

# Herausforderungen bei der Integration von 3D: Daten, Devices und Dimensionen im AR-VR-Kontinuum

**CLAUDIA LINDNER<sup>1</sup>, JOHANNES SCHULTZ<sup>1</sup>, ANNETTE ORTWEIN<sup>1</sup>, KILIAN STAAR<sup>1</sup>,  
HENRYK HODAM<sup>1</sup>, CARSTEN JÜRGENS<sup>1</sup>, FABIAN SELG<sup>1</sup> & ANDREAS RIENOW<sup>1</sup>**

*Für den Einsatz von Geodaten im Schulbetrieb verfügen die wenigsten Schulen über ausreichende Hard- oder Software. Auch die mobilen Endgeräte der Schülerinnen und Schüler sind hierzu nur bedingt fähig, hier können jedoch zusätzliche Funktionen der Augmented und Virtual Reality (AR und VR) genutzt werden. In beiden Fällen liegt eine heterogene Ausstattung vor. Geodaten müssen daher aufwendig in GIS selektiert, vorprozessiert, exportiert, in Spieleentwicklungsumgebungen importiert und in 2D- und 3D-Umgebungen implementiert werden. Aufgrund von Datenformatinterkompatibilitäten und Dateigrößenbegrenzungen, Versionsunterschieden und variierenden Verfügbarkeiten von Daten müssen zielführende Lösungswege entwickelt werden.*

*Einige Lösungsmöglichkeiten bezüglich der genannten Daten, Devices und Dimensionen, die im Rahmen des Projektes KEPLER ISS in der Entwicklung von AR- und VR-Applikationen für den Schulunterricht gefunden und entwickelt wurden, werden in diesem Beitrag vorgestellt.*

## 1 Einleitung

Der Digitalpakt zwischen Deutschem Bund und Ländern ist nach langjährigen Verhandlungen endlich zu Stande gekommen. Ziele des Digitalpaktes sind der Aufbau schulischer Serverlösungen, W-LAN, digitale Lehr-Lern-Infrastrukturen, digitale Arbeitsgeräte und schlussendlich schulgebundene mobile Endgeräte, sofern das W-LAN bereits fertig ausgebaut ist (BMBF 2019). Bis zur endgültigen Umsetzung verbleibt die IT-Infrastruktur an vielen Schulen jedoch mangelhaft, mit einer heterogenen Ausstattung, was Hard- und Software angeht, denn an Schulen ist die hohe Diskrepanz zwischen dem aktuellen Stand der Technik und dem tatsächlichen Ausstattungsniveau besonders ausgeprägt (SÖBKE & STREICHER 2016).

Die Nutzung von Satellitendaten im Geographie-Unterricht wird von Bildungsstandards seit Jahren gefordert (DGFG 2008, DGFG 2014). Doch für die Verarbeitung oder auch nur Visualisierung von Geodaten müssen Rechner Anforderungen bezüglich Arbeitsspeicher, Prozessoren, Speicherplatz und einer Grafikkarte erfüllen, die deutlich über die Grundausstattung hinausgehen. Gleichzeitig erfordert das Lehren von Geoinformationssystemen (GIS) ein Vorwissen, das nicht in jedes Lehramtsstudium integriert ist, und das Erlernen mehr Zeit, als in regulären Stundenplänen und Schuljahren vorhanden ist. Um also Geodaten sinnvoll in den regulären Schulunterricht einzubringen, ist es von Vorteil, diese soweit zu reduzieren, dass sie auf mobilen Endgeräten funktionieren, und zwar auch auf denen der Schülerinnen und Schüler (SuS), während der notwendige Ausbau der IT-Infrastruktur der Schulen noch nicht abgeschlossen ist (LINDNER et al. 2019a).

---

<sup>1</sup> Ruhr-Universität Bochum, Geographisches Institut, Arbeitsgruppe Geomatik, Universitätsstr. 150, D-44801 Bochum, E-Mail: [claudia.lindner, johannes.schultz, annette.ortwein]@rub.de, kstaar@uni-bonn.de, [henryk.hodam, carsten.juergens, fabian.selg, andreas.rienow]@rub.de

Augmented Reality (AR) und Virtual Reality (VR) auf von SuS mitgebrachten Geräten („Bring your own device“, BYOD) gewinnen daher zunehmend an Bedeutung, auch zur Vermittlung wissenschaftlicher Inhalte. Die Voraussetzungen dieser mitgebrachten Geräte variieren jedoch stark, was bspw. das Betriebssystem oder dessen Version angeht. Apps für den Schulunterricht müssen dies mit einbeziehen, ebenso wie unterschiedliches Nutzerverhalten. Auf der anderen Seite werden Geodaten mit immer höherer raumzeitlicher Auflösung bereitgestellt, die theoretisch in den Unterricht einfließen könnten, die besagten mitgebrachten mobilen Geräte der LuL und SuS jedoch Hard- und Softwaretechnisch überfordern und auch ihre Nutzerinnen und Nutzer (NuN) vor Herausforderungen stellen. Individuelle Lösungsansätze müssen geschaffen werden, die auf so vielen Geräten und von so vielen NuN wie möglich angewendet werden können.

Im Rahmen der Projekte KEPLER ISS (ehemals Columbus Eye) und Fernerkundung in Schulen (FIS), die an den Universitäten Bochum und Bonn durchgeführt und vom Deutschen Zentrum für Luft- und Raumfahrt (Förderkennzeichen 50JR1701 & 50EE1703) gefördert werden, entstehen verschiedene AR- und VR-Lerneinheiten rund um die Themenfelder Erdbeobachtung für die Fächer Geographie und Physik. Durch das European Space Education Resource Office (ESERO) Germany werden sie in ganz Deutschland in die Schulen integriert und ebenfalls europaweit disseminiert. Anhand von zwei Beispielen für interdisziplinäre Lernmaterialien für den Schulunterricht mit Geodaten wird verdeutlicht, wie die o.g. Herausforderungen die Entwicklung von immersiven Applikationen beeinflussen und welche Lösungsansätze zielführend sind. Die hierzu verwendeten Programme sind Unity und Unreal, bei denen es sich um 3D-Spiele-Entwicklungsumgebungen handelt, Vuforia, welches die Bilderkennung für die AR bereitstellt, und AgiSoft, eine Photogrammetrie-Software.

## **2 Ausgangssituation: Raum für Verbesserung - Raum für Neues**

### **2.1 Nutzung von Synergieeffekten zwischen diversen AR-EO-Apps**

Im Projekt Columbus Eye wurden insgesamt sechs alleinstehende AR-Apps mit Erdbeobachtungsdaten entwickelt, davon vier auf Deutsch und zwei komplementäre englische Versionen (siehe Tabelle 1). Alle Apps sind jeweils einem Arbeitsblatt zugeordnet, das die Funktionen sinnvoll in den Unterricht einbringt (LINDNER et al. 2019a). Die Gesamtgröße aller sechs Apps beträgt fast 200 MB. Darin sind zahlreiche Redundanzen inkludiert: Die Inhalte und Funktionen des Nutzerinterfaces (User Interface, UI), das über die App und das Projekt informiert, Grundfunktionen für 3D-Animation, zusätzliche Funktionen, etc. werden zu jedem einzelnen App-Paket hinzugefügt. Auf den mobilen Endgeräten der Lehrerinnen und Lehrer (LuL) sowie der SuS sind auch zahlreiche andere Applikationen für verschiedenste Zwecke installiert. Manches beliebte Spiel nimmt allein 1-2 GB an Speicherplatz ein. Hinzu kommen Fotos, Videos, Musikdateien, System- und voreingestellte Apps des Herstellers, die sich zu mehreren GB aufsummieren und selbst großzügige Speicherplätze füllen. Für den Schulbetrieb ist es daher zwingen notwendig, die Apps kompakt zu implementieren. Während das W-LAN noch nicht flächendeckend ausgebaut ist, müssen sie zudem ohne Streaming auskommen: Nicht alle SuS haben unbegrenzten Internetzugang oder Netzabdeckung in ihrem Klassenzimmer. Sämtliche Teile einer modularen App müssen daher vorbereitend im heimischen W-LAN herunterladbar sein und sollten sich für den Fall, dass doch im Unterricht etwas nachgeladen werden muss, auf wenige MB beschränken.

Tabelle 1: Im Projekt Columbus Eye veröffentlichte alleinstehende Apps. Diese wurden am 27.06.2019 aus dem Store entfernt und durch die Gesamt-App ersetzt (vgl. Kap. 4.2) (Datenquelle: Google Play Console des Projekt-Accounts, Stand: 14.01.2020).

Bezeichnung	Veröffentlichung	Aquirierte neue Nutzer (kumuliert) bis 27.06.2019	Dateigröße (letzte Version)	Details
Im Auge des Sturms	19.07.2016	414	14,3 MB	ORTWEIN et al. 2017
(EN) Eye of the Cyclone	02.11.2016	97	24,9 MB	s.o.
Aralkum	22.06.2017	399	30,5 MB	SCHULTZ et al. 2017
(EN) Aralkum	15.05.2018	13	30,5 MB	s.o.
Erde bei Nacht	15.05.2018	44	61,9 MB	LINDNER et al. 2018
Erde-Mond-System	21.06.2018	152	37,4 MB	LINDNER et al. 2019b

Auf Fortbildungen wurde von teilnehmenden LuL wiederholt eine vereinigende App erbeten. Dabei erwarteten die LuL sowohl eine Reduzierung der Gesamdateigröße, als auch eine vereinfachte Nutzung. Dies ist auch im Sinne des Projektes, da diverse Lizenzmodelle sich inzwischen nach der Anzahl der veröffentlichten Apps richten. Weiterhin wünschten sich die LuL eine Version für iOS.

Die Überarbeitung und Zusammenfassung der sechs Apps soll zusammen mit einem Versionsupdate der verwendeten Software einhergehen. Es soll ein gemeinsames UI zur Übersicht entstehen, über die jeder App-Part, der eine der vorherigen individuellen Apps darstellt, einzeln heruntergeladen und gelöscht werden kann. Weiterhin soll es nicht mehr eigene Apps auf Englisch geben, sondern die gesamte App mit allen Parts lokalisiert, d.h. eine Übersetzung zur Verfügung gestellt werden.

## 2.2 Migration und Integration von 3D Modellen in AR- und VR-Applikationen

Die virtuelle Abbildung von komplexen dreidimensionalen Objekten ist heute eine gängige Methode, um Simulationen durchzuführen, wissenschaftliche Inhalte zu vermitteln, oder Daten im alltäglichen Leben zu visualisieren. Dass dreidimensionale Modelle auch in Zukunft eine immer wichtigere Rolle spielen werden, ist unbestreitbar: Umso wichtiger ist es, dass SuS bereits in der Schule mit der Thematik der 3D-Modellierung und den damit oft verbundenen Visualisierungstechniken für die Erstellung von VR- und AR-Applikationen in Kontakt kommen.

Im Zuge des KEPLER-ISS-Projektes werden zwei konkrete Maßnahmen zur Verbesserung des Verständnisses der 3D-Modellierung bei SuS verfolgt: Zum einen die 3D-Modellierung markanter Strukturen der Erdoberfläche und die anschließende Bereitstellung mittels VR-Anwendungen, zum anderen die Vermittlung von Kompetenzen zur 3D-Modellierung mittels Arbeitsblätter, die

über Smartphone-basierte AR-Apps die Inhalte anschaulich vermitteln. Dieser integrative Ansatz impliziert, dass sowohl PC-, Smartphone- als auch Tablet-basierte Anwendungen entwickelt werden müssen.

Aufgrund der Heterogenität der Endgeräte und vor allen Dingen der verwendeten Betriebssysteme ist es erstrebenswert, dass Anwendungen und die damit verbundenen Datensätze möglichst interoperabel in Bezug auf Hard- und Software entwickelt werden. Universelle Datenaustauschformate oder direkte Schnittstellen zwischen den benötigten Programmen sind jedoch nur bedingt vorhanden. Eine sehr große Anzahl von verschiedenen Standards und die fehlende Interkompatibilität erschweren insbesondere den Austausch und die Integration von 3D-Geodaten in VR-Umgebungen. Besonders die steigende raumzeitliche Auflösung von Geodaten, einhergehend mit zunehmender Datengröße, sowie unterschiedliche Datenformate und die unterschiedliche Leistungsfähigkeit der Endgeräte stellen hierbei große Herausforderungen dar. Je nach Anwendung müssen die Programme und auch Datensätze individuell auf das verwendete Endgerät angepasst werden. Basierend auf Daten des NASA-HDEV-Experiments und Astronautenbildern werden die dabei auftretenden Herausforderungen im Folgenden dargelegt.

### **3 Herausforderungen der Aufbereitung von 2D- und 3D-Daten**

Für die einzelnen Module müssen die jeweiligen Erdbeobachtungsdaten so in GIS-Programmen vorprozessiert werden, dass die benötigte Rechenkapazität für die Anzeige und Verarbeitung auf ein für Endgeräte nutzbares Maß reduziert wird. Gleichzeitig muss eine möglichst große Detailtreue in Bezug auf die räumliche und die spektrale Auflösung erhalten bleiben. Es werden die Prozessierung von hypso- und bathymetrischen Daten zur simplifizierten Simulation von Gezeiten in der Deutschen Bucht unter einer variablen Gravitationskraft und von Hyperspektraldaten des ISS-gebundenen Sensors Hyperspectral Imager for the Coastal Ocean (HICO) zur Unterscheidung potentiell giftiger und ungiftiger Algenblüten im Lake Erie vorgestellt. Beide Datensätze finden Verwendung in AR-Apps zum Einsatz von ISS- und Satellitenbildern im Schulunterricht. Die photogrammetrische Aufbereitung von Astronautenbildern in 3D-Modelle dient als Beispiel für die Vorprozessierung von Daten, die als 3D-Modelle in VR-Anwendungen Verwendung finden. Durch die Verbindung der Faszinationsfelder „Immersion“ und „Erdbeobachtung“ entstehen so Synergieeffekte, die didaktisch in Wert gesetzt werden, um die SuS für die MINT-Bildung begeistern, LuL in Bereich Virtual und Mobile Learning fortzubilden und zusätzlich Themen der nachhaltigen Entwicklung in den alltäglichen Unterricht zu integrieren.

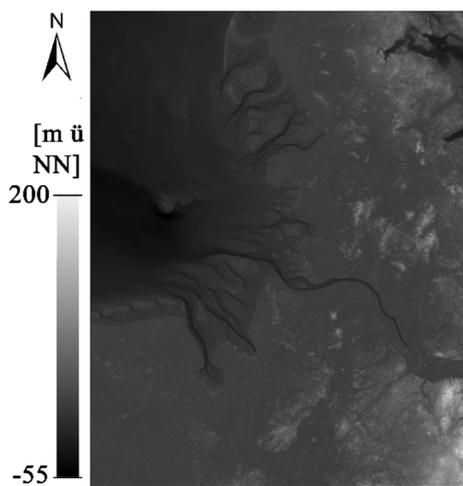
#### **3.1 Zweidimensionale Daten**

##### **3.1.1 Höhenmodell**

Für die Erde-Mond-System-App wurde ein Ausschnitt des ASTER GDEMs (METI & NASA 2011) der Deutschen Bucht als Basis genutzt. Vektordaten der Bathymetrie der Deutschen Nordsee (GEOPOTENTIAL DEUTSCHE NORDSEE 2013) wurden zu Rasterdaten mit gleichen Georeferenz-Daten umgewandelt und in das GDEM eingefügt. Rasterdaten des Baltikums waren nur mit erheblich geringerer räumlicher Auflösung zu erhalten (BALTIC HYDROGRAPHIC COMMISSION 2014), was

aufgrund der schlussendlichen Bildgröße in der Anwendung vernachlässigbar ist. Nach der Integration der drei Datensätze verblieben zahlreiche kleine Teile des Wattenmeers ohne Höhen- oder Tiefendaten. Diese wurden in originaler Auflösung der umgebenden Daten von 30m bilinear interpoliert. Aus dem Geotiff-Datensatz von 196 MB wurde zur Verarbeitung in der App ein 24-bit PNG von 61 KB generiert. Dabei entspricht jeder Graustufenwert von 0-255 einem Meter Höhen- bzw. Tiefenunterschied, beginnend beim tiefsten Punkt im Kartenausschnitt von 55 m unter NN (direkt südlich von Helgoland). Dies ist nur möglich, weil der Gradient im Bildausschnitt weniger als 256 m beträgt.

In Unity sind beim Einladen von Bilddaten Einstellungen vorzunehmen, die sich auf die Speicherung und Darstellung beziehen. Sie werden auf „Sprite (2D and UI)“ eingestellt, welches in der Entwicklungsumgebung für 3D-Welten die für 2D-Abbildungen optimierte Einstellung ist. Eine weitere mögliche Einstellung ist, die Pixeldaten der Bilder auszulesen und zu verändern („Read/Write enabled“). Dies zu aktivieren, bedeutet, dass stets eine Kopie des jeweiligen Bildes vorliegt; die vorgehaltene Datenmenge wird also verdoppelt. Angesichts der Anforderung an die App, Speicher- und Rechenleistungsverbrauch zu reduzieren, ist die Aktivierung dieser Einstellung nur bei absoluter Notwendigkeit vorzunehmen.



In der Erde-Mond-System-App ist daher nur ein Bild „Read/Write enabled“: Das finale Höhen- und Tiefenmodell, auf dem die Berechnungen für die Tiden simulation basieren (Abb. 1). Mit knapp 88.000 Pixeln und 61 KB hat es gerade so eine ausreichende Auflösung für die Betrachtung auf Tablets. Die Berechnung und Erstellung der überlagernden Wasserspiegel-Ebene bringt einige moderne Geräte an ihre Grenzen, die gleichzeitig auch noch die Bilderkennung mit der Kamera durchführen müssen. Daher wird nicht in jedem Frame (ca. 30/Sekunde), sondern nur in jedem dritten der Tidenhub neu berechnet und überlagert.

Abb. 1: Digitales Höhen- und Tiefenmodell der Deutschen Bucht (Eigene Darstellung in ArcGIS, Datenquellen: METI & NASA 2011, Geopotential Deutsche Nordsee 2013, Baltic Hydrographic Commission 2014)

### 3.1.2 Hyperspektraldaten

Die neueste App dreht sich um eine massive Algenblüte im Erie-See in Nordamerika im Jahr 2011. Aus einer HICO-Szene, die im Original über 128 Bänder von 353 bis 1080 nm mit den Ausmaßen von 512×2000 Pixeln verfügt, wurden die 87 Bänder zwischen 400 und 900 nm, welche über ein ausreichendes Signal-to-Noise-Ratio verfügen (LUCKE et al. 2011), ausgewählt und entsprechend den Anforderungen der Problemstellung zurechtgeschnitten. Die Daten enthalten eine Georeferenzierung, die sie jedoch aufgrund der Flugbahn vergleichsweise stark verzerrt und auf einer genordeten Karte sehr schief darstellt. Da den SuS die Daten so original wie möglich zur Verfügung gestellt werden sollen, aber auch aufgrund der begrenzten Größe und Auflösung von Smartphone-Bildschirmen, wurde stattdessen ein 1-dimensionales Resampling durchgeführt, das das Bild nicht

verzerrt aber die der Aufnahme inhärente Streckung der Pixel in Reihenrichtung annähernd ausgleicht.

Die resultierende Szene ist nur noch  $491 \times 725$  Pixel groß, passt in Originalgröße auf den Smartphone-Bildschirm und als 24-bit-PNG ist jedes einzelne Band nur 200-230 KB groß. Bei 87 Bändern summiert sich dies jedoch auf fast 19 MB.

Keines der HICO-Bänder ist Read/Write enabled, d.h., sämtliche Einfärbungen müssen ohne das Auslesen der Bilder funktionieren. Die Lösung hierfür sind spezielle Shader, die lediglich die Darstellung auf dem Bildschirm definieren.



Abb. 2: HICO-Szene des Westendes des Erie-Sees vom 03.09.2011 in Echtfarben, wie sie in der App eingesetzt wird: nicht genordet, in horizontaler Richtung eindimensional resampled (Eigene Darstellung, Datenquelle: NASA)

## 3.2 Dreidimensionale Modelle

### 3.2.1 3D-Modelle aus HDEV-Daten

Von April 2014 bis Oktober 2019 waren an der Außenseite des Columbus-Modul der ISS im Rahmen des NASA-HDEV-Experiments vier HD-Videokameras installiert, die die Erde aus verschiedenen Blickwinkeln durchgehend aufgenommen haben (RUNCO 2015). Bei einer durchschnittlichen Flughöhe der ISS von circa 400 km (STEFANOV et al. 2011), ergibt sich im Nadir eine Ground Sampling Distance (GSD) von 500 m bei einer Videoauflösung von  $1280 \times 720$  Pixeln.

Der ISS-Orbit ist sonnenasynchron und in Bezug auf Flughöhe und Lage hoch variabel (STEFANOV et al. 2011). Da HDEV über keinen eigenen Sternensensor verfügt und die Aufnahmezeit nicht gespeichert wird, ist eine automatische Georeferenzierung der Videos nicht möglich. Aufgrund der geringen Auflösung der Bilddaten und der fehlenden Georeferenzierung ist die Erstellung von 3D-Modellen möglich, aber aufwändig (vgl. hierzu SCHULTZ et al. 2018a; SCHULTZ et al. 2019). Die Videos werden im Super-Resolution-Verfahren verbessert, bei welchem die Pixelinformationen aus 15 Einzelbildern in einem Bild kombiniert werden, was die Auflösung der Bilder und somit die Bodenauflösung des Modells von 500 m (Nadir) auf 150 m erhöht. Es handelt sich dabei um ein Multiframe-Verfahren (IRANI 2002), welches mit mehreren niedrigauflösenden Bildern

derselben Szene arbeitet. Hierzu wurden in einem Abstand von 200 Frames 15 Frames aus dem Videomaterial (60 fps), mit MATLAB extrahiert und eine einfache Atmosphärenkorrektur (vgl. hierzu SCHULTZ et al. 2018) durchgeführt. Da sich die ISS zwischen den einzelnen Frames fortbewegt, sind die Einzelbilder um Subpixel verschoben und enthalten geringfügig veränderte Informationen zum Folgebild, die sich das Verfahren zu Nutze macht. Von jeder ausgewählten Szene wurden mittels Agisoft Photoscan Professional ein Höhenmodell, ein Orthomosaik und ein 3D-Modell erstellt und anschließend georeferenziert (AGISOFT PHOTOSCAN 2019). Dabei konnte die Ungenauigkeit der Referenzierung bei bisherigen Modellen auf etwa ein Drittel der Bodenauflösung, also circa 50 m reduziert werden. Das in Abb. 3 dargestellte Modell wurde aus HDEV-Daten eines Überfluges über Chile am 29.08.2018 erstellt hat eine Ausdehnung von 1.500 km in der Länge und einer Fläche von circa 346.500 km<sup>2</sup>.

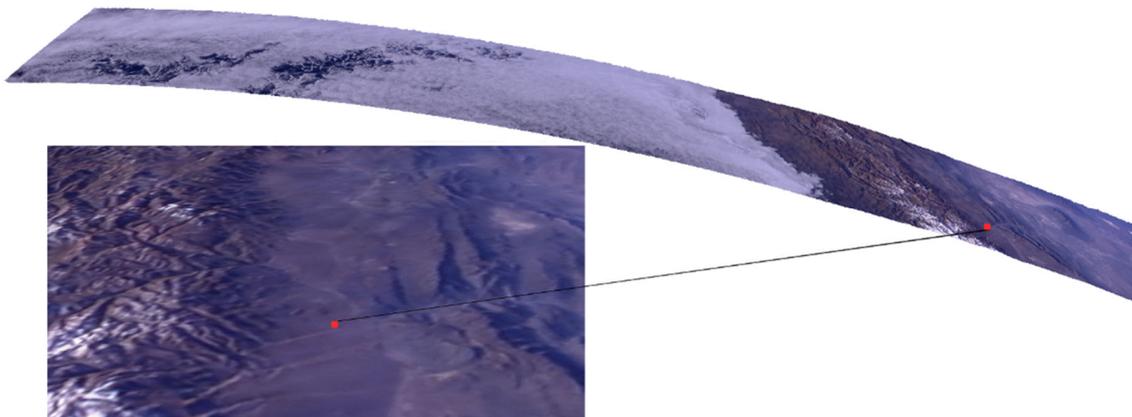


Abb 3: 3D-Modell abgeleitet aus HDEV Daten. 1500 km langer ISS Überfluges über Chile (Eigene Darstellung, Datenquelle: NASA)

Da die Höhenunterschiede im Modell schwierig zu erkennen sind, wurde ein weiteres 3D-Modell, basierend auf dem Orthomosaik und SRTM-Daten (Shuttle Radar Topography Mission, <https://www2.jpl.nasa.gov/srtm>), erstellt. Das photogrammetrische erstellte Höhenmodell und das auf SRTM-Daten basierende Höhemodell wiesen große Unterschiede auf. Diese Differenzen sind hauptsächlich auf Schatteneffekte und die geringe Auflösung der HDEV-Daten zurückzuführen. Allein der Vergleich der beiden Modelle gestaltet sich kompliziert, da sowohl im Im- als auch Export nur eine kleine und nicht notwendigerweise mit anderer verwendeter Software kompatible Auswahl von Dateiformaten zur Verfügung steht. So unterstützt ArcGIS nicht ohne Weiteres den Import von 3D-Modellen aus dem Wavefront OBJ Format, besonders Texturen und auch die Projektion führen zu Problemen. Desweiteren erschwert die ungenaue Georeferenzierung des aus HDEV-Daten abgeleiteten Modells den Vergleich mit den SRTM-Daten. Relativ einfach kann jedoch das aus HDEV-Daten mit Agisoft Photoscan Professional abgeleitete DGM mit den SRTM Daten verglichen werden.

### 3.2.2 3D-Modelle aus Astronautenbildern

Alle von Astronauten aufgenommenen Bilder von 1961 bis heute sind im „Gateway to Astronaut Photography of Earth“ der NASA frei verfügbar (<https://eol.jsc.nasa.gov>). Die hochauflösenden

Astronautenbilder der Erde eignen sich besonders gut für die Modellierung einzelner Strukturen in hoher Auflösung, wie z. B. Vulkane oder Bergzüge. Für die Entwicklung eines 3D-Modelles des 3.776 m hohen in Japan gelegenen Mt. Fuji wurden 27 Bilder (ISS046-E-35801 bis ISS046-E-35828), die am 08.02.2016 von der ISS aufgenommen wurden verwendet (SCHULTZ et al. 2019). Astronautenbilder eignen sich für die Erstellung eines 3D-Modells unter den Voraussetzungen, dass die Bilder in einem geringem zeitlichen Abstand aufgenommen wurden, sodass sie über einen großen Überlappungsbereich verfügen, einen geringfügig unterschiedlichen Aufnahmewinkel aufweisen und eine gleichbleibende Kameraeinstellung verwendet wurde (SCHULTZ et al. 2018b). Gleichzeitig haben die hier verwendeten Astronautenbilder eine hohe Auflösung von 4.928 x 3.280 Pixel. Durch die Verwendung eines 1.150 mm Objektivs liegt die GSD bei circa 4 m. Das in Abb. 4 dargestellte 3D-Modell des Mt. Fuji wurde mit Agisoft Photoscan Professional erstellt. Agisoft Photoscan errechnet basierend auf der Überlappung der Bilder die genaue Position der Kamera. Zur Verbesserung des Modells wurden in den Ausgangsbildern 10 Marker manuell gesetzt, die auch die Georeferenzierung des 3D-Modells ermöglichen. Aus diesen Daten wird eine 3D-Punktwolke und ein Mesh berechnet. Hierbei führen besonders Bildbereiche mit Wolken zu großen Fehlern, die manuell korrigiert werden müssen. Auf das Mesh werden die einzelnen Bilder projiziert und daraus die Textur erzeugt. Die Ergebnisse können als einfaches, frei drehbares 3D-Modell (Abb. 4) dargestellt werden. Ferner stehen verschiedene Austauschformate in Agisoft Photoscan zur Verfügung. Ergänzend kann auch ein animiertes PDF oder ein Anaglyphenbild (MICHEL 2013) erstellt werden, welches einen besseren dreidimensionalen Seheindruck vermittelt. Durch die Vielzahl der verwendeten Bilder konnte die GSD von 4 m auf 2.68 m verbessert werden. Das Modell hat  $6804 \times 6628$  Pixel und besteht aus 6.105.526 Punkten.



Abb. 4: 3D-Modell des Mt. Fujis, erstellt aus Aufnahmen von der ISS (Eigene Darstellung, Datenquelle: NASA)

### 3.3 Aufbereitung für VR

Für die Verarbeitung eines AgiSoft-Modells in Unreal Engine bestehen grundsätzlich zwei Möglichkeiten; das Modell kann entweder als Mesh oder als Landscape mit direkter Einbindung in übergeordnete Landschaft eingeladen werden. Die Einbettung als Landscape hat einige Vorteile, darunter eine einfachere Bearbeitung direkt im Editor, bessere Performance und standardisierte

Werte für Interaktion und Kollisionsabfragen (UNREAL 2020a). Hierbei entstehen jedoch Skalierungsprobleme beim Import in die Engine, da metrische Einheiten nicht 1: 1 übertragen werden. Um die Größenverhältnisse zwischen Hintergrund, Spielerfigur und Landscape korrekt darzustellen, werden die Elemente relativ zueinander durch einen Skalierungsfaktor festgesetzt. Dies muss bei Anwendungen (messen etc.) berücksichtigt werden, da die Einheiten nicht den realistischen Entfernungs- und Größenmaßen entsprechen und dementsprechend a priori umgerechnet werden sollten.

Eine weitere Herausforderung ist die Anpassung der Datenkomplexität des zu Grunde liegenden Modells. Ein extrem hochaufgelöstes Höhenmodell ist zwar für die NuN der VR-Anwendung realistischer, erfordert aber entsprechende Hardware (und damit Zeit) zur Berechnung; für detaillierte Anwendungen können daher verschiedene Level of Detail eingebracht werden (UNREAL 2020a). Die Vereinfachung des Modells kann z.B. indirekt durch Umwandlung in PNG (256 - 766 Höhenabstufungen) erfolgen, dies macht jedoch eine manuelle Glättung der entstandenen Kanten für realistischere Darstellung ohne Terrassen notwendig, da, anders als im GDEM in Kapitel 3.1.1 der Gradient im Bild deutlich größer ist als die Anzahl der zur Verfügung stehenden Graustufen. Bessere Auflösungen können durch r16-Rohdaten (raw-Format 16bit) erreicht werden, die jedoch nur durch wenige Bildverarbeitungsprogramme überhaupt generiert, und nur eingeschränkt in die Engine importiert werden können – sie führen häufig zu lückenhafter Darstellung, große Höhenunterschiede werden z.T. nicht korrekt übertragen.

Daher hat sich im Anwendungsfall des Mt. Fuji eine manuelle Anpassung des aus AgiSoft im PNG-Format exportierten Graustufen-Höhenmodells durch Smoothing und Sculpturing im Editor der Unreal Engine als praktikablere Alternative herausgestellt. Bei dieser Vorgehensweise werden die Konvertierungsschritte zwischen den einzelnen Dateiformaten auf ein Minimum beschränkt und somit der Workflow effizient gestaltet, ohne zu viele Details des zu Grunde liegenden Modells zu verlieren. Der Informationsverlust ist hierbei abhängig von der Spannweite der Werte des generierten Höhenmodells, daher ist dieses Vorgehen für höher aufgelöste Höhenmodelle mit großen Höhenunterschieden eher nicht zu empfehlen – hier bietet sich ein Tiling der hochaufgelösten Höhenwerte an (UNREAL 2020a).

## 4 Herausforderungen des Anwendungsdesigns

### 4.1 Entwicklungsumgebungs-Versions-Upgrade

Die AR-Apps werden in der 3D-Spieleentwicklungsumgebung Unity erstellt. Die eigentliche AR entsteht durch die Bilderkennung der Software Vuforia, welche vordefinierte Zielbilder anhand von Kantendetektion identifiziert. In Unity können beliebig Videos, 3D-Elemente usw. durch die Zielbild-Erkennung aktiviert werden. Hier werden auch zweidimensionale Inhalte im UI implementiert.

Zu Beginn des Unterfangens war Vuforia in die neueste Unity-Version, 2017, integriert und beide Programme umfassend überarbeitet worden. Dabei wurde auch eine für die Columbus-Eye-Apps wichtige Schlüsselfunktion überarbeitet: In der AR dargestellte Videos nutzten zuvor instanziierte Funktionen, i.e. mehrere Videos konnten eingeladen und individuell auf ihren zugehörigen Zielbildern abgespielt werden. Nach der Überarbeitung wurde jedoch die Funktion, die das Abspielen der Videos steuert, statisch, und so pro Szene nur noch ein Video möglich. Da jede der Apps über

mindestens zwei abzuspielende Videos verfügt, war das Upgrade daher nicht möglich und es wird die aktuelle Version von Unity 5 genutzt. Diese wird vom Hersteller aktuell noch insofern unterstützt, als dass Sicherheitslücken geschlossen werden. Die Vorgabe des Google Play Store, allen Apps eine 64-bit-Version beizufügen, kann jedoch nicht eingehalten werden. Es existiert eine Ausnahmeregelung für Apps, die mit derartigen Unity-Versionen erstellt wurden, sodass diese noch bis 01.08.2021 als reine 32-bit-Version im Store veröffentlicht werden dürfen. Ob bis dahin Unity 5 entsprechend überarbeitet wird, um auch 64-bit-App-Versionen generieren zu können, ist nicht geklärt (UNITY TECHNOLOGIES 2019).

Ähnliche Probleme ergeben sich auch bei der Nutzung anderer Entwicklungsumgebungen wie der Unreal Engine; ein Level erfordert die Darstellung eines Videos auf einem Laptopbildschirm. Hierfür wurde ein entsprechendes Mesh erstellt und der „virtuellen Laptopbildschirm“ mit einer Funktion versehen, welche eine Videodatei (.mp4) abspielt, sobald bestimmte Voraussetzungen im Spiel erfüllt wurden. Ein Problem dieser angebotenen Funktionen ist, dass diese versionsabhängig sein können. In der Folge mussten sowohl die Videoabspieelfunktion als auch die Hierarchien der Objekte mehrfach angepasst werden, um die Kompatibilität mit der neuesten Unreal-Version sicherzustellen.

## 4.2 Gemeinsames Nutzerinterface

Das neue UI der AR-App hat die Hauptaufgabe, den NuN übersichtlich Informationen bereitzustellen. Beim erstmaligen Öffnen der App müssen die NuN darüber informiert werden, dass die App nur mit den entsprechenden Arbeitsblättern funktioniert. Danach wird dieses Wissen als gegeben angesehen und soll nicht mehr bei jedem Öffnen der App erscheinen (dies war ein expliziter Wunsch von LuL auf Fortbildungen). Beim Wegklicken der Nachricht wird eine Einstellung in den Android PlayerPrefs vorgenommen. Dadurch bleibt der Wert auch beim Neustart oder der Neuinstallation der App erhalten. Die Nachricht kann jedoch über einen der obersten Buttons (Abb. 5, „Willkommen!“) wieder abgerufen werden.

Weitere Buttons enthalten die gleichen Inhalte oder Links, wie schon in den individuellen Apps, führen zu den Arbeitsblättern und den rechtlichen Hinweisen. Nahe dem unteren Bildrand findet sich auch ein Button zum Beenden der App, der äquivalent ist mit dem „Stopp erzwingen“ im Android Anwendungsmanager oder dem Wegwischen in der App-Fenster-Ansicht, die App aber auch im Hintergrund direkt wieder startet. Dies ist erforderlich, wenn die App unerwünschtes Verhalten zeigt („Bugs“), und beendet werden muss. Nach Erfahrung in LuL-Fortbildungen und SuS-App-Tests kennen manche Smartphone-NuN die Funktion des „Stopp erzwingen“ bzw. den Unterschied zwischen dem in Android üblichen Verhalten, die Apps im Hintergrund weiterlaufen zu lassen, um sie schneller wieder starten zu können, und dem tatsächlichen Beenden der Apps, nicht. Auch der „Beenden“-Button war daher ein expliziter Wunsch von LuL.

Wie auch auf zahlreichen Webseiten üblich, sind die Spracheinstellungen am rechten oberen Rand des UIs zu finden. Zwei Flaggen stehen repräsentativ für die genutzten Sprachen: Die deutsche Flagge für die deutsche Sprache und der Union Jack des Vereinigten Königreiches für die britisch-englische Sprache (Abb. 5, siehe Kap. 4.4).

Die Haupt-App „Columbus Eye“ hat eine Dateigröße von 40,9 MB. Weitere 61,6 MB können hinzugeladen werden, um sämtliche App-Teile zur Verfügung zu stellen. Davon entfallen allein 30 MB auf den Part „Algenblüte“ mit den HICO-Daten und einem Landsat-Szenenausschnitt in Originalauflösung. „Columbus Eye“ wurde seit ihrer Veröffentlichung am 27.06.2019 bisher 122-Mal heruntergeladen (Stand: 14.01.2020, vgl. Tab 1).



Abb. 5: Ausschnitte des Nutzerinterfaces des Hauptmenüs der „Columbus Eye“-App (Eigene Darstellung)

### 4.3 Dynamisches Zu- und Entladen der App-Inhalte

Es sind eine Reihe zusätzlicher Skripte notwendig, die sich mit dem Finden, Herunterladen, Speichern und Löschen der Daten befassen. Eine Liste mit den Download-Links auf einem sftp-Server und ihrer Zugehörigkeit zu den einzelnen App-Parts ist in einer separaten Klasse gespeichert. Beim Betätigen des Download-Buttons für den jeweiligen Part wird abgefragt, welche Links heruntergeladen werden müssen, und diese, sofern nicht schon vorhanden, mit den jeweiligen Unity-Funktionen heruntergeladen. Bildsammlungen wie die HICO-Bänder sind dabei in AssetBundles zusammengefasst, einem Unity-eigenen Archivdateiformat. Videos werden einzeln heruntergeladen. Alles wird zusammen in einem gesonderten Ordner auf dem Smartphone gespeichert, der in Abhängigkeit von Hersteller und Betriebssystem variiert (bspw. unter Android 9: /Interner Speicher/Android/data/com.ColumbusEye.Main/files). Bei Bedarf können NuN so auch außerhalb der App bzw. ohne die Marker auf die Videos zugreifen und lernen dabei noch, durch die Eigenen Dateien auf dem mobilen Endgerät zu navigieren. Es wurde für den Download in Einzelteilen

gegenüber dem Download ganzer App-Parts entschieden, da sich diese einzelnen Parts zahlreiche Funktions-Skripte und sonstige Dateien teilen und es erneut zu Redundanzen gekommen wäre. Es ist möglich, den Download für mehrere Parts gleichzeitig durchzuführen, da es sich um eine asynchrone Funktion handelt, jedoch kann während eines aktiven Downloads kein anderer App-Part geöffnet werden. Dies würde den Download unterbrechen. Ein Download im Hintergrund ist nicht erwünscht, da dies die Apps verlangsamen würde. Sind alle Einzelteile eines App-Parts heruntergeladen wird der Download-Button durch einen Öffnen-Button ersetzt und der nebenstehende Info-Button, über den die NuN über das dynamische Zu- und Entladen informiert werden, durch einen Löschen-Button (vgl. Abb. 5). Bei Löschen der Dateien werden diese aus dem Speicher gelöscht und die Herunterladen- und Info-Buttons wieder aktiviert.

#### **4.4 Lokalisation**

Vorgefertigte Lokalisationseinstellungen von Unity sind erst ab der 2018er-Version verfügbar und können daher in der Columbus-Eye-App nicht genutzt werden (UNITY TECHNOLOGIES 2019). Die in den individuellen Apps enthaltenen Texte und Beschriftungen waren über das Unity-Developer-UI zunächst als konstante Werte hinterlegt. Für die Lokalisation wurde jedes textenthaltende UI-Element mit einem Skript versehen, das die Beschriftung aus einer Liste ausliest. Aufgrund der Simplizität wurde für diese Liste je eine JSON-Datei pro Sprache angefertigt, in dem sämtliche Texte sowie ihre Formatierung in html-Format hinterlegt sind, sofern etwas vom Standardformat abweicht. Dies wird auch die Übersetzung in weitere Sprachen erleichtern, die so auch von Personen ohne Programmierkenntnisse durchgeführt werden kann. Die zusätzlich notwendigen Skripte sind zudem sehr kompakt und benötigen insgesamt nur 11 KB an Speicherplatz. Die Umstellung von einer Sprache zur anderen erfolgt instantan beim Drücken des entsprechenden Flaggen-Buttons im Hauptmenü.

Innerhalb der App-Parts reicht die Methode jedoch nicht immer aus. Sobald die Zielbilder oder Bildinhalte sprachlich eindeutigen Text enthalten, müssen sie separat implementiert werden. Gleiches gilt für fest in Videos eingeblendeten Text (im Gegensatz zu einem in der App übergelagerten Text). Davon sind nur zwei Videos mit einer Größe von insgesamt ca. 3 MB betroffen. Zwar enthalten noch mehr Videos Text, dieser besteht jedoch überwiegend aus Zahlen und Einheiten und enthält wenige englische Worte, die für die deutschsprachigen SuS im Zielgruppen-Alter unproblematisch sind. Bildinhalte der App werden grundsätzlich ohne Text entwickelt, welcher dann Lokalisationsabhängig eingeblendet wird. Für Zielbilder der AR ist dies jedoch nicht möglich. Fast alle App-Parts benötigen daher einen separaten „übersetzten“ Part, in dem die Bilderkennung nach den lokalisierten Zielbildern sucht. In Abhängigkeit der gewählten Lokalisation wird beim Drücken des „Öffnen“-Buttons der deutsche oder englische Part geladen und ein Ladebildschirm mit ebenfalls lokalisiertem Text angezeigt. Um die Sprache zu wechseln, müssen NuN den Part wieder verlassen. Die meisten Zielbilder sind sich zwischen ihren jeweiligen Lokalisationen jedoch so ähnlich, dass sie auch in der anderssprachigen Szene funktionieren.

#### **4.5 Texturierung, Materialien und Interaktion in der VR**

Eine zusammengesetzte Luftbildaufnahme dient als Grundlage für die Textur des virtuellen Mount Fuji. Damit der visuelle Eindruck des Mount Fuji aus den Astronautenbildern erhalten bleibt, wurde eine große Gesamttextur erstellt, welche das Luftbild 1:1 ohne Kachelungseffekt auf die

erstellte Landschaft projiziert. Im Nahbereich führt diese Herangehensweise jedoch zu verwaschenen Texturen, da die Auflösung des Luftbildes zu gering ist. Durch Überblenden der Gesamttextur mit künstlichen, hochauflösenden Einzeltexturen kann ein realistischerer Eindruck erzeugt werden. Hierzu wurde ein Material erstellt, welches das Verhalten der einzelnen Texturen („texture blending“) in Bezug auf Entfernung zum Betrachter, Beleuchtungseffekte und die Lokation der Einzelkomponenten auf dem Modell steuert (UNREAL 2020b). Grundlage des letzten Punktes ist eine Minimum-Distance-Klassifikation der Oberflächen in die Klassen Schnee, Stein, und Vegetation, aus der eine RGB-Karte erstellt wurde. Für einen dreidimensionalen Eindruck der kleinteiligen Texturen wurde zudem eine Normal Map mit der Software Crazybump erstellt (<http://www.crazybump.com/>), um das Gelände auch im großen Maßstab realistisch wirken zu lassen. Sowohl für Materialien als auch Texturen sind frei verfügbare Bibliotheken erhältlich, die es ermöglichen, in kurzer Zeit auf das eigene Vorhaben angepasste eigene Materialzusammenstellungen zu entwickeln (UNREAL 2020c). Für eine stimmige Einbettung des Fuji-Modells in eine Szenerie wurden eine Umgebungslandschaft in geringerer Auflösung sowie eine Skybox erstellt. Hierbei sind Tag-Nacht-Wechsel ebenso möglich wie Wettereffekte.

Ein Vorteil der Verwendung von Spieleengines ist, dass sich vielfältige Eingabemethoden einrichten und gleichzeitig verwenden lassen. Somit lassen sich bereits vorhandene (und vertraute) Eingabemethoden wie Tastatur und Maus, Gamepad, etc. genauso verwenden wie VR-Motion Controller, was das Zurechtfinden innerhalb der Anwendung erleichtert. Die Bewegung innerhalb der VR-Anwendung erfolgt als Freiflug in Ich-Perspektive. Für den Actor werden verschiedene Funktionen definiert, unter anderem das Verhalten bei Kollision mit Landschaftselementen, und Steuerungsparameter wie Geschwindigkeit, Beschleunigung und Lage. Die Ausdehnung der Spielszene wird durch eine Blocking Volume begrenzt, welche den Spieler in seinem Bewegungsradius einschränkt und somit die Erstellung einer unendlichen Welt obsolet macht.

Die interaktiven Elemente der Anwendungen wurden in ein separates Level eingebunden, welches durch eine Teleportfunktion angesprochen wird. Fliegt der Spieler auf den Gipfel des Mt. Fuji, löst er eine Triggerfunktion aus, wird in das o.g. Level transportiert und wechselt die Perspektive von „fliegend“ auf „laufend“, was die Interaktion (in diesem Fall mit einem Laptop und einem Tastenfeld) intuitiver gestaltet. Diese Lösung wurde gewählt, um die Flugsimulation so authentisch wie möglich zu gestalten und die Aufmerksamkeit der SuS in dieser Anwendungsphase nicht durch interaktive Elemente abzulenken.

#### **4.6 Plattformübergreifendes Design**

Das verwendete Fuji-Modell der Desktopversion stellt, aufgrund der hohen Auflösung, zu hohe Ansprüche an die Leistung von aktuellen Smartphones und Tablets. Um das Problem zu lösen, wurde das Modell vereinfacht, indem die Anzahl an Vertices stark verringert und das Material auf die Verwendung von nur einer Textur (die des zusammengesetzten Astronautenbilds) begrenzt wurde. So konnte ein sinnvolles Gleichgewicht zwischen Informationsgehalt und Performance gewahrt werden (GOOGLE DEVELOPER 2020; UNREAL 2020e). Da es nicht ohne externe Eingabe möglich ist, mit der Anwendung zu interagieren, wurde auch die Interaktionskomponente eingeschränkt. Die Flugsteuerung wurde auf automatische Fortbewegung umgestellt, bei der Richtungsänderungen durch Kopfbewegung eingeleitet werden. Ein Stoppen ist somit nicht möglich. Auch die Interaktion mit Spielelementen erfolgt nun ausschließlich durch Fokussierung eines Objekts

für mehrere Sekunden. Dies schränkt bislang die Möglichkeiten ein, immersive Anwendungen mit interaktiven Elementen auszustatten. Bluetooth-Controller können diese Limitation beheben, sind dann jedoch unabdingbar und müssen von den Anwendern bereitgestellt werden. Dies widerspricht der Idee der ubiquitären Verfügbarkeit von Smartphones, die der Grund für die Portierung auf mobile Endgeräte ist.

## 5 Integration ins Klassenzimmer

SuS der Klassen 10-11 testeten die Erde-Mond-System-App auf ihre Funktionalitäten. Reaktionen auf die App fielen positiv aus und brachten selbst Physik-desinteressierte SuS dazu, sich für Themen wie Gravitation und Astronomie zu interessieren. Im Nachgang wurden zahlreiche Verbesserungsvorschläge implementiert, die sich auf Versionsunterschiede, Datengröße, Zielbilder und Übersichtlichkeit des UIs beziehen (LINDNER et al. 2019b). Dabei wurde auch klar, dass SuS sehr unterschiedliche Erwartungen an Lern-Apps haben. In der Diskussion mit der Klasse wurde bspw. der Wunsch nach einer Vertonung einer Funktion laut, bei der der Mond in die Erde stürzt: Während manche SuS sich eine comic-artige Explosion wünschten, diskutierten andere realistische Alternativen und kamen zu dem Schluss, dass sich im annähernden Vakuum des Weltraums kein Schall ausbreitet und in der Perspektive der NuN von hinter dem Mond daher kein Geräusch zu hören wäre.

Die erste lauffähige Version der VR-Anwendung wurde auf Publikumstagen ebenfalls von SuS getestet. Ihr mündliches Feedback fiel überwiegend positiv aus. Gelobt wurde die Verwendung von VR-Elementen im Bildungsbereich und den Vorstoß neuer Technologien in das Klassenzimmer. Das Thema, das als eher trocken empfunden wurde, konnte durch die Anwendung anschaulicher dargestellt werden und wurde mit größerem Enthusiasmus aufgenommen. Bemängelt wurde die Verfügbarkeit der benötigten Hardware in der Schule sowie die limitierten Interaktionsmöglichkeiten innerhalb der Anwendung. Um diesen Anregungen Rechnung zu tragen, wird die Anwendung aktuell auf mobile Endgeräte portiert. Um den immersiven VR-Charakter zu erhalten, können VR-Halterungen für Smartphones (z.B. Google Cardboard) eingesetzt werden. Hierbei treten einige Hürden auf, insbesondere hinsichtlich der Komplexität des Modells und der Heterogenität der Smartphone-Hard- und -Software, u.a. unterschiedliche Programmierschnittstellen wie OpenGL ES2 vs. 3.1, sowie unterstützte Multitexturformate (ETC1/2, DXT, ASTC, ...) (GOOGLE DEVELOPER 2020, UNREAL 2020d). Die Lösung liegt hierbei (noch) in der Bereitstellung multipler Formate für die unterschiedlichen Endgerätagenerationen.

## 6 Fazit und Ausblick

Die Entwicklung von AR- und VR- Anwendungen für den Schulunterricht steht in der Praxis besonders an den Schnittstellen von Daten, Devices und Dimensionen vor großen Herausforderungen. Für den sinnvollen Einsatz von Erdbeobachtungsdaten im Schulunterricht reicht die aktuelle Schul-IT-Infrastruktur nur in Ausnahmefällen aus und wird es auch in Zukunft aufgrund der erhöhten Anforderungen an Hard- und Software nicht flächendeckend. Eine Möglichkeit, derartige Daten trotzdem einzubringen, sind die mobilen Endgeräte von SuS im BYOD-Verfahren. Auch diese sind für herkömmliche Geodaten aufgrund der immer höheren raumzeitlichen Auflösung nur

bedingt geeignet, verfügen jedoch über zusätzliche Möglichkeiten, die Daten SuS-gerecht aufzuarbeiten und darzustellen.

Hierzu müssen Geodaten sorgfältig individuell selektiert und prozessiert werden, um sie in ihrer Datengröße zu reduzieren und in Dateiformate zu überführen, die von den weiterprozessierenden Programmen gelesen werden können. Oftmals geschieht dies über Umwege, da GIS- und Photogrammetrieanwendungen für die Visualisierung und Auswertung der Daten nur über wenige Schnittstellen-Formate mit Spielentwicklungsumgebungen verfügen, die für die Einbindung in Unterrichtsaufgaben genutzt werden. Eines der wenigen, nahezu universell verfügbaren Formate ist das PNG. So können Farbabstufungen unter bestimmten Voraussetzungen zur Übertragung von Höhendaten, auch im 3D-Format, genutzt werden.

Im Projekt KEPLER ISS wurden eine Reihe existenter AR-Applikationen für Android-Smartphones zusammengefügt, um Redundanzen zu reduzieren und die Bedienbarkeit zu verbessern. Dabei wird den NuN die Möglichkeit gegeben, die Anzeigesprache instantan zu wechseln, einzelne Inhalte dynamisch hinzuzuladen oder zu entfernen sowie auf einzelne Inhalte auch außerhalb der App zuzugreifen. Weiterhin wurde eine VR-Anwendung basierend auf Bildaufnahmen von Astronauten an Bord der ISS entwickelt, für die die besagten Aufnahmen zu einem hochauflösenden 3D-Modell prozessiert und für eine 3D-Umgebung exportiert wurden. Auch dieses soll auf den mobilen Endgeräten der SuS anwendbar sein, was jedoch zu Einschränkungen in den Bedienmöglichkeiten führt.

Im AR-Bereich ist in naher Zukunft die Migration der App in iOS vorgesehen, damit sie einer breiteren Masse an LuL und SuS zur Verfügung stehen. Als nächster Inhalt sind Interferometrie-Daten in einer Lektion über Vulkane vorgesehen, bei dem auch das 3D-Fuji-Modell zum Einsatz kommen wird. Bis dahin werden weiter die Möglichkeiten der Übertragung des VR-Erlebnisses auf Smartphones in Cardboards exploriert. Weitere 3D-Modelle für die Anwendung im Schulunterricht werden kontinuierlich erstellt. Eine formative Evaluation dieser Lernmaterialien in AR und VR findet kontinuierlich statt.

## 7 Danksagungen

Die Projekte Kepler ISS und FIS werden vom Deutschen Zentrum für Luft- und Raumfahrt e.V. mit Mitteln des Bundesministeriums für Wirtschaft und Energie aufgrund eines Beschlusses des Deutschen Bundestages gefördert (Förderkennzeichen 50JR1701 & 50EE1703).

Weiterhin danken wir der Remote Sensing Research Group der Universität Bonn, in der die Projekte für viele Jahre ausgeführt wurden, für den andauernden hervorragenden fachlichen Austausch.

## 8 Literaturverzeichnis:

AGISOFT PHOTOSCAN, 2019: Agisoft Metashape, Professional Edition, Version 1.5  
[https://www.agisoft.com/pdf/metashape-pro\\_1\\_5\\_en.pdf](https://www.agisoft.com/pdf/metashape-pro_1_5_en.pdf)

BALTIC HYDROGRAPHIC COMMISSION, 2014: Baltic Sea Bathymetry Database. <http://data.bshc.pro>

- BUNDESMINISTERIUM FÜR BILDUNG UND FORSCHUNG (BMBF), 2019: Verwaltungsvereinbarung: DigitalPaktSchule 2019 bis 2024. [https://www.bmbf.de/files/VV\\_DigitalPaktSchule\\_Web.pdf](https://www.bmbf.de/files/VV_DigitalPaktSchule_Web.pdf)
- DEUTSCHE GESELLSCHAFT FÜR GEOGRAPHIE (DGFG), 2008: Bildungsstandards im Fach Geographie für den Mittleren Schulabschluss, Bonn.
- DEUTSCHE GESELLSCHAFT FÜR GEOGRAPHIE (DGFG), 2014: Bildungsstandards im Fach Geographie für den Mittleren Schulabschluss, Bonn.
- GEOPOTENTIAL DEUTSCHE NORDSEE, 2013: Bathymetrie. [www.gpdn.de/](http://www.gpdn.de/)
- GOOGLE DEVELOPER, 2020: Unreal: Development for Android. <https://developers.google.com/ar/develop/unreal>
- IRANI, M., 2002: Multi-Frame Correspondence Estimation Using Subspace Constraints. International Journal of Computer Vision **48**(3), 173-194, doi: 10.1023/A:1016372015744.
- LINDNER, C., JÜRGENS C. & RIENOW, A., 2018: Fernerkundung und Raumfahrt in Schulen unter besonderer Berücksichtigung des Ruhrgebiets. In: Schulbuchinformationsdienst Ruhrgebiet **91**, 2-3.
- LINDNER, C., MÜLLER, C., ORTWEIN, A., HODAM, H., JÜRGENS, C., SCHULTZ, J., SELG, F. & RIENOW, A., 2019a: Das Klassenzimmer im Weltraum – Anwendungen zur Erdbeobachtung zwischen Realität und Virtualität. In: T. Kersten (Hrsg.): Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., Band **28**, 241-255.
- LINDNER, C., RIENOW, A. & JÜRGENS, C., 2019b: Augmented Reality applications as digital experiments for education – An example in the Earth-Moon System. In: Acta Astronautica **161**, 66-74, doi: 10.1016/j.actaastro.2019.05.025 .
- LUCKE, R.L., CORSON, M., MCGLOTHLIN, N.R., BUTCHER, S.D., WOOD, D.L., KORWAN, D.R., LI, R.R., SNYDER, W.A., DAVIS, C.O. & CHEN, D.T., 2011: Hyperspectral Imager for the Coastal Ocean: instrument description and first images. Applied optics **50**(11), 1501-1516, doi: 10.1364/AO.50.001501.
- MICHEL, B., 2013: Digital Stereoscapy: Scene to Screen 3D Production Workflows. Stereoscapy News, Sprimont-Belgium.
- MINISTRY OF ECONOMY, TRADE, AND INDUSTRY & UNITED STATES NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (METI & NASA), 2011: ASTER Global Digital Elevation Model V002. doi: 10.5067/ASTER/ASTGTM.002
- ORTWEIN, A., KREBS, B., MEHRING, J., SCHULTZ, J. & Rienow, A., 2017: Neue Dimensionen der Erdbeobachtung im Schulunterricht: Raumfahrt, 3-D und Augmented Reality. In: T. Kersten (Hrsg.): Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., Band **26**, 332-341.
- RUNCO, S., 2015: International Space Station – High Definition Earth Viewing (HDEV), [http://www.nasa.gov/mission\\_pages/station/research/experiments/917.html](http://www.nasa.gov/mission_pages/station/research/experiments/917.html)
- SCHULTZ, J., ORTWEIN, A., LINDNER, C., HODAM, H., WEPPLER, J. & RIENOW, A., 2017: Augmenting Pupil's Reality from Space – Learning with Digital Media based on Earth Observation Data from the ISS. Proceedings of the 68 the International Astronautical Congress 2017 25-29 September in Adelaide, Australia, 1-6.

- SCHULTZ, J.A., ORTWEIN, A. & RIENOW, A. 2018a: Technical Note: Using ISS Videos in Earth Observation - Implementations for Science and Education. *European Journal of Remote Sensing* **51**(1), 28–32. doi:10.1080/22797254.2017.1396880
- SCHULTZ, J., HODAM, H., LINDNER, C., ORTWEIN, A., SELG, F. & RIENOW, A., 2018b: Ableitung von 3D-Modellen aus Daten des High Definition Earth Viewing-Experiments (ISS) –Anwendungen für den Schulunterricht. In: T. Kersten, Gülch, E., Schiewe, J., Kolbe, T., Stilla, U. (Hrsg.): *Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V.*, Band **27**, 129-140.
- SCHULTZ, J.A., HARTMANN, M., HEINEMANN, S., JANKE, J., JÜRGENS, C., OERTEL, D., RÜCKER, G., THONFELD, F., & RIENOW, A. 2019: DIEGO: A Multispectral Thermal Mission for Earth Observation on the International Space Station. *European Journal of Remote Sensing*, Online-Ausgabe, 1-11, doi: 10.1080/22797254.2019.1698318
- SÖBKE H. & STREICHER A., 2016: *Serious Games Architectures and Engines*. Dörner R., Göbel S., Kickmeier-Rust M., Masuch M., Zweig K. (Hrsg.): *Entertainment Computing and Serious Games*. *Lecture Notes in Computer Science* **9970**, Springer.
- UNITY TECHNOLOGIES, 2019: Unity User Manual. <https://docs.unity3d.com/Manual/index.html>
- UNREAL, 2020a: Unreal Engine 4 Documentation: Landscape Outdoor Terrain. <https://docs.unrealengine.com/en-US/Engine/Landscape/index.html#landscapefeatures>
- UNREAL, 2020b: Unreal Engine 4 Documentation: Landscape Materials. <https://docs.unrealengine.com/en-US/Engine/Landscape/Materials/index.html>
- UNREAL, 2020c: Unreal Engine 4 Documentation: Materials. <https://docs.unrealengine.com/en-US/Engine/Rendering/Materials/index.html>
- UNREAL, 2020d: Unreal Engine 4 Documentation: Android Game Development. <https://docs.unrealengine.com/en-US/Platforms/Mobile/Android/index.html>
- UNREAL, 2020e: Mobile Performance Tips and Tricks. <https://docs.unrealengine.com/en-US/Platforms/Mobile/Performance/TipsAndTricks/index.html>