

Erstellung und Untersuchung einer VR- Anwendung auf die realistischen Visualisierungsmöglichkeiten von Materialien

Bachelor-Thesis

Zur Erlangung des akademischen Grades B.Sc.

Peter Oetker

2240231



Hochschule für Angewandte Wissenschaften Hamburg
Fakultät Design, Medien und Information
Department Medientechnik

Erstprüfer: Prof. Dr. Roland Greule
Zweitprüfer: Prof. Dr. -Ing. Sabine Schumann

Hamburg, 25. August 2020

Inhaltsverzeichnis

1	Einleitung	5
2	Grundlagen	6
2.1	Virtual Reality	6
2.1.1	Begriffserklärung von Virtual Reality.....	6
2.1.2	Präsenz und Immersion	7
2.1.3	Head-Mounted Displays	7
2.2	Photogrammetrie	8
2.2.1	Funktionsprinzip	8
2.2.2	Photogrammetrische Verfahren.....	10
2.3	Terrestrisches 3D-Laserscanning (TLS).....	11
2.3.1	Funktionsprinzip	11
2.3.2	Messverfahren.....	12
2.3.2.1	Impulslaufzeitverfahren.....	12
2.3.2.2	Phasenvergleichsverfahren.....	13
2.3.3	Lichtablenkung	14
2.3.4	Georeferenzierung der Punktwolken.....	14
2.3.5	Scann-Systeme	15
2.3.5.1	Airborn Laserscanner	15
2.3.5.2	Handgeführte 3D-Laserscanner	15
2.3.5.3	Tablets mit Photogrammetrie.....	16
2.3.6	Gerätetypen	17
2.3.6.1	FARO Focus ^s 70	17
2.3.6.2	Geoslam ZAB-REVO.....	18
2.3.6.3	EVA Handheldscanner.....	19
3	Umsetzung.....	20
3.1	Datenerfassung	20
3.2	Datenauswertung.....	21
3.2.1	Artec Professional 10.....	21
3.2.2	Blender	23
3.2.3	Photoshop	26
3.3	Evaluation der Scans	27
3.3.1	Analyse der Meshes	27
3.3.2	Analyse der Eva Scan Texturen	30
4	Simulation	33
4.1	Zielhardware	34
4.1.1	HTC Vive.....	34
4.1.2	Oculus Quest	35
4.2	FTZ Digital Reality Hafencity.....	36
4.3	Unity Engine	37
4.4	Implementation der Zielhardware	38
4.5	Assets/Interaktionsmöglichkeiten	38
4.6	Implementation der Modelle.....	39
4.7	Umgebung.....	41
4.8	Beleuchtung	44

5	Fazit	45
6	Aussichten	46
	Abbildungsverzeichnis	47
	Tabellenverzeichnis	48
	Literatur	49

Abstract

This bachelor thesis analyses the possibilities of visualization of materials in a virtual reality application. The models and textures are developed through a handheld photo- and laserscanner. The records and the processing will happen in Artec 3D software in real-time. The further processing and polishing will be made in blender.

The simulation is done in the Unity engine and will take place in the Forschungs- und Transferzentrum Digital Reality in the Speicherstadt Hamburg.

Zusammenfassung

In dieser Bachelorarbeit wird die Möglichkeit der Visualisierung von Materialien in einer Umgebung untersucht. Die 3D-Modelle und die dazugehörigen Texturen werden mit einem tragbaren Foto- und Laserscanner erstellt. Die Aufnahme und Bearbeitung der Rohdateien erfolgt in der Artec 3D Software. Die weitere Bearbeitung und Feinarbeit werden in Blender stattfinden.

Die Darstellung erfolgt in einer virtuellen interaktiven Simulation des Forschungs- und Transferzentrums Digital Reality der Speicherstadt in der Unity Engine.

1 Einleitung

Game-Engines bieten durch eine fortlaufende Entwicklung immer bessere Möglichkeiten zur Erzeugung und Visualisierung digitaler Inhalte. Die Virtuelle Realität (VR) spielt dabei eine immer größer werdende Rolle für simulierte Umgebungen. Scanner mit Laser- und Bildsensoren bieten einen weiteren Schritt für die Erstellung realistisch wirkender 3D-Objekte.

Die Photogrammetrie (auch Fotogrammetrie) wurde ursprünglich für die Auswertung fotografischer Bilder für die Vermessung und Lagebestimmung von Objekten genutzt.

Durch moderne terrestrische Lasercanner und Bildverarbeitungssoftware hat die Photogrammetrie neue digitale Anwendungsbereiche geschaffen. So können aus Scannerdatensätzen erstellte Objekte, Zeit- und Kostenfaktoren einsparen. Architekturbüros können Kunden*innen ihr zukünftiges Haus bereits veranschaulichen, ohne dass dieses gebaut werden musste.

Automobilkonzerne nutzen diese auch für dreidimensionale Darstellungen im Verkauf. können sich ein Wunschauto in passender Farbe direkt auf einem Touchscreen zusammenstellen. Durch die Einbindung in VR bietet sich die Möglichkeit diese Erfahrungen hautnah zu erleben. Im Bereich der Stadt- und Gebäudeplanung wird dies immer wichtiger. Sieht man sich in einer Augmented Reality (AR) oder einer VR-Anwendung bereits ein 3D-Model des Gebäudes, mit verlaufenden Rohren, Kabeln und Leitungen an, lässt sich zeit- und kosteneffizienter planen und arbeiten. Analoge Pläne auf Papier fallen weg und entlasten die Bürokratie.

Durch ein sogenanntes BIM-Model (Building-Information-Modelling), kann das zuständige Personal jederzeit über einen Computer zugreifen und sieht sofort, wo sich die einzelnen Elemente befinden.

Für die VR-Anwendung haben im Rahmen dieser Arbeit die Firma Dataport und MINIMYSET die Räumlichkeiten des FTZ Digital Reality in der Hafencity gescannt und die Rohdateien an die Studierenden Digital Reality weitergegeben. Diese erstellen aus den Rohdateien ein 3D-Model, welches in eine interaktive VR-Anwendung eingebettet wird.

Die Untersuchung dieser Arbeit liegt darin, zu analysieren wie gut sich Materialien auf gescannten Objekten in VR visualisieren lassen. Genutzt wird hierfür die Unity-Engine. Die VR-Interaktion erfolgt über die HTC-Vive und die Oculus Quest. Es werden hierfür mehrere eigene Objekte mit einem tragbaren Scanner Eva erstellt und überarbeitet. Für eine größere Auswahl werden mehrere Varianten der originalen Materialien erstellt. Über ein User Interface (UI), das über die Controller gesteuert wird, sollen diese an einigen Objekten frei wählbar sein.

In dieser Arbeit wird am Anfang auf die Grundlagen der Photogrammetrie und Laserscanner eingegangen. Hier werden verschiedene Messverfahren erläutert. Neben dem Artec Eva Handheld Scanner werden auch der FARO Scanner und der Geoslam Zeb Revo vorgestellt. Im darauffolgenden Kapitel wird auf die Durchführung der Datenerfassung des Eva Scanners eingegangen. In der Datenauswertung werden die Arbeitsschritte von der Punktwolke, bis hin zum fertigen Model erläutert. Hier wird auch auf die Artec Studio 10 Software und blender eingegangen. Im letzten Kapitel wird die Erstellung der VR-Anwendung in Unity beschrieben, auf die ein Analyse der Umsetzung der Materialien folgt.

2 Grundlagen

In dem folgenden Kapitel wird auf die Grundlagen der im Rahmen dieses Projektes genutzten Technologien und Software eingegangen. Beispiele und Vergleiche werden aufgeführt, sofern sie vorgenommen wurden

2.1 Begriffserklärung Virtual Reality

Unsere Sinneswahrnehmung verbindet unser Gehirn, über ausgelöste Impulse mit unserer Umwelt. Hätte der Mensch seine fünf Sinne nicht, könnte er nicht mit der Umgebung interagieren. Dies führt zu einem fehlenden Feedback dieser und somit zu fehlenden Informationen für das menschliche Gehirn.

Die wichtigste Wahrnehmung ist dabei das Sehvermögen. Unser Auge kann über Millionen Rezeptoren eine Fülle an Informationen aus Umgebung und Objekten filtern. Diese kann das menschliche Gehirn in ein dreidimensionales Bild umwandeln.

In der VR beschäftigt man sich mit der Frage, wie unserer Sinneswahrnehmungen künstlich ausgelöst werden können. Dem Nutzenden soll eine möglichst realistische Umgebung simuliert werden, welche die Sinne täuscht und die Grenzen zwischen Simulation und Realität verwischen lässt. Diese Täuschung der Reize wird auch als „Suspension of Disbelief“ bezeichnet. Die Person wird über ihre Sinne so sehr getäuscht, dass sie die Simulation nicht mehr von der Realität unterscheiden kann¹

2.1.1 Begriffserklärung von Virtual Reality

Die technologische Charakterisierung von VR Umgebungen bezieht sich auf die Ein- und Ausgabegeräte. Dazu gehören die VR-Brillen (Head-Mounted Displays), Controller-Typen, Datenhandschuhe und haptische Anzüge, die eine Interaktion mit der virtuellen Umgebung möglich machen. Somit lässt sich eine Charakterisierung anhand technologischer Aspekte vollziehen. Jedoch wird dies auch durch den Fakt infrage gestellt, dass sich Technologien rasant weiterentwickeln und somit neue technologische Möglichkeiten, alte Systeme und Technologien ablösen.

Für zukunftssichere Definitionen sollten mögliche visionären Vorstellungen mit einbezogen werden. Ein Beispiel hierfür wäre das Holodeck aus Star Trek. Einige dieser technologieorientierten Charakterisierungen aus den vorherigen Jahren treffen jedoch noch heute zu.²

„Virtual Reality refers to immersive, interactive, multi-sensory, viewer-centered, three-dimensional computer generated environments and the combination of technologies required to build these environments.“

(Carolina Cruz-Neira, Quelle: Dörner 2017)

¹ Dörner, 2017.

² Dörner, 2017.

2.1.2 Präsenz und Immersion

Wie bereits zu Anfang beschrieben liegt ein wesentlicher Teil von VR in der Illusion einer realistischen Umwelt in einer künstlichen 3D-Umgebung.

Nutzende tauchen vollständig in die virtuelle Welt ein und werden von der realen Welt losgekoppelt. Der Begriff Präsenz erläutert dabei die eigene Wahrnehmung in der virtuellen Umgebung. Je höher der Grad der Immersion, desto höher ist die subjektive Wahrnehmung (Präsenz) in einer VR Umgebung. Präsenz wird durch ein bewusstes Eintauchen in die virtuelle Welt erzeugt, in der sich die Testpersonen wie in der realen Welt verhalten.

Die subjektive Wahrnehmung der eigenen Präsenz ist jedoch individuell stark ausgeprägt. Nutzende mit einem hohen Grad an Präsenz werden körperlich entsprechend auf bestimmte Situationen reagieren. So weisen Menschen bei Höhengsimulationen (Fliegen, Flugzeug, Wolkenkratzer), bei einer stark ausgebildeten Selbstwahrnehmung, einen erhöhten Herzschlag oder ein Zittern auf.³

2.1.3 Head-Mounted Displays

Head-Mounted Display (HMDs) sind spezielle visuelle Ausgabegeräte für die Erzeugung einer immersiven virtuellen Umgebung. Für die Erzeugung eines räumlichen Eindruckes werden zwei Bilder aus unterschiedlichen Perspektiven erzeugt (Steropjektion).

Das Display sowie integrierte Sensoren für die Erkennung der Position und das Controller-Tracking werden über die HMD interne Elektronik gesteuert. Die HMDs und dem Computer wird über ein USB.3- zu USB.C- Kabel hergestellt. Da es eine breite Auswahl an HMDs und deren Anwendungsbereichen existieren, werden folgend einige allgemeine Kenngrößen und Eigenschaften erläutert.

Das Gefühl der Präsenz wird stark von dem sogenannten FOV (Field of View) bestimmt (siehe Abb. 2.1). In diesem werden die horizontalen und vertikalen Winkel ausgehend vom Auge beschrieben. Je größer der FOV ist desto größer die Weiträumigkeit der virtuellen Anwendung. Dies führt zu einer stärkeren Wahrnehmung. Bei durchsichtigen HMDs wird der FOV der realen Welt mit in den Bezug aufgenommen. Weitere Einflussfaktoren sind die Homogenität des Bildes und die Frontleuchte des HMDs. Die Frontleuchte bestimmt dabei die Helligkeit des Bildes.⁴

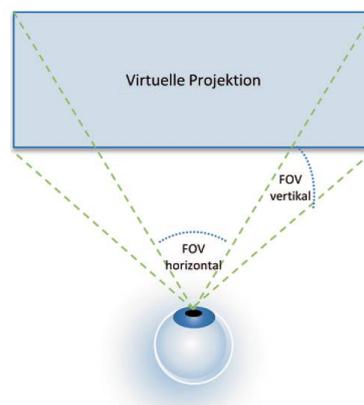


Abbildung 2.1: Funktionsprinzip FOV (Quelle: Dörner, 2017)

³ Dörner, 2017.

⁴ Dörner, 2017.

2.2 Photogrammetrie

Die Photogrammetrie ist ein berührungsloses optisches Messverfahren für die Bestimmung der Lage, Form und Größe eines Objektes aus zweidimensionalen Fotografien. Diese entstehen durch die Beleuchtung des Objektes durch eine Lichtquelle. Weitere wichtige Einflussfaktoren sind die Oberflächenbeschaffenheit des Objektes, Eigenschaften der Hardware und vorhandene Umweltfaktoren. Dazu zählen die Luft und transparente Materialien (Wasser, Kameralinse). Unterschieden wird meistens zwischen der terrestrischen Photogrammetrie und der Aerophotogrammetrie.

Bei der terrestrischen Photogrammetrie werden Bilder von stationären Standpunkten vom Boden aus aufgenommen. Die Aerophotogrammetrie beschäftigt sich meist mit der Aufnahme von topografischen Karten, die von Flugzeugen aus aufgenommen werden.

Für die Erstellung des Bildmaterials werden heutzutage Digitalkameras zur Rekonstruktion des Objektes verwendet. Nicht sichtbare Elemente können nicht rekonstruiert werden. Dieser Informationsverlust gilt auch für verdeckte Teile und Oberflächen die aufgrund ihrer Beschaffenheit (zu hoher Kontrast, zu klein) nicht erfasst werden können. Beispielsweise einzelne Bauteile von Gebäuden oder industriellen Objekten. Berücksichtigt werden neben den genannten Faktoren auch Erfahrung und Handeln des Menschen.⁵

2.2.1 Funktionsprinzip

Das Messverfahren der Photogrammetrie wird abgeleitet von dem mathematischen Verfahren der zentralprojektiven Abbildung. Ein abgebildeter Punkt P im Raum wird durch eine Gerade von dem Projektionszentrum O' als Punkt in der Ebene dargestellt. Jeder Punkt P im Raum entspricht so einem Punkt auf der Ebene im Bild P' . Die zurückgelegte Strecke der Gerade die durch das Projektionszentrum O' und den Bildpunkt P' entspricht somit unendlich vielen Punkten des Raumes (siehe Abbildung 2.2). Die Lage, Form und Größe eines Objektes werden mithilfe überschneidender Strahlenbündel bestimmt. Diese müssen unterschiedlich positioniert sein. Der Schnittpunkt zweier sich überschneidender Strahlen ergibt einen Punkt (Objektpunkt) im Bildkoordinatensystem der Kamera. Für die Bestimmung der 3D-Koordinaten aus den Strahlen müssen die Parameter der Kamera (innere Orientierung) und die Lage der Kamera im Raum (äußere Orientierung) bekannt sein.

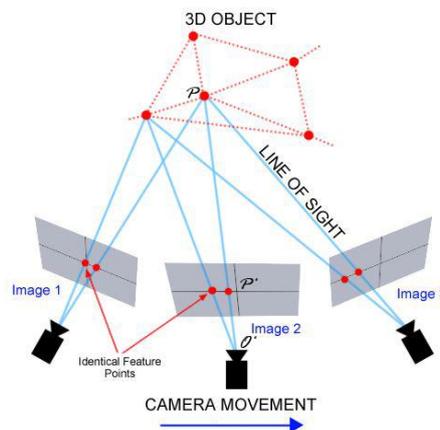


Abbildung 2.2 Funktionsprinzip der zentralprojektiven Abbildung (Quelle: researchgate, Bitenc)

⁵ Luhmann, 2018.

Die innere Orientierung beschreibt die Abweichungen (Abbildungsfehler) und die Lage des Punktes O' in einem Bildkoordinatensystem. Das Projektionszentrum O' erhält die höchste Wertigkeit. Wichtigste Bezugsgröße der inneren Orientierung ist die Kamerakonstante c . Diese definiert den Abstand zwischen Bildpunkt P' und dem Projektionszentrum O' . Durch ihn verlaufen alle Strahlen geradlinig. Diese Annahme entspricht dem Model einer Lochkamera (siehe Abbildung: 2.3). Diese weicht jedoch stark von heutigen Kamertypen ab. Instabilität und Ausrichtung der Kamera auf einem Stativ können dafür Auslöser sein. Um Abweichungen zu vermeiden werden die inneren Parameter für jedes Gerät entsprechend kalibriert.

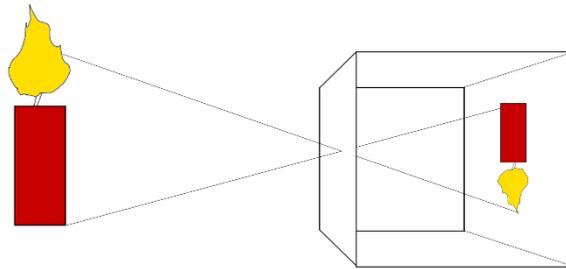


Abbildung 2.3: Lochkamera Funktionsprinzip (eigene Darstellung)

Die Lage der Kamera in einem Raum wird über die Informationen der äußeren Orientierung abgerufen und in einem übergeordneten Koordinatensystem festgelegt. Diese werden über die bekannten Objektpunkte ermittelt. Das übergeordnete Koordinatensystem und das Bildkoordinatensystem werden nun mit dem Projektionszentrum als gemeinsamen Bezugspunkt in einen Zusammenhang gebracht. Die gemessenen Bildpunkte entsprechen dabei einer räumlichen Richtungsmessung vom Projektionszentrum aus. Die Länge des Strahls ist dabei erstmal nicht bekannt, so dass auf dem Strahl liegende Objektpunkte zu den ihnen zugewiesenen Bildpunkten führen. Für die absolute Festlegung des Objektpunktes sind weitere Parameter erforderlich (zum Beispiel.: eine Raumgerade oder Objektebene).⁶

⁶ Kraus, 2004.

2.2.2 Photogrammetrische Verfahren

Unterschieden wird in drei technische Verfahren. Die analoge Photogrammetrie befasst sich mit der Auswertung der Bilder von optischen Geräten. Diese ist jedoch veraltet und wurde durch die analytische und digitale Photogrammetrie komplett ersetzt. Die analytische Photogrammetrie bezieht sich dabei auf die Auswertung analoger Bilder unterstützt durch digitale Rechenverfahren. Die digitale Photogrammetrie umschreibt die Auswertung digitaler Bilder mit digitalen Systemen. Die digitale Photogrammetrie hat die beiden anderen Verfahren heutzutage fast vollkommen abgelöst.⁷

Analytische Photogrammetrie	Digitale Photogrammetrie
Auswertung analoger Bilder	Digitalisierung der Bilder erforderlich
Sehr gute Bildwiedergabe	Sehr gute Bildwiedergabe
Sehr guter räumlicher Eindruck	Sehr guter räumlicher Eindruck
Keine nachträgliche Bildverbesserungen möglich	Nachträgliche digitale Bildverbesserung möglich
	<ul style="list-style-type: none"> • Kein physischer Umgang mit Fotografien • Speicherung der Bilder • Kombination mehrerer Messverfahren • Mehrbildverarbeitung • Digitale Editierung • Keine Kalibrierung der Auswertegeräte nötig

Tabelle 2.1: Unterschiede analytischer und digitaler Photogrammetrie
(Quelle: Bundesministerium der Verteidigung, 2015)

⁷ Bundesministerium der Verteidigung, 2015

2.3 Terrestrisches 3D-Laserscanning (TLS)

Laserscanning ist eine polare Messmethode, die Daten flächenhaft aufnimmt und auswertet. Sie wird deshalb auch als polare Photogrammetrie bezeichnet. Laserscanning- und Photogrammetrieverfahren werden auch kombiniert in modernen terrestrischen gestützten Laserscann-Systemen genutzt. Diese werden in erster Linie für die Gewinnung geometrischer Information für die Generierung von 3D-Punktwolken genutzt, die in ein dreidimensionales Koordinatensystem übertragen werden. Laserscanner nutzen einen ausgesendeten Lichtstrahl für die aktive Beleuchtung eines Objektes. Dabei wird aus der reflektierten Strahlung die Distanz zu dem Objekt gemessen.

Ein Unterschied zu der klassischen Photographie ist die sequenzielle Messung der aufgenommenen Daten. In der Photographie ist die Messung der einfallenden Strahlung für alle Pixel gleich.

2.3.1 Funktionsprinzip

Laserscanner bestehen aus einem System zur elektro-optischen Distanzmessung (EDM) und einer Ablenkeinrichtung, welche Raumsegmente in regelmäßigen Intervallen abtastet. Jeder durch den Laser abgetastete Punkt des Objektes wird dabei in ein elektrisches Signal umgewandelt. Passiert das Licht die Glasscheibe des Laserkopfes, werden Umgebungsstrahlen aus externen Lichtquellen herausgefiltert. Messergebnisse werden dabei in der Matrix eines Koordinatensystems gespeichert. Die Zeilen- und Spaltenrichtung entsprechen dabei zueinander orthogonalen Raumwinkeln. Die gespeicherten Daten entsprechen dabei zumindest der Strecke zur Objektoberfläche, die mit einem Phasenvergleichsverfahren oder einem Laufzeitverfahren gemessen wird. Diese werden in den folgenden Kapiteln näher beleuchtet.

Das vertikale Ablenksystem des Laserscanners bestimmt die genaue vertikale Ausrichtung des Strahls zu dem gewünschten Messpunkt auf dem Objekt. Das horizontale Ablenksystem ist für die Drehung des Kopfes auf der horizontalen Achse zuständig. Beide Systeme werden über interne Motoren für die Ausrichtung gesteuert und korrigieren die jeweilige Ausrichtung anhand von Spiegeln. Systeme mit integrierter Kamera nehmen zusätzlich Fotos von dem Objekt auf, um später Texturinformationen über das Objekt (Mesh) legen zu können.

Dabei wird nicht jeder aufgenommen Scan im Raum mit einer Texturinformation (Farbwert) versehen. Die Zuordnung dieser erfolgt automatisiert im eingebauten Computerteil. Für eine Echtzeitgenerierung von Punktwolken wird ein externer Computer oder ein Tablet benötigt. Dies hat auch den Vorteil, den Verlauf zu beobachten und direkt korrigieren zu können. Dies ist vor allem ein Vorteil bei Handheld- Laserscannern (tragbare Laserscanner).

Manche Laserimpulse gelangen jedoch nie zurück. Dies liegt an stark reflektierenden oder zu transparenten Oberflächen. Zu stark Licht absorbierende Materialien sind ebenso ein Grund für Fehlinterpretationen des Lasers. So können Entfernungen zu groß oder zu klein berechnet werden. Wasser und Glas sind Beispiele dafür.

Zu dem weiteren Zubehör gehören eine Speicherkarte, Akku und auch teilweise direkt integrierte Touch-Displays für die Anpassung der internen Kalibrierung der Komponenten. Moderne Systeme verfügen jedoch meist über eine automatisierte Kalibrierung.⁸

⁸ Luhmann, 2018.

2.3.2 Messverfahren

Die Basis der Laserscanning-Funktionsweise stellt das Aussenden, Empfangen und Auswerten eines Lichtsignals dar. Dabei wird ausgenutzt, dass die meisten Materialien in der Lage sind, ein Lichtsignal reflektieren zu können.

Beim Laserscanning können verschiedene Parameter beim ausgesendeten Lichtsignal und bei der Art des Empfangs verändert werden. Daraus leitet sich wiederum die Gliederung in Impulsmessverfahren und Phasenvergleichsmessverfahren ab.

2.3.2.1 Impulslaufzeitverfahren

Das Impulslaufzeitverfahren nutzt für die Messung der Distanz Lichtimpulse, die in kurzen Abständen ausgesendet werden (siehe Abb. 2.4). Dabei wird der zeitliche Unterschied zwischen dem Senden und dem Empfangen des reflektierten Lichtsignals gemessen.

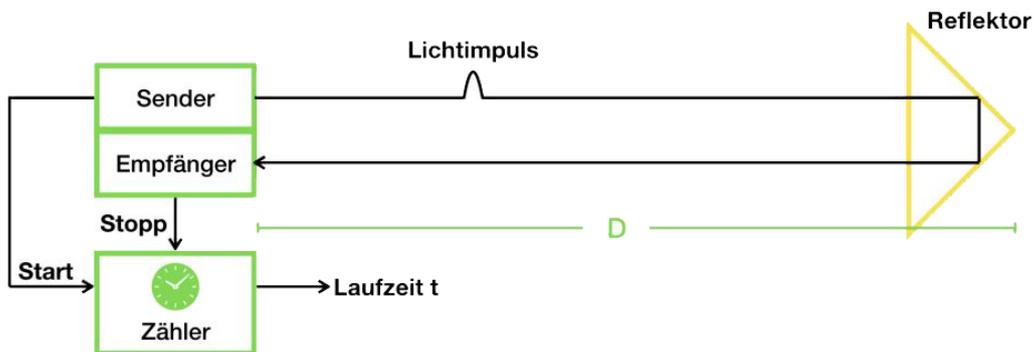


Abbildung 2.4: Funktionsprinzip des Impulszeitlaufverfahrens (Quelle: Möllring, 2018)

Ein Zähler berechnet dabei die Laufzeit t . Die Laufzeit entspricht dabei der Zeit, die der Lichtimpuls für die Distanz D benötigt. Die Distanz berechnet sich dabei aus der Laufzeit t und der Lichtgeschwindigkeit c .⁹

$$D = \frac{c * t}{2} \quad (1)$$

⁹ Möllring, 2018.

2.3.2.2 Phasenvergleichsverfahren

Das Phasendifferenzverfahren nutzt für die Berechnung von Strecken die einzelnen Phasen von Lichtwellen. Die Veränderungen dieser Phasen sind abhängig von der Distanz des Objektes. Dabei wird ein Lichtstrahl von einem Sender zu einem Objekt ausgesendet. Das reflektierte Licht wird von einem Empfänger in einem Tachymeter aufgenommen und seine Phasenunterschiede analysiert.

Durch die Distanz verändern sich diese Phasen. Diese Distanz lässt sich wiederum aus den Phasenveränderungen ableiten.¹⁰

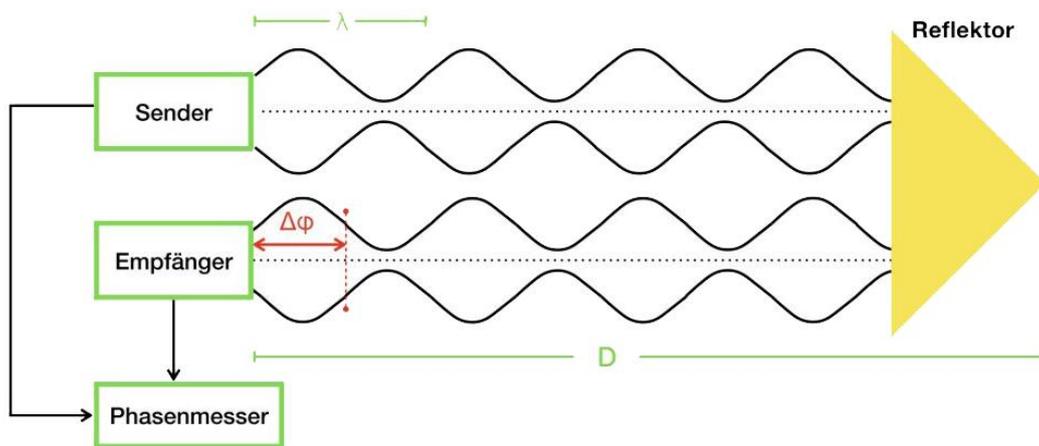


Abbildung 2.5: Funktionsprinzip des Phasenvergleichsverfahrens (Quelle: Möllring, 2018)

Für die Berechnung der Strecke werden die Messgrößen des Phasenunterschiedes $\Delta\phi$ und die Anzahl der ganzen Wellenlängen u benötigt.

$$2D = u \cdot \lambda + \frac{\Delta\phi}{2\pi} \cdot \lambda \quad (1)$$

Die Wellenlänge wird dabei aus den Größen der Modulationsfrequenz und der Lichtgeschwindigkeit festgelegt.

$$\lambda = \frac{c}{f} \quad (2)$$

Die Lichtgeschwindigkeit bezieht sich dabei auf die Ausbreitung von Licht im Vakuum c_0 und unter der Berücksichtigung des Brechungsindex n . c_0

$$c = \frac{c_0}{n} \quad (3)$$

¹⁰ Möllring, 2018.

2.3.3 Lichtablenkung

Die Generierung von 3D-Objekten setzt voraus, dass Laserstrahlen über die Umgebung geführt werden, um diese profiartig abtasten zu können. Hierfür werden speziell beschichtete Spiegel eingesetzt (siehe Abb. 2.6). Diese besitzen eine stark reflektierende Oberfläche und werden auch als Strahlableinheiten bezeichnet.

Die Winkelposition des Reflektors wird dabei benötigt, um die Ausrichtung des Laserstrahls im Raum bestimmen zu können. Aus diesem Grund sind die Ablenkeinrichtungen und die dazugehörigen Drehmotoren mit hochauflösenden Encodern versehen. Die Scanfrequenz ist dabei in einer Abhängigkeit zu der Rotationsfrequenz der Ablenkeinrichtung.¹¹

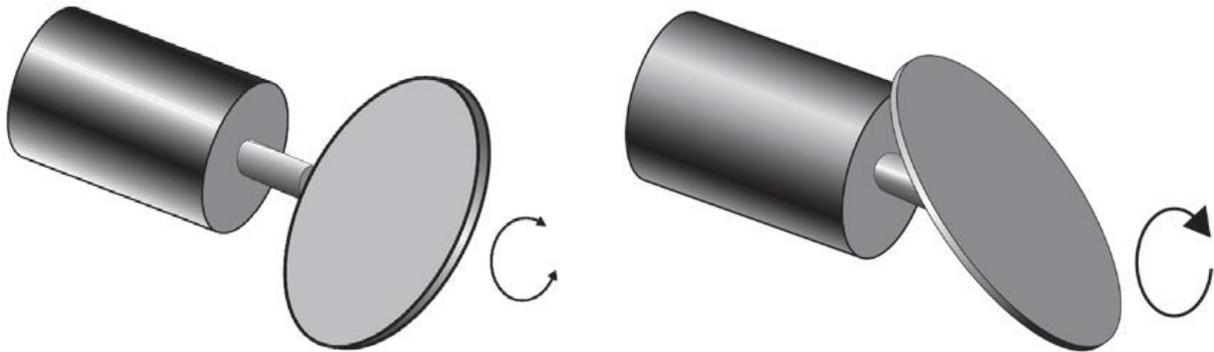


Abbildung 2.6: Scanspiegel (Quelle: DVW e.V, 2017)

2.3.4 Georeferenzierung der Punktwolken

Um einzelne Punktwolken verknüpfen und das erfasste Objekt später im Raum verorten zu können, muss die Position des Scanners in Bezug auf ein globales Koordinatensystem exakt bekannt sein. Dazu werden heute unterschiedliche Methoden und Ansätze verfolgt. So kann der Laserscanner direkt mit einem entsprechenden Orientierungs- und Positionierungssystem verbunden werden (direkte Registrierung und Georeferenzierung), entsprechende Messmarken mit bekannten Objektkoordinaten einbezogen werden (indirekte Registrierung und Georeferenzierung) oder homologe Punkte in den Punktwolken identifiziert werden (datengetriebene Registrierung und Referenzierung).¹²

¹¹ DVW e.V, 2017.

¹² DVW e.V, 2017.

2.3.5 Scann-Systeme

Laserscanner werden in verschiedene Typen unterteilt. Diese werden im Folgenden weitestgehend beschrieben und deren technische Daten erläutert. In dem darauffolgendem Kapitel wird auf einige der im Umfang dieses Projekts verwendeten Gerätetypen eingegangen.

2.3.5.1 Airborne Laserscanner

Flugzeuge nutzen Airborne Laserscanner Systeme (ALS) als Methode der Fernerkundung. Diese nutzen Höheninformationen für die Erstellung von Scans diverser Anwendungsgebiete. Man unterscheidet hierbei zwischen zwei Typen von Sensoren. Discret Echo Sensoren und den Full-waveform Systemen. Ersteres erfasst nur eine geringe Anzahl von Echos, während zweiteres die gesamte Signalstärke permanent erfassen kann. Das Full-waveform System kann dadurch zusätzlich die Signalamplitude und die Echobreite ableiten.

Wichtige Parameter für die Erfassung sind die Punktdichte, die Laser Footprints und die Signalamplitude. Die Punktdichte ist abhängig von weiteren Parametern wie der Flughöhe, Sichtfeld und Geschwindigkeit. Die Laser Footprint beschreibt die beleuchtete Erdoberfläche, resultierend aus der Strahlqualität. Diese beschreibt die Fokussierbarkeit des Laserstrahls und wie schnell er sich bei der Ausbreitung relativ zu seinem Durchmesser ausbreitet.

ALS Systeme werden vor allem in Flugzeugen und Helikoptern benutzt. Heutzutage sind mit Drohnen ein weiteres Anwendungsfeld hinzugekommen. Benutzt werden ALS Systeme überwiegend für die Erfassung von Terrain in der Geografie aber auch für die Erfassung von Gebäuden, sowie in Bereichen von Forst- und Landwirtschaft.

2.3.5.2 Handgeführte 3D-Laserscanner

Sogenannte Handheld Laserscanner (HLS) sind kleine Ausführungen für den handlichen Gebrauch und gehören zu den terrestrischen Laserscannern. Diese funktionieren mit einem mobilen Akku und Tablet auch ohne Verbindung mit einem Computer.

HLS haben dabei eine Reichweite zwischen 0,5m und 2m Entfernung, bei einer Auflösung zwischen 0,1mm und 0,5mm. Die Genauigkeit kann ebenso zwischen 0,05mm und 0,1mm betragen. HLS besitzen dazu eine Echtzeitgenerierung der Daten, die eine permanente Beobachtung des Fortschritts erlauben. Ihre Beschaffenheit erlaubt die Aufnahme für kleine bis mittlere Objekte. Die häufigsten Anwendungsgebiete finden sich daher in der Medizin, der industriellen Fertigung und dem Denkmalschutz.¹³

¹³ Artec3D, 2020.

2.3.5.4 Tablets mit Photogrammetrie

Im Vergleich zu den hohen Anschaffungskosten üblicher Scanner, können kostengünstigere Tablets mit Photogrammetrie Software ausgestattet werden. Dies hat hohe Qualitätseinbußen zur Folge, ist jedoch für den Gebrauch bei Amateuraufnahmen die bessere Alternative. Dafür muss das Tablet nur eine Kamera mit guter Qualität besitzen.

2020 stellte Apple sein neues iPad Pro mit neuer Ausstattung vor. Dieses enthält ein Kamerasystem mit Weitwinkel- und Ultraweitwinkelkamera, sowie einen LiDar-Sensor („*Light Detection and Ranging*“). Im Funktionsprinzip agiert dieser ähnlich wie ein Radar, außer dass ein Laserstrahl für die Entfernungsbemessung angewendet wird. Die Nutzung ist jedoch auf eine Reichweite von 5 Metern beschränkt. Ein weiteres Anwendungsfeld findet sich in der Automobilindustrie (siehe Abb. 2.7).¹⁴

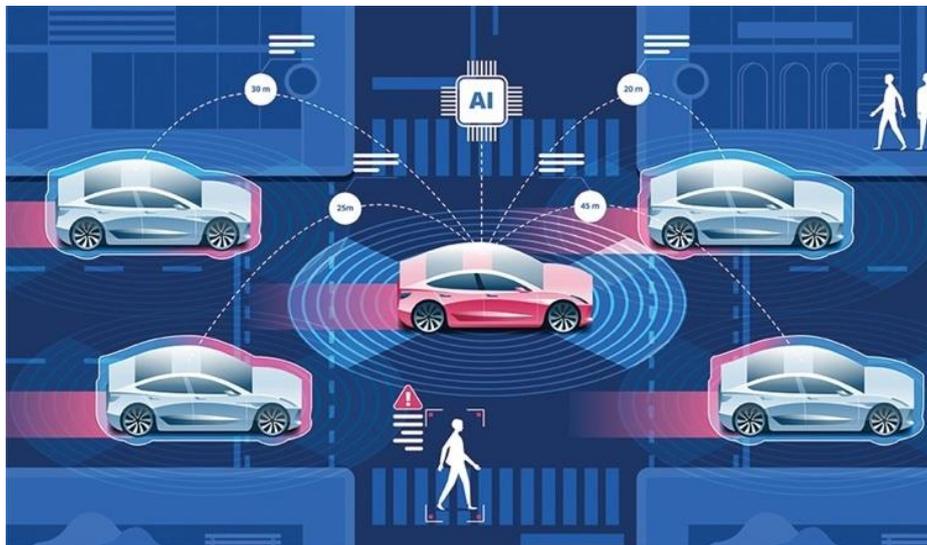


Abbildung 2.7: Anwendungsgebiet eines LiDar-Sensors in der Automobilbranche (Quelle: Smith, 2019)

¹⁴ Smith, 2019.

2.3.6 Gerätetypen

2.3.6.1 FARO Focus^s 70

Der FARO Focus^s 70 (siehe Abb. 2.8) wurde von der Firma Dataport für die Erstellung der Punktwolken des FTZ verwendet. Der Scanner wurde für kurze Reichweiten entwickelt (70m) und ist damit vor allem für den Einsatz im Bereich des Ingenieurs – und Bauwesens geeignet. Zudem entspricht er der Schutzklasse IP54. Dies bedeutet er hat einen Klasse 5 Schutz gegen äußere Umwelteinflüsse wie Staub oder Wasser, die zwar eindringen können, aber dem Gerät aufgrund innerer Abschirmung keinen Schaden zufügen. Ein Klasse 6 Schutz hieße kein Eindringen von Staub möglich wäre. Zudem verfügt der Focus^s 70 über eine HDR-Bildfunktion (High Dynamic Range) und einen erweiterten Temperaturbereich¹⁵

Nutzende haben eine freie Wahl der Software für die Punktwolkengenerierung. FARO bietet hier die eigene Software FARO SCENE oder Autodesk Recap an. Entwickelt mit einer Reichweite von 70m verfügt der FARO Scanner über eine Genauigkeit im Bereich von +/- 1mm. Die Scangeschwindigkeit beträgt bis zu 1 Millionen Punkte die Sekunde.

Dank der kürzlich hinzugefügten FARO SCENE 7.0 kann der Ablauf der Scans auch in Echtzeit verfolgt werden. Die Winkelgenauigkeit des FOCUS^s 70 19 arcsec für vertikale und die horizontalen Winkel. Die Kamera verfügt über 165 Megapixel. Die Abmessungen werden in einem Größenbereich von 230 x 183 x 103mm vorgenommen. Das Gewicht des Scanners beträgt 4,2kg. Des Weiteren verfügt die Vorrichtung über ein GPS, Kompass, einen Höhengsensor und einen Zweiachsenkompensator. Die Handhabung wird über einen integrierten Touchscreen oder über eine WLAN-Verbindung gesteuert¹⁶



Abbildung 2.8: FARO Focus^s 70 (Quelle: FARO, 2020)

¹⁵ FARO, 2020.

¹⁶ FARO, 2020.

2.3.6.2 Geoslam ZAB-REVO

Der ZAB-REVO von der Firma Geoslam gehört wie der Eva Hand-Scanner zu den tragbaren Scanner-Typen (siehe Abb. 2.8) und ist optimal für Räume und Höhlen geschaffen. Das Gewicht beträgt 850g.

Das Innere ist dabei mit der IP 64 Klasse gegen das Eindringen von Partikeln abgesichert. Der Scanner verfügt hierbei eine Geschwindigkeit von 43.200 registrierten Punkten pro Sekunde, bei einer Speicherkapazität von 10MB die Minute und einem Datenspeicher von 55GB bei 55HZ. Die Einsatzfähigkeit liegt bei vier Stunden.

Die Genauigkeit des ZAB-REVO umfasst einen prozentualen Bereich um $\pm 0,1\%$. Die relative Genauigkeit entspricht dabei ca. 1-3 cm. Das Sichtfeld des Scanners beträgt 360 Grad in der Vertikalen und 270 Grad in der Horizontalen. Die Registrierung und Georeferenzierung erfolgt automatisiert. Aufgrund seiner Größe werden die Scans während des Gehens aufgenommen. Dank seiner Echtzeitvisualisierung wird der Fortschritt über ein gekoppeltes Tablet oder ein Notebook verfolgt.

Neben den standartmäßig integrierten Komponenten verfügt das ZAB-REVO System über eine Stange für eine bessere Handhabung für erhöhte und für schwer zu erreichende Bereiche, einen Rucksack für die sichere Lagerung und seit neuestem über eine integrierbare ZAB-CAM. Diese kann Farbinformationen aufzeichnen, die für die Informationen der Textur benötigt werden. Diese Faktoren bieten eine optimale Einsatzmöglichkeit für Räume und Höhlen.¹⁷



Abbildung 2.9: Geoslam ZEB-Revo (Quelle: geoslam, 2020)

¹⁷ Geoslam, 2020.

2.3.6.3 EVA Handheldscanner

Der tragbare Eva Scanner von der Firma Artec 3D ist für die Erstellung von kleinen bis mittelgroße Objekte entwickelt worden. Die Aufnahme findet bei einer Geschwindigkeit von statt. Dabei werden bis zu zwei Millionen Punkten erfasst, bei einer Genauigkeit von bis zu +/- 0,1mm. Die Verarbeitungsgeschwindigkeit beträgt dabei bis zu einer Millionen Punkte die Sekunde in Echtzeit, bei einer Auflösung von bis zu 0,1mm und einem Arbeitsabstand von 0,4 – 1m.

Die Maße betragen 261,5 x 158,2 x 63,7 mm bei einem tragbaren Gewicht von 0,9kg. Kombiniert werden kann dieser mit einem tragbaren Akku und einem Tablet per USB-3 Anschluss. Dies macht den Scanner unabhängig von stationären Rechnern und mobiler in seiner Handhabung.

Neben den optischen Sensoren enthält die Ausstattung eine integrierte Kamera für die Erstellung von farblichen Oberflächeninformationen. Im Vergleich dazu hat der Eva Scanner light nur einen schwarz-weißen Abgleich. Die Verarbeitungskapazität umfasst 40 Millionen Dreiecke die Sekunde bei 1GB RAM. Empfohlen werden jedoch 12GB – 16GB Ram.

Die Auswertung findet in der eigenen Artec Studio Professional Software statt. Alternativ bietet sich auch RealityCapture an. Die Formate der Polygonnetze können ausgegeben werden als: OBJ, PLY, STL, AOP, ASC Disney PTX, E57 oder XYZRGB. Die Anwendungsbereiche finden sich hier vielseitig in der Industrie, Medizin, Kulturschutz, Archäologie oder sogar für die Erstellung von CGI Modellen von Charakteren wieder.¹⁸



Abbildung 2.10: Artec Eva (Quelle: artec3D, 2012)

¹⁸ Artec3D, 2012.

3 Umsetzung

Für die Simulation wurden eigene Testobjekte der ihrer Materialien gewählt Drei Kernobjekte sind ein Hai aus Stoff und zwei Leuchten aus Metall und Glas.

Topografie und Materialeigenschaften beeinflussen den Scanner maßgeblich. Materialien, die zu stark reflektieren oder zu transparent sind brechen das Licht und erschweren es den Sensoren die Oberfläche richtig zu erfassen. Eine weitere Leuchte wurde aufgrund ihrer stark reflektierenden und teils transparenten Oberfläche fast gar nicht erkannt.

3.1 Datenerfassung

Die Aufnahme der notwendigen Rohdaten, bestehend aus Bildmaterial und Laserscan hat mehrere Tage. Die Handhabung braucht einige Zeit der Einarbeitung. Über einen Bildschirm kann man in Echtzeit beobachten, wie der Scanner Teil für Teil scannt und zu einem Objekt zusammensetzt.

Die Schwierigkeit waren hier Platz- und Kabellängenmangel. Die Kabel waren sehr kurz und der Scanner braucht 1m bis 2m Radius Abstand. So verlor er oft den Anschluss und brauch direkt ab. Dadurch entstanden des Öfteren nur Teil-Meshes mit riesigen Löchern.

Der Eva-Handscanner hat neben einer integrierten Kamera und einem Laser-Sensor auch ein Blitzlicht, dass versucht während der Aufnahme ein gleichmäßiges Licht zu erzeugen. Zu dunkle oder zu helle Orte führen zur Abnahme der Qualität. Ein gleichmäßiges Licht erzeugt optimale Bedingungen für die Sensoren und Kamera.

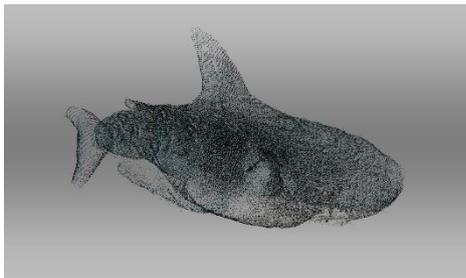
Entwickelt wurde Eva speziell für Objekte von kleiner bis mittlere Größe Er eignet sich damit optimal für die Anwendung an Personen, in der Architektur und der Industrie. Es wurden drei Objekte gescannt, um den Arbeitsablauf nachzuvollziehen und eigene Materialien erstellen zu können. Für die Scans wurden pro Objekt mehrere Anläufe benötigt, da die Qualität ursprünglicher Scans nicht zufriedenstellend war und ein besseres Gefühl der Handhabung auch mit der Übung kommt.

Die Auflösung wurde in 4k (1048x 1048 Pixel) bei einer Punktwolkendicht von ca. 50.000 Polygonen pro Scan erstellt. Da der Eva Scanner in der Handhabung während des Vorganges aufgrund fehlender Präzision öfter Abbruch aufwies, wurden über mehrere Tage mehrere Anläufe benötigt, um ein zufriedenstellendes Ergebnis zu erzielen. Es war schwierig ein Objekt in einem Scan direkt zu erfassen und so wurden meist einzelne Teilwolken erstellt, die zusammengefügt und zu einem Mesh verschmolzen wurden.

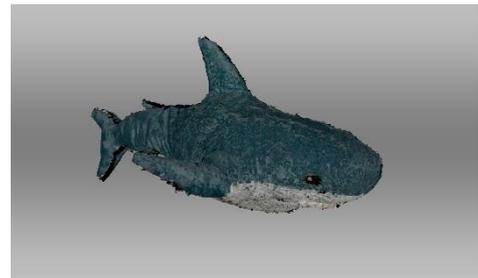
3.2 Datenauswertung

3.2.1 Artec Professional 10

Die aus Scanner und Photogrammetriedaten gewonnenen Informationen können in der Artec eigenen Software zu Punktwolken und fertigen Meshes generiert werden. Diese können für weitere Bearbeitung exportiert und umgewandelt werden. Artec Professional 10 enthält ebenso einige Tools für die Bearbeitung der fertigen Meshes. Folgende Abbildungen zeigen alle Ansichten der Modelle.

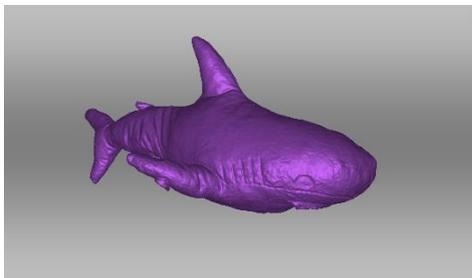


(a) Point Cloud Ansicht

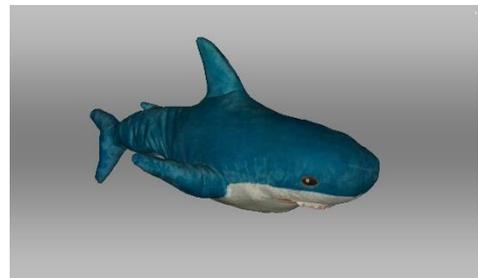


(b) Frame Ansicht

Abbildung 3.1: Darstellung in Schritten (eigene Darstellung des Hai-Scans)



(c) Mesh Ansicht



(d) Mesh Ansicht mit Textur

Meshes die aus zwei oder mehr Teilen bestehen, konnten automatisch oder per Zuweisung gleicher markanter Punkte auf dem Objekt zusammengefügt werden. Im darauffolgenden Schritt wurden die Teilobjekte als Ganzes registriert.

Unnötige Punkte außerhalb des Meshes werden mit dem Eraser-Tool entfernt. Mit dem Tool Align werden bei mehreren Meshes Punkte als Marker an einzelne Stellen platziert, die exakt so auch auf dem Gegenstück liegen müssen. Das Programm fügt diese Teile danach automatisch zusammen. Outlier-Removal ist ein Tool, das sich wie das Eraser-Tool, störende außenstehende Punkte zu entfernen und das Objekt förmiger zu machen.

Die Mesh Erstellung erfolgt in dem Tool Fusion. Hier werden die Registrierten Flächen des Objektes zu einer Oberfläche zusammengefügt. Für die meisten Objekte empfiehlt sich die Sharp Fusion. Sie berücksichtigt auch starke Kanten mehr und rundet diese nicht ab. Mit dem Mesh-Simplification-Tool können die Polygone reduziert werden. Der Hai wurde von ca. 60.000 Polygonen auf ca. 10.000 Polygone reduziert.

Diese Schritte können jederzeit reproduziert werden. Artec Studio Professional 10 enthält weitere Tools für die Reparatur von Mesh und Textur. Im letzten Schritt werden die Dateien als OBJ-Dateien exportiert. OBJ-Dateien können von den meisten 3D-Softwares verwendet werden und liefern auch die Texturinformationen mit.



(a) Mesh Ansicht mit Textur

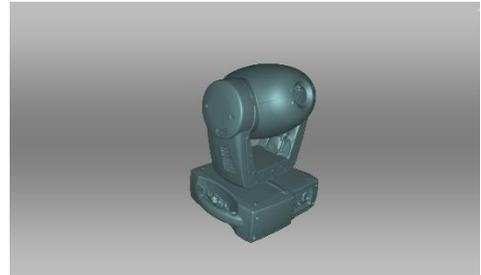


(b) Mesh Ansicht ohne Textur

Abbildung 3.2: Darstellung der Lampen als Mesh und mit Textur (eigene Darstellung)



(c) Mesh Ansicht mit Textur



(d) Mesh Ansicht ohne Textur

3.2.2 blender

Die OBJ-Dateien wurden für den Feinschliff in blender Version 2.83.1 importiert. Hier wurden die Meshes weiter herunterskaliert und Unebenheiten geglättet. Einige Konturen mussten noch gezogen werden, die in Artec Studio Professional 10 zu glatt verschmolzen wurden.

Der Bereich mit den Zähnen des Hais wurde abgeglättet. Hier waren zu viele Störelemente übriggeblieben, dass die Zähne eine pelzige Struktur aufwiesen. Auch bei den Leuchtelementen gab es einige Punkte, die zu eng aneinander und somit zu klein für die Strahlen des Lasers waren. Hier entstanden entweder Löcher oder zu unebene Oberflächen (siehe Abb. 3.3 und 3.4).

Als Lösung wurden identische Stellen kopiert und mit dem Mesh verschmolzen. Hier zeigt sich ein Nachteil bei gescannten Meshes die über eine Netz an Faces verfügen, im Vergleich zu modellierten Objekten, die ein perfekt aufgebautes Netz an Faces haben. Das Netz von Scans ist somit nicht so einfach zu entkoppeln von dem Rest des Modells, da ungerade Löcher und Risse dabei entstehen, die eine Reparatur sogar verschlimmern können.

Eine Lösung hierfür ist das Darüberlegen der kopierten oder neumodellierten Teile. Einzelne Reste darunter können im Anschluss gelöscht und unsichtbar gemacht werden. Die Einzelteile werden dann zu Child-Objekten des Meshes gemacht, damit sie fixiert bleiben.

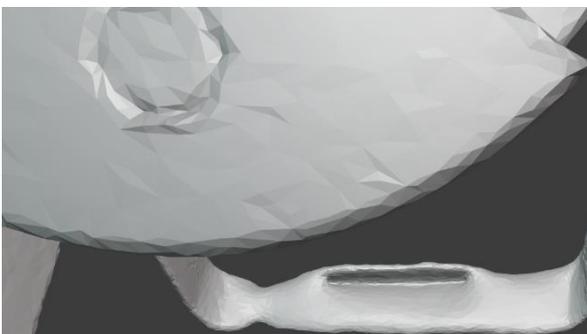


(a) Löcher im Boden der kleinen Lampa

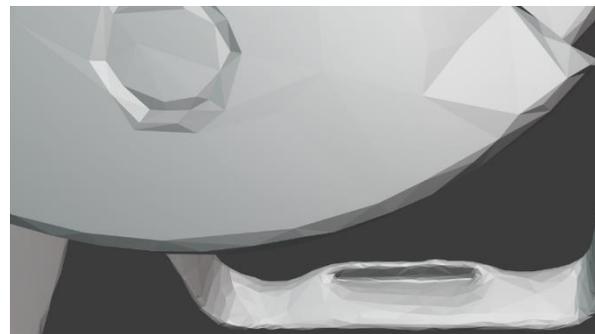


(b) Reparierte Löcher

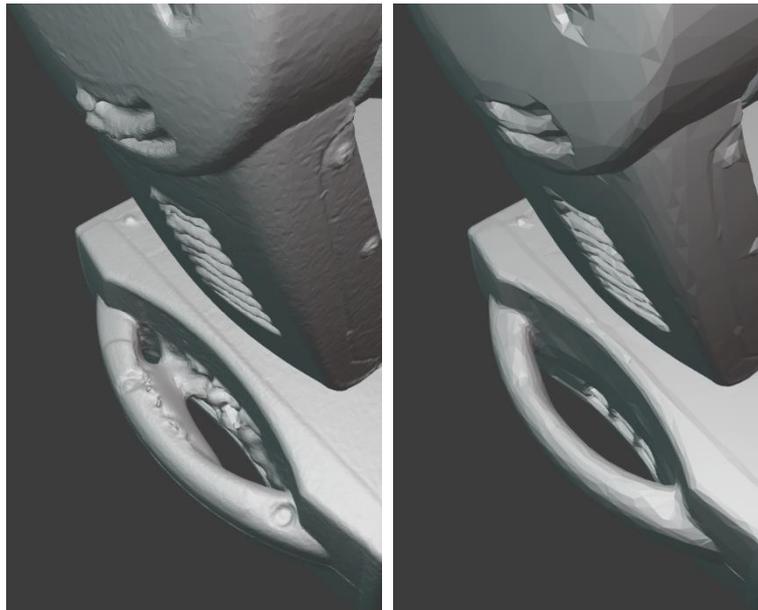
Abbildung 3.3: Mesh Fehler und Korrekturen (eigene Darstellung)



(c) Mesh-Fehler in der Topologie



(d) Korrigiert Stelle des Meshes



(a) Fehler der Fusion aus Artec

(b) Korrigiert und geglättet

Abbildung 3.4: Fusions Fehler aus Artec (eigene Darstellung)

Für die Erstellung der Texturen muss das Model erst im UV-Editor aufgeklappt (unwrapping) werden. Die Oberflächentextur wird aufgeklappt und auf ein 2D-Image gelegt. Wird ein Model im UV-Editor nicht unwrapped, lassen sich die Texturen auch nicht speichern und exportieren.

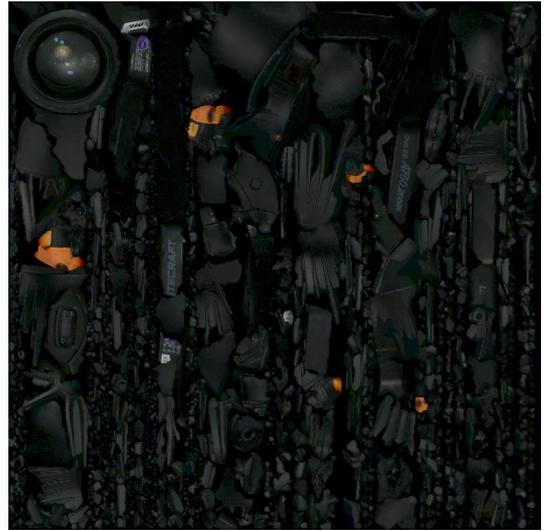
Fehler in der Textur wurden in Blender mit dem Repair-Brush Tool und dem Pinsel Tool bearbeitet. Der Repair-Brush kopiert die Informationen der umliegenden Textur und übermalt damit den ausgewählten Bereich. Die fertige Textur kann nun auf das Mesh gebacken (baking) werden und als PNG abgespeichert werden. Dies wird auch als sogenannte Normal-Map getan. Diese enthält die Höhen und Tiefeninformationen des Meshes und täuscht die gleiche Form der Oberflächenstruktur vor. Dies ist hilfreich für einen weiteren Schritt der Reduzierung der Polygone. Das reduzierte Duplikat dient ebenso als eine weitere Absicherung. Das zweite Mesh kann nun weiter in seiner Polygonanzahl mit dem Dezimierung Tool reduziert werden.

Die verschiedenen Maps der Texturen werden im 2D-Format gespeichert und können von dort direkt exportiert werden. Für die gescannten Objekte wurden Textur-Maps erstellt. Die eigentliche Textur mit Farbwerten. Eine Normal-Map mit den Höhen- und Tiefenwerten der Oberfläche. Diese simuliert die 3D-Struktur eines Objektes. Schatten- und Lichtwerte wurde auch gebaked aber herausgelassen, um eine gleichmäßig beleuchtete Textur zu erhalten.

Unity wird hierfür später auf seine eigene Render Pipeline zurückgreifen. Die Texturen wurden direkt in den Unity Asset Ordner exportiert. Unity aktualisiert so automatisch hinzugefügte Dateien und spart Zeit. In der folgenden Abbildung wird die generierten Maps beispielhaft aufgezeigt.



Texture-Map Lampe 1



Texture-Map Lampe 2

Abbildung 3.5: Texture-Maps der Lampen (eigene Darstellung)



Abbildung 2

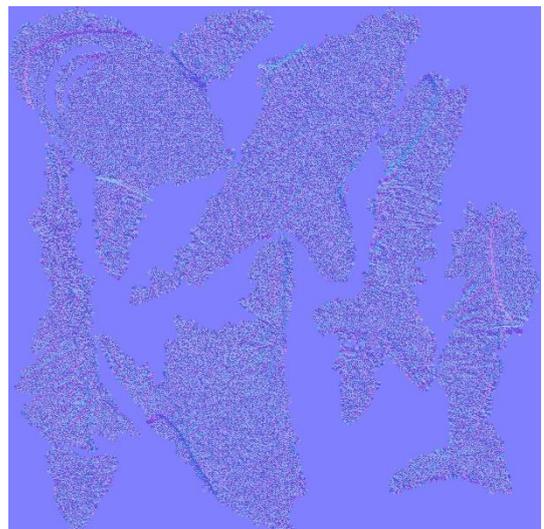


Abbildung 1

Abbildung 3.6: Texture- und Normal-Map des Hais (eigene Darstellung)

3.2.3 Photoshop

Die unwrappten Texture-Maps wurde zwischenzeitlich in Photoshop als PNG-Dateien importiert. Einige Texturen enthielten kleine Stellen mit vermischten RGB-Werten oder Texturfehlern, die in Blender nicht gut genug überarbeitet werden konnten. Dafür verfügt Photoshop über ein besseres Archiv an Tools und Werkzeugen für die Feinarbeit.

Zu unscharfe Elemente der Textur wurden aufgehellt und nachgezeichnet. Hierzu zählen beispielsweise Etiketten und Beschriftungen. Für den Arbeitsschritt wird ein Grafik-Tablet der Marke Medion benutzt. Texturen mit falschen Farbinformationen werden mit dem Repair-Tool and ihre Umgebungstextur angeglichen. Dafür werden Informationen aus der umliegenden Textur genutzt und auf die angewendete Stelle gelegt. Dies ist jedoch mit Vorsicht zu machen, da die Textur auch mit falschen Informationen repariert werden kann weiter man in eine bestimmte Richtung zeichnet, desto mehr Texturinformationen werden aus den umliegenden Bereichen gezogen. Das kann Texturinformationen beinhalten die unerwünscht sind, bzw. Bereichen entsprechen, die wiederum andere Farbbereiche betreffen.

Jedes Model hatte dabei unterschiedliche Texturbereiche, die überarbeitet werden mussten. So mussten einzelne Farbliche Bereiche der Textur repariert werden. Dazu gehörten dunkle Stellen des Bauches (siehe Abb. 3.7). Diese entstanden durch Texturinformationen, die durch die Beine eines Stuhles entstanden. Dieser wurde für eine gute Fixierung des Objektes genutzt, damit es sich einfacher von allen Seiten einscannen ließ. Einige Farbbereiche waren auch schwach im Kontrast oder verwaschen, wie Zähne und Augen.

Da der Scanner nicht jeden Bereich zu hundert Prozent erfasst, da oft Stellen versteckt liegen und zu klein sind, mischt der Algorithmus der Scanner Software umliegende Farbinformationen hinzu. Auch kann die eingebaute Leuchtquelle des Eva Handscanners nicht alle Bereiche maximal aufhellen. Zu guter Letzt wurden die Texturen als Kopie abgespeichert. Die 2D-Texture-Maps konnten so wieder in Blender oder direkt in Unity für die Erstellung der Materialien importiert werden.



(a) Original Textur des Scans



(b) Überarbeitete Textur aus Photoshop

Abbildung 3.7: Texture-Map vor (a) und nach (b) der Überarbeitung in Photoshop

3.3 Evaluation der Scans

Für die Erstellung der Scans wurden Punktwolken mit über einhunderttausend Polygonen bei einer Framerate von 16 Bit mit dem Eva Handheld-Scanner aufgenommen. Jeder zwanzigste Scan enthielt dabei Informationen für die Textur. Die Scan-Dateinsätze des FTZ DR betragen mehrere Gigabyte, die jedoch im Vorfeld für eine bessere Verarbeitung stark reduziert wurden. In dem folgenden Kapitel werden die Meshes und Texturen anhand ihrer Arbeitsschritte und Fehlerquellen verglichen.

3.3.1 Analyse der Meshes

Die gescannten Meshes des Eva Hand-Scanners wurden mit sehr hoher Genauigkeit automatisiert von der Artec Professional 10 Software generiert. Die Option Sharp-Fusion sorgt dabei für sehr glatte und markante Oberflächen der Leuchten, die sowohl Runde als auch viele eckige Bereiche im Original aufweisen. Die Form wurde sehr gut übernommen, jedoch musste die Größe der Objekte händisch in Blender angepasst werden, um näher am Original sein zu können.

Modelle müssen so in Blender nicht komplett neu modelliert werden, wodurch viel Zeit der 3D-Artists eingespart wird. Jedoch kam es auch bei einigen Scans zu Fehlern in den Meshes. So wurden einige Höhen und Tiefeninformationen zB. der Augen des Hais oder der Lampen nicht richtig übertragen und mussten in Blender mit dem Sculpting-Tool überarbeitet werden. Auch fehlende Informationen der stützenden Metallstriemen der Lampe kamen vor. So wurden Löcher oder zu kleine Partien an bestimmten Stellen der Meshes generiert. An diesen Stellen war der Bereich für die Abtastung des Lasers zu klein was auch durch fehlende Farbinformationen gestützt wird. Die meisten Fehlerkorrekturen ließen sich jedoch fast problemlos nachbearbeiten.

Dank der Informationen aus den Texture- und Normal-Maps können die Objekte auch bei reduzierter Polygonanzahl, fast identisch zum Original dargestellt werden. Die Darstellung des Hai Meshes ist unter allen Modellen am besten gelungen. Es ließen sich nach der Überarbeitung, kaum Fehlerstellen nachweisen, was auch an der einfachen Topologie lag.

Das Mesh wies keine markanten Ecken und Kanten auf und ließ sich somit optimal als Objekt mit einem weichen Material darstellen. Auch aus näherer Betrachtung fallen kaum Fehlerpunkte auf. Nur bei genauer Betrachtung sind einige Unebenheiten zu erkennen. In der folgenden Abbildung werden die einzelnen Schichten des Hai Meshes aufgezeigt.



(a) Drahtgitteransicht



(b) Mesh Ansicht



(c) Mesh Ansicht mit Textur

Abbildung 3.8: Die verschiedenen Schichten des Hais visualisiert (eigene Darstellung)

Die Modelle der Lampen wiesen im Vergleich zu dem Stoff-Hai jedoch mehrere Fehlerquellen auf (siehe Abb. 3.9), die selbst in der Nachbearbeitung nur zum Teil ausgeglichen werden konnten und bei genauerer Betrachtung auffallen. Jedoch war der größte Teil der Oberflächentopologie so gut intakt und geglättet, dass eine Neumodellierung auch hier nicht zwingend notwendig ist. Durchschnittlicherweise sehen die 3D-Objekte sehr gut aus, dass sie für die weitere Verwendung exportiert werden können.



(a) Unscharfe Textelemente



(b) Ansicht ohne Textur

Abbildung 3.9: Unscharfer Schriftzug mit Textur (a) und ohne Textur (b) (eigene Darstellung)



(a) Unscharfe Rückseite



(b) Unscharfe Rückseite ohne textur

Abbildung 3.10: Unscharfer Schriftzug mit Textur (a) und ohne Textur (b) (eigene Darstellung)

3.2 Analyse der Eva Scan Texturen

Die Texturen wurden sehr gut übertragen. Einige Stellen wiesen Fehlinformationen und Flecken auf, die durch die Fusionierung mehrerer Mesh-Einzelteile entstanden sind. Artec Studios hauseigene Software bietet dafür ein Textur-Repair Tool an. Dieses funktioniert im Grunde wie das Texture-Repair Tool in Blender. Die Anpassung der fehlerhaften Texturinformationen an die Umgebungsinformationen intakter Texturen konnte somit sehr gut visualisiert werden. Einige Texturinformationen enthalten dennoch bei genauer Ansicht Vermischungen aus Farben des RGB Bereichs. Diese fallen jedoch bei äußerer Betrachtung nicht auf.

In Blender wurden Fehler, die in Artec Studio nicht korrigiert werden konnten, zusätzlich angepasst. Dafür wurde das Model unwrapped (die Textur entpackt). Eine 2D-Ansicht konnte dadurch unterstützend genutzt werden, Texturen anzupassen als würde man sie in Photoshop exportieren und dort in der 2D-Ansicht anpassen.

Eine Methode bietet Blender jedoch im Texture-paint Fenster. So können Texturen mit dem Pinsel und Repair Tools direkt auf dem 3D-Model angewendet werden, um so eine genauere Anpassung vorzunehmen. Texturen, die nur in der 2D-Ansicht überarbeitet werden, können aufgrund der Topologie des Objektes abweichen. Beispiel hierfür sind die Etiketten des ASCII-Codes auf den jeweiligen Leuchten. Diese verloren an Konturen und wurden eher verwaschen durch die Überarbeitung in der 2D-Ansicht, der das aufliegende 3D-Model fehlt. Diese Anpassung wird direkt am Model durchgeführt und liefert eine besser aufgetragene Textur für das Baking (siehe Abb. 3.11).

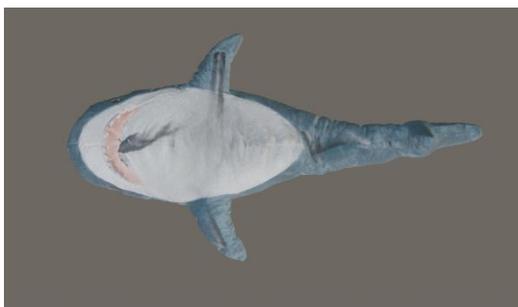


(a) Original Textur (Augen und Zahnpartien)

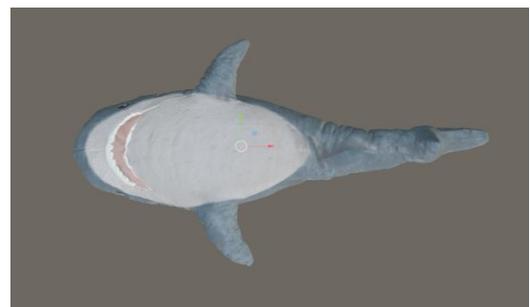


(b) Überarbeitete Textur

Abbildung 3.11: Texturfehler und bearbeitete Textur (eigene Darstellung in blender)



(a) Original Textur Unterseite



(b) Überarbeitete Textur

Die überarbeiteten Texturen des Hais ließen sich sehr gut im Nachhinein darstellen. Augen- und Zahnpartien waren sehr unscharf in Auflösung und Farbe, ließen sich jedoch sehr gut in Photoshop überarbeiten. Dafür wurden die originalen Farben mit der Pipette aufgenommen, aufgehellt und überzeichnet.

Da das stoffliche Material keinerlei Reflexionen enthält, wird der Metallic- und der Specular-Anteil auf null gesetzt. Das Material könnte realistischer wirken, wenn man in Blender Härchen hinzufügen würde, aus denen das Original besteht. Dies würde zu einem weitaus höherem Rechenaufwand führen.

Die Darstellung über eine Normal-Map (siehe Abb. 3.12) enthielt jedoch genug Tiefen- und Höheninformationen, um das Material im Kontext seiner Größe realistisch genug darzustellen. Bei genauerer Betrachtung fällt erst die tatsächliche Oberflächen Struktur auf und ist somit akzeptabel.

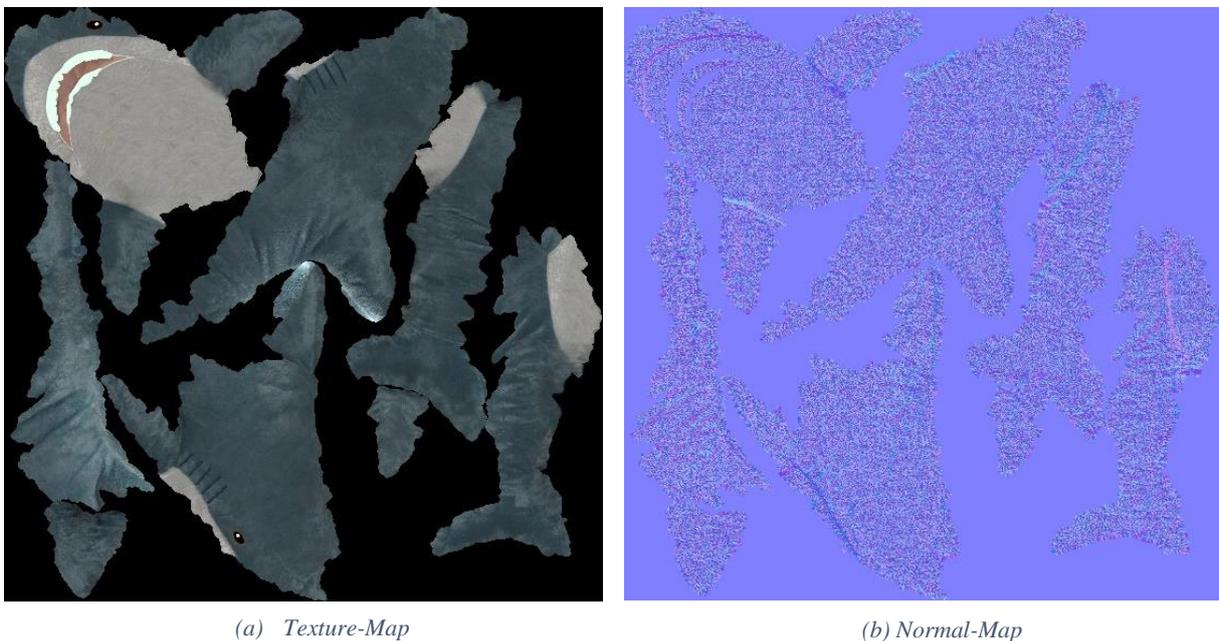


Abbildung 3.12: Darstellung in Schritten (eigene Darstellung)

Die Überarbeitung der Lampen erwies sich als etwas schwieriger. Die komplexe Oberfläche, bestehend aus vielen Ecken, Kanten und toten Winkeln, erforderte einen höheren Arbeitsaufwand, im Vergleich zu dem Hai. Texturen die Beschriftungen enthielten, könnten erst ab einer gewissen Größe rekonstruiert werden, in der sie vor der Überarbeitung gut erkennbar waren. Diese verwaschenen Schriftzeichen sind jedoch auch in VR nur bei sehr genauer Betrachtung zu erkennen und fallen bei der Größe des kleinen Objektes nicht weiter auf (siehe Abb. 3.13). Die Linsen der Leuchten konnten während des Scan Vorganges nicht erfasst werden. Diese waren zu transparent und reflektierten das Licht zu stark. Hier konnte durch eine Modellierung Abhilfe geschaffen werden. So wurden runde Linsen in Unity nachmodelliert. Das Material wurde in Unity über einen Glass-Shader visualisiert, der Reflexions- und Transparenzwerte enthält.



(a) Original mit fehlerhafter Textur der Linse



(b) Original mit unscharfer Beschriftung

Abbildung 3.13: Fehlerquellen in den Texturen der Lampen (eigene Darstellung)



(c) nachbearbeitete unscharfe Beschriftung

4 Simulation

Für die Simulation des FTZ wurde die Game Engine Unity ausgewählt. Als Render-Engine wurde die Universal Render Pipeline benutzt, da diese im Vergleich zu der Built-in-Render Pipeline eine der ausgereiftesten zur Verfügung stehenden Render Pipelines in Unity ist, die erst seit einigen Jahren in Unity verwendet werden. Diese verfügt über Echtzeitrendering. Als Zielhardware wurden die HTC Vive und die Oculus Quest gewählt. Dafür wurden auch die jeweiligen Assets für die Implementierung benötigt. SteamVR und das Oculus VR-Plugin.

Da nicht jede Asset Version in jeder Unity Version im Abgleich der Universal Render Pipeline funktioniert hat, wird die Unity 2019.4f1 Version aufgrund ihres Long-Term-Supports (Langzeitunterstützung der Version) ausgewählt. Da für die Materialauswahl das Asset VR Material Manager ausgewählt wurde und die URP (Universal Render Pipeline), sowie das VRTK (Virtual Reality Toolkit) SteamVR ab der Version 2.0 keine Implementierung und Support mehr erhalten, wird die SteamVR Version v1.2.3 ausgewählt. Diese unterstützt die Implementierung des VRTK Assets und somit den VR Material Manager.

4.1 Zielhardware

Als Zielhardware werden die HTC Vive und die Oculus Quest als HMDs für die Umsetzung verwendet. Im folgenden Abschnitt wird auf die technischen Eigenschaften der HMDs eingegangen.

4.1.1 HTC Vive

Die HTC Vive ist ein Head-Mounted Display von HTC das in Zusammenarbeit mit der Firma Valve produziert wird und das erste Mal 2015 auf dem Mobile World Congress präsentiert wurde.

Das HDM deckt einen Spielbereich von bis zu 15m² (3.5m x 3.5m) ab und hat ein Sichtfeld von 110 Grad. Neben den beiden Controllern werden Basis-Stationen für eine drahtlose Synchronisierung der Geräte benötigt. Das HDM hat zwei integrierte Linsen für die visuelle Darstellung. Eine Linse hat 1080 x 1200 Pixel (2160 x 1200 Pixel für beide Linsen). Die HTC Vive verfügt darüber hinaus über einen SteamVR Tracking Sensor, G-Sensor, Gyroskop, und einen Näherungssensor der einem das Blickfeld und die Drehung mit der Brille für ein immersives Erlebnis erlauben. Die HTC Vive lässt sich über eine Verbindung per Kabel mit jedem Rechner verbinden. Voraussetzung für die Nutzung ist SteamVR über Steam und die Registrierung der Hardware mit den zugehörigen Controllern.¹⁹



Abbildung 4.1: HTC Vive (Quelle: Vive; 2020)

¹⁹ Vive, 2016

4.1.2 Oculus Quest

Die Quest ist ein HMD von der Firma Oculus. Das HMD verfügt über zwei Linsen mit einer Auflösung von je 1440 x 1600 Pixel. Diese werden mit einer Aktualisierungsrate von 72 Herz betrieben. Das Tracking erfolgt über diverse Sensoren, die es dem HMD erlauben auch Bewegungen des Körpers im Spielbereich zu erfassen. Anders als bei der HTC Vive ist die Quest ein Android gestütztes System und funktioniert auch kabellos über den App-Store der mit 64 oder 128GB ausgestattet ist.²⁰



Abbildung 4.2: Oculus Quest (Quelle: Oculus, 2020)

²⁰ Oculus, 2019

4.2 FTZ Digital Reality Hafencity

Das Forschungs- und Transferzentrum des Studienganges Digital Reality befindet sich in der alten Speicherstadt der Hafencity Hamburg. Auf 600m² liegend bilden die zwei großen Räumlichkeiten Platz für Studierendenbereiche der Hochschule für angewandte Wissenschaften (HAW) und vier ansässige Unternehmensstellen. Dazu zählen zum Beispiel shape-shift und Macom. Zu den Bereichen gehören ein VR-Bereich, Arbeitsbereiche und Konferenzzimmer für Workshops und Seminare.

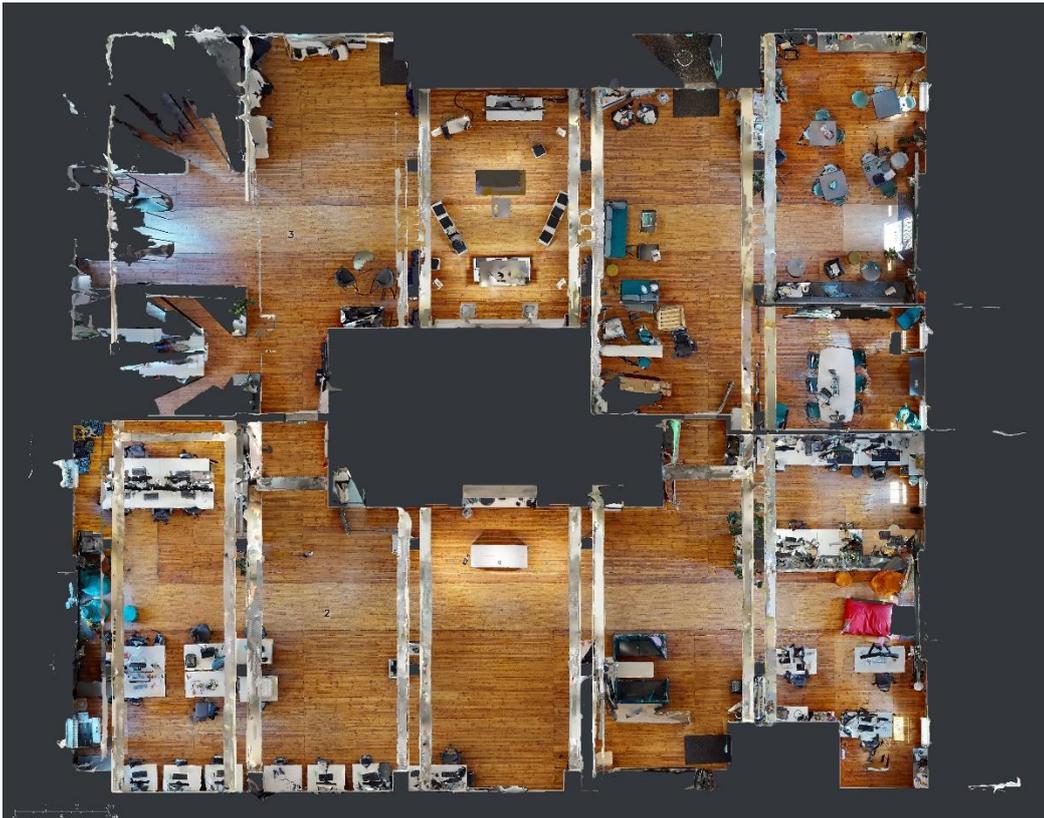


Abbildung 4.3: FTZ Point cloud Oben-Ansicht (Quelle: Dataport)

Dieser Bereich wurde von der Firma Dataport mit dem FARO S70 Scanner eingescannt und den Studierenden des Masterstudienganges Digital Reality als Projekt zur Generierung eines 3D-Modelles aller Bereiche übergeben. Inklusiv einer Variante, die zeigt wie die alten Räume vor dem Umbau ausgesehen haben. Die Studierenden haben mit den generierten Rohdaten der Punktwolken, eine fertige Architektur mit Meshes und Texturen in Revit und Blender erstellt. Die Möbel wurden neu modelliert.

4.3 Unity Engine

Als Entwicklungsumgebung für die VR-Anwendung wurde die Unity Engine benutzt. Gewählt wurde hierfür die Unity Version 2019.4.2f1 mit Long Time Support (LTS). Diese enthält vielerlei Möglichkeiten HDMs über Plugins der Hersteller oder externe Toolkits zu implementieren. Die Umsetzungsmöglichkeit reiner VR-Anwendungen auf Basis der Toolkits ist dabei sehr entwicklerfreundlich, muss jedoch im Zusammenhang der Kompatibilität verschiedener Versionen der einzeln genutzten Komponenten analysiert werden.

Die Unity Engine bietet für die Erstellung von Projekten direkt eine Kamera, eine Skybox und ein Directional Light als Lichtquelle. Diese Komponenten haben jedoch keinerlei Mehrwert für eine VR-Anwendung. Die Kamera wird in einer VR Umgebung durch das Kamera-Rig der jeweiligen Zielhardware ersetzt, da das Sichtfeld auf, das des HMDs geändert werden muss. Sonst sieht man durch die HMD nichts. Außerdem lassen sich verschiedenste Lichtquellen in der Unity Engine erstellen. Dazugehörige Parameter wie die Streuung und Lichtkegel können manuell angepasst werden.

Die Modelle lassen sich in Unity als FBX oder OBJ Dateien importieren. Da FBX Dateien aber direkt die Texturinformationen mitliefern, macht es durchaus Sinn das OBJ Format zu wählen. Dieses erlaubt es dank auch verschiedene Texturinformationen anhand von Maps in Unity zu importieren. Die Maps können dann in ein entsprechend erstelltes Material eingepflegt werden, um später direkt auf das Objekt gelegt werden zu können. Dafür wird für die einzelnen Maps das PNG Format gewählt. Die Darstellung erfolgt dabei in 1028 x 1028 Pixel (4k). Die Größe der Objekte weicht jedoch auch bei genauester Beachtung der Skalierung in der Unity Engine ab, wodurch diese nachskaliert werden müssen. Solche Unterschiede bestehen häufiger zwischen verschiedenen 3D-Visualisierungsprogrammen. Positionierung und Drehung stimmen jedoch meist überein.

Ein wichtiger Vorteil der Unity Engine ist auch der Asset Store. So können Privat- und Hobbyentwickler eigene erstellte Assets für die Allgemeinheit zugänglich machen. Dies kann in vielerlei Entwicklungsprozessen Zeit und Aufwand sparen, um sich auf bedeutendere Schwerpunkte konzentrieren zu können. Für das Projekt wurden Assets aus dem VR Bereich genutzt. Diese werden in dem Kapitel 4.5 vorgestellt.

4.4 Implementation der Zielhardware

Grundlage für die Funktionalität der Oculus Quest sind im Grunde zwei Pakete. Das eine ist Oculus Integration und Oculus Utilities. Oculus Integration und Oculus Utilities sind beides Pakete die Prefabs und Skripte für die Entwicklungsumgebung bereithalten. In der neuesten Version des Oculus Integration Paketes im Unity Store wurden alle Inhalte, für eine Handhabung zusammengefasst. Enthalten sind dabei das Framework, Prefabs, Beispielszenen und weitere Entwicklertools für die Entwicklung in Unity.

Hier ist anzumerken, dass für die Entwicklung mit der Oculus Quest auch eine erstellte ID und die Desktop App notwendig sind. Die Oculus Quest ist dank ihrer Android Umgebung im Stande auch ohne Verbindung mit einem Computer ausgeführt werden zu können. Jedoch für die Entwicklung und Nutzung am Computer ist ein USB-3 zu USB-C Kabel notwendig. Dies führt häufiger dazu, dass Geräte nicht erkannt und des Öfteren neu gekoppelt werden müssen.

Grundlage für die Verwendung der HTC Vive ist ein Steam Account und SteamVR. Unity braucht für die Integration das SteamVR Plugin aus dem Asset Store. Dieses beinhaltet und Prefabs mit Beispielszenen und bildet die Schnittstelle für die Anweisungen der API (application programming interface) über OpenGL. Die API beschreibt hierbei die Programmierschnittstelle.

Die HTC Vive ist eine rein Desktop orientierte Hardware. Die Schnittstelle wird hierfür über ein HDMI und ein USB-3 Kabel hergestellt. Das hat den Nachteil, dass diese nicht plattformunabhängig ist und ein Kabel als Schnittstelle unabdinglich ist. Dies schränkt auch den Bewegungs- und Sichtbereich ein.

4.5 Assets/Interaktionsmöglichkeiten

Für die Interaktion wurde das VRTK Asset gewählt. Dieses bietet Plattformübergreifend Interaktionsmöglichkeiten für diverse HMDs und dazugehöriger Controller. Für eine beispielhafte Interaktion wurde der VR Material Manager des Unity Asset Stores gewählt. Dieser nutzt das VRTK Asset als Schnittstelle für die Interaktion der HMDs und baut mit eigenen Skripten auf deren Interaktion auf. Für den Material Manager werden dabei ein Laserpointer und eine UI Interaktion benötigt, die über ein objektgebundenes Interface eine Auswahl von Materialien vorgibt. Dadurch lassen sich originale Scan Texturen in verschiedenen Materialien darstellen.

Basierend auf den Avataren der einzelnen HMDs lassen sich so über das VRTK beide Brillen ansteuern. Da VRTK mit der Version 3.3.0 endete und Version 4.0 derzeit in der Betaphase steht, wurde 3.30 als verwendete Version gewählt. Die Interaktion hervorragend jedoch hat sich seit dem Einzug von SRPs die Darstellung der Materialien geändert. Diese müssen in SRP Materialien umgewandelt werden, dass die SRP diese auch rendert. Alle Materialien ließen sich dabei über eine automatisierte Auswahl direkt in das URP Format übermitteln, in dem die Materialien als Lit ausgegeben wurden. Lit ist das Standardformat der URP.

Jedoch ergab sich hier folgende Problematik. Da beide Zielhardware Geräte Materialien der Controller und Hände in Pink dargestellt wurden. In Verbindung mit VRTK ist nicht fähig diese Materialien richtig über die SRP auszugeben. Die Lösung war relativ simpel bei der HTC Vive. So konnte im Skript das Material auf die „*Universal Render Pipeline Lit*“ geändert werden. Dadurch konnte das ursprüngliche Default Material der Controller wiederhergestellt werden.

Für die Oculus Quest hat sich das Ganze als etwas schwieriger herausgestellt. Da Oculus keinerlei Scripting Anweisung für die Materialien ausgibt, wurden die Standardcontroller den Händen als Child Objekt angefügt und das OVRAvatar Rig wurde deaktiviert. Dadurch konnten die Controller normal visualisiert dargestellt werden, jedoch ohne Animation der einzelnen Tasten und Knöpfe. Für die Interaktion funktionierte dies dennoch sehr gut.

Das Menü befindet sich jedes Mal seitlich neben dem jeweiligen Objekt, das das Material Manager Prefab angehängt bekommt. Dadurch lassen sich die Materialien über eine Listenauswahl ändern. Die Menüs können über den X-Button des Controllers ein- und ausgeblendet werden. Per Laserpointer werden diese ausgewählt und mit Drücken der hinteren Taste bestätigt.

Eine weitere Komponente ist der Teleporter für die Navigation durch das FTZ. Dieser wurde als VRTK auf den linken Controller gelegt, damit der rechte Controller für die Material und UI Interaktion genutzt werden kann.

4.6 Implementation der Modelle

Die in Blender überarbeiteten Modelle wurden als OBJ Dateien importiert. Es ist auch möglich diese als Blender File oder FBX File zu importieren jedoch fehlen hier die gebakten Informationen der verschiedenen Texturen. Für eine genaue Visualisierung der Materialien wurden die erstellten Texturen als Maps importiert. Die Textur Map gibt die Informationen der reinen Oberflächentextur. Die Normal Map gibt die Tiefeninformationen des Meshes wieder, sodass die Textur genauso aussieht, wie sie in Blender war. Die Materialien wurden aufgrund der URP in URP Lit Materialien überführt. Damit sich die Materialien ändern lassen, wurde für das Hai Modell eine weitere Textur erstellt, die als Emissionswert ausgegeben wird (siehe Abb. 4.4 und 4.5). Das führt dazu dass die Textur des Emissionswertes immer ausgegeben wird, wenn sich die Farbe des Materials ändert (siehe Abb. 4.6 bis 4.8). Die verschiedenfarbigen Materialien werden in den darauffolgenden Abbildungen beispielhaft aufgezeigt.



Abbildung 4.4 Textur des Materials

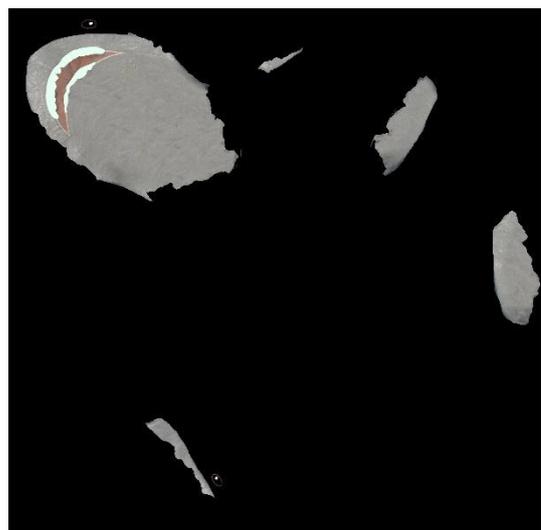


Abbildung 4.5 Emissionswert des Materials



Abbildung: 4.6 Hai mit originaler Textur



Abbildung: 4.7 Hai mit roter Textur



Abbildung: 4.8 Hai mit grüner Textur

4.7 Umgebung

Für die Umgebung wurde das FTZ DR in der Hafencity gewählt. Da die Abgabe der Studierenden des Masterstudienganges etwas kurzfristig war wurde ein modelliertes 3D-Objekt von Anna Wolf genommen. Für die Ausstattung wurden bereits fertig modellierte Möbel der Digital Reality Studenten und der Besprechungsraum des FTZ genutzt. Dazu zählen Stühle und Möbel, die häufig vorkommen und somit einmal modelliert werden mussten. Alle Objekte des FTZ zu modellieren wäre ein sehr großer Aufwand. Die Möbel wurden nach Gefühl und Augenmaß platziert, um den Maßen des FTZ Modells zu entsprechen (siehe folgende Abbildungen).



Abbildung: 4.9: Visualisierung des FTZ Modells in Unity (eigene Darstellung)



Abbildung: 4.10: Besprechungsraum (eigene Darstellung)

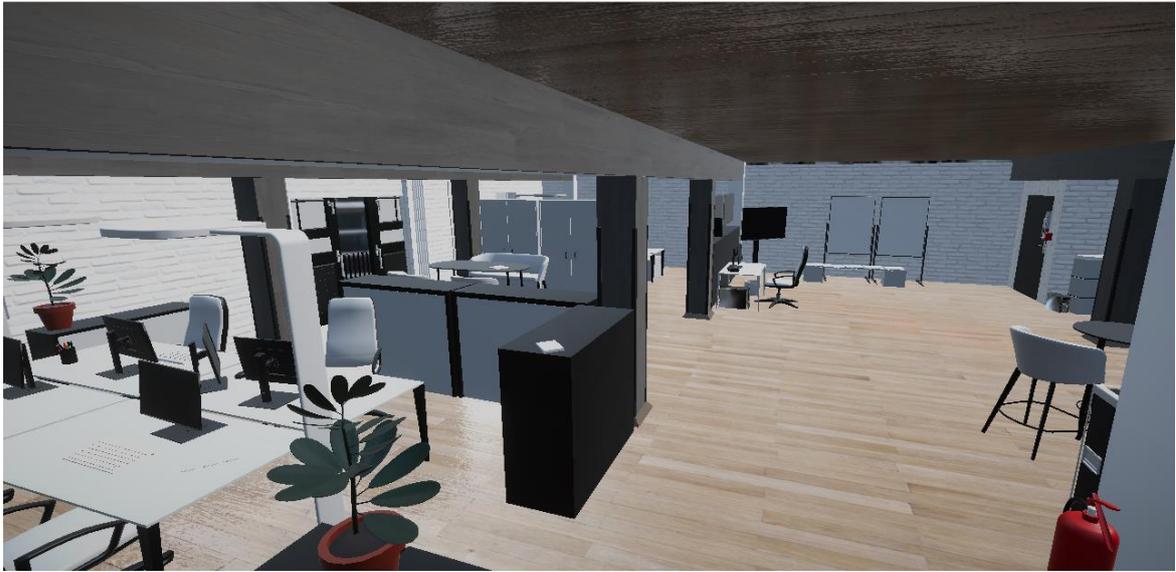


Abbildung: 4.11: Vorderer Arbeitsbereich (eigene Darstellung)



Abbildung: 4.12.: Hintere Küchenzeile (eigene Darstellung)



Abbildung: 4.13.: Seminarraum (eigene Darstellung)



Abbildung: 4.14 vorderer Arbeitsbereich Beleuchtung (eigene Darstellung)

4.8 Beleuchtung

Eine realistische Beleuchtung sorgt erst für die richtige Wahrnehmung eines Objektes. Dies ist so in der Realität, wie auch in der Virtuellen Welt. Die Modelle der gescannten Lampen wurden mit Lichtquellen eines Directional Lights ausgestattet. Dieses hat einen geradeausgerichteten Leuchtkegel, mit anpassbaren Parametern für Intensität, Breite des Kegels und Farbbereich. Einzelne Deckenbeleuchtungen wurden mit Point Lights (siehe Abb. 4.11 und 4.12) versehen. Den Lampen wurden Directional Lights zugewiesen, da diese eine intensivere Lichtquelle besitzen.



Abbildung: 4.15 Beleuchtungsquellen im Besprechungsraum



Abbildung: 4.16 Beleuchtungsquellen der vorderen Küche

5 Fazit

Das Ziel dieser Arbeit war die Untersuchung der Visualisierungsmöglichkeiten von Materialien in einer VR-Anwendung. Die Objekte wurden samt ihrer Textur gescannt und bearbeitet. Die Simulation in der Unity Engine war dabei ein voller Erfolg. Die Implementation von 3D-Objekten und Materialien funktioniert ohne komplizierte Problematik. Die gescannten Texturen konnten sehr gut als Materialien in Unity reproduziert werden. Die einzige Anpassung, die vorgenommen werden musste, war die Umwandlung aller Materialien in kompatible Lit-Formate der Universal Render Pipeline, damit diese gerendert werden.

Hier ergab sich eine Schwierigkeit in der Darstellung der Controller. Die zuständigen Materialien ließen sich nicht so einfach in das URP zuständige Lit-Format übertragen. Für die HTC Vive, musste eine Code-Zeile von Standard-Material auf Lit geändert werden.

Die größere Herausforderung war jedoch die Visualisierung der Controller der Oculus Quest. Da diese über einen Shader Graph funktionieren konnten die Materialien nicht im Code angepasst werden. Nach vielen Experimenten fand sich jedoch eine Lösung. Der Oculus Avatar wurde in der Szene deaktiviert und Prefabs den Händen als Child übergeben. So wurden die Controller letztendlich auch richtig in ihrem ursprünglichen Material visualisiert, jedoch mit dem Nachteil des Fehlens der Animation, die sich nur über das Avatar-Rig animieren lassen.

Somit hat sich die HTC Vive über SteamVR als die bessere Entwicklungshardware bewiesen. Sie ist einfacher zu nutzen und auch nur mit Computern kompatibel, im Gegensatz zu der Oculus Quest die ein Android gestütztes System ist.

Abgesehen davon muss bei der Verwendung in der Unity Engine jedes Mal die Version dieser und der Assets abgeglichen werden. So funktionieren Skripte und Schnittstellen nicht mehr auf die gleiche Weise wie in anderen Versionen. So wurden SRPs erst mit der Unity Version 2018 eingeführt. Auch das VRTK funktioniert nur bedingt in den neuesten Unity Versionen. Dieser Abgleich muss leider jedes Mal vorgenommen werden und macht eine Zusammenarbeit verschiedenen Komponenten der Unity Engine schwierig, bzw. erfordert einen hohen Aufwand die Kompatibilität einzelner Komponenten im Vorfeld festzulegen.

Damit nicht zu viel Rechenleistung benötigt wird, muss darauf geachtet werden, dass die Meshes so weit wie möglich herunterskaliert werden, ohne das große Qualitätseinbußen bei der Visualisierung der Materialien entstehen. Die Erstellung der Point Clouds hat sich als schwieriger herausgestellt als gedacht. Der Fortschritt muss permanent bei der Aufnahme verfolgt und neu kalibriert werden, da der Eva Hand-Scanner häufiger das Tracking der Oberfläche verliert. Dafür ist eine genaue Haltung bei einem gleichbleibenden Winkel erforderlich und somit auch Erfahrung und Gefühl in die Handhabung des Scanners.

6 Aussichten

Die Ergebnisse zeigen, dass die Kombination Photogrammetrie und Laserscanning sich zum Erstellen von 3D-Objekte von VR-Anwendungen lohnen. Jedoch muss abgewogen werden, ob es sich nicht eher lohnt, Objekte mit einfacher Geometrie nicht selbst zu modellieren. Die einzigartigen Texturinformationen können jedoch nur durch den Photogrammetrischen Anteil erfüllt werden.

Abstriche müssen jedoch in der physikalischen Exaktheit gemacht werden. Trotz guter Skalierung im cm Bereich in Blender, werden diese nicht exakt so in Unity wiedergegeben. Dies muss per Hand und Augenmaß in Unity nachskaliert werden. Aus den Texturen ließen sich jedoch sehr gut Materialien rekonstruieren.

Das im Rahmen dieser Bachelorthesis entwickelte Projekt dient gut für die Vorstellung des FTZ Digital Reality. Die Fülle an Objekte zeigt in Kombination mit dem Material Manager ein gutes Beispiel wie Materialien visualisiert und auch verändert werden können. Diese Arbeit bietet mit der Fülle an Objekte und Informationen viele Möglichkeiten für die Weiterentwicklung von Interaktionen und der Nutzung der Erstellten Modelle für weitere VR-Anwendungen.

Des Weiteren wird dieses Projekt in die Hände eines Studierenden übergeben. Dieser wird sich in seiner Thesis mit den Größeneinheiten beschäftigen und wie gut sich Maße anhand der Realität in VR anpassen lassen, um ein besseres immersives Gefühl zu ermöglichen.

Abbildungsverzeichnis

2.1 Funktionsprinzip FOV	8
2.2 Lochbildkamera Funktionsprinzip.....	9
2.3 Funktionsprinzip der zentralprojektiven Abbildung.....	9
2.4 Funktionsprinzip des Impulszeitlaufverfahrens.....	12
2.5 Funktionsprinzip des Phasenvergleichverfahrens	13
2.6 FARO Focus ^s 70	1
2.7 Geoslam ZAB-Revo	18
2.8 Artec Eva	19
3.1 Darstellung in Schritten	21
3.2 Darstellung der Lampen als Mesh und mit Textur	22
3.3 Mesh Fehler und Korrekturen.....	23
3.4 Mesh Fehler und Korrekturen.....	24
3.5 Textur-Maps	25
3.6 Texture- und Normal-Maps	26
3.7 Texture-Maps vor und nach der Überarbeitung in Photoshop	28
3.8 Die verschiedenen Schichten des Hais visualisiert.....	28
3.9 Unschärfe Schriftzug mit Textur (a) und ohne Textur (b).....	29
3.10 Unschärfer Schriftzug mit Textur (a) und ohne Textur (b)	29
3.11 Texturfehler und bearbeitete Textur	30
3.12 Darstellung in Schritten	31
3.13 Fehlerquellen in den Texturen der Lampen.....	32
4.1 HTC Vive	29
4.2 Oculus Quest	30
4.3 FTZ Point cloud Obenansicht.....	31
4.4 Textur des Materials	32
4.5 Emissionswert des Materials	29
4.6 Hai mit originaler Textur	30
4.7 Hai mit roter Textur.....	31
4.8 Hai mit grüner Textur	32
4.9 Visualisierung des FTZ Models in Unity	29
4.10 Besprechungsraum	30
4.11 Vorderer Arbeitsbereich Beleuchtung	31
4.12 Hintere Küchenzeile	32
4.13 Seminarraum.....	29
4.14 Vorderer Arbeitsbereich	30
4.15 Beleuchtungsquellen im Besprechungsraum.....	31
4.16 Beleuchtungsquellen der vorderen Küche.....	32

Tabellenverzeichnis

2.1 Unterschiede analytischer und digitaler Photogrammetrie.....	11
--	--------------------

Literaturverzeichnis

- Artec3D. (2012). *Artec Eva 3D-Scanner*. Artec 3D. URL: <https://www.artec3d.com/de/portable-3d-scanners/artec-eva-v2>
- Artec3D. (2012). *Artec Laserscanner Eva technisches Datenblatt*. Artec 3D.
- Babov, D. (2017). *VR Material Manager*. Unity Asset Store.
URL: <https://assetstore.unity.com/packages/tools/gui/vr-material-manager-92908>
- Bitenc, M. (2019). *Three-Dimensional UAV-Based Photogrammetric Structural Models for Rock Slope Engineering*. researchgate. URL: https://www.researchgate.net/figure/3D-reconstruction-from-multiple-images-wherein-point-coordinates-are-calculated-by-the_fig1_327234684
- Dörner, R., Broll, W., Grimm, P., & Jung, B. (2017). *Virtual Reality (VR/AR) Grundlagen und Methoden der Virtuellen und Augementierten Realität 1. Auflage*. Springer.
- DVW - Gesellschaft für Geodäsie, G. u. (2017). *Terrestrisches Laserscanning 2017 (TLS 2017) Band 88*. Augsburg: Wißner-Verlag.
- FARO. (2016). *FARO Focus^s 70 Laserscanner technisches Datenblatt*. FARO.
- FARO. (2020). *FARO Focus 3D Laser Scanner Serie*. FARO. URL: <https://www.faro.com/de-de/produkte/bausektor-bim-cim/faro-focus/>
- Geoslam. (o.J). *ZEB REVO*. Geoslam. URL: <https://geoslam.com/solutions/zeb-revo/>
- Geoslam. (2020). *ZEB REVO Datenblatt*. Geoslam.
URL: <https://geoslam.com/wp-content/uploads/2020/08/ZEB-Revo-RT-product-card-1.pdf>
- Heipke, C., Freeden, W., & Rummel, R. (2017). *Photogrammetrie und Fernerkundung, Handbuch der Geodäsie 1. Auflage*. Springer.
- Laserscanning-europe. (2020). *Artec Eva*. Laserscanning-europe. URL: <https://www.laserscanning-europe.com/de/3d-scanner/model/3d-scanner-artec-3d-eva>
- Laserscanning-europe. (2020). *ZEB-REVO*. Laserscanning-europe. URL: <https://www.laserscanning-europe.com/de/3d-scanner/model/mobile-vermessung-mit-dem-zeb-revo-von-geoslam>
- Luhmann, T. (2018). *Nahbereichsphotogrammetrie Grundlagen - Methoden - Beispiele*. Wichmann.
- Mason, A. (2015). *The Haskins Society - Making 3D Models with Photogrammetry - Getting started with Agisoft Photoscan*. URL: <https://thehaskinssociety.wildapricot.org/photogrammetry>
- Möllring, L. (2018). *Impulsverfahren/Impulslaufzeitverfahren*.
URL: <https://messpanda.de/strecken/impulsverfahren/>
- Möllring, L. (2018). *Phasenvergleichsverfahren*.
URL: <https://messpanda.de/strecken/phasenvergleichsverfahren/> abgerufen
- Oculus. (2020). *Integration*. URL: <https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022>
- Otterstein, B. (2020). *iPad Pro: Das kann der neue LiDAR-Sensor*.
URL: <https://www.maclife.de/news/ipad-pro-kann-neue-lidar-sensor-100116388.html>
- Rechenbach, J. *Funktionsweise Laserscanner*. Hannover: rechenbach architecture. URL: <https://www.laser-scanning-architecture.com/funktionsweise-laserscanner/>

blender Dokumentation. 2020 URL: <https://docs.blender.org/manual/en/latest/render/cycles/index.html>
abgerufen

Baking: <https://docs.blender.org/manual/en/latest/render/cycles/baking.html>

Color Management: https://docs.blender.org/manual/en/latest/render/color_management.html

Settings: https://docs.blender.org/manual/en/latest/render/cycles/material_settings.html

Simplify: https://docs.blender.org/manual/en/latest/render/cycles/render_settings/simplify.html

Subdivision:

https://docs.blender.org/manual/en/latest/render/cycles/render_settings/subdivision.html

workbench. URL: <https://docs.blender.org/manual/en/latest/render/workbench/index.html>

Thomas, L., & Schumacher, C. (2016). *Photogrammetrie - Laserscanning - Optische 3-D messtechnik: Beiträge der Oldenburger 3D-Tage 2016*. Wichmann.

Valve. (2018). *SteamVR Unity Plugin Version 1.2.3*.

URL: https://github.com/ValveSoftware/steamvr_unity_plugin/releases/tag/1.2.3

Valve. (2020). *SteamVR Unity Plugin*.

URL: https://valvesoftware.github.io/steamvr_unity_plugin/articles/intro.html

Verteidigung, B. d. (2015). *Photogrammetrie: Luftbildorientierung und technische Grundlagen der Luftbildstereoauswertung*. Bundesministerium der Verteidigung. URL: https://www.bfr-kmr.de/anhang_2.3.2.html

VRTK. (2019). *VRTK Version 3.3.0 Download*.

URL: <https://github.com/ExtendRealityLtd/VRTK/releases/tag/3.3.0>

VRTK. (2020). *VRTK Version 3.3.0 Dokumentation*. URL: <https://vrtoolkit.readme.io/docs>

Taylor-Smith, Kerry. (2019) *What is a LiDAR Senso?*

URL: <https://www.azosensors.com/article.aspx?ArticleID=1110>

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelor-Thesis mit dem Titel:

selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen wie z. B. Internetseiten übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

Ort, Datum

Peter Oetker