
Bachelorarbeit

zur Erlangung des Grades
Bachelor of Science (B. Sc.)
im Studiengang Mensch-Computer-Systeme
an der Universität Würzburg

Immersive Interaktion mit einer virtuellen Erdkugel mit Darstellung meteorologischer Daten

vorgelegt von
Daniel Rapp
Matrikelnummer: 1954753

am 15. April 2016

Betreuer/Prüfer:
Prof. Marc Erich Latoschik, Informatik IX, Universität Würzburg
M.-Space Tech. Borja Garcia, Informatik VIII, Universität Würzburg

Zusammenfassung

Wetterdaten von Satelliten liefern heutzutage wichtige Informationen für Wissenschaft und Meteorologie und sind inzwischen auch für universitäre Zwecke verfügbar. Da bisher Aufnahmen dieser Satelliten an der **Julius-Maximilians-Universität (JMU)** in Würzburg lediglich zweidimensional dargestellt werden, verfolgt diese explorative Arbeit eine neue, dreidimensionale Repräsentation. Im Zuge dessen verarbeitet ein hierfür selbstgeschriebenes Programm die Aufnahmen von fünf ausgewählten Satelliten und kombiniert diese zu farbigen Texturen. Es sind in dieser Arbeit fünf verschiedene Möglichkeiten implementiert, die schwarzweißen Aufnahmen einzufärben. Diese werden im Anschluss auf einer virtuellen Erdkugel innerhalb einer, durch die Unreal Engine dargestellte, Szene präsentiert. Die Interaktion erfolgt über eine Wii Mote, während Nutzer die Erde über die Oculus Rift betrachten können. Zukünftige Arbeiten können die Anwendung mit Nutzern testen.

Abstract

Today, data from satellites are not only important for meteorology or science but also largely available for universities and research purposes. Up until now, the **JMU** in Wuerzburg only showed these pictures in a two-dimensional way, so this work explores a new three-dimensional representation. The final program uses images from five satellites and converts these black and white images into colored textures, while providing five different approaches to do so. The textures are then used within a scene rendered by the Unreal Engine to present a virtual earth. Users might view it wearing the Oculus Rift and can interact with it using the Wii Mote Controller. The program is in its first iteration and therefore may be evaluated with users in future work.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	ix
Acronyms	xi
1 Einleitung	1
1.1 Wetterbeobachtung	1
1.2 Oculus Rift	1
1.3 Beteiligung der Lehrstühle	2
1.4 Hintergrund der Arbeit	3
2 Aufgabenstellung	5
3 Stand der Forschung	7
3.1 Empfang von Satellitendaten	7
3.2 MSG Data Manager	7
3.3 MSG Animator	8
3.4 Bisherige Verarbeitung der Daten	8
3.5 Datenvisualisierung in virtuellen Realitäten	9
4 Rahmenbedingungen	11
4.1 Anforderungen	11
4.2 Zeitplan	13
5 Konzept	15
5.1 Überblick	15
5.2 Auswahl der Satelliten	15
5.2.1 Meteosat 10	17
5.2.2 Meteosat 7	17
5.2.3 Geostationary Operational Environmental Satellite West	17
5.2.4 Geostationary Operational Environmental Satellite East	18
5.2.5 Himawari 8	18
5.3 Übertragung	18
5.4 Verarbeitung	19
5.5 Visualisierung	19
5.6 Interaktion	19

6	Implementierung	21
6.1	Empfang und Entschlüsselung	21
6.2	Übertragung	21
6.2.1	Datenmanagement	21
6.3	Verarbeitung	22
6.3.1	Clipping	23
6.3.2	Neuprojektion	23
6.3.3	Farben verfälschen	25
6.3.4	Kombinieren	28
6.4	Visualisierung	28
6.5	Interaktion	31
6.5.1	Hardware	31
6.5.2	Software	32
7	Evaluation	35
8	Zusammenfassung und Ausblick	37
8.1	Zusammenfassung	37
8.2	Ausblick	38
	Literatur	39
	Anhang	40

Abbildungsverzeichnis

1.1	Oculus Rift mit den 6 Freiheitsgraden X, Y, Z, Yaw, Pitch und Roll .	2
3.1	Zwölf verschiedene Channels zur Auswahl im Manager	8
3.2	Animation mit hochauflösenden Daten des Meteosat-8 und Land/Wasser Farbverfälschung	9
5.1	Konzept der einzelnen Projektkomponenten	15
5.2	Schematische Darstellung des Längengrades der ausgewählten Satel- liten und deren Abdeckung der Erde	16
6.1	Sequenzdiagramm des FileWatchers	22
6.2	Sequenzdiagramm des Bildverzerrers	23
6.3	Bild vor der Bearbeitung	23
6.4	Textur nach dem Projizieren	24
6.5	Textur nach dem Kombinieren	29
6.6	Timeline für den alpha-Wert im Material der Erdkugel mit den Events W, X, Y und Z	30
6.7	Stereoskopische Darstellung des Menüs	31
6.8	Stereoskopische Darstellung der Szene mit Erde und Satelliten im Hintergrund	31
6.9	Wii mit sechs Freiheitsgraden und sieben Buttons zur Eingabe (B auf der Rückseite)	32

Tabellenverzeichnis

4.1	Zeitplan	13
5.1	Position der Satelliten	16
6.1	Eingabemöglichkeiten	29
6.2	Events der Timeline	30
6.3	Eingabemöglichkeiten	33

Akronyme

DVB Digital Video Broadcasting. 7

EUMETSAT European Organization for the Exploitation of Meteorological Satellites. 5, 7, 17

GOES-E Geostationary Operational Environmental Satellite East. 18

GOES-W Geostationary Operational Environmental Satellite West. 17, 18

HCI Human-Computer Interaction. 2, 5, 11, 13, 15, 18, 19, 21, 37, 41

Him8 Himawari 8. 18

HMD Head Mounted Display. 1, 5

HSL Modell Farbmodell auf Basis von Hue (H), Saturation (S), und Lightness (L). 26, 28

HUD Head Up Display. 31

JMA Japan Meteorological Agency. 18

JMU Julius-Maximilians-Universität. iii, 5

JPEG Joint Photographic Experts Group. 21

LNB Low Noise Block. 7

LuRI Luft und Raumfahrt. 2, 3, 5, 8, 11, 13, 15, 18, 21, 37

Met10 Meteosat 10. 7, 17

Met7 Meteosat 7. 17

MSG Meteosat Second Generation. 7, 8

MTSAT-2 Multifunctional Transport Satellite 2. 18

NOAA National Oceanic and Atmospheric Administration of USA. 5, 17, 18

PC Personal Computer. 2, 7, 13, 18, 19, 21, 31, 32

RAM Random Access Memory. 37

TCP Transmission Control Protocol. 21, 37

Acronyms

USB Universal Serial Bus. 32

YCbCr Modell Farbmodell auf Basis von Helligkeit (Y), Blautönen (Cb) und Rottönen (Cr). 25, 26, 28

1 Einleitung

1.1 Wetterbeobachtung

Die Satellitenmeteorologie als Teilgebiet der Meteorologie bereichert diese schon seit langer Zeit. Bereits 1987 berichtet Veronika Zwatz-Meise in ihrem Buch Satellitenmeteorologie: Satelliten beobachten das Wetter von den Einblicken und Methoden dieses Forschungsgebiets, das damals bereits fast 30 Jahre alt und somit heute über 55 Jahre Teil der Wissenschaft ist. Visuelle Aufnahmen der Erde, gepaart mit Informationen zu Wasserdampf und Infrarotwerten helfen so nicht nur das tägliche Wetter vorherzusagen (Zwatz-Meise, 1987), sondern außerdem Klimaveränderungen zu beobachten, Ozeane zu überwachen oder Informationen zu Luftströmungen und Verschmutzung zu sammeln (EUMETSAT, 2016).

Heutzutage ist die Wettervorhersage aus dem alltäglichen Leben nicht mehr wegzudenken und neben neuen technischen Möglichkeiten, wie die Verbreitung der Daten auf Smartphones, werden vermehrt Satelliten in den Orbit gebracht, um die Genauigkeit jener Vorhersagen weiter zu erhöhen und bisher fehlende Regionen der Erde ebenfalls abzudecken (Kettig, 2015). Weiterhin ergibt sich inzwischen die Möglichkeit die Satellitendaten auch für private und universitäre Zwecke zu nutzen, was unter anderem die Durchführung dieser Arbeit ermöglicht hat.

1.2 Oculus Rift

Virtuelle Realität als eine vom Computer simulierte Umgebung, die dem Nutzer das Gefühl gibt in jener präsent zu sein (Desai, Desai, Ajmera und Mehta, 2014), gewann in den letzten Jahren immer mehr an Popularität, unter anderem aufgrund der Tatsache, dass ihre technische Umsetzung möglich wurde. Als ein Beispiel verwendet das **Head Mounted Display (HMD)** Oculus Rift zwei Displays, um jedem Auge einzeln ein leicht verschobenes Bild zu präsentieren, um einen dreidimensionalen, stereoskopischen Eindruck zu erzeugen. Weiterhin besitzt das **HMD** die Möglichkeit

seine Position durch ein Trackingsystem und seine Orientierung durch ein Gyroskop festzustellen (Desai u. a., 2014).

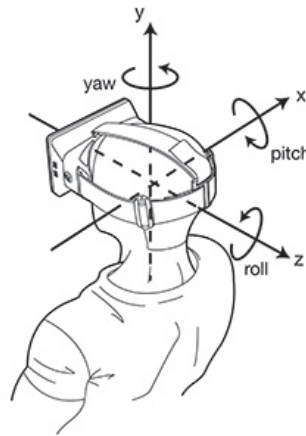


Abbildung 1.1. Oculus Rift mit den 6 Freiheitsgraden X, Y, Z, Yaw, Pitch und Roll

Diese technische Errungenschaft ermöglicht es Nutzer in beliebige Szenen zu versetzen und kann nicht nur in der Spieleindustrie verwendet werden, sondern auch im wissenschaftlichen oder industriellen Kontext zum Einsatz kommen, wie zum Beispiel bei der Fernkontrolle von Robotern (Kot und Novák, 2014).

1.3 Beteiligung der Lehrstühle

An dieser Arbeit sind sowohl der Lehrstuhl für **Luft und Raumfahrt (LuRI)** (Informatik VIII) und der Lehrstuhl für **Human-Computer Interaction (HCI)** (Informatik IX) beteiligt.

Um die Daten der Satelliten empfangen zu können, wird eine Bodenstation benötigt, die der Lehrstuhl für **LuRI** zur Verfügung stellt. Weiterhin verwaltet der Lehrstuhl die Daten und speichert diese für einen Zeitraum von aktuell drei Tagen. Als Ansprechpartner aus diesem Lehrstuhl ist für diese Arbeit M.-Space Tech. Borja Garcia ausgewählt, der dort unter anderem für die Hard- und Software zum Empfangen der Daten verantwortlich ist.

Für die konzeptionelle Betreuung und die Bereitstellung der Hardware ist in dieser Arbeit der Lehrstuhl für **HCI** durch Prof. Marc Erich Latoschik verantwortlich. Neben der Oculus Rift, stellt der Lehrstuhl ebenfalls einen hardwarestarken **Personal Computer (PC)** und eine Wii Mote, da all diese Komponenten für die Umsetzung

nötig sind. Weiterhin erfolgt die Hauptbetreuung durch diesen Lehrstuhl, was sich in regelmäßigen Treffen auszeichnet, um inhaltliche Fragestellungen zu klären.

1.4 Hintergrund der Arbeit

Der Lehrstuhl für **LuRI** empfängt über die bereits erwähnte Bodenstation meteorologische Daten diverser Satelliten. Bisher erfolgte nur eine Livepräsentation auf einem Bildschirm im Mathematik-Gebäude, nun wird jedoch im Zuge dieser Arbeit ein neues Programm zur interaktiven Visualisierung einer virtuellen Erdkugel geschaffen werden. Eine dreidimensionale Repräsentation der Daten wird es den Nutzern ermöglichen diese auf eine neue, natürliche Weise zu sehen und mit ihnen interagieren zu können. Weiterhin wird vor allem auch die Möglichkeit eröffnet, rückwirkend auf die Satellitendaten zugreifen zu können, um so auch Tage später die Daten zu einem bestimmten Zeitpunkt visualisieren zu können.

2 Aufgabenstellung

Als Hauptaufgabe wird in dieser Arbeit ein Programm geschaffen werden, welches die Satellitendaten des Lehrstuhls für **LuRI** dreidimensional darstellt. Hierfür wird auf Methodik und Technik des Lehrstuhls für **HCI** zurückgegriffen. Daher erfolgt die Arbeit in Kooperation zwischen diesen beiden Lehrstühlen der **JMU** in Würzburg. Während erstgenannte vor allem die Daten zur Verfügung stellen, sorgt der zweite Lehrstuhl für Bereitstellung der Hardware und Hauptbetreuung.

Zum Einsatz werden unter anderem die Bilddaten der Organisationen **European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)** und **National Oceanic and Atmospheric Administration of USA (NOAA)** kommen, wobei weitere Quellen denkbar sind. Diese Daten liegen auf einem speziellen Rechner des Lehrstuhls für **LuRI**, wobei neue Bilder alle 15 Minuten bis drei Stunden empfangen werden. Diese Daten werden im Anschluss auf einen Rechner übertragen werden, der sie verarbeitet und über das **HMD** Oculus Rift ausgibt. Weiterhin werden sich die ausgegebenen Daten automatisch mit der Zeit verändern, um eine Animation zu simulieren. Als Eingabekomponente wird ein System verwendet werden, welches Interaktion zur Verschiebung, Rotation und zum Zoomen der virtuellen Erde erlaubt (z.B. WiiMote, Razor Hydra, Trackingsystem, Speech Recognition).

3 Stand der Forschung

3.1 Empfang von Satellitendaten

Wettersatelliten, wie beispielhaft die der Organisation **EUMETSAT**, senden in regelmäßigen Abständen Sets aus Bildern an Bodenstationen. Im Fall des Satelliten **Meteosat 10 (Met10)** empfängt zum Beispiel das zentrale Rechenzentrum in Darmstadt die Rohdaten und verarbeitet diese, ehe sie global über das sogenannte EumetCast-System zur Verfügung gestellt werden. Bodenstationen, wie die der Universität Würzburg, können so über eine Parabolantenne mit **Low Noise Block (LNB)** und einem **PC** mit einer **Digital Video Broadcasting (DVB)**-Karte die Daten empfangen und entschlüsseln (Kettig, 2015).

3.2 MSG Data Manager

Die einzeln entschlüsselten Daten setzt im Anschluss der **Meteosat Second Generation (MSG)** Data Manger (Taylor, 2015) zusammen. Der Manager, der ursprünglich von SatSignal Software in Edinburgh entwickelt wurde, wandelt die komprimierten Daten, die lediglich Zahlen enthalten, als erstes in Bilder um und löscht danach wahlweise die sehr großen Quelldateien. Neben diesem zentralen Schritt erlaubt er es auch auszuwählen, welche der zur Verfügung stehenden Kanäle umgewandelt werden sollen. So bieten die Satelliten Meteosat 8, 9 und 10 zwölf verschiedene Kanäle an, deren Daten sich in Auflösung und Aufnahme (visueller Bereich, Infrarotwerte, Wasserdampf) unterscheiden (siehe Bild 3.1). Neben den europäischen (wie Meteosat), empfängt der **MSG** Data Manager auch Informationen von weltweiten Satelliten. Dies ermöglicht, dass er als Empfänger für sämtliche Daten, die für die Visualisierung der dreidimensionalen Erde dieser Bachelorarbeit benötigt werden, dienen kann.

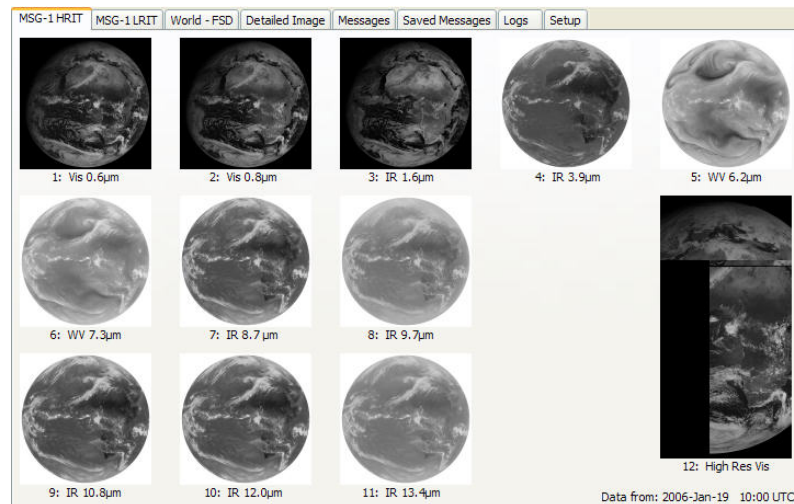


Abbildung 3.1. Zwölf verschiedene Channels zur Auswahl im Manager

3.3 MSG Animator

Neben dem MSG Data Manager, benutzt das Institut für LuRI bisher ebenfalls den MSG Animator (Taylor, 2015). Dieses Programm greift auf die Daten des erstgenannten zurück, um zweidimensionale Echtzeitanimationen zu erzeugen. Darüber hinaus besitzt es die Fähigkeit die Farben der schwarzweißen Aufnahmen der Satelliten zu verfälschen, um so farbige Darstellungen der Erde zu erhalten. Die genaue Vorgehensweise wurde von den Autoren und Programmierern jedoch nicht veröffentlicht, sodass im Zuge dieser Bachelorarbeit nicht darauf zurückgegriffen werden kann. Da der Animator weiterhin nur zweidimensionale Bilder erzeugt, wird er für die Visualisierung der dreidimensionalen Erde nicht weiterverwendet. Bild 3.2 zeigt einen Ausschnitt aus einer Animation mit visuellen Daten und Farbverfälschung, die der MSG Animator produzieren kann.

3.4 Bisherige Verarbeitung der Daten

Neben der Darstellung der Daten über den MSG Animator, verwendet der Lehrstuhl für LuRI bereits andere Programme, um die Daten zu verarbeiten und auszuwerten. So werden beispielsweise die Bilddaten der Meteosat Wettersatelliten automatisch auf Unregelmäßigkeiten überprüft. Es gäbe immer noch Unklarheiten über viele Phänomene, die sich in der Erdatmosphäre abspielen und deshalb sei es wichtig

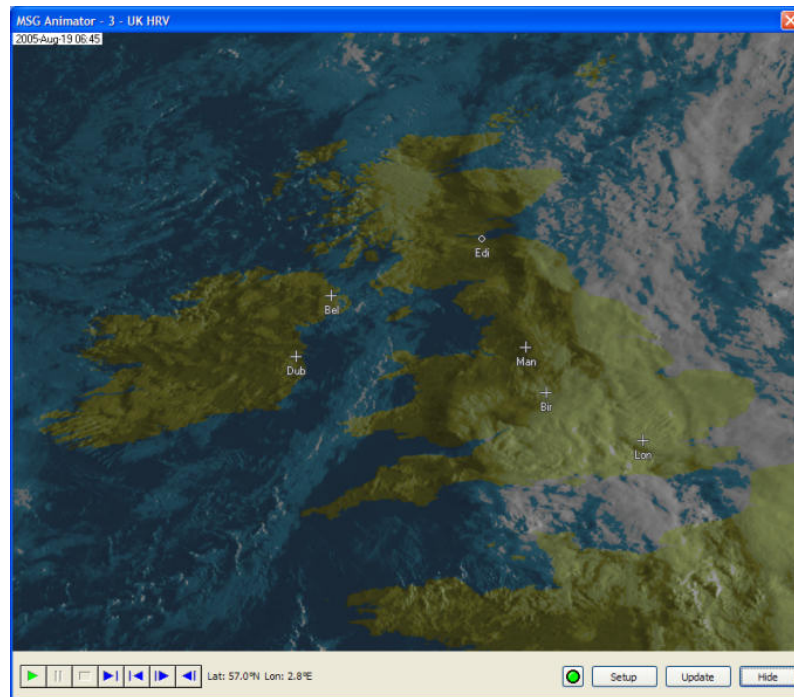


Abbildung 3.2. Animation mit hochauflösenden Daten des Meteosat-8 und Land/Wasser Farbverfälschung

die Bilddaten von Wettersatelliten nach spezifischen Eigenschaften von Anomalien zu untersuchen. Eine automatische Erkennung dieser Unregelmäßigkeiten würde es Menschen deutlich erleichtern, jene später zu untersuchen (Kettig, 2015). Das geschaffene Programm speichert lokal seine Ergebnisse, zeigt diese in einer Benutzeroberfläche an und versendet nach Wahl Emails an eingetragene Nutzer.

3.5 Datenvisualisierung in virtuellen Realitäten

Mit immer größeren Datenmengen steigt das Problem diese alle zu verstehen beziehungsweise die wichtigen Informationen aus dieser Menge zu entnehmen. Neben der klassischen zweidimensionalen Visualisierung mit zum Beispiel Graphen, Karten und Bildern, ermöglicht die Technik heutzutage eine dreidimensionale Darstellung innerhalb einer virtuellen Realität. Tom Erickson argumentiert unter anderem, dass Visualisierung mit Manipulation und Interaktion verbunden sei. Eigenschaften, die eine virtuelle Realität durch ihre hohe Interaktion, direkte Feedbackrückkopplung und auch durch ihr Embodiment unterstütze (Stone, Erickson, Bederson, Rothman und Muzzy, 1994).

Auch in der Kartographie werden bereits virtuelle Realitäten zur Visualisierung von Daten verwendet. So erhöhe die dreidimensionale Darstellung von Objekten durch virtuelle Realität das Level der Wahrnehmung und dynamische Änderungen, wie Regen oder Wolken, sollen es der Karte der Zukunft ermöglichen, die Aufmerksamkeit des Nutzers gezielter zu steuern. Außerdem helfe die Tatsache, dass Nutzer in der virtuellen Realität einen Großteil, wenn nicht sogar die gesamte Szene überblicken können, sich zurecht zu finden. Die Verarbeitung dreidimensionaler Strukturen und Szenen sei zwar komplexer als deren zweidimensionale Gegenstücke, da wir jedoch in einer dreidimensionalen Welt leben, könne unser Gehirn jene Darstellung besser verstehen (Bidoshi, [2003](#)).

4 Rahmenbedingungen

4.1 Anforderungen

Im Zuge dieser Arbeit werden die folgenden Funktionen und Anforderungen umgesetzt, um die in Kapitel 2 genannten Ziele zu erreichen:

AK1 Auswahl der zu darstellenden Daten:

AK1.1 Auswahl geeigneter Satelliten unter Berücksichtigung einer möglichst vollständigen Abdeckung der Erde

AK1.2 Auswahl geeigneter Kanäle der Satelliten, da diese neben einer Verarbeitung im visuellen Bereich auch zum Beispiel Infrarotbilder senden

AK2 Kommunikation der Daten zwischen dem Empfangsrechner des Instituts für **LuRI** und dem Rechner des Lehrstuhls für **HCI**, der die Daten verarbeitet und ausgibt:

AK2.1 Auswahl und Implementierung einer Überwachung des Empfangsrechners, um neue Bilder direkt zu versenden

AK2.2 Auswahl des Übertragungsweges unter Berücksichtigung der Tatsache, dass die Satellitenaufnahmen das Netzwerk der Universität nicht verlassen dürfen

AK2.3 Auswahl und Implementierung einer Überwachung des Verarbeitungs- und Ausgaberechners, um neue Bilder direkt in eine Textur umwandeln zu können

AK3 Verarbeitung (mehrerer) Satellitenbilder zu einer Textur, die auf eine Kugel projiziert werden kann:

AK3.1 Änderung der Projektion der Satellitenbilder, um diese von ihrer azimuthalen Form in eine sphärische zu tauschen

AK3.2 Kombination mehrerer neu projizierter Satellitenbilder zu einer Textur, die die gesamte Erde abdeckt

- AK3.3** Auswahl und Implementierung einer Möglichkeit der Farbverfälschung der schwarzweißen Textur, um eine kolorierte Version der Erde zu bekommen
- AK4** Erzeugen einer Speicherstruktur, die es erlaubt beliebig viele Texturen zu beliebig vielen Zeitpunkten zu hinterlegen, und zur Laufzeit des Programms wieder abzurufen
- AK5** Ausgabe der Daten mithilfe Technologie der immersiven virtuellen Realität:
 - AK5.1** Auswahl der Engine, um die Erde darzustellen
 - AK5.2** Anschluss einer Oculus Rift, um die Szene als immersive virtuelle Realität darzustellen
 - AK5.3** Implementierung einer Textur, die sich während der Laufzeit ändert, um den zeitlichen Verlauf zwischen den Zeitpunkten der Texturen zu simulieren
- AK6** Anschluss einer Interaktionskomponente, die es Nutzern erlaubt mit der virtuellen Erde zu interagieren:
 - AK6.1** Auswahl der Hardware für die Interaktion
 - AK6.2** Implementierung einer räumlichen Interaktion mit der Erde im dreidimensionalen Raum
 - AK6.3** Implementierung einer zeitlichen Interaktion mit der Erde, um die Auswahl des zu betrachtenden Zeitpunkts wählen zu können

Besondere Schwierigkeiten ergeben sich dabei im Bezug auf die folgenden Punkte:

- S1** Die meisten Satelliten senden Daten in unterschiedlichen Intervallen. Dadurch kann es zu Diskontinuitäten in der Textur kommen. (**AK3.2**)
- S2** Es ist nicht gesichert, dass eine Auswahl getroffen werden kann, welche die Erde vollständig abdeckt. (**AK1.1**)
- S3** Da die Daten rechtlich geschützt sind, müssen diese zu jeder Zeit innerhalb der Universität bleiben. (**AK2.2**)
- S4** Das Projizieren ist ein komplexes Problem, bei welchem vor allem die Randbereiche der Aufnahmen stark verzerrt werden. (**AK3.1**)
- S5** Das Zusammenfügen einzelner Aufnahmen von unterschiedlichen Satelliten soll nach Möglichkeit keine Diskontinuitäten zeigen. Denkbar wäre außerdem, dass dieser Schritt von der Ausgabeengine übernommen wird. (**AK3.2**)

S6 Es wird Stellen und Zeitpunkte geben, zu denen keine Textur vorliegt. (**AK3.2**)

S7 Eine 4D-Interaktion (Raum- und Zeitverschiebung) ist eine komplexe Interaktion, die trotzdem intuitiv umgesetzt sein soll. (**AK6**)

4.2 Zeitplan

Der Zeitplan gibt an welche Aufgaben in welcher Woche erledigt werden. Der Bearbeitungszeitraum erstreckt sich über mehr als zehn Wochen, da lediglich eine Teilzeitbearbeitung möglich ist.

Tabelle 4.1. Zeitplan

Arbeit	Anforderung	Aufwand
Einarbeitung Unreal	AK5.1	1 Woche
Ausschluss der Oculus	AK5.2	1/2 Woche
Literatursuche 1: Thema Visualisierung	-	1 Woche
Auswahl der Satelliten	AK1.1	1/2 Woche
Auswahl der Kanäle	AK1.2	1/2 Woche
Einarbeitung, um Daten zu kopieren	AK2.2	1 Woche
Programm zur Projektion einbauen	AK3.1	1 Woche
Programm, um Kombination erweitern	AK3.2	1/2 Woche
Programm, um Farbverfälschung erweitern	AK3.3	1/2 Woche
Darstellung der Erde mit Beispieldaten	AK5.3	1 Woche
Literatursuche 2: Thema Interaktionen	AK6	1 Woche
Eingabehardware anschließen	AK6.1	1/2 Woche
Räumliche Interaktion einbauen	AK6.2	1/2 Woche
Programme auf dem LuRI PC einrichten	AK2.1	1 Woche
Verarbeitung auf dem HCI PC automatisieren	AK2.3	1 Woche
Anbindung der Datenstruktur an Unreal	AK4	1 Woche
Zeitliche Interaktion einbauen	AK6.3	2 Wochen
Puffer für unvorhergesehene Probleme	-	1 Woche
Schriftliche Teile finalisieren	-	1 Woche
Abgabe	-	Mitte April

5 Konzept

Um das Gesamtprojekt zu meistern, ist es, wie Abbildung 5.1 zeigt, in einzelne Projektkomponenten zerlegt. Diese gestalten sich modular, sodass sie unabhängig voneinander bearbeitet werden können.

5.1 Überblick

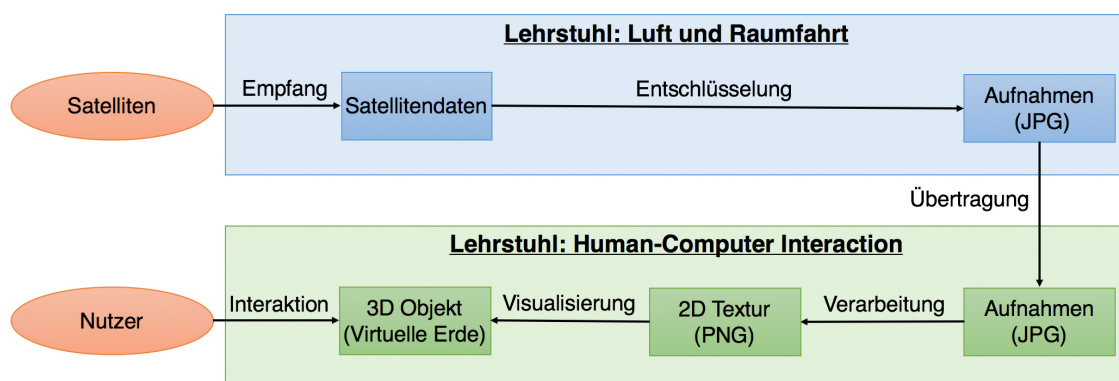


Abbildung 5.1. Konzept der einzelnen Projektkomponenten

Die Phasen Empfang und Entschlüsselung sind nicht Teil der Arbeit und sind bereit kurz in Kapitel 3.1 angesprochen. Dahingegen finden sich Informationen zu den ausgewählten Satelliten in Kapitel 5.2 sowie das Konzept für das Programm der Datenübertragung in Kapitel 5.3. Die Aufgaben, die sich eher auf Seiten des Lehrstuhls für HCI befinden, sind durch die Erstellung der Texturen (siehe Kapitel 5.4) sowie die Darstellung der virtuellen Szene (siehe Kapitel 5.5) und die Interaktion mit jenen (siehe Kapitel 5.6) vollständig in dieser Arbeit aufgeführt.

5.2 Auswahl der Satelliten

In Kooperation mit dem Institut für LuRI sind fünf Wettersatelliten ausgewählt, die sich in einem geostationären Orbit befinden (AK1.1 und S2). Die genauen Daten

sind in Tabelle 5.1 gelistet.

Tabelle 5.1. Position der Satelliten

Satellit	Breitengrad	Längengrad	Bildfrequenz
Met10	0	0	15 Minuten
Met7	0	57.5E	1 Stunde
GOES-W	0	135W	1 Stunde (Vollständig nur 3 Stunden)
GOES-E	0	75W	1 Stunde (Vollständig nur 3 Stunden)
Him8	0	140.7E	30 Minuten

Eine schematische Darstellung wie in Abbildung 5.2 zeigt, dass durch diese Auswahl eine nahezu komplette Darstellung der Erde möglich ist, da die fünf Satelliten die Erde aus jeder Position bei null Grad Breite aufnehmen. Außerdem gewährleistet die Überlappung der einzelnen Bilder, dass Fehler aufgrund von starker Verzerrungen am Rand der Aufnahmen minimiert werden können. Auf der anderen Seite bedeutet dies jedoch auch, dass am Nord- und Südpol keine Daten zur Verfügung stehen werden. Pro Satellit werden die Daten eines visuellen Kanals pro benutzt (AK1.2).

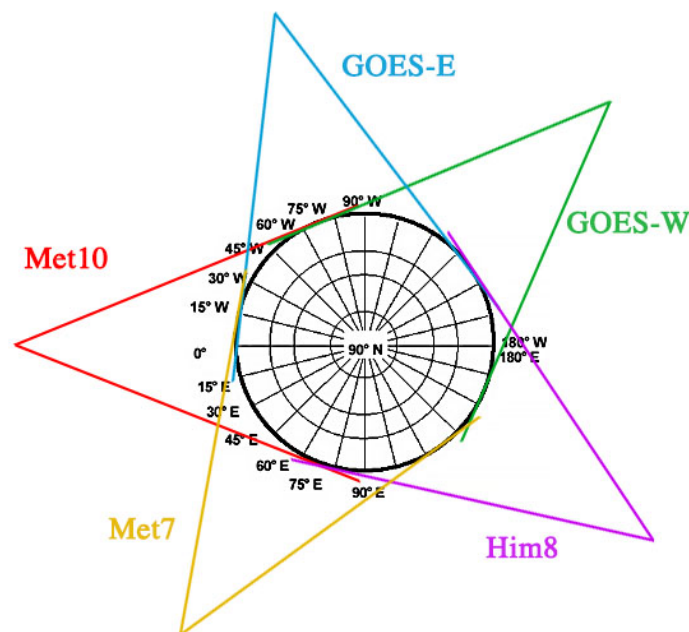


Abbildung 5.2. Schematische Darstellung des Längengrades der ausgewählten Satelliten und deren Abdeckung der Erde

5.2.1 Meteosat 10

Der **Met10** ist Teil der zweiten Generation an Wettersatelliten, die durch das Institut **EUMETSAT** betrieben werden. Mit einem Arbeitsbeginn am 05. Juli 2012 (Launch im Guiana Space Centre in Kourou), wird der **Met10** laut Angaben vermutlich bis 2022 weiterarbeiten können. In einer Höhe von 36000 km über dem Äquator nimmt dieser Satellit neben visuellen Aufnahmen auch Informationen im Infrarotbereich und Daten über Wasserdampf auf. Die Hauptverwendung für die Aufnahmen durch diesen Satelliten ist vor allem die (auch kurzfristige) Vorhersage von Wetter. Hierzu sendet der **Met10** alle 15 Minuten ein vollständiges Bild der Erde (EUMETSAT, 2015).

5.2.2 Meteosat 7

Der **Meteosat 7 (Met7)** ist ebenfalls Teil des Instituts **EUMETSAT**, gehört jedoch noch der ersten Generation an Wettersatelliten an. Er begann seine Arbeit am 02. September 1997, wird jedoch erst seit dem 01. November 2006 über dem Indischen Ozean eingesetzt. Seine Arbeit soll 2016 eingestellt werden, sodass er 2017 seinen stationären Orbit verlassen kann. Auch dieser Satellit dient zur Wettersvorhersage und deckt mit dem Indischen Ozean die östlichsten Bereiche der Satelliten vom Institut **EUMETSAT** ab. Er produziert jede Stunde einen vollständigen Scan (EUMETSAT, 2015).

5.2.3 Geostationary Operational Environmental Satellite West

Der **Geostationary Operational Environmental Satellite West (GOES-W)** oder auch **GOES-15** ist Teil der Organisation **NOAA** und wird von den Vereinigten Staaten Amerikas operiert. In einem geostationären Orbit, ca. 35780 km über dem Äquator bei 135 Grad West, überwacht dieser Nordamerika und Teile des pazifischen Ozeans. Der **GOES-W** wurde am 4. März 2010 von der Cape Canaveral Air Station aus in den Orbit geschossen. Er produziert jede Stunde ein Bild der Erde, dieses ist jedoch nur alle drei Stunden vollständig (NOOA, 2014).

5.2.4 Geostationary Operational Environmental Satellite East

Ähnlich zum [GOES-W](#), ist der [Geostationary Operational Environmental Satellite East \(GOES-E\)](#) oder GOES-13 ebenfalls ein Wettersatellit der Organisation [NOAA](#) und überwacht für die USA Nord- und Südamerika sowie den atlantischen Ozean. Mit ähnlichen Spezifikationen wie der [GOES-W](#), sendet auch dieser Satellit jede Stunde ein Teil-, beziehungsweise alle drei Stunden ein vollständiges Bild mit der Position 75 Grad West, 0 Grad Nord im Zentrum ([NOOA, 2014](#)).

5.2.5 Himawari 8

Zu Beginn dieser Arbeit wurde der Satellit [Multifunctional Transport Satellite 2 \(MTSAT-2\)](#) ausgesucht, der auch unter dem Namen Himawari 7 bekannt ist und sich bei 145 Grad Ost auf einer Höhe von 35800 km über dem Äquator ([Japan Meteorological Agency, 2015b](#)) befand. Die [Japan Meteorological Agency \(JMA\)](#) ersetzte diesen Satelliten jedoch am 4. Dezember durch den neuen [Himawari 8 \(Him8\)](#), sodass auch in dieser Arbeit auf den neuen Satelliten gewechselt wird. Dieser befindet sich bei ca. 140.7 Grad Ost über dem Äquator und sendet alle dreißig Minuten eine Aufnahme ([Japan Meteorological Agency, 2015a](#)).

5.3 Übertragung

Die Daten werden im Anschluss durch ein Programm vom [LuRI PC](#) auf den [HCI PC](#) kopiert ([AK2.1](#) und [AK2.3](#)) werden, wobei die Übertragung über das Netzwerk der Universität erfolgt ([AK2.2](#) und [S3](#)).

Eine Alternative wäre es, dass der [HCI PC](#) lediglich alle 15 Minuten überprüft, ob neue Daten vorhanden sind, da dies die schnellste Frequenz neuer Bilder ist ([AK2.3](#)).

Im Zuge dieser Arbeit wird jedoch die erste Methode implementiert, da diese im Vergleich zur zweiten kein Polling verwendet und Daten sofort verschickt, sobald dieser zur Verfügung stehen.

5.4 Verarbeitung

Auf dem **HCI PC** angekommen, werden die Bilder mithilfe einer selbstgeschriebenen Anwendung zuerst in die sphärische Projektion gebracht (**AK3.1**). Der Verzerrung an den Randbereichen der Aufnahme wird dabei entgegengewirkt, da Daten von genug Satelliten zur Verfügung stehen und sich deren Bilder stark genug überlappen (**S4**). Diese Projektionen werden als Einzeldateien in einer Ordnerstruktur hinterlegt (**AK4**). Zu jedem Zeitpunkt, an welchem Daten vorhanden sind, werden diese zu einer Textur kombiniert (**AK3.2** und **S1** und **S5**) und mit einer Farbvorlage verbunden (**AK3.3**). Stellen ohne Daten werden dabei gesondert markiert (**S6**). Weiterhin würden sich die Möglichkeiten ergeben, dass die Engine die einzelnen Texturen kombiniert oder ein externes Programm über Stitching diese zusammenfügt. (**AK3.2** und **S1** und **S5**).

Das finale Konzept setzt die erste Idee um, da so die kompletten Texturen abgespeichert werden können. Dies entlastet die Engine, die dann lediglich die fertigen Daten lädt.

5.5 Visualisierung

In einem letzten Schritt greift die Engine Unreal 4 (**AK5.1**) bei der Darstellung auf die Texturen zu und projiziert diese auf eine Kugel innerhalb einer dreidimensionalen Szene. Die Engine ersetzt in regelmäßigen Abständen die Textur der Erde mit der nachfolgenden (**AK5.3**).

5.6 Interaktion

Der Nutzer sieht die Szene durch eine Oculus Rift, welche über das Development Kit 2 mit der Engine verbunden ist (**AK5.2**). Die Interaktion erfolgt mit einer Wii Mote (**AK6.1** und **S7**), die hierfür sowohl Knöpfe als auch ein Gyroskop zur Verfügung stellt (**AK6.2** und **AK6.3**). Zur Auswahl stehen bei dieser Eingabemethode eine Interaktion 0. oder 1. Ordnung, da beide auf den ersten Blick intuitiv erscheinen. Bei erstgenannter würde die Rotation der Erde immer der Rotation der Wii Mote entsprechen, während bei zweitgenannter die Geschwindigkeit der Rotation der Erde über die Rotation der Wii Mote kontrolliert wird.

Eine mögliche alternative Interaktion wäre eine Kugel, die der Nutzer selbst in der Hand hält. Die räumliche Manipulation entsteht durch direkte Bewegung dieser Kugel (entspricht 0. Ordnung für die Wii Mote). Für die zeitliche Interaktion müssten jedoch zwei Knöpfe vorhanden sein (**AK6** und **S7**).

Die Interaktion wird im Zuge dieser Arbeit durch die Wii Mote Steuerung 1. Ordnung realisiert. Neben den sechs Freiheitsgraden, die die Wii Mote durch Beschleunigung in die jeweilige Richtung erkennt, stellt sie ebenfalls ein Steuerkreuz sowie sieben Buttons zur Verfügung, was vor allem in der Menüauswahl unverzichtbar ist. Daraus ergab sich vor allem der Vorzug im Vergleich zur Kugel. Weiterhin ist die Idee der 0. Ordnung in dieser Arbeit verworfen, da diese dazu führen würde, dass die Wii Mote in bestimmten gedrehten Position verbleiben muss, wenn Nutzer bestimmte Orte der virtuellen Erde länger betrachten wollen.

Für die zeitliche Steuerung wird dem Nutzer einerseits ein Startmenü zur Verfügung stehen, um einen Zeitrahmen auszuwählen und andererseits die Möglichkeit geboten werden die Geschwindigkeit der Animation zu verändern, bis hin zu einer Pause und einen Modus, in welchem die Zeit rückwärts verläuft.

6 Implementierung

Im Folgenden werden die einzelnen Schritte, wie sie im konzeptionellen Teil genannt wurden, erneut aufgegriffen, um die konkrete Umsetzung in dieser Arbeit zu erläutern.

6.1 Empfang und Entschlüsselung

Die Aufgaben Empfang und Entschlüsselung sind nicht Teil dieser Arbeit und werden vom Lehrstuhl für LuRI übernommen. Dies stellt keine zusätzlichen Anforderungen, da die Daten bereits für andere Anwendungen bis ins Joint Photographic Experts Group (JPEG) Format aufbereitet werden.

6.2 Übertragung

Ein Java Programm kopiert und verschickt mithilfe einer Transmission Control Protocol (TCP) Verbindung die Bilddaten vom Rechner des Lehrstuhls für LuRI über das Netzwerk der Universität. Es basiert dabei auf den Klassen innerhalb des Pakets java.nio.file, welche unter anderem einen Pfad auf Dateiänderungen überwachen können. Da beim Entschlüsseln der Satellitendaten ein neues Bild im JPEG Format generiert wird, aktiviert dies ebenfalls den Watcher, der daraufhin das Bild, die Aufnahmezeit und den Namen des Aufnahmesatelliten verschickt.

6.2.1 Datenmanagement

Sobald die Daten am PC des Lehrstuhls für HCI empfangen werden, wird das Bild in einer speziellen Ordnerstruktur abgelegt. Diese sieht den Aufbau wie folgt vor: root/year/month/day/timeOfDay/. Im Anschluss wird eine Textur auf Basis aller Satellitenbilder im aktuellen Order erstellt. Sollte bereits eine Textur vorhanden

sein, so wird diese lediglich überschrieben, da jene zu einem frühen Zeitpunkt erstellt wurde und nun mehr Daten bzw. Satellitenbilder vorliegen. Die Sequenzdiagramme in 6.1 veranschaulichen den Aufbau dieses Programms.

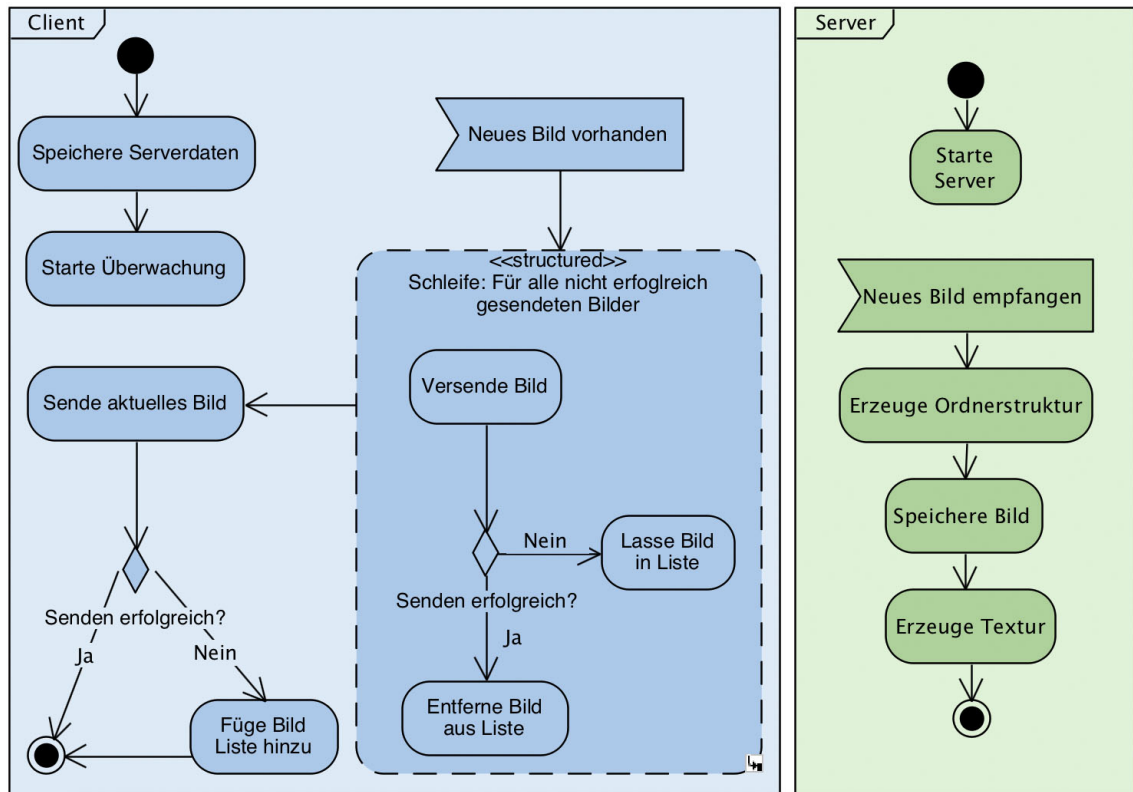


Abbildung 6.1. Sequenzdiagramm des FileWatchers

6.3 Verarbeitung

Das Programm, welches die Texturen erstellt, verarbeitet die Satellitenaufnahmen in mehreren Schritten, wie Sequenzdiagramm 6.2 zeigt.

Zu Beginn werden die Daten geladen. Um dann die Bilder der Satelliten zu einer Textur kombinieren zu können, werden diese zuerst zurechtgeschnitten (siehe Kapitel 6.3.1), bevor sie in eine sphärische Projektion gebracht (siehe Kapitel 6.3.2) und mit einer Farbvorlage kombiniert werden können (siehe Kapitel 6.3.3). Im Anschluss werden mehrere solcher neu projizierter Bilder zu einer Textur verbunden (siehe Kapitel 6.3.4), die dann gespeichert wird.

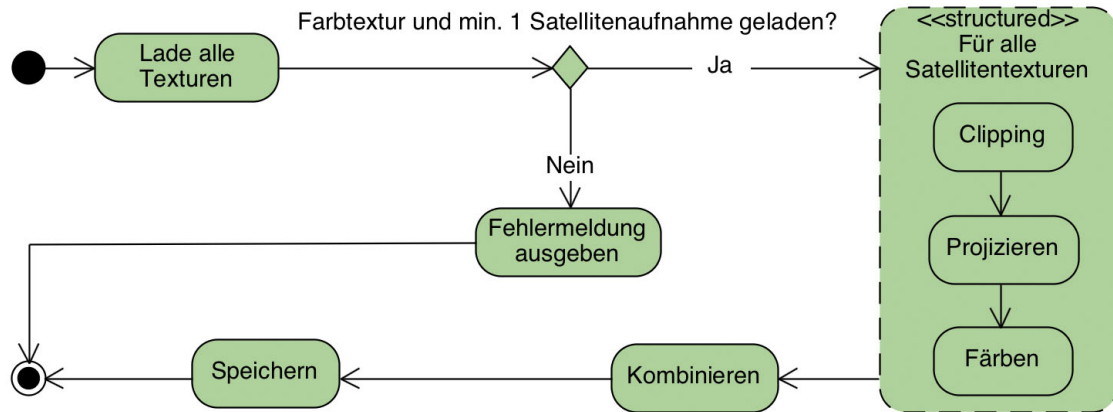


Abbildung 6.2. Sequenzdiagramm des Bildverzerrers

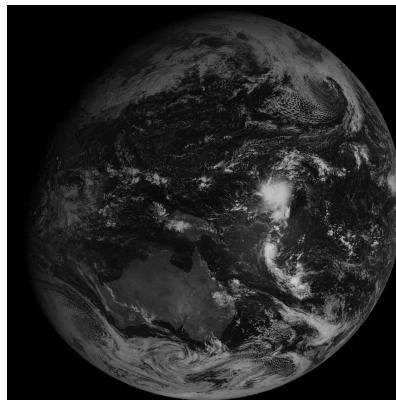


Abbildung 6.3. Bild vor der Bearbeitung

6.3.1 Clipping

Im ersten Schritt werden die Bilder der Satelliten auf ein quadratisches Format geschnitten, sodass die Erde gerade in das Bild passt. Dies ist nötig, da das Programm im nächsten Schritt davon ausgeht, dass der Radius der Erde die halbe Bildhöhe bzw. Bildbreite ist. Weiterhin wird so sichergestellt, dass der einzige Punkt, von dem das Programm den Längen- und Breitengrad ohne weitere Berechnung kennt, sich in der Mitte des Bildes befindet. Dies ist gewährleistet, weil alle verwendeten Satelliten sich im geostationären Orbit befinden.

6.3.2 Neuprojektion

Für jedes Bild eines Satelliten wird in diesem Schritt ein neues angelegt, das doppelt so breit wie hoch ist. Dieses neue Bild stellt eine sphärische Projektion dar, wobei

sich links oben -180 Grad Breite und -90 Grad Länge und rechts unten +180 Grad Breite und +90 Grad Länge befinden. Für jeden Pixel (an der Position x und y) dieses neuen Bildes (der Höhe *image.height* und Breite *image.width*) wird nach den Formeln (6.1) und (6.2) zuerst der entsprechende Längengrad ϕ und Breitengrad λ berechnet.

$$\phi = \frac{y}{\text{image.height}} * 180 - 90; \quad (6.1)$$

$$\lambda = \frac{x}{\text{image.width}} * 360 - 180; \quad (6.2)$$

Die Gradzahlen ϕ und λ werden im Anschluss über die Formeln (6.3) und (6.4) in Koordinaten für das azimuthal projizierte Bild (x und y) umgerechnet (Weisstein, o.D.). Dabei gilt, dass das Pixel in der Mitte des azimuthal projizierten Bildes den Längengrad ϕ_0 und Breitengrad λ_0 hat.

$$x = k' * \cos(\phi_0) * \sin(\lambda - \lambda_0) \quad (6.3)$$

$$y = k' * (\cos(\phi_0) * \sin(\phi) - \sin(\phi_0) * \cos(\phi) * \cos(\lambda - \lambda_0)) \quad (6.4)$$

Mit diesen Pixelkoordinaten kann dann im Satellitenbild der Farbwert nachgeschlagen und in das neue Bild übertragen werden.



Abbildung 6.4. Textur nach dem Projizieren

6.3.3 Farben verfälschen

Im Anschluss wird die schwarzweiße Textur mit einer farbigen Vorlage verbunden. Diese Vorlage folgt den gleichen Rahmenbedingungen. So beginnt diese beispielsweise ebenfalls links oben bei -90 Grad Breite und -180 Grad Länge. Damit beschreiben Pixel, die an der gleichen X und Y Koordinate liegen auf beiden Bildern den gleichen Punkt.

Im Zuge dieser Arbeit sind verschiedene Methoden exploriert, um die beiden Bilder zu verbinden. Es folgt nun eine genauere Beschreibung von fünf Möglichkeiten, aus denen im Anschluss eine gewählt wird (siehe 6.3.3).

Verfälschung basierend auf dem YCbCr Modell

Für diese Verbindung werden die Farbwerte (r = Rot, g = Grün, b = Blau) der Farbvorlage und des schwarzweißen Satellitenbilds mithilfe der Formeln (6.5), (6.6), und (6.7) in Werte des Farbmodell auf Basis von Helligkeit (Y), Blautönen (Cb) und Rottönen (Cr) (YCbCr Modell)s umgewandelt (Clarke, 2011). Für jeden Pixel ($final$) des finalen Bildes wird anschließend ein neuer Farbwert gebildet, der aus dem Y-Wert des Satellitenbilds (Y_{sw}) sowie dem Cb- und Cr-Wert der Farbvorlage (Cb_{farbe} und Cr_{farbe}) besteht. Somit wird zwar überall die Farbe ins finale Bild übernommen, jedoch an Stellen mit Wolken stark aufgehellt bzw. an Stellen mit fehlenden Daten stark verdunkelt. Das Ergebnis dieses Vorgehens ist in Abbildung 6.5a zu sehen.

$$Y = 0.257 * r + 0.504 * g + 0.098 * b + 16 \quad (6.5)$$

$$Cb = -0.148 * r - 0.291 * g + 0.439 * b + 128 \quad (6.6)$$

$$Cr = 0.439 * r - 0.368 * g - 0.071 * b + 128 \quad (6.7)$$

$$final = (Y_{sw}, Cb_{farbe}, Cr_{farbe}) \quad (6.8)$$

Verfälschung basierend auf einzelne Kanäle

Bei diesem Vorgehen werden die Farbkanäle einzeln kombiniert. Für jeden Pixel (*final*) des Zielbilds wird als erstes der Rot-, Grün- und Blauwert aus der Farbvorlage und dem schwarzweißen Satellitenbild bestimmt. Sind an einer Stelle keine Daten vorhanden und damit die Helligkeit des Satellitenbilds ($Helligkeit_{sw}$) gering (schwarz), so wird für jeden Farbkanal der kleinere Wert ($\min(\dots)$) aus beiden Bildern ($quelle_{sw}$ oder $quelle_{farbe}$) übernommen. Andernfalls wird der größere Wert ($\max(\dots)$) genutzt (vergleiche exemplarisch (6.9)). Dunkle Stellen übernehmen somit geringe Werte und bleiben schwarz, während helle Stellen die größeren Werte übernehmen und damit eingefärbt werden. Das Ergebnis ist sichtbar in Abbildung 6.5b.

$$final = \begin{cases} \min(quelle_{sw}, quelle_{farbe}) & \text{if } Helligkeit_{sw} \text{ gering} \\ \max(quelle_{sw}, quelle_{farbe}) & \text{Sonst} \end{cases} \quad (6.9)$$

Verfälschung basierend auf dem HSL Modell

Ähnlich zu der Kombination mit dem YCbCr Modell wird auch mit dieser Methode das finale Bild erzeugt. Der einzige Unterschied ist das Farbmodell, welches hierbei das Farbmodell auf Basis von Hue (H), Saturation (S), und Lightness (L) (HSL Modell) ist. Berechnet wird dies mit den Formeln (6.10), (6.11) und (6.12), wobei $\min$ und $\max$ der jeweils kleinste bzw. größte Wert aus den drei Farbkanälen (r = Rot, g = Grün, b = Blau) darstellt (Camick, 2015). Wiederum wird für jeden Pixel (*final*) Helligkeit (l) aus dem schwarzweißen Satellitenbild (l_{sw}) und Farbwerte (h und s) aus der Farbvorlage (h_{farbe} und s_{farbe}) zu einem neuen Farbwert kombiniert. Dabei ergibt sich ein finales Bild wie in Abbildung 6.5c zu sehen ist.

$$h = \begin{cases} 0 & \text{if } \max = \min \\ ((60 * (g - b) / (\max - \min)) + 360) \% 360 & \text{if } \max = r \\ (60 * (b - r) / (\max - \min)) + 120 & \text{if } \max = g \\ (60 * (r - g) / (\max - \min)) + 240 & \text{Sonst} \end{cases} \quad (6.10)$$

$$l = (\max + \min) / 2; \quad (6.11)$$

$$s = \begin{cases} 0 & \text{if } \max = \min \\ (\max - \min)/(\max + \min); & \text{if } l \leq 0.5 \\ (\max - \min)/(2 - \max - \min); & \text{Sonst} \end{cases} \quad (6.12)$$

$$final = (h_{farbe}, s_{farbe}, l_{sw}) \quad (6.13)$$

Verfälschung basierend auf der Methode Hartes Licht

Diese Methode verbindet zwei Ebenen, indem die Farbkanäle einzeln zu einer neuen Farbe (*final*) multipliziert werden. Ist die Helligkeit der schwarzweißen Satellitenaufnahme (*quelle_{sw}*) dabei gering, wird positiv mit der Farbvorlage (*quelle_{farbe}*) multipliziert, wie die Formel (6.14) beispielhaft zeigt. Dies führt zu einer Abdunklung der Farben. Ist hingegen die Aufnahme hell, so wird negativ multipliziert, wie die Formel (6.15) exemplarisch andeutet, wodurch das Bild aufgehellt wird. Dunkle Bereiche mit fehlenden Daten werden dadurch dunkler, helle Bereiche mit Wolken gleichzeitig heller. Das Ergebnis in Abbildung 6.5d zeigt, dass der Eindruck eines härteren Lichts entsteht (Altmann, o.D.).

$$final = quelle_{sw} * quelle_{farbe} \quad (6.14)$$

$$(1 - final) = (1 - quelle_{sw}) * (1 - quelle_{farbe}) \quad (6.15)$$

Verfälschung basierend auf einfacher Helligkeit

Diese sehr einfache Methode übernimmt für jeden Pixel (*final*) den Farbwert aus dem schwarzweißen Satellitenbild (*quelle_{sw}*), wenn dieses sehr dunkel (schwarz, keine Daten) oder hell (weißgrau, Wolken) ist (*Helligkeit_{sw}*). Andernfalls übernimmt es den Farbwert aus der farbigen Vorlage (*quelle_{farbe}*), wie Formel (6.16) zeigt. Das Ergebnis ist in Abbildung 6.5e festgehalten.

$$final = \begin{cases} quelle_{sw} & \text{if } Helligkeit_{sw} \text{ gering} \\ quelle_{sw} & \text{if } Helligkeit_{sw} \text{ groß} \\ quelle_{farbe} & \text{Sonst} \end{cases} \quad (6.16)$$

Finale Auswahl

Wie die Bilder in Abbildung 6.5 zeigen, haben die Kombinationsmöglichkeiten über einzelne Farbkanäle und einfache Helligkeit das Problem, dass die Wolken lediglich grau sind und abgeschnitten werden, wenn sie sich der Tag/Nacht-Grenze an der Textur nähern. Dies liegt an der Tatsache, dass die Wolken an diesen Stellen in der schwarzweißen Satellitenaufnahme bereits zu dunkel sind. Auch die Idee mit hartem Licht hat dieses letztgenannte Problem, aber durch die helleren Wolken fällt dies weniger auf. Gute Ergebnisse erzielen hingegen das **HSL Modell** und das **YCbCr Modell** mit dem wichtigen Unterschied, dass fehlende Daten beim erstgenannten komplett schwarz dargestellt werden, während bei der anderen Möglichkeit die Erdtextur stark verdunkelt zu sehen ist.

Für die weitere Implementierung wurde daher die Methode auf Basis des **YCbCr Modells** gewählt, da es die Wolken gut darstellt und nicht die Hälfte der Textur schwarz lässt.

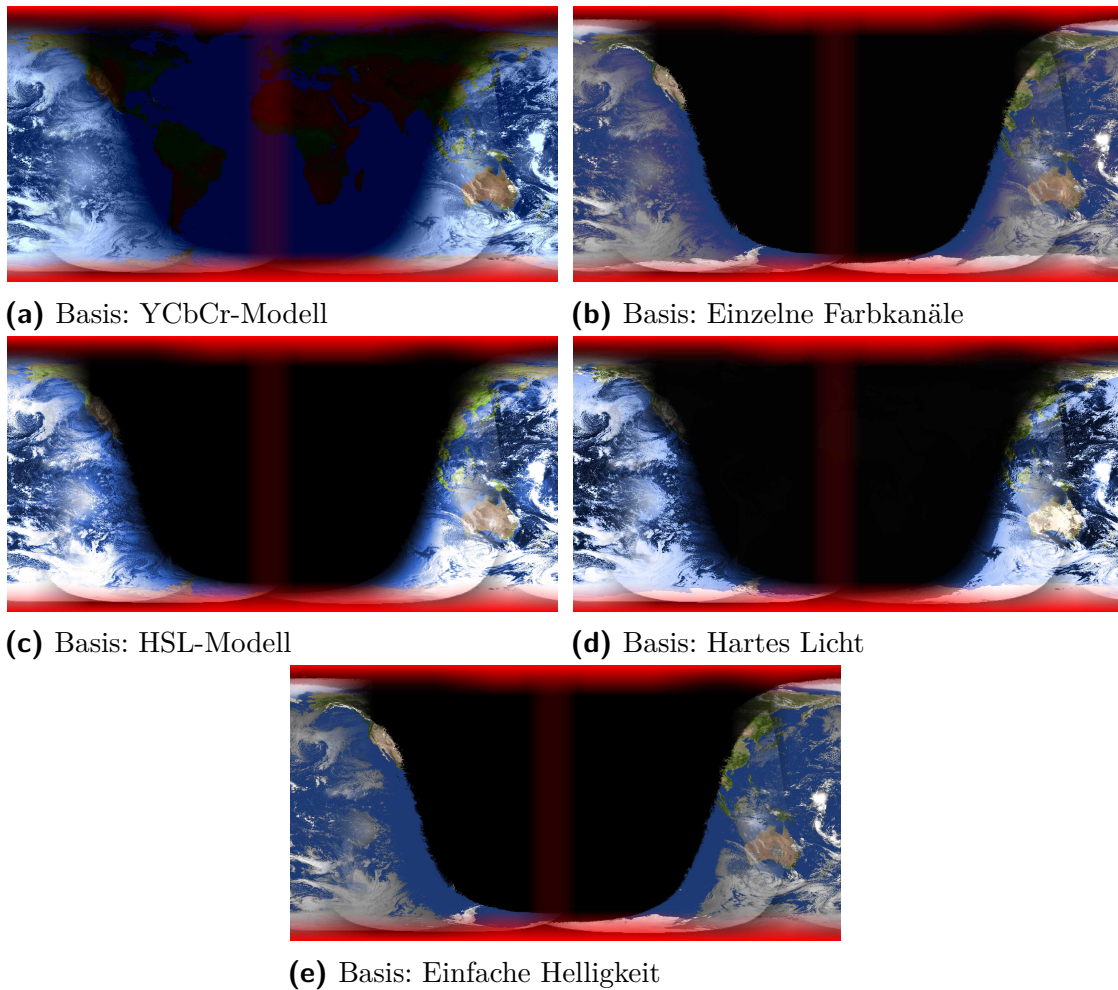
6.3.4 Kombinieren

Für das Kombinieren werden alle vorhandenen, bereits neu projizierten Bilder, übereinander gezeichnet. Da im vorherigen Schritt schon sicher gestellt ist, dass an Stellen ohne Daten die Bilder durchsichtig sind, entsteht kein Problem. Der Hintergrund wird rot und markiert damit Pixel, die von keinem Satellitenbild Daten entnehmen konnten.

6.4 Visualisierung

Hardware

Neben einem Windows PC, dessen Hardware in Tabelle 6.1 einzusehen ist, wird die Oculus Rift mit dem Development Kit 2 verwendet. Weitere Hardware ist für die Darstellung der virtuellen Erde nicht nötig.

**Abbildung 6.5.** Textur nach dem Kombinieren**Tabelle 6.1.** Eingabemöglichkeiten

Hardware	Verwendet
Betriebssystem	Windows 8.1 64-Bit
Prozessor	4x Intel(R) Core(TM) i5-4460, CPU 3.20 GHz
Arbeitsspeicher	8.00 GB
Grafikkarte	NVIDIA GeForce GTX 970

Software

Nach der Auswahl einer Zeitspanne durch den Nutzer, werden alle Texturen, die bereits zu dieser Zeitspanne vorhanden sind, in die Anwendung geladen. Weiterhin kann die Anwendung aus dem Dateipfad der Textur den Aufnahmezeitpunkt jener auslesen. Ist der Ladevorgang abgeschlossen, beginnt sofort die Animation der Erde, welche über den Timeline-Knoten der Unreal Engine realisiert ist. Das Material der

Erdkugel besteht dabei aus zwei Texturen A und B , die übereinander geblendet werden. Ein alpha-Wert α , zwischen null und eins entscheidet, wie viel Prozent der Textur A bzw. der Textur B zu sehen ist (Summe der beiden alpha-Werte immer eins). Dieser alpha-Wert wird über eine periodische Timeline (vergleiche Abbildung 6.6) verändert, sodass zwei Texturen konstant ineinander übergehen.

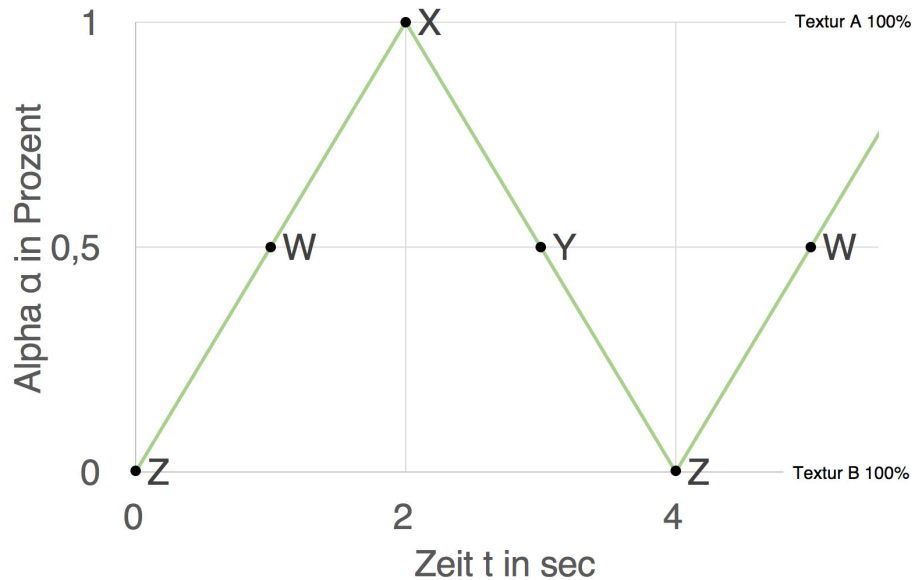


Abbildung 6.6. Timeline für den alpha-Wert im Material der Erdkugel mit den Events W, X, Y und Z

Innerhalb einer Periode feuert die Timeline außerdem die vier Events W, X, Y und Z, deren Effekt davon abhängt ob die Timeline gerade vorwärts oder rückwärts abläuft. Die Effekte sind in Tabelle 6.2 festgehalten:

Tabelle 6.2. Events der Timeline

Event	Effekt ...	... bei Vorwärts	... bei Rückwärts
W	Setze Datum auf das ...	... der Textur A	... der Textur B
X	Ersetze Textur ...	... B auf die folgende	... A auf die vorherige
Y	Setze Datum auf das ...	... der Textur B	... der Textur A
Z	Ersetze Textur ...	... A auf die folgende	... B auf die vorherige

Die Bilder 6.7 und 6.8 zeigen das finale Interface in den zwei Zuständen Menü und Animation der Erde.

Der Nutzer bekommt nichts vom Austausch der Textur mit und gewinnt den Eindruck einer konstanten Animation, da die Textur A bzw. B immer zu dem Zeitpunkt ausgetauscht wird, wenn sie nicht sichtbar ist (Event X und Z). Darüber hinaus wird



Abbildung 6.7. Stereoskopische Darstellung des Menüs

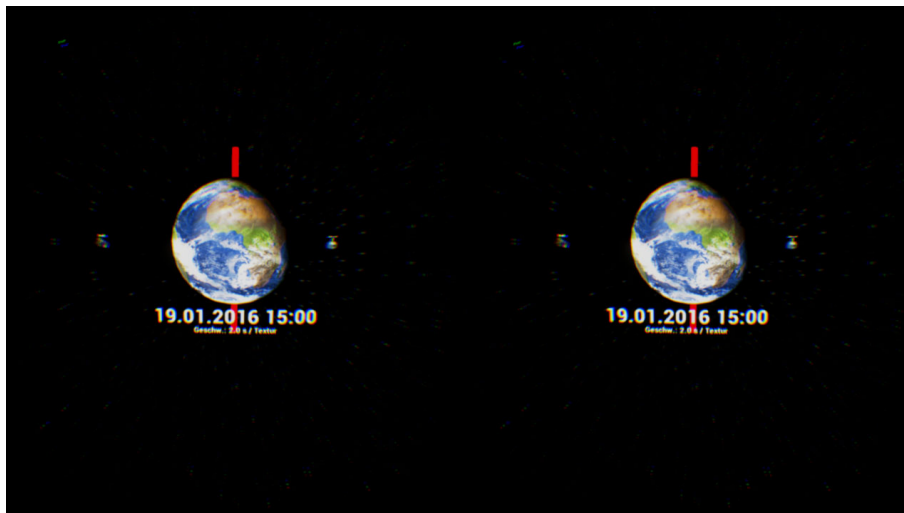


Abbildung 6.8. Stereoskopische Darstellung der Szene mit Erde und Satelliten im Hintergrund

das Datum im **Head Up Display (HUD)** durch die Events W und Y angepasst. Zeit und Datum werden nicht interpoliert.

6.5 Interaktion

6.5.1 Hardware

Als Steuerungseinheit wird eine Wii Remote Plus (RVL-CNT-01-TR) verwendet, wie sie in Bild 6.9 zu sehen ist. Die Verbindung zum PC ist über Bluetooth realisiert. Da

der benutzte PC selbst jedoch kein Bluetooth besitzt, wird ein externer Universal Serial Bus (USB) - Bluetooth Adapter verwendet.

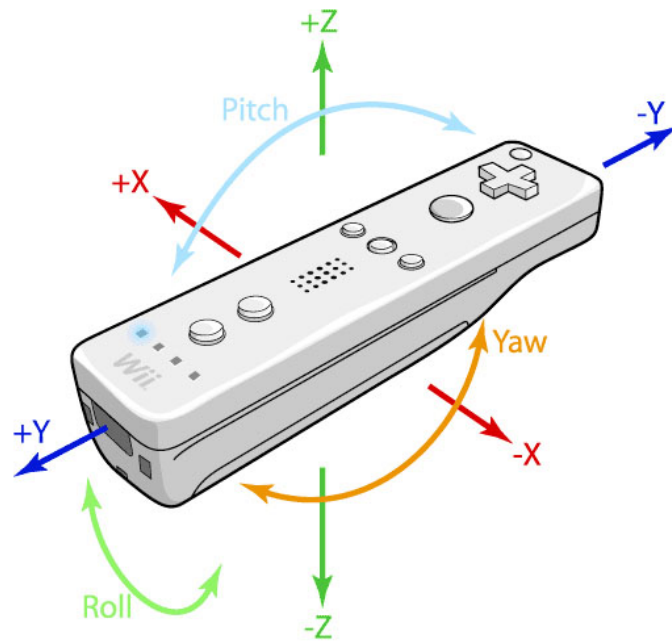


Abbildung 6.9. Wii mit sechs Freiheitsgraden und sieben Buttons zur Eingabe (B auf der Rückseite)

6.5.2 Software

Die Eingaben der Wii Mote werden durch das Programm GlovePIE (Kenner, 2009) verarbeitet. Dieses ermöglicht unter anderem die von der Wii gelieferten Werte auszuwerten und Tastendruck auf einer Tastatur zu simulieren. Diese beiden Eigenschaften werden genutzt, um die in Unreal implementierte Steuerung per Tastatur auf die Wii zu übertragen. Das Zwischenprogramm GlovePIE erwies sich als nötig, da direkte Plugins wie das UE4-Wiimote-Plugin (xoyojank, 2015) nicht stabil genug laufen oder von der Engine nicht erkannt werden. Tabelle 6.3 zeigt die finale Verbindung zwischen Wii, Tastatur und Engine. So lässt sich die Rotation Erde über Rotation der Wii Mote und der Zoom über die Knöpfe A und B kontrollieren. Für die Zeitsteuerung stehen die Knöpfe Plus und Minus zur Verfügung. Eine schrittweise Verlangsamung über den Minus Button führt dabei erst zur Pausierung und dann zur Investierung des Zeitverlaufs. Der Plus Button führt mit einer schrittweisen positiven Beschleunigung zum umgekehrten Effekt.

Diese Art der Implementierung lässt auch die schnelle Steuerung per Tastatur weiter zu, die sich vor allem beim Testen als praktisch erweist.

Tabelle 6.3. Eingabemöglichkeiten

Effekt	Tastatur	Wii
Menüsteuerung	Pfeiltasten	Steuerkreuz
Bestätigen	Enter	Home
Rotation um Z	A und D	Roll
Rotation um X	Q und E	Pitch
Zoom	W und S	A und B
Zeitsteuerung	Z und X	+ und -

7 Evaluation

Eine iterative Evaluation mit Nutzern wie sie als Kernprozess im User Center Design vorhanden ist, wird aus Zeitgründen nicht als Teil mit in diese Arbeit aufgenommen. Jedoch ist durch mehrfache Messung der Ressourcen- und Zeitverbrauch der Anwendung bestimmt. Es ergeben sich dabei die Formeln 7.1 (Ladezeit t in Sekunden in Abhängigkeit der Anzahl zu ladender Texturen n) und 7.2 (Ressourcenverbrauch r in Megabyte in Abhängigkeit der Anzahl geladener Texturen n) als lineare Regression.

$$t = 0.22 * n + 0.90 \quad (7.1)$$

$$r = 33 * n + 286 \quad (7.2)$$

Der Speicherbedarf s in Megabyte für die Aufnahmen der Satelliten plus den erstellen Texturen in Abhängigkeit von n Tagen lässt sich in Formel 7.3 ablesen, die ebenfalls durch mehrfache Messung und lineare Regression entstanden ist.

$$s = 1080 * n + 520 \quad (7.3)$$

8 Zusammenfassung und Ausblick

8.1 Zusammenfassung

Die Satellitenmeteorologie und der Wissensgewinn daraus gehören schon lange zur Meteorologie, und werden durch technische Neuerungen stetig erweitert. Im Rahmen dieser Bachelorarbeit ist daher ein Programm geschaffen, welches Wetterdaten auf einer virtuellen Erdkugel in einer dreidimensionalen Szene darstellt. Diese Szene kann von einem Nutzer über die Oculus Rift betrachtet und durch eine Wii Mote manipuliert werden. Die Daten, die bereits vom Lehrstuhl für **LuRI** verwendet werden, werden über ein weiteres Programm in Java von einer Empfangsstation per **TCP** Verbindung auf ein Rechner des Lehrstuhls für **HCI** übertragen. Dort werden sie in einer Ordnerstruktur gespeichert und zu einer Textur kombiniert. Ein großer Teil der Arbeit beschäftigte sich mit verschiedenen Verfahren, um die schwarz-weißen Satellitenaufnahmen mit einer Farbvorlage zu kombinieren. Die Szene ist mit der Unreal Engine 4 programmiert und simuliert eine Animation, indem konstant zwischen zwei ständig wechselnden Texturen interpoliert wird. Der Nutzer kann die Erde drehen und zoomen sowie den Verlauf der Zeit beschleunigen, verlangsamen, pausieren und invertieren. Die Szene enthält außerdem die verwendeten Satelliten als Objekte an der korrekten Position im All.

Zeitliche Gründe erlaubten es nicht das Programm mit Nutzern zu evaluieren, so dass das Interface, die Interaktion mit der Wii Mote und die Wahl der Kombination der Texturen die erste Iteration darstellt. Eine Messung der Ressourcen zeigt jedoch, dass die Anwendung für einen vollständigen Datensatz von drei Tagen ca. 3.8 Gigabyte Speicherplatz benötigt sowie 32.8 Sekunden Ladezeit und 5.1 Gigabyte **Random Access Memory (RAM)** für die Darstellung verbraucht.

8.2 Ausblick

Das Programm bietet viele Möglichkeit der Erweiterung an, die bereits teilweise als konzeptuelle Ideen in dieser Arbeit enthalten sind, jedoch nicht umgesetzt sind. So wäre es unter anderem möglich weitere Satelliten als Datenquellen heranzuziehen. Weiterhin könnten andere Strategien implementiert werden, nach welcher Regel die Texturen zeitlich kombiniert werden. Neben der in dieser Arbeit verwendeten Technik sicherzustellen, dass nur Daten angezeigt werden, die zum angezeigten Zeitpunkt passen, könnten auch immer die aktuellsten Daten verwendet werden. Auch wären verschiedene Optimierungsaufgaben denkbar, um Ressourcen wie Zeit- und Speicherverbrauch zu minimieren. Als einer der wichtigsten Punkte sollte die Anwendung jedoch mit Nutzern getestet werden, um dann weitere Designzyklen durchlaufen zu können.

Literatur

- Altmann, R. (o.D.). Die mischmodi von photoshop und co. URL: <http://www.simplefilter.de/grundlagen/mischmodi.html> (Zugriff am 08.12.2015).
- Bidoshi, K. (2003). *Virtual reality visualization for maps of the future* (Magisterarb., The Ohio State University).
- Camick, R. (2015 Juli). Hsl color. URL: <https://tips4java.wordpress.com/2009/07/05/hsl-color/> (Zugriff am 08.12.2015).
- Clarke, R. (2011 August). Convert rgb to ycbcr - c code. URL: <http://stackoverflow.com/questions/7086820/convert-rgb-to-ycbcr-c-code> (Zugriff am 08.12.2015).
- Desai, P. R., Desai, P. N., Ajmera, K. D. & Mehta, K. (2014 Juli). A review paper on oculus rift-a virtual reality headset. *International Journal of Engineering Trends and Technology*, 13(4).
- EUMETSAT. (2015 Juni). Meteosat. URL: <http://www.eumetsat.int/website/home/Satellites/CurrentSatellites/Meteosat/index.html> (Zugriff am 06.12.2015).
- EUMETSAT. (2016). Eumetsat is the european operational satellite agency for monitoring weather, climate and the environment. URL: <http://www.eumetsat.int/website/home/AboutUs/WhatWeDo/index.html> (Zugriff am 20.02.2016).
- Japan Meteorological Agency. (2015a). Himawari-8 spacecraft overview. URL: <http://www.jma-net.go.jp/msc/en/general/system/himawari89/index.html> (Zugriff am 20.02.2016).
- Japan Meteorological Agency. (2015b Juli). Meteorological satellites. URL: <http://www.jma.go.jp/jma/jma-eng/satellite/> (Zugriff am 06.12.2015).
- Kenner, C. (2009 Dezember). Glovepie. URL: <https://sites.google.com/site/carlkenner/glovepie> (Zugriff am 18.02.2016).
- Kettig, P. (2015). *Entwicklung, implementierung und test von algorithmen zur erkennung von unregelmäßigkeiten in den bilddaten von meteosat wettersatelliten*, Universität Würzburg. Bachelor Thesis.
- Kot, T. & Novák, P. (2014). Utilization of the oculus rift hmd in mobile robot teleoperation. In *Applied mechanics and materials* (Bd. 555, S. 199–208). Trans Tech Publ.

- NOOA. (2014 März). Noaa's geostationary and polar-orbiting weather satellites. URL: <http://noaasis.noaa.gov/NOAASIS/ml/genlsatl.html> (Zugriff am 06.12.2015).
- Stone, L. M., Erickson, T., Bederson, B. B., Rothman, P. & Muzzy, R. (1994). Visualizing data: is virtual reality the key? In *Proceedings of the conference on visualization'94* (S. 410–413). IEEE Computer Society Press.
- Taylor, D. (2015 Dezember). Msg data manager. URL: http://www.satsignal.eu/software/msg_dm.htm (Zugriff am 10.01.2016).
- Weisstein, E. W. (o.D.). Azimuthal equidistant projection. URL: <http://mathworld.wolfram.com/AzimuthalEquidistantProjection.html> (Zugriff am 08.12.2015).
- xoyojank. (2015 November). Ue4-wiimote-plugin. URL: <https://github.com/xoyojank/UE4-Wiimote-Plugin> (Zugriff am 10.02.2016).
- Zwatz-Meise, V. (1987). *Satellitenmeteorologie: satelliten beobachten das wetter*. Springer Verlag.

Anhang A. Bereitstellung der Arbeit

Diese Bachelorarbeit ist ebenfalls auf dem Git-Server des Lehrstuhls für HCI gesichert. Die schriftliche Arbeit kann unter <http://win9148.informatik.uni-wuerzburg.de/daniel.rapp/2016-ba-mcs-rapp-daniel-virtuelle-erde-thesis> eingesehen werden, während die Implementierungen unter <http://win9148.informatik.uni-wuerzburg.de/daniel.rapp/2016-ba-mcs-rapp-daniel-virtuelle-erde-code> abgerufen werden können.

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht.

Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Würzburg, 15. April 2016

Daniel Rapp