Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., Band 23, Potsdam, Hrsg. E. Seyfert,E. Gülch, C. Heipke, J. Schiewe, M. Sester, 34. Wissenschaftlich-Technische Jahrestagung der DGPF, 26.-28. März 2014 an der HafenCity Universität Hamburg, Tagungsband auf CD-ROM, Beitrag 173.

Interaktive Visualisierung und Steuerung von 3D-Modellen und Punktwolken mit Low-Cost-Systemen

FELIX TSCHIRSCHWITZ[1], THOMAS P. KERSTEN[1] & KAY ZOBEL[1]

Zusammenfassung: Für die interaktive Visualisierung von texturierten 3D-Landschafts- und Architekturmodellen, die an der HCU Hamburg als CAD-Modell oder Dreiecksvermaschung erstellt wurden, wird die moderne Spiel-Engine Unity eingesetzt. Hierbei werden texturierte Modelle oder auch Punktwolken in einem Viewer so dargestellt, dass eine Spielfigur wie bei einem Computerspiel durch die virtuelle Welt bewegt werden kann. Der Blickwinkel kann vom Anwender selbst gewählt werden, um so einen virtuellen Rundgang im entsprechenden Modell zu ermöglichen. Für die realistischen 3D-Darstellungen werden Low-Cost-Hardware-Komponenten genutzt (z.B. 3D-Monitore). Die Daten werden mit einer entsprechenden App auf einem Android-Smartphone dargestellt, das als Head-Mounted-Display (HMD) genutzt wird. Die Inertialsensoren des Telefons werden für das Headtracking genutzt. Um Bewegungen der Spielfigur in den visualisierten 3D-Daten auszulösen bzw. zu steuern, kommen unterschiedliche Controller zum Einsatz, die bei klassischen Spielkonsolen oder als PC Zubehör "off-the-shelf" Verwendung finden (Sony Dualshock3, Nintendo WiiMote & Nunchuk, Microsoft Kinect, Leap Motion, XSens MTi). Aktuell werden auf unterschiedlicher Hardware vermaschte Modelle im FBX Format verarbeitet und visualisiert, während zur Darstellung von farbigen Punktwolken die Daten im ASCII-Format PLY (mit XYZRGB) auf Basis von DirectX 11 verwendet werden.

1 Einleitung

Durch Visualisierung können im Allgemeinen abstrakte Daten und Zusammenhänge in eine graphische bzw. visuell erfassbare Form gebracht werden, in der sich mithilfe einer speziellen Prozesskette die Daten in Bilder überführen lassen. Seit Jahrhunderten ist die dreidimensionale Darstellung der Landschaft und Architektur in Form von perspektivischen Abbildungen (Zeichnungen) eine verbreitete Technik zur Visualisierung des geographischen Raumes (ZANINI 1998). Durch die rasante Entwicklung der Computertechnologie erlebte die digitale Modellierung und Visualisierung von Architekturobjekten und Landschaften jedoch erst vor einigen Jahren ihren Durchbruch. Denn beim "Rendern" solcher perspektivischen Ansichten spielen neben der Computerleistungsfähigkeit vor allem auch die Qualität der verwendeten Materialien für die Texturen der Objekte und die Beleuchtung eine entscheidende Rolle, um das Endergebnis realistisch wirken zu lassen. Entsprechende 3D-Visualisierungsprogramme wie Autodesk 3ds Max, Autodesk Maya oder Cinema 4D haben sich nicht nur seit mehr als 10 Jahren in der Filmindustrie etabliert, sondern sind mittlerweile auch fester Bestandteil in Architektur- oder Visualisierungsbüros geworden. War vor vielen Jahren noch die gerenderte perspektivische Ansicht (Standbild) das vorzeigbare Ergebnis, so stehen heute Animationen (fly-through, walk-through) und interaktive Visualisierungen im Vordergrund, denn Videospiele ermöglichen schon seit langem das Eingreifen des Betrachters in die virtuelle Welt. Während bei

[1] HafenCity Universität Hamburg, Hebebrandstraße 1, 22297 Hamburg,
E-Mail: [felix.tschirschwitz,thomas.kersten,kay.zobel]@hcu-hamburg.de

den Animationen der Anwender die Visualisierung von Objekten und Landschaften als Video in einem vordefinierten virtuellen Kamerapfad mit einem zeitaufwendigen Renderprozess umsetzt, wird bei der interaktiven Visualisierung die Zeit zum Rendern durch den Einsatz von entsprechender Computergrafik-Hardware in Echtzeit gespart. Dadurch hat der Betrachter die Freiheit, sich selbständig und frei durch das Modell zu bewegen, um den Raum zu erkunden und zu erleben.
In diesem Beitrag werden verschiedene Low-Cost-Systeme zur 3D-Visualisierung von texturierten CAD- und vermaschten Modellen vorgestellt, die eine sehr kostengünstige Alternative zu komplexen Virtual-Reality-Systemen wie einem CAVE Automatic Virtual Environment oder einer 3D-Powerwall darstellen. Als Low-Cost-Systeme für die Visualisierung (Betrachtung und Steuerung) sind Hardwarekomponenten definiert, die weniger als 2000 Euro kosten, wie z.B. ein 3D-Monitor oder ein Smartphone für weniger als 500 Euro. Es werden die Aufbereitung der zu visualisierenden 3D-Daten, die Spiel-Engine Unity zur Generierung der 3D-Szenen und die Implementation der Software zur Steuerung der interaktiven Visualisierung auf Low-Cost-Ausgabegeräten vorgestellt.

2 Datengrundlagen

An der HafenCity Universität Hamburg werden im Labor für Photogrammetrie und Laserscanning seit 13 Jahren im Rahmen von Lehrveranstaltungen und diversen Projekten 3D-Modelle von Objekten aus Architektur und Archäologie generiert. Diese Objekte werden durch Photogrammetrie (KERSTEN et al. 2004, KERSTEN 2006), durch terrestrisches Laserscanning (KERSTEN et al. 2009, KERSTEN et al. 2010), durch beide Verfahren in Kombination (KERSTEN et al. 2006, LINDSTAEDT et al. 2008) oder durch Computer Vision Methoden (KERSTEN & LINDSTAEDT 2012a) erfasst bzw. erstellt. Die 3D-Modellierung der jeweiligen Objekte zu einem CAD-Volumenmodell erfolgte mit den photogrammetrisch gemessenen Punkten in AutoCAD, während aus Punktwolken 3D-Modelle sowohl durch Dreiecksvermaschungen in Geomagic als auch im CAD erstellt wurden. Beispiele für eine CAD-Modellierung nach photogrammetrischer Aufnahme sind das Celler Schloss (KERSTEN et al. 2003) und das Ahrensburger Schloss (KERSTEN & ACEVEDO PARDO 2003), während eine CAD-Modellierung aus Punktwolken für den Kaiserdom in Königslutter durchgeführt und dokumentiert wurde (KERSTEN & LINDSTAEDT 2012b).
Eine 3D-Modellierung durch Dreiecksvermaschung aus Punktwolken ist für die Projekte Statuen der Osterinsel (KERSTEN et al. 2009), Bismarck-Denkmal in Hamburg (KERSTEN et. al. 2010) und Fort Al Zubarah in Katar (KERSTEN & MECHELKE 2013) publiziert worden. Nach der Generierung der 3D-Modelle erfolgte die manuelle Texturierung der Oberflächen in verschiedenen Visualisierungssoftwarepaketen wie Cinema 4D, 3ds Max, Maya und anderen. Für die Implementierung der interaktiven Visualisierung auf den verschiedenen (Low-Cost-) Systemen kam in erster Linie das in Cinema 4D texturierte 3D-Modell des Alt-Segeberger Bürgerhauses zum Einsatz (siehe Abb. 1 oben). Das 3D-Modell wurde aus photogrammetrisch mit einer Nikon D90 erfassten Bilddaten in AutoCAD modelliert. Die Generierung der RGB-Punktwolke (Abb. 1 unten) für das Bürgerhaus erfolgte aus denselben Bilddaten durch Computer Vision Algorithmen (z.B. Bundler/PMVS2) (KERSTEN & LINDSTAEDT 2012a).

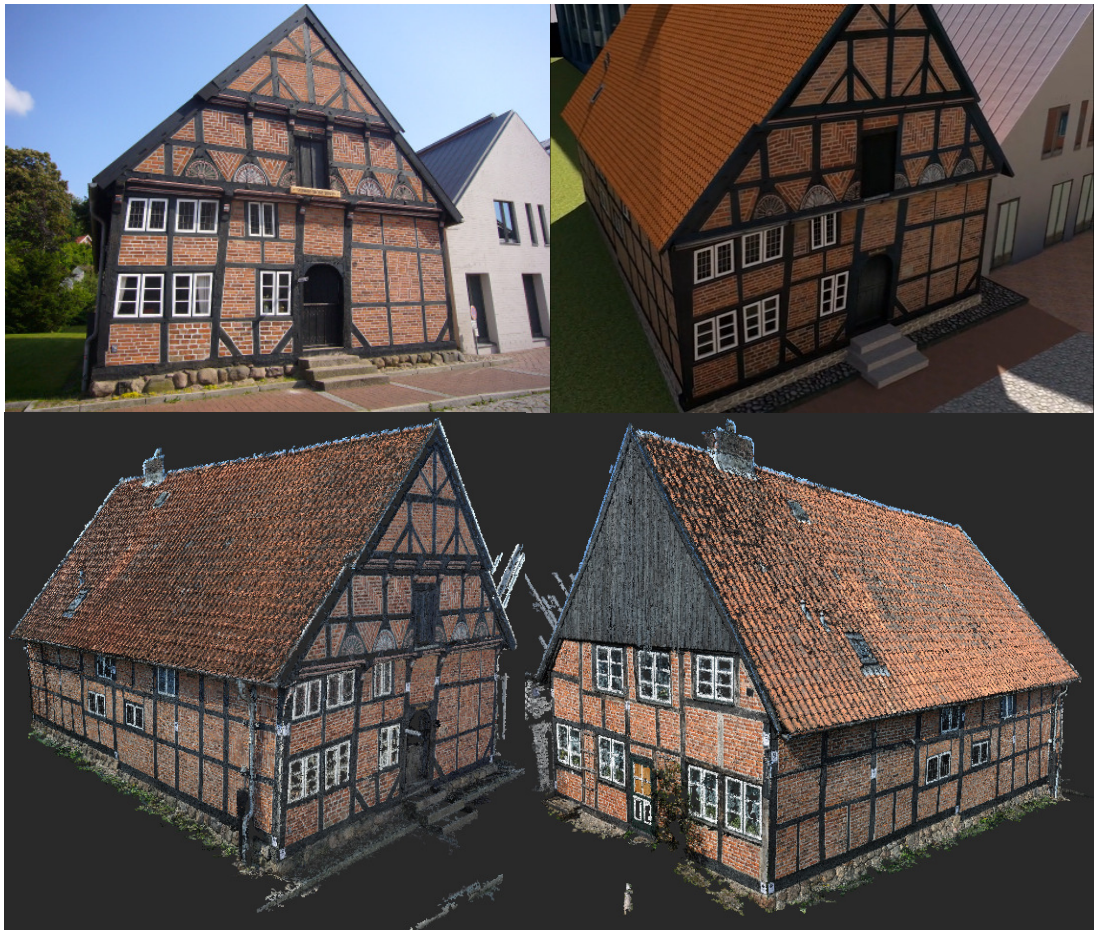

Abb. 1: Alt-Segeberger Bürgerhaus – Foto der Frontfassade (links oben), texturiertes 3D-Modell (rechts oben) und RGB-Punktwolke (unten)

3 Visualisierung mit der Spiel-Engine Unity

Durch die Visualisierung können erfasste 3D-Daten dem Nutzer angezeigt werden. Im Rahmen dieser Implementierungen werden als Daten texturierte CAD-Modelle, Modelle aus einer Dreiecksvermaschung von Punktwolken und reine Punktwolken betrachtet. Die Visualisierung erfolgt dabei als Darstellung der Geometrie und Radiometrie der Objekte in Form von dreidimensionalen Szenen. Zur Erzeugung dieser Szenen wird die Spiel-Engine Unity von Unity Technologies aus San Francisco, USA, benutzt, die die Entwicklung von Computerspielen und anderen interaktiven 3D-Grafik-Anwendungen unter Windows und Mac OS X erlaubt. Die Berechnung mit Unity erfolgt dabei in Echtzeit. Zielplattformen sind neben Arbeitsplatzrechnern auch Spielkonsolen, mobile Geräte und Webbrowser. Während die Basisversion kostenlos ist und für eigene Entwicklungen genutzt werden kann, bietet die kostenpflichtige Pro-Version erweiterte Funktionalität. DELOURA (2011) gibt einen Überblick über weit verbreitete, verfügbare und bekannte Game-Engines, die den Bereich von lizenzierten High-End bis freien Open-Source Game-Engines abdecken.

Obwohl die genannten Daten geometrische Informationen in drei Dimensionen liefern, werden

sie auf einem Standard-Monitor nur zweidimensional dargestellt. Erst durch stereoskopisches Sehen auf einem 3D-Bildschirm oder einer Projektionsleinwand entsteht ein entsprechender Tiefeneindruck, wenn stereoskopische oder 3D-Bilder dargestellt werden. Die Technik der stereoskopischen Darstellung besteht darin, zwei für das linke und rechte Auge leicht unterschiedliche Bilder anzuzeigen. Die dafür erforderlichen Bilder mit geringer horizontaler Parallaxe werden in Unity durch zweifache Berechnung der Szene von leicht versetzten virtuellen Kamerastandpunkten erstellt. Die Trennung der Bilder erfolgt dabei für die Augen in den eingesetzten Systemen durch spezielle Brillen. Bei sequentiell wechselnden Bildern für das rechte und linke Auge werden die Brillengläser mit einer Frequenz von 120 Hz synchronisiert verschlossen und geöffnet, sodass das Bild nur vom entsprechenden Auge wahrgenommen werden kann. Die beiden Bilder werden dann im Gehirn für die Tiefenwahrnehmung kombiniert, da das Gehirn die hohe Frequenz nicht so schnell verarbeiten kann. Bei nebeneinander dargestellten Bildern dient die Brille als Okular und fokussiert das linke Auge auf das linke Bild und das rechte Auge auf das rechte Bild.

3.1 3D-Modell

Das 3D-Modell wird durch manuelle Vorverarbeitung auf Fehler und Modellinkonsistenzen überprüft und anschließend in Unity eingebunden. Inkonsistenzen können durch fehlerhafte Texturzuweisung, Flächenüberscheidungen und uneinheitliche Flächennormalen entstehen. Unity bietet dabei den Modellimport über gängige Datenformate wie Wavefront (obj), 3DS, Collada (dae) und Filmbox (fbx) für den Austausch von 3D-Daten an. Das Format fbx bietet dabei für das CAD-Modell einen optimalen Datenfluss zwischen der Texturierung-Software (Cinema 4D) und Unity.

3.2 Punktwolke

Die Darstellung von Punktwolken erfolgt ebenso in einem Programm, welches mit Unity erzeugt wurde. Der Import der Punktwolke erfolgt aktuell über ASCII-Textdateien (ply-Format), die aufgrund der Verzahnung von Darstellung und Modell eine Größe von 500 MB pro Einzeldatei und 45 Millionen Punkte insgesamt nicht übersteigen dürfen.

4 Steuerung der interaktiven 3D-Visualisierung

Da die interaktive Visualisierung eine freie Bewegung durch das 3D-Modell ermöglichen soll, sind dafür Steuerbefehle durch geeignete Hardware zu erfassen und als Eingabedaten der Visualisierungssoftware zur Verfügung zu stellen. Als Steuerungsparameter sind Positions- und Rotationsänderungen einer frei beweglichen Kamera definiert, die durch den Anwender interaktiv festgelegt wird. Bislang wurden zwei solche Konzepte mit unterschiedlichen Optionen implementiert. Beim ersten Konzept werden die Positions- und Rotationsinformationen in Echtzeit direkt an eine virtuelle Kamera übergeben, die durch das Modell schwebt und eine Sicht aus jedem Blickwinkel (z.B. als Grundriss-Darstellung, Ansicht oder Isometrie) ermöglicht. Weiterhin nutzt diese Kamera keine Kollisionsabfrage für die Bewegung, sodass auch ein Durchfliegen von Flächen (Dächern und Wänden) zum Betreten bzw. Verlassen von Gebäuden möglich ist. Während das erste Konzept also ein „Durchfliegen“ des Modells erlaubt, bietet das

zweite Konzept ein „Durchlaufen“ der Szene an. Zur Steuerung wird die Kamera an einem Objekt in menschlicher Augenhöhe (~1,70m) befestigt, das durch das 3D-Modell bewegt wird. Die Schwerkraft ist dabei modelliert, sodass das Objekt nach Sprüngen wieder auf der Oberfläche zurückkehrt. Aus dieser Bindung an die Oberfläche folgt die Anforderung an das 3D-Modell, dass eine geschlossene Fläche zum Betreten vorhanden sein muss, die nicht stark von der Horizontalen abweicht (±20%). Da bei Punktwolken grundsätzlich keine Oberflächen vorhanden sind, erfolgt die Bewegung durch die jeweilige Punktwolke ohne Bezug zu einer Fläche. Die so erzielbare Steuerung bietet die Möglichkeit zu virtuellen Rundgängen durch Stadt- und Architekturmodelle aus der gleichen Perspektive wie ein realer Betrachter. Durch diese Art der Bewegung durch das 3D-Modell oder durch die Punktwolke erhöht sich der Grad der Immersion, also des Gefühls der Verbindung mit dem Objekt. In beiden Konzepten werden die Positions- und Rotationsparameter den Kamerakoordinaten bzw. Rotationsmatrizen angehängt, wobei im zweiten Konzept noch Bewegungseinschränkungen durch Kollisionsabfragen berücksichtigt werden müssen.
Für die Konzeption des Systems wurde vorab festgelegt, dass die Software zu Erfassung der Steuerungsinformationen von der Visualisierungssoftware getrennt erstellt wird. So ist eine Nutzung der erfassten Daten in anderen Programmen und eine Implementation für neue Steuerungshardware leichter möglich. Diese logische Trennung der Steuerungs- und Visualisierungssoftware stellt auch eine physikalische Trennung dar, indem die Steuerungshardware an einem anderen System als dem Visualisierungssystem angeschlossen werden kann. Zur Übertragung der Steuerungsparameter wird das Netzwerkprotokoll UDP bzw. TCP des Ethernet-Protokollstapels TCP/IP eingesetzt. Exemplarisch für die Fülle an möglicher Steuerungshardware werden hier das Gamepad der Playstation3 (Dualshock3) und die natürliche Benutzerschnittstelle (Natural User Interface NUI) der XBox360, die Kinect, vorgestellt.

4.1 Playstation3-Controller Dualshock3

Der Playstation3-Controller (Abb. 2 links) besteht aus einem Steuerkreuz, zwei Joysticks und einem zweiachsigen Neigungssensor sowie einigen Tasten. Die für die Visualisierung notwendigen Steuerungsparameter stehen relativ simpel zur Verfügung, da eine Auslenkung der Joysticks die erforderlichen Informationen direkt bereitstellt. So wurde für eine Positionsänderung der linke Joystick gewählt, während die Rotation über den rechten Joysticks gesteuert wird, wobei eine Gierbewegung der Vertikalachse durch Auslenkung nach links und rechts und eine Nickbewegung der Querachse durch Auslenkung nach vorne und hinten entsteht. Die Implementation wurde als Software für den Windows-PC in C# mit der DirectX API SharpDX erstellt. Die Orientierungsdaten werden anschließend intern im PC oder im Netzwerk versandt. Für die mobile Version auf einem Raspberry Pi (scheckkarten-großer Computer mit einer Platine) mit dem Debian Wheezy Betriebssystem dient ein Python-Skript auf Basis von Pygame, ein für die Spieleprogrammierung dienendes Set von plattformübergreifenden Python-Modulen, für das Auslesen der Steuerungsparameter des Gamepads. Die Steuerungsdaten werden über die Netzwerkkarte oder einen WLAN-Stick an die Visualisierungssoftware übertragen.

Abb. 2: Playstation3 Gamepad Dualshock3 (links) und XBox360 NUI Kinect (rechts)

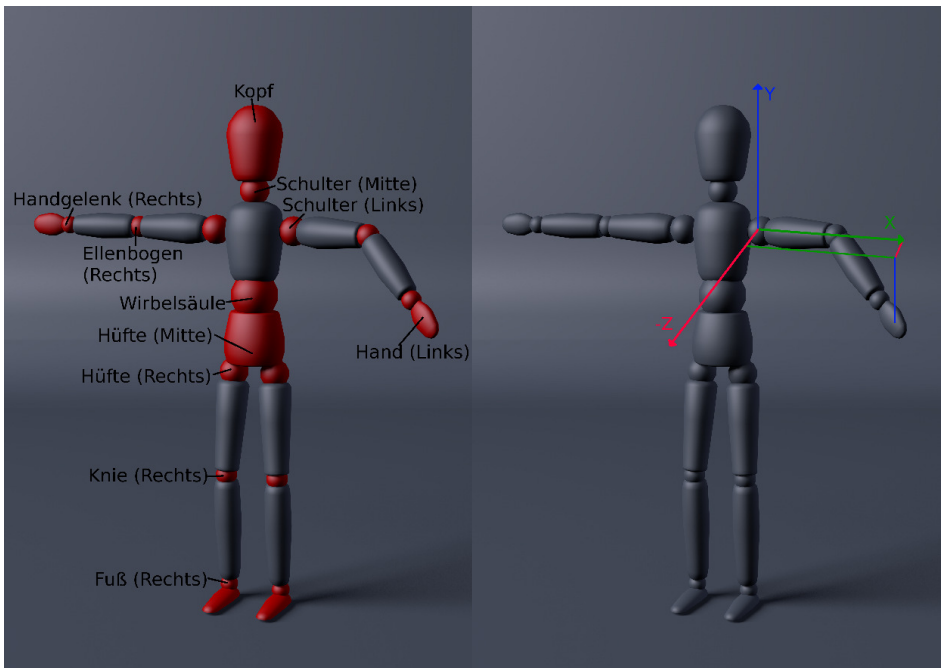

Abb. 3: Körperknoten der Kinect (links), Koordinatendefinition zur Ableitung von Steuerungsparametern (rechts)

4.2 Microsoft Kinect

Sehr viel schwieriger stellt sich die Auswertung der Steuerungsdaten mit der Microsoft Kinect (Abb. 2 rechts) dar. Die Kinect (abgeleitet vom Englischen kinetic connect) ist eine von Microsoft zusammen mit der Firma PrimeSense entwickelte Hardware zur Steuerung der Videospielkonsole Xbox 360 (FREEDMAN et al. 2008; MICROSOFT 2010), die seit Anfang November 2010 verkauft wird. Spieler können damit anstatt mittels herkömmlicher Gamepads allein durch Körperbewegungen die Software bedienen (WEBB & ASHLEY 2012). Sie besteht aus einem Mikrofonarray, einer Farbkamera, einer Tiefenkamera und einem Neigungssensor. Daher wird sie auch als RGB-D Sensor klassifiziert, bei dem RGB die drei Farbkomponenten und D die Tiefe beschreibt. Algorithmen werten die so erfassten Daten aus und stellen als Ergebnis die Position einer Person sowie seiner einzelnen Körperteile dar. Die ausgewerteten Positionen

beschreiben den menschlichen Körper anhand von 20 Knoten (Position und Funktion, siehe Abb. 3 links). Um diese Daten zur Steuerung nutzen zu können, müssen die einzelnen Körperteile in Bezug zueinander gesetzt und als Gesten ausgewertet werden. Für den Steuerungsparameter Position wird die linke Hand verwendet. Dafür wird die Koordinate der linken Hand in das Koordinatensystem transformiert, welches als Verbindung von rechter zu linker Schulter (horizontal) aufgespannt wird. Der Abstand der Hand zur linken Schulter ergibt dabei die Positionsänderung entsprechend der definierten Koordinatenachsen (Abb. 3 rechts). Die Auswertung für die Rotation erfolgt analog mit der rechten Hand und als Bezugspunkt dient die rechte Schulter. Die Positionsdifferenz wird in Polarkoordinaten umgewandelt und als Steuerungsparameter übertragen.

5 Implementierungen der interaktiven Visualisierung

Als erstes System wurde eine Implementierung auf einem kommerziellen 3D-Monitor mit dem 3D Vision System von Nvidia (Abb. 4 links) untersucht. Dank des Einsatzes von 3D Vision ist keine stereoskopische Implementation notwendig, da die 3D-Information im Rahmen des Echtzeitrenderings bekannt ist und so ein zweites Bild als Stereopartner automatisch generiert werden kann (NVIDIA, 2010). Die interaktive Darstellung von 3D-Modellen und Punktwolken ist auf dem 3D-Monitor möglich, in dem wahlweise mit Maus und Tastatur oder mit dem Gamepad die Steuerung übernommen wird. Eine Nutzung der Kinect bietet sich hier nicht an, da der Betrachter am PC sitzt und so nur ein begrenzter Bereich der Körpers für die mögliche Steuerung sichtbar ist. Weiterhin ist der Abstand des Anwenders zum System zu gering, um eine gute Ausnutzung des Gesichtsfeldes der Kinect-Kamera zu erreichen.
Die zweite Implementation erfolgte für ein elektronisches Whiteboard-System. Es handelt sich um ein eBoard von Legamaster mit einer Projektionsfläche (diagonal) von 88“ (http://www.e-legamaster.com/de/). Dieses System unterstützt nur die Darstellung der Bilder als Top-Bottom-3D, was eine simultane Berechnung der beiden virtuellen Kamerastandpunkte zwingend notwendig macht. Weiterhin werden die beiden Halbbilder in einem Vollbild übereinander dargestellt, was die reale Auflösung in vertikaler Richtung halbiert. Da die Darstellung im Seitenverhältnis von 16:10 erscheint, muss das Rendering dieses bereits einplanen und die Bilddaten in 16:20 bereitstellen. Zur Steuerung eignet sich hier weder Maus noch Tastatur, da keine Abstellfläche (Tisch) vorhanden ist und der Nutzer meistens steht. Entsprechend bieten sich hier vor allem berührungslose und drahtlos verbundene Eingabegeräte wie die Kinect oder das Gamepad als optimale Steuerung der Visualisierung an. Da die Darstellung der Objekte vom angeschlossenen PC auf dem Beamer erfolgt, können alle Modelle visualisiert werden.

Abb. 4: 3D-Monitor mit Visualisierung durch Nvidia 3D Vision (links) und Smartphone mit stereoskopischen Bildpaar in einer Halterung als Virtual-Reality-Brille (rechts)

Als drittes System wurde die interaktive 3D-Visualisierung auf einem Smartphone implementiert. Der Erfindung und dem Konzept von OpenDive folgend wird aus einem Android-Smartphone und einem Plastikgestell aus dem 3D-Drucker eine Virtual-Reality-Brille als einfaches Head-Mounted-Display (HMD) (JANSSEN, 2013). Dabei dient das Plastikgestell als Halterung, mit der das Smartphone, auf dem zwei Teilbilder nebeneinander dargestellt werden, am Kopf befestigt wird (Abb. 4 rechts). Die Halterung kann als Bausatz in Form einer STL-Datei durch einen 3D-Drucker ausgegeben und dabei entsprechend auf das Ausgabegerät Smartphone noch angepasst werden. Die Fokussierung erfolgt mit einem in der Halterung befestigten Linsenstereoskop, dessen Linsen individuell in Augenabstand und Objektabstand verstellbar sind. Die Steuerung erfolgt zweigeteilt, d.h. die Rotationsänderungen werden vom im Smartphone enthaltenen Gyroskop erfasst und direkt zur Anzeige genutzt. Für die Positionsänderung kommt ein externes Steuerungsgerät zum Einsatz. Aufgrund der Tatsache, dass man unter dem HMD die Umgebung nicht mehr sehen kann, erscheint für die Steuerung nur ein Gerät mit entsprechender haptischer Wahrnehmung sinnvoll zu sein. In dieser Untersuchung hat sich nur das Gamepad als ideale Steuerungsmöglichkeit herausgestellt, da die individuelle Form des Gamepads den ergonomischen Einsatz ohne Sichtkontakt möglich macht. Die Darstellung von Punktwolken ist bei diesem System jedoch noch nicht möglich, da die Berechnung für die Darstellung der Punktwolken mit DirectX 11 erfolgt und das genutzte Android-Smartphone mit OpenGL ES arbeitet. Die Darstellung von 3D-Modellen ist möglich, allerdings sind gewisse Einbußen in der Darstellungsqualität hinzunehmen, um sie auf dem Smartphone berechnen zu lassen.

6 Fazit und Ausblick

In diesem Beitrag wurden unterschiedliche Steuerungsmöglichkeiten für die interaktive 3D-Visualisierung von an der HCU Hamburg erzeugten 3D-Modellen und Punktwolken vorgestellt. Für die Echtzeit-Generierung der 3D-Szenen in den bereitgestellten 3D-Modellen wurde die Spiel-Engine Unity verwendet, durch die die interaktive 3D-Visualisierung auf verschiedenen Zielplattformen wie PCs und mobilen Geräten gezeigt werden konnte. Die Implementierung der Steuerungssoftware auf verschiedene Low-Cost-Systeme (3D-Monitor und Smartphone) wurde beschrieben. Dazu konnten auch die bereits an der HCU Hamburg vorhandenen Systeme zur Visualisierung der 3D-Objekte (3D-Powerwall, Whiteboard, 3D-Monitor und Smartphone) und Steuerungssensoren (Playstation3-Controller Dualshock3 und Microsoft Kinect) verwendet werden. In zukünftigen Arbeiten könnten die 3D-Modelle oder Punktwolken mit der realen Umgebung verknüpft werden (Augmented Reality). So könnte der Einsatz von 3D-Modellen oder Punktwolken z.B. für museale oder archäologische Anwendungen eingebettet in Präsentationen in situ ermöglicht werden.
Weiterhin werden im nächsten Jahr einige neue Produkte auf dem Markt kommen, die die Darstellung z.B. durch die VR-Brille Oculus Rift, die ein besonders großes Sichtfeld und sehr schnelle Bewegungssensoren aufweist, oder auch die Steuerung z.B. durch das Myo-Armband von ThalmicLabs (THALMICLABS, 2014) signifikant verbessern könnten. Zuletzt bietet gerade das OpenDive-Konzept eine Visualisierungsplattform zu einem günstigen Preis an, da viele potentielle Anwender das Smartphone als Ausgabegerät bereits in der Tasche tragen. Nur das entsprechende Plastikgestell als Halterung der OpenDive-Brille müsste noch angeschafft werden. Somit könnte eine günstige Form der 3D-Visualisierung von Objekten einer breiten Öffentlichkeit zugänglich gemacht werden, zumal Wissenschaftler der ETH Zürich bereits eine App entwickelt haben, mit der ein Smartphone zum 3D-Scanner wird (TANSKANEN et al. 2013).

7 Literaturverzeichnis

DELOURA, M., 2011: Game Engine Survey 2011, Game Developers Magazine, May, 7-12, http://impactjs.com/files/GDM_May_2011.pdf

FREEDMAN, B., SHPUNT, A., MACHLINE, M., & ARIELI, Y., 2008: Depth mapping using projected patterns. October 9 2008, WO Patent WO/2008/120,217.

JANSSEN, J.-K., 2013: Zum Eintauchen: Android-Smartphone als Virtual-Reality-Brille. c't - Magazin für Computertechnik, Heft 15, 64-65.

KERSTEN, TH., ACEVEDO PARDO, C., 2003: Wie kommt das Ahrensburger Schloss in den Computer? - 3D-Gebäudeerfassung und Visualisierung durch digitale Photogrammetrie. DenkMal! Schleswig Holstein, Zeitschrift für Denkmalpflege in Schleswig Holstein, Landesamt für Denkmalpflege in Schleswig-Holstein (Hrsg.), Jahrgang 10, 2003, Westholsteinische Verlagsanstalt Boyens & Co., Heide., 48-54.

KERSTEN, TH., EILMUS, B., LINDSTAEDT, M., ACEVEDO PARDO, C., 2003: 3D-Erfassung und Visualisierung des Celler Schlosses durch digitale Architekturphotogrammetrie. Photogrammetrie, Laserscanning, Optische 3D-Messtechnik - Beiträge der Oldenburger 3D-Tage 2003, Th. Luhmann (Hrsg.), Wichmann Verlag, Heidelberg, 213-222.

KERSTEN, TH., ACEVEDO PARDO, C., LINDSTAEDT, M., 2004: 3D Acquisition, Modelling and

Visualization of north German Castles by Digital Architectural Photogrammetry. IAPRS, Vol. XXXV, Part B2, 126-132.

KERSTEN, TH., BIEBERMANN, M., SCHNEIDER, M., 2006: 3D-Erfassung und Modellierung des Duderstädter Westerturmensembles durch Kombination von digitaler Architekturphotogrammetrie und terrestrischem Laserscanning. Photogrammetrie, Laserscanning, Optische 3D-Messtechnik - Beiträge der Oldenburger 3D-Tage 2006, Th. Luhmann/C. Müller (Hrsg.), Wichmann Verlag, Heidelberg, 254-263.

KERSTEN, TH., 2006: Kombination und Vergleich von digitaler Photogrammetrie und terrestrischem Laserscanning für Architekturanwendungen. Publikationen der DGPF e.V., Band 15, E. Seyfert (Hrsg.), 247-254.

KERSTEN, TH., BÜYÜKSALIH, G., BAZ, I., JACOBSEN, K., 2009: Documentation of Istanbul Historic Peninsula by Kinematic Terrestrial Laser Scanning. The Photogrammetric Record, 24(126): 122-138 (June 2009).

KERSTEN, TH., TILSNER, A., JAQUEMOTTE, I., SIEH, W., 2010: 3D-Erfassung und Modellierung des Bismarck-Denkmals durch terrestrisches Laserscanning zur Integration in das Hamburger Stadtmodell. AVN - Allgemeine Vermessungs-Nachrichten, 5/2010, 163-169.

KERSTEN, TH., LINDSTAEDT, M., 2012a: Automatic 3D Object Reconstruction from Multiple Images for Architectural, Cultural Heritage and Archaeological Applications Using Open-Source Software and Web Services. PFG, Heft 6, 727-740.

KERSTEN, TH., LINDSTAEDT, M., 2012b: Virtual Architectural 3D Model of the Imperial Cathedral (Kaiserdom) of Königslutter, Germany through Terrestrial Laser Scanning. EuroMed 2012 - Int. Conference on Cultural Heritage, Ioannides, M.; Fritsch, D.; Leissner, J.; Davies, R.; Remondino, F.; Caffo, R. (Eds.), Lecture Notes in Computer Science (LNCS), Volume 7616, Springer-Verlag Berlin Heidelberg, 201-210.

KERSTEN, TH., MECHELKE, K., 2013: Fort Al Zubarah in Katar – 3D-Modell aus Scanner- und Bilddaten im Vergleich. AVN - Allgemeine Vermessungs-Nachrichten, 2/2013, 50-58.

LINDSTAEDT, M., KERSTEN, TH., MECHELKE, K., GÖTTING, M., HEIDEN, R., 2008: Virtuelles 3D-Modell der antiken Tempelanlage in Sirwah/Jemen zur archäologischen Objektdokumentation durch terrestrisches Laserscanning und Photogrammetrie. Publikationen der DGPF e.V., Band 17, E. Seyfert (Hrsg.), 59-68.

MICROSOFT, 2010: PrimeSense Supplies 3-D-Sensing Technology to "Project Natal" for Xbox 360. http://www.microsoft.com/en-us/news/press/2010/mar10/03-31primesensepr.aspx

NVIDIA, 2010: Nvidia 3D Vison Pro And Stereoscopic 3D – White Paper. http://www.nvidia.com/docs/IO/40505/WP-05482-001_v01-final.pdf, abgerufen am 28.01.2014.

TANSKANEN, P., KOLEV, K., MEIER, L., CAMPOSECO, F., SAURER, O., POLLEFEYS, M., 2013: Live Metric 3D Reconstruction on Mobile Phones. IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, December 1-8.

THALMICLABS, 2014: Introducing Myo™ - Gesture Control Armband. www.thalmic.com/en/myo/ (abgerufen am 31.01.2014)

WEBB, J. & ASHLEY, J., 2012: Beginning Kinect Programming with the Microsoft Kinect SDK. Apress, Berkley, CA, USA, 85-100.

ZANINI, M., 1998: Dreidimensionale synthetische Landschaften - Wissensbasierte dreidimensionale Rekonstruktion und Visualisierung raumbezogener Informationen. Diss. ETH Nr. 12893, IGP Mitteilungen Nr. 66, ETH Zürich, 179 S.