



Vergleich der Methoden zur Erweiterung von 360° Panoramatauren durch die Verwendung von Tiefeninformationen in krpano

Studiengang Medieninformatik

Bachelorarbeit

vorgelegt von

Marc Henrik Plüddemann

geb. in Gießen

durchgeführt bei
mi360 GmbH, Lich

Referent der Arbeit: Prof. Dr. Cornelius Malerczyk
Korreferent der Arbeit: M. Sc. Hans Christian Arlt
Betreuer: Lorenz Rings

Friedberg, 2022

Danksagung

Ich möchte meiner Familie für die Unterstützung während der Abfassung der Arbeit danken. Meiner Mutter für die seelische und moralische Aufmunterung, und meinem Vater für das Korrekturlesen.

Selbstständigkeitserklärung

Ich erkläre, dass ich die eingereichte Bachelorarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die von mir angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Gießen, Juni 2022

Marc Henrik Plüddemann

Inhaltsverzeichnis

Danksagung	i
Selbstständigkeitserklärung	iii
Inhaltsverzeichnis	v
Abbildungsverzeichnis	vii
Tabellenverzeichnis	viii
1 Einleitung	1
1.1 Motivation	1
1.2 Problemstellung	3
1.3 Zielsetzung der Arbeit	4
1.4 Aufbau der Arbeit	4
1.5 Zusammenfassung	5
2 Grundlagen	7
2.1 Digitale Bilddaten	7
2.2 Panoramabilder	8
2.2.1 Panoramafotografie	8
2.2.2 Panorama-Rendering	10
2.2.3 Cube-Maps	10
2.3 Tiefeninformationen	10
2.3.1 3D-Modelle	11
2.3.2 Z-Depth-Bilder	12
2.4 Panoramatouren mit krpano	13
2.4.1 Panoramatouren	14
2.4.2 Übergang zwischen Panoramabildern	14
2.5 krpano	15
2.5.1 Panoramabilder in krpano	16
2.5.2 Hotspots in krpano	16
2.5.3 Depthmap-Implementationen in krpano	17
3D-Modell-Depthmaps	17

	Depthmap-Bilder	18
2.5.4	3D-Übergangseffekt in krpano	19
2.5.5	Dollhouses	21
3	Stand der Technik und Wissenschaft	23
3.1	Erweiterung von Panoramabildern durch 3D-Laserscan-Daten	23
3.2	Rekonstruktion von 3D-Geometrie aus Bildern	26
3.3	Räumlicher 3D-Übergangseffekt als Video	28
3.4	Zusammenfassung	28
4	Methodik	29
4.1	Aufstellung von Vergleichskriterien	29
4.2	Konzeption von Testumgebungen	31
4.3	Umsetzung	33
4.3.1	Erstellung der Testumgebungen	33
	Rendern von Panoramabildern in Unreal Engine 4	34
	Rendern von 32-Bit Z-Buffer-Bildern	34
	Zusammenfügen von Cube-Face-Bildern in PTGui	37
	Konvertieren von Z-Depth Panoramabildern mit ImageMagick	37
4.3.2	Testumgebung: Performance	38
4.3.3	Testumgebung: Präzision	40
4.3.4	Testumgebung: Ästhetik	40
4.3.5	Szenarien mit gerenderten Panoramabildern	41
4.3.6	Szenarien mit fotografierten Panoramabildern	42
5	Datenanalyse	45
5.1	Vergleich Speicherbedarf und Ladezeiten	45
5.2	Bildfrequenz	48
5.3	Präzision	52
5.4	Ästhetik	57
5.4.1	Vergleich bei gerenderten Panoramabildern	57
5.4.2	Vergleich bei fotografierten Panoramabildern	60
6	Ergebnisse	63
6.1	Zusätzliche Erkenntnisse	63
6.1.1	Arbeitsaufwand	63
6.1.2	Integration von Hotspots	64
6.2	Übersicht der Vor- und Nachteile	65
6.3	Einordnung der Ergebnisse	67
7	Zusammenfassung und Ausblick	69
7.1	Zusammenfassung	69
7.2	Ausblick	71
	Literaturverzeichnis	73

Abbildungsverzeichnis

1.1	Milgram's Realitäts-Virtualitäts-Kontinuum	1
1.2	Google Street View	2
1.3	Dollhouse Ansicht	3
2.1	RGB-Farbraum Modellierung	8
2.2	Vergleich Rektangularprojektion	9
2.3	Cube-Map Beispiel	11
2.4	sphärische Projektion auf 3D-Modell	12
2.5	Z-Bildebene	12
2.6	z-Depth Beispiel	13
2.7	Panorama-Übergangs-Interpolationen	15
2.8	Panorama-Übergangs-Interpolationen Beispiel	15
2.9	krpano Hotspots	16
2.10	3D-Axen Vergleich	17
2.11	Depthmap-Kanäle Vergleich	18
2.12	Dollhouse Beispiel	21
3.1	Street View Lidar Punktwolke	24
3.2	Panorama-Laserscanner Produkte	25
3.3	Matterport 3D-Model	25
3.4	DeepPanoContext: 3D-Rekonstruktion	26
3.5	Matterport Capture Screenshot	27
4.1	Sechs-Seiten Kamera Setup in Unreal	34
4.2	Blueprint Z-Depth gesamt	35
4.3	Blueprint Z-Depth links	36
4.4	Blueprint Z-Depth rechts	36
4.5	Testumgebung Präzision	40
4.6	Komplexe 3D-Szene	41
4.7	Einfache 3D-Szene	41
4.8	Indoor-Szene 3D-Rekonstruktion	42
4.9	Outdoor-Szene 3D-Rekonstruktion	43
5.1	Diagramm Performance Depthmap-Bilder	47
5.2	Diagramm Performance Depthmap-3D-Modelle	51

5.3	Vergleich Projektionsgenauigkeit 3D-Modelle	53
5.4	Vergleich Projektionsgenauigkeit Depthmap-Bilder	54
5.5	Vergleich Präzision Geometrie 3D-Modelle	55
5.6	Vergleich Präzision Geometrie Depthmap-Bilder	56
5.7	Vergleich Ästhetik einer komplexen 3D-Szene	58
5.8	Vergleich Ästhetik bei Weglassung von Objekten	59
5.9	Ästhetik Depthmaps Innenaufnahme	60
5.10	Ästhetik Depthmaps Außenaufnahme	61

Tabellenverzeichnis

5.1	Dateigrößen von Depthmaps (3D-Modelle)	46
5.2	Dateigrößen von Depthmaps (Bilder) einer einfachen Szene	46
5.3	Dateigrößen von Depthmaps (Bilder) einer komplexen Szene	46
5.4	Downloadzeiten (t) von Depthmaps (3D-Modelle) bei verschiedenen Internetgeschwindigkeiten	48
5.5	(Einfache Szene) Downloadzeiten von Depthmap-Bildern bei verschiedenen Internetgeschwindigkeiten	49
5.6	(Komplexe Szene) Downloadzeiten von Depthmap-Bildern bei verschiedenen Internetgeschwindigkeiten	49
5.7	Bildfrequenzmessung von Depthmaps verschiedener Komplexität bei drei Geräten	50
5.8	Durchschnittlich Bildfrequenz bei der Verwendung von Depthmap-Bildern mit verschiedenen Geräten.	52

Kapitel 1

Einleitung

Mixed-Reality-Anwendungen stellen eine Verbindung zwischen Realität und Virtualität her. Bezogen auf das Realitäts-Virtualitäts-Kontinuum von Milgram et al. [MTUK95] lässt sich das Thema dieser Arbeit im Gebiet der Augmented Reality (AR) verorten. In der AR wird die Realität durch virtuelle Objekte erweitert, weshalb sie sich im Spektrum des Kontinuums in der Nähe der realen Umgebung befindet (vgl. Abbildung 1.1).

Diese Arbeit befasst sich mit einem Sonderfall der AR, da die virtuellen Objekte nicht separat in eine reale Umgebung eingebaut werden, sondern die reale Umgebung (Panoramataour) wird selbst virtuell durch das Einbeziehen von Tiefeninformationen erweitert.

1.1 Motivation

Insbesondere der Online-Dienst *Google Street View*¹ zeigt die Möglichkeiten von virtuellen Panoramatauren auf (siehe Abbildung 1.2). Die Anwender können sich in den 360°-Panoramabildern frei umschaun und in nahegelegene Bilder springen. Zudem gibt es Interaktionsmöglichkeiten, wie z.B. Schaltflächen mit Informationen zu einzelnen Geschäften, Restaurants etc. Die Interaktivität in Verbindung mit hochauflösenden Panoramabildern ermöglicht ein virtuelles Eintauchen und realitätsnahes Erleben (Immersion) in den jeweili-

¹<https://www.google.de/streetview/>



Abbildung 1.1: Milgram's Realitäts-Virtualitäts-Kontinuum [MTUK95]

1. EINLEITUNG



Abbildung 1.2: *Google Street View* ermöglicht das Erkunden von Straßen auf dem Großteil der Erde.

gen, als drei-dimensional wahrgenommenen Ort, der vom Anwender auf spielerische Weise erkundet werden kann.

Diese Immersion kann jedoch beim Übergang zwischen zwei Panoramabildern gebrochen werden, da der Anwender das Gefühl bekommt an den nächsten Ort zu springen, ohne dabei eine räumliche Bewegung zu vernehmen. Zur Lösung dieses Problems verwendet Street View beim Fotografieren der Straßen zusätzlich Lidar-Scanner [Ars22], mit welchen zusätzliche Tiefeninformationen erfasst werden. Aus diesen Daten wird eine 3D-Geometrie erstellt, welche es ermöglicht die Panoramabilder beim Übergang zwischen zwei Bildern auf die Geometrie zu projizieren (mapping), sodass kurzzeitig ein drei-dimensionaler Bewegungseffekt entsteht. Die aus den Tiefeninformationen gewonnenen Daten sollen im Folgenden durch den Begriff *Depthmaps* referenziert werden, wobei die getrennte Schreibweise *Depth-Maps* (mit oder ohne Bindestrich) eigentlich die geläufigere Schreibweise ist, jedoch normalerweise für die Beschreibung des *Depth-Buffers* (also von Tiefeninformationen als Bild) verwendet wird. Die Schreibweise *Depthmap* bezieht sich hierbei explizit auf die im KRPAÑO-Kontext verwendeten Tiefeninformationen, die sowohl als 3D-Modell, als auch als Bild vorliegen können.

Das von STREET VIEW verwendete Vorgehen ist weitgehend automatisiert, es gibt jedoch auch die Möglichkeit *Depthmaps* manuell mit einer 3D-Software zu erstellen. Dazu muss die Geometrie eines Raumes ausgehend von einem Panoramabild händisch nachmodelliert werden, wobei dafür weniger-komplexe Szenen mit einfachen geometrischen Formen besser geeignet sind.

Die Software *krpano*² ermöglicht das Erstellen von webbasierten, virtuellen Panoramatauren und unterstützt dabei das Einbinden von *Depthmaps* zur Darstellung von 3D-Übergangseffekten und sogenannten „Dollhouse“-Ansichten (Außensicht der gesamten Panoramataur als 3D-Modell, vgl. Abbildung 1.3). Auf diese Weise können immersive Panoramatauren mit 3D-Effekten auch ohne teures Scan-Equipment hergestellt und veröffentlicht

²<https://krpano.com> (In dieser Arbeit wurde die Version 1.20.11 verwendet)



Abbildung 1.3: Beispiel einer Dollhouse-Ansicht in einer Demo von Everpano³

werden.

Das Erstellen von, durch Depthmaps erweiterten, virtuellen Panoramatauren zur Darstellung in einem Browser, ist mit einigen technischen Herausforderungen verbunden. Diese Herausforderungen fallen in die Bereiche Fotografie, 3D-Modellierung/Texturierung und Webprogrammierung. Während die Herausforderungen bei der Fotografie und im Umgang mit 3D-Software eher handwerklicher Natur sind, beinhaltet die Webumgebung einige technische Limitierungen, die es auch im Umgang mit Depthmaps zu beachten gilt. Besonders die sehr unterschiedlichen Leistungsfähigkeiten der Endgeräte⁴ stellen bei der rechenintensiven Darstellung von hochauflösenden Panoramabildern ein Problem dar. Durch die Verwendung von Depthmaps wird dieser Rechenaufwand nochmals erhöht. Ein weiterer zu beachtender Aspekt ist zudem der Speicheraufwand, der bei Webanwendungen wichtig ist, da die Daten erst von einem Server heruntergeladen werden müssen.

1.2 Problemstellung

Zur Erweiterung einer Panoramatur durch Tiefeninformationen mithilfe von Depthmaps besteht in KRPAÑO grundlegend die Wahl zwischen der Verwendung von Depthmap-Bildern und der Verwendung von Depthmap-3D-Modellen. Dabei sind die Vor- und Nachteile dieser beiden Depthmap-Typen nicht ausführlich dokumentiert, geschweige denn erforscht. Demnach haben Entwickler bei der Wahl zwischen diesen beiden Optionen nur wenige Anhaltspunkte, um eine fundierte Entscheidung zu treffen. Zusätzlich besitzen beide Depthmap-Typen nochmals eigene Aspekte, die für die Qualität und Performance der Panoramatur entscheidend sind, aber in ihren Vor- und Nachteilen ebenfalls noch nicht genauer dokumentiert sind. Gerade in Bezug auf die erwähnten Limitierungen bei der Verwendung von Panoramatauren als Webanwendungen, ist es wichtig eine fundierte Entscheidung treffen zu können, auch in Hinblick auf einen effektiven Arbeitsverlauf. Zusätzlich können auch

³https://everpano.s3.eu-central-1.amazonaws.com/3d/sant_marti/

⁴Zum Zeitpunkt des Schreibens (März 2022) besteht der Markt laut statcounter weltweit zu 56.05% aus Mobilgeräten, in Deutschland liegt der Anteil bei 42.8% [Sta22]

ästhetische Faktoren für die Wahl der einen oder anderen Variante bei der Nutzung von Depthmaps entscheidend sein.

1.3 Zielsetzung der Arbeit

Das Ziel dieser Arbeit besteht in der Erstellung eines aussagekräftigen Vergleichs der Vor- und Nachteile der beiden in KRPANO nutzbaren Depthmap-Typen. Dabei sollen die beiden Typen und ihre jeweiligen Varianten auf die Aspekte Performance, Speicheraufwand, Genauigkeit und Ästhetik hin untersucht und verglichen werden. Um dies zu erreichen werden im Rahmen dieser Arbeit verschiedene Testumgebungen entwickelt, mit deren Hilfe vergleichbare Datensätze erstellt werden. Zudem soll ein Blick auf den heutigen Stand der Technik insbesondere unter der Maßgabe des automatisierten Erstellens von Depthmaps geworfen und als Alternative zum manuellen Erstellen von Depthmaps evaluiert werden.

1.4 Aufbau der Arbeit

Die Arbeit ist in sieben Kapitel aufgeteilt, die im Folgenden, beginnend mit dem zweiten Kapitel, kurz eingeführt werden sollen:

Kapitel 2, Grundlagen: In diesem Kapitel werden die für das Verständnis der Arbeit nötigen Grundlagen aus den Bereichen der grafischen Datenverarbeitung und Panoramafotografie aufgezeigt, sowie eine Einführung in die Software KRPANO gegeben.

Kapitel 3, Stand der Technik und Wissenschaft: Die aktuelle Forschungslage zu Themen aus der Peripherie dieser Arbeit, insbesondere mit Blick auf die automatisierte Gewinnung von Tiefeninformationen, sowie der Stand der Technik in Bezug auf aktuell verfügbare Technologien werden in diesem Kapitel vorgestellt.

Kapitel 4, Methodik: Die zur Messung und Evaluation von Depthmaps aufgestellte Methodik wird hier erläutert und begründet, sowie das Verfahren zur Gewinnung der Testdaten vorgestellt.

Kapitel 5, Datenanalyse: Die Daten, die aus dem in der Methodik vorgestellten Verfahren gewonnenen wurden, werden in diesem Kapitel analysiert und interpretiert.

Kapitel 6, Ergebnisse: In diesem Kapitel werden die aus der Datenanalyse gewonnenen Erkenntnisse in einem Vergleich gegenübergestellt und evaluiert.

Kapitel 7, Zusammenfassung und Ausblick: Die gesamte Arbeit und die Ergebnisse werden hier zusammengefasst und ein Ausblick auf mögliche, weitere Forschungsaufgaben geworfen.

1.5 Zusammenfassung

Virtuelle Panoramarundgänge gewinnen mit dem Aufkommen von leistungsstarken Mobilgeräten und VR-Brillen in den letzten Jahren immer mehr an Bedeutung. Anwendungen wie Google's *STREET VIEW* erfreuen sich dabei aufgrund der Möglichkeit zur Erkundung ferner Orte großer Beliebtheit.

In dieser Arbeit wird ein Problem beleuchtet, das beim Übergang zwischen zwei Panoramabildern entsteht. Hierbei wird beim Nutzer ein räumlicher Sprung vernommen, der die Immersion in die Anwendung brechen kann. Die Software *KRPANO*, mit der webbasierte Panoramatauren erstellt werden können, bietet als einzige auf dem Markt eine Lösung des Problems an, welche in der Verwendung von 3D-Modellen oder Z-Depth-Bildern zur Erweiterung von Panoramatauren mit Tiefeninformationen besteht. In dieser Arbeit wurden diese beiden Möglichkeiten der Nutzung von sogenannten Depthmaps in *KRPANO* untersucht und Daten erfasst, anhand derer die Depthmap-Typen miteinander verglichen werden konnten. Zum Vergleich der Depthmap-Typen wurden zunächst folgende Vergleichskriterien aufgestellt:

1. Speicheraufwand und Downloadzeiten
2. Performance
3. Präzision von Projektion und Geometrie
4. Ästhetik
5. Arbeitsaufwand
6. Integration von Hotspots

Zur Ermittlung der Performance, der Präzision und Ästhetik wurden schließlich Testumgebungen erstellt, die jeweils aus Panoramatauren mit Panoramabildern bestehen, die größtenteils mit 3D-Programmen gerendert wurden. Das Rendern von Panoramabildern brachte dabei den Vorteil, dass die 3D-Modelle gleich als Depthmaps weiterverwendet werden konnten, und somit Fehler bei der manuellen Erstellung von 3D-Modellen ausgeschlossen wurden.

Die aus den Testumgebungen erfassten Daten wurden anschließend ausgewertet und miteinander verglichen. Dabei ergab sich, dass die Verwendung von Depthmap-3D-Modellen gegenüber Depthmap-Bildern einige gravierende Vorteile bringt. So erreichen 3D-Modelle eine höhere Präzision und sind dabei in den meisten sowohl performanter als auch effizienter (bezogen auf den Speicheraufwand). Zudem lassen sich in Depthmap-3D-Modelle besser *KRPANO*-Hotspots integrieren und der Arbeitsaufwand zur Erstellung von Depthmap-3D-Modellen ist im Vergleich zu Depthmap-Bildern geringer.

Auch in Bezug auf die Ästhetik sind Depthmap-3D-Modelle in manchen Aspekten überlegen. So liefern sie eine Bildqualität, die aufgrund der Genauigkeit der Geometrie deutlich schärfer, und somit hochwertiger, wirkt, als bei Depthmap-Bildern. Zudem können sie Flächen und vor allem schräge Kanten deutlich besser darstellen.

Allerdings kann in Bezug auf die Ästhetik auch ein Argument für die Verwendung von Depthmap-Bildern gemacht werden, da sie zwar im Gegensatz zu Depthmap-3D-Modellen die Geometrie verzerren, allerdings nicht wie diese eine verdoppelte Darstellung des Panoramabildes in den Schattenbereichen der Projektion hervorbringt. Diese Verdoppelung bei der Darstellung mit Depthmap-3D-Modellen kann dabei sehr irritierend wirken, wodurch die Verwendung von Depthmap-Bildern und der damit einhergehenden Verzerrung der Geometrie aus ästhetischen Gesichtspunkten durchaus in Betracht kommen kann.

Aus den Ergebnissen ließen sich drei verschiedene Anwendungsfälle folgern, die als Entscheidungsgrundlage zur Auswahl einer der beiden Depthmap-Typen verwendet werden können, wobei die meisten Anwendungsfälle bereits durch den ersten Punkt abgedeckt sein sollten.

1. Verwendung von Depthmap-3D-Modellen bei einer Polygonanzahl von bis zu 32.000 Tris (keine Latenz) oder 250.000 Tris (geringfügige Latenz).
2. Verwendung von Depthmap-Bildern aufgrund der möglicherweise ansprechenderen Ästhetik.
3. Verwendung von Depthmap-Bildern bei Panoramabildern, die aus hochkomplexen 3D-Szenen gerendert wurden und eine Polygonanzahl von über 250.000 Tris aufweisen.

Kapitel 2

Grundlagen

In der Erweiterung von Panoramabildern durch Tiefeninformationen zur Darstellung in web-basierten virtuellen Touren kommen Technologien aus den Bereichen Fotografie, grafische Datenverarbeitung und Webprogrammierung zum Einsatz. Um die Methoden zu eben dieser Erweiterung vergleichen und evaluieren zu können, ist ein Grundverständnis der Technologien in diesen Bereichen vonnöten. In diesem Kapitel sollen daher zunächst allgemeinere Grundlagen erläutert werden, worauf aufbauend spezielle, für die Erstellung von Panoramatouren mit KRPAÑO, notwendige Voraussetzungen aufgezeigt werden.

2.1 Digitale Bilddaten

Üblicherweise erfolgt die Darstellung der Panoramabilder schlussendlich über einen Webbrowser, auf der Basis von digitalen und web-kompatiblen Formaten. Dafür wird bei Bildern der RGB-Farbraum verwendet, welcher als additiver Farbraum die drei Grundfarben Rot, Grün und Blau als einzelne Werte zwischen 0 und 1 abbildet. Dabei bedeutet der Wert 0, dass die entsprechende Farbe nicht verwendet wird, wobei 1 die volle Verwendung der Farbe bedeutet. Durch die Addition der Farbwerte entsteht die Farbe Weiß, wenn die einzelnen Farbwerte jeweils den Wert 1 besitzen und es entsteht Schwarz, wenn alle Werte auf 0 sind. Dieser Sachverhalt lässt sich durch die Verwendung eines Würfels modellieren (vgl. dazu Abbildung 2.1). Meist werden diese Farben im digitalen Raum in einem 8-Bit-Format verwendet, was bedeutet, dass die einzelnen Farbwerte auf ganzzahlige Werte zwischen 0 und 255 (2^8) abgebildet werden. Die Werte werden separat in einzelnen Kanälen gespeichert.

Neben dem RGB-Farbraum ist für diese Arbeit noch die Verwendung von Bildern mit nur einem Kanal interessant. Die Verwendung von nur einem Kanal ermöglicht dabei lediglich die Darstellung als Graustufenbild mit Helligkeitsabstufungen in der Höhe der Verwendeten Bittiefe (8-Bit = 256, 16-Bit = 65536, usw.). Neben der Darstellung von Schwarz-Weiß-Bildern werden diese Art von Bilddaten meist zur Speicherung anderweitiger Daten, wie z.B. von Tiefeninformationen, verwendet (mehr dazu in Abschnitt 2.3.2).

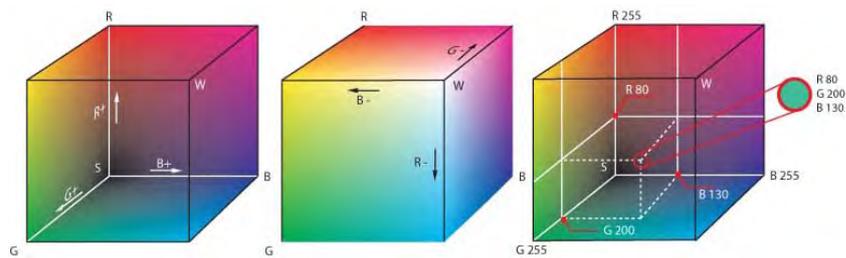


Abbildung 2.1: Modellierung des RGB-Farbraums als linearen Raum, dargestellt als Würfel [Wik21]. Die drei Achsen repräsentieren dabei die einzelnen Farbkanäle.

2.2 Panoramabilder

Panoramabilder sind Bilder, die durch ein weites (bzw. hohes) Sichtfeld einen großen Raum darstellen können. Solche Bilder können auf verschiedene Weisen produziert werden. Sie können als einzelnes Bild von einer Kamera mithilfe eines Weitwinkelobjektivs fotografiert werden, oder sie können aus mehreren Fotografien zusammengesetzt werden. In dieser Arbeit werden in Bezug auf Panoramabilder ausschließlich solche betrachtet, die einen 360° Blickwinkel besitzen. Das heißt, bei diesen Panoramabildern wird jeder Teil des umgebenen Raumes sphärisch abgedeckt und als Fotografie abgebildet. Aus diesem Grund wird diese Art von Panoramabild auch *Kugelpanorama* genannt.

Ein solches Kugelpanorama lässt sich aufgrund seiner Form jedoch nicht einfach mit einem rechteckigen Bildformat darstellen. Um es so speichern zu können, muss das Panoramabild zunächst so verzerrt werden, dass es auf ein rechteckiges Bildformat passt, wobei keine Bildinformationen verloren gehen sollen. Diese Verzerrung erfolgt üblicherweise durch eine Zylinderprojektion (Rektangularprojektion), bei welcher vor allem die oberen und unteren Teile des Bildes, wie in Abbildung 2.2 zu sehen, stark verzerrt dargestellt werden. Ein so transformiertes Panoramabild besitzt dabei ein Seitenverhältnis von 2:1, was den Winkeln 360° in der Horizontalen und 180° in der Vertikalen entspricht. Der rechte und linke Rand des Bildes gehen nahtlos übereinander über, sodass bei der Betrachtung eines Panoramabildes in einem *Panorama-Viewer* (eine Software zum interaktiven Betrachten eines 360° Panoramabildes ohne Verzerrung), kein Anfang und Ende des Bildes zu erkennen sind. Dabei wird oft die rektilineare Projektion verwendet (Abbildung 2.2, rechts), welche es bis zu einem Winkel von 120° erlaubt Bildbereiche ohne Verzerrung darzustellen [Wit19].

Wenn bei einer Betrachtung des Panoramabildes in einem Panorama-Viewer von einem Blickwinkel von 90° ausgegangen wird, muss das Panoramabild, für das Erreichen einer bestimmten Auflösung im Panorama-Viewer, in der Breite die vierfache Auflösung besitzen. Bei einer Zielauflösung von 1920×1080 Pixel würde dies für das Panoramabild eine Auflösung von 7680 Pixel in der Breite bedeuten.

2.2.1 Panoramafotografie

Für das Fotografieren von 360° Panoramabildern wird spezielle Hardware verwendet, die sowohl aus vollständig automatisierten Systemen bestehen kann, als auch aus einfachen



Abbildung 2.2: Vergleich eines 360°-Panoramabildes in Rektangularprojektion (links) mit einer geradlinigen Projektion aus demselben Standpunkt (rechts). Die Wölbung des Gebäudes und der Straße entstehen erst durch die Verzerrung der Rektangular-Projektion.

Aufsätzen für Kamerastative (Panoramaköpfe/Nodalpunktadapter). Das Prinzip ist bei beiden Ansätzen jedoch das Gleiche. Es werden zeitversetzt oder zeitgleich in verschiedenen Rotationen genug Bilder geschossen, dass die 360-Grad um die Kamera herum komplett abgedeckt werden und die Bilder sich zusätzlich überlappen. Eine Überlappung von 30% ist hierbei ein gebräuchlicher Wert [Jac04]. Es ist zudem wichtig, dass der Rotationspunkt so nah wie möglich am Sensor (Nodalpunkt) der Kamera ist. Je besser der Nodalpunkt erreicht wird, desto identischer wirken zwei Objekte, die aus unterschiedlichen Kamerarotationen entstanden sind (und somit einen anderen Teil im Bildausschnitt besetzen). Je identischer die Bildobjekte sind, desto besser lassen sich zwei Bilder im nächsten Schritt zusammenfügen. Dies geschieht nämlich mithilfe von Kontrollpunkten, die anhand von Ähnlichkeiten in den sich überlappenden Bildteilen gesetzt werden.

Eine Software wie PTGUI¹ kann solche Kontrollpunkte größtenteils automatisiert setzen, und anhand dieser Kontrollpunkte die Bilder zu einem Panorama zusammenfügen. Dieser Vorgang nennt sich *Stitching*. Das Programm muss dabei auch die durch das verwendete Objektiv entstandenen Verzerrungen und die Größe des Sensors berücksichtigen, dafür werden intern Kameraprofile verwendet, die für neue Hardware deshalb auch beständig aktualisiert werden müssen. Zusätzlich müssen Effekte wie chromatische Aberrationen und Vignettierungen herausgerechnet werden, da diese zu sichtbaren Kanten innerhalb des Panoramabildes führen können.

Ein weiterer Teil der Erstellung von Panoramabildern — besonders wichtig für das spätere Erstellen von Depthmaps — ist das *Ausrichten*, welches ebenfalls mit der Software PTGUI geschehen kann. Dabei wird versucht den Horizont im Bild möglichst geradezustellen. Dies kann entweder manuell nach Augenmaß erfolgen oder durch das Setzen von vertikalen Linien, anhand derer PTGUI die korrekte Ausrichtung ermittelt.

Bei der Fotografie von Panoramabildern besteht zudem das Problem, dass das Kamerastativ beim Fotografieren des unteren Bereiches (des Nadirs) selbst im Bild ist. Abhilfe kann hier entweder durch das nachträgliche Retuschieren mit einem Bildbearbeitungsprogramm, oder durch das ungenaue Fotografieren des Nadirs aus der Hand, geschaffen werden.

¹PTGUI (<https://ptgui.com/>) ist eine kostenpflichtige Software, die auf dem Plugin PANORAMA TOOLS (<http://panotools.sourceforge.net/>) basiert und laut [Ack22] derzeit im Vergleich mit der Konkurrenz die beste Stitchingqualität liefert.

2.2.2 Panorama-Rendering

Ein großer Nachteil des Renderns von Panoramabildern ist der große Rechenaufwand und die damit verbundene Renderdauer. Diese entsteht vor allem durch die hohe Auflösung der Bilder (im Rahmen dieser Arbeit meist 8192 x 4096 Pixel). In dieser Arbeit wurde daher zum Rendern von Panoramabildern die UNREAL ENGINE 4² verwendet. Der gesamte Prozess wurde dabei in Kapitel 4 Methodik, Abschnitt (4.3) ausführlich dargestellt. Die UNREAL ENGINE 4 wurde primär aus dem Grund ausgewählt, dass sie im Vergleich zu herkömmlichen 3D-Renderengines, wie AUTODESK ARNOLD oder Chaos V-RAY, eine drastisch kürzere Renderzeit bei einer trotzdem noch guten Qualität ermöglicht. Dabei wird ein Hybrid aus Ray-Tracing und Rasterungsverfahren verwendet [Epi21].

Herausforderungen, wie das Entfernen des Stativs bei der Fotografie von Panoramabildern, gibt es beim Rendern von Panoramabildern in einem 3D-Programm nicht. Hier wird eine virtuelle Kamera eingesetzt, die nicht an physikalische Gegebenheiten gebunden ist. Sie kann eine unendlich kleine Sensorgröße besitzen und ein Objektiv mit einem Blickwinkel von 360°, was das Rendern eines Panoramabildes in einem Stück ermöglicht. Eine solche Implementation einer sphärischen 360°-Kamera gibt es nativ beispielsweise in den schon genannten Renderengines AUTODESK ARNOLD und Chaos V-RAY, aber nicht bei den Renderengines der kostenfreien Alternative BLENDER oder auch der UNREAL ENGINE 4. Es gibt jedoch auch Plugins für diese Programme, die das direkte Rendern von Panoramabildern ermöglichen. Alternativ lassen sich gerenderte Panoramabilder auch auf ähnliche Weise wie fotografierte Panoramabilder erstellen, durch das Rendern einzelner Flächen einer Cube-Map.

2.2.3 Cube-Maps

Cube-Maps stellen eine Möglichkeit der Speicherung von 360°Panoramabildern ohne Verzerrung dar. Wie in Abbildung 2.3 zu sehen werden dabei sechs einzelne quadratische Flächen zu einer Kreuzform zusammengesetzt. Diese sechs einzelnen Flächen repräsentieren dabei jeweils eine Seite eines 360°-Panoramas mit einem Blickwinkel von 90°. Umgekehrt lassen sich in einem 3D-Programm auch einzelne Flächen einer solchen Cube-Map rendern und zu einem gesamten Cube-Map-Panorama zusammenfügen. Dafür muss lediglich eine Kamera mit einem Blickwinkel von 90° und einem quadratischen Seitenverhältnis die sechs Seiten einzeln abbilden. Die sechs einzelnen Flächen lassen sich zudem unter anderem mit der Software PTGUI direkt in ein rektangulares Panorama umwandeln.

2.3 Tiefeninformationen

Zur Erweiterung von Panoramabildern durch Tiefeninformationen bestehen im Wesentlichen zwei Möglichkeiten: die Projektion von Panoramabildern auf ein 3D-Modell und die

²Version 4.27: <https://docs.unrealengine.com/4.27/>

³Bildquelle: <https://onix-systems.medium.com/how-to-use-360-equirectangular-panoramas-for-greater-realism-in-games-55fadb0547da>



Abbildung 2.3: Beispiel einer Cube-Map, die sich aus einzelnen Würfelflächen zusammensetzt.³

Zuordnung von Tiefeninformationen zu jedem Pixel eines Panoramabildes (Z-Buffer). Beide Varianten verändern dabei nicht das Aussehen eines Panoramabildes, wenn es aus der zentralen Perspektive heraus (beispielsweise in einem Panorama-Viewer) betrachtet wird, sondern erst, wenn ein anderer Standpunkt angenommen wird. Das liegt natürlich daran, dass die Projektion des Panoramabildes aus exakt dem Standpunkt erfolgt, aus dem es auch Betrachtet wird.

2.3.1 3D-Modelle

3D-Modelle beschreiben allgemein Körper im dreidimensionalen Raum. Dabei gibt es viele verschiedene Ansätze zur Beschreibung von solchen 3D-Modellen, im Kontext dieser Arbeit (Verwendung von 3D-Modellen in einer Web-Umgebung) ist jedoch nur die Verwendung von 3D-Modellen als Polygonnetz (Mesh) relevant. Polygone sind dabei Flächen, die durch mehrere Eckpunkte im 3D-Raum beschrieben werden. Üblicherweise werden Polygone mit drei (Tris) oder vier (Quads) Eckpunkten verwendet. Einzelne Polygone können dabei Eckpunkte teilen und somit ein Netz bilden, um zusammen einen Körper zu repräsentieren. Jedes Polygon kann zudem einen Normalvektor haben, der die Richtung beschreibt, aus welcher dieses Polygon als solches sichtbar ist. Zusätzlich kann ein dreidimensionales Polygon auf ein 2D-Koordinatensystem gemappt werden (UV-Mapping), um eine 2D-Textur auf dem 3D-Modell darzustellen. Dies kann im Kontext von *KRPANO* beispielsweise für die Darstellung von *Dollhouses* (vgl. Abschnitt 2.5.5) verwendet werden.

3D-Modellierung anhand von Panoramabildern

Zur Verwendung von 3D-Modellen als Tiefeninformation für Panoramabilder muss zunächst die Geometrie der vom Panoramabild abgebildeten Szene nachmodelliert werden. Dafür wird zunächst das Panoramabild in ein 3D-Modellierungs-Programm geladen und auf eine Kugel projiziert. Die Kamera des 3D-Programms wird in das Zentrum dieser Kugel gestellt. So kann nun ein 3D-Modell nachgebaut werden, indem beispielsweise die Kanten der Polygone

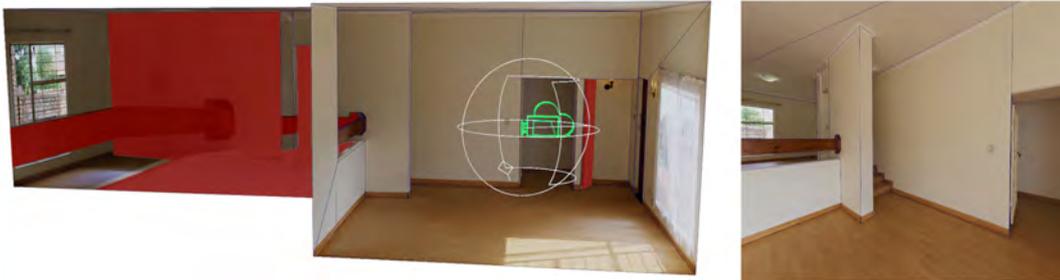


Abbildung 2.4: In der 3D-Software Autodesk Maya modellierter Raum, ausgehend von einem Kugelpanoramabild⁴, welches vom Kamerastandpunkt (grünes Symbol) aus sphärisch auf die Geometrie projiziert wird. Die rot-markierten Bereiche liegen im Schatten der Projektion, somit fehlen an diesen Stellen Bildinformationen. Das Bild rechts ist dabei die Sicht aus diesem Kamerastandpunkt heraus, das Wireframe des 3D-Modells überlagert das unverzerrte Panoramabild

auf Kanten von Wänden in der Szene des Panoramabildes gesetzt werden (vgl. Abbildung 2.4). Sofern die Szene ebene Böden und Wände aufweist, ist es lediglich notwendig, die Höhe des Kamerasensors zu definieren und diese auf die 3D-Szene zu übertragen. So ergibt sich eine korrekte Skalierung. Bei unebenen Böden oder nicht-rechtwinkligen Objekten können jedoch weitere Messdaten vonnöten sein, da hier sonst keine eindeutige Aussage in Bezug auf die geometrische Lage von Objekten getroffen werden kann.

2.3.2 Z-Depth-Bilder

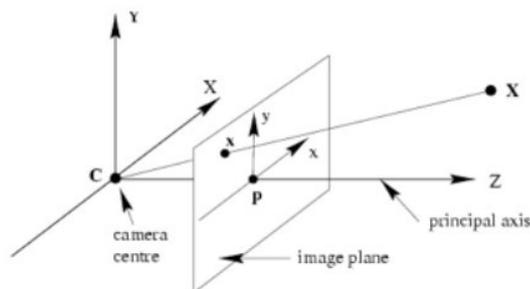


Abbildung 2.5: Werte auf der Z-Achse einer Bildebene können Pixeln eine Tiefe zuordnen [HZ04].

Der Buchstabe Z weist in diesem Zusammenhang auf die z-Koordinate, also der Tiefe der Bildebene hin (vgl. Abbildung 2.5). Einem Pixel wird hierbei ein z-Wert zugeordnet, der die Entfernung des dort abgebildeten *Polygonpunktes* [BB06] bzw. des Hintergrunds, in Relation zur Bildebene darstellt. Diese Art der Speicherung von Tiefeninformationen findet meist während der Rasterung von 3D-Szenen statt. Hier werden die Tiefeninformationen unter anderem verwendet, um zu determinieren, in welcher Ordnung die Polygone einer 3D-Szene vor- bzw. hinter-einander liegen. Der zu diesem Zweck verwendete Algorithmus heißt Z-Buffer/Tiefenpuffer-Algorithmus. Beim Berechnen der Tiefe mit einer sphärischen

⁴Kugelpanorama veröffentlicht von Greg Zaal, frei verfügbar unter: <https://polyhaven.com/a/lebombo>

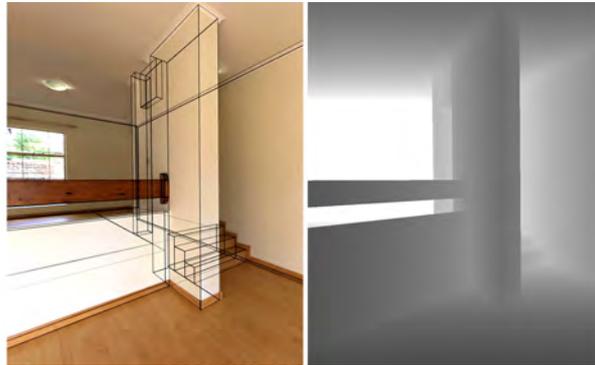


Abbildung 2.6: Aus einem Panoramabild 3D-Modellierter Raum (links) und die korrespondierende Depth-Map (rechts)

Kamera kann die z-Achse der Bildebene jedoch nicht verwendet werden, da es nicht nur eine Blickrichtung gibt. Jeder Pixel weist eine leicht verschiedene Blickrichtung auf, sodass hier die Distanz zwischen der Kameraposition (anstatt der Kamera-/Bildebene) und dem Polygonpunkt des Pixels verwendet werden muss.

Da jedem Pixel über diese Verfahren ein Tiefenwert zugewiesen wird, lässt sich das Ergebnis als ein Graustufenbild darstellen (siehe Abbildung 2.6). Die Bittiefe eines solchen Bildes entscheidet dabei maßgeblich über die Genauigkeit der Tiefeninformationen, da sich mit jedem weiteren Bit die Anzahl der möglichen Werte pro Pixel verdoppelt.

Bei der Division der maximalen Distanz der Kamera zu einem Polygonpunkt und der Bittiefe (Anzahl möglicher Grauwerte), lässt sich die Genauigkeit der Tiefeninformationen als minimal-mögliche Distanz (s_{min}) darstellen. Exemplarisch wurde die Rechnung hier für eine maximale Distanz von 10 m vollzogen:⁵

$$s_{min}(8bit) = 10m \div 2^8 = \underline{0,039m}$$

$$s_{min}(16bit) = 10m \div 2^{16} = 0,153 \cdot 10^{-3} = \underline{0,153mm}$$

$$s_{min}(24bit) = 10m \div 2^{24} = 0,596 \cdot 10^{-6} = \underline{0,596\mu m}$$

$$s_{min}(32bit) = 10m \div 2^{32} = 2,328 \cdot 10^{-9} = \underline{2,328nm}$$

2.4 Panoramatouren mit krpano

In dieser Arbeit soll ein Vergleich zwischen den, in der Software `KRPANO` vorgefundenen, Implementationen der im vorigen Abschnitt beschriebenen Varianten der Darstellung von Tiefeninformationen, vorgenommen werden. Um dies nachvollziehen zu können ist ein Grundverständnis der Funktionsweise von `KRPANO` vonnöten. In diesem Abschnitt sollen diese notwendigen Kenntnisse erläutert, sowie allgemein das Prinzip von Panoramatouren erklärt werden.

⁵Üblicherweise werden in Renderengines als Grenzwerte die in der Kamera eingestellten Near- und Far-Clipping Werte verwendet.

2.4.1 Panoramatouren

Als Panoramatouren oder auch *virtuelle Rundgänge* werden interaktive Anwendungen bezeichnet, in welchen mehrere Panoramabilder miteinander verknüpft sind, sodass ein Anwender zwischen den verschiedenen Panoramabildern wechseln kann. Die Panoramabilder werden dabei mithilfe eines Panorama-Viewers unverzerrt dargestellt und der Anwender kann sich mithilfe eines Eingabegeräts — beispielsweise einer Maus — in dem Panoramabild umschauen und Interaktionselemente wie Pfeile oder andere Buttons bedienen. Das bekannteste Beispiel für Panoramatouren ist Google's `STREET VIEW`.

2.4.2 Übergang zwischen Panoramabildern

Panoramabilder sind in der Lage die Realität aus einem Standpunkt heraus perfekt abzubilden. Demnach kann durch das Verwenden von Panoramatouren ein hoher Grad an Immersion erreicht werden, welche zusätzlich durch die Interaktionsmöglichkeiten eines Panorama-Viewers verstärkt wird. Die Schwäche von Panoramatouren liegt allerdings im Übergang zwischen zwei Panoramabildern. Hierbei erfährt der Nutzer das Gefühl eines Bruchs in der Realität dadurch, dass er in das nächste Panoramabild *springt*, anstatt sich räumlich in den nächsten Bereich zu *bewegen*. Meist wird beim Übergang zweier Panoramabilder lediglich ein einfacher Überblendungseffekt verwendet, teilweise wird auch einfach das nächste Bild ohne Übergangseffekt direkt angezeigt. Es gibt allerdings auch Tricks, mit denen eine Bewegung vorgetäuscht werden kann. Dies kann beispielsweise mithilfe eines Zoom-Effekts getan werden, bei dem der Panorama-Viewer beim Verlassen eines Bildes dieses Bild im Blickwinkel verengt, während das neu geladene Bild eingeblendet und aus einer weiteren Sicht heraus in den normalen Blickwinkel geführt wird.

Eine weitere, und weitaus effektivere Methode zum Erreichen eines Bewegungseffektes kann durch das Verwenden von Depthmaps erreicht werden. Hierbei werden, wie in Abschnitt 2.3 beschrieben, entweder 3D-Modelle oder Depthmap-Bilder dazu verwendet Panoramabilder durch Tiefeninformationen zu erweitern. Die Panoramabilder erhalten so eine 3D-Geometrie mit deren Hilfe dem Panoramastandpunkt ein Punkt im dreidimensionalen Raum zugewiesen werden kann. Nun kann beim Übergang zwischen zwei Panoramabildern eine räumliche Bewegung von Panorama A zu Panorama B stattfinden, während die Bilder selbst gleichzeitig überblendet werden. Wie in den Abbildungen 2.7 und 2.8 zu sehen, entstehen somit bei diesem Übergang viele temporäre Interpolationen von Panoramabildern, welche aus zugleich zwei überblendeten, und auf eine 3D-Geometrie projizierten Panoramabildern bestehen.

Dieser Bewegungseffekt simuliert somit nicht eigentlich eine Bewegung, sondern es passiert (virtuell) eine tatsächliche, räumliche Bewegung zwischen zwei Standpunkten im dreidimensionalen Raum. Allerdings ist diese Lösung auch nicht perfekt. Da die Bildinformationen nur von den beiden Panoramabildern kommen, wird nicht unbedingt jeder Bereich der Geometrie durch eigene Bildinformationen abgedeckt. Es entstehen hinter Objekten Schattenbereiche, in denen keine Bildinformationen vorhanden sind, und die daher auch kein akkurates Abbild der Wirklichkeit darstellen können. Bei der Verwendung von Depthmap-Bildern bestehen zudem in diesen Bereichen auch keine Tiefeninformationen, da diese an

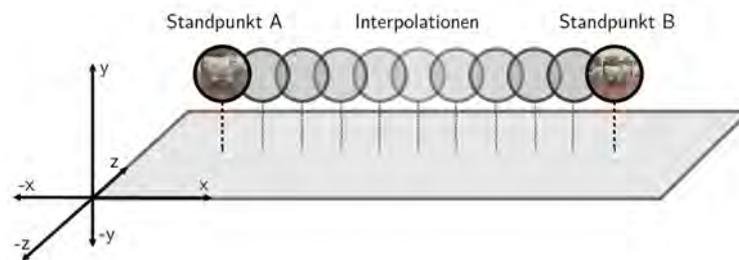


Abbildung 2.7: Beim Übergang der beiden Panorambilder von Standpunkt A zu Standpunkt B entstehen durch die räumliche Bewegung und die Überblendung zweier Panorambilder Interpolationen.



Abbildung 2.8: Beispiel einer sich in der Bewegung befindlichen Interpolation. Es Überlagern sich zwei Panorambilder aus zwei stark verschiedenen Standpunkten (Standpunkt A ist nah hinter der aktuellen Perspektive; Standpunkt B liegt auf dem weißen Kreis (zentral, rechts))

die Pixel des Panorambildes gekoppelt sind. Aus diesem Grund sind auch die Geometrien zweier unterschiedlicher Panorambilder bei der Verwendung von Depthmap-Bildern nicht identisch.

2.5 krpano

KRPANO ist eine proprietäre Software zum Erstellen von Panoramatauren. Dabei können Panoramatauren sowohl als HTML5- oder als ADOBE FLASH⁶-basierte Webanwendung erstellt werden. KRPANO selbst ist eine JavaScript-Anwendung mit einem umfangreichen Angebot an Features und Tools. Der Panorama-Viewer von KRPANO kann an beliebiger Stelle einer Webseite über einen HTML-`<div>` Tag eingebunden⁷ und direkt an den meisten Browsern und Geräten angezeigt werden.

Zur Beschreibung von Panoramatauren in KRPANO wird eine auf Text-basierende XML-Struktur verwendet, über welche alle Inhalte eingefügt und modifiziert werden können. Über eine eigene Skriptsprache — und auch über JavaScript selbst — können zudem eigene Funk-

⁶Adobe Flash wird jedoch seit dem 31.12.2020 nicht mehr fortgesetzt (<https://www.adobe.com/de/products/flashplayer/end-of-life.html>)

⁷<https://krpano.com/docu/html/#syntax>



Abbildung 2.9: Verwendung von Hotspots in krpano als klickbare Grafik (links) und als polygonale Form mit mouse-hover-Effekt (rechts).

tionalitäten (*Actions*) programmiert werden. Eine HTML5-Schnittstelle ermöglicht weiterhin auch das Erstellen eigener Plugins, die auf den Framebuffer im WebGL-Rendering-Context⁸ der Anwendung zugreifen, und somit auch beispielsweise nachträgliche Farbkorrekturen anwenden können.

2.5.1 Panoramabilder in krpano

Panoramabilder können in `KRPANO` über den `<image>` Tag der XML-Datei eingebunden werden. Dabei werden sowohl viele Bildformate, als auch einige verschiedene Projektionsarten bzw. Panoramaformate unterstützt. Für die Verwendung von Kugelpanoramen sind dabei im Besonderen die Panoramaformate `<cube>` und `<sphere>` relevant. Beide Formate ermöglichen die Verwendung der *Multi-Resolution* Einstellung, sowie die Verwendung von Depthmaps. Die Verwendung von *Multi-Resolution*-Bildern hilft durch die Verwendung von verschiedenen Bildauflösungen vor allem dabei, die Performance auf schwächeren Geräten zu verbessern, indem für manche Zwecke auf geringere Auflösungen zurückgegriffen werden kann. Dies kann auch dazu beitragen die Ladezeiten von Bildern zu verringern, was zu einer besseren Immersionserfahrung beitragen kann.

In diesem Kontext ist auch die Möglichkeit zur Verwendung eines gering-auflösenden Preview-Bildes zu nennen. Dieses wird beim Laden eines Panoramabildes zuerst angezeigt und kann durch die schnelle Ladezeit verhindern, dass es bei Übergängen zu Verzögerungen kommt. Das eigentliche Panoramabild wird geladen, während das Preview-Bild bereits angezeigt wird.

Bei der Verwendung des `<cube>` Panoramaformats können die einzelnen Würfelflächen zudem separat angezeigt werden, sodass nicht das gesamte Panoramabild auf einmal geladen werden muss, wodurch ebenfalls Daten und Ladezeiten gespart werden.

2.5.2 Hotspots in krpano

Hotspots (`<hotspot>`) sind Elemente, die zusätzlich in der Panoramatour platziert werden und für Interaktionen zuständig sind. Durch den Klick auf einen Hotspot können dabei verschiedene Aktionen vollzogen, sowie auch selbst-programmierte *Actions* aktiviert werden. Durch Hotspots werden beispielsweise auch andere Panoramabilder verlinkt. Hotspots können dabei sowohl aus Grafiken, Videos oder aus polygonalen Formen bestehen und können

⁸<https://developer.mozilla.org/en-US/docs/Web/API/WebGLRenderingContext>

sowohl in der Bildebene (at_h , at_v), als auch im dreidimensionalen Raum (tx , ty , tz) platziert werden.

2.5.3 Depthmap-Implementationen in krpano

Seit Version 1.20⁹ unterstützt KRpano das Einbinden von Depthmaps. Dabei können sowohl Z-Depth-Bilder als auch 3D-Modelle als Depthmaps verwendet werden. Diese Erweiterung durch Tiefeninformationen ermöglicht dabei neben einem 3D-Übergangseffekt auch das stereoskopische Rendern bei nur einem einzigen Bild zur Darstellung mit VR-Brillen.

Die Depthmaps werden dabei über den `<depthmap>`-Tag innerhalb eines `<image>`-Tags eingebunden und können somit nur in Kombination mit einem Panoramabild verwendet werden. Die Depthmaps können dabei unabhängig vom Panoramabild im Raum transformiert werden, allerdings können sie nicht unabhängig rotiert werden, weshalb dies im Fall von 3D-Modellen bereits in der 3D-Software erfolgen muss. Dies muss insbesondere aufgrund der verschiedenen Verwendung von Achsen berücksichtigt werden, welche, wie in Abbildung 2.10 zu sehen, je nach Software unterschiedlich bezeichnet werden.

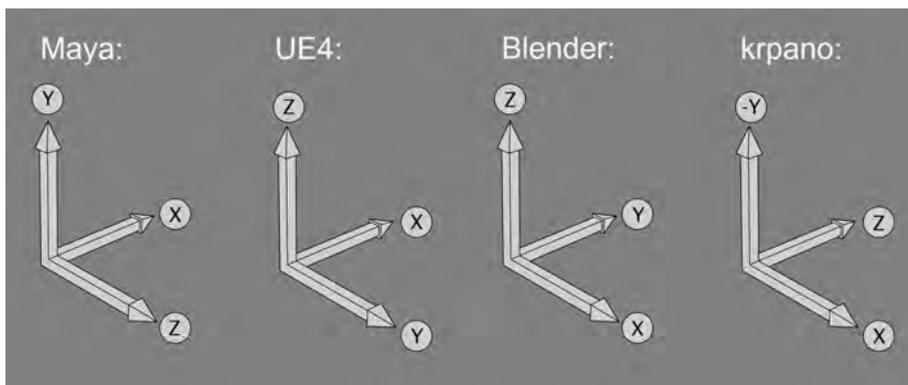


Abbildung 2.10: Vergleich der verschiedenen Achsenbeschriftungen in unterschiedlichen 3D-Programmen. Keines der Programme verwendet dieselben Achsen.¹⁰

3D-Modell-Depthmaps

Zur Verwendung von 3D-Modellen als Depthmaps in KRpano können die drei Dateiformate OBJ, STL (ASCII) und DEPTH zum Einsatz kommen. Das DEPTH Dateiformat ist dabei ein spezielles KRpano-Format, welches schnelles Laden und einfaches *Parsen* (lesen des Codes durch einen Computer) ermöglichen soll¹¹. Sowohl das STL-, als auch das DEPTH-Dateiformat sind Textformate, die lediglich die Verwendung von Dreiecken als Grundform unterstützen. STL bietet dabei zusätzlich noch die Angabe eines Normalvektors.

⁹Veröffentlicht am 9.2.2022 (<https://krpano.com/docu/releasenotes/#krpano120>)

¹⁰Ursprüngliche Grafik ergänzt durch Maya und Unreal Engine 4 (Quelle: <https://krpano.com/forum/wbb/index.php?page=Thread&postID=84177#post84177>)

¹¹<https://krpano.com/docu/xml/#depthmap>

Eine DEPTH-Datei wird aus einer STL-Datei mittels eines von KRPANO bereitgestellten Tools¹² konvertiert.

Lediglich das OBJ-Format unterstützt die Verwendung von UV-Maps, wodurch nur dieses Format für die Verwendung von Dollhouses (siehe Abschnitt 2.5.5) infrage kommt.

Depthmap-Bilder

Für die Verwendung von Depthmap-Bildern in KRPANO können zwei verschiedene Formate verwendet werden. Zum einen kann ein 8-Bit Graustufenbild mit einem einzigen Kanal verwendet werden, zum andern ein RGB-Farbbild mit drei Kanälen von jeweils 8-Bit. Das RGB-Format kann dabei verwendet werden, um ein Z-Depth-Bild mit 24-Bit auf die drei Farbkanäle aufzuteilen. Dabei werden die acht in der Wertigkeit höchsten Bits in den roten Kanal, die nächsten acht in den Grünen und die Bits mit der geringsten Wertigkeit in den blauen Kanal gelegt. Der blaue Kanal kann jedoch auch genullt werden (alle Werte auf 0 gesetzt werden), um Speicherplatz zu sparen, da die Genauigkeit bei der Verwendung von 16 effektiven Bits schon sehr hoch ist (siehe die Rechnung in Abschnitt 2.3.2). Die jeweiligen Depthmap-Bilder und der einzelnen Kanäle werden in Abbildung 2.11 dargestellt.

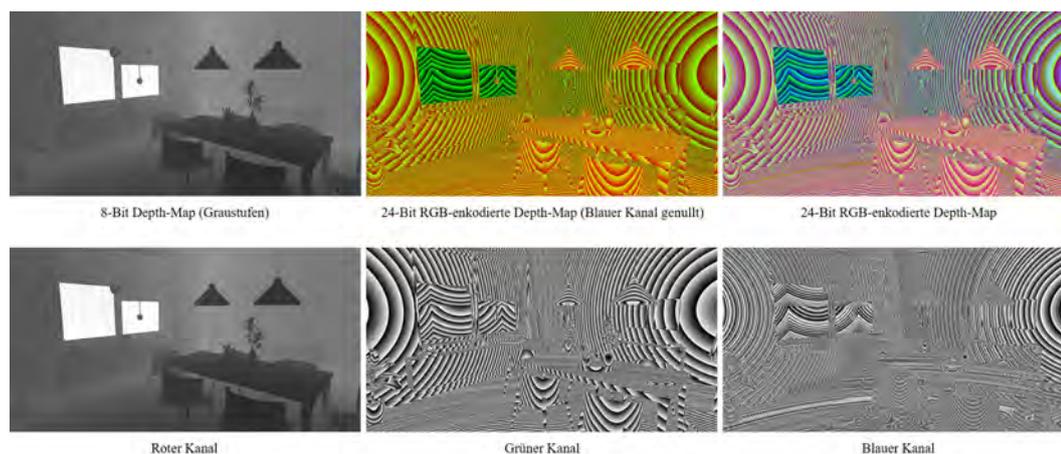


Abbildung 2.11: Vergleich der in krpano verwendbaren Depthmap-Bilder und die jeweiligen Kanäle einer RGB-kodierten Depthmap. Die Muster bei den RGB-kodierten Bildern entstehen dadurch, dass beispielsweise beim grünen Kanal jeder Wert von 0 bis 255 in den Bereich eines einzigen Grauwerts des roten Kanals fällt. Der blaue Kanal bildet dasselbe beim in Bezug auf den grünen Kanal ab, wobei hier jedoch die Auflösung der Pixel nicht annähernd ausreicht, um den vollen Wertumfang auszunutzen.

Bei der Nutzung von Depthmap-Bildern wird in KRPANO eine Kugel als Grundmesh verwendet, auf welche die Tiefeninformationen der Depthmap angewendet werden. Dabei werden ausgehend von den Tiefenwerten der Depthmap-Pixel, einzelne Punkte des Meshs der Kugel entlang ihrer Normalvektoren verschoben. Die Kugel besitzt dabei eine Polygonanzahl von 800.000 Quads.

¹²<https://krpano.com/docu/depth-file-format/>

2.5.4 3D-Übergangseffekt in krpano

Über solche Hotspots kann auch der Übergang zwischen zwei Bildern mit den in Abschnitt 2.4.2 aufgezeigten räumlichen Bewegungseffekt aktiviert werden. Der folgende Code stellt ein Beispiel mit einer Szene, einem Hotspot und einer *Action* dar, in welchem diesen Über-

gangseffekt implementiert ist:^{13 14}

```
1   <!-- Szene mit Panoramabild und Depthmap -->
2
3   <scene name="Standpunkt_01" >
4     <image>
5       <view tx="10" ty="-5" tz="5" />
6       <sphere url="panos/standpunkt_01.jpg" ox="10" oy="-5" oz="5" />
7       <depthmap url="depthmaps/model_3d.obj"
8         rendermode="3dmodel"
9         scale="1"
10      />
11    </image>
12  </scene>
13
14  <!-- Hotspot aktiviert bei Klick den 3D-Übergangseffekt -->
15
16  <hotspot name="navigation_01" tx="-81.33" ty="15.0" tz="-36.0"
17    onclick="tour3d_loadscene('Standpunkt_01')"
18  />
19
20  <!-- Die Action übernimmt den Namen der Szene und Interpoliert
21    die räumliche Bewegung zu diesem -->
22
23  <action name="tour3d_loadscene" args="scene">
24    loadscene(get(scene), null, MERGE|KEEPVIEW|KEEPMOVING, BLEND(0.50));
25    if(global.customtransition != 'SKIP',
26      if(global.customtransition !== null,
27        global.customtransition(); ,
28      tween(view.tx|view.ty|view.tz,
29        calc(image.ox + '|' + image.oy + '|' + image.oz),
30        2.0, easeinoutsine);
31    );
32  );
33  delete(global.customtransition);
34  delete(global.customtransitiontime);
35  </action>
```

¹³Der Code stammt ursprünglich von „DerMische“ und wurde hier nur leicht adaptiert. <https://dermische.de/2021/01/02/3d-depthmap-tour-panocamadder-tutorial/> (zuletzt abgerufen am 3.6.2022)

¹⁴Die ungewöhnliche Syntax der Action entstammt der krpano-Skriptsprache (<https://krpano.com/docu/actions/>)

2.5.5 Dollhouses

KRPANO unterstützt weiterhin das Verwenden von sogenannten *Dollhouses* im Zusammenhang mit Depthmaps. Diese Funktion wird in der Regel dazu verwendet ein 3D-Modell der Panoramatur von außen zu zeigen, und somit wie ein Puppenhaus darzustellen.

Um ein Dollhouse darzustellen, kann ein texturiertes 3D-Modell verwendet werden. Die Texturierung muss dabei bereits im Voraus in einem 3D-Programm erfolgen und kann beispielsweise über die Kombination aus der Projektion mehrerer Panoramaprojektionen auf eine Geometrie erfolgen. Diese Synthese kann schließlich auf eine UV-Map gemappt und so eine zweidimensionale Textur erstellt werden. Diese Textur kann in KRPANO als im `<image>`-Tag als `<sphere>` hinzugefügt und auf das als Depthmap eingefügte 3D-Modell projiziert werden. Dabei ist es jedoch nicht möglich im Hintergrund noch ein weiteres Panoramabild anzuzeigen.



Abbildung 2.12: Beispiel eines sogenannten Dollhouses, für welches die Textur (rechts) mit Panoramabildern aus mehreren Standpunkten erzeugt wurde.

Kapitel 3

Stand der Technik und Wissenschaft

Für die Integration von Tiefeninformationen in webbasierte Panoramatauren werden Technologien aus verschiedenen Bereichen verwendet. Im vorigen Kapitel wurden die Grundlagen dieser Technologien und die Funktionsweise der Software `KRPANO` erläutert, welche, bei der Verwendung von Depthmaps, Elemente aus den Bereichen Panoramafotografie, grafische Datenverarbeitung, 3D-Modellierung und Webprogrammierung zu einem gesamten Produkt vereint, dessen Neuartigkeit genau aus dieser Verbindung besteht. Denn diese einzelnen Elemente sind für sich genommen keine Neuheiten: sowohl die Panoramafotografie, als auch webbasierte Panorama- und 3D-Model-Viewer, sind Technologien, die seit vielen Jahren im Einsatz sind. Aber technische Entwicklungen der letzten Jahre, z.B. bei der Hardware der Endverbraucher oder das Aufkommen von HTML5, ermöglichen die Darstellung immer komplexerer und rechenintensiverer Anwendungen. Zusätzlich tragen vergleichsweise neue Technologien wie Smartphones und VR-Brillen zur wachsenden Popularität und Zugänglichkeit von Panoramatauren bei.

Allerdings ist die manuelle Erstellung von Depthmaps mit einer 3D-Software — zur Erweiterung von Panoramatauren mit Tiefeninformationen — arbeitsaufwändig und fehleranfällig. Zur Lösung dieses Problems gibt es verschiedene Ansätze, welche in diesem Kapitel aufgezeigt werden sollen. Dazu werden sowohl die aktuellen Entwicklungen von Hardware-basierten Ansätzen, die in Form von kombinierten Systemen aus Kameras und Laserscannern auftreten, als auch Forschungen im Bereich der Bildverarbeitung (bzw. Computer Vision) zur Konstruktion von Tiefeninformationen aus (Panorama-)Bildern heraus, dargestellt. Weiterhin sollen einige Forschungen aus der Peripherie des Themas der Erweiterung von Panoramatauren mit Tiefeninformationen, vorgestellt werden.

3.1 Erweiterung von Panoramabildern durch 3D-Laserscan-Daten

Da sich diese Arbeit mit einem Anwendungsorientierten Problem, nämlich dem Vergleich von verschiedenen Möglichkeiten der Verwendung von Tiefeninformationen mit der Software `krpano`, beschäftigt, sollen in diesem Abschnitt andere aktuelle Möglichkeiten zur Erstellung



Abbildung 3.1: Callum Prentice's *Street Cloud Simple*¹ stellt extrahierte Lidar-Scan-Daten aus Street View in einer 3D-Ansicht dar.

von Panoramatauren mit Tiefeninformationen aufgezeigt werden. Da die manuelle Erstellung von 3D-Modellen für die Verwendung als Depthmaps schwierig und zeitaufwändig ist, wurden Verfahren entwickelt, um die Gewinnung von zu Bilddaten korrespondierenden Tiefeninformationen automatisiert zu ermöglichen. Dabei besteht eine technische Möglichkeit zur Gewinnung von Tiefeninformationen unter anderem in der Verwendung von Laserscannern mit der Lidar-Methode. Solche Systeme werden beispielsweise von Google verwendet, um die Panoramabilder ihres Produkts *STREET VIEW* mit 3D-Geometrie zu ergänzen. Da die Bilder dabei mit auf Autos montierten Kameras während der Fahrt aufgenommen werden, und somit bestmöglich zeitgleich gemacht werden müssen, verwendet Google gleich neun Kameras und zusätzlich einen Lidar-Scanner zum Messen der Tiefendaten [Esc19]. Die Tiefendaten bestehen dabei aus Punktwolken und werden schließlich dazu verwendet, um bei *STREET VIEW* einen 3D-Übergangseffekt zwischen zwei Panoramabildern zu ermöglichen, sowie für die Darstellung eines dreidimensionalen Mauszeigers und beispielsweise Markierungen zu Geschäften. Die 3D-Daten werden dabei stark vereinfacht, um die Ladezeiten beim Download zu minimieren [ADF⁺10]. Wie in Abbildung 3.1 ersichtlich wird, sind die in diesem Kontext von *STREET VIEW* verwendeten 3D-Daten aber eher grob und bilden die umliegende Geometrie nur rudimentär ab.

Neben diesen von Google für die Betrachtung durch einen Endnutzer bereitgestellten Produkt, gibt es auch Hersteller von kombinierten Kamera-Laserscanner Systemen, die für die selbstständige Anwendung in Innenräumen gedacht sind. So werden von Firmen wie Matterport² oder HottScan³ komplette Systeme inklusive Software angeboten, wobei die Firmen damit werben, dass es teilweise selbst für Laien möglich sein soll Panoramatauren mit korrespondierenden 3D-Modellen zu erstellen. Dabei gibt es bei den beiden genannten Beispielen im Ansatz jedoch leichte Unterschiede, da Matterport auf das vollautomatisierte Erstellen

¹https://callumprentice.github.io/apps/street_cloud_simple/

²<https://matterport.com/de>

³<https://hottscan.de/>

3.1. Erweiterung von Panoramabildern durch 3D-Laserscan-Daten

von texturierten 3D-Modellen setzt, während HottScan eher die Möglichkeit dazu bereiten will, anhand der automatisiert-generierten Tiefeninformationen präzise 3D-Modelle nachzumodellieren. Die Systeme setzen dabei jedoch weiterhin auf eine Kombination der Tiefeninformationen mit 360°-Panoramabildern, denn auch bei diesen Systemen ist die Präzision der Laserscanner nicht sonderlich hoch. Wie in Abbildung 3.3 zu erkennen ist die Geometrie stark vereinfacht und stellt das Objekt nur sehr verzerrt dar, sodass die Verwendung der Tiefeninformationen als texturiertes 3D-Modell nur aus weit entfernten Ansichten und zur Ergänzung der Panoramabilder geeignet ist.



Abbildung 3.2: Beispiele für Produkte zum Erstellen von 360°Panoramen mit Tiefeninformationen durch Laserscan-Technologie. Links: Matterport Pro2, Mitte: iStar Pulsar+, Rechts: HottScan 3D-Raumscanner.



Abbildung 3.3: Beispiel einer Panoramatour von Matterport, mit einem texturierten 3D-Modell (oben) und der Sicht aus einem Panoramabild (unten).⁴

⁴<https://matterport.com/discover/space/v4LWLiLDm3s>

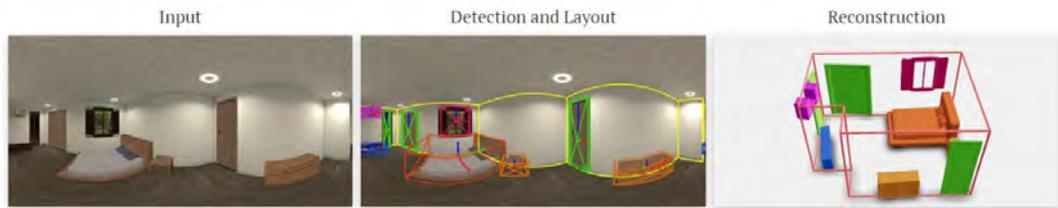


Abbildung 3.4: Rekonstruktion einer 3D-Szene aus Panoramabildern [ZCC⁺21].

Von Matterport gibt es zudem ein für Forschungszwecke erstelltes Datensatz namens *Habitat*, der in Zusammenarbeit mit Meta AI veröffentlicht wurde und aus 1000 texturierten 3D-Modellen besteht [RGW⁺21]. Ein weiterer solcher Datensatz heißt *Matterport3D* und besteht sowohl aus texturierten 3D-Modellen, als auch korrespondierenden Panoramabildern, Depthmap- und semantischen Bildern (farbliche Trennung einzelner Objekte) [CDF⁺17], wobei die Panoramabilder mit der Kamera Matterport Pro gemacht wurden.

3.2 Rekonstruktion von 3D-Geometrie aus Bildern

Ein weiterer Ansatz zur automatisierten Generierung von Tiefeninformationen, und somit der Rekonstruktion von 3D-Geometrie, besteht im Feld der Computer Vision durch die Anwendung verschiedener Algorithmen, sowie KI-basierter Methoden, auf Bilder, die teilweise nur Farbinformationen beinhalten. Die Aufgabe dieser Methoden ist es daher, die in den Bildern vorkommenden Formen, Farben und andere Merkmale zu analysieren und in 3D-Geometrie zu interpretieren (Image-Based-Modeling). Ein System zur Rekonstruktion von 3D-Geometrie stellen Cabral und Furukawa [CF14] vor, in welchem sie zunächst einen SfM-Algorithmus (Structure from Motion) auf mehrere Panoramabilder und einen MVS-Algorithmus (Multi-View Stereo) auf die ungestitchten Bilder der Panoramen anwenden. Aus den 3D-Daten wird eine Punktwolke mit den Klassen Boden, Decke und Wand zugeordneten Punkten erstellt und aus diesen Daten ein 2D-Flurplan ermittelt, woraus wiederum 3D-Wände extrudiert werden. Somit werden alle anderen Objekte im Raum vernachlässigt und nur die Wände, Boden und Decke werden als 3D-Objekt konstruiert. Dieses Verfahren geht dabei von orthogonalen Wänden und geraden Böden aus, wodurch es für Außenszenen ungeeignet ist.

Ein weiteres Verfahren, das aus Panoramabildern eine 3D-Geometrie rekonstruiert wird von Yang et al. [YJL⁺18] vorgeschlagen und ist nicht auf die Verwendung von überlappenden Bildern angewiesen. Das Verfahren untersucht Bilddaten auf semantische Hinweise, um ein Panoramabild in Vordergrund (Objekte) und Hintergrund (Layout) aufzuteilen und aus diesen Informationen eine Depth-Map zu gewinnen. Aufgrund der Rücksichtnahme auf Objekte in der Szene kann dieses Verfahren für Innenräume mit komplexer Geometrie zu guten Ergebnissen führen.

Zhang et al. [ZCC⁺21] schlagen ein Verfahren zur Rekonstruktion von 3D-Geometrie aus einem singulären Panoramabild vor. Dabei werden mehrere andere Methoden aus dem aktuellen Stand der Technik zur Gewinnung von geometrischen, semantischen und Merkmalen von Objekten kombiniert und mithilfe eines neuronalen Netzwerkes verfeinert (vgl.

Abbildung 3.4). Zudem werden die Beziehungen von Objekten zur Welt und zueinander ermittelt. Beispielsweise kann so eine Tür erkannt und in der Lage angepasst werden, da sie als Rotationskörper nicht unbedingt orthogonal zu einer Wand steht. Dieses Verfahren basiert, wie die vorher genannten, auch auf der „Manhattan World“ Annahme [CY99], die von einer Welt ausgeht, in der Wände und Objekte grob gesehen aufrecht und orthogonal zueinander stehen.

Eine Methode, die im Gegensatz zu den vorigen, auch zur Tiefenerfassung von Außenaufnahmen geeignet ist und ebenfalls Panoramabilder als Input verwendet, wird von Yan et al. vorgeschlagen, wobei zur Ermittlung der Tiefe gleich mehrere neuronale Netze verwendet werden, wobei eines auf die Maskierung des Himmels in Außenaufnahmen trainiert ist.

Neben diesen Forschungsergebnissen gibt es von Matterport bereits eine Anwendung für Mobilgeräte, die die Rekonstruktion von 3D-Geometrie aus Panoramabildern ermöglicht. Mit der App names MATTERPORT CAPTURE⁵ lassen sich mit Android- und iOS-Geräten Panoramatauren erstellen, die sich mithilfe des neuronalen Netzwerks CORTEX AI durch ein texturiertes 3D-Modell ergänzen lassen. Wie in Abbildung 3.5 zu sehen ist, ist die Auflösung und Komplexität des so erstellten 3D-Modells jedoch nicht sehr hoch und es weist einige Verzerrungen sowie Löcher auf. Die Qualität der Panoramabilder selbst ist dabei überraschend gut, vor allem unter Berücksichtigung der Tatsache, dass die Bilder aus der Hand mit einem Smartphone geschossen wurden, wo aufgrund der Verschiebung des Nodalpunkts mit vielen Stitchingfehlern zu rechnen wäre.



Abbildung 3.5: Screenshot einer Beispiel-Panoramatur von Matterport mit Panoramabildern, die aus der Hand mit einem Mobilgerät bei der Verwendung der App Matterport Capture entstanden sind.⁶

⁵<https://matterport.com/de/3d-camera-app>

⁶<https://matterport.com/discover/space/v4LWLiLDm3s>

3.3 Räumlicher 3D-Übergangseffekt als Video

In dieser Arbeit werden hauptsächlich Methoden zur Erweiterung von Panoramatauren durch das Hinzufügen von Tiefeninformationen untersucht. Einer der Beweggründe ist dabei das Problem des sprunghaften Übergangs zwischen zwei Panoramabildern zu lösen, indem stattdessen eine Bewegung — oder die Simulation einer Bewegung — erfolgt. Anstatt diese Bewegung nachträglich virtuell zu erstellen, lässt sie sich jedoch auch aus der Realität selbst aufzeichnen. Hierfür schlagen Tao und Wand [TW15] vor mit einer Kamera ein Video aufzunehmen, welches eben diese Bewegung zwischen zwei Standpunkten nachstellt. Zur Integration in einen Panorama-Viewer ist es hierbei notwendig, dass der Zustand des Panorama-Viewers (Kamerarichtung, Sichtfeld) beim Start des Übergangs in ein anderes Panoramabild mit dem Zustand des Videobeginns korrespondiert, sodass möglichst kein Unterschied zu erkennen ist. Um diesen Zustand zu erreichen, wird der Panorama-Viewer zunächst in die richtige Richtung und in das richtige Sichtfeld gebracht. Dann wird das Video eingeblendet, welches die Kamerafahrt zum nächsten Standpunkt darstellt, wo, wenn die Position erreicht und das Video zu Ende ist, wieder das Panoramabild eingeblendet wird. Bei idealer Umsetzung stellt diese Methode somit eine reibungslose Darstellung eines Übergangs zwischen Panoramen dar, allerdings nimmt sie dem Benutzer für die Zeit des Übergangs auch die Kontrolle über den Panorama-Viewer.

Die Autoren verwenden für ihre Beispiele Panoramabilder und Videos, die aus 3D-Szenen gerendert wurden, und somit ideal passgenau zugeordnet werden können. Allerdings könnte auch hier eine Diskrepanz zwischen Video und Panoramabild entstehen, wenn das Video ein anderes Bildverhältnis aufweist, als das Gerät, auf dem der Panorama-Viewer angezeigt wird. Bei der Verwendung von Kameras könnte es jedoch eine technische Herausforderung werden die verschiedenen Bilder in Einklang zu bringen. Es wäre daher zu untersuchen wie gut der Effekt auch bei einer nicht idealen Umsetzung wirkt. Alternativ könnten mit dem Aufkommen von Panorama-Videokameras Übergangsanimationen als interaktive 360-Grad-Videos erstellt werden. Dabei muss jedoch bedacht werden, dass 360-Grad-Videos im Internet noch nicht flächendeckend in der gleichen Auflösung dargestellt werden können, wie 360-Grad-Panoramabilder.

3.4 Zusammenfassung

Es wird viel Forschung im Bereich der sogenannten Computer Vision betrieben, um Verfahren zu entwickeln, mit denen sich 3D-Szenen aus Bildern und rekonstruieren lassen. Die Ergebnisse sind bereits sehr beeindruckend, so lassen sich teilweise aus einzelnen Panoramabildern unter Verwendung von Algorithmen und neuronalen Netzwerken komplexe und genaue 3D-Modelle rekonstruieren. Zusätzlich gibt es bereits Verbraucherprodukte, die das Erstellen von Panoramatauren mit texturierten 3D-Modellen ermöglichen, wobei sowohl spezielle Hardwarelösungen, als auch Softwarelösungen in Form von mobilen Anwendungen existieren. Die Qualität der resultierenden 3D-Modelle ist dabei jedoch (noch) nicht sehr hoch, aber sie können eventuell als gute Grundlage zur manuellen Erstellung einer sauberen Geometrie dienen.

Kapitel 4

Methodik

In Kapitel 2 wurden die technischen Voraussetzungen für das Erstellen von virtuellen Panoramatauren mit Tiefeninformationen erläutert. In diesem Kapitel soll nun die Methodik erläutert werden, welche verwendet wurde, um quantitative und qualitative Vergleiche der beiden Implementationsvarianten, zur Verwendung von Depthmaps in `KRPANO`, zu ermöglichen. Die Methodik unterteilt sich dabei in vier Teilschritte:

1. Aufstellung von Vergleichskriterien (4.1)
2. Konzeption von Testumgebungen (4.2)
3. Erstellung der Testumgebungen und Datenerfassung (4.3)
4. Datenanalyse (Kapitel 5)

Diese Teilschritte werden in den folgenden Abschnitten näher erläutert.

4.1 Aufstellung von Vergleichskriterien

In diesem Teilschritt der Methodik werden Vergleichskriterien erstellt, mit deren Hilfe die beiden Implementationsvarianten auf ihre Vor- und Nachteile hin untersucht werden sollen. Dabei ist die Auswahl dieser Kriterien wegweisend für die Richtung, die diese Arbeit nimmt. In diesem Fall sollen insbesondere Kriterien ausgewählt werden, die für konkrete Anwendungszwecke relevant sind und somit zum Beispiel auch Voraussetzungen berücksichtigen, die durch mögliche Limitierungen aufseiten der Hardware von Endnutzern entstehen können. Zusätzlich wird von der Nutzung von virtuellen Touren im Kontext des Internets ausgegangen, weshalb Aspekte wie die Größe von Dateien, welche erst von einem Server heruntergeladen werden müssen, ebenfalls berücksichtigt werden. Diese technischen Limitationen setzen den Rahmen, in welchem die Depthmaps schließlich in Bezug auf ihre Qualität hin verglichen werden sollen. Um empirische Daten für solch einen Vergleich zu ermitteln, sollen schließlich Testumgebungen erstellt werden, die sich auf die Vergleichskriterien beziehen.

Daraus schließend soll der Vergleich zum großen Teil anhand folgender vier Kriterien vollzogen werden:

- Speicherbedarf und Ladezeiten der bei den verschiedenen Methoden verwendeten Depthmaps (4.1)
- Performance bei der Verwendung von Depthmaps (4.1)
- Präzision der Projektion des Panoramabildes auf die Geometrie der Depthmaps und Präzision der Geometrie an sich (4.1)
- Ästhetik (4.1)

Diese Kriterien sollen folgend im Einzelnen näher erläutert werden.

Vergleichskriterium: Speicherbedarf und Ladezeiten

Die Minimierung des Speicheraufwandes ist für eine flüssige Darstellung einer virtuellen Panoramatur essentiell, da Bilder und Depthmaps im Normalfall erst von einem Server heruntergeladen werden müssen, bevor sie angezeigt werden können. Große Dateien können beispielsweise das Laden des nächsten Panoramabildes verzögern und somit eine potentiell immersions-brechende Verzögerung (kurzer Stillstand in der Bewegung des Bildes) verursachen.

Bei der Verwendung einer Webanwendung besteht außerdem das Risiko von weiteren Verzögerungen aufgrund einer langsamen Internetverbindung.

Vergleichskriterium: Bildfrequenz

Ein weiterer Faktor für das Entstehen von Verzögerungen im Bild, welche zu einer nicht-flüssigen Wahrnehmung beim Anwender führen können, ist die Verwendung rechenintensiver Assets, wie z.B. sehr hochauflösender Bilder oder komplexer Depthmaps. Hierbei soll untersucht werden, inwiefern sich die Komplexität der Depthmaps (Polygonanzahl bei 3D-Modellen, Auflösung und Bittiefe bei Depthmap-Bildern), auf die Bildfrequenz auswirkt. Dabei soll auch die Leistungsstärke des Endgeräts als Faktor berücksichtigt werden.

Vergleichskriterium: Präzision der Projektion und Geometrie

Ein weiteres Kriterium, nach dem die Qualität der verschiedenen Depthmaps verglichen werden soll, ist die Präzision der Projektion des Panoramabildes auf die Geometrie der Depthmap. Dabei stellt sich vor allem die Frage, wie — beziehungsweise wie gut — die

Kanten eines Objekts im Bild auf die entsprechenden Kanten in der Geometrie projiziert werden.¹

Hier spielt neben der Projektionspräzision auch die Präzision der Geometrie der Depthmaps an sich eine Rolle. Die in *KRPANO* verwendeten Depthmaps bestehen entweder aus Depthmap-Bildern oder aus 3D-Modellen, die in verschiedenen Formaten vorliegen können. Es soll untersucht werden, ob und wie diese verschiedenen Formate die Präzision der Geometrie verändern.

Vergleichskriterium: Ästhetik

Als Letztes sollen auch allgemein ästhetische Aspekte bei der Verwendung verschiedener Depthmaps in *KRPANO* bewertet werden. Diese Bewertungen sollen sich auf der ästhetischen Ebene mit den Stärken und Schwächen der verschiedenen Varianten befassen und sie für unterschiedliche Szenarien abwägen. Zusätzlich sollen die visuellen Unterschiede bei der Verwendung von einfachen, detailarmen Modellen und komplexen Modellen herausgearbeitet werden, sowie die Unterschiede bei der Verwendung von Depthmaps für Innen- und Außenaufnahmen.

Weitere Vergleichskriterien

Zusätzlich zu den oben genannten Vergleichskriterien sollen noch allgemeinere Facetten der Integration von Depthmaps in *KRPANO* verglichen werden. Dazu sollen anhand der Erfahrungen, die beim Erstellen von Testumgebungen gesammelt wurden, die Aspekte *Arbeitsaufwand* und *Integration von Hotspots* bewertet und verglichen werden.

4.2 Konzeption von Testumgebungen

Dieser Abschnitt befasst sich mit allgemeinen theoretischen Überlegungen zur Konzeption von Testumgebungen, mithilfe derer die Erfassung von Daten, zum Vergleich der verschiedenen Depthmap-Varianten, ermöglicht werden soll. Die Testumgebungen sollen sich dabei an den praxisnahen, oben bereits beschriebenen, technischen Limitierungen orientieren und dabei gleichzeitig die Grenzen innerhalb dieses Rahmens austesten.

Um dies zu erreichen, ist es unabdingbar, dass die Testumgebungen selbst webbasierte, mit *KRPANO* erstellte Panoramatauren sind. Zudem müssen sie repräsentativ für reale Anwendungszwecke sein, weshalb sie auch immer mindestens ein Panoramabild beinhalten. Dies muss insbesondere bei der Messung der Performance berücksichtigt werden.

Für die Evaluierung der Bildqualität ist es ein klarer Vorteil, wenn Faktoren ausgeschlossen werden können, die aus der Ungenauigkeit menschlichen Arbeitens heraus entstehen können. So ist es hier konkret der Fall, dass das manuelle Erstellen von Depthmaps —

¹Dieses Kriterium steht in einem reziproken Verhältnis zu den beiden vorigen Kriterien, da es in Richtung der Maximierung der Bildqualität zeigt, während die oberen beiden Kriterien das Ziel verfolgen eine möglichst gute Performance zu ermöglichen. Dennoch haben alle Kriterien auch ästhetische Aspekte, denn die Bildfrequenz und die Verhinderung von Verzögerungen sind ebenso wichtig für das ästhetische Erfahren wie die Bildqualität.

vor allem bei *fotografierten* Panoramabildern — bezogen auf die Genauigkeit eine gewisse Fehleranfälligkeit aufweist.

Denn sowohl die Ausrichtung des Bildes ist meist nicht perfekt (selbst geringe Abweichungen von 3 bis 4 Pixeln können hier auffällig sein), aber auch die reale Welt an sich weist meist keine perfekten Geraden Linien oder perfekte 90° Winkel etc. auf, sodass selbst Depthmaps, die mit eigens dafür gemachten Tools erstellt wurden, oft auffällige Fehler aufweisen.²

Um die Fehler, die durch das manuelle Erstellen von Depthmaps entstehen können, als Faktor auszuklammern, sollen hier für die Testumgebungen Panoramabilder verwendet werden, die aus 3D-Szenen heraus gerendert wurden. Denn im Vergleich zum Fotografieren hat das Rendern von Panoramabildern den Vorteil, dass die zum Rendern der Bilder genutzte Geometrie ebenfalls zum Erstellen der Depthmaps genutzt werden kann. Somit können Panoramabilder und Depthmaps erstellt werden, die exakt aufeinander passen.

Unter Berücksichtigung dieser Aspekte wurden die folgenden drei Testumgebungen konzipiert:

- **Testumgebung I (Performance):** Diese Testumgebung soll die Möglichkeit zur Messung der Bildfrequenz bei der Verwendung von unterschiedlichen Depthmaps schaffen. Dazu soll eine Funktion entwickelt werden, die die Nutzung der Panoramatour durch eine Bewegung und Drehung im 3D-Raum simuliert und für alle Depthmaps wiederholt werden kann. Die für die Messung der Bildfrequenz hergestellten Depthmaps können zusätzlich in Bezug auf ihren Speicherbedarf und ihre daraus resultierenden Ladezeiten hin untersucht werden. Die ästhetischen Aspekte können bei diesem Prototyp vernachlässigt werden.
- **Testumgebung II (Präzision):** Zur Ermittlung der Präzision soll eine Testumgebung erstellt werden, die primitive Geometrische Formen in unterschiedlicher Ausrichtung verwendet. Schließlich sollen Positionen ausgewählt werden, aus denen uniforme Screenshots angefertigt werden können, um einen visuellen Vergleich zu ermöglichen.
- **Testumgebung III (Ästhetik):** Zum Vergleich ästhetischer Aspekte bei der Verwendung von Depthmaps soll eine Panoramatour einer detailreichen, visuell-anspruchsvollen Szene angefertigt werden. Dafür sollen auch Depthmaps in der bestmöglichen Qualität hergestellt werden. Hierbei soll ebenfalls eine Untersuchung durch die Verwendung uniformer Screenshots stattfinden, wobei ein Fokus auf den Vergleich zwischen der in Abschnitt 2.5.3 beschriebenen, bei 3D-Modellen stattfindenden Projektionsverzerrung und der bei Depthmap-Bildern stattfindenden geometrischen Verzerrung, gelegt werden soll. Weiterhin sollen Depthmaps für Szenen aus Innen- und Außenaufnahmen erstellt und die Wirkung der verschiedenen Methoden verglichen werden.

Die Umsetzung der Testumgebungen nach der in diesem Abschnitt aufgestellten Konzeption, wird im folgenden Abschnitt dargestellt.

²Ein Beispiel dafür sind die Depthmap-Panoramatouren von Everpano (<https://everpano.com/demos>). Dabei muss natürlich berücksichtigt werden, dass viele Fehler durch das Weglassen von Details entstehen, um Arbeitszeit zu sparen.

4.3 Umsetzung

In diesem Abschnitt sollen die nächsten Schritte der Methodik (Erstellung der Testumgebungen, sowie Datenerfassung) dargestellt werden. Dabei wird der Produktionsprozess der Depthmaps an dieser Stelle ausführlich dokumentiert, um die Arbeitsschritte der Datenerhebung nachvollziehbar und überprüfbar zu machen. Dies ist wichtig, da es an einigen Stellen technische Herausforderungen gibt, bei deren Bewältigung es durch die Anwendung fehlerhafter Methoden zu schlechten Ergebnissen kommen kann.

Es werden folgend zunächst die gemeinsamen Aspekte bei der Erstellung aller drei Testumgebungen dargestellt. Daraufhin werden die speziellen Aspekte der einzelnen Testumgebungen aufgezeigt, sowie die Datenerfassung erläutert.

4.3.1 Erstellung der Testumgebungen

Die Erstellung aller Testumgebungen beinhaltet folgenden Ablauf:

1. Das Erstellen einer 3D-Szene mit 3D-Geometrie und zusätzlich, je nach Bedarf, UV-Maps, Texturen, Lichter usw.
2. Das Rendern von Panoramabildern als *Cube-Faces* (einzelne Bilder einer Cube-Map) aus der 3D-Szene mithilfe der UNREAL ENGINE 4.
3. Das Rendern von 32-bit Z-Buffer Panoramabildern als Cube-Faces.
4. Das Zusammenfügen der Cube-Face-Bilder zu 360° Panoramabildern mit PTGUI.
5. Das Konvertieren der Z-Buffer-Bilder in KRPANO-kompatible Formate mithilfe der Software IMAGEMAGICK.³
6. Das Exportieren/Konvertieren der 3D-Geometrie als OBJ-, STL- oder DEPTH-Datei.⁴
7. Das Komprimieren der 3D-Geometrie durch das KRPANO „Encrypt Tool“.⁵
8. Das Einpflegen der Bilder und Depthmaps in eine KRPANO-Panoramatour.
9. Das Erstellen der Funktionalität zur Erfassung der Testdaten.

Die Schritte 2 bis 4 sollen in den folgenden Abschnitten näher erläutert werden.

³<https://imagemagick.org/>

⁴Zum Konvertieren in das DEPTH-Format bietet krpano ein Webtool an: <https://krpano.com/docu/depth-file-format/>

⁵Das Encrypt Tool ist Teil der krpano Tools, durch welches eine auf Text basierende Datei (z.B. auch .obj oder .stl Dateien), in eine, nur durch krpano lesbare, ASCII-Zeichenfolge konvertiert und komprimiert wird.

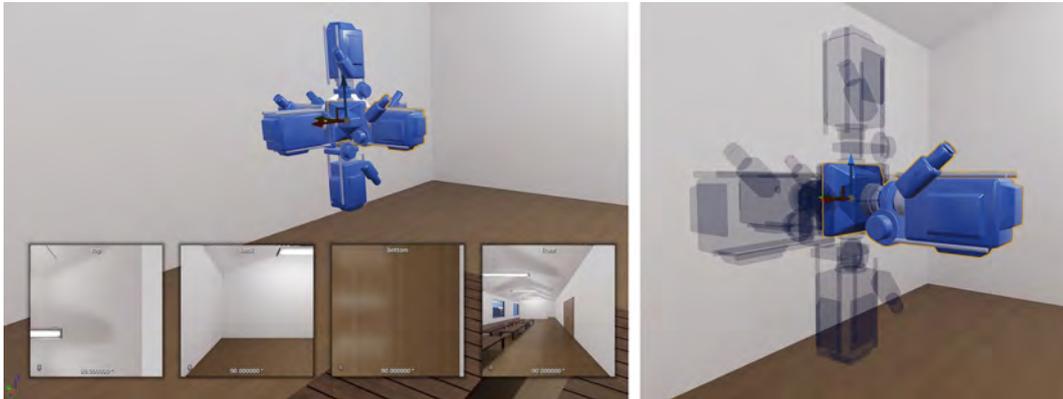


Abbildung 4.1: Sechs-Kamera-Setup zum Rendern einzelner Würfelflächen in der UNREAL ENGINE 4. Die virtuellen Kamerasensoren befinden sich alle im exakt selben Punkt im Raum und besitzen selbst keine Ausdehnung.

Rendern von Panoramabildern in Unreal Engine 4

Im Abschnitt *Cube-Maps* (Grundlagen 2.2.3) wurde erklärt, dass die Verwendung von Cube-Maps eine Möglichkeit der Speicherung von Panoramabildern darstellt. Die einzelnen Flächen einer Cube-Map können dabei auch als eine 90° Grad geradlinige Projektion aus der Mitte des Kugelpanoramas heraus, in jede Richtung, gedacht werden. Umgekehrt ist es daher möglich aus einer 3D-Szene heraus einzelne Flächen einer Cube-Map (Cube-Faces) zu rendern, indem mit einer virtuellen Kamera, mit einem Blickwinkel von 90° Grad, entsprechend den Cube-Faces in sechs verschiedenen Rotationen „fotografiert“ wird (vgl. Abbildung 4.1). Das direkte Rendern von sphärischen Panoramen ist in der UNREAL ENGINE 4 nur über Plugins von Drittanbietern möglich, welche allerdings nicht das Rendern von Depthmap-Bildern unterstützen, weshalb für diese Arbeit auf das Rendern von Cube-Maps zurückgegriffen wurde.

Für das Rendern der Bilder mit diesem Setup wird die *MovieRenderQueue* verwendet. Falls nicht anders angegeben wurde die Auflösung 8192 x 4096 Pixel verwendet, was für die quadratischen Cube-Faces eine Auflösung von 2048 x 2048 Pixeln ergibt.

Rendern von 32-Bit Z-Buffer-Bildern

Das Rendern der Depthmap-Bilder in der UNREAL ENGINE erfolgt mit dem gleichen Kamerasetup, das im vorigen Abschnitt beschrieben wurde. Die Depthmaps werden dabei aus exakt derselben Position wie die korrespondierenden Panoramabilder gerendert.

Um das finale Bild jedoch als Depthmap zu rendern, muss zunächst ein sogenanntes „Post-Process-Material“ erstellt werden. Die UNREAL ENGINE bietet zwar bereits ein solches Material an,⁶ jedoch erfüllt es nicht exakt den Zweck, für welchen es in diesem Fall benötigt wird. Die benötigte Funktionalität lässt sich aber aus diesem Material ableiten. Wie das daraus erstellte Material genau funktioniert, wird unter den Abbildungen 4.2 bis 4.4

⁶Das Material „*MovieRenderQueue.WorldDepth*“ ist Teil des *MovieRenderQueue* Plugins, welches standardmäßig installiert, aber nicht aktiviert ist.

erklärt. Als Resultat liefert dieses Material jedoch ein 32-Bit Z-Buffer-Bild mit Gleitkommazahlen zwischen 0 und 1. Der Wert 0 stellt dabei die kürzeste Distanz zur Kamera dar und der Wert 1 die Weitesten. Ab welcher Distanz ein Pixel den Wert 1 annimmt, wird im Blueprint des Materials festgelegt. Der optimale Wert liegt dafür bei der maximalen Distanz, die eine Kamera zu irgendeinem Punkt eines Meshes in der Szene haben kann.⁷

Die Bilder werden schließlich mit der MovieRenderQueue gerendert und im 32-Bit Format in einer EXR-Datei gespeichert.

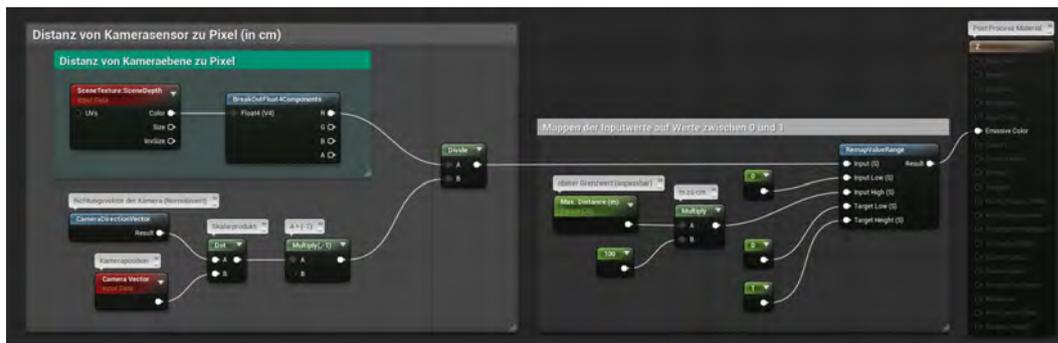


Abbildung 4.2: Blueprint des selbstgebauten Z-Depth Materials: Der Plan wird von links nach rechts gelesen. Die einzelnen Blöcke (Knoten) können dabei sowohl konstante Werte, als auch Funktionen, Texturen etc. darstellen, wobei die weißen Linien zwischen den Blöcken immer von einem Output zu einem Input führen. Schlussendlich führt diese Verkettung von Werten und Funktionen dazu, dass jedem Pixel im Bild ein Wert zwischen 0 und 1 zugeordnet wird, der im Resultatknoten (ganz rechts) als Emissionsfarbe eingespeist wird.

⁷Falls die Kamera nicht vollständig von Meshes umschlossen ist und es unendliche Distanzen gibt, wird die maximale Distanz durch diesen Wert festgelegt.

4. METHODIK

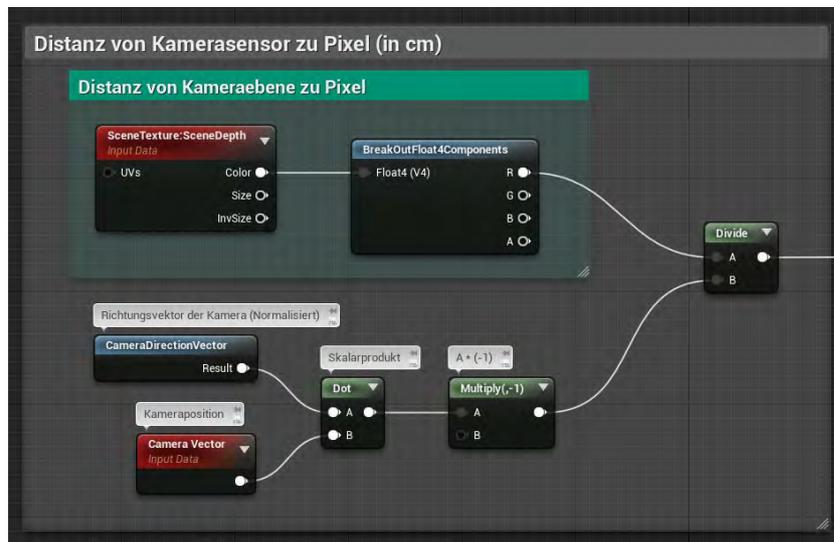


Abbildung 4.3: In diesem Teil des Blueprints wird oben zunächst der rote Kanal des SceneTexture-Knotens isoliert. Dieser beinhaltet die Distanz von der Kameraebene (der Richtungsvektor der Kamera ist hier der Normalvektor der Ebene) bis zum nächsten Mesh. Im unteren Teil wird das Skalarprodukt zwischen dem Kameravektor (Position) und dem Richtungsvektor der Kamera (Rotation) gebildet und anschließend mit -1 multipliziert. Der obere Teil wird anschließend durch den unteren Teil dividiert, sodass der Output-Wert nun die Distanz von der Kamera (Ursprung bzw. Sensor der Kamera) bis zum Mesh zeigt, welches vom jeweiligen Pixel abgedeckt wird.

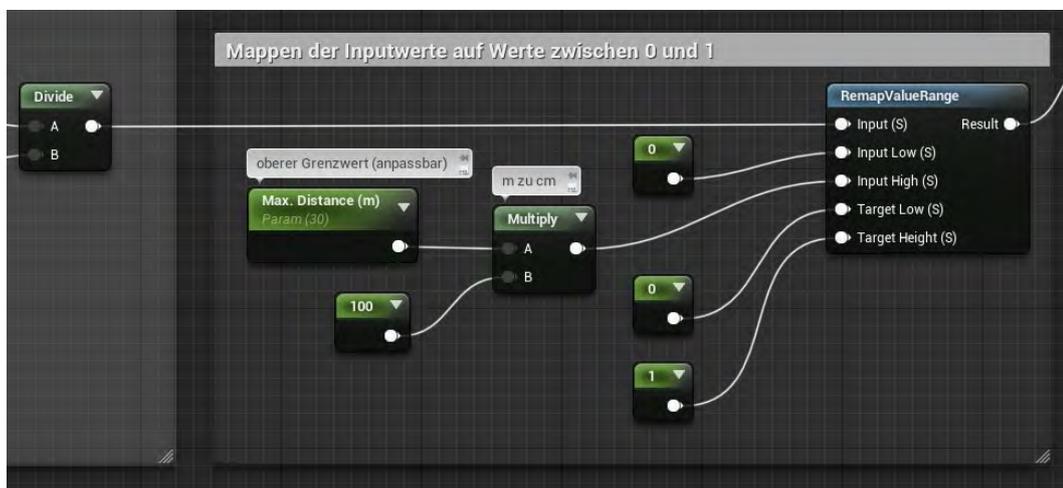


Abbildung 4.4: Der in Abbildung 4.3 ermittelte Quotient wird in diesem Teil des Blueprints im Knoten „RemapValueRange“ (Blau) auf Zahlenwerte zwischen 0 und 1 neu zugeordnet (Re-mapping). Dabei bleibt das Verhältnis (die relativen Abstände) jedoch erhalten. Zusätzlich bestimmt der Input *Input High (s)* den maximalen Wert, den eingehende Werte haben können, wodurch er sich anbietet, um eine maximale Distanz festzulegen, die die Kamera zu irgendeinem Punkt im Raum haben kann. Im Idealfall können so 2^{32} verschiedene Pixelwerte zwischen 0 und 1 als Distanz in der Depth Map dargestellt werden.

Zusammenfügen von Cube-Face-Bildern in PTGui

PTGui bietet die Möglichkeit zum Zusammenfügen einzelner Cube-Face-Bilder zu einem gesamten equirektangularen Panoramabild. Dazu müssen lediglich die sechs Bilder mit speziellen Suffixen⁸ versehen werden, sodass das Programm sie nach dem Import richtig zuordnen kann.

Die Bilder werden dabei *perfekt* zusammengesetzt; sie weisen keine sichtbaren Nähte auf und es gibt auch keine Überlappung der einzelnen Bilder, sodass auch keine Kontrollpunkte benötigt werden.

Konvertieren von Z-Depth Panoramabildern mit ImageMagick

In Abschnitt 2.5.3 wurden die verschiedenen Formate beschrieben, die für Depthmaps zum Gebrauch in KRPAHO verwendet werden können. Dabei stellten die 24-Bit Formate, die nach Stellenwerten in die drei 8-Bit RGB Kanäle kodiert werden, eine besondere Herausforderung dar, da diese nicht mit einem herkömmlichen Bildbearbeitungsprogramm wie ADOBE PHOTOSHOP erstellt werden konnten. Daher wurde für die Konvertierung auf die Software IMAGEMAGICK zurückgegriffen, welche das Bearbeiten von Bildern durch die Verwendung mathematischer Funktionen und benutzerdefinierter Algorithmen über die Kommandozeile ermöglicht.

Für das Umwandeln der 32-Bit Z-Buffer-Bilder in 24-Bit RGB-encodierte Depthmaps, wurde folgender Kommandozeilenbefehl angewandt:⁹

```

1  magick Panorama_32-Bit_Depth_Map.exr \
2  -channel red -fx '(int(u*(2^24)>>16)/255.0' \
3  -channel green -fx '(int(u*(2^24)>>8)&255)/255.0' \
4  -channel blue -fx '(int(u*(2^24))&255)/255.0' \
5  Panorama_24bit_rgb-encoded.png

```

Bei:

- channel folgende Argumente werden auf den angegebenen Kanal angewendet¹⁰
- fx '..' Verwendung eines mathematischen Ausdrucks
- int(..) Abrundung auf nächsten Integer
- >> bitweise Verschiebung nach rechts
- & binäres UND

Zur Verwendung der Depthmap als 24-Bit Bild mit genulltem Blaukanal musste die vierte Zeile wie folgt geändert werden:

```

4  -channel blue -fx '0'

```

⁸z.B. [name.front.jpg, name.left.jpg, name.right.jpg, name.back.jpg, name.top.jpg, name.bottom.jpg]

⁹Diese Codezeile ist eine Abwandlung eines im krpano-Forum veröffentlichten Codes (verfügbar unter: <https://krpano.com/forum/wbb/index.php?page=Thread&postID=86260#post86260>, zuletzt abgerufen am 19.5.22)

¹⁰Die Argumente des -fx Befehls aufgelistet: <https://imagemagick.org/script/fx.php#anatomy>

KRPANO verwendet Depthmaps mit invertierten Werten (weiß = null), weshalb die gerenderten Z-Buffer-Bilder hier zuerst invertiert werden mussten. Dies erfolgte mit dem Befehl `-negate`.

Falls das Z-Buffer-Bild beim Rendern noch nicht normalisiert (auf Zahlenwerte zwischen 0 und 1 gemappt) wurde, kann dies auch über den mathematischen Ausdruck `-fx 'u/maxima'` erfolgen. *maxima* ist hierbei der maximale Wert, den irgendein Pixel im Bild hat. Die Funktionen `-normalize` und `-auto-level` erfüllen ähnliche Funktionen, bei ihnen wird jedoch ein maximaler/minimaler Wert vorbestimmt. Dies führt zu Fehlern, da der minimale Wert der Depthmap äquivalent zur geringsten Distanz der Kamera zu irgendeinem Punkt eines Meshes sein muss.

Bei der Umwandlung einer Depthmap in ein 8-Bit Graustufen-Bild muss beachtet werden, dass nur ein Kanal (grau) verwendet werden darf. Die Umwandlung konnte hierbei mit dem Bildbearbeitungsprogramm ADOBE PHOTOSHOP erfolgen.¹¹

4.3.2 Testumgebung: Performance

Die Erstellung der Testumgebung, die in diesem Abschnitt dargestellt werden soll, erfolgte ebenfalls über die in Abschnitt 4.3 aufgelisteten Arbeitsschritte. Die Konzeption für diese Testumgebung wurde zudem bereits in Abschnitt 4.2 erläutert. Hier sollen Details der Umsetzung ergänzt und die für die Datenerfassung verwendeten Methoden erläutert werden. Diese Testumgebung dient der Erstellung von Messdaten zur Evaluation der Performance von Panoramatouren bei der Verwendung von Depthmaps. Dabei sollen Depthmaps verschiedenen Typs und von verschiedener Komplexität in Bezug auf die mit ihnen erreichbare Bildfrequenz hin untersucht werden. Dafür wurde folgende KRPANO-Action¹² geschrieben, die alle halbe Sekunde die durchschnittliche Bildfrequenz im Konsolenfenster des Browsers

¹¹<https://www.adobe.com/de/products/photoshop.html>

¹²Actions werden für das Programmieren eigener Funktionen verwendet. Dafür wird von KRPANO eine eigene Skriptsprache angeboten. (<https://krpano.com/docu/actions/>)

ausgibt:

```

1 <krpano>
2   <action name="fps_install">
3     set(view.hlookat, 0); set(view.tx, 0);
4     jscall(console.clear());
5     jscall(console.log("animation start"));
6     delayedcall(2,
7       setInterval(fps_plugin, 0.5,
8         if(view.hlookat LT 360.0,
9           fps_log(calc((display.currentfps+0.5) BOR 0));,
10          jscall(console.log("animation finished")););
11         exitcall();
12       );
13     );
14   );
15 </action>
16
17   <action name="fps_log" type="Javascript" ><![CDATA[
18     console.log(args[1]);
19   ]]></action>
20 </krpano>

```

Während diese *Action* läuft, bewegt sich die Kamera im Raum und dreht sich dabei langsam um 360° Grad. Die Action endet, wenn die Drehung um 360° Grad erreicht ist (Zeile 8 im Codeblock).

Dieses Experiment wurde auf drei verschiedenen Geräten (Smartphone, Tablet, Tower-PC) mit jeweils zehn verschiedenen, als Depthmap verwendeten 3D-Modellen (mit zwischen zweitausend und einer Million Polygonen), und den drei Varianten von Depthmap-Bildern bei vier unterschiedlichen Auflösungen, vollzogen. Die Messungen wurden alle mithilfe des `GOOGLE CHROME` Browsers ausgeführt, wobei Messungen beim Smartphone über die *remote-debugging*-Funktion¹³ des Browsers erfolgte. Bei dieser wird ein Mobilgerät per USB an einen PC angeschlossen, von welchem aus auf die Konsole zugegriffen werden kann.

Die für diese Testumgebung erstellten Depthmaps wurden schließlich zusätzlich verwendet, um Berechnungen über die Ladezeiten in Bezug auf mögliche Downloadgeschwindigkeit anzustellen. Dafür wurden diese drei, in Deutschland möglichen Übertragungsgeschwindig-

¹³<https://developer.chrome.com/docs/devtools/remote-debugging/>

keiten ausgewählt:

- 16 MBit/s (vergleichsweise langsamer Breitband-Anschluss)
- 75 MBit/s (Median der Breitband-Downloadgeschwindigkeit 2019/2020 in Deutschland)¹⁴
- 1000 MBit/s (eine mögliche 5G-Downloadgeschwindigkeit in 2022 in Deutschland)¹⁵

4.3.3 Testumgebung: Präzision

Die Testumgebung für den Vergleich der Präzision der Projektion und der Geometrie besteht aus dem Panoramabild einer virtuellen Szene, die einen großen Quader mit Schachbrettmuster als äußeren Raum verwendet. In dessen Inneren sind vier dreidimensionale Primitive aufgestellt (siehe Abbildung 4.5), die den Vergleich bei schrägen, geraden und runden Kanten, sowie der Genauigkeit der Geometrie mittels einer Wireframe-Textur ermöglichen. Das Schachbrettmuster des Quaders sorgt zudem dafür, dass sich gut Verzerrungen in der Projektion des Panoramabildes erkennen lassen.

Um die Depthmaps nachvollziehbar vergleichen zu können wurden konstante Kamerapositionen ausgewählt, aus denen Screenshots bei Verwendung verschiedener Depthmaps angefertigt wurden. Die Depthmaps sollen anhand der Screenshots danach bewertet werden, inwiefern die Geometrie der Depthmap und die Projektion des Panoramabildes auf der Depthmap der 3D-Szene aus dem 3D-Programm entspricht, mit welcher das Panoramabild gerendert wurde.

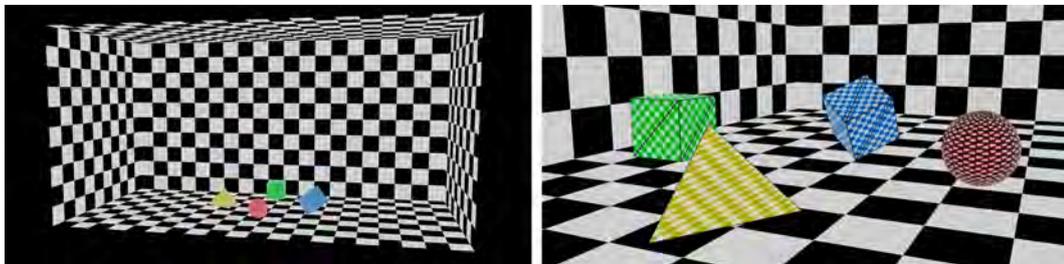


Abbildung 4.5: Testumgebung für den Vergleich der Präzision von Projektion und Geometrie.

4.3.4 Testumgebung: Ästhetik

Die Testumgebung zum Vergleich der Depthmaps in Bezug auf ihre Ästhetik wurde anhand mehrerer Szenarien vollzogen, wobei zwischen der Verwendung von Depthmaps mit gerenderten Panoramabildern und der Verwendung von Depthmaps mit fotografierten Panoramabildern unterschieden wird.

¹⁴<https://www.breitbandmessung.de/interaktive-darstellung>

¹⁵<https://www.5g-anbieter.info/speed/wie-schnell-ist-5g.html>

¹⁶Die Szene ist frei verfügbar unter:

<https://www.unrealengine.com/marketplace/en-US/product/archvis-interior-rendering>



Abbildung 4.6: Erstes Szenario für den Vergleich der Ästhetik. Die Szene entstammt einem Beispielprojekt für das hochqualitative Rendern von Innenarchitektur in der UNREAL ENGINE 4 und besitzt eine Polygonanzahl von ungefähr 5,5 Millionen Tris.¹⁶

4.3.5 Szenarien mit gerenderten Panoramabildern

Im ersten Szenario wurde die Verwendung von Depthmap Bildern in einer komplexen 3D-Szene mit einer hohen Anzahl von Polygonen getestet. Hierbei wurde die in Abbildung 4.6 abgebildete Szene genutzt. Da die Polygonanzahl der eigentlichen Szene jedoch zu hoch für die Verwendung in KRPANO war, mussten einige detailreiche Meshes für die Nutzung als 3D-Modell aus der Depthmap entfernt werden. Somit wurde jedoch gleichzeitig ein Fall geschaffen, der eher einem realen Anwendungsfall ähnelt, in dem Depthmaps für fotografierte Panoramabilder erstellt werden, da hier aus praktischen Gründen ebenfalls kleine Details vernachlässigt werden können. Es wurden in diesem Szenario also folgende Dinge verwendet:

- Depthmap-Bilder, die aus der vollständigen Szene heraus gerendert wurden und in verschiedenen Formaten vorliegen
- ein Depthmap 3D-Modell, welches jedoch für KRPANO in der Komplexität reduziert wurde, wobei die gerenderten Panoramabilder noch die fehlenden Meshes beinhalten

Auch hier wurden uniforme Screenshots aus identischen Positionen und Kamerarotationen erstellt, um den Vergleich zwischen den verschiedenen Depthmaps nachvollziehbar zu machen.



Abbildung 4.7: Szenario für den Vergleich der Ästhetik bei der Verwendung von Depthmaps mit einer einfachen, gerenderten Szene.

Dies gilt auch für das Szenario, welches in Abbildung 4.7 gezeigt wird. Diese Szene ist im Vergleich deutlich weniger komplex, insbesondere die Geometrie ist sehr simpel gehalten.

Dadurch konnte hier auch ein 3D-Modell mit der originalen Geometrie als Depthmap verwendet werden. In dieser Szene liegt bei der Untersuchung der Ästhetik das Augenmerk auf den Unterschied, den die Depthmaps bei der Darstellung der Bänke machen. Hier gibt es drei zu unterscheidende Fälle:

- Verzerrung der Geometrie durch die Verwendung von Depthmap-Bildern
- Verzerrung der Projektion durch das Fehlen der Geometrie der Bänke
- Verdoppelung und Verzerrung der Projektion durch das Fehlen von Bildinformationen hinter Objekten

4.3.6 Szenarien mit fotografierten Panoramabildern

Ein weiteres zum Vergleich der ästhetischen Eigenschaften verschiedener Depthmaps erstelltes Szenario, wird in Abbildung 4.8 abgebildet. Hierbei anhand eines fotografierten Panoramabildes mithilfe der 3D-Software AUTODESK MAYA manuell ein 3D-Modell modelliert, woraus wiederum ein Depthmap-Bild gerendert wurde. Das Panoramabild stellt dabei einen größtenteils rechteckigen Raum mit geraden Linien und aufrechten Wänden dar, was die Erstellung eines präzisen 3D-Modell ermöglichte.



Abbildung 4.8: Szenario zum Testen der Ästhetik bei der Verwendung von Depthmaps mit einer Indoor-Szene eines fotografierten Panoramabildes.

Für das letzte Szenario (vgl. Abbildung 4.9) wurden Depthmaps einer Außenaufnahme erstellt. Hierbei wurde wiederum anhand eines Panoramabildes ein 3D-Modell nachmodelliert, wobei größtenteils primitive Formen verwendet wurden, um Objekte wie Bäume und Autos nachzustellen. Folglich ist die Geometrie nur sehr grob und entspricht nur ungefähr der eigentlichen Form der abgebildeten Objekte.

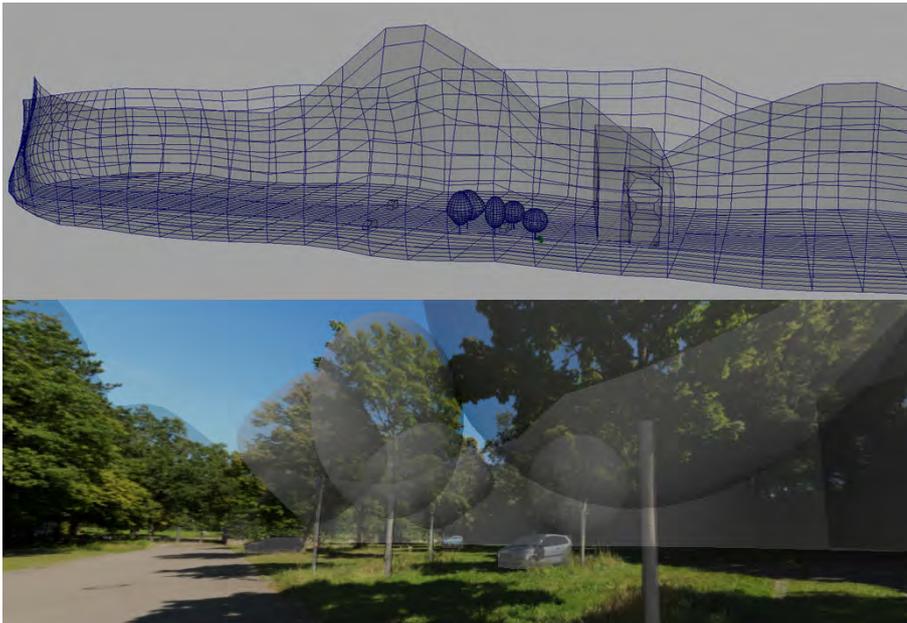


Abbildung 4.9: Szenario zum Testen der Ästhetik bei der Verwendung von Depthmaps mit einer Outdoor-Szene eines fotografierten Panoramabildes.¹⁷

Der letzte Teilschritt der Methodik besteht in der Analyse und Interpretation der erhobenen Daten und wird im nächsten Kapitel vollzogen.

¹⁷Kugelpanorama veröffentlicht von Andreas Mischok, frei verfügbar unter: https://polyhaven.com/a/park_parking

Kapitel 5

Datenanalyse

In diesem Kapitel sollen die, mithilfe der im vorigen Kapitel beschriebenen Methoden, erhobenen Daten dargestellt und analysiert werden. Die einzelnen Abschnitte laufen dabei Parallel zu den in 4.1 beschriebenen Vergleichskriterien.

5.1 Vergleich Speicherbedarf und Ladezeiten

Für die Testumgebung *Performance* (4.3.2) wurden Depthmap-3D-Modelle von verschiedener Komplexität in den Formaten STL, OBJ und DEPTH hergestellt und Depthmap-Bilder gerendert und in verschiedene Formate konvertiert. Um festzustellen, welches dieser Formate in Bezug auf den Speicherbedarf am effizientesten ist, wurden die Dateigrößen verglichen.

Speicherbedarf 3D-Modelle

Die in Tabelle 5.1 dargestellten Daten zeigen dabei eindeutig auf, dass das DEPTH-Format für die Verwendung von 3D-Objekten in Bezug auf den Speicherbedarf das Effizienteste der drei Formate ist.^{1 2} Die Verwendung des OBJ-Formats ist jedoch weiterhin relevant, da Dollhouses in diesem Format gespeichert werden müssen.

Speicherbedarf Depthmap-Bilder

In den Tabellen 5.2 und 5.3 werden die Dateigrößen von Depthmap-Bildern in Bezug auf ihre Auflösung gezeigt. Daraus wird ersichtlich, dass die Dateigröße ungefähr linear mit der Auflösung abnimmt und sich mit einer höheren Anzahl an verwendeten Bits vergrößert. Zudem spielt die Komplexität des Bildes eine maßgebliche Rolle für die Dateigröße. Erwähnenswert ist zudem, dass der Faktor, mit welchem die Komplexität des Bildes die Dateigröße beeinflusst, selbst sowohl abhängig von der Auflösung, als auch von der Kodierung des

¹Der Export der OBJ und STL geschah aus der Unreal Engine 4, die DEPTH-Dateien wurden von den STL-Dateien heraus konvertiert.

²in der Tabelle gibt es in jeder Spalte einen deutlichen *Sprung* zwischen zwei Werten, welcher durch eine Limitierung der den KRPANO-Encryption Tools entsteht, die nur Dateien unter einer Dateigröße von 10.000kb verarbeiten können.

Polygone (Tris)	Größe .obj (kb)	Größe .stl (kb)	Größe .DEPTH (kb)
1.932	80,11	53,66	23,99
4.016	150,55	137,73	48,85
8.044	279,26	302,65	93,50
16.002	535,73	619,34	182,48
32.026	953,75	1224,98	361,70
64.078	1961,64	11603,46	738,76
125.230	13475,90	22722,01	1468,54
250.196	26423,16	45320,72	2785,39
500.138	51657,51	91195,85	14769,25
1.015.554	106111,60	184111,33	30523,99

Tabelle 5.1: Dateigrößen von Depthmaps (3D-Modelle)

Bildes ist. Das heißt, ein komplexes Bild mit hoher Auflösung ist nicht nur größer als ein dementsprechendes Bild einer einfachen Szene, sondern es hat auch einen größeren Multiplikationsfaktor als ein entsprechendes Bild von niedrigerer Auflösung. Das Gleiche gilt für die Höhe der Verwendeten Bits. Dies lässt sich eventuell durch die beim Export verwendeten verlustfreien Kompressionsalgorithmen des PNG-Formats erklären.

Auflösung (px)	8-Bit Graustufen (kb)	16-Bit RGB-kodiert (kb)	24-Bit RGB-kodiert (kb)
8192x4096	531	1580	1855
4096x2048	211	1030	1381
2048x1024	85	572	836
1024x512	33	160	235

Tabelle 5.2: Dateigrößen von Depthmaps (Bilder) einer einfachen Szene

Auflösung (px)	8-Bit Graustufen (kb)	16-Bit RGB-kodiert (kb)	24-Bit RGB-kodiert (kb)
8192x4096	712	3999	5468
4096x2048	297	2109	3207
2048x1024	127	1045	1965
1024x512	43	257	451

Tabelle 5.3: Dateigrößen von Depthmaps (Bilder) einer komplexen Szene

Downloadzeiten 3D-Modelle

Für die Bewertung der Ladezeiten von 3D-Modellen wurde eine Grenze festgelegt, ab dieser eine Ladezeit als optimal gilt. Diese Grenze wurde auf 40ms gelegt, da dies der Wert ist, der bei einer Bildfrequenz von 25 Hertz das Zeitintervall zwischen zwei Bildern ausmacht. Somit könnte im Idealfall davon ausgegangen werden, dass keine Ladezeit, und somit keine Verzögerung im Bild, wahrgenommen werden kann.

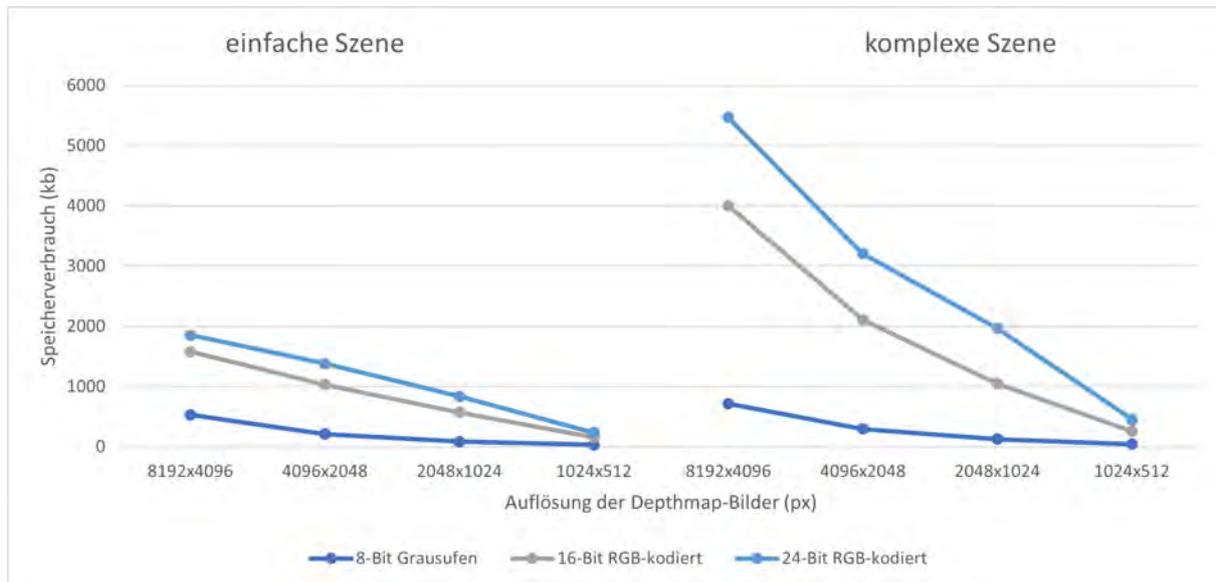


Abbildung 5.1: Diagramm zum Vergleich der Dateigrößen von Depthmap-Bildern einer einfachen und einer komplexen Szene.

Bei der Rechnung wurden dabei nur die Dateien des DEPTH-Formats berücksichtigt, da diese, wie oben festgestellt wurde, das optimale Format für Depthmaps in Bezug auf die Dateigröße sind. In der Tabelle 5.4 werden diese Berechnungen aufgezeigt, wobei alle unter dem Grenzwert liegenden Werte fett gedruckt sind. Hier fällt auf, dass der Download der zwei 3D-Modelle mit der geringsten Polygonanzahl, auch bei der hier langsamsten Downloadgeschwindigkeit, immer noch unter dem Grenzwert liegen. In diesem Zusammenhang sei auch noch einmal erwähnt, dass es beim manuellen Erstellen von Depthmaps einen sehr hohen Arbeitsaufwand bedeuten würde, überhaupt Modelle mit solch hohen Polygonanzahlen zu erstellen, sodass man dieses Ergebniss durchaus so bewerten kann, dass die Ladezeiten im Normalfall (und bei Nutzung von verschlüsselten DEPTH-Dateien) kein Problem darstellen sollten.

Downloadzeiten Depthmap-Bilder

Parallel zum vorigen Abschnitt wurde auch hier verfahren und die Downloadzeiten für Depthmap-Bilder errechnet. Hierbei fällt besonders auf, dass bei den langsamen Downloadgeschwindigkeiten fast keine Bilder die Anforderungen des Grenzwerts von 40ms erfüllen. Nur die Graustufenbilder von geringer Auflösung können demnach innerhalb dieser Zeit geladen werden. Aber auch bei einer Geschwindigkeit von 75MBit/s erfüllen die hochauflösenden Depthmaps meist nicht dieses Kriterium. Nur bei der schnellsten hier getesteten Downloadgeschwindigkeit können fast alle Bilder innerhalb dieses Zeitraums heruntergeladen werden.

Größe .DEPTH (kb)	t bei 16 Mbit/s (ms)	t bei 75 Mbit/s (ms)	t bei 1 Gbit/s (ms)
23,99	11,99	2,56	0,19
48,85	24,43	5,21	0,39
93,50	46,75	9,97	0,75
182,48	91,24	19,46	1,46
361,70	180,85	38,58	2,89
738,76	369,38	78,80	5,91
1468,54	734,27	156,64	11,75
2785,39	1.392,70	297,11	22,28
14769,25	7.384,63	1.575,39	118,15
30523,99	15.261,99	3.255,89	244,19

Tabelle 5.4: Downloadzeiten (t) von Depthmaps (3D-Modelle) bei verschiedenen Internetgeschwindigkeiten

Speicherbedarf und Downloadzeiten Zwischenergebnis

Für die hier festgelegte untere Grenze von 40ms legen die erhobenen Daten nahe, dass für Szenen mit geringer geometrischer Komplexität die Nutzung von 3D-Modellen als Depthmaps aufgrund der erheblich geringeren Dateigröße deutlich vorteilhafter ist. Allerdings lässt sich auch feststellen, dass es bei der Nutzung von 3D-Modellen mit einer Polygonanzahl von über 100000 Tris, ebenfalls zu Problemen mit den Downloadzeiten kommen kann. Ebenfalls sei hier angemerkt, dass der Speicherbedarf von Depthmap-Bildern bei detailreichen Szenen durchaus geringer sein kann als bei 3D-Modellen. Allerdings muss in diesem Fall auch beachtet werden, dass ein 3D-Modell für alle Panoramabilder zusammen benutzt werden kann, während ein Depthmap-Bild jeweils nur für ein Panoramabild Verwendung findet.

5.2 Bildfrequenz

In Abschnitt 4.3.2 wurde die Methodik erläutert, mit welcher Daten für den Vergleich von Depthmaps in Bezug auf ihre Bildfrequenz erhoben wurden. Die Datenerhebung geschah dabei mit den gleichen Depthmaps, die im vorherigen Abschnitt zum Vergleich des Speicheraufwands und der Ladezeiten verwendet wurden. Sie wurden für das Simulieren eines Ablaufs verwendet, in welchem die Bildfrequenz stetig gemessen wurde. Dieser Ablauf wurde mit drei verschiedenen Geräten vollzogen. Diese Geräte waren ein Tower-PC³, ein Tablet-PC⁴ und ein Smartphone.⁵

³Prozessor: AMD Ryzen 5 5600X 6-Core (3,7 GHz); Arbeitsspeicher: 32 GB; Grafikkarte: NVIDIA GeForce RTX 3060

⁴Microsoft Surface 3, Prozessor: Quad Core Intel Atom x7-Z8700-CPU (1,6GHz)

⁵Nokia 8, Prozessor: Qualcomm Snapdragon 835 (4*2,5 GHz, 4*1,8 GHz)

Auflösung in px	Downloadgeschwindigkeit t in ms		
	bei 16 MBit/s	bei 75 MBit/s	bei 1 GBit/s
8-Bit Graustufenbild			
8192x4096	265,50	56,64	4,25
4096x2048	105,50	22,51	1,69
2048x1024	42,50	9,07	0,68
1024x512	16,50	3,52	0,26
16-Bit RGB-kodiert			
8192x4096	790,00	168,53	12,64
4096x2048	515,00	109,87	8,24
2048x1024	286,00	61,01	4,58
1024x512	80,00	17,07	1,28
24-Bit RGB-kodiert			
8192x4096	927,5	197,87	14,84
4096x2048	690,5	147,31	11,05
2048x1024	418	89,17	6,69
1024x512	117,5	25,07	1,88

Tabelle 5.5: (Einfache Szene) Downloadzeiten von Depthmap-Bildern bei verschiedenen Internetgeschwindigkeiten

Auflösung in px	Downloadgeschwindigkeit t in ms		
	bei 16 MBit/s	bei 75 MBit/s	bei 1 GBit/s
8-Bit Graustufenbild			
8192x4096	356,00	75,95	5,70
4096x2048	148,50	31,68	2,38
2048x1024	63,50	13,55	1,02
1024x512	21,50	4,59	0,34
16-Bit RGB-kodiert			
8192x4096	1.999,50	426,56	31,99
4096x2048	1.054,50	224,96	16,87
2048x1024	522,50	111,47	8,36
1024x512	128,50	27,41	2,06
24-Bit RGB-kodiert			
8192x4096	2.734,00	583,25	43,74
4096x2048	1.603,50	342,08	25,66
2048x1024	982,50	209,60	15,72
1024x512	225,50	48,11	3,61

Tabelle 5.6: (Komplexe Szene) Downloadzeiten von Depthmap-Bildern bei verschiedenen Internetgeschwindigkeiten

Bildfrequenz 3D-Modelle

Die Daten von der Messung der Bildfrequenz bei der Verwendung von 3D-Modellen als Depthmap, befinden sich in Tabelle 5.7. Zur Veranschaulichung wurden zusätzlich die letzten fünf Zeilen der Tabelle als Diagramm in Abbildung 5.2 dargestellt. Daraus lässt sich erkennen, dass es erst bei der Verwendung von sehr detailreichen 3D-Modellen zu einem Abfall der Bildfrequenz unter die vom Bildschirm festgelegte maximale Frequenz kommt. Bei der Verwendung eines leistungsstarken Tower-PCs war überhaupt kein Leistungsabfall messbar, während beim Smartphone die Bildfrequenz erst ab 500.000 Tris unter 25fps fällt. Beim Tablet-PC war ein gravierender Leistungsabfall bei der Verwendung des 3D-Modells mit über einer Million Tris zu beobachten, wobei die Bildfrequenz auch hier sogar noch über 30fps liegt. Somit sollte der Performance-Einschnitt bei der Verwendung von 3D-Modellen als Depthmaps, in den allermeisten Anwendungsfällen nicht sehr beachtenswert zu sein. Allerdings sei hier noch angemerkt, dass die Auswahl der maximal-akzeptablen Bildfrequenz auch Geschmackssache ist. Vor allem bei interaktiven Anwendungen können daher auch höhere Bildfrequenzen wünschenswert sein.

Tris	fps PC	% von 144Hz	fps Smartphone	% von 30Hz	fps Tablet-PC	% von 60Hz
2.000	139,75	97,05%	29,66	98,88%	59,35	98,91%
4.000	144,00	100,00%	29,73	99,09%	59,48	99,13%
8.000	143,98	99,98%	29,65	98,84%	58,85	98,08%
16.000	143,93	99,95%	29,72	99,06%	59,51	99,18%
32.000	143,98	99,98%	29,68	98,95%	59,51	99,18%
64.000	143,98	99,98%	27,73	92,43%	59,80	99,67%
125.000	143,98	99,98%	26,43	88,12%	58,35	97,26%
250.000	143,96	99,97%	25,12	83,73%	56,22	93,70%
500.000	143,93	99,95%	22,29	74,31%	55,14	91,90%
1.000.000	143,98	99,98%	18,13	60,43%	33,13	55,22%

Tabelle 5.7: Bildfrequenzmessung von Depthmaps verschiedener Komplexität bei drei Geräten

Bildfrequenz Depthmap-Bilder

Die im vorigen Abschnitt beschriebenen Messungen wurden auf diese selbe Weise auch mit Depthmap-Bildern durchgeführt. Dabei wurden wiederum einige Variationen bei der Verwendung verschiedener Geräte verwendet. Die Bildfrequenz wurde dabei auch für den Tower-PC getestet, wobei jedoch kein signifikanter Verlust gemessen werden konnte (die Bildfrequenz lag für alle Depthmaps stabil bei ungefähr 144Hz). Die Testdaten für die anderen Geräte befinden sich in Tabelle 5.8. Die Bildfrequenz bei der Verwendung von Bildern mit der Auflösung 8192x4096px konnte dabei nicht gemessen werden, da der mobile CHROME-Browser die Bilder nicht verwenden konnte.⁶ Bei den Messungen mit dem Smartphone

⁶Es trat folgende Fehlermeldung auf: „WebGL: INVALID_VALUE: texImage2D: width or height out of range“

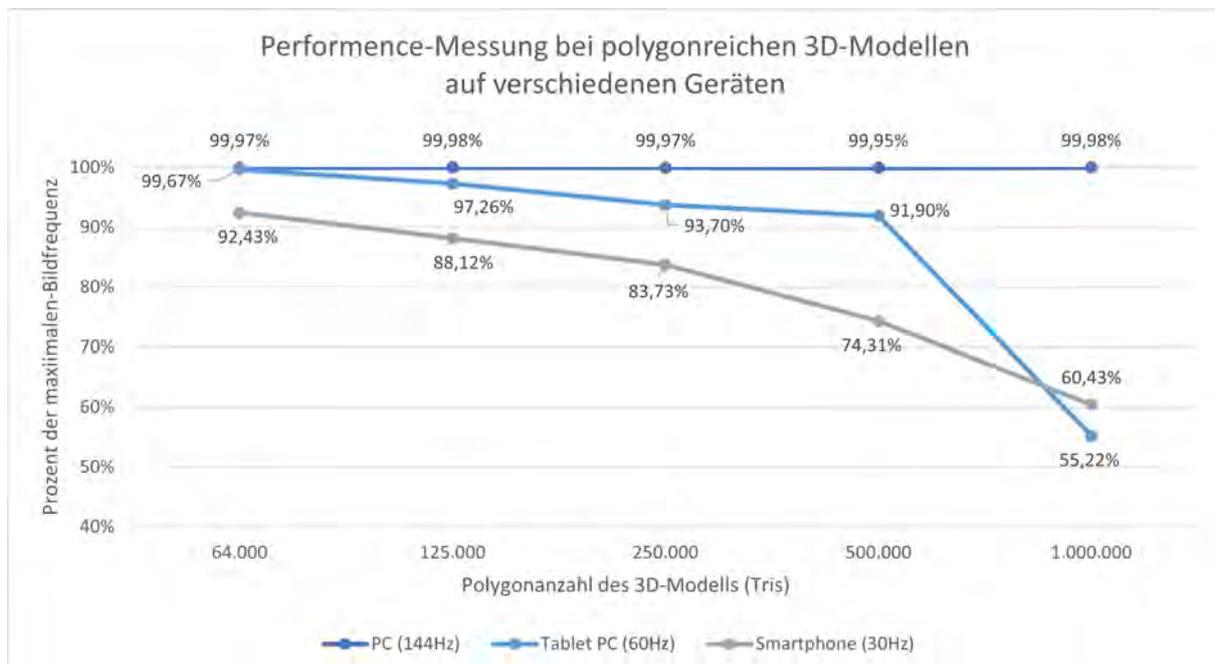


Abbildung 5.2: Diagramm zum Vergleich der Bildfrequenzen im Verhältnis zur Bildfrequenz des Monitors bei der Verwendung von Depthmaps mit hohen Polygonanzahlen.

fällt auf, dass die Werte alle sehr ähnlich sind und relativ nahe an der maximalen Bildfrequenz liegen. Die Verwendung verschiedener Depthmap-Typen und Auflösungen scheint hier demnach nicht ausschlaggebend zu sein. Beim Tablet PC finden sich jedoch einige Schwankungen, die sich nicht einfach erklären lassen. So sind beispielsweise bei den mit 16- und 24-Bit kodierten Bildern die vorletzte Reihe performanter als die letzte Reihe, die eine noch geringere Auflösung verwendet. Beim 8-Bit Graustufenbild ist dies umgekehrt, und der Wert erreicht sogar fast die maximale Bildfrequenz.

Desweiteren fällt auf, dass in zwei der drei Fälle die Bildfrequenz bei der höchsten Auflösung höher ist, als die bei der zweithöchsten. Somit lässt sich sagen, dass die Messungen in diesem Fall kein kongruentes Bild zur Bewertung der Depthmap-Bilder anhand ihrer Performance abgeben. Allerdings lässt sich für die Depthmap-Bilder allgemein feststellen, dass es bei allen Messungen zu akzeptablen Bildfrequenzen kommt. Das heißt die Verwendung von Depthmap-Bildern stellt in Bezug auf ihre Performance in der Regel kein Problem dar.

Performance Zwischenergebnis

Die Verwendung von Depthmaps jeder Art in krpano stellt in Bezug auf die Performance in der Regel kein Problem dar. Lediglich bei der Verwendung von sehr hochauflösenden Modellen kann es zu erheblichen Performanceeinbußen kommen. Dies ist aber kein üblicher Fall, da die Verwendung solcher 3D-Modelle auch aus anderen, bereits genannten Gründen, sehr unvorteilhaft ist.

Auflösung in px	Durchschnittliche Bildfrequenz (fps)			
	Smartphone	% (30Hz)	Tablet PC	% (60Hz)
8-Bit Graustufenbild				
8192x4096	ERROR	-	46,64	77,73%
4096x2048	27,00	90,00%	43,03	71,71%
2048x1024	27,51	91,71%	49,71	82,84%
1024x512	27,72	92,41%	59,03	98,39%
16-Bit RGB-kodiert				
8192x4096	ERROR	-	42,92	71,53%
4096x2048	26,37	87,89%	39,50	65,83%
2048x1024	27,19	90,63%	53,33	88,89%
1024x512	27,46	91,53%	42,26	70,44%
24-Bit RGB-kodiert				
8192x4096	ERROR	-	42,24	70,39%
4096x2048	27,43	91,44%	44,22	73,70%
2048x1024	26,96	89,86%	58,10	96,83%
1024x512	27,55	91,85%	43,39	72,32%

Tabelle 5.8: Durchschnittlich Bildfrequenz bei der Verwendung von Depthmap-Bildern mit verschiedenen Geräten.

5.3 Präzision

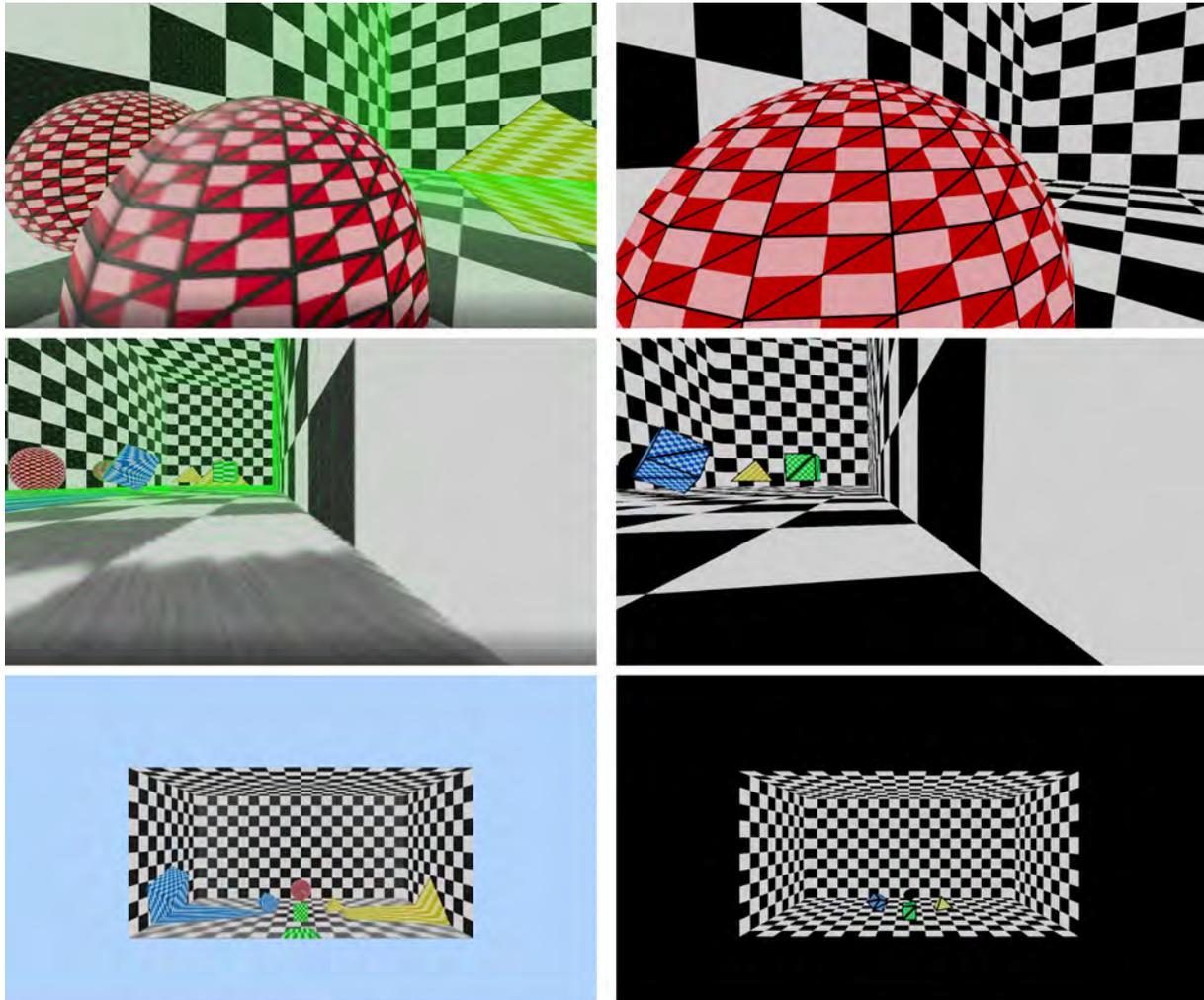
Für die Untersuchung der Präzision der Projektion des Panoramabildes auf die Geometrie einer Depthmap, sowie der Präzision der Geometrie an sich, wurde die in 4.3.3 vorgestellte Testumgebung verwendet. Die daraus entstandenen Screenshots sollen hier aufgezeigt und analysiert werden.

Präzision 3D-Modelle

In Abbildung 5.3 sieht man den Vergleich einer Depthmap mit der Szene, aus der diese Depthmap erstellt wurde. Die Depthmap ist dabei die 3D-Geometrie der Szene, die rechts im Bild durch Screenshots aus der 3D-Software AUTODESK MAYA dargestellt wird. Der Grünstich in den Screenshots der Depthmaps rührt von der Darstellung des Wireframes des Modells her (je dichter das Mesh, desto grüner ist das Bild an dieser Stelle).

Aus den Screenshots wird ersichtlich, dass zwar die Geometrie der beiden 3D-Modelle identisch ist, aber die Projektion an einigen Stellen auffällige Ungenauigkeiten aufweist. So verwischt sich das Bild vor allem an Stellen, wo weniger Pixelinformationen des Panoramabildes vorhanden sind. Dies ist besonders am linken Rand der Kugel zu beobachten, wo die Farbe zudem etwas ausbleicht. Im mittleren, linken Bild ist so ein Effekt ebenfalls zu beobachten. Hier ist die Kamera, aus der der Screenshot aufgenommen wurde, lediglich weit weg von dem Ausgangspunkt des Panoramabildes. Dies führt allerdings ebenfalls dazu, dass die Pixeldichte bei solchen Teilen der Geometrie deutlich niedriger ist, als in Bereichen, die näher

am Projektionszentrum sind. Dabei sei erwähnt, dass dies nur so auffällig ist, da die Kamera diese Bereiche aus einer sehr kurzen Distanz erfasst. Aus einer größeren Distanz heraus betrachtet wirkt das Schachbrettmuster des Quaders beispielsweise noch verhältnismäßig scharf.



3D-Modell als Depthmap in krpano

3D-Modell in Maya

Abbildung 5.3: Vergleich der Projektionsgenauigkeit und der Genauigkeit der Geometrie bei der Verwendung eines 3D-Modells als Depthmap und in der ursprünglichen Szene in der 3D-Software AUTODESK MAYA.

Präzision Depthmap-Bilder

Die nächste Abbildung (5.4) zeigt die Verwendung von Depthmap-Bildern im Vergleich miteinander. Die Kameraposition ist dabei identisch mit den ersten Screenshots der vorherigen

Abbildung. Durch diese Gegenüberstellung der Depthmap-Bilder werden mehrere Dinge ersichtlich:

- Der Unterschied zwischen 24-Bit und 16-Bit enkodierten Depthmap-Bildern ist in diesem Fall fast unbemerkbar.
- Die Präzision der Geometrie nimmt bei den ersten beiden Reihen mit der Auflösung erheblich ab, wobei zusätzlich kleine stachelartige Formen entstehen.
- Die Präzision der Geometrie bei Verwendung von 8-Bit-Graustufenbildern ist sehr limitiert, dafür ist der Qualitätsverlust bei einer geringeren Auflösung nicht so gravierend.
- Die Verwendung von Depthmap-Bildern ist nicht dazu geeignet Formen wie Kugeln darzustellen, da die Rückseite keine Tiefeninformationen enthält und somit nicht dargestellt werden kann.

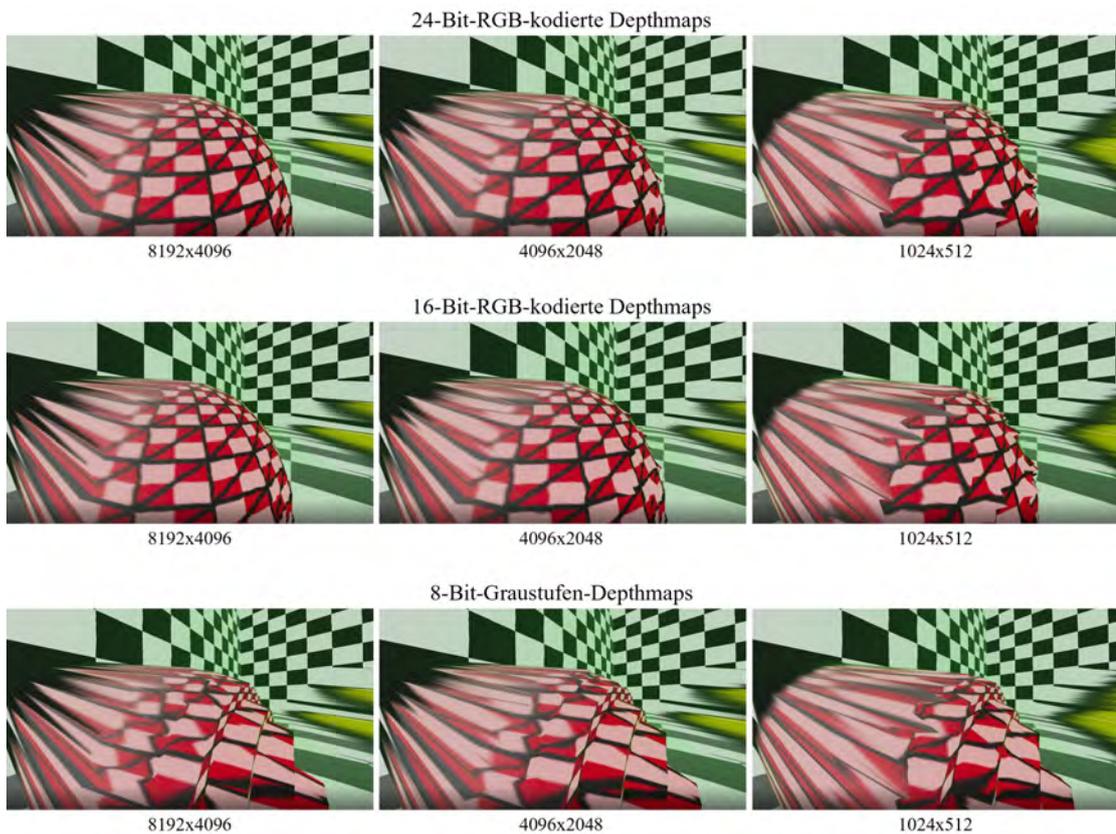


Abbildung 5.4: Vergleich der Genauigkeit der Geometrie bei der Verwendung von Depthmap-Bildern. (Position 1)

Aus der Abbildung 5.6 wird zudem ersichtlich, dass die Auflösung für weit vom Projektionszentrum entfernte Bereiche kein großer Faktor zu sein scheint. Die Screenshots der ersten

beiden Reihen aus dieser Position ähneln sich alle sehr und weisen eine starke Verzerrung der Projektion auf. Die Screenshots der 8-Bit-Graustufen-Depthmaps haben rechts im Bild eine veränderte Geometrie und die Verzerrung der Projektion scheint hier nicht so stark zu sein. Dies lässt sich allerdings durch einen Blick auf die Abbildung 5.5 erklären. Dort sieht man nämlich, dass sich die Geometrie selbst stark verzerrt und nur aufgrund dessen die Projektion nicht so stark verzerrt werden muss.

Besonders hier fällt auch auf, dass die Präzision im Vergleich zum 3D-Modell aus Abbildung 5.3, sowohl in Bezug auf die Projektion, aber auch in Bezug auf die Geometrie, nicht so hoch ist.

Präzision Zwischenergebnis

Die Depthmap-Bilder weisen im Vergleich zu, als Depthmap verwendeten 3D-Modellen, klare Defizite in der Präzision der Geometrie auf. Besonders auffällig ist dies bei eigentlich geraden Flächen, die bei der Verwendung von Depthmap-Bildern jedoch immer Unebenheiten aufweisen. Aber auch in Bezug auf die Präzision der Projektion lassen sich bei den Depthmap-Bildern stärkere Verzerrungen, vor allem in weiter vom Projektionszentrum entfernten Bereichen, feststellen. Zudem variieren Depthmap-Bilder bei der Verwendung unterschiedlicher Auflösungen und Bittiefen, eine gute Präzision wird dabei nur bei der Auflösung 8192x4096 Pixeln und einer Bittiefe von mindestens 16-Bit erreicht.

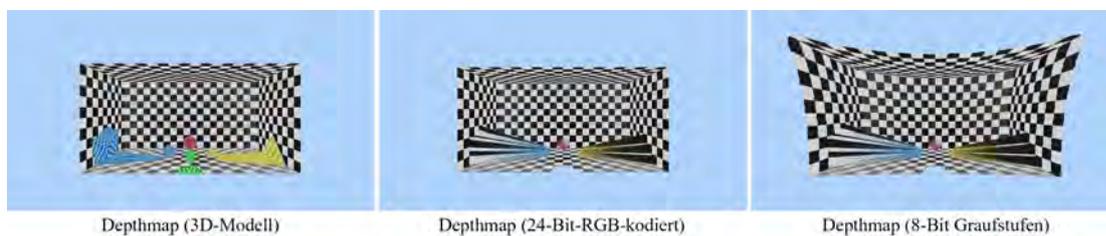


Abbildung 5.5: Vergleich der Genauigkeit der Geometrie bei der Verwendung von Depthmap-Bildern. (Position 3)

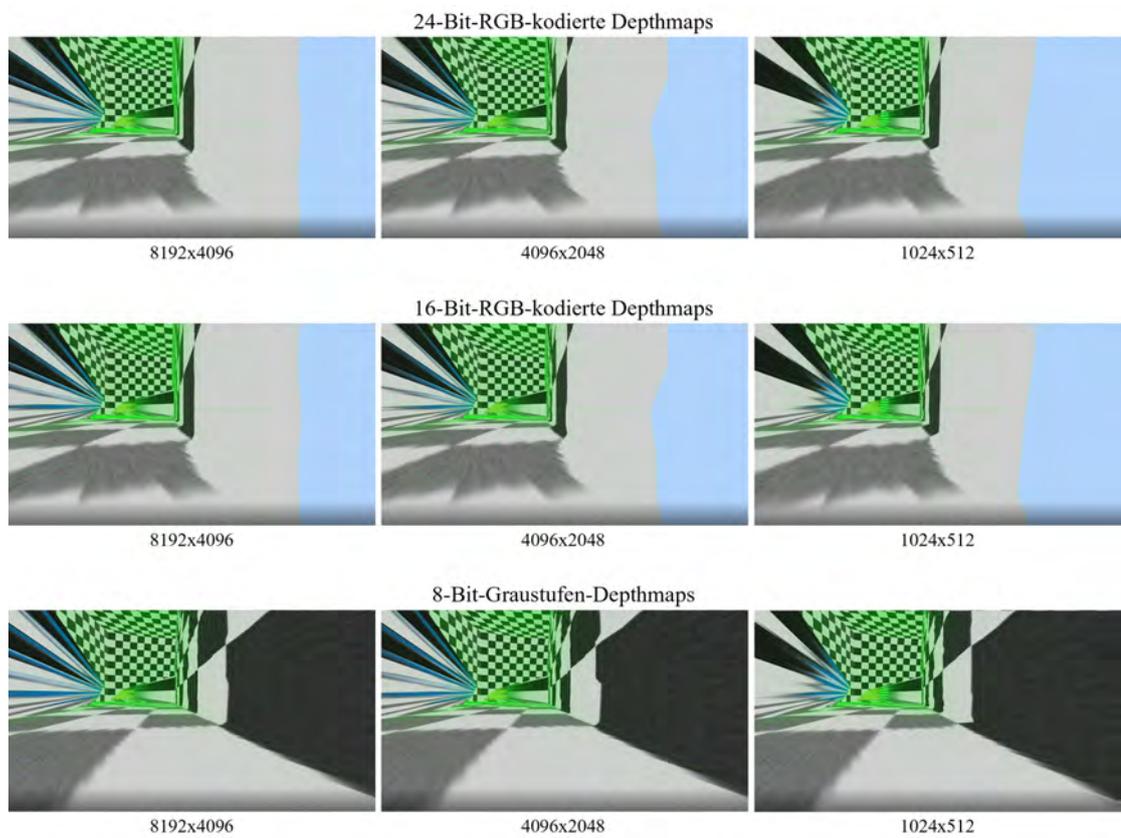


Abbildung 5.6: Vergleich der Genauigkeit der Geometrie bei der Verwendung von Depthmap-Bildern. (Position 2)

5.4 Ästhetik

Für den Vergleich der Ästhetik wurden vier, in Abschnitt 4.3.4 Szenarien erstellt, wobei sich zwei mit gerenderten Panoramabildern und daraus resultierenden idealen Depthmaps befassen. Die anderen beiden verwenden fotografierte Panoramabilder und Depthmaps, die anhand dieser Panoramabilder manuell modelliert wurden.

5.4.1 Vergleich bei gerenderten Panoramabildern

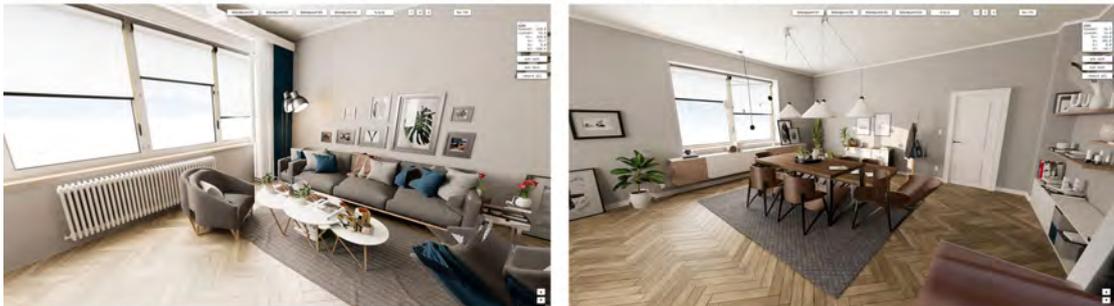
Der Vergleich der Ästhetik von idealen Depthmaps bei gerenderten Panoramabildern wurde anhand zweier Reihen von Screenshots vollzogen. Die erste davon (abgebildet in Abbildung 5.7), befasst sich mit dem Unterschied in der Darstellung von Depthmap-Bildern und 3D-Modellen als Depthmaps. In der Abbildung lässt sich erkennen, dass beide Varianten ihre ästhetischen Eigenschaften haben. So findet sich bei der Verwendung des 3D-Modells eine Verdopplung vor, bei welchem Objekte nochmals auf weiter hinten liegende Geometrie projiziert werden. Bei der Verwendung von Depthmap-Bildern hingegen entsteht eine Verzerrung der Geometrie, bei der eben jene im Schatten der Projektion liegende Bereiche verdeckt werden, wobei die äußeren Pixel eines solchen Objekts die Farben dieser verzerrten Geometrie entscheiden. Zudem entsteht an den Rändern von Objekten ein Treppeneffekt, der auf die unzureichende Auflösung des Meshes zurückzuführen ist, welches bei der Verwendung von Depthmap-Bildern als Projektionsfläche dient (vgl. dazu Abschnitt 2.5.3 des Grundlagenkapitels). Dieser Treppeneffekt führt zudem zusammen mit Verzerrungen der Geometrie dazu, dass an einigen Stellen gezackte Muster auftreten, welche zudem noch langgezogene Texturen, bzw. einfarbige Flächen, haben. Diese Muster werden umso auffälliger, desto weiter die Kamera vom Projektionszentrum entfernt ist, was im Normalfall bei der Verwendung von Depthmaps für Übergangsanimationen jedoch nicht passieren sollte.

Um den Verdopplungseffekt bei der Verwendung von 3D-Modellen als Depthmaps zu verhindern, können die betroffenen Objekte aus dem 3D-Modell entfernt werden. Ein solcher Fall ist in Abbildung 5.8 abgebildet. Hier wurden die Bänke aus dem 3D-Modell entfernt, sodass die Projektion der Bänke stattdessen auf den Boden des Raums fällt. Somit werden die Bänke bei einer Übergangsanimation verzerrt dargestellt und die dreidimensionale Wahrnehmung geschwächt. Der Verdopplungseffekt ist besonders bei Objekten wie Bänken oder Stühlen auffällig, sodass die Weglassung dieser Objekte eine sinnvolle Alternative darstellen kann.

Unreal Rendering



3D-Modell als Depthmap in krpano



Depthmap-Bild in krpano

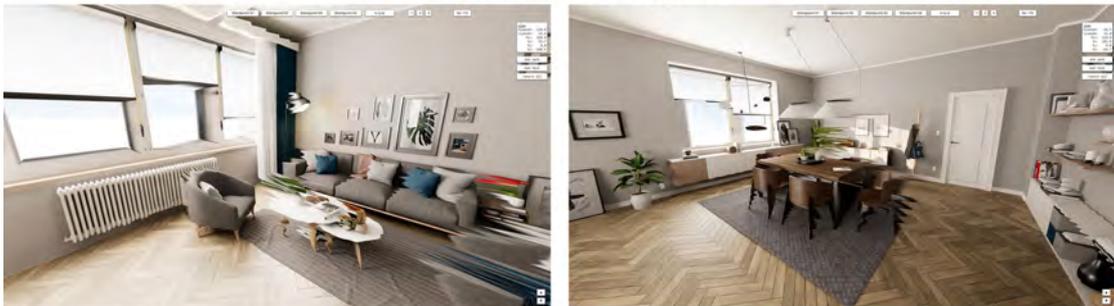
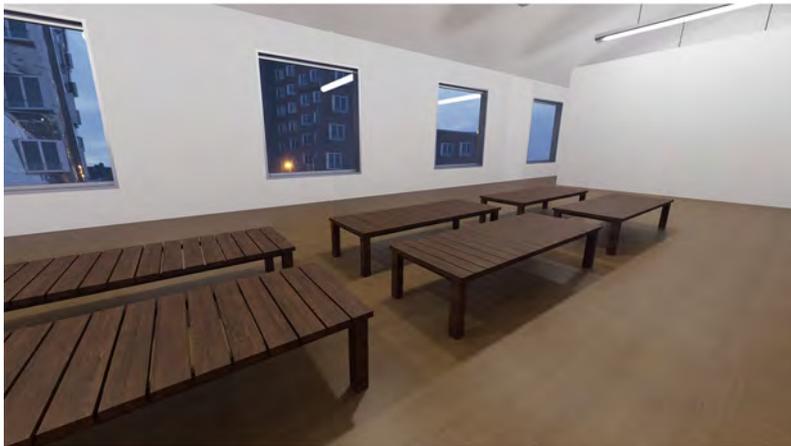


Abbildung 5.7: Vergleich der Ästhetik von Depthmaps bei einer hochwertigen 3D-Szene.

Unreal Rendering



3D-Modell-Depthmap (mit Bänken in der Geometrie)



3D-Modell-Depthmap (ohne Bänke in der Geometrie)

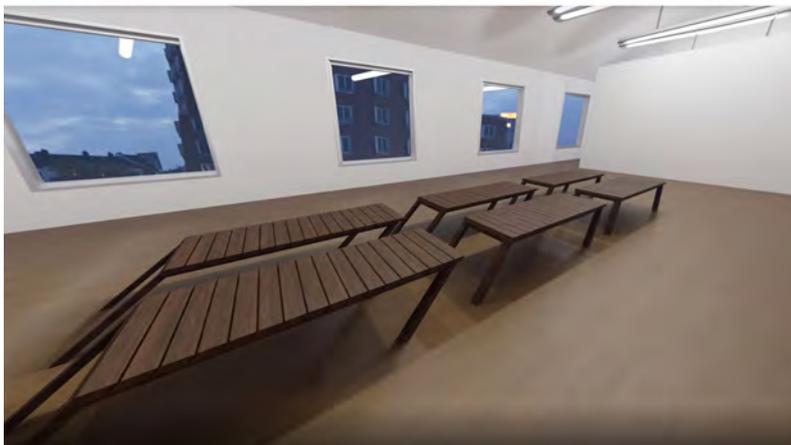


Abbildung 5.8: Vergleich der Ästhetik bei Weglassung bestimmter Objekte im 3D-Modell.

5.4.2 Vergleich bei fotografierten Panoramabildern

Bei der manuellen Erstellung von 3D-Modellen anhand von Panoramabildern gilt, wie bereits weiter oben erwähnt, dass der Prozess sehr fehleranfällig und aufwendig ist. Dies gilt im Besonderen für Szenen mit vielen Objekten und anderen Details, sodass im Modellierungsprozess Entscheidungen bezüglich des Weglassens von Details getroffen werden müssen. Dies lässt sich auch in der oberen Reihe der Abbildung 5.9 beobachten, da hier Details, wie Lampen, Türrahmen oder Vorhänge beim Modellieren weggelassen wurden. Allerdings fallen diese Details bei der Darstellung als Depthmap nicht sehr auf. Das Wesentliche besteht hier in der Darstellung der grundlegenden Geometrie, welche in diesem Szenario durch die einfache Struktur des Raumes gut nachgestellt werden konnte. In diesem Beispiel ist auch der Unterschied zwischen Depthmap-Bild und Depthmap-3D-Modell nicht so gravierend, wie in detailreicheren Szenen. Nur in den verdeckten Bereichen wird hier ein kleiner Unterschied sichtbar, der, wie im vorigen Abschnitt beschrieben, aus der Verzerrung der Geometrie bei Depthmap-Bildern und der Verzerrung der Projektion bei Depthmap-3D-Modellen liegt und besonders in der unteren Reihe bemerkbar ist, wo die Verwendung des Depthmap-Bildes zu einer verschwommen-wirkenden Kante führt. Auffallend ist zudem eine Lücke im 3D-Modell, welche die hellblaue Hintergrundfarbe zum Vorschein bringt. Dahingegen wird beim Depthmap-Bild dieser Bereich automatisch verdeckt.



Abbildung 5.9: Vergleich der Ästhetik von Depthmaps eines fotografierten Panoramas von einem Raum.

Als weiteres Szenario wurde eine Outdoor-Szene verwendet, welche sich nicht durch einfache Geometrische Formen präzise nachmodellieren ließ. Zur Annäherung der tatsächlichen Geo-

metrie wurden daher lediglich primitive Formen, wie Kugeln und Quader verwendet und leicht angepasst (vgl. Abbildung 5.10). Es wird schnell ersichtlich, dass die manuelle Erstellung von Depthmaps hier an ihre Grenzen stößt, da eine genaue Rekonstruktion der komplexen Struktur von Objekten wie Bäumen manuell nicht möglich ist. Besonders durch die Projektion des Himmels auf die Geometrie der Bäume (vgl. Abbildung 5.10 (oben rechts)) verliert das Bild an Glaubwürdigkeit. Allerdings wirkt der Verdopplungseffekt in Bezug auf die Bäume nicht so stark, da sich in diesem Fall die grünen Farbtöne nicht so stark voneinander abheben.

Besonders auffällig ist hingegen die starke geometrische Verzerrung der Depthmap-Bilder, was jedoch auch dadurch zu erklären ist, dass (besonders in der oberen Reihe) ein Kamerastandpunkt gewählt wurde, der weit vom Projektionsmittelpunkt entfernt ist. Diese sehr starken Verzerrungen führen zu einem fast surreal anmutenden Anblick. Aber auch die klar erkennbare Kugel, die bei der Verwendung des 3D-Modells einen Baum abbildet, wirkt sehr unnatürlich. Dabei muss jedoch gesagt werden, dass die Erstellung von 3D-Modellen, die die Form eines Baumes besser wiedergeben als eine Kugel natürlich möglich ist.

Besser abgebildet wird hingegen das Auto, welches dabei lediglich aus einem Mesh mit nur 10 Polygonen besteht, und dennoch in der Form klar erkennbar ist. Außerdem wird der flache Boden trotz kleiner Unebenheiten ebenfalls bei beiden Depthmap-Varianten überzeugend dargestellt und sogar die Grundform lässt sich bei der Verwendung der 3D-Modell-Depthmap gut nachvollziehen.

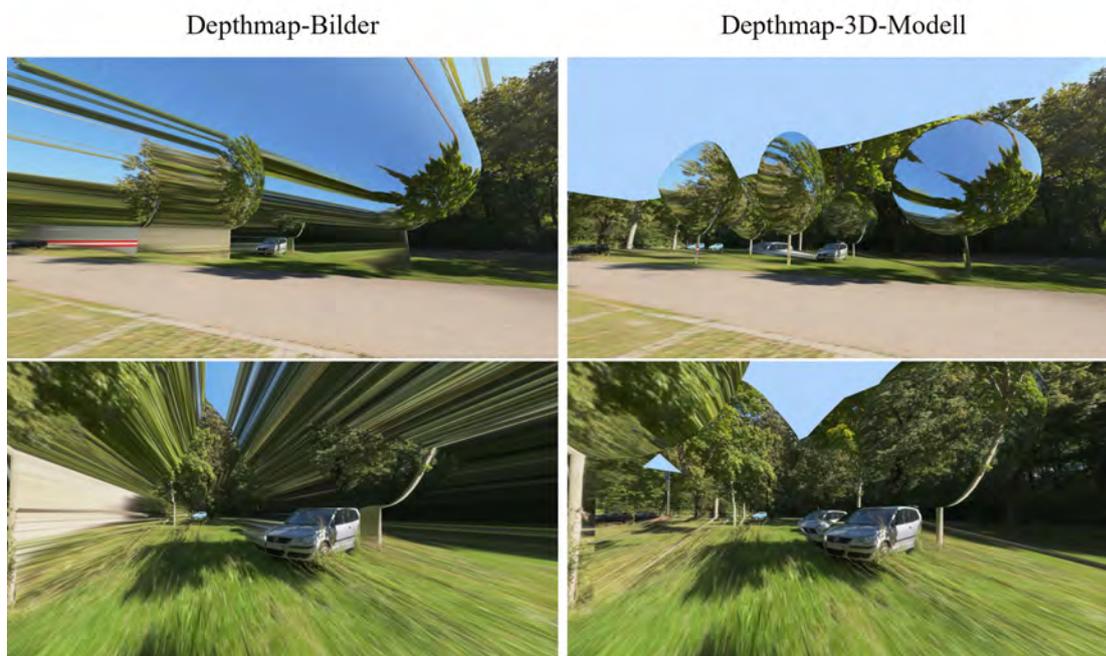


Abbildung 5.10: Vergleich der Ästhetik von Depthmaps eines fotografierten Panoramas einer Außenaufnahme mit komplexer Geometrie.

Ästhetik Zwischenergebnis

In Bezug auf die Ästhetik haben die verschiedenen Depthmap-Typen ihre Vor- und Nachteile. Depthmap-Bilder erzeugen eine verzerrte Geometrie, die durch das Entstehen von Zacken besonders auffällig ist. Gerade bei weitläufigen Szenen fällt dies besonders auf, weshalb es gerade bei Outdoor-Szenen vermieden werden sollte weit vom Projektionspunkt abzuweichen.

Anstatt einer verzerrten Geometrie entsteht bei der Verwendung von 3D-Modellen als Depthmaps hingegen eine Verdopplung, welche einen irritierenden Effekt haben kann, da das verdoppelte Objekt zusätzlich noch verzerrt dargestellt wird. Dies kann allerdings durch das Weglassen von 3D-Objekten verhindert werden kann, wobei jedoch andere Verzerrungen bei der Projektion entstehen.

Die Untersuchung der fotografierten Panoramabilder bei der Verwendung von Depthmaps ergab zudem, dass das manuelle Modellieren von 3D-Modellen bei Szenen, in welchen komplexe Objekte wie Bäume vorkommen, an seine Grenzen stößt. Eine glaubwürdige Repräsentation der Formen einer solchen Szene ist vielleicht überhaupt nicht durch das manuelle Modellieren der 3D-Geometrie möglich. Allerdings lässt sich bei der Verwendung von 3D-Modellen als Depthmaps dennoch die Grundform des abgebildeten Raumes erkennen und könnte somit bei der räumlichen Orientierung helfen. Außerdem zeigte es sich, dass die Wirkung des Verdopplungseffekts bei Szenen mit vielen ähnlichen Objekten, wie beispielsweise Bäumen, nicht so stark ist, da sich die Objekte durch einen geringen Kontrast nicht stark voneinander abheben.

Insgesamt lässt sich feststellen, dass die Verwendung von 3D-Modellen als Depthmaps in KRPAÑO aufgrund der Präzision der Geometrie, sowie der daraus folgend wahrgenommenen Schärfe der Bilder in den meisten Szenarien die bessere Wahl ist. Auch bei weitläufigen Szenen mit ungenauen 3D-Modellen können 3D-Modell-Depthmaps im Gegensatz zu Depthmap-Bildern zumindest noch die Wahrnehmung der grundlegenden Form des umgebenen Raumes ermöglichen. Allerdings lässt sich nach Ansicht des Autors kein klares Urteil über die ästhetischen Vor- und Nachteile der jeweiligen Depthmap-Typen in Bezug auf ihre Verzerrungseigenschaften fällen.

Kapitel 6

Ergebnisse

In diesem Kapitel sollen die Ergebnisse der im vorigen Kapitel beschriebenen Datenanalyse nochmals übersichtlich dargestellt und verglichen werden. Dafür werden die Ergebnisse der einzelnen Untersuchungen aus den Testumgebungen zusammengefasst in einer Gegenüberstellung dargestellt. Zuvor werden noch andere, für den Vergleich der Methoden zur Implementation von Depthmaps in *KRPANO*, relevante Aspekte diskutiert.

6.1 Zusätzliche Erkenntnisse

In diesem Abschnitt sollen noch einige andere Aspekte erwähnt werden, die für die Beurteilung der Vor- und Nachteile von Depthmaps eine Rolle spielen können. Dies sind Erkenntnisse, die durch das Arbeiten mit Depthmaps in *krpano* gewonnen wurden, aber nicht durch zusätzliche empirische Methoden überprüft wurden.

6.1.1 Arbeitsaufwand

Der Arbeitsaufwand allgemein ist für die Erstellung von Depthmaps schwierig zu bewerten, da es viele Faktoren gibt, die den Arbeitsaufwand maßgeblich beeinflussen. Allerdings lässt sich klar sagen, dass im Vergleich der beiden Depthmap-Typen (Bilder und 3D-Modelle), die Erstellung von 3D-Modellen die weniger aufwendigere ist, da die Depthmap-Bilder nur aus 3D-Modellen heraus gerendert werden können. Bei der Einpflegung der Depthmaps in die *KRPANO*-Panoramatur haben beide Typen jedoch ihre Vor- und Nachteile. Depthmap-Bilder werden immer vom Projektionszentrum aus auf die Geometrie projiziert, sodass die Geometrie nicht zusätzlich im 3D-Raum verschoben werden muss. Bei der Verwendung von 3D-Modellen kann dies der Fall sein, wenn nur ein einzelnes 3D-Modell verwendet wird. Hier muss das Modell in Bezug auf die Kameraposition verschoben werden. Bei der Verwendung von einzelnen 3D-Modellen für jedes Panoramabild würde diese Arbeit bereits beim Erstellen der Depthmap im Voraus in der 3D-Software geschehen.

Einen Vorteil, den Depthmap-Bilder gegenüber 3D-Modellen haben, ist, dass Details nachträglich gut angepasst werden können. So können beispielsweise kleine Fehler behoben werden, indem in einem Bildbearbeitungsprogramm einzelne Formen aus dem Panoramabild

ausgeschnitten und in das Depthmap-Bild eingefügt werden kann, wobei nur noch die Farbe an einen passenden Tiefenwert angepasst werden muss. Somit kann der Arbeitsaufwand zur nachträglichen Anpassung in diesem Fall geringer sein, als ein 3D-Modell zu bearbeiten. Diese Art der Bildbearbeitung kann effektiv jedoch nicht mehr bei den 16- und 24 Bit Depthmap-Bildern erfolgen.

6.1.2 Integration von Hotspots

Einen nicht unerheblichen Vorteil hat das Verwenden von 3D-Modellen als Depthmap in Bezug auf die Kombination mit dreidimensionalen KRPAÑO-Hotspots.¹ Hierbei entsteht der Vorteil vor allem dadurch, dass die geometrischen Daten des 3D-Modells für die Platzierung von Hotspots genutzt werden können. Wenn es beispielsweise bekannt ist, dass der Boden im 3D-Modell 1,5 m unter der Kamera ist, kann einem dreidimensionalen Hotspot der Y-Wert -1.5 zugewiesen werden, sodass dieser auf dem Boden aufliegend dargestellt wird. Bei der Verwendung von Depthmap-Bildern ist dies zwar auch möglich — um die Depthmap richtig zu skalieren gibt es extra ein Tool² — jedoch gibt es auch bei einer idealen Skalierung zum einen das Problem, dass ebene Flächen oft nicht präzise genug dargestellt werden, und zum anderen, dass Hotspots, die im Schattenbereich der Projektion liegen, von der verzerrten Geometrie verdeckt werden.

¹Hotspots können z.B. Grafiken, Bilder oder Videos sein

²https://krpano.com/viewsource.html?releases/1.20.11/examples/./viewer/examples/depthmap/depthmap-helper-tool/depthmap_helptool.xml

6.2 Übersicht der Vor- und Nachteile

Im Folgenden werden die Vor- und Nachteile der beiden Depthmap-Typen in Bezug auf die oben aufgeführten Kriterien übersichtlich dargelegt.

Depthmap-Bilder

Speicherbedarf und Ladezeiten

Vorteile: Die Dateigrößen von Depthmap-Bildern vergrößern sich aufgrund der Kompressionsalgorithmen bei detailreichen Szenen zwar erheblich (ungefähr bis zum Faktor 2), trotzdem können sie in diesem Fall im Vergleich zu 3D-Modellen geringer ausfallen. Zudem ist der Speicherbedarf bei niedrig-Auflösenden 8-Bit Graustufenbildern mit ungefähr 30-50 kb sehr gering.

Nachteile: Bei weniger detailreichen Szenen (<2000 Tris) sind Depthmap-Bilder in jeder Form größer als ihr Pendant im 3D-Modell. Zudem muss für jedes Panoramabild ein eigenes Depthmap-Bild verwendet werden, sodass der Speicherbedarf mit der Anzahl an Panoramabildern linear steigt.

Performance

Vorteile: Die Performance, gemessen an der Bildfrequenz, war in jedem Fall ausreichend. Nur auf schwächeren Geräten wurden bei der Verwendung hochauflösender, mindestens 16-Bit tiefer Bilder, Einbußen in der Performance gemessen, die jedoch nicht sehr erheblich waren.

Nachteile: Es kann angenommen werden, dass bei noch schwächeren Geräten Probleme auftreten können, da das Mesh, das für die Geometrie von Depthmap-Bildern verwendet wird, eine Kugel mit einer

3D-Modell-Depthmaps

Speicherbedarf und Ladezeiten

Vorteile: 3D-Modelle als Depthmaps zu Verwenden ist für die meisten Szenen deutlich weniger speicherintensiv. Auch Ladezeiten sind somit kürzer als bei der Verwendung von Depthmap-Bildern. Dies gilt in der Regel für Szenen, die eine Polygonanzahl von weniger als 500000 Tris haben, was beim manuellen Erstellen von Depthmaps den Normalfall darstellen sollte.

Nachteile: Bei sehr detailreichen 3D-Modellen können Dateigrößen sehr groß werden. Dies gilt insbesondere, wenn die Dateigröße vor der Kompression mit dem krpno-Encryption Tool über 10mb liegt, da die Dateien ab diesem Limit nicht mehr komprimiert werden können.

Performance

Vorteile: Bis zu einer Polygonanzahl von 32.000 Tris konnten keine nennenswerten Performance-Einbußen gemessen werden.

Nachteile: Auf schwächeren Geräten wirken sich detailreiche 3D-Modelle auf die Bildfrequenz aus. Dies wird jedoch erst bei einer Polygonanzahl von über 1.000.000 Tris problematisch, da hier bei schwächeren Geräten auch Bildfrequenzen von unter 25 Hz erreicht werden können.

Auflösung von 800.000 Polygonen (Quads) ist. Inwiefern sich dies konkret auswirkt, konnte hier jedoch nicht festgestellt werden.

Präzision

Depthmap-Bilder weisen lediglich bei der Verwendung einer hohen Auflösung und einer Bittiefe von mindestens 16-Bit, bezogen auf die Geometrie, gute Ergebnisse auf. Jedoch entstehen selbst bei solchen Depthmaps keine komplett geraden Flächen und besonders in vom Projektionszentrum weiter entfernten Bereichen fällt die fehlende Genauigkeit auf. Die Verwendung von 8-Bit Graustufenbildern als Depthmaps kann zudem dazu führen, dass größere Räume in ihrer Geometrie stark verzerrt werden.

Ästhetik

Vorteile: Je nach Geschmack kann die Verdeckung der Bereiche, die bei der Verwendung von Depthmap-Bildern im Schatten der Projektion liegen, auch ein Vorteil gegenüber der verdoppelten Darstellung bei 3D-Modell-Depthmaps sein.

Nachteile: Die oben genannten Ungenauigkeiten der Geometrie und der Projektion auf die Geometrie wirken sich auch negativ auf die Ästhetik der Depthmaps aus. So entsteht an den Rändern von Objekten beispielsweise eine Art Treppeneffekt, der zusätzlich zu einem Zackenmuster in den verzerrten Bereichen der Geometrie führen kann. Diese Verzerrungen der Geometrie befinden sich im Schatten der Projektion.

Arbeitsaufwand

Vorteile: Bei Depthmap-Bildern können Details im Nachhinein noch mithilfe eines Bildbearbeitungsprogramms angepasst werden, wobei

Präzision

Die 3D-Modelle sind sowohl in Bezug auf die Präzision der Geometrie genauer, als auch in der Präzision der Projektion des Panoramabildes auf diese Geometrie. Zudem weisen sie auch bei der Verwendung verschiedener Formate dieselben Ergebnisse auf.

Ästhetik

Vorteile: Die Genauigkeit der Depthmaps ist hier ein klarer Vorteil gegenüber Depthmap-Bildern.

Nachteile: Als Nachteil lässt sich der Effekt der Verdopplung von Objekten auslegen. Dabei erscheinen Objekte, die, vom Projektionszentrum aus gesehen, einen Schatten werfen, doppelt. Zudem erscheint die zweite Projektion verzerrt auf einer weiter hinten liegenden Geometrie.

Arbeitsaufwand

Vorteile: Das Erstellen von 3D-Modellen bedeutet in der Regel einen hohen Arbeitsaufwand. Dafür können 3D-Modelle jedoch direkt als

diese Arbeit nicht so genau ist, wie das Rendern dieser Details in einer 3D-Software. Je nachdem kann das Hinzufügen und Rendern neuer Geometrie auch einen geringeren Arbeitsaufwand bedeuten. Zusätzlich müssen Depthmap-Bilder bei Einpflegen in die Panoramatour nicht mehr verschoben werden.

Nachteile: Depthmap-Bilder werden aus 3D-Modellen heraus gerendert, wodurch der Arbeitsaufwand auf dem der 3D-Modelle aufbaut und somit allein logisch betrachtet höher sein muss.

Integration von Hotspots

Nachteile: Für die Platzierung von Hotspots im dreidimensionalen Raum können nicht einfach die Positionsdaten aus dem 3D-Modell verwendet werden, sondern es müssen entweder Daten bei der Verwendung des Depthmap-Bildes manuell gefunden werden, oder die Depthmap muss mithilfe eines Tools skaliert werden.

Depthmaps verwendet werden und müssen nicht erst wie Depthmap-Bilder zuerst noch gerendert werden.

Nachteile: Für das Einpflegen von 3D-Modell-Depthmaps müssen diese, im Fall der Verwendung eines 3D-Modells für mehrere Panoramabilder, in KRPAÑO erst an die richtige Stelle im 3D-Raum transformiert werden. Zusätzlich muss sichergestellt sein, dass die Panoramabilder dabei in dieselbe Richtung ausgerichtet sind, da 3D-Modell-Depthmaps innerhalb von KRPAÑO nicht unabhängig vom Panoramabild gedreht werden können. Im Fall der Verwendung von je einem 3D-Modell pro Panoramabild, ist das nicht nötig.

Integration von Hotspots

Vorteile: 3D-Hotspots können anhand der Mesh-Informationen des 3D-Modells im Raum platziert werden. Dabei muss nur beachtet werden, dass die Achsen und die Skalierung bei KRPAÑO gegebenenfalls unterschiedlich vom 3D-Programm sind.

Nachdem die Vor- und Nachteile der Depthmaps in dieser Gegenüberstellung dargestellt worden sind, sollen diese Ergebnisse im nächsten Abschnitt noch in verschiedene Anwendungskontexte eingeordnet werden.

6.3 Einordnung der Ergebnisse

In diesem Abschnitt sollen die Ergebnisse des Vergleichs der beiden Depthmap-Typen in KRPAÑO in den Kontext von konkreten Anwendungsfällen gesetzt werden. Dazu ist es notwendig die, nach den Vergleichskriterien getrennten, einzelnen Ergebnisse in Bezug zueinander zu setzen. So steht die Maximierung der Bildqualität beispielsweise im wechselseitigen Verhältnis zur Maximierung der Performance. Allerdings geht aus den Ergebnissen auch hervor, dass es eindeutige Vorteile bei der Verwendung mancher Depthmaps gegenüber anderen gibt. So lässt sich aus aufgelisteten Ergebnissen eindeutig erkennen, dass die Verwendung von 3D-Modellen als Depthmaps in KRPAÑO in den meisten Fällen deutlich von Vorteil ist. Das liegt größtenteils daran, dass 3D-Modelle dreidimensionale Geometrie-Daten deut-

lich effizienter speichern als es Depthmap-Bilder tun. Zugleich muss bei Depthmap-Bildern für die Darstellung als 3D-Geometrie noch ein zusätzlicher Schritt vollzogen werden, der darin besteht die Tiefeninformationsdaten auf ein schon bestehendes Mesh anzuwenden, welches in der Auflösung notwendigerweise begrenzt ist. Im Fall von KRPAÑO reicht die dafür verwendete Kugel mit einer Polygonanzahl von 800.000 Quads nicht dazu aus eine vergleichbar-gute Präzision der Geometrie zu gewährleisten. Auch aufgrund dieser Ineffizienz weisen Depthmap-Bilder in der Regel höhere Dateigrößen auf. Lediglich bei stark in der Auflösung reduzierten, oder bei der Verwendung von 8-Bit Graustufenbildern konnten Dateigrößen erreicht werden, die dazu ausreichen ein Depthmap-Bild bei durchschnittlichen Downloadgeschwindigkeiten innerhalb von 40 ms (ein Frequenzintervall bei einer Bildrate von 25 Hz) herunterzuladen. Im Vergleich dazu konnte dies noch bei 3D-Modell-Depthmaps mit bis zu 32.000 Tris bewerkstelligt werden. An dieser Stelle soll jedoch erwähnt werden, dass die Einhaltung der Downloadzeit von unter 40ms keine Anforderung ist, sondern als Hilfestellung zum Vergleich der Performance auf Basis der Erreichung einer komplett ruckelfreien Verwendung von Depthmaps gesetzt wurde. Je nach persönlichem Empfinden können jedoch auch kurze Ladezeiten (zudem noch im Bereich von Millisekunden) in Kauf genommen werden, auch ohne möglicherweise ein Immersionsempfinden zu brechen.

Ein typischer Anwendungsfall für die Verwendung von Depthmaps besteht in einer Panoramatour, die aus fotografierten Panoramabildern besteht und für die die Depthmaps erst noch hergestellt werden müssen. Die Panoramabilder werden in diesem Fall also nicht wie in dieser Arbeit aus den 3D-Modellen heraus gerendert, sondern die Geometrie wird anhand der existierenden Panoramabilder in einer 3D-Software nachgebaut. Dabei ist zu erwarten, dass die 3D-Modelle keine hohen Polygonanzahlen haben werden, da der Vorgang der manuellen Herstellung arbeitsaufwändig und fehleranfällig ist. Für diesen typischen Anwendungsfall wäre somit die direkte Verwendung der 3D-Modelle als Depthmaps die eindeutig performantere Option.

Ein möglicher Anwendungsfall, bei dem die Verwendung von Depthmap-Bildern bezogen auf die Performance die bessere Wahl wäre, wäre eine virtuelle Panoramatour, bei der die Panoramabilder, wie in Abschnitt 4.3.4 gezeigt, aus einer hochkomplexen 3D-Szene heraus gerendert werden. In dem erwähnten Beispiel besaß die Szene viele hochauflösende Modelle, sodass das 3D-Modell mit einer Polygonanzahl von über 2 Millionen Tris und einer Dateigröße von über 60 mb nicht mehr sinnvoll als Depthmap verwendet werden konnte. In einem solchen Fall können Depthmap-Bilder eine sinnvolle Alternative bieten.

Von diesem Fall abgesehen kann die Verwendung von Depthmap-Bildern nur in Bezug auf die Ästhetik als Alternative angesehen werden. Hierbei ist es der subjektiven Wahrnehmung eines jeden Überlassen, ob die Verzerrungen, die bei der Verwendung von Depthmap-Bildern entstehen, oder ob die Verdopplungen, die bei der Nutzung von 3D-Modell-Depthmaps aufkommen, einen größeren Störfaktor für das jeweilige ästhetische Empfinden darstellt.

Kapitel 7

Zusammenfassung und Ausblick

7.1 Zusammenfassung

In dieser Arbeit wurde eine Methodik zum Erfassen von Daten entwickelt, mit deren Hilfe ein Vergleich zweier verschiedener Möglichkeiten der Erweiterung von KRPANO-Panoramatauren, durch die Integration von Tiefeninformationen, aufgestellt wurde. Hierzu wurden zunächst einige Vergleichskriterien aufgestellt, welche auch besonders Rücksicht darauf nehmen, dass eine KRPANO-Panoramataur eine Webanwendung ist und somit auch einigen technischen Limitationen unterliegt.

1. Speicheraufwand und Downloadzeiten
2. Performance
3. Präzision von Projektion und Geometrie
4. Ästhetik
5. Arbeitsaufwand
6. Integration von Hotspots

Für die Kriterien 2-4 wurden zur Erfassung von Daten schließlich drei Testumgebungen konzeptioniert und umgesetzt:

1. Testumgebung zum Vergleich der Performance (Bildfrequenz)
2. Testumgebung zum Vergleich der Präzision von Projektion und Geometrie
3. Testumgebung zum Vergleich der Ästhetik

Die Testumgebungen bestehen dabei alle aus Panoramatauren, die aus gerenderten Panoramabildern bestehen. Das Rendern von Panoramabildern hat den Vorteil, dass die zum Rendern verwendete Geometrie ebenfalls zur Erstellung der Depthmaps genutzt werden kann,

und sie somit die anschließende Projektion der Panoramabilder exakt auf die Geometrie der Depthmaps passt. Somit wurden Fehler, die beim manuellen Erstellen von Depthmaps entstehen können, ausgeschlossen. Für das Rendern von Depthmap-Bildern wurde eigens ein Post-Process-Material in der Unreal Engine 4 erstellt und in dieser Arbeit zusammen mit einigen anderen Teilschritten dokumentiert.

Die Testumgebung zum Vergleich der Performance besteht aus einer virtuellen Panoramatour mit verschiedenen Depthmap-Typen und -Qualitäten. Hier wurde eine KRPANO-Action programmiert, mit deren Hilfe ein Bedienverhalten simuliert wurde, wobei gleichzeitig die durchschnittliche Bildfrequenz gemessen wurde. Die Bildfrequenz wurde an drei Geräten mit unterschiedlicher Leistung gemessen (Smartphone, Tablet-PC und Tower-PC). Zum Vergleich der Präzision der Projektion und der Geometrie von Depthmaps wurde eine Testumgebung entwickelt, in welcher verschiedene Formen und Muster verwendet wurden, um mögliche Verzerrungen der Geometrie, oder der Projektion des Bildes auf die Geometrie, anhand von visuellen Vergleichen mithilfe von Screenshots festzustellen. Die Ästhetik wurde auf ähnliche Weise mithilfe von Screenshots verglichen. Hierfür wurden zwei Szenarien angelegt. Im ersten wurde ein hochkomplexes Modell zum Rendern von Panoramabildern verwendet, um die Bildqualität bei der Verwendung von Depthmaps in detailreichen Szenen zu vergleichen; im zweiten wurde eine einfache Szene erstellt, bei der die Wirkung von einzelnen Objekten auf die Darstellung der Depthmaps genauer untersucht wurden.

Aus der Analyse der Daten ergab sich, dass die Verwendung von 3D-Modellen als Depthmaps in den meisten Anwendungsfällen klare Vorteile aufweist. Diese Vorteile bestehen vor allem in der Genauigkeit der Geometrie und aus den relativ kleinen Dateigrößen, die 3D-Modelle im Vergleich zu Depthmap-Bildern haben. Die Dateigröße ist hier besonders relevant, da die Depthmaps zunächst von einem Server heruntergeladen werden müssen, ehe sie angezeigt werden können. Aus diesem Grund wurden die Dateigrößen in der Datenanalyse mit verschiedenen Downloadgeschwindigkeiten in Relation gesetzt. Daraus ergab sich, dass bei einer üblichen Downloadgeschwindigkeit 75 MBit/s 3D-Modelle mit einer Polygonanzahl von bis zu 32.000 Tris noch innerhalb eines Frequenzintervalls bei einer Bildfrequenz von 25 Hz heruntergeladen werden konnten. Was einem Wert entspricht, bei dem eine Verzögerung vom Laden des 3D-Modells kaum bemerkbar sein sollte. Bei den Depthmap-Bildern finden sich hingegen nur bei stark verkleinerten Bildern, oder solchen die nur 8-Bit als Bittiefe verwenden, Werte, die dieses Kriterium erfüllen. Im Vergleich der Performance konnten nur auf schwächeren Geräten Einbußen gemessen werden, wobei hier auch erwähnt sein muss, dass die Bildfrequenz auch bei komplexeren Depthmaps noch in Bereichen von über 25 Hz lag. Bei der Verwendung von hochauflösenden Depthmap-Bildern oder 3D-Modellen mit über 32.000 Tris, konnten in Kombination mit schwächeren Geräten leichte Verluste in der Bildfrequenz gemessen werden. In Bezug auf die Genauigkeit der Projektion und Geometrie weisen die Depthmap-3D-Modelle ebenfalls klar bessere Ergebnisse als die Depthmap-Bilder auf. Die Geometrie von 3D-Modellen ist ideal, während die der Depthmap-Bilder selbst bei glatten Flächen und der Verwendung hochauflösender mit 24-Bit kodierten Bilder, leichte Wellen aufweist und besonders an den Rändern von Objekten Treppeneffekte entstehen. Bei der Ästhetik der Depthmaps besteht entweder das Problem der Verzerrung der Geometrie bei der Verwendung von Depthmap-Bildern oder das Problem der Verdoppelung von Objek-

ten bei der Verwendung von Depthmap-3D-Modellen. Alternativ können jedoch bestimmte Objekte auch ganz aus der Depthmap herausgelassen werden, sodass statt einer Verzerrung der Geometrie oder einer Verdoppelung von Objekten eine Verzerrung der Projektion entsteht. Ein weiterer Faktor ist zudem die Ungenauigkeit der Geometrie bei der Verwendung von Depthmap-Bildern. Die Integration von *KRPANO*-Hotspots in die durch Tiefeninformationen aus Depthmaps erweiterte Panoramatur, funktioniert bei der Verwendung von Depthmap-3D-Modellen auch besser als bei Depthmap-Bildern, bei denen die Skalierung erst manuell angepasst werden muss, und glatte Flächen leichte Wellen aufweisen. In Bezug auf den Arbeitsaufwand sind 3D-Modelle ebenfalls im Vorteil, da für die Erstellung von Depthmap-Bildern selbst 3D-Modelle verwendet werden müssen. Allerdings muss auch beachtet werden, dass bei der Einpflegung von Depthmap-Bilder in eine Panoramatur an sich keine Veränderung an der Position der Geometrie vorgenommen werden muss, da die Projektion des Depthmap-Bildes aus der gleichen Position, wie die Projektion des Panoramabildes stammt. Bei 3D-Modellen (zumindest bei der Verwendung eines einzelnen 3D-Modells für mehrere Panoramabilder) muss hingegen die Geometrie in Bezug auf die Kameraposition an die richtige Stelle im dreidimensionalen Raum transformiert werden.

Aus den Ergebnissen lassen sich drei verschiedene Anwendungsfälle folgern, die als Entscheidungsgrundlage zur Auswahl einer der beiden Depthmap-Typen verwendet werden können.

1. Verwendung von Depthmap-3D-Modellen bei einer Polygonanzahl von bis zu 32.000 Tris (keine Latenz) oder 250.000 Tris (geringfügige Latenz).
2. Verwendung von Depthmap-Bildern aufgrund der möglicherweise ansprechenderen Ästhetik.
3. Verwendung von Depthmap-Bildern bei Panoramabildern, die aus hochkomplexen 3D-Szenen gerendert wurden und eine Polygonanzahl von über 250.000 Tris aufweisen.

7.2 Ausblick

Die Ergebnisse aus dieser Arbeit zeigen auf, dass die Verwendung von 3D-Modellen als Depthmaps in *KRPANO* in den meisten Anwendungsfällen die sowohl effizientere, als auch qualitativ hochwertigere Variante ist. Daher erscheint es sinnvoll, weiter in Richtung der Verwendung von 3D-Modellen als Depthmaps zu forschen. Vor allem die Verdoppelung der Projektion sorgt noch für auffallende Bildfehler, sodass dies als ein zu lösendes Problem angesehen werden kann. Mögliche Lösungen könnten dabei in der Ergänzung durch Dollhouses bestehen, welche mithilfe einer 2D-Textur die fehlenden Bildinformationen durch die Synthese aus mehreren Panoramabildern abdecken können. Da die Dollhouses in der Bildqualität jedoch schlechter als Panoramabilder sind, könnten sie auch ausschließlich dafür verwendet werden, um, beispielsweise bei einem räumlichen 3D-Übergangseffekts, die fehlenden Bereiche abzudecken, sodass kein Verdopplungseffekt entsteht und die gute Bildqualität der Panoramabilder erhalten bleibt. Alternativ könnten auch Verfahren entwickelt werden, die anstelle der verdoppelten Projektion versuchen eine Annäherung der tatsächlichen Farben

und Formen in diesen Schattenbereichen zu erreichen.

Ein weiteres Problem besteht unterdessen in der manuellen Herstellung von 3D-Modellen aus Panoramabildern. Dieser Prozess ist vor allem für detailreiche Szenen und für Szenen, in welchen die Geometrie nicht durch einfache, orthogonal zueinander stehende Formen repräsentiert werden können, sehr fehleranfällig und arbeitsaufwendig. Hierzu gibt es bereits Lösungsansätze, die entweder durch die Verwendung spezieller Hardware, wie Laserscanner, Tiefeninformationen in der echten Welt messen und Panoramabildern zuordnen [ADF⁺10]. Oder mit der Verwendung von Algorithmen und neuronalen Netzwerken, wie sie in aktueller Forschung vorgeschlagen werden, aus Bildern selbst Tiefeninformationen generieren.

Um das Problem des Sprungs beim Übergang zwischen zwei Panoramabildern zu lösen, könnten in Zukunft zudem vermehrt 360-Grad-Videos zum Einsatz kommen, welche durch die aktuellen Grenzen in der Hardware der Endverbraucher, sowie der Übertragungsraten im Internet, noch nicht flächendeckend in hoher Auflösung genutzt werden können.

Literaturverzeichnis

- [Ack22] ACKERMANN, Ralf M.: *Panorama-Software im Test (2020 / 2021)*. <https://www.ralf-michael-ackermann.de/Software%20Kugelpanoramen.htm>. Version: 2022
- [ADF⁺10] ANGUELOV, Dragomir ; DULONG, Carole ; FILIP, Daniel ; FRUEH, Christian ; LAFON, Stéphane ; LYON, Richard ; OGALE, Abhijit ; VINCENT, Luc ; WEAVER, Josh: Google Street View: Capturing the World at Street Level. In: *Computer* 43 (2010), Nr. 6, S. 32–38. <http://dx.doi.org/10.1109/MC.2010.170>. – DOI 10.1109/MC.2010.170
- [Ars22] ARS TECHNICA: *Google's Street View cars are now giant, mobile 3D scanners*. <https://arstechnica.com/gadgets/2017/09/googles-street-view-cars-are-now-giant-mobile-3d-scanners/>. Version: 3/18/2022
- [BB06] BENDER, Michael ; BRILL, Manfred: *Computergrafik Ein Anwendungsorientiertes Lehrbuch*. Hanser, 2006
- [CDF⁺17] CHANG, Angel ; DAI, Angela ; FUNKHOUSER, Thomas ; HALBER, Maciej ; NIESSNER, Matthias ; SAVVA, Manolis ; SONG, Shuran ; ZENG, Andy ; ZHANG, Yinda: Matterport3D: Learning from RGB-D Data in Indoor Environments. In: *International Conference on 3D Vision (3DV)* (2017)
- [CF14] CABRAL, Ricardo ; FURUKAWA, Yasutaka: Piecewise Planar and Compact Floorplan Reconstruction from Images. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, S. 628–635
- [CY99] COUGHLAN, James M. ; YUILLE, Alan L.: Manhattan World: compass direction from a single image by Bayesian inference. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision 2* (1999), S. 941–947 vol.2
- [Epi21] EPIC GAMES: *Real-time Ray Tracing*. <https://docs.unrealengine.com/4.27/en-US/RenderingAndGraphics/RayTracing/>. Version: Sep 2021
- [Esc19] ESCOBAR, Thomas: *Google maps 101: How imagery powers our map*. <https://www.blog.google/products/maps/google-maps-101-how-imagery-powers-our-map/>. Version: Dec 2019

- [HZ04] HARTLEY, Richard ; ZISSERMAN, Andrew: *Multiple view geometry in Computer Vision*. Cambridge University Press, 2004
- [Jac04] *Kapitel Grundlagen*. In: JACOBS, Corinna: *Digitale Panoramen*. Springer Berlin Heidelberg, 2004, S. 18
- [MTUK95] MILGRAM, Paul ; TAKEMURA, Haruo ; UTSUMI, Akira ; KISHINO, Fumio: Augmented reality: a class of displays on the reality-virtuality continuum. In: DAS, Hari (Hrsg.): *Telemanipulator and Telepresence Technologies*, SPIE, 1995 (SPIE Proceedings), S. 282–292
- [RGW⁺21] RAMAKRISHNAN, Santhosh K. ; GOKASLAN, Aaron ; WIJMANS, Erik ; MAKSYMETS, Oleksandr ; CLEGG, Alexander ; TURNER, John M. ; UNDERSANDER, Eric ; GALUBA, Wojciech ; WESTBURY, Andrew ; CHANG, Angel X. ; SAVVA, Manolis ; ZHAO, Yili ; BATRA, Dhruv: Habitat-Matterport 3D Dataset (HM3D): 1000 Large-scale 3D Environments for Embodied AI. In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021
- [Sta22] STATCOUNTER GLOBAL STATS: *Desktop vs Mobile vs Tablet Market Share Germany — Statcounter Global Stats*. <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/germany>. Version: 3/18/2022
- [TW15] TAO, Lingjun ; WANG, Yigang: A WebGL-based 3D virtual home roaming by seamlessly connecting videos to panoramas. In: *2015 8th International Congress on Image and Signal Processing (CISP)*, 2015, S. 498–503
- [Wik21] WIKIPEDIA: *RGB-Farbraum — Wikipedia, die freie Enzyklopädie*. <https://de.wikipedia.org/w/index.php?title=RGB-Farbraum&oldid=215777086>. Version: 2021. – [Online; Stand 29. Mai 2022]
- [Wit19] WITZANI, Chris: *Projektionen in der Panoramafotografie*. <https://360.schnurstracks.de/projektionen-in-der-panoramafotografie/>. Version: Oct 2019
- [YJL⁺18] YANG, Yang ; JIN, Shi ; LIU, Ruiyang ; KANG, Sing B. ; YU, Jingyi: Automatic 3D Indoor Scene Modeling from Single Panorama. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, S. 3926–3934
- [ZCC⁺21] ZHANG, Cheng ; CUI, Zhaopeng ; CHEN, Cai ; LIU, Shuaicheng ; ZENG, Bing ; BAO, Hujun ; ZHANG, Yinda: DeepPanoContext: Panoramic 3D Scene Understanding with Holistic Scene Context Graph and Relation-based Optimization. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, S. 12612–12621