

# Evaluation eines Hochschulprojektes zum Thema 3D-Raumüberwachung

David Alexander Kring  
david\_alexander.kring@smail.th-koeln.de  
Technische Hochschule Köln  
Gummersbach, Germany

Lukas Martin Munz  
lukas\_martin.munz@smail.th-koeln.de  
Technische Hochschule Köln  
Gummersbach, Germany

Moritz Langer  
moritz\_sylvius.langer@smail.th-koeln.de  
Technische Hochschule Köln  
Gummersbach, Germany

Merle Struckmann  
merle.struckmann@smail.th-koeln.de  
Technische Hochschule Köln  
Gummersbach, Germany

## ZUSAMMENFASSUNG

Die folgende Ausarbeitung stellt eine qualitative Evaluation eines Hochschulprojektes zum Thema 3D-Raumüberwachung dar. Dabei werden die Problematiken und deren Ursachen erörtert, um die Fragestellung, wieso das erreichte Projekt der Phase 2 nicht den geplanten Konzept aus Phase 1 entspricht, zu beantworten. Die Evaluation wurde in drei Bereiche unterteilt: *Anforderung & Planung, Implementierung und Projektmanagement*. Abschließend wird mithilfe der Evaluation das erreichte Ergebnis eingeschätzt.

## CCS CONCEPTS

• **Evaluation**; • **Room-Surveillance** → *Three-dimensional*; • **Projectmanagement**;

## KEYWORDS

evaluation, room surveillance, three-dimensional, camera, project-management

### ACM Reference Format:

David Alexander Kring, Moritz Langer, Lukas Martin Munz, and Merle Struckmann. 2022. Evaluation eines Hochschulprojektes zum Thema 3D-Raumüberwachung. In *ACM*, New York, NY, USA, 7 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 EINLEITUNG

Die vorliegende Arbeit stellt einen Teil der dritten und letzten Projektphase des semesterübergreifenden Projekts *3D Room Surveillance* im Medieninformatik Master der technischen Hochschule Köln dar. Genauer dokumentiert diese Arbeit eine Evaluation der vorangegangenen Projektphasen.

Das Projekt beschäftigt sich im Allgemeinen mit der Fragestellung, wie ein festgelegter Raum mithilfe von 3D-Technologie überwacht

werden kann. Zusätzlich ist das Projekt in drei Phasen unterteilt, die jeweils einen anderen Schwerpunkt der Bearbeitung betrachten. In der ersten Phase des Projekts *Vision & Konzept* soll ausgehend von einer Problemstellung eine Recherche der Domäne, sowie des aktuellen Stands der Wissenschaft und Technologie durchgeführt werden, mithilfe dieser dann eine Vision und Konzeption für ein System entwickelt oder weiterentwickelt werden kann. In der zweiten Phase *Entwicklung* soll darauf aufbauend aus dem erarbeiteten Konzept ein „Minimum Viable Product“ entstehen. In der dritten Phase *Forschung, Evaluation, Assessment, Verwertung* soll dann eine Auswertung bzw. Verwertung des Projekts stattfinden.

Die folgende Evaluation betrachtet die erste sowie die zweite Projektphase. Zu Beginn wird in Kapitel 2 die Ausgangssituation der Arbeit erläutert, bevor in Kapitel 3 die ausgewählte Evaluationsmethode genauer betrachtet wird. Im darauffolgenden Kapitel 4 findet dann die eigentliche Evaluation des Projekts statt, die in Kapitel 5 nochmals zusammengefasst und eingeordnet wird.

## 2 AUSGANGSSITUATION

Die Projektidee war ein Raumüberwachungssystem mithilfe von 3D-Technologien zu bauen. Dabei sollte erörtert werden, ob sich 3D-Kameras für den Einsatz als Live-Überwachungssystem eignen und wie ein System in diese Richtung aufgebaut werden könnte. Zudem sollten passende Einsatzgebiete gefunden werden und 3D-Verarbeitungstechniken herausgearbeitet werden, die die Arbeit vereinfachen.

Das hier dargelegte Projekt beschreibt eine 3D-Überwachung eines Supermarkt-Regals. Dabei soll mithilfe der Azure Kinect [1] festgestellt werden, ob eine Veränderung im Kamerabereich, z.B. durch das Entnehmen eines Produktes, aufgetreten ist und dies dem Nutzer mitgeteilt werden. Die Veränderungen und Kameraaufnahmen sollen archiviert werden, sodass der Nutzer auch zu einem späteren Zeitpunkt auf diese zugreifen kann.

Die erste Phase beschäftigte sich mit der Vision und dem Konzept des Projektes. Ausgehend von der Problemstellung wurde eine Recherche der Domäne sowie des aktuellen Stands der Wissenschaft und Technologie durchgeführt. Zudem wurde ein Konzept ausgearbeitet und dokumentiert, so dass in einem nachfolgenden Projekt

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

Conference XX, Technische Hochschule Köln

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

ein Entwicklungsteam in der Lage ist das System zu realisieren. Für die konkrete Problemstellung mussten Hard- und Software für 3D-Scan und Objekt-Extraktion untersucht werden. Zudem sollte ein Konzept für eine Systemarchitektur und ein Benutzer-Interface entworfen werden. Abschließend sollte eine Abschätzung über die Realisierbarkeit und den Nutzen des entworfenen Konzepts abgegeben werden. Das entworfenen Konzept sollte anschließend in der zweiten Phase entwickelt und als "Minimal Viable Product" umgesetzt werden. Entsprechend mussten die gestellten Anforderungen aus der ersten Phase priorisiert und die ausgewählten Hard- und Softwaretools bewertet werden. Zudem musste eine Systemarchitektur, der 3D-Workflow und ein Benutzerinterface entworfen und implementiert werden.

In der dritten Projektphase sollen die Ergebnisse aus den Projektphasen *Vision & Konzept* und *Entwicklung* untersucht und evaluiert werden. Dabei soll das Projekt hinsichtlich verschiedener Fragestellungen betrachtet werden. Zunächst soll geklärt werden, wieso der Prototyp nicht die gestellten Anforderungen aus der ersten Phase erfüllt. Um dies zu erörtern, wird die Evaluation in drei Phasen unterteilt: Anforderung & Planung, Implementierung und Projektmanagement. Dabei liegt der Fokus im Bereich *Anforderung & Planung* bei der Formulierung der gesetzten Anforderungen, der technische Ausarbeitung und Analyse der Hard- und Softwaretools. Im Bereich *Implementierung* soll die Entwicklungsphase hinsichtlich der verwendeten Software, der Arbeitsaufteilung und Integration der Teilprojekte erörtert werden. Abschließend soll im Bereich *Projektmanagement* die Teamdynamik zwischen den zwei Projektphasen untersucht werden, da zum Zeitpunkt der zweiten Projektphase neue Teammitglieder dem Team beigetreten sind. Das Ziel der Evaluation ist die Schwachstellen der vorangegangenen Phasen zu extrahieren, sodass bei einem Redesign des Projektes diese vermieden werden können.

### 3 EVALUIERUNGSMETHODE

Zu Beginn einer Evaluation ist es notwendig die Rahmenbedingungen sowie die zu verwendenden Methoden bei der Durchführung selbiger zu definieren. Aus diesem Grund beschäftigt sich das folgende Kapitel mit der Planung und den Methoden der Evaluation, darüber hinaus wird das Ziel der Evaluation festgelegt. Die Vorgehensweise bei der Projektevaluation stammt dabei aus dem *Leitfaden für die Projektevaluation* des Eidgenössischen Büros für die Gleichstellung von Menschen mit Behinderungen [7].

#### 3.1 Ziel der Evaluation

Um das Ziel der Evaluation zu definieren, ist es zunächst erforderlich den Gegenstand der Evaluation zu erörtern und festzustellen, warum evaluiert werden muss, welche Teile evaluiert werden müssen und welche Aspekte dabei eine genauere Betrachtung benötigen.

Die Gründe für die Durchführung einer Evaluation des Projekts lassen sich in drei Bereiche einteilen. Zum Einen, ist es das subjektive Empfinden des Projektteams, dass im Rahmen der Projektarbeit, besonders in Phase 2, die Arbeit im Team sowie die Arbeitsweise an dem Projekt selbst nicht zufriedenstellend abgelaufen ist. Hier

soll mithilfe der Evaluation klar herausgearbeitet werden, wo und warum dies so passiert ist. Zudem weicht die Durchführung des Projekts in Phase 2 stark von der eigentlichen Planung in Phase 1 ab. Die Evaluation soll aufzeigen, warum es zu diesem Unterschied gekommen ist. Zuletzt ist auch das erarbeitete Ergebnis aus Phase 2 ein Grund für die Evaluation, da die angedachte Konzeption nur in Teilen im Ergebnis wiederzufinden ist. Hier soll beleuchtet werden, warum dies so ist.

Wie aus den oben beschriebenen Gründen bereits abzulesen ist, liegt der Fokus der Evaluation auf dem Teil einer qualitativen Betrachtung des Projekts. Der subjektive Eindruck des Projektteams ist es, dass sich die Koordination im Team, sowie die eigentliche Arbeitsweise negativ auf das Ergebnis ausgewirkt haben. Dementsprechend sollen diese Aspekte genauer untersucht werden, um festzustellen, wie gearbeitet wurde, wie es geplant war zu arbeiten und wie sich mögliche Unterschiede der beiden Bereiche auf das Ergebnis des Projekts ausgewirkt haben.

Aus den festgelegten Gegenständen der Evaluation ergeben sich nun die Fragestellungen der Evaluation. Hierbei muss einerseits festgelegt werden, welche Fragen die Evaluation beantworten muss und ob das Ziel der Evaluation mit der Beantwortung dieser Fragen erreicht wird.

Eine Frage, die beantwortet werden muss, ist, wie das Projekt in Phase 2 im Sinne von Phase 1 umgesetzt wurde. Ob und welche Unterschiede aufgetreten sind sowie begründen, wie es zu diesen gekommen ist. Des Weiteren soll die Frage beantwortet werden, wie das Ergebnis aus Phase 2 eingeschätzt werden kann, entsprechend der betrachteten Bereiche der Evaluation.

Mit der Beantwortung dieser Fragen kann das Ziel der Evaluation, im Gesamtkontext des durchgeführten Redesigns, erreicht werden, da so die Schwierigkeiten des Projekts herausgestellt werden können und einen essentiellen Teil für eine Verbesserung von internen Prozessen beziehungsweise der Projektdurchführung, allgemein darstellen.

#### 3.2 Planung und Methoden

Nachdem das Ziel der Evaluation definiert wurde, folgt nun die Planung, sowie die Festlegung, auf die zu verwendenden Methoden, der Evaluation.

Es handelt sich um eine interne Evaluation, da sie durch Teile des Projektteams selbst durchgeführt wird. Dies hat den Vorteil, dass bereits vertiefte Kenntnisse des Projekts vorliegen.

Weiter wird als Methode der Datenerhebung eine qualitative Betrachtung durchgeführt, da die erhobenen Daten entsprechender Natur sind. Es handelt sich um individuelle Daten, die nicht in Zahlen messbar sind. Diese werden aus direkten Beobachtungen und Erfahrungen des Projekts gewonnen. Quantitative Daten, die möglicherweise das Ergebnis des Projekts beschreiben, werden nicht für die Evaluation hinzugezogen, da der Fokus, wie in der Sektion Ziel der Evaluation beschrieben, auf dem Prozess der Projektarbeit liegt. Dies spiegelt auch die Einordnung der Evaluation in den Kontext des Redesigns wieder.

Die erhobenen Daten werden anschließend in der Analyse ausgewertet, so dass Schlussfolgerungen gezogen werden können. Mithilfe dieser Schlussfolgerungen kann wiederum das Ziel der Evaluation erreicht und die damit verbundenen Fragen beantwortet werden.

## 4 EVALUATION

Im Folgendem wird eine Evaluation zu den Bereichen *Anforderung & Planung*, *Implementierung* und *Projektmanagement* durchgeführt. Dabei soll die Problematik bei der Bearbeitung des Projektes erörtert und analysiert werden. Für den Abschnitt *Anforderung & Planung* werden die gestellten Anforderungen in Projektphase 1 mit den umgesetzten Aspekten in Projektphase 2 abgeglichen. Für den Abschnitt *Implementierung* wird die Arbeitsaufteilung, die Softwareentwicklung und die Integration der Teilgruppen erörtert. Abschließend wird im Abschnitt *Projektmanagement* die Projektbearbeitung hinsichtlich der Teamdynamik, der Gruppenaufteilung und der Kommunikation analysiert.

### 4.1 Anforderung & Planung

Die im Projekt notwendigen Anforderungen des zu entwickelnden Systems wurden in Phase 1 der Projektarbeit definiert und sollten anschließend in Phase 2 umgesetzt werden. Folgend soll aus diesem Grund evaluiert werden, inwieweit die Anforderungen aus Phase 1 in Phase 2 umgesetzt wurden. Zusätzlich wurde, in der ersten Phase des Projekts, aus den Anforderungen eine Planung für Hardware- / Softwarekomponenten erstellt, welche im Hintergrund der Durchführung der zweiten Phase ebenfalls in der Evaluation berücksichtigt wird.

**4.1.1 Anforderungen.** Wie bereits erwähnt, wurden die Anforderungen des Systems in der ersten Phase des Projekts aufgestellt und formuliert. Insgesamt wurden 23 funktionale und nicht-funktionale Anforderungen an das System gestellt. Von diesen 23 definierten Anforderungen aus Phase 1 wurden acht in Phase 2 umgesetzt.

Umgesetzte Anforderungen:

- Das System muss dem Benutzer die Möglichkeit geben ausgewählte Objekte zu tracken.
- Das System muss dem Benutzer auf neue Objekte aufmerksam machen.
- Das System soll die Kameraaufzeichnung für einen Zeitraum speichern.
- Das System kann dem Benutzer eine Tiefenansicht vom Kamerabild zur Verfügung stellen.
- Das System muss die Objekterkennung und das Object-tracking in Echtzeit durchführen.
- Das System soll eine konsistente Benutzeroberfläche aufweisen.
- Die Berechnungsquote für fehlerhafte Ergebnisse muss im festgelegten Akzeptanzbereich liegen.
- Das System soll die Videoaufzeichnungen löschen, wenn der Zweck der Speicherung erfüllt wurde.

Die umgesetzten Anforderungen definieren mehrheitlich das Erkennen von Veränderungen im Kamerabereich, entsprechend der

Hauptaufgabe des Systems. Die Liste der nicht umgesetzten Anforderungen besteht beispielsweise aus folgenden Anforderungen:

- Das System kann die Möglichkeit bieten neue Berechtigungsgruppen zu erstellen.
- Das System soll Personen im Kamerabereich unkenntlich machen.
- Das System muss dem Benutzer die Möglichkeit geben sich zu authentifizieren

Die nicht umgesetzten Anforderungen umfassen mehrheitlich Anforderungen, die Bezug auf die Idee der ersten Projektphase im Sinne von einer Anwendung für einen Supermarkt, mit unterschiedlichen Berechtigungsgruppen, nehmen.

Der Vergleich der beiden Listen zeigt, dass in Phase 2 des Projekts der Fokus auf die Anforderungen, welche die Hauptaufgabe des Systems beschreiben, gelegt wurde. Diese konnten wie in der Konzeption beschrieben auch umgesetzt werden. Anforderungen zu den Bereichen Anwendungskontext (Supermarkt), Interface, Mensch-Computer-Interaktion und Datenschutz wurden nicht umgesetzt. Das Fehlen dieser Anforderungen hat unterschiedliche Ursachen, wie beispielsweise mangelhaftes Projektmanagement und Kommunikation (siehe Kapitel 4.3). Der größte Einflussfaktor war dennoch die Priorisierung des Projektteams auf Anforderungen, welche die Hauptaufgabe des Systems definieren.

**4.1.2 Planung.** Wie bereits einleitend erwähnt, wurde in der ersten Phase des Projekts aus den Anforderungen eine Planung für Hardware- / Softwarekomponenten erstellt. Dabei wurden jeweils unterschiedliche Hardware- / Softwarekomponenten miteinander verglichen und daraufhin entschieden. Die Auswahl erfolgte mithilfe von Recherchen und dem Vergleichen von Datenblättern (bei Hardwarekomponenten). Es wurden keine Tests mit den jeweiligen Komponenten durchgeführt, was im Verlauf von Phase 2 zu weiteren Komplikationen geführt hat (siehe Kapitel 4.2).

### 4.2 Implementierung

Die Implementierung des Projektes fand in der zweiten Phase, der Entwicklungsphase, statt. Hier sollte das erstellte Konzept in einen Prototypen umgesetzt werden. Für die Evaluierung werden die Bereiche Arbeitsaufteilung, Softwareentwicklung und Integration des Projektes betrachtet. Dabei soll erörtert werden, welche Aspekte der einzelnen Bereiche zu Problemen geführt haben und wie diese hätte vermieden werden können.

**4.2.1 Arbeitsaufteilung.** Zu Beginn der Entwicklungsphase wurde die sechser Gruppe in drei Teilgruppen à zwei Personen aufgeteilt. Dabei bearbeitete jede Teilgruppe einen anderen Aspekt des Projektes. Anfänglich hat sich eine Teilgruppe um die Verarbeitung der Kamera gekümmert. Währenddessen hat eine zweite Teilgruppe das Surface Matching implementiert und die dritte Teilgruppe Möglichkeiten zur Verbesserung der Punktwolke recherchiert.

Die Aufteilung in Teilgruppen wurde zu Beginn sowie im Nachhinein als sinnvoll betrachtet. Mit sechs Gruppenmitgliedern an denselben Aufgaben wäre es aus Zeitmanagement-Sicht problematisch geworden alle Aufgaben zu erfüllen. Aus den Erfahrungen bei der Integration der verschiedenen Teilprojekte kann gesagt

werden, dass innerhalb dieser größeren Gruppenphase sich nicht alle Gruppenmitglieder einbringen konnten. Das hat dazu geführt, dass nur 2-3 Personen aktiv gearbeitet haben und mehr Zeit für die Aufgaben gebraucht wurde.

Die Aufgabenverteilung kann an sich positiv gewertet werden. Es wurden klare Grenzen zwischen den verschiedenen Teilaufgaben gezogen, sodass die einzelnen Teilgruppen unabhängig voneinander zu Ergebnissen kommen konnten. Allerdings wurden die Stärken der einzelnen Gruppenmitglieder bei der Verteilung der Aufgaben nicht optimal genutzt, da diese durch die kurze Zusammenarbeit nicht bekannt waren. Zudem war die komplette Abspaltung von den Teilprojekten zu alleinstehenden Codebereichen ein großes Problem. Dies führte zu einer erschwerten Zusammenführung der einzelnen Projektteile zu einem vollständigen Projekt. Dadurch konnten Fehler wie zum Beispiel die falschen Positive beim Surface Matching [5], also Situationen, in denen das System eine falsche Veränderung erkannt hat, länger unentdeckt bleiben.

Was die Arbeit an dem Projekt ebenfalls erschwert hat, war die Begrenzung auf nur eine 3D-Kamera, der Azure Kinect. Dieser Fehler ist auf ein Versäumnis der Gruppe zu Beginn des Semesters zurückzuführen. Eine zusätzliche Kamera hätte es erlaubt, dass andere Mitglieder der Gruppe mithilfe der Kamera, Tests durchführen können und die Abhängigkeit von einem anderen Teammitglied wäre stark reduziert worden. Die Möglichkeit des Austauschs der Hardware innerhalb der Teilgruppen war aufgrund der aktuellen Covid Situation nicht möglich.

Abschließend lässt sich zu der Aufgabenverteilung sagen, dass vor allem eine stärkere Kommunikation und mehr Austausch zwischen den Teilgruppen nötig gewesen wäre. Dadurch hätten alle Teammitglieder die Gelegenheit gehabt, die Ergebnisse der anderen Teilgruppen besser zu verstehen. Es ist anzunehmen, dass so, zum einen Fehler schneller entdeckt worden wären und zum anderen der Prozess der Zusammenführung der verschiedenen Teilergebnisse schneller und effizienter abgelaufen wäre.

**4.2.2 Softwareentwicklung.** Im Bereich der Softwareentwicklung sind verschiedene Probleme aufgetreten, die im Folgendem evaluiert werden.

**PointCloudLibrary.** Bereits zu Beginn der Entwicklungsphase gab es Probleme mit der Point Cloud Library (PCL) [8]. Der Teil der Gruppe, welcher auf MacOS arbeitete, war nicht in der Lage die Bibliothek zu installieren. Der Rest der Gruppe hatte ebenfalls Probleme, da Teile der Library nicht korrekt heruntergeladen wurden. Dies wurde durch das manuelle Hinzufügen behoben. Mithilfe der PCL war es möglich Surface Matching auszuführen, jedoch gab es, wie erwartet, einige *falsche Positive*. Ein erweitertes Surface Matching wird angeboten, um diese fehlerhaften Ergebnisse zu eliminieren, jedoch führte dies zu weiteren Fehlern. Nach längerer Untersuchung wurde festgestellt, dass einige Teile von PCL nicht richtig funktionierten. Es ist anzunehmen, dass dies mit der manuellen Installation zusammenhängt. Diese Phase dauerte bereits mehr als zwei Monate und Teile der Gruppe konnten PCL überhaupt nicht benutzen. Aus diesen Gründen wurde sich entschieden einen neuen Ansatz zu wählen.

Im Rückblick hätte man hier eine erneute Analyse durchführen sollen, um herauszufinden, ob Surface Matching der richtige Ansatz

für die Lösung des vorliegenden Problems ist. Stattdessen wurde sich entschieden, die gleiche Kernidee mit einer anderen Bibliothek umzusetzen.

**OpenCV.** Für die weitere Bearbeitung des Projekts wurde OpenCV [6] benutzt, diese Library war den meisten Gruppenmitgliedern unbekannt, kostete also viel Einarbeitungszeit. Die Installation lief dafür deutlich fehlerfreier, abgesehen von einigen Problemen mit CMake. Diese Probleme konnten aber schnell von dem Teil der Gruppe behoben werden, die sich in diesem Bereich bereits auskannten.

Mit dieser Bibliothek konnte das Surface Marching ebenfalls erfolgreich umgesetzt werden. Die einzelnen Teile, wie beispielsweise das Trainieren des Modells, konnten aufgeteilt werden, um im späteren Verlauf des Programmcodes einiges an Zeit zu sparen. Anschließend wurde durch Multithreading ein Gerüst gebaut, welches erlaubt, mehrere Szenen und Objekte gleichzeitig abzugleichen. Während der Entwicklung dieses Programnteils wurde jedoch der Fehler gemacht, dass nur mit den von OpenCV bereitgestellten Beispielen gearbeitet wurde und nicht mit den selbst aufgenommenen. Dies war notwendig, da, wie bereits erwähnt, nur eine Kamera vorhanden war und die anderen Gruppenmitglieder sonst nicht unabhängig weiterarbeiten konnten. Bei dem Testen des Programmes mit dem eigenen Point Clouds wurde festgestellt, dass es zu vielen falschen Ergebnissen kam.

Die Kamera wurde mittels OpenCV live dargestellt, zusätzlich wurde mit der Library ermöglicht, Aufnahmen und Point Clouds<sup>1</sup> zu speichern. Das Problem war jedoch, dass die erstellten Point Clouds keine Oberflächeninformationen mitspeichern. Diese Oberflächeninformationen werden in herkömmlichen 3D-Anwendungen als sogenannte *Faces* mitgespeichert. Der Surface Matching Ansatz konnte jedoch keine Point Clouds ohne die Oberfläche verarbeiten. Aus diesem Grund funktionierte das Programm mit den eigenen Daten nicht. Die Aufteilung der Arbeit und fehlende Kommunikation verhinderte, dass dieses Problem früher gefunden wurde. Die Lösung des Problems war es die Oberfläche nachträglich zu errechnen. Dies erwies sich aber als sehr kompliziert.

**Open3D.** Die Umsetzung der Erstellung der Faces wurde mithilfe von Open3D[4] erstellt. Genutzt wurde der Poisson Algorithmus[3], welcher sich im Vergleich mit dem Ball Pivoting[3] durchsetzte. Die mit dieser Methode verbesserten Point Clouds führten jedoch immer noch zu falschen Positiven, in der Ausführung des Surface Matching. Nach einiger Recherche stellte sich heraus, dass die Auflösung der Point Clouds nach den Speichern und Bearbeiten zu gering war, um ein sinnvolles Ergebnis zu bekommen. Folglich wurde eine Recherche in andere Möglichkeiten der Umsetzung durchgeführt. Dies hätte, wie bereits erwähnt, deutlich früher im Projektverlauf geschehen müssen. Teilerfolge haben aber davon abgehalten, da durch diese der Anschein entstand, dass der gewollte Lösungsweg funktionieren könnte.

In der Recherche wurde die Methode des *Point Cloud Matching* entdeckt und anschließend umgesetzt. Diese Umsetzung lieferte richtige Ergebnisse und sehr wenige falsche Positive (abhängig vom selbst gesetzten Threshold). Das dadurch entstandene Ergebnis war jedoch nicht mehr das Ergebnis, welches aus den Anforderungen

<sup>1</sup>Dreidimensionale Punktwolken die von der Kamera aufgenommen werden.

aus der ersten Projektphase erwartet wurde, da sich der Fokus im Verlauf der Projektphase 2 verändert hat.

**Windows Presentation Foundation.** Mithilfe des Frameworks Windows Presentation Foundation (kurz WPF) [2] sollte das Interface erstellt werden. Da es zu Anfang Probleme mit der Benutzung und Installation gab, hätte eine neue Recherche nach Alternativen betrieben werden müssen. Dies ist allerdings nicht geschehen. Die Integration des Interface und Backends hätte vorab oder im Verlauf der Arbeit besser geplant werden müssen, da dies zu vielen Problemen führte. Dadurch konnte das Interface im Endprodukt nicht mit eingebunden werden.

**4.2.3 Integration.** Die Zusammenführung der einzelnen Projektteile zu einem vollständigen Projekt wurde am Ende der Implementierung durchgeführt. Wie in den vorherigen Kapiteln bereits erwähnt, führte die späte Zusammenführung zu einigen Problemen, was dazu führte, dass das Endprodukt nicht den anfangs definierten Projektzielen entsprach. Zunächst wurden sich vorab keine Gedanken um die Zusammenführung der Teilaufgaben gemacht, wodurch die Integrationsphase unstrukturiert ablief. Fehlende Kommentare bzw. die schlechte Dokumentation des Codes erschwerten zudem das Verständnis der importierten Code-Teile. Ein weiterer Aspekt, welcher schon im Kapitel 4.2.1 angesprochen wurde, war die Schwierigkeit von jedem Teammitglied Input zu erhalten, bedingt durch die Größe der Gruppe. Jedoch konnte aufgrund der Beschränkung auf eine Kamera keine Aufteilung in Teilgruppen durchgeführt werden. Dadurch, dass nur ein Programmteil auf die Kamera zugreifen konnte, stoppte die Übertragung der Kamera bei der Erstellung einer Point Cloud. Durch die Verwendung von Threads konnte das Problem gelöst werden. Jedoch führte dieser Lösungsansatz zu Problemen, indem die Videoaufzeichnung sich überschchnitt. Die Überschneidungen lagen daran, dass die Azure Kinect keine konstante Framerate bei Full-HD aufrechterhalten kann. Jedoch arbeitet die Funktion zu dem Abspeichern der Videoaufnahmen mit genau dieser Framerate. Während der Implementierungsphase wurde das Feststellen von Veränderungen mithilfe von verschiedenen Bibliotheken ausprobiert, letztendlich wurde Open3D verwendet. Die Kamera wurde mithilfe der Bibliothek OpenCV implementiert. Die beiden unterschiedlichen Bibliotheken konnten jedoch nicht den gleichen Compiler benutzen. Dies musste durch die Erstellung einer Executable des Point Cloud Matching gelöst werden. Das Problem entstand ebenfalls durch die Zerstückelung der Aufgaben und die späte Zusammenführung aller Teilprojekte. Zuletzt muss negativ bewertet werden, dass in der Integrationsphase effektiv immer nur 2-4 Personen arbeiten konnten. Andere Gruppenmitglieder, die sich zu den Zeitpunkten schlecht integrieren konnten, hätten sich bereits um andere Teile wie zum Beispiel Ideen zur Integration des Interfaces beschäftigen können. Alle diese Probleme führten dazu, dass die Integration des Projektes nicht so gelang wie gewollt. Das Endergebnis entsprach nicht den Erwartungen und Zielen des Teams.

## 4.3 Projektmanagement

In der ersten Projektphase *Vision & Konzept* wurde die Thematik der 3D-Raumüberwachung in zwei Schwerpunkte à drei Personen aufgeteilt. Jedes der beiden Teams entwickelte ein eigenes Konzept

für den gewählten Anwendungsbereich. Zudem wurden innerhalb der Gruppen Rollen verteilt, um die verschiedenen Zuständigkeiten festzulegen. Für das Zeitmanagement wurde das Gantt-Chart Model von Agantty verwendet. Hier wurden Meilensteine und Aufgaben eingefügt und die zuständige Person eingetragen. Zusätzlich wurde der Zeitplan regelmäßig überarbeitet, um diesen an den aktuellen Stand anzupassen. Tools wie zum Beispiel Agantty, GitHub Issues, die als Unterstützung für die kommende Entwicklungsphase angedacht waren, wurden gemeinsam als Team ausgesucht und vorbereitet. Durch die Gruppengröße von drei Personen wurde in der Projektphase *Vision & Konzept* die Aufgabenteile vordergründig als Gruppe bearbeitet. Dies konnte durchgeführt werden, da es in der Projektphase nur um die Konzeption des Projektes ging. So war der Austausch von Ideen und Lösungen ein wichtiger Arbeitsbestandteil, um auf das entsprechende Endergebnis zu kommen. Zum Teil gab es zudem kleinere Teilaufgaben, die nach den regelmäßigen Wochenmeetings wieder zusammengeführt wurden, wie zum Beispiel die Recherche von verschiedenen Bereichen. Außerdem wurde einige Aufgaben separat bearbeitet, um verschiedene Lösungsansätze zu bekommen wie zum Beispiel beim Design des Interface. Neben den wöchentlichen Meetings innerhalb des Teams wurden in einem 14-Tage-Rhythmus Meetings mit dem Projekt Owner abgehalten. Hier wurde der aktuelle Projektstand mit den Betreuern, sowie einem parallel arbeiteten Teams ausgetauscht und neue Ideen und Anregung mitgeteilt. Das zweite Konzept wurde jedoch nicht in der Projektphase *Entwicklung* weitergeführt, wodurch der Austausch zwischen den Teams nicht mehr stattfinden konnte. Da das Projektmanagement in der ersten Projektphase als gut eingestuft werden kann, soll in den nächsten Unterkapiteln evaluiert werden, welche Problematiken in der zweiten Projektphase aufgetreten sind und welche Ursachen diese haben.

**4.3.1 Teamgröße.** Zu Beginn der zweiten Projektphase *Entwicklung* hat sich die Gruppengröße von drei auf sechs Teammitglieder verändert. Für die Einarbeitung standen den neuen Teammitgliedern die Dokumentation aus der Projektphase *Vision & Konzept* zur Verfügung, zudem konnten Fragen an die bisherigen Gruppenmitglieder gestellt werden. Jedoch hatten die neuen Mitglieder nicht ausreichend Zeit, um sich einen klaren Überblick über das Projekt anzueignen. Um die Einarbeitung der neuen Teammitglieder in den Entwicklungszyklus zu integrieren, hätte eine neue Projektstruktur, wie zum Beispiel Scrum, angefertigt werden müssen. Zusätzlich wurde die Rollenverteilung aus der vorherigen Phase nicht angepasst und bestehende Rollen wurden vernachlässigt. Ebenfalls wurden verschiedene Tools wie die Agantty oder die GitHub Issues, die den Entwicklungszyklus vereinfachen sollten, nicht verfolgt und gepflegt.

**4.3.2 Gruppenaufteilung.** In der Projektphase *Entwicklung* wurde sich aufgrund der Gruppengröße für eine Aufgabenteilung in Teilgruppen von zwei Personen entschieden. Dies ermöglicht die Bearbeitung von Aufgaben parallel und unabhängig zueinander, jedoch entstand durch die fehlende Durchmischung der Teilgruppen eine Abkapselung von einer der drei Teilgruppen. Die Gruppenmitglieder außerhalb der Teilgruppen kannten sich nicht was weiterhin zu Kommunikationsproblemen führte. Durch diese Verständigungsprobleme spaltete sich eine Zweierpaarung ab und lieferte keine

konkreten Arbeitsergebnisse. Aufgrund des großen Arbeitsaufwandes konnte sich nicht vollständig mit der Arbeit der anderen Teilgruppen beschäftigt werden. Zwar wurde versucht Lösungsansätze darzubieten, jedoch konnte sich nicht intensiv mit der Problemlösung der andere Teilgruppen beschäftigt werden. Dies führte dazu, dass eine Teilgruppe aufgrund der Probleme keine Ergebnisse liefern konnte. Ein Ansatz, um dies zu vermeiden wären die Teilgruppen durchzumischen oder aufzulösen. Ein anderer Ansatz, der auch die Einarbeitungsphase der neuen Mitglieder unterstützt hätte, wäre die Teilgruppen in Abhängigkeit von den Erfahrungen mit dem Projekt zu bilden. Dies wurde jedoch aufgrund von persönlichen Präferenzen nicht durchgeführt. Zusätzlich konnte der Scope der Arbeit zu Beginn der Entwicklungsphase nicht eingeschätzt werden, was zu zeitlichem Druck gegen Ende des Projektes führte.

**4.3.3 Kommunikation.** Für die Planung wurde das Gantt-Chart Tool Agantty aus der Projektphase *Vision & Konzept* weitergeführt. Hier fehlte jedoch die regelmäßige Überarbeitung. Dadurch entsprachen die dort verzeichneten Arbeitspakete oft nicht dem eigentlichen Projektablauf. Dies hatte zur Folge, dass das Tool sich nicht in den Arbeitsablauf der Gruppe integrieren lies und der Aufwand die Arbeitspakete zu überarbeiten immer größer wurde, ohne einen sichtlichen Mehrwert zu bieten. Parallel zu Agantty wurden für den begleitenden Design Sprint GitHub Issues angelegt. Mithilfe dieser Issues wurde die Implementationsphase geplant und Arbeitspakete in verschiedenen Kategorien erstellt und aufgeteilt. Dies stellte in Teilen einen doppelten Aufwand dar, da hierdurch Aufgabenpakete in verschiedenen Tools synchron gehalten werden mussten und die hierfür benötigte Zeit keinen Nutzen für das eigentliche Projekt ergeben hat. Durch das zusätzlich im GitHub erstellte Wiki, in dem diverse Informationen und zusätzliche Deliverables aus dem Design Sprint abgelegt wurden, wurde Agantty letztlich nicht weiter genutzt. Nach Erstellung dieser GitHub-Ressourcen wurde es zusätzlich versäumt den neuen Gruppenmitgliedern die Navigation durch diese Ressourcen zu erleichtern, wodurch die Ressourcen wenig genutzt und nicht angemessen gepflegt wurden. So konnten andere Gruppenmitglieder den Fortschritt einzelner Teammitglieder nicht einschätzen, was gegenseitige Hilfestellungen erschwerte.

Aufgrund dieser Probleme hätte eine Neustrukturierung des Projektes stattfinden müssen, in der sich alle Gruppenmitglieder auf Tools und Ressourcenstrukturen einigen, um die Planung nachvollziehen zu können. Hier wäre ein Projektleiter notwendig, der diesen Prozess überwacht und zuständig ist, die Probleme der Projektplanung zu bearbeiten. Aufgrund der Tatsache, dass sich die Gruppenmitglieder vor Projektbeginn nicht kannten, konnte kein geeigneter Projektleiter gewählt werden. Letztendlich führte eine Kombination aus diesen Faktoren zu einer Abkapslung der einzelnen Teilgruppen, wo lediglich in den wöchentlichen Meetings der Projektstand der jeweiligen Teilbereiche kommuniziert wurde.

Für die Kommunikation im Team wurden wöchentliche Meetings gehalten, in denen sich das Team über den derzeitigen Fortschritt und Aufgabenverteilung abgesprochen hat. Zusätzlich dazu wurde der Fortschritt in einem Zwei-Wochen-Rhythmus dem Project Owner vorgestellt und so ein Feedback-Loop sichergestellt. Die wöchentlichen Meetings können jedoch als vorteilhaft angesehen

werden, da hierdurch die Kommunikation zwischen den einzelnen Teilgruppen bestehen blieb. Zudem bietet der Wochenabstand genügend Zeit, um an dem Teilbereich zu arbeiten und bei den Meetings einen Fortschritt bzw. Problematiken aufzeigen zu können. Zukünftig sollte die Projektplanung allerdings besser behalten und gemanagt werden, sodass alle Teammitglieder über die klare Struktur und den Ablauf informiert sind. Empfehlen würde sich, die Festlegung eines Projektmanagers und Zeitmanagement-Tools. Zudem sollten alle Arbeitspakete bereits vor der Projektphase *Entwicklung* definiert werden, sodass hier genügend Zeit für die Bearbeitung eingeplant werden kann.

## 5 FAZIT

Für die abschließende Evaluierung sollen die gestellten Fragen im Bereich Evaluierungsmethode aufgegriffen und erörtert werden. Die Fragen dienen als Leitfaden und sollten durch die Evaluierung beantwortet werden. Zunächst sollte erörtert werden, warum die Evaluierung durchgeführt wurde. Hier wurden die drei Bereiche, das subjektive Empfinden des Teams, Abweichung der Phasen 1 & 2 und das Erreichen von Teilergebnissen in Phase 2 genannt. Gründe für diese Problematiken wurden bereits vielfältig im Abschnitt der Evaluation genannt. Als einer der Hauptgründe könnte man das fehlende Projektmanagement, insbesondere in der zweiten Projektphase, identifizieren. Zusätzlich gab es viele technische Schwierigkeiten, die vorab in der ersten Projektphase nicht ausreichend getestet bzw. untersucht wurden, sodass während der zweiten Projektphase hier spontan neue Ansätze verwendet werden mussten. Die Konzeption wurde bereits in der Projektphase *Vision & Konzept* sehr technisch ausgeführt, wodurch eine Abweichung der vorgeschlagenen Technik nicht alle gestellten Anforderung abdeckt bzw. nur teilweise abdeckt. Das Ergebnis der Projektphase *Entwicklung* kann als nicht vollendet bzw. nur teilweise gelungen eingestuft werden. Ansätze aus der Konzeption wurden umgesetzt bzw. im Rahmen der Kernidee wurde das Projekt umgesetzt. Jedoch fehlte von den drei festgelegten Bereichen, *Kamera*, *Objekterkennung* und *Benutzerschnittstelle*, der dritte Bereich, sodass das Projekt unvollständig ist.

Die gewonnen Erkenntnisse aus der Evaluation können für die Erstellung eines Redesigns verwendet werden und können auch für andere Projekte als Leitfaden zur Problemvermeidung dienen. Dabei ist darauf hinzuweisen, dass hier nicht alle Aspekte der drei Schwerpunkte Planung, Implementierung und Managements erörtert wurden.

## LITERATUR

- [1] Microsoft. 2022. Azure Kinect DK. <https://azure.microsoft.com/de-de/services/kinect-dk/> Zuletzt abgerufen 23 Februar 2022.
- [2] Microsoft. 2022. Windows Presentation Foundation. <https://docs.microsoft.com/de-de/visualstudio/designers/getting-started-with-wpf?view=vs-2022> Zuletzt abgerufen 23 Februar 2022.
- [3] Open3D. [n.d.]. Surface Reconstruction. [http://www.open3d.org/docs/latest/tutorial/Advanced/surface\\_reconstruction.html](http://www.open3d.org/docs/latest/tutorial/Advanced/surface_reconstruction.html) Zuletzt abgerufen 20 Februar 2022.
- [4] Open3D. 2022. Open3D Website. <http://www.open3d.org/> Zuletzt abgerufen 26 September 2021.
- [5] OpenCV. 2022. OpenCV Surface Matching. [https://docs.opencv.org/4.x/d9/d25/group\\_\\_surface\\_\\_matching.html](https://docs.opencv.org/4.x/d9/d25/group__surface__matching.html) Zuletzt abgerufen 23 Februar 2022.
- [6] OpenCV. 2022. OpenCV Website. <https://opencv.org/> Zuletzt abgerufen 23 Februar 2022.

[7] Martina Pezzati. 2015. Leitfaden für die Projektevaluation. [https://www.bne-brandenburg.de/materialien/2015\\_leitfaden\\_projektevaluation.pdf](https://www.bne-brandenburg.de/materialien/2015_leitfaden_projektevaluation.pdf) Zuletzt abgerufen 20 Februar 2022.

[8] PointCloudLibrary. 2022. PointCloudLibrary Website. <https://pointclouds.org/> Zuletzt abgerufen 23 Februar 2022.