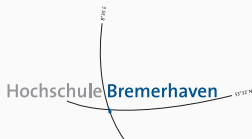


Digital Humanities im Museum 4.0

betreut durch Prof. Dr. Oliver Radfelder



Julia Albers, Jannick Bock, Felix Harms,
Luca Junge, Pepijn Kampschöer, Leon Leymann,
Christian Lüth, Björn Schultz, Fabian Sehm, André Stadtmeyer

Einleitung

Photogrammetrie

Rekonstruktionssoftware

AliceVision

Meshoptimierung

Modelldarstellung & -Interaktion

Einleitung Übersicht

Digital Humanities

Museum 4.0

Open Source

Gängige Lizenzen

Einleitung

Digital Humanities

- Digitale Geisteswissenschaften
- Digital Humanities, eHumanities (enhanced Humanities)
- Geisteswissenschaften umfassen alle Aspekte der menschlichen Gesellschaft
- Nutzung der digitalen Arbeits- und Medienwelt

Der Verband DHd über digitale Geisteswissenschaften [1]

Einleitung

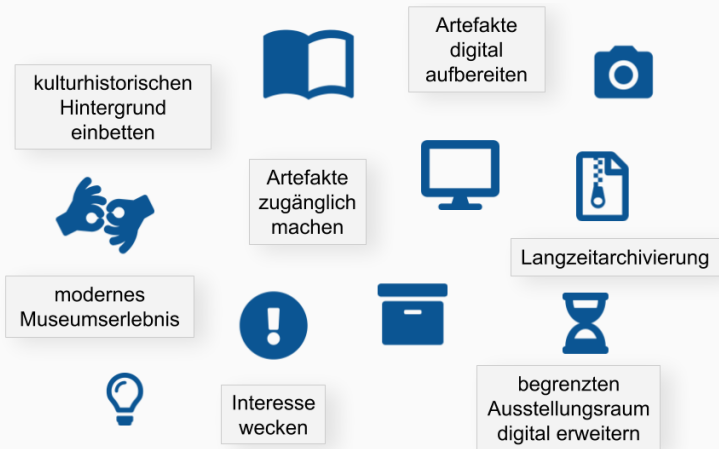
Museum 4.0

“Unser Ziel ist es, Menschen auf das Museum als Erkenntnis- und Erlebnisraum neugierig zu machen, die Auseinandersetzung mit musealen Themen zu fördern sowie Möglichkeiten für aktive Teilhabe zu schaffen.”

Ziele des Projekts “museum4punkt0.de” [2]

Einleitung

Museum 4.0



Einleitung

Open Source

- Unterstützung des Open Source Gedanken
- Verfügbarkeit für jedermann (z.B. Museen)
- Definition [3]
 - Freier Zugang zum Quellcode
 - Freie Weitergabe der Software
 - Beliebige Modifikation der Software (Copyleft)
 - Keine Einschränkungen der Nutzung

Einleitung

Gängige Lizenzen

- GNU General Public License (GPL) [4]
- MIT-Lizenz [5]
- Apache-Lizenz [6]
- BSD-Lizenz [7]

Photogrammetrie Übersicht

Grundlagen

Bildserie

Erfahrungswerte zu den Fotos

Verarbeitung der Bild-Serie

Photogrammetrie

Grundlagen

- Ursprünge in der Vermessungstechnik
- Bilderserie
- Extraktion der Bildinformationen
- Bildung der Punktwolke
- Erzeugung des Wireframe
- Erzeugung des Meshs

Photogrammetrie

Bildserie



Photogrammetrie

Erfahrungswerte zu den Fotos

- Spiegelnde Flächen und Fenster erzeugen Fehler
- Gleichmäßiger Abstand beim Fotografieren
- Stativ wirkt unterstützend
- Direkte Sonneneinstrahlung vermeiden

Photogrammetrie

Verarbeitung der Bild-Serie

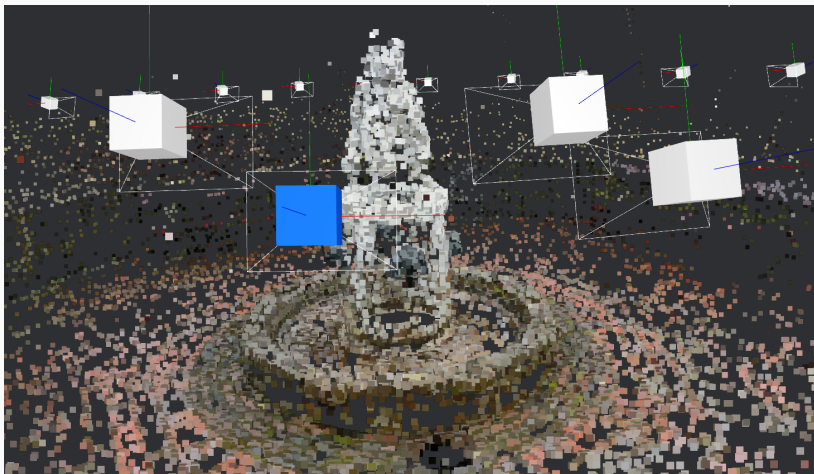


Abbildung 1: Punktwolke des Klabaüterbrunnens

Photogrammetrie

Erzeugung des 3D-Modells

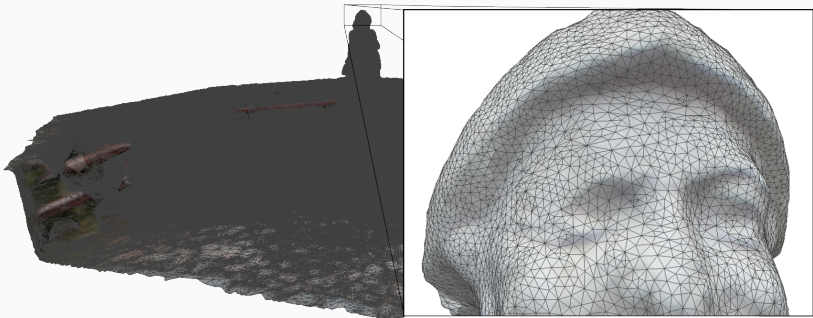


Abbildung 2: Wireframe des Klabauterbrunnen

Photogrammetrie

Erzeugung des 3D-Modells

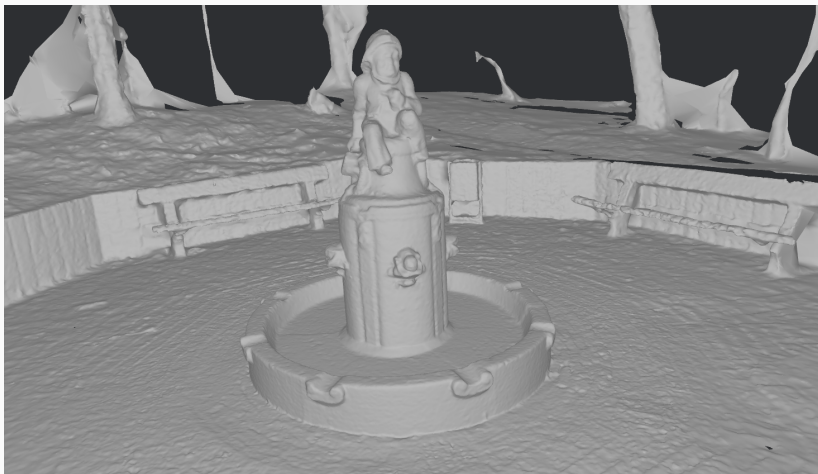


Abbildung 3: untexturiertes Mesh

Photogrammetrie

Erzeugung des 3D-Modells



Abbildung 4: texturiertes Mesh

Rekonstruktionssoftware Übersicht

Was ist Rekonstruktionssoftware?

Welche Software haben wir genutzt?

Rekonstruktionssoftware

Was ist Rekonstruktionssoftware?

- Software zum Berechnen von 3D-Modellen aus Bildern
- Open Source in unserem Projekt
- Pipeline Konzept
- Im Projekt wurden mehrere Anwendungen genutzt

Rekonstruktionssoftware

Welche Software haben wir genutzt?

- VisualSfM [8]
- MVE [9]
- COLMAP [10] [11]
- AliceVision [12]

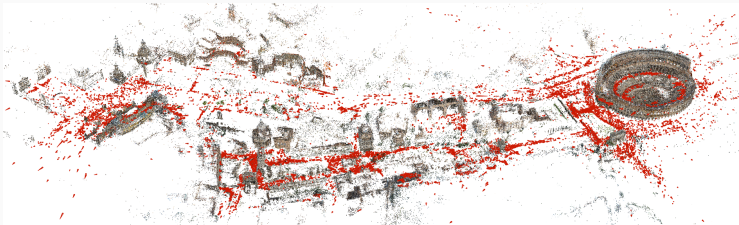


Abbildung 5: Punktwolke Roms Innenstadt mit COLMAP [10]

AliceVision Übersicht

Modularität

Meshroom

AliceVision

Modularität

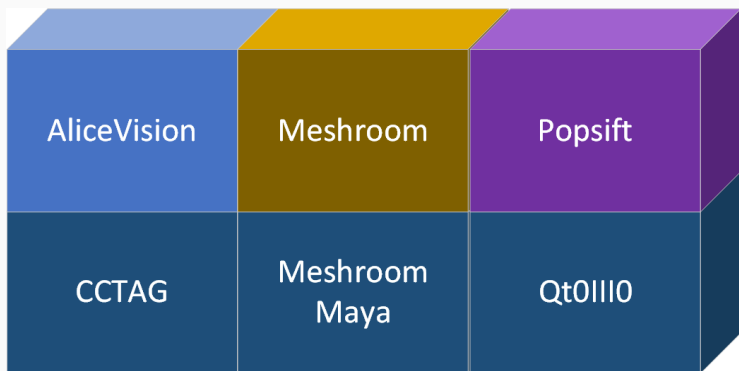


Abbildung 6: Einzelne Module des Frameworks

AliceVision

Meshroom

Features:[13]

- Python
- Bietet eine intuitive Benutzeroberfläche
- Out-of-the-Box-Lösung
- Graphenorientierte Modellerstellung

AliceVision Meshroom

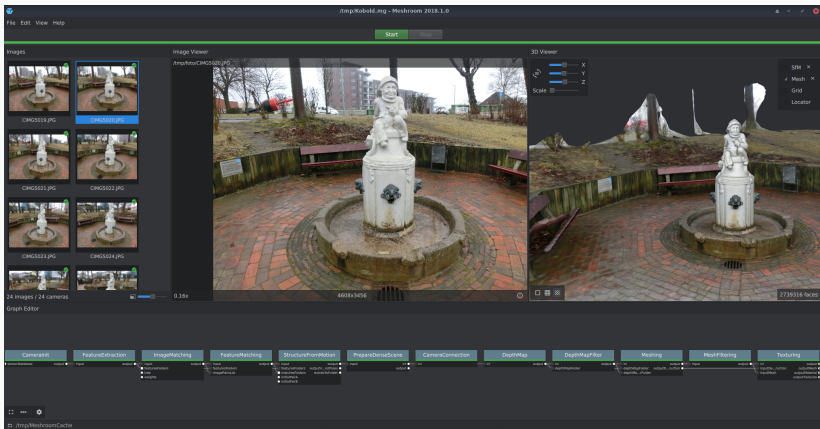


Abbildung 7: Meshroom

AliceVision

Meshroom

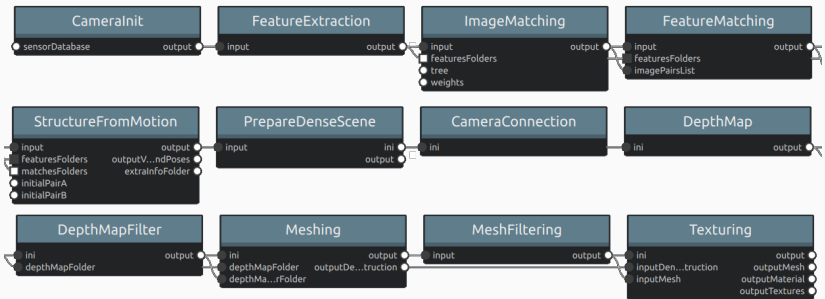


Abbildung 8: Rekonstruktionsgraph aus Meshroom

Meshoptimierung Übersicht

Ergebnis aus AliceVision

Optimierung

Komprimierung

Ergebnis der Optimierung

Meshoptimierung

Ergebnis aus AliceVision

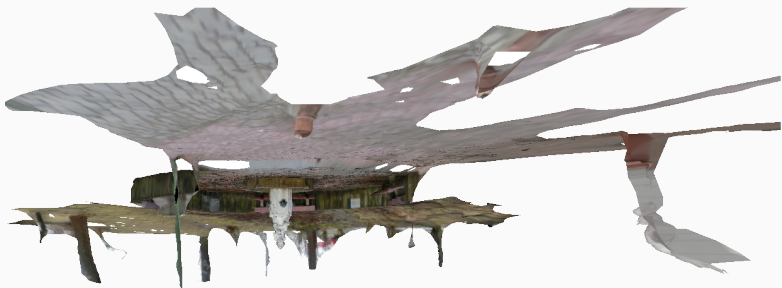


Abbildung 9: Ergebnis aus AliceVision

Meshoptimierung

Optimierung

- Probleme:
 - Fehlerhafte Ausrichtung
 - Überflüssige "Inhalte" eines Modells
- Lösung:
 - Bearbeitung des Modells mit Blender & Meshlab

Meshoptimierung

Komprimierung

- Ziel: Modellgröße verringern
- Punkte werden aus Wolke entfernt
 - Veränderungen an der Oberfläche möglichst gering
- Diverse Methoden zur Komprimierung
 - Clustering Dezimierung
 - Kanten-Kollaps-Transformation
 - Quadric Edge Collapse Dezimierung

Meshoptimierung

Ergebnis der Optimierung



Abbildung 10: Aufbereitetes Modell

Modelldarstellung & -Interaktion

Übersicht

Darstellung des Modells

Entwickelte Infrastruktur

Interaktion durch ein eingebettetes System

Interaktion durch touchfähige Endgeräte

Interaktion durch WebVR

Modelldarstellung & -Interaktion

Darstellung des Modells

Modelle sollen im Web dargestellt werden

- three.js als JavaScript-Bibliothek zur Darstellung[14]
 - arbeitet Browserübergreifend
 - Open Source Software
- Webbasierte Darstellung ermöglicht eine Client-Server-Struktur

Modelldarstellung & -Interaktion

Darstellung des Modells

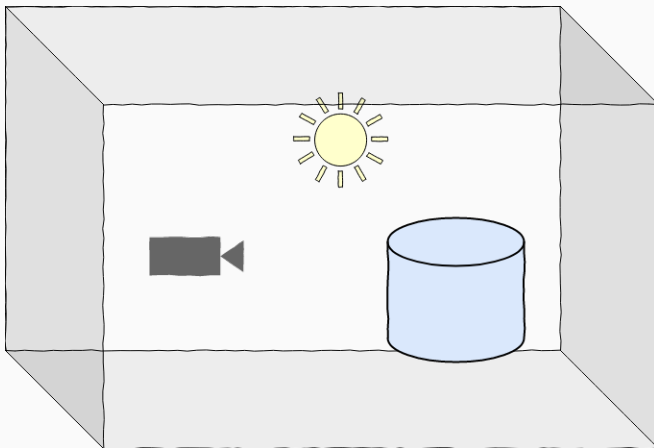


Abbildung 11: Grundaufbau einer three.js Szene

Modelldarstellung & -Interaktion

Entwickelte Infrastruktur

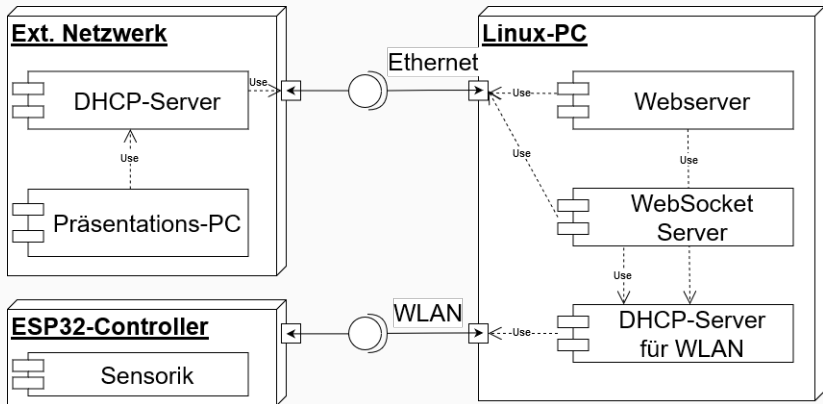


Abbildung 12: Netzwerkdiagramm am Controller-Beispiel

Modelldarstellung & -Interaktion

Interaktion durch ein eingebettetes System

- Controller aus ESP32-Chip, Gyroskop und Tastern
 - Gyroskop steuert Drehung von Modellen
 - Taster steuern Modellwechsel oder Zoom
- Controller sendet nur bei Statuswechsel der Komponenten
- 2.000 mAH PowerBank reicht für mehrere Tage Betrieb

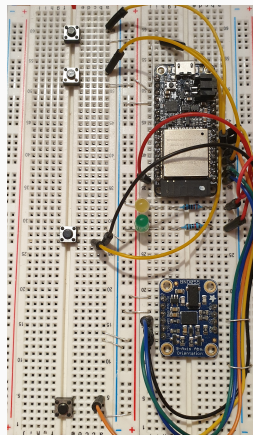


Abbildung 13:
Controller auf einem
Breadboard

Modelldarstellung & -Interaktion

Interaktion durch touchfähige Endgeräte

- Steuerung durch Smartphone oder Tablet
- 3D-Objekte mit three.js auf Modell-Webseite
- Steuerungs-Webseite mit Touch-Bedienelementen
- Übertragung der Steuerungsdaten als JSON über Web-Socket-Verbindung

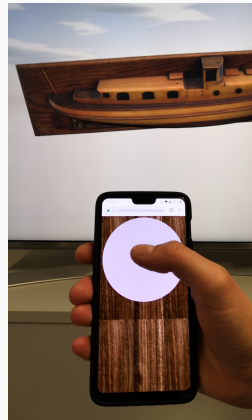


Abbildung 14:
Touch-Steuerung mit
Modell

Modelldarstellung & -Interaktion

Interaktion durch WebVR

Features:[15]

- Nutzung von VR-Brillen bietet neue Dimensionen der Wahrnehmung
- Anwendung in vielen Bereichen
 - Stadtführungen, Konzerte, Kunstausstellungen
- three.js stellt Schnittstelle zur Verfügung

**Vielen Dank
für Ihre Aufmerksamkeit!**

- [1] SAHLE, Patrick: *Digitale Geisteswissenschaften*. <https://dig-hum.de/digitale-geisteswissenschaften>. Version: 2011. – besucht am 22. Januar 2019
- [2] MUSEUM4PUNKT0: *Über uns*. <https://www.museum4punkt0.de/>. Version: 2019. – abgerufen am 15. Januar 2019
- [3] RENNER, Thomas ; VETTER, Michael ; REX, Sascha ; KETT, Holger: *Open Source Software*. <https://wiki.iao.fraunhofer.de/images/6/63/Fraunhofer-Studie-Open-Source-Software.pdf>. Version: 2005
- [4] STALLMAN, Richard: *GNU General Public License - GPL*. <https://www.gnu.org/licenses/gpl-3.0>. – besucht am 22. Januar 2019
- [5] OPEN SOURCE INITIATIVE: *The MIT License*. <https://opensource.org/licenses/MIT>. – besucht am 22. Januar 2019
- [6] APACHE SOFTWARE FOUNDATION: *Apache License*. <https://www.apache.org/licenses/LICENSE-2.0>. – besucht am 22. Januar 2019
- [7] UNIVERSITY OF CALIFORNIA, BERKELEY: *Berkeley Software Distribution - BSD*. <https://de.wikipedia.org/wiki/BSD-Lizenz>. – besucht am 22. Januar 2019
- [8] WU, Changchang: *VisualSFm: A Visual Structure from Motion System*. <http://ccwu.me/vsfm/>. – besucht am 22. Januar 2019
- [9] FUHRMANN, Simon ; LANGGUTH, Fabian ; GOESELE, Michael: MVE – A Multi-View Reconstruction Environment. In: *Eurographics Workshop on Graphics and Cultural Heritage*, 2014
- [10] SCHÖNBERGER, Johannes L. ; FRAHM, Jan-Michael: Structure-from-Motion Revisited. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016
- [11] SCHÖNBERGER, Johannes L. ; ZHENG, Enliang ; POLLEFEYS, Marc ; FRAHM, Jan-Michael: Pixelwise View Selection for Unstructured Multi-View Stereo. In: *European Conference on Computer Vision (ECCV)*, 2016

- [12] ALICEVISION: *About*. <https://alicevision.github.io/#about>. Version: 2018. – besucht am 17. Dezember 2018
- [13] ALICEVISION: *Meshroom*. <https://alicevision.github.io/#meshroom>. Version: 2018. – besucht am 17. Dezember 2018
- [14] THREEJS: *three.js*. <https://threejs.org/>. – besucht am 22. Januar 2019
- [15] IMMERSIVE WEB COMMUNITY GROUP: *Welcome to the immersive web*. <https://webvr.info/>. Version: Mai 2018