Photogrammetry using Stereo Image Pairs

Anfängerpraktikum • Justin Sostmann • Manuel Trageser

Photogrammetrie

- **Definition:**
  “Photogrammetrie […] ist eine Gruppe von berührungslosen Messmethoden und Auswerteverfahren, um aus Fotografien eines Objektes durch Bildmessung seine Lage und Form indirekt zu bestimmen […].”

https://de.wikipedia.org/wiki/Photogrammetrie

Photogrammetrie

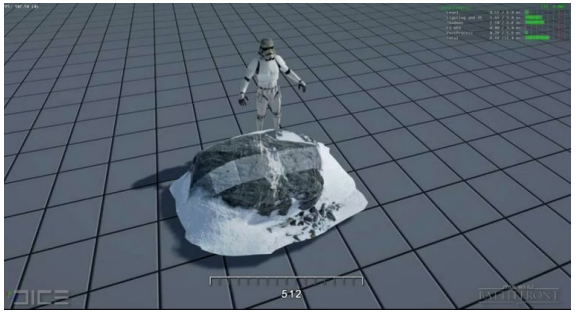
- **Definition:**
  “Photogrammetrie […] ist eine Gruppe von berührungslosen Messmethoden und Auswerteverfahren, um aus Fotografien eines Objektes durch Bildmessung seine Lage und Form indirekt zu bestimmen […].”
- **Generell:**
  Tiefen-Rekonstruktion aus (Stereo) Bildern.

https://de.wikipedia.org/wiki/Photogrammetrie

Anwendung

- Landvermessung
  - Google Earth
- Film- und Spielindustrie
  - High-Poly Meshes
- Autonomes Fahren

https://earth.google.com



https://www.ea.com/frostbite/news/photogrammetry-and-star-wars-battlefront

Arten der Photogrammetrie

- Shape from shading
  - Licht und Schatten
- Structure from motion
  - Überlappung zeitversetzter Bilder

Arten der Photogrammetrie

- Shape from shading
  - Licht und Schatten
- Structure from motion
  - Überlappung zeitversetzter Bilder
- **Stereophotogrammetry**
  - Keypoint Triangulation

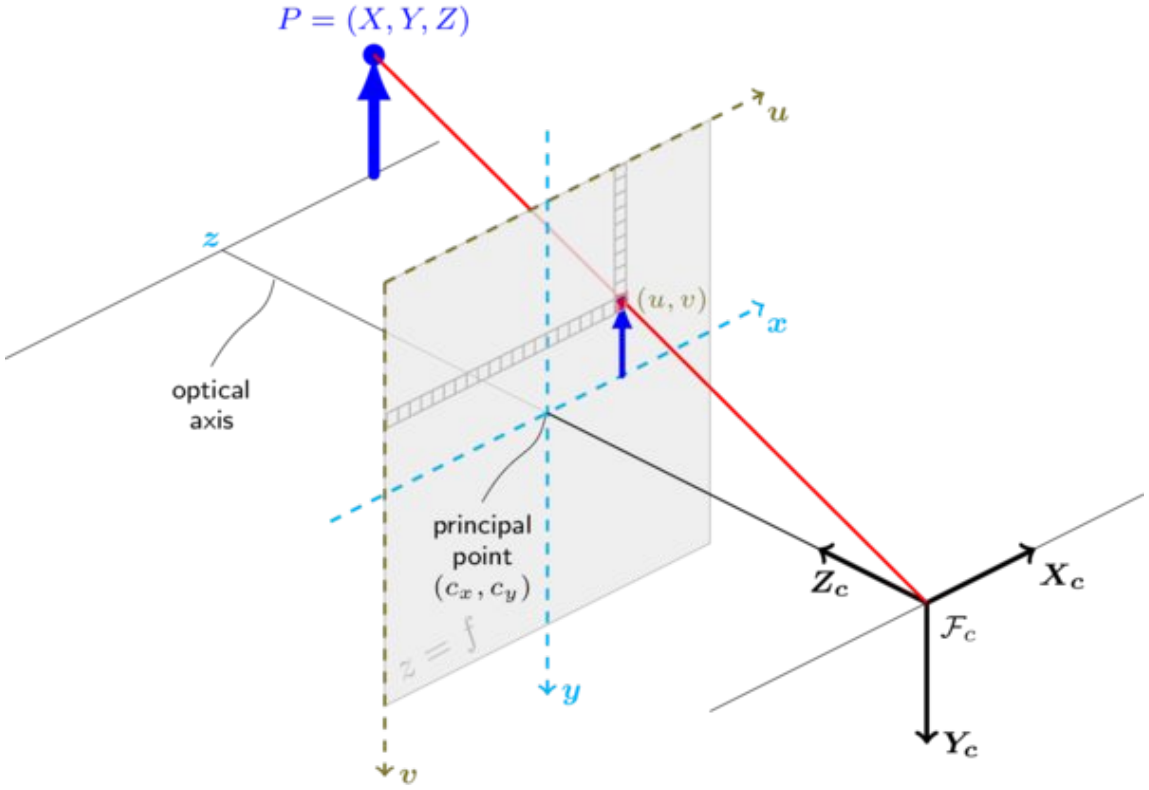
Ziel des Praktikums

- Methoden zur Bildverarbeitung in Python
- Mesh Generation in Python
- Pinhole Camera Model

➜ 3D Modelle aus Bildern

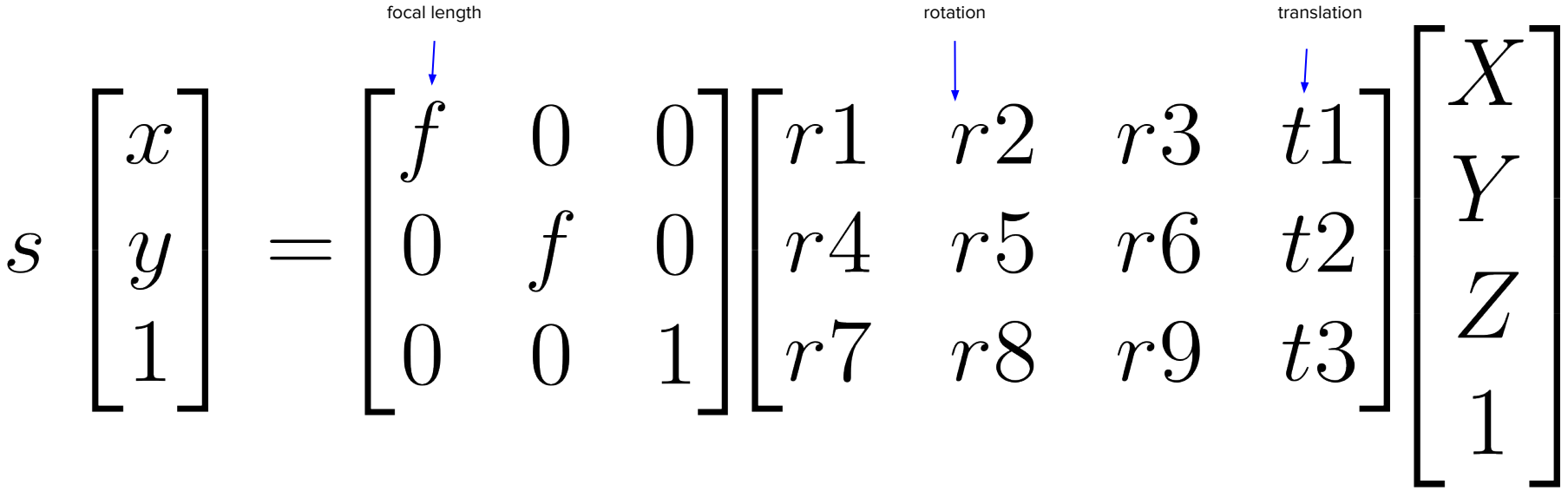
Theorie

Pinhole Camera

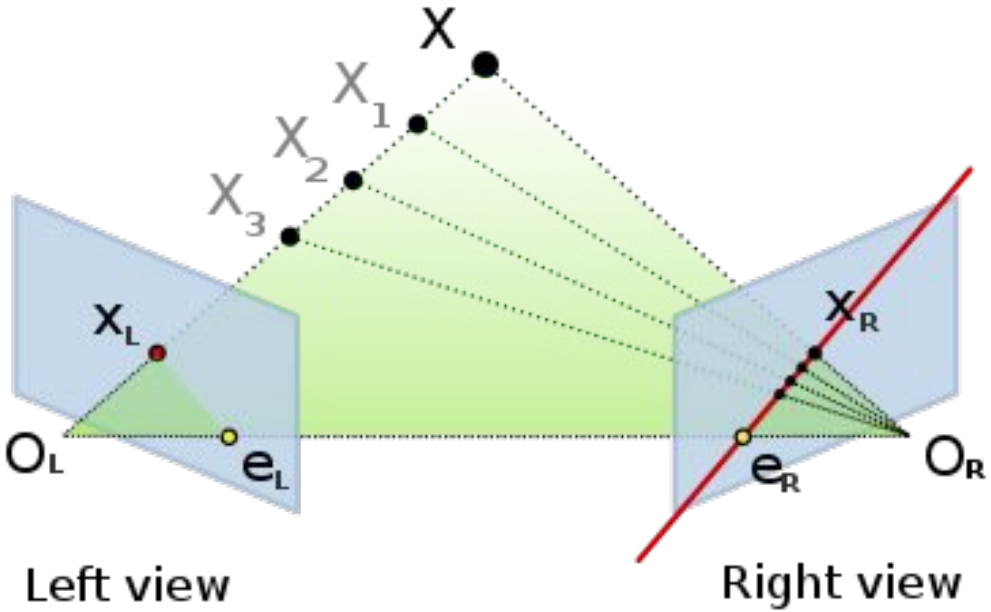


https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

Intrinsic/Extrinsic Matrix

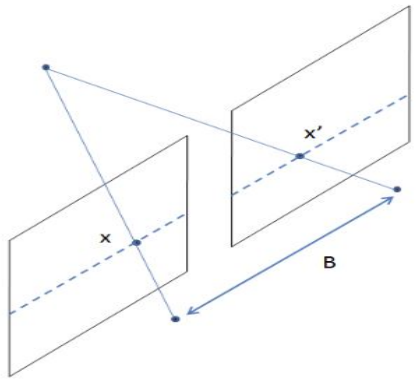


Stereo Vision



https://upload.wikimedia.org/wikipedia/commons/1/14/Epipolar_geometry.svg

Ideal Case



Co-Planar

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u0 \\ 0 & f & v0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -B \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\longrightarrow Z = f * \underbrace{(x - x')}_{\text{Disparity}} / B$$

https://towardsdatascience.com/3-d-reconstruction-with-vision-ef0f80cbb299

Example

Example

Example

Playground

https://ksimek.github.io/perspective_camera_toy.html

Praxis

Image Processing

- Python
- OpenCV
  - Bilder einlesen (BGR)
- Plotly
  - Bilder anzeigen (RGB)

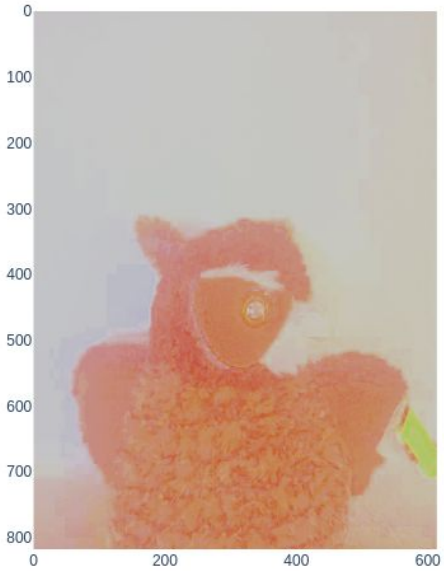
Segmentation

- Vordergrund vom Hintergrund trennen
- Shadow Reduction
- Grayscale
- Threshold Value
- Largest Contour

Segmentation

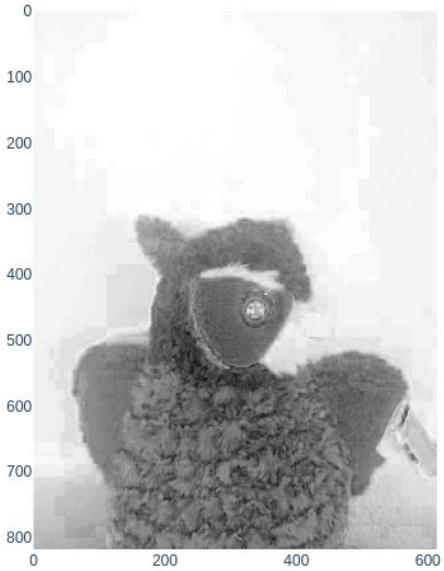
Segmentation

Reduced Shadows



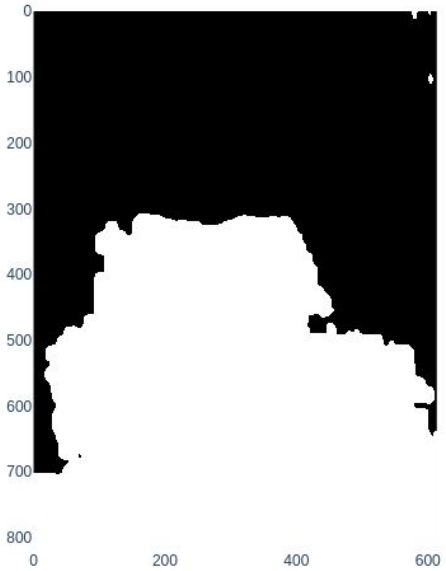
Segmentation

Grayscaled



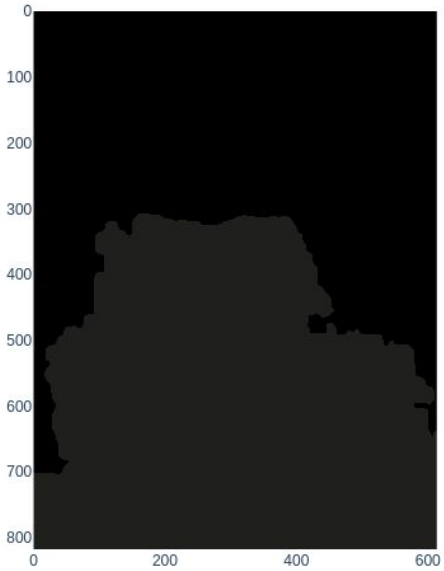
Segmentation

Mask



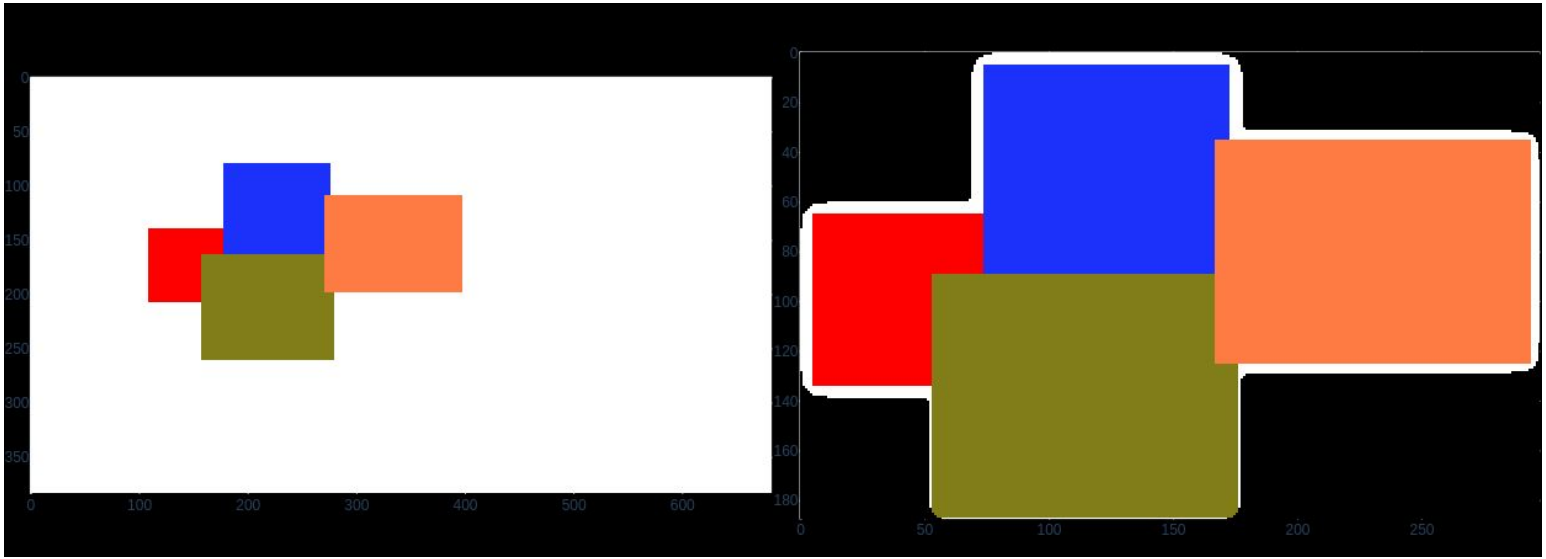
Segmentation

Largest Contour



Segmentation

Segmentation

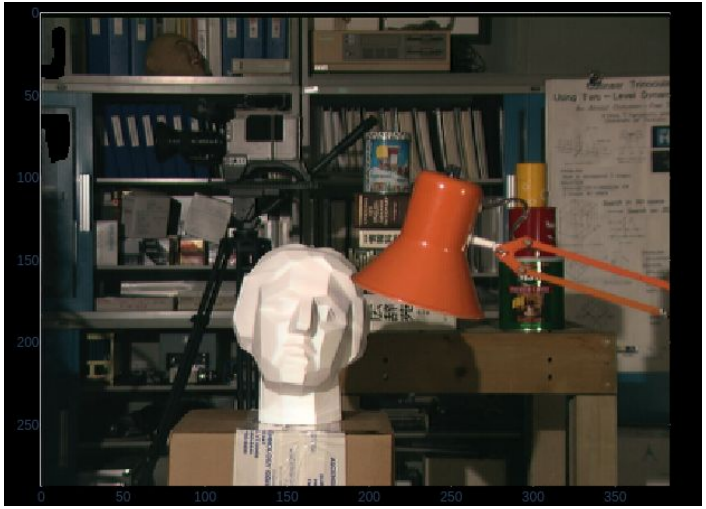


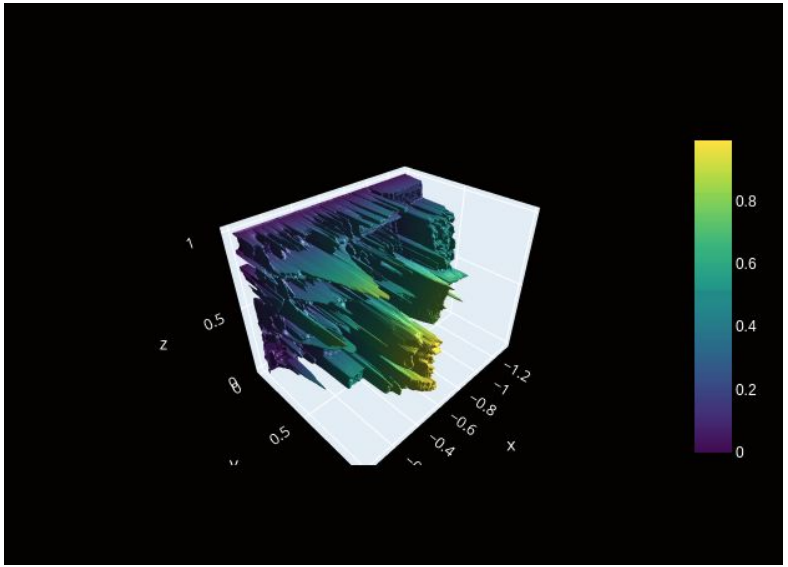
Meshes

- open3d Modul in Python
- Point Clouds
- STL
- Andere Datentypen

Lithophane

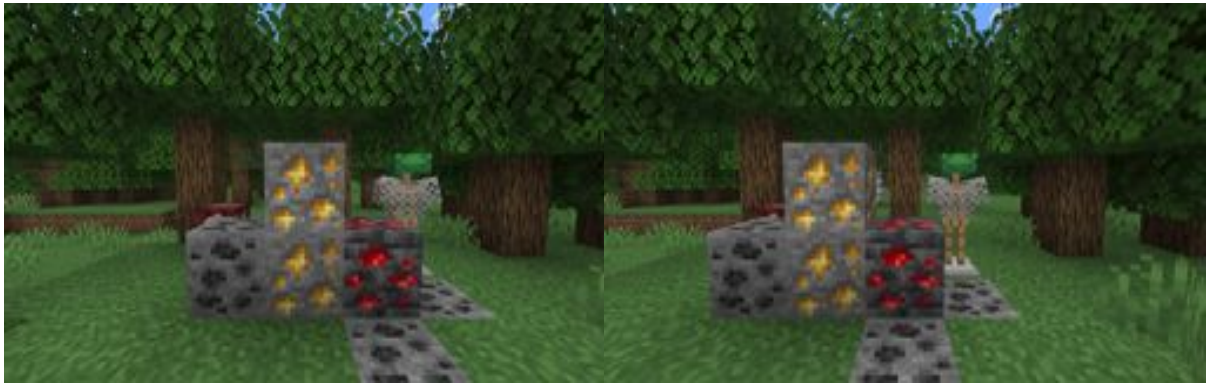
- Licht durch Platte
- Tiefe über Farbwerte





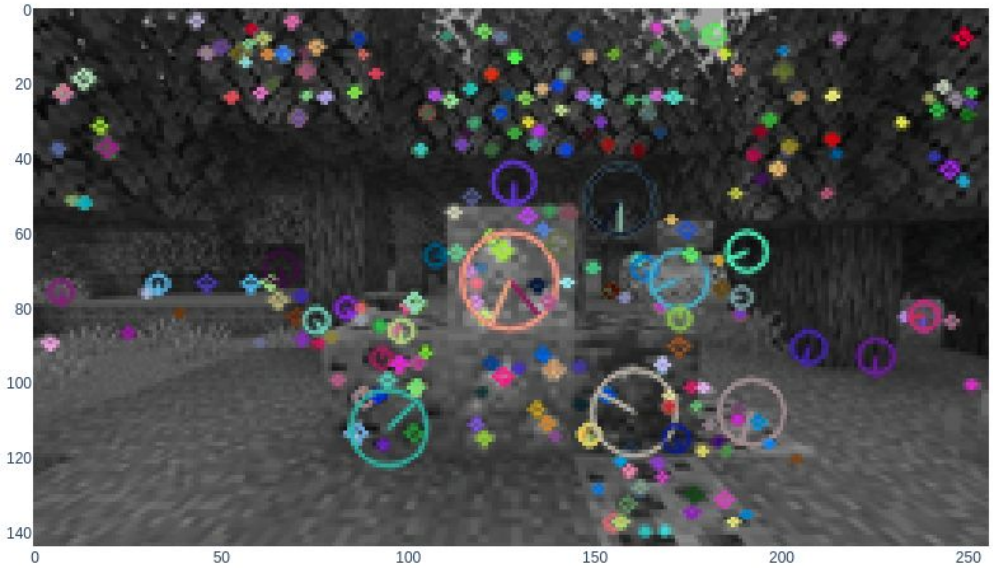
Stereo Bildpaare

- horizontal verschoben (Baseline)
- Rectification, um nicht perfekte Verschiebung zu korrigieren

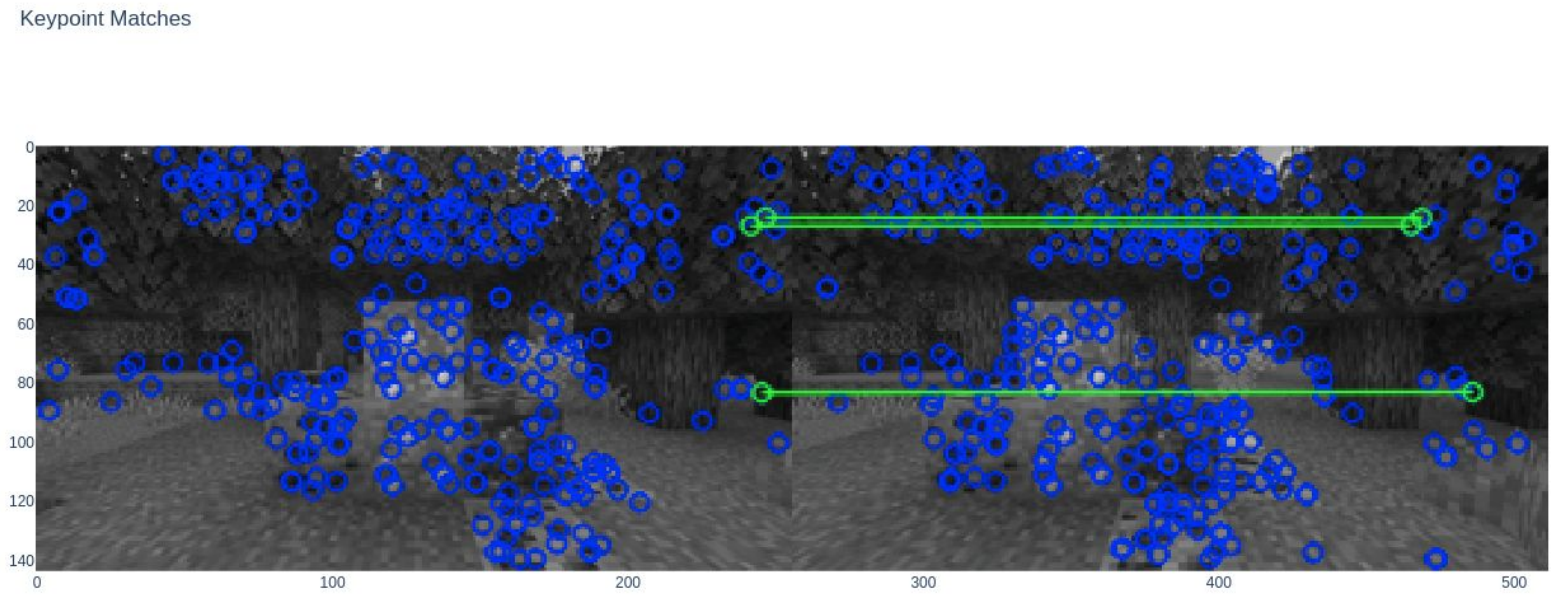


Keypoints

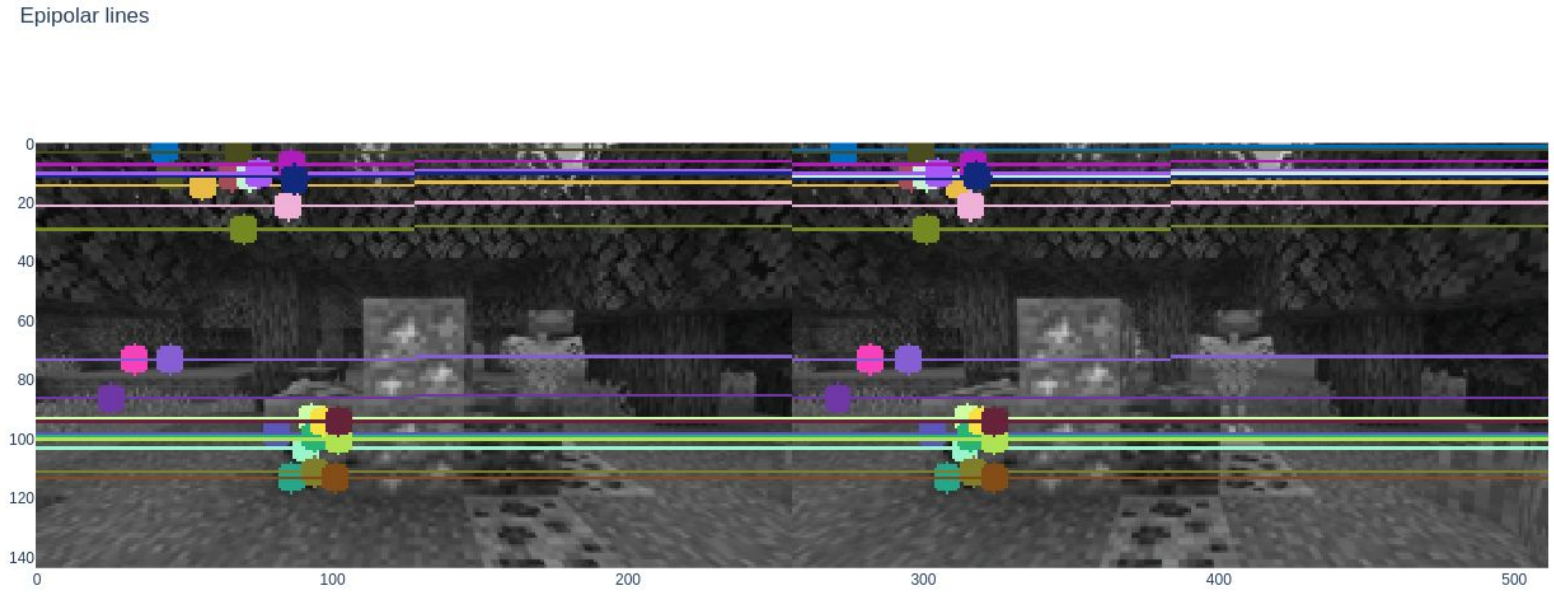
SIFT keypoints



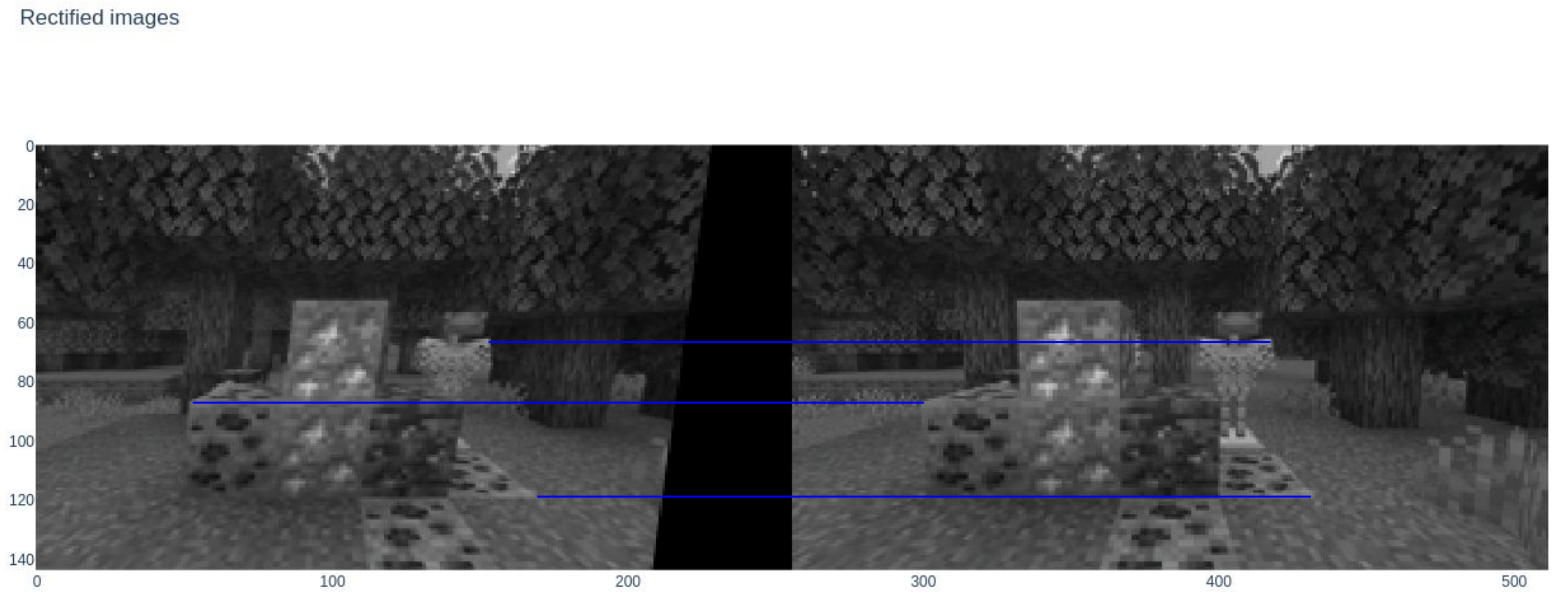
Keypoint Matching



Epipolar Lines

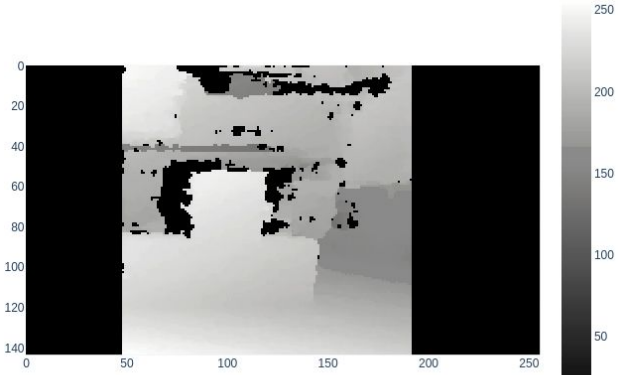


Rectification



Disparity Map

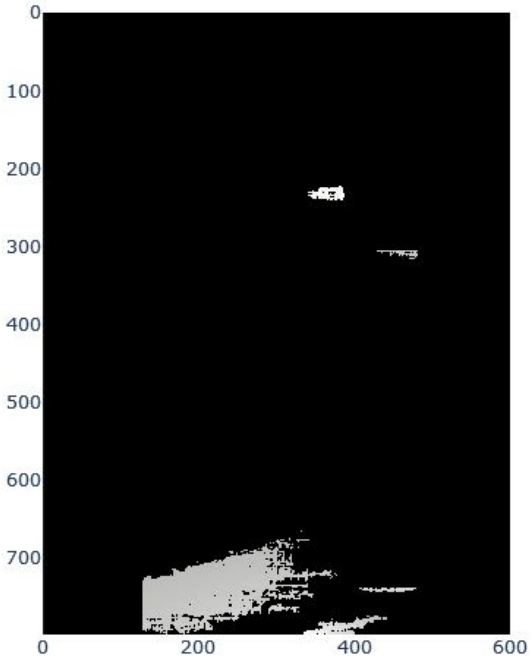
- cv2::StereoSBGM
  - Pixel/Blocks werden zwischen beiden Bildern gematched
  - Verschiebungsvektor zu selbem Pixel/Block im anderen Bild wird berechnet



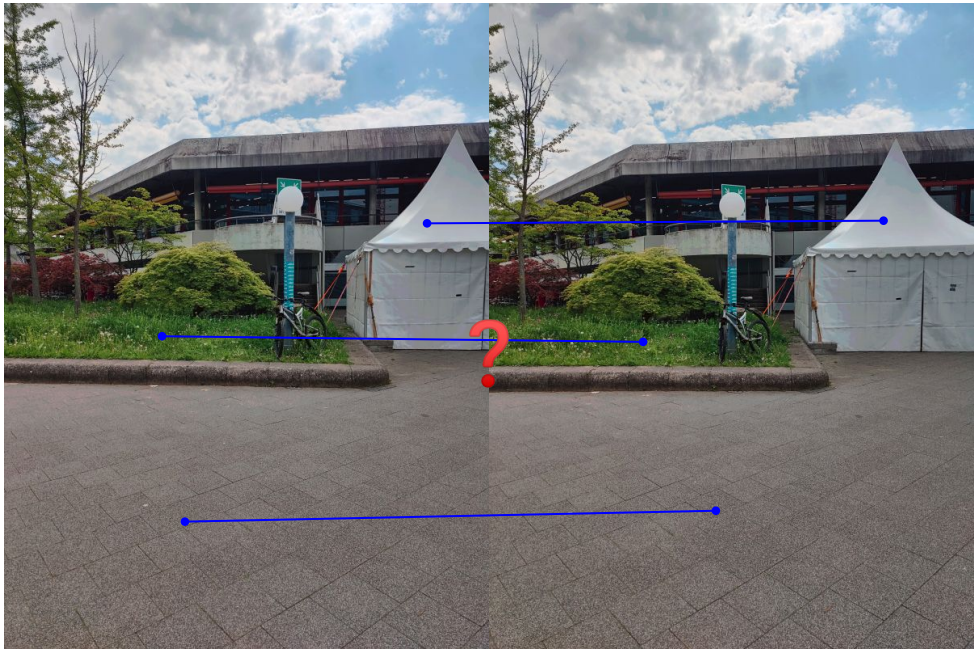
$$Z = f * (x - x') / B$$

Stereo Bildpaar Handykamera

Disparity Map?

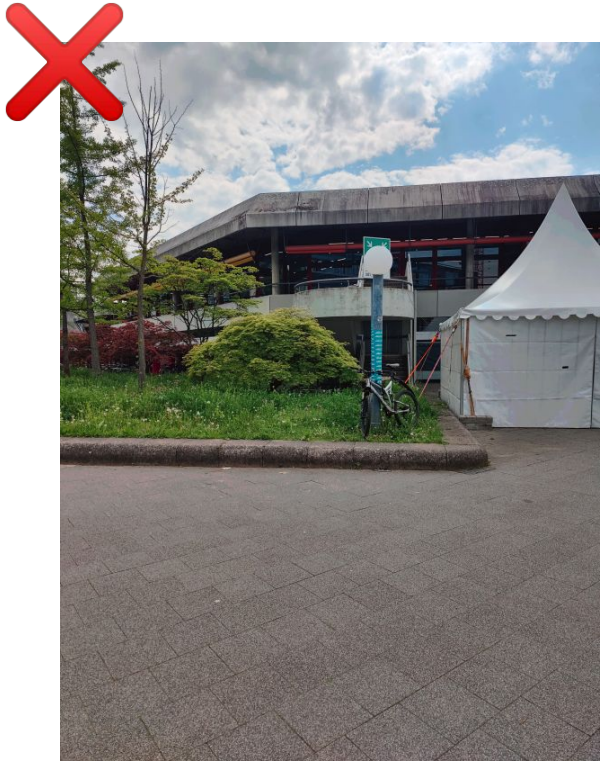


Keypoints?



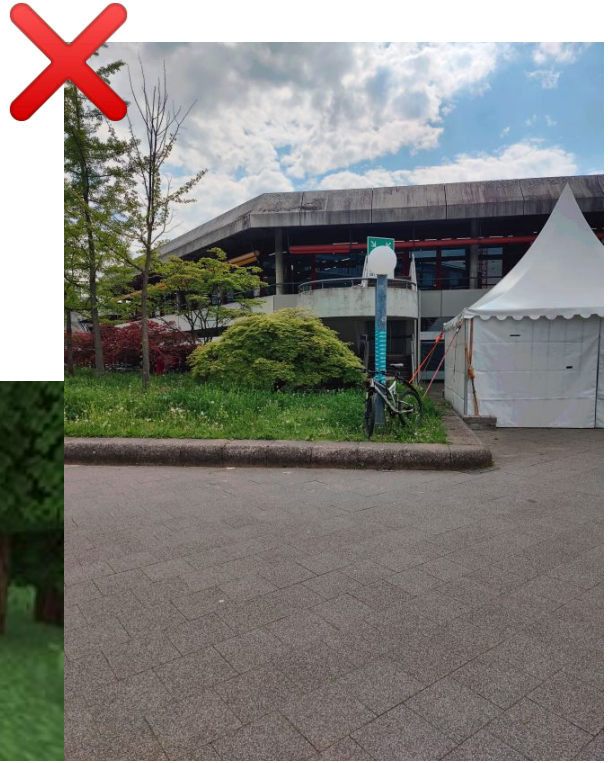
Bildeigenschaften

- keine eindeutigen Muster
  - schwieriges keypoint matching
  - schwierig Pixel/Block Similarity zu vergleichen für Disparity
- keine Tiefeninformation
  - keine Parallaxe



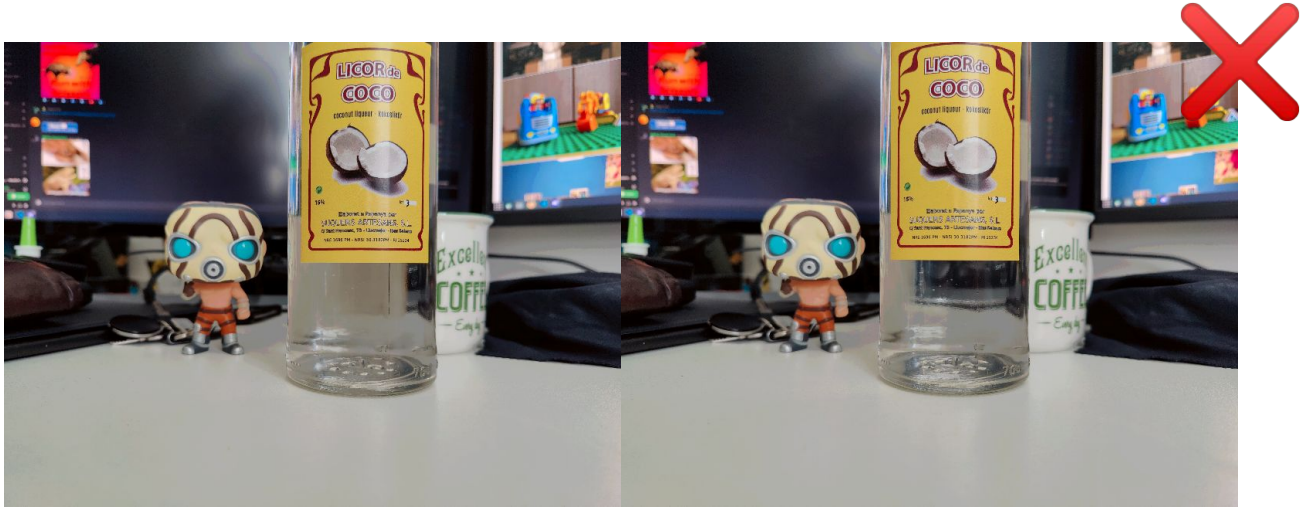
Bildeigenschaften

- keine eindeutigen Muster
  - schwieriges keypoint matching
  - schwierig Pixel/Block Similarity zu vergleichen für Disparity



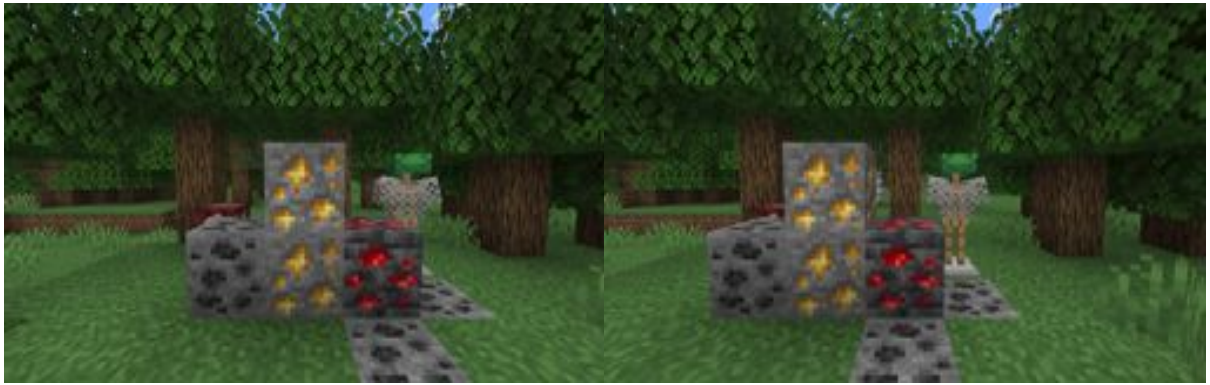
Bildeigenschaften

- Tiefenschärfe/unschärfe



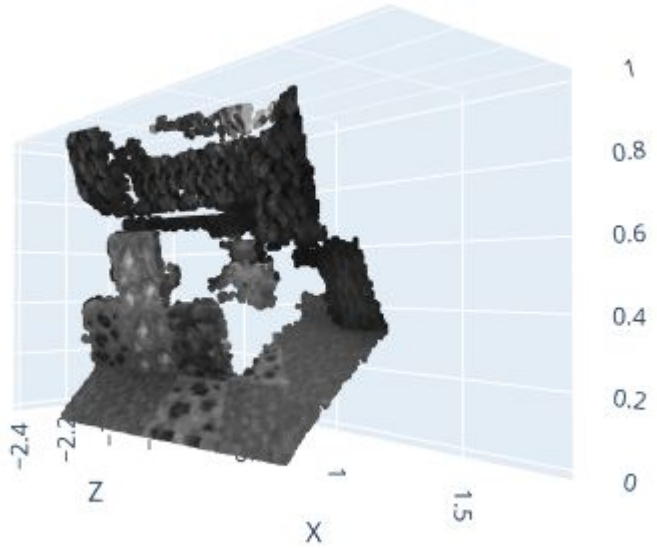
Bildeigenschaften

- Tiefenschärfe
- Muster
- Parallaxe

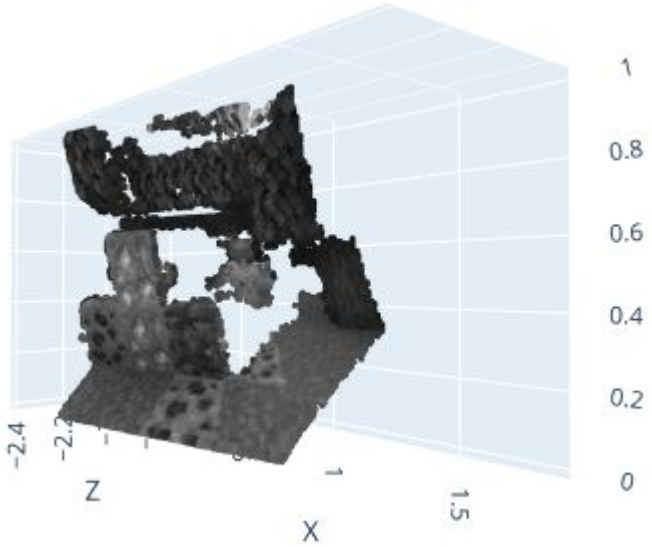


Stereo Point Cloud

- Tiefe aus Disparity Map über $Z = f * (x - x')/B$

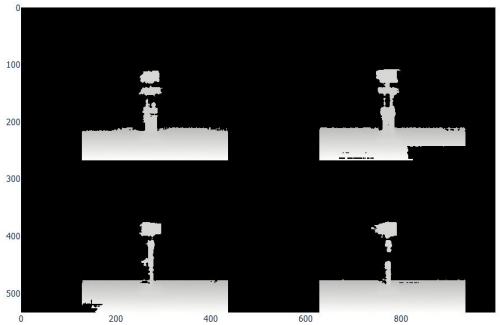


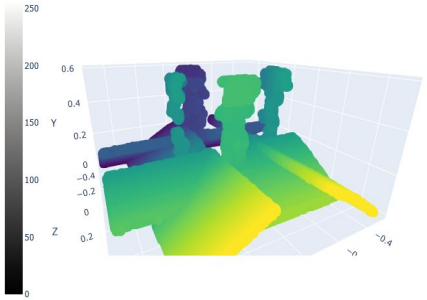
Stereo Point Cloud

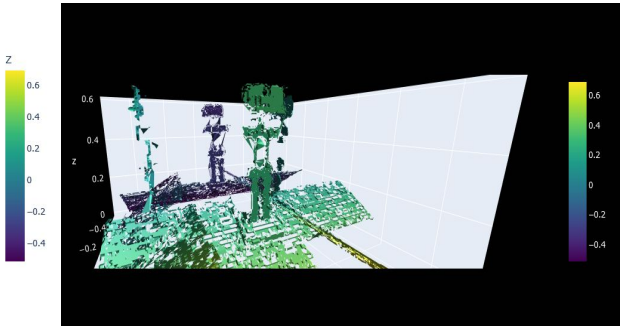


3D Modell / Mesh

- 4 Stereo Image Pairs
  - Vorne, Links, Hinten, Rechts
- Generiere 4 unabhängige Point Clouds
- Problem: Wie fügt man alle 4 Point Clouds in ein zusammenhängendes Modell zusammen?







Ergebnisse

Ergebnisse

https://github.com/PPBP-2021/photogrammetry

Fazit

- Bildverarbeitung in Python generell über OpenCV einfach möglich
  - Fine-Tuning / Anpassung an konkrete Probleme teilweise schwierig
- Stereo Bilder am Handy schwer machbar
  - Tiefenschärfe, Color Correction
- Photogrammetrie ist ein mächtiges Tool
- Einzelne Algorithmen sind leicht anwendbar und verständlich
  - Generalisierung der Algorithmen ist schwer

\+ Gute 3D Point Clouds
\- Rekonstruieren der 3D Szene

Quellen

https://ksimek.github.io/2013/08/13/intrinsic/

https://towardsdatascience.com/3-d-reconstruction-with-vision-ef0f80cbb299

https://docs.opencv.org/4.x/dd/d53/tutorial_py_depthmap.html

https://learnopencv.com/depth-perception-using-stereo-camera-python-c/

https://learnopencv.com/introduction-to-epipolar-geometry-and-stereo-vision/

https://en.wikipedia.org/wiki/HSL_and_HSV

https://en.wikipedia.org/wiki/Binocular_disparity