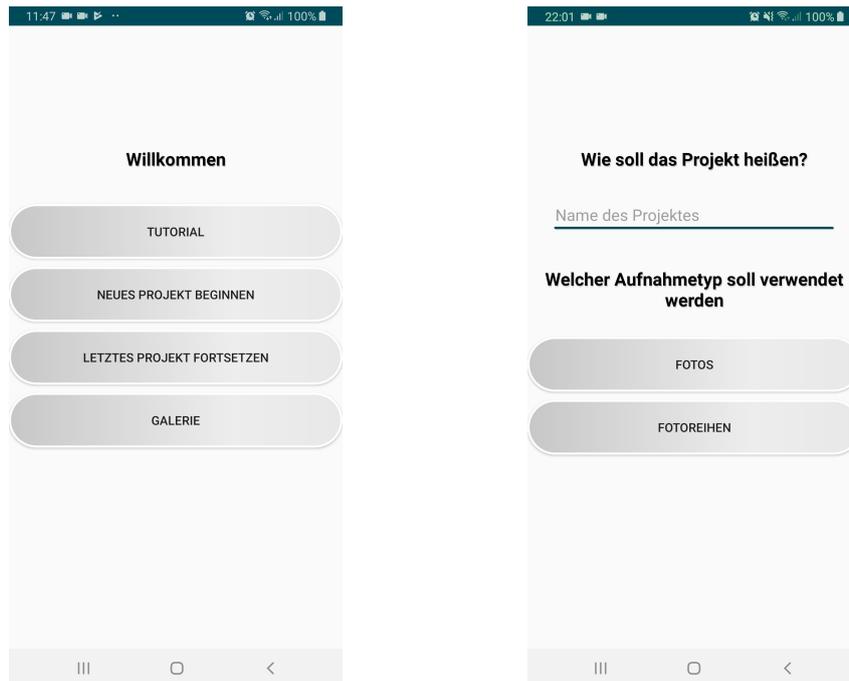


Abbildung C.1: Links: Hauptmenü auf der Smartphone-Anwendung. Rechts: Auswahlbildschirm des Aufnahmetyp auf der Smartphone-Anwendung.

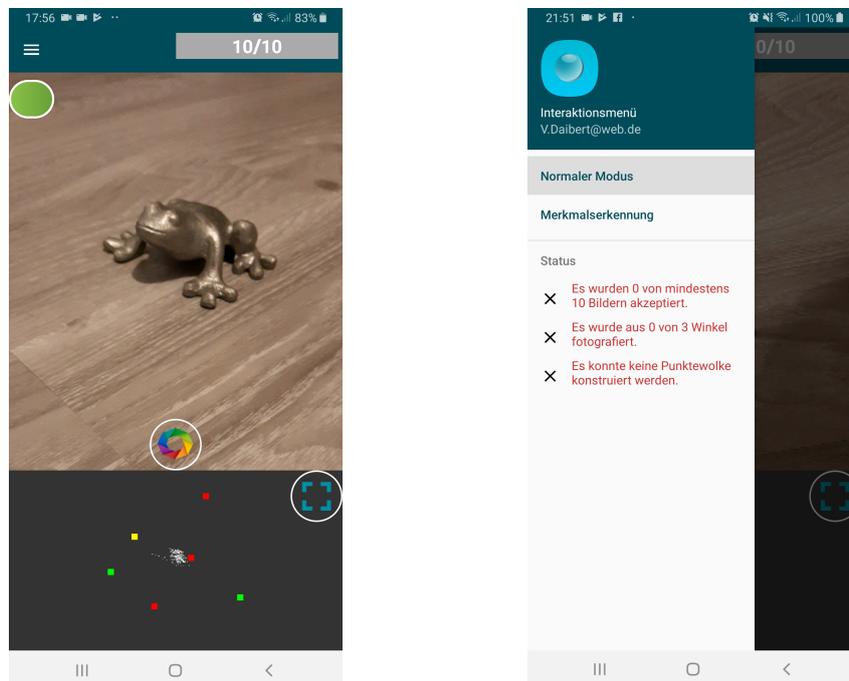


Abbildung C.2: Links: Normaler Aufnahmemodus auf der Smartphone-Anwendung. Rechts: Geöffnete Menü im Aufnahmemodus der Smartphone-Anwendung.

## C Anhang C: Grafische Benutzeroberfläche der Smartphone-Anwendung

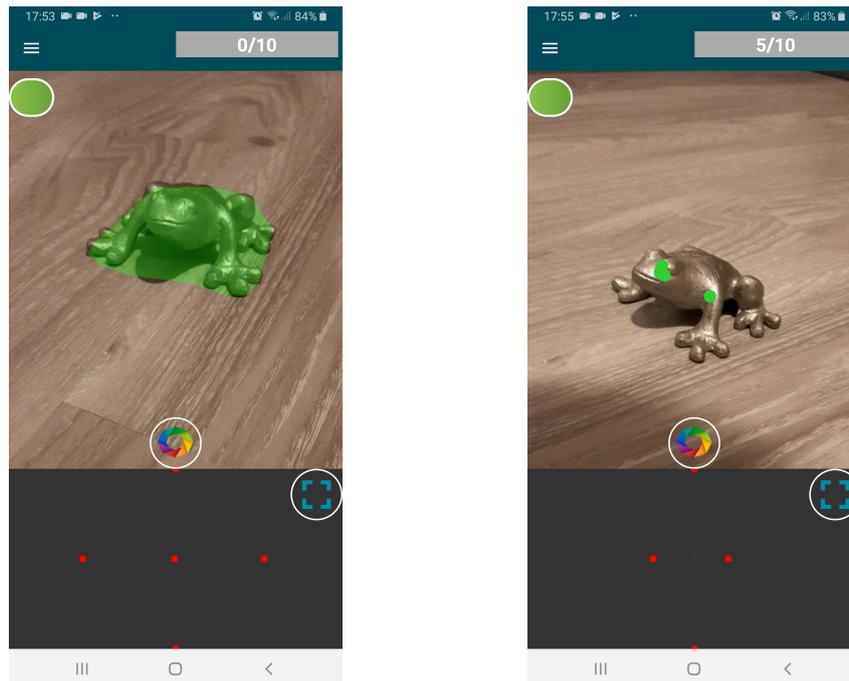


Abbildung C.3: Links: Objekterkennung auf der Smartphone-Anwendung. Rechts: Merkmalsvergleich auf der Smartphone-Anwendung.

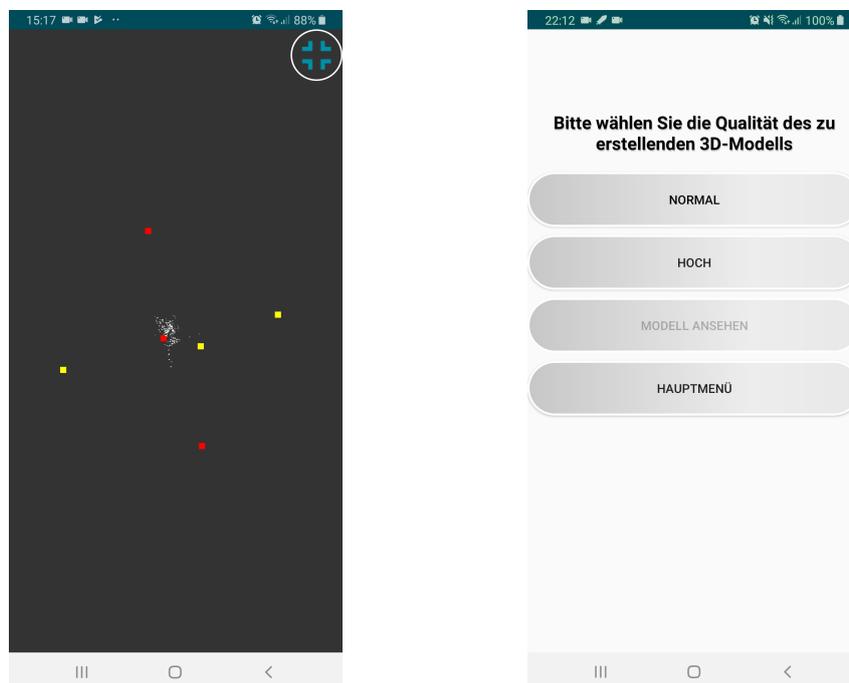


Abbildung C.4: Links: Vollbildmodus zum Betrachten der Punktwolke auf der Smartphone Anwendung. Rechts: Modellerstellungsmenü auf der Smartphone Anwendung.

## C Anhang C: Grafische Benutzeroberfläche der Smartphone-Anwendung

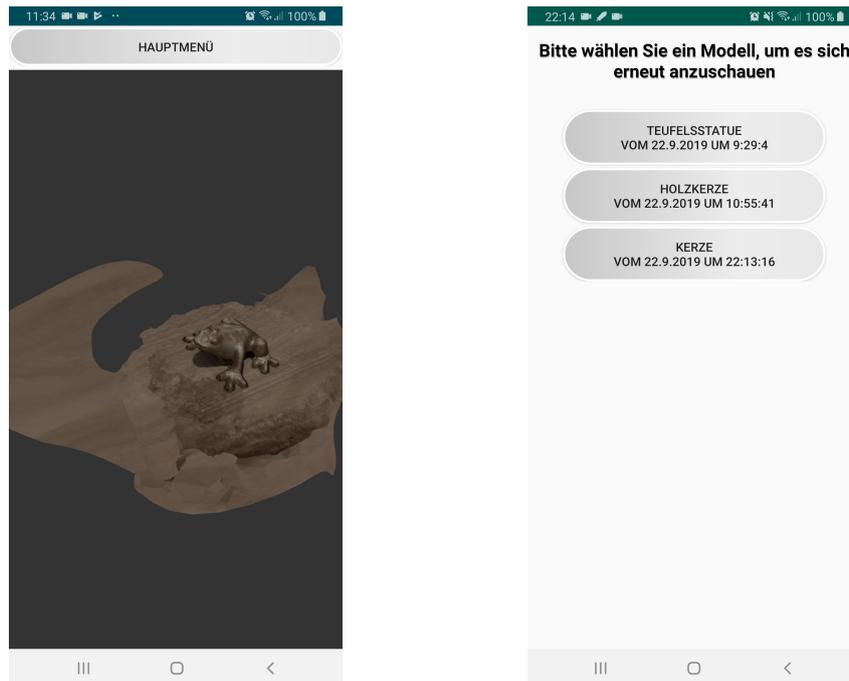


Abbildung C.5: Links: Modellbetrachtung auf der Smartphone Anwendung. Rechts: Galleriemodus auf der Smartphone Anwendung.



Abbildung C.6: Einführungsfenster auf der Smartphone Anwendung.

# D

## Anhang D: Belastungsanalyse

In diesem Abschnitt wird eine Belastungsanalyse sowohl auf dem Smartphone-, als auch auf der Server-Anwendung unternommen. In 'Normaler Zustand' finden keine Berechnungen auf den Geräten statt, 'Erstellen der Fotos' werden alle 0.5 Sekunden ein Foto vom Smartphone auf den Server gesendet. Schnelleres versenden der Fotos sollte nicht möglich sein, da die Fotos vorab auf dem Smartphone selektiert werden müssen. 'Foto-matching' führt den Modus 'Matches' und 'Objektanalyse' den Modus 'Objekterkennung' auf einem Kontrastreichem Foto aus. Bei 'Modellerstellung' wird der Erstellungsprozess eines 3D-Modells mit Meshroom und dem rechenintensivem Prozessen OpenMVG, OpenMVE und SMVS anhand 50 kontrastreicher Bilder untersucht. 'Betrachten des Modells' betrachtet die Belastung während der Betrachtung eines 3D-Modells, welches aus kontrastreichen Fotos erstellt wurde und die maximalste Anzahl an darstellbare Polygonetze enthält.

### Normaler Zustand

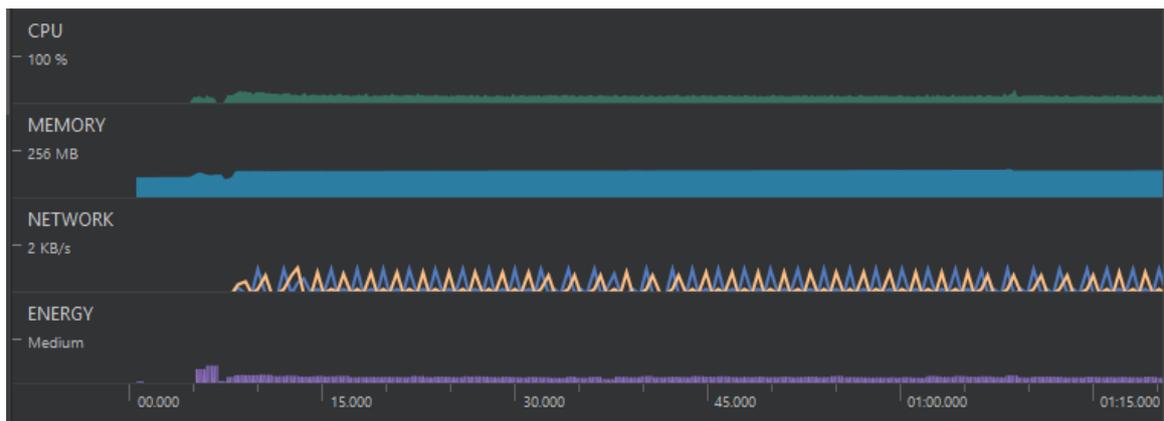


Abbildung D.1: Belastungsanalyse des Smartphones im Normalzustand.

## D Anhang D: Belastungsanalyse

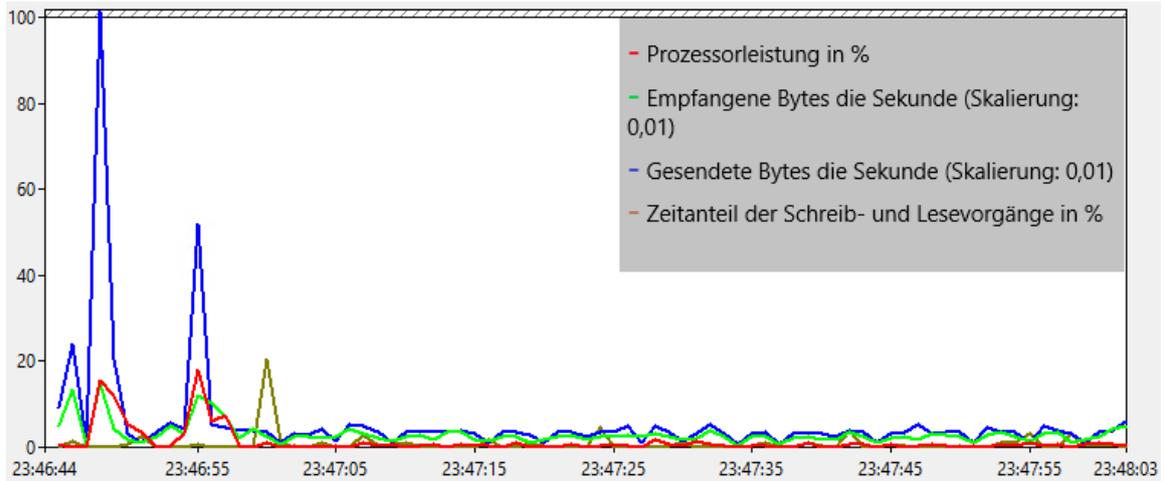


Abbildung D.2: Belastungsanalyse des Servers im Normalzustand.

### Erstellen der Fotos

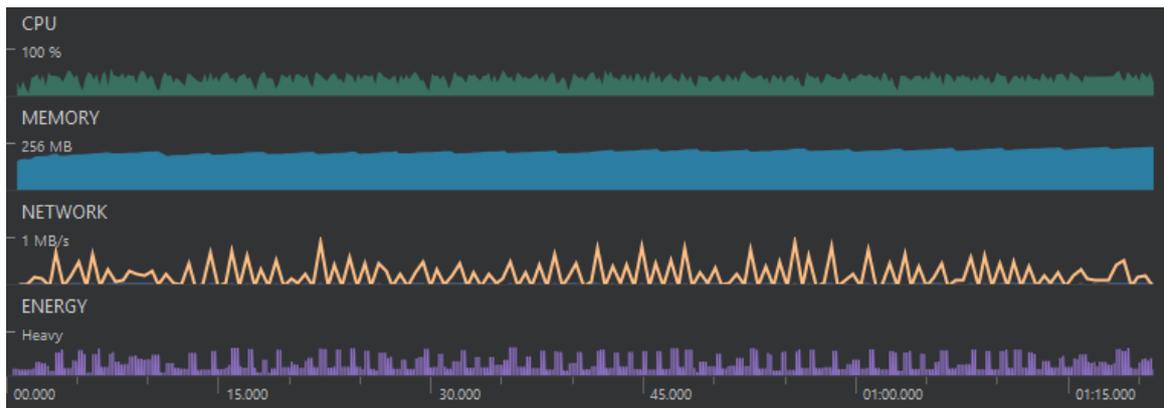


Abbildung D.3: Belastungsanalyse des Smartphones beim Erstellen und hochladen auf den Server aller 500ms eines Fotos.

## D Anhang D: Belastungsanalyse

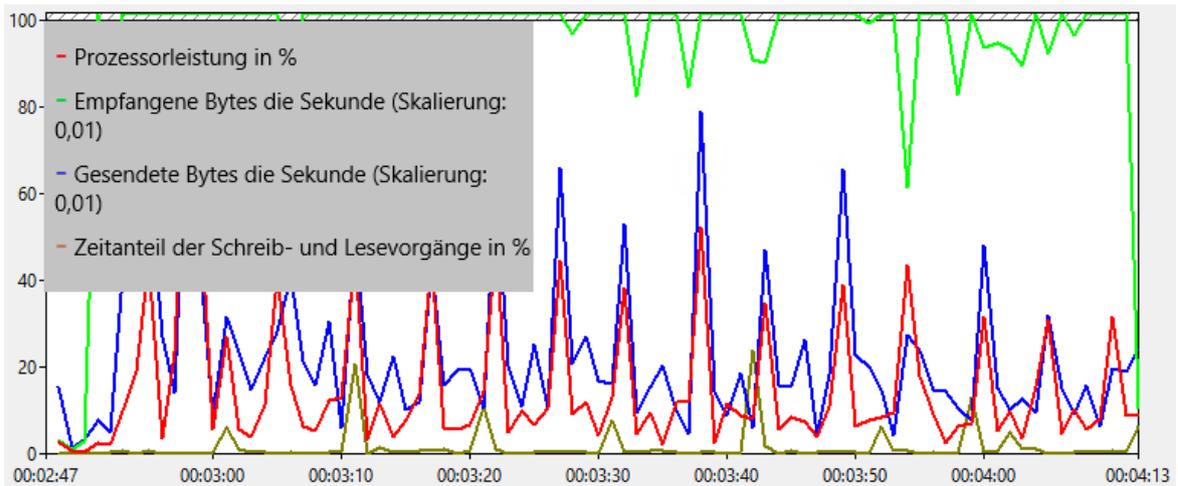


Abbildung D.4: Belastungsanalyse des Servers beim Erstellen einer Punktwolke. Dabei wird alle 500ms ein Foto von Smartphone auf den Server hochgeladen.

## Fotomatching

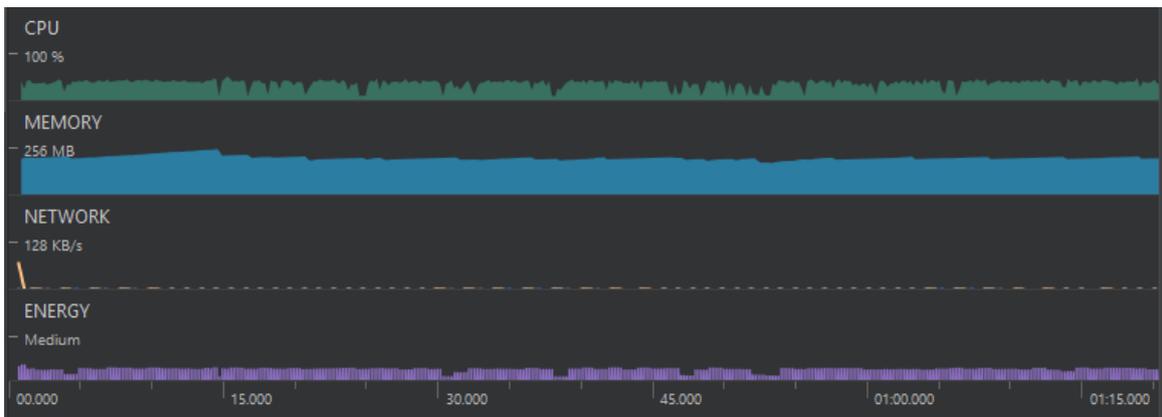


Abbildung D.5: Belastungsanalyse des Smartphones im Matching-Modus.

## Objektanalyse

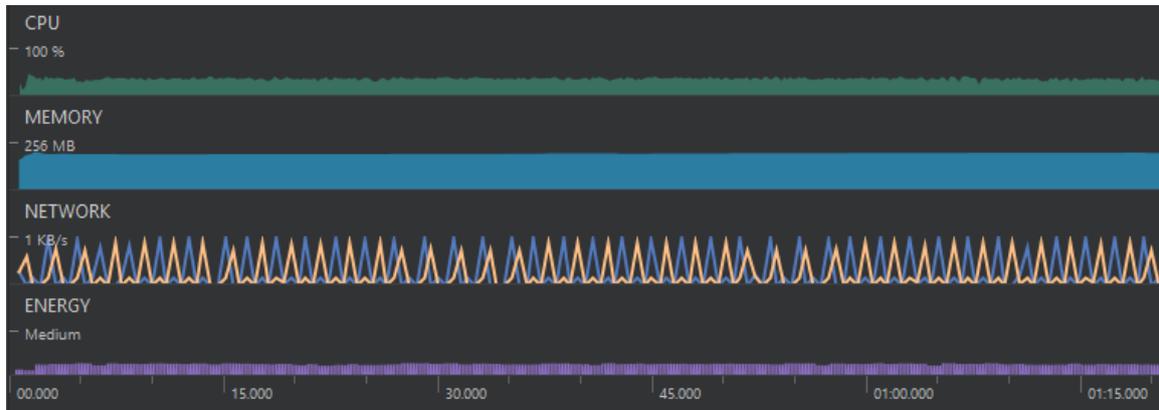


Abbildung D.6: Belastungsanalyse des Smartphones während im Merkmalerkennungs-Modus.

## Modellerstellung

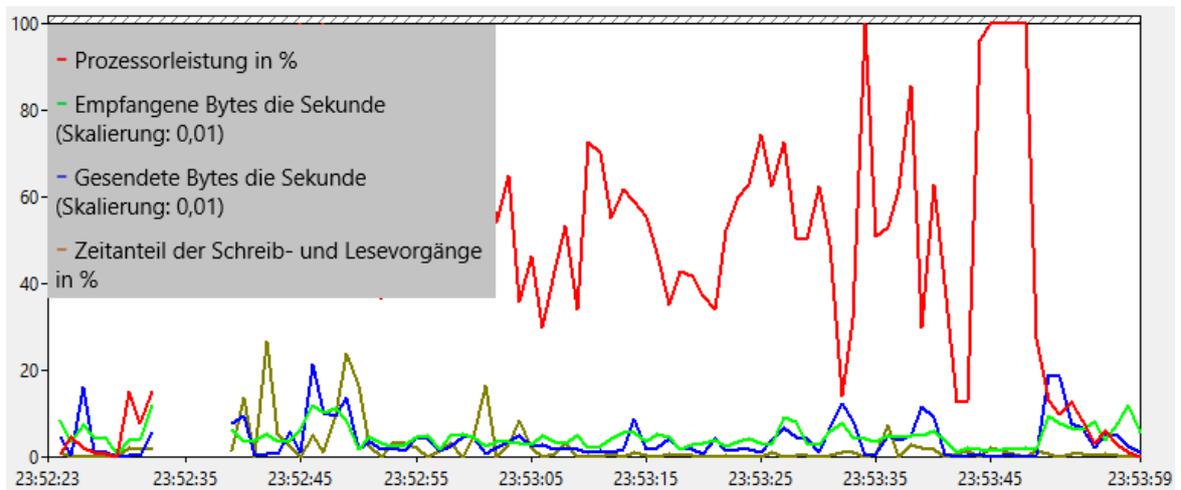


Abbildung D.7: Belastungsanalyse des Server während der 3D-Modellerstellung mit OpenmVG, OpenMVE und SMVS.

## D Anhang D: Belastungsanalyse

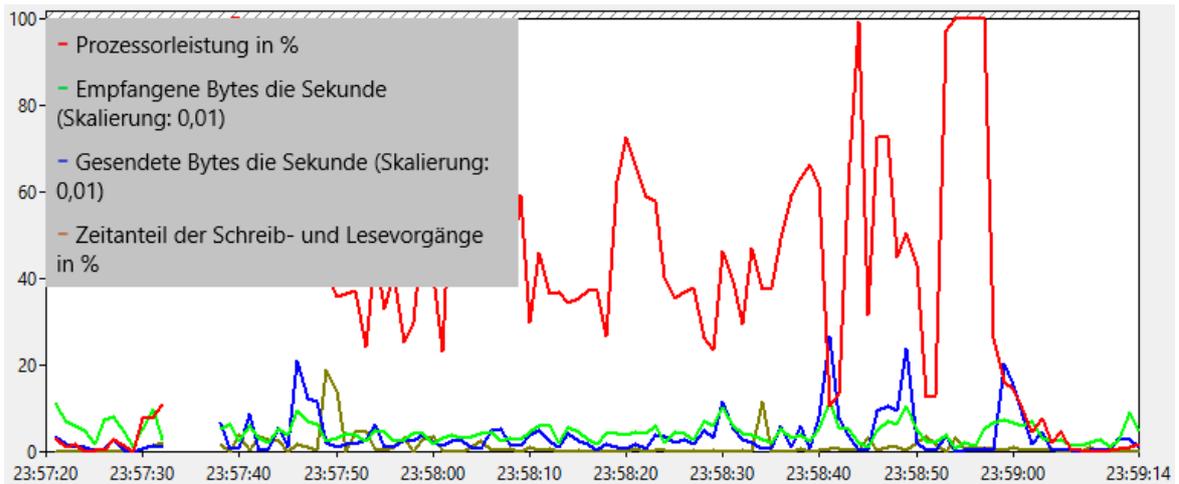


Abbildung D.8: Belastungsanalyse des Server während der Modellerstellung mit Meshroom.

### Betrachten des Modells

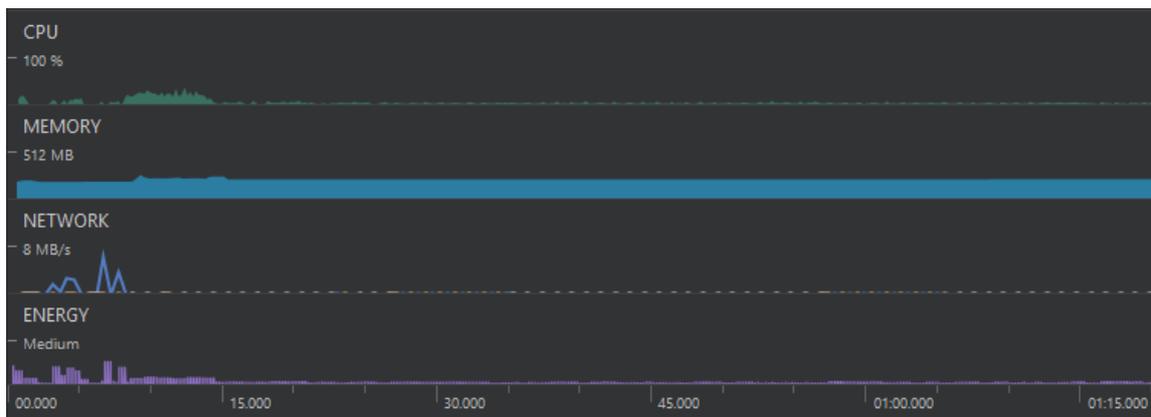


Abbildung D.9: Belastungsanalyse des Smartphones während der Modellerbetrachtung

Die hier dargestellten Ergebnisse sind erwartungskonform und werden im Abschnitt 5.2 diskutiert.