3D Scanning & Motion Capture

Exercise - 1

Dejan Azinović, Manuel Dahnert

Team

Lecturers

Dr. Justus Thies

Angela Dai

Teaching Assistants

Dejan Azinović

Manuel Dahnert

Tutorials

- Wednesday, 10:15 – 11:45
  - Introduce new exercise
  - Present solution from previous exercise
  - Until start of final projects

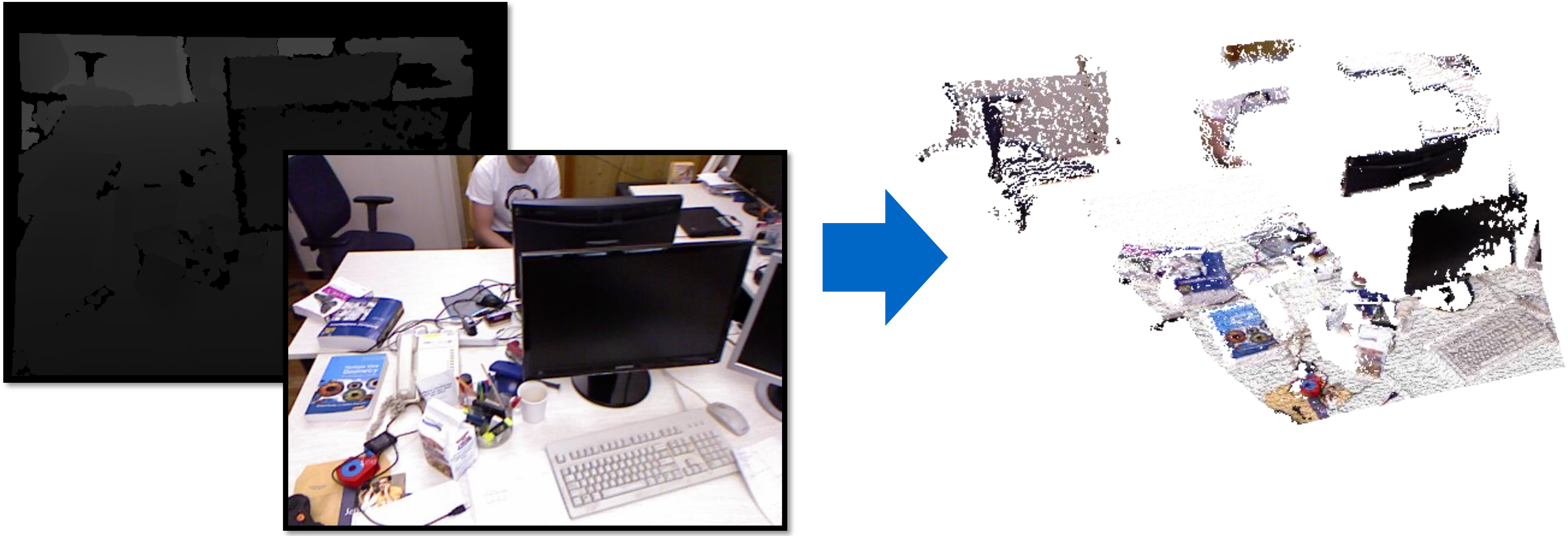
- Office Hours
  - Monday, 14:00 – 15:00
  - Moodle

Exercises – Purpose

- Implement basic 3D reconstruction concepts yourself
  - **Groups of two** are allowed
- Learning by doing, code in C++
- 5 small, self-contained exercises
  - **1 – 2 weeks** of working time
- Grade Bonus (on passed exam) of 0.3
  - Submit all 5 exercises, **pass at least 4 exercises**, **5th exercise passed/borderline**

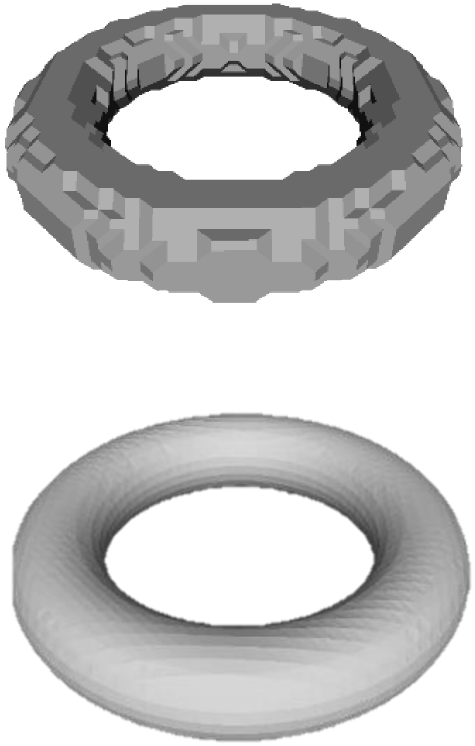
Exercises – Overview (1/5)

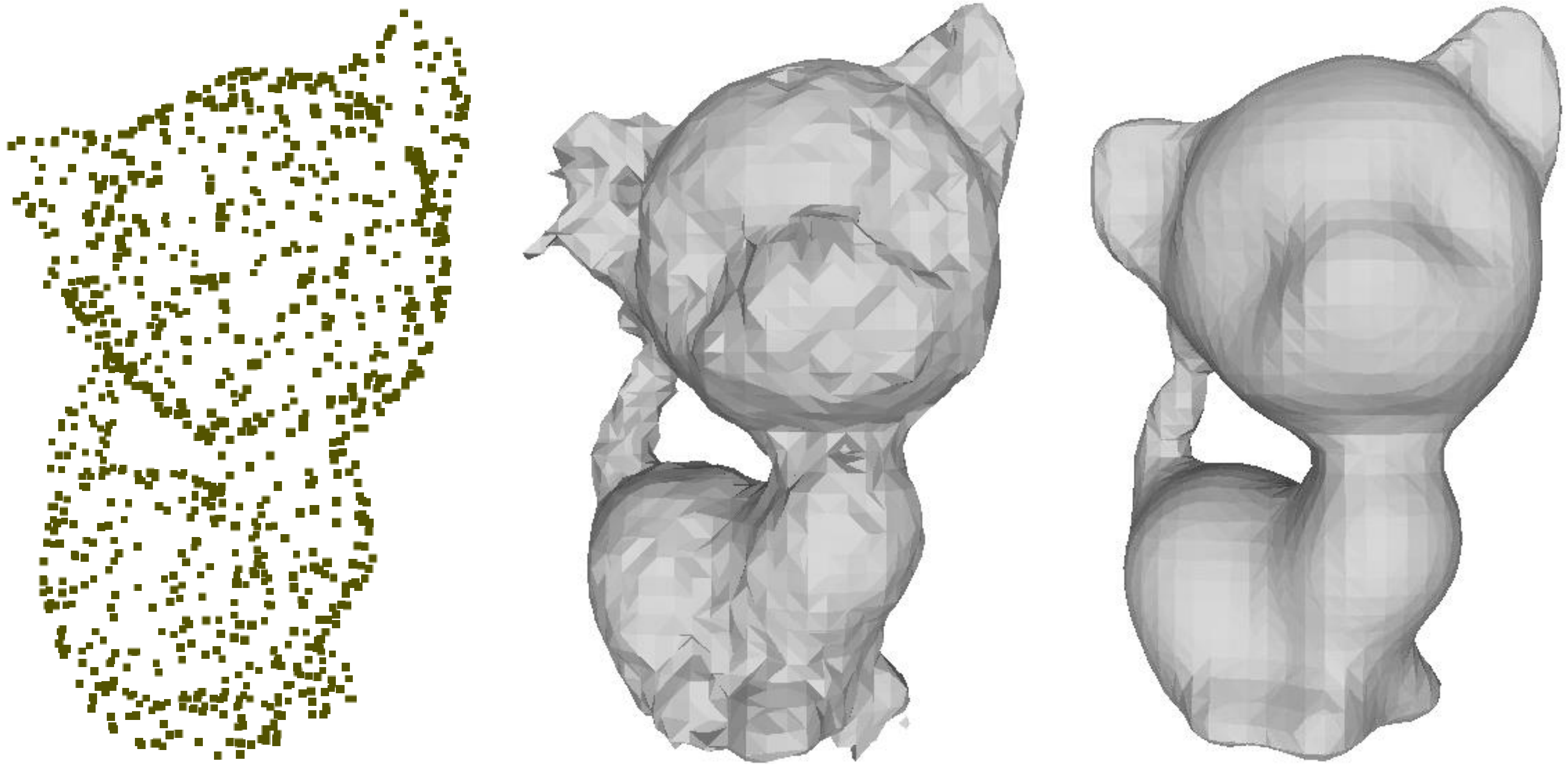
1. Exercise → Camera Intrinsics, Back-projection, Meshes



Exercises – Overview (2/5)

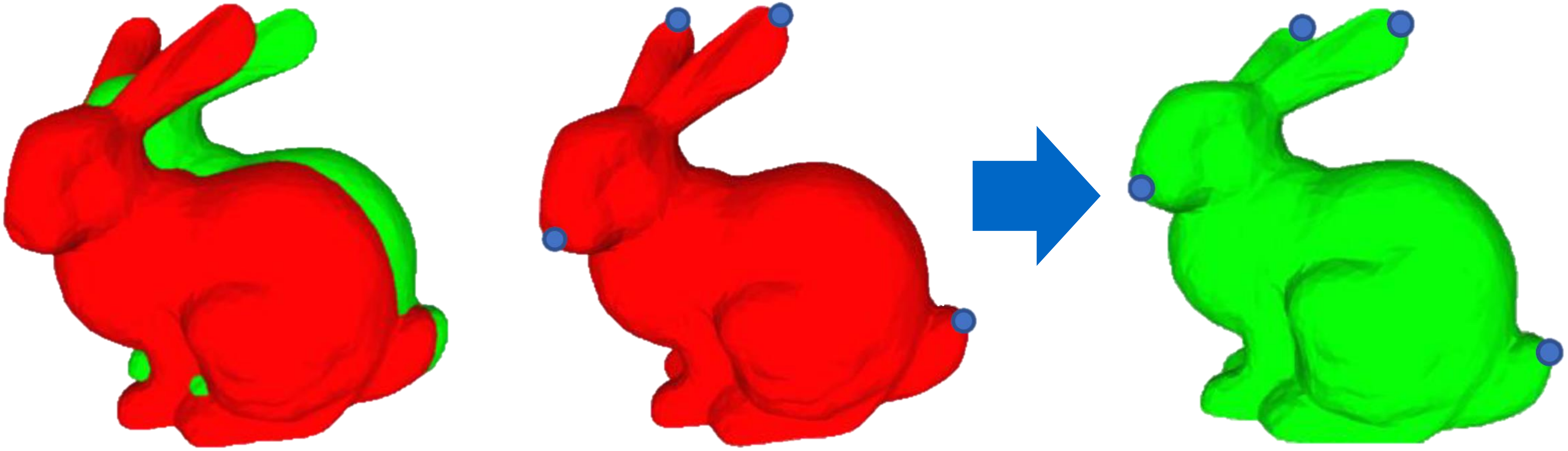
2. Exercise → Surface Representations





Exercises – Overview (3/5)

3. Exercise → Coarse Alignment (Procrustes)

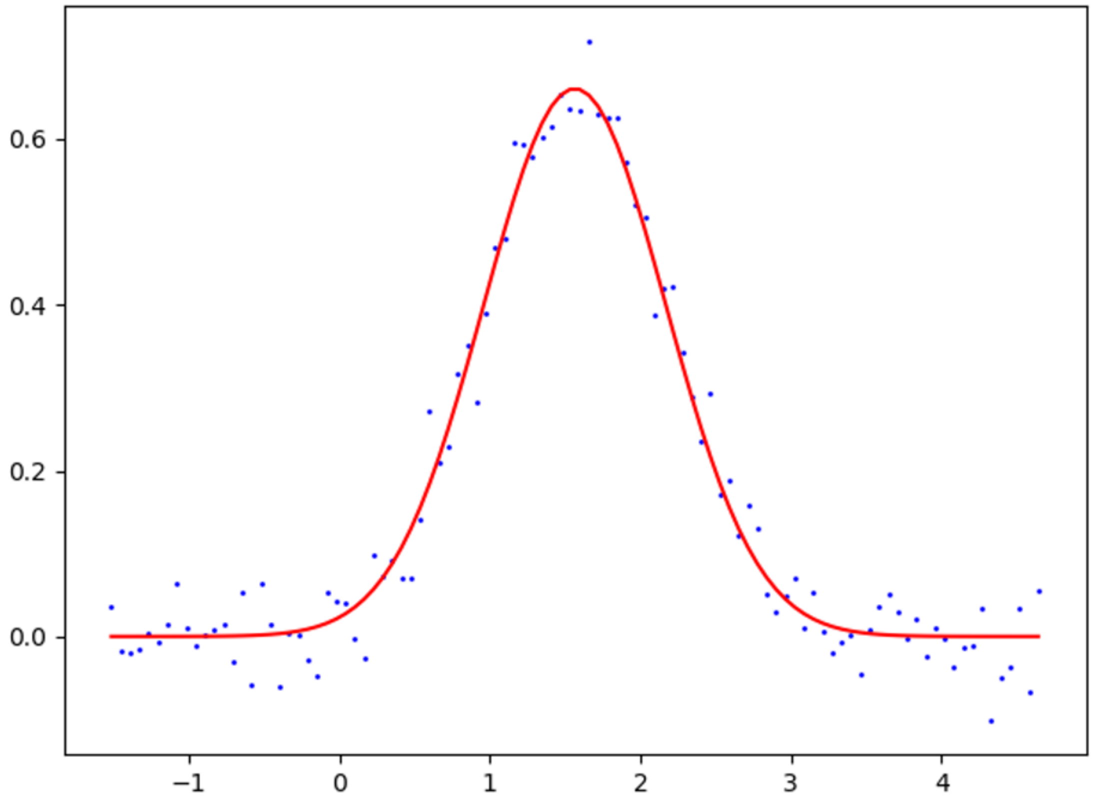


Exercises – Overview (4/5)

4. Exercise → Optimization

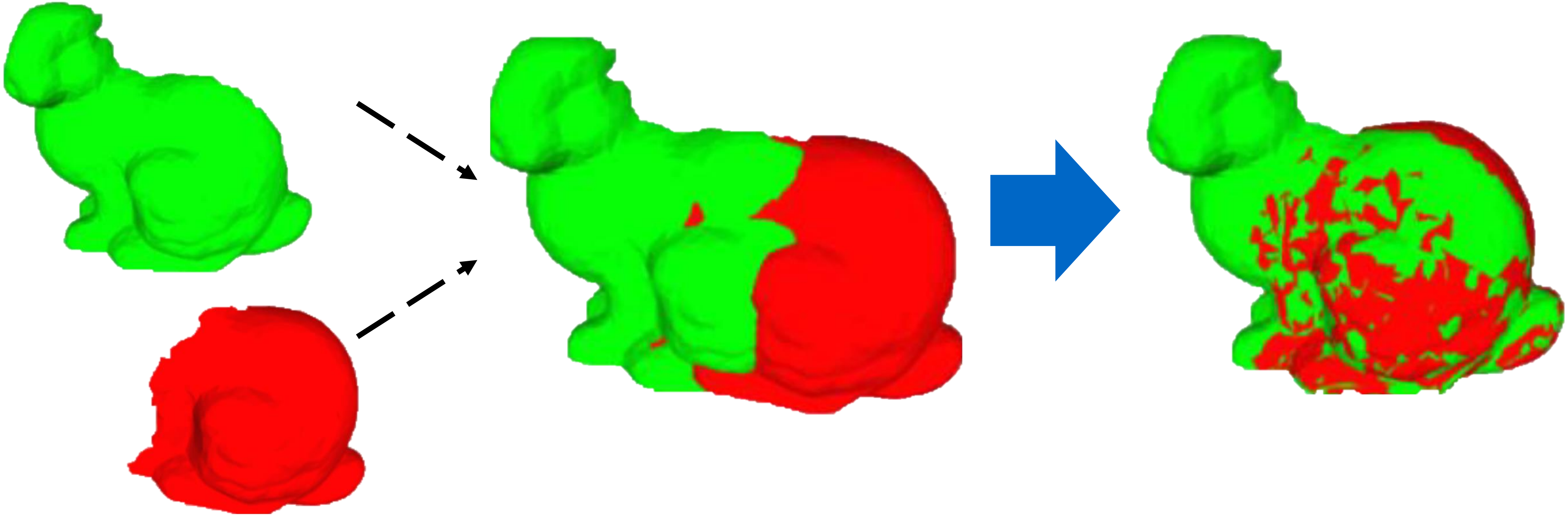
$$f(x) = \frac{1}{\sqrt{2\,\pi\,\sigma^2}}\; e^{-\frac{(x-\mu)^2}{2\,\sigma^2}}$$

Find $\boldsymbol{\mu}$ & $\boldsymbol{\sigma}$ for points



Exercises – Overview (5/5)

5. Exercise → Object Alignment, ICP



Exercises – Overview

1. Exercise → Camera Intrinsics, Back-projection, Meshes
2. Exercise → Surface Representations
3. Exercise → Coarse Alignment (Procrustes)
4. Exercise → Optimization
5. Exercise → Object Alignment, ICP

Final Project

- Start: December, 5+ weeks working time
- 3D reconstruction / tracking project (KinectFusion, Face Fitting, Bundling etc.)
- Groups of 3 - 4
- Proposal (abstract 1-2 pages)
- Presentation of the project (poster) + abstract (2 pages with results)
- 40% of the exam

Exercise 1

Camera Intrinsics – Back-Projection – Meshes

TUM-RGB-D SLAM Dataset

- https://vision.in.tum.de/data/datasets/rgbd-dataset
- 39 sequences
- Recorded using Kinect v.1
  - Structured light
  - Calibrated
  - Aligned depth and color maps
- Camera trajectory

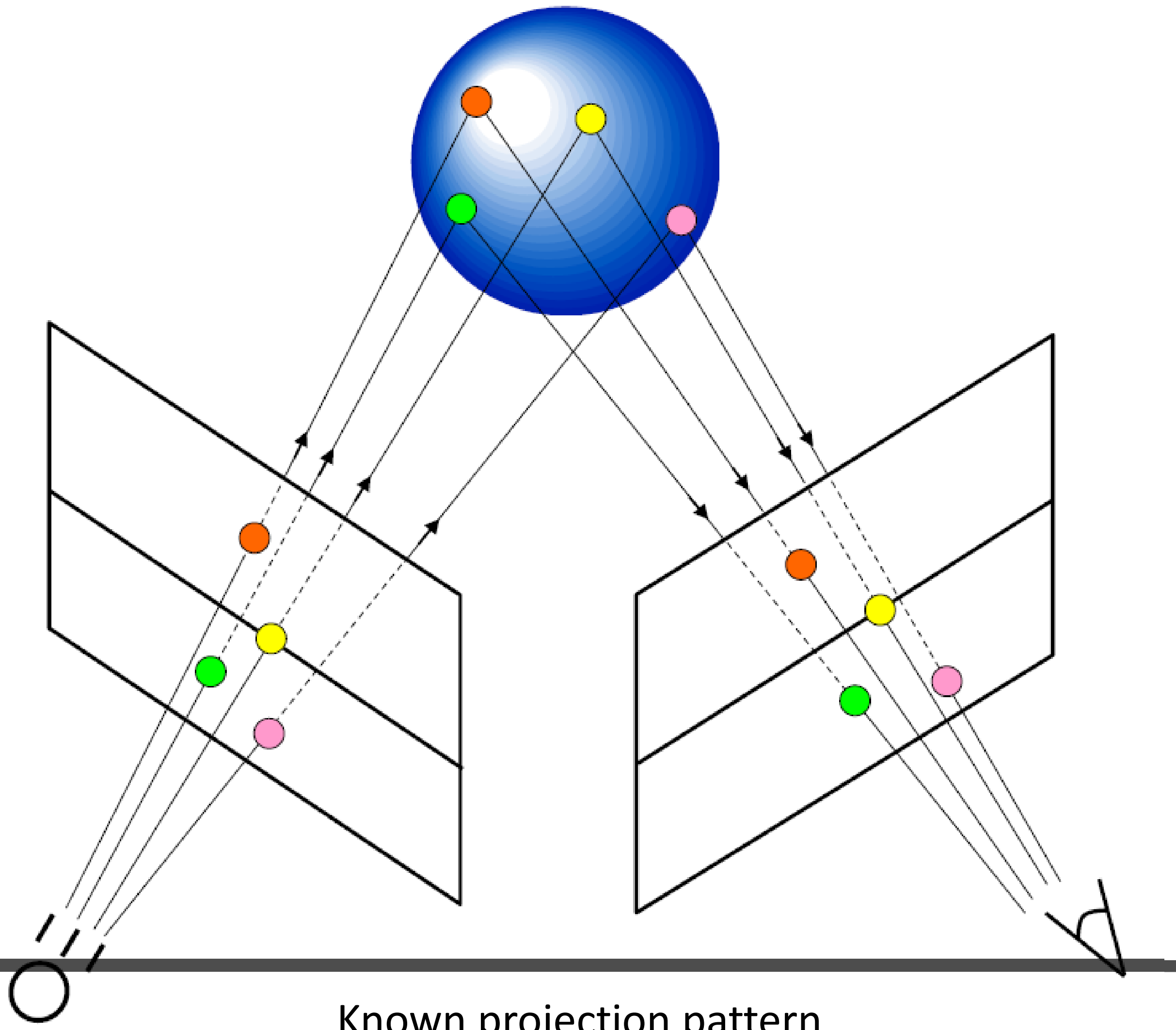



TUM-RGB-D SLAM Dataset

Scene: “fr1/xyz”

Kinect v.1 – Depth and Color Information

Kinect v.1 – Structured Light



Known projection pattern

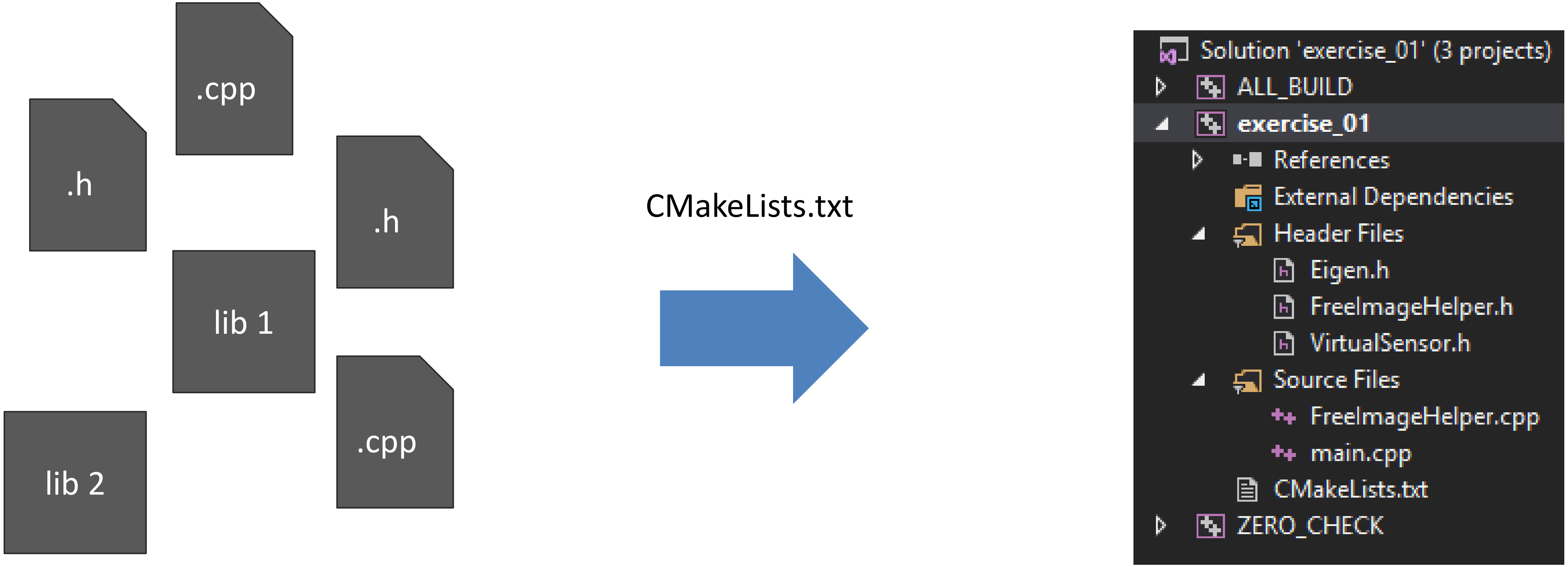
Tasks

1. Project dependencies & CMake configuration
2. Back-Projection
   – Use the given intrinsics, extrinsics and the camera trajectory to project the camera observation back to world space
   – Assign the color to the back-projected points
3. Write a 3D mesh
   – Write an OFF file containing the back-projected position and color information
   – Make use of the grid structure of the observation to perform the triangulation

Task 1) Project dependencies

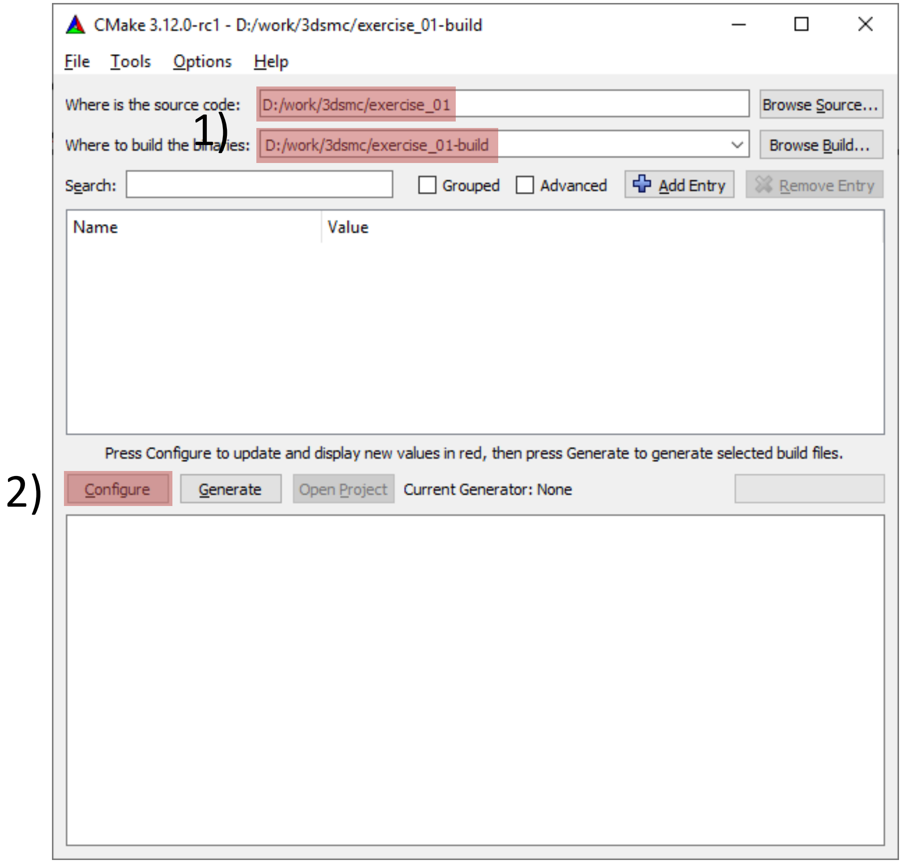
- Eigen   http://eigen.tuxfamily.org
  – Headers-only
  – Linear Algebra library
  – Matrix, Vector, Solvers, …
  – TIP: Do not use C++'s `auto`
- FreeImage   http://freeimage.sourceforge.net/
  – Support for many image formats
  – Windows: We provide a pre-compiled binary
  – Linux: `$ sudo apt-get install libfreeimage3 libfreeimage-dev`

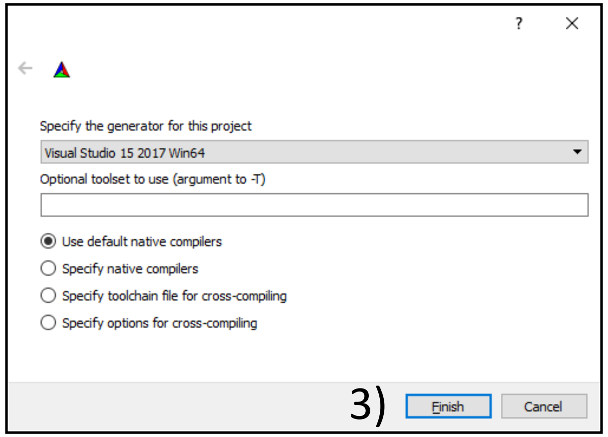
Task 1) CMake

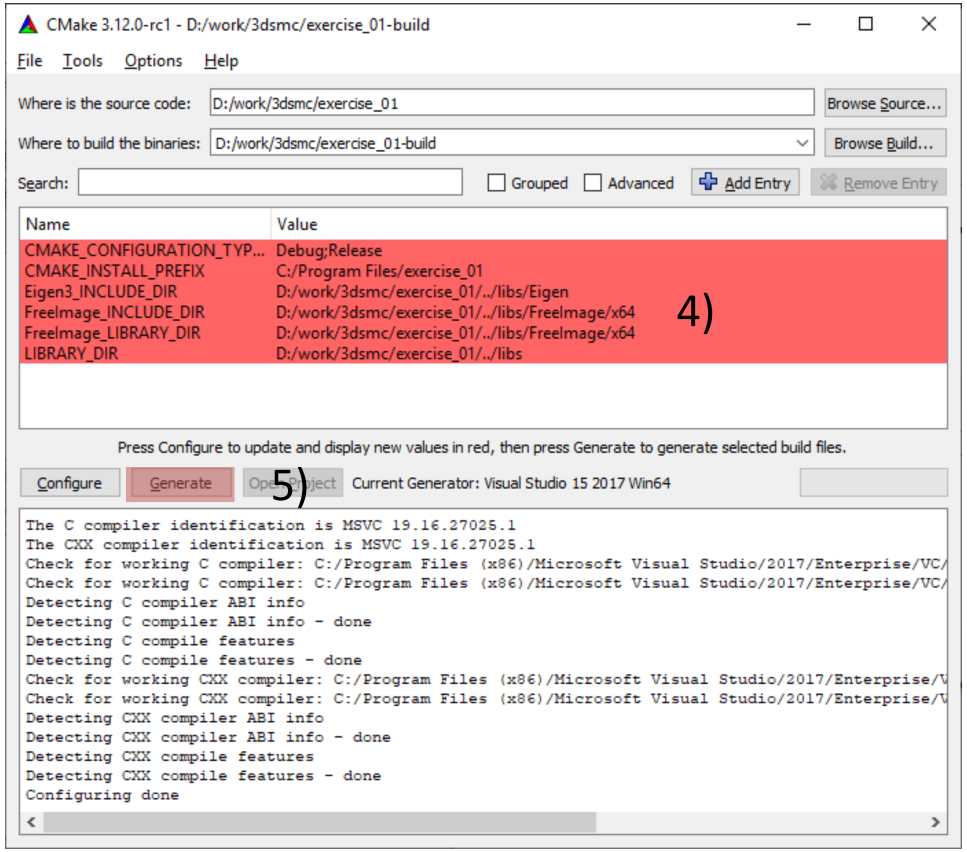


e.g. Visual Studio .sln

Task 1) CMake GUI







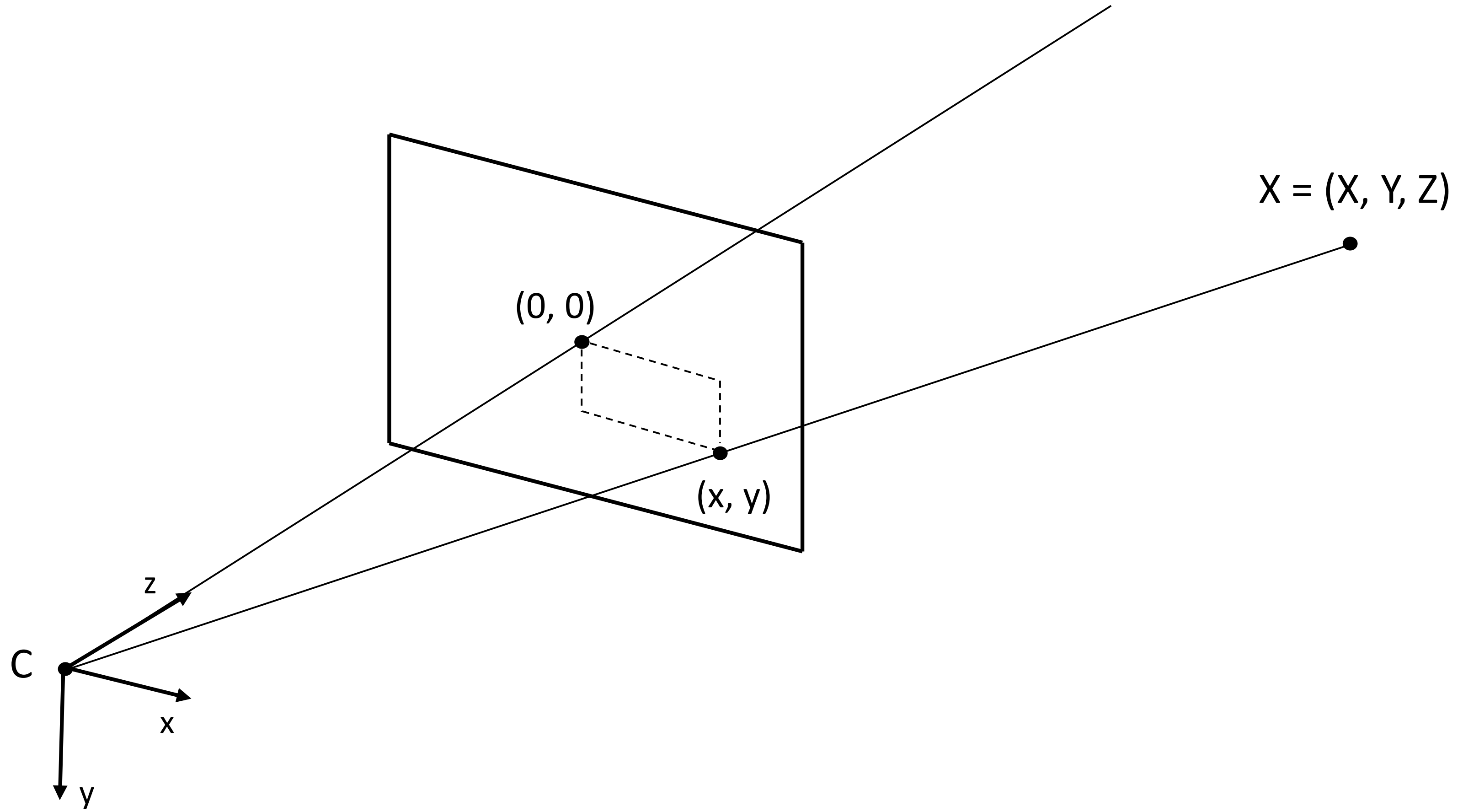
Task 2) Back-Projection

- Use depth map, camera intrinsics and trajectory to project points from 2D → 3D.

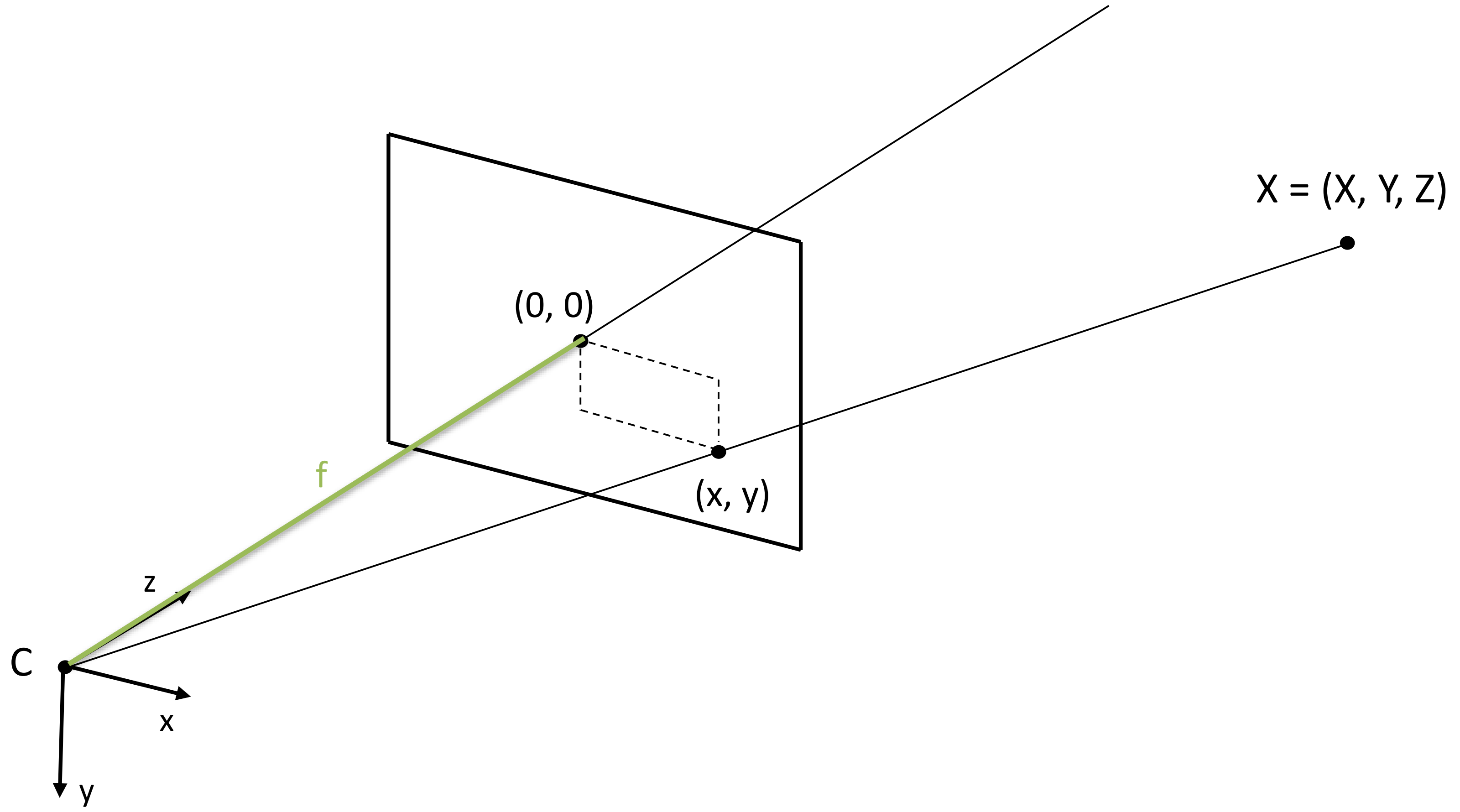
1 float / pixel (z)

4 chars / pixel (R, G, B, A)

Task 2) Pinhole camera model

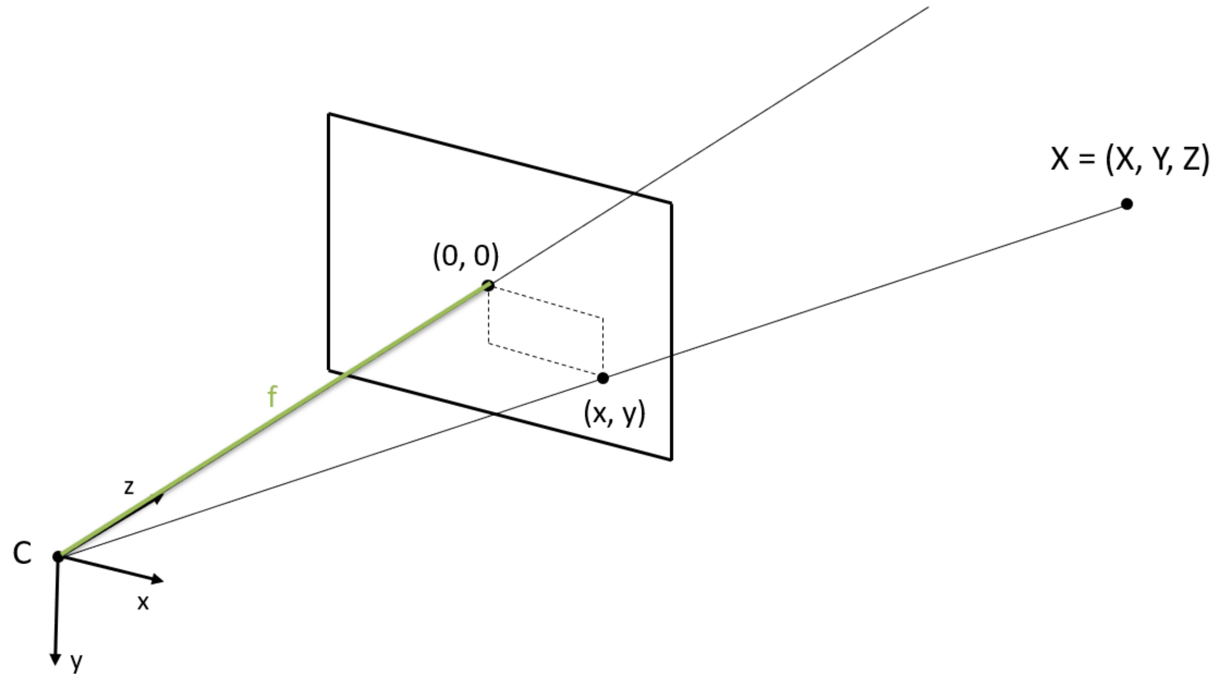


Pinhole camera model

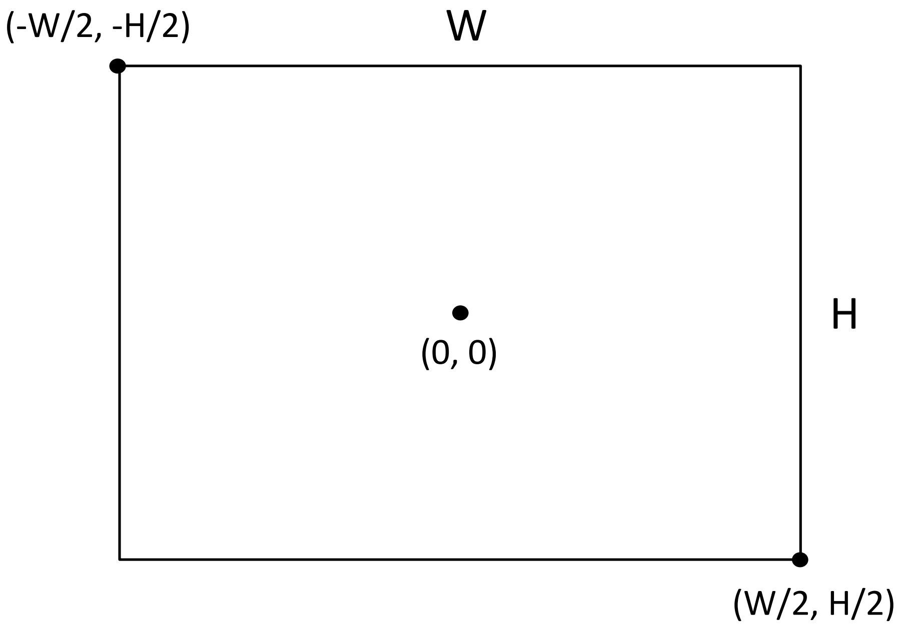


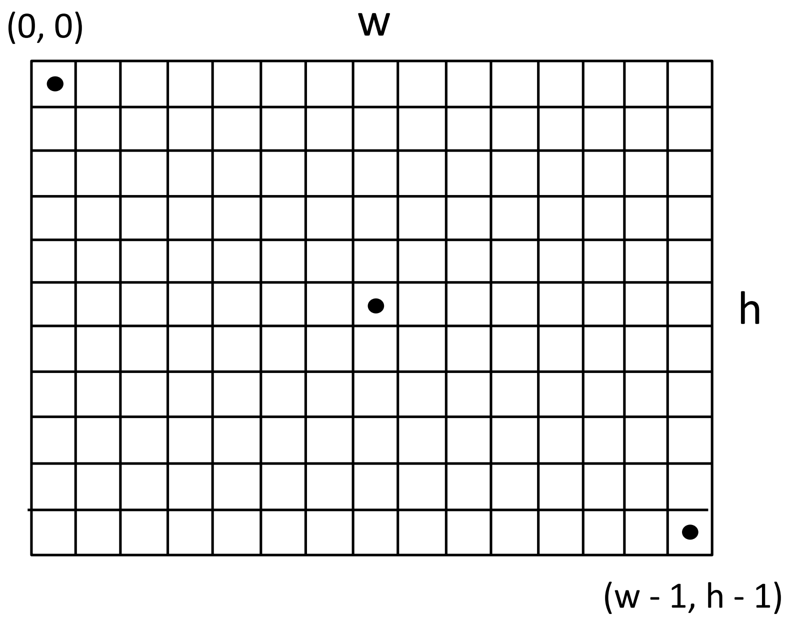
Pinhole camera model

$$\begin{pmatrix} x \\ y \end{pmatrix} = f \cdot \begin{pmatrix} X/Z \\ Y/Z \end{pmatrix}$$

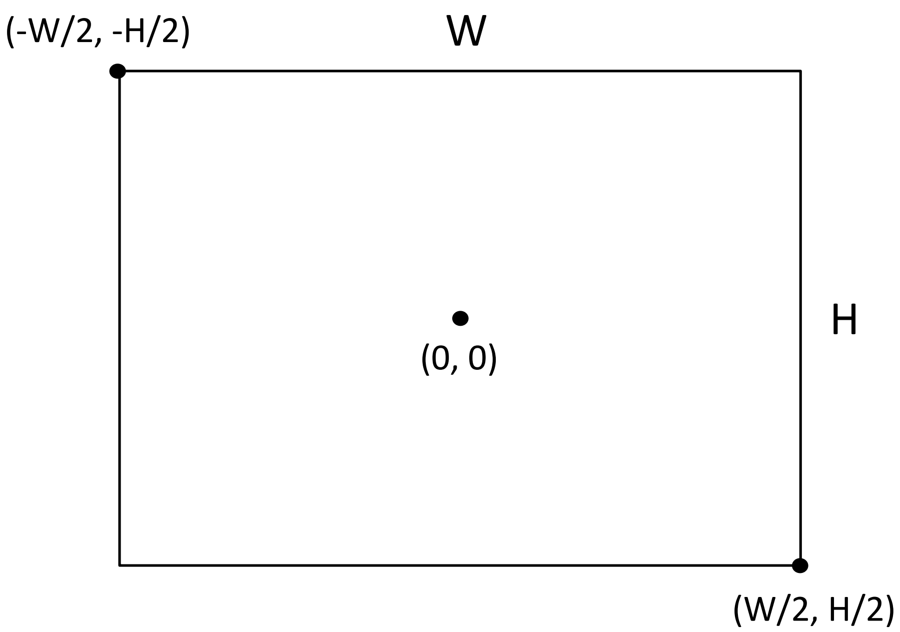


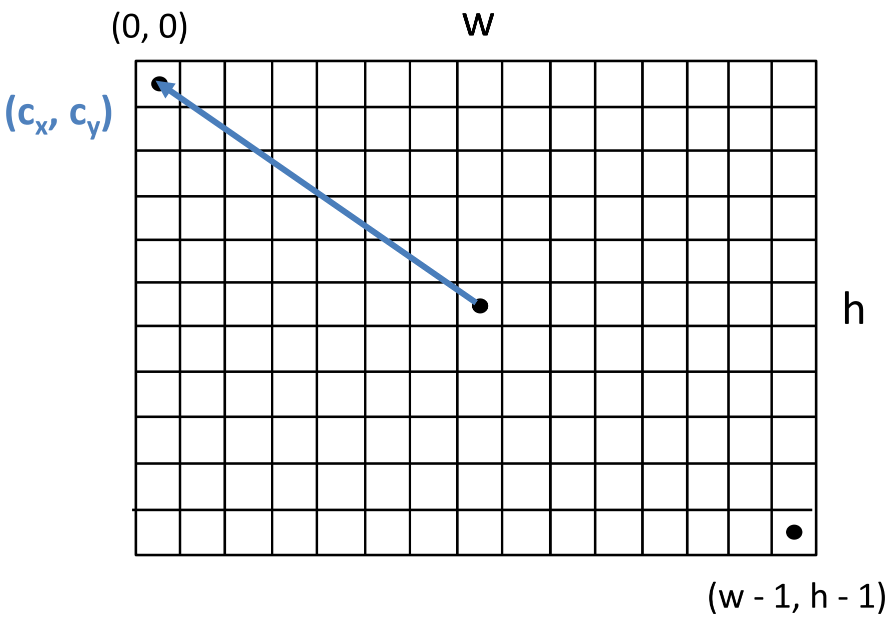
From sensor to pixels





From sensor to pixels





Intrinsic matrix

$$\begin{aligned} f &:= \text{focal length} = 4.1mm \\ W &:= \text{sensor width} = 4.54mm \\ H &:= \text{sensor height} = 3.42mm \\ w &:= \text{image width} = 640 \\ h &:= \text{image width} = 480 \\ c_x &:= \text{image center x} = 320 \\ c_y &:= \text{image center y} = 240 \end{aligned}$$

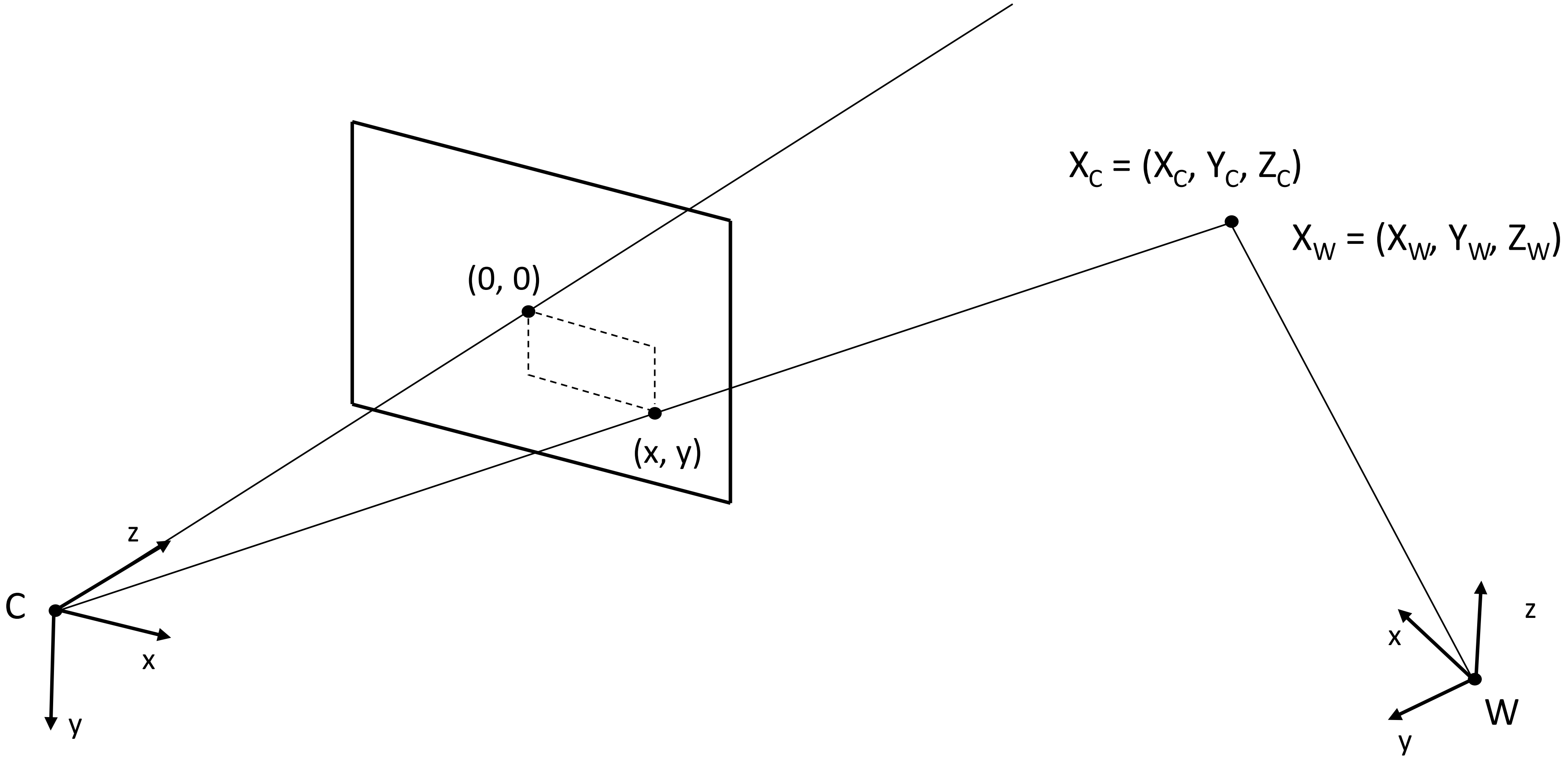
$$\text{Resulting intrisic matrix}: \begin{bmatrix} \frac{f \cdot w}{W} & 0 & c_x \\ 0 & \frac{f \cdot h}{H} & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Perspective Projection in CV

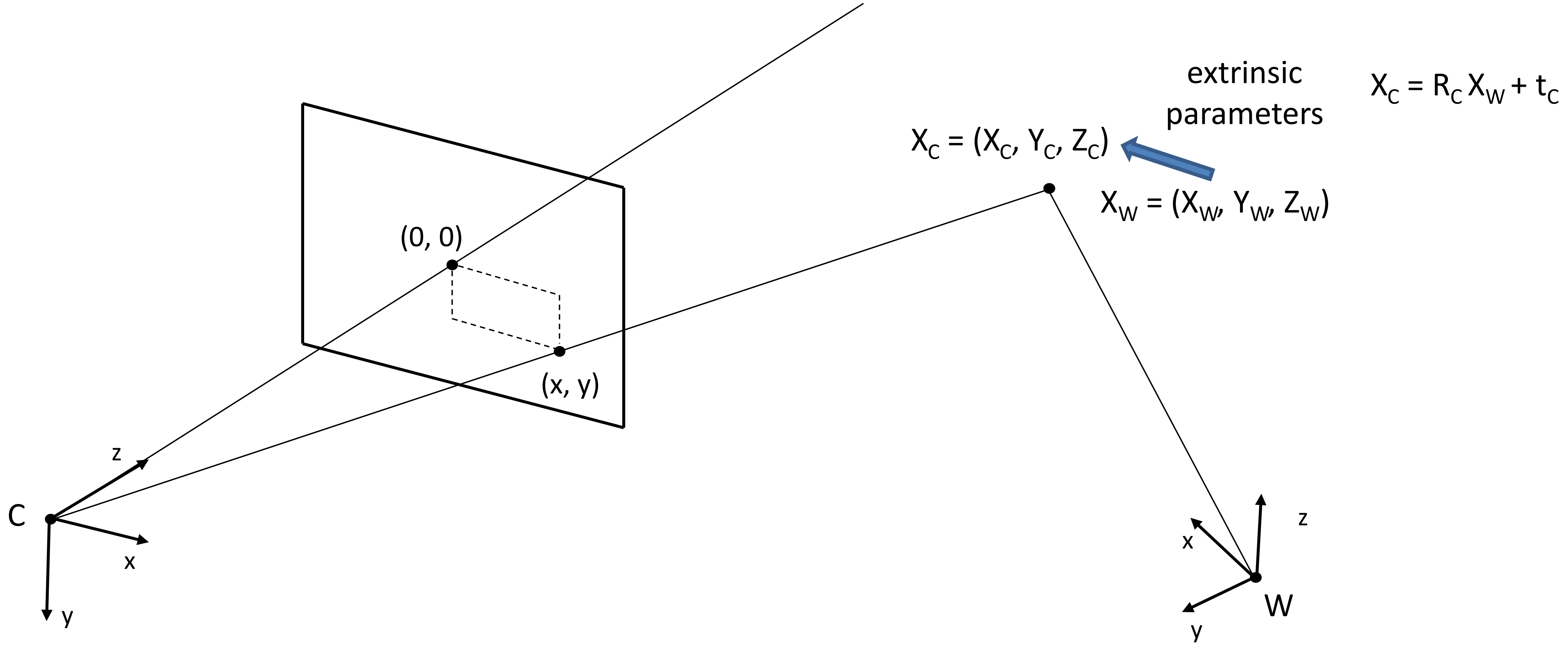
$$\begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ \boldsymbol{z} \end{pmatrix} = \begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} \xrightarrow{\textbf{Dehomogenization}} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} = \begin{pmatrix} u'/w' \\ v'/w' \end{pmatrix}$$

- Keep track of the unmapped z values!
- For backprojection, perform the transformations in reverse order

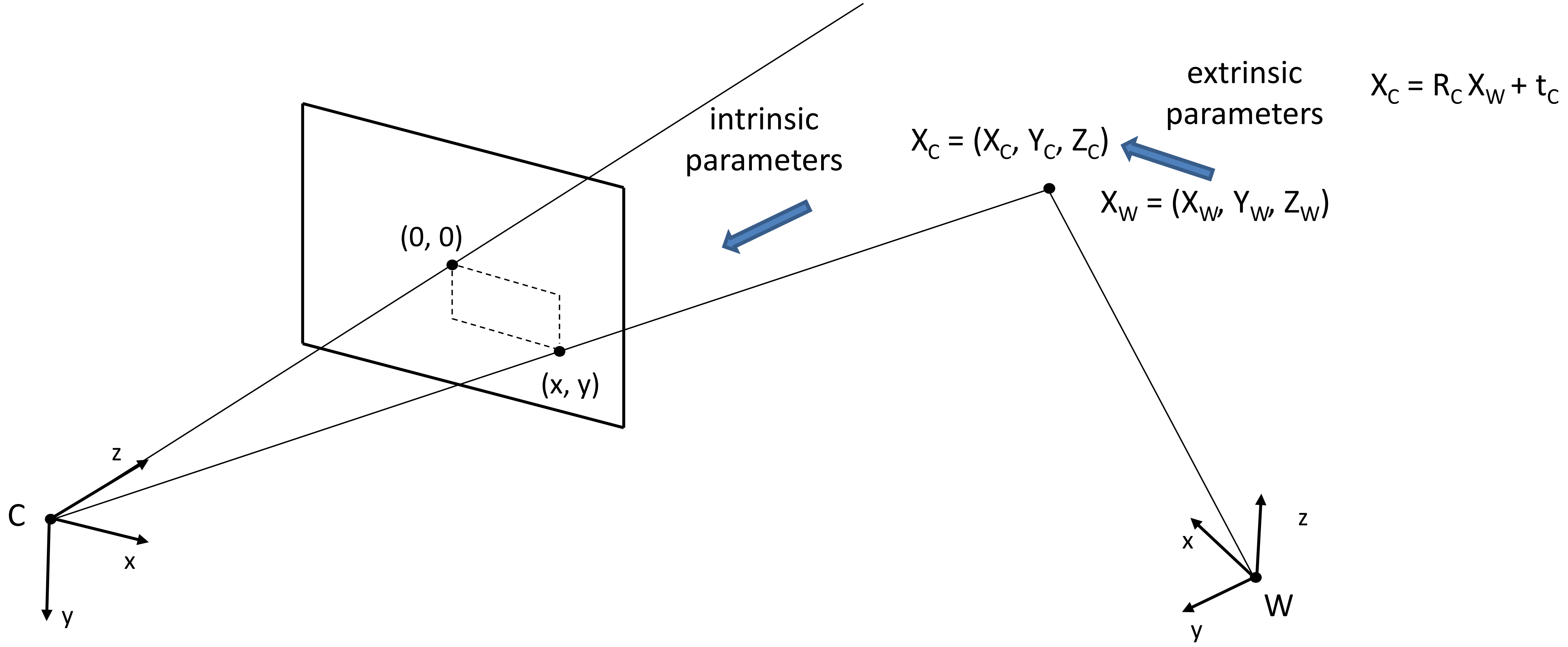
Extrinsic matrix



Extrinsic matrix



Projection Pipeline



Projection Pipeline (from World to Pixels)

$$p_{world} := \text{arbitary point} \in \mathbb{R}^4 = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}^T$$

$$M_{camera} := \text{camera pose} \in \mathbb{R}^{4\times4}$$

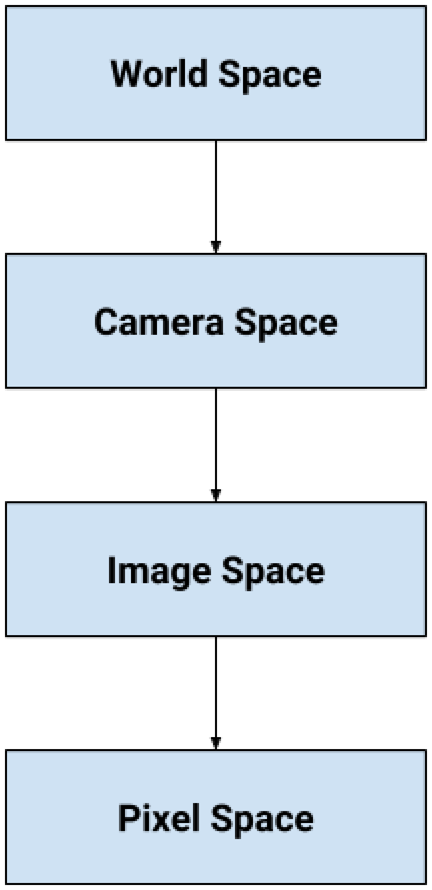
$$K := \text{intrinsic matrix} \in \mathbb{R}^{3\times3} = \begin{bmatrix} \frac{f\cdot w}{W} & 0 & c_x \\ 0 & \frac{f\cdot h}{H} & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$E := \text{extrinsic matrix} \in \mathbb{R}^{4\times4} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

$$p_{camera} = E \cdot M_{camera} \cdot p_{world} \in \mathbb{R}^4$$

$$p_{image} = K \cdot \begin{bmatrix} I_3 & 0 \end{bmatrix} \cdot p_{camera} \in \mathbb{R}^3$$

$$p_{pixel} = \frac{1}{p_{image,z}} \begin{bmatrix} p_{image,x} \\ p_{image,y} \end{bmatrix} \in \mathbb{R}^2$$

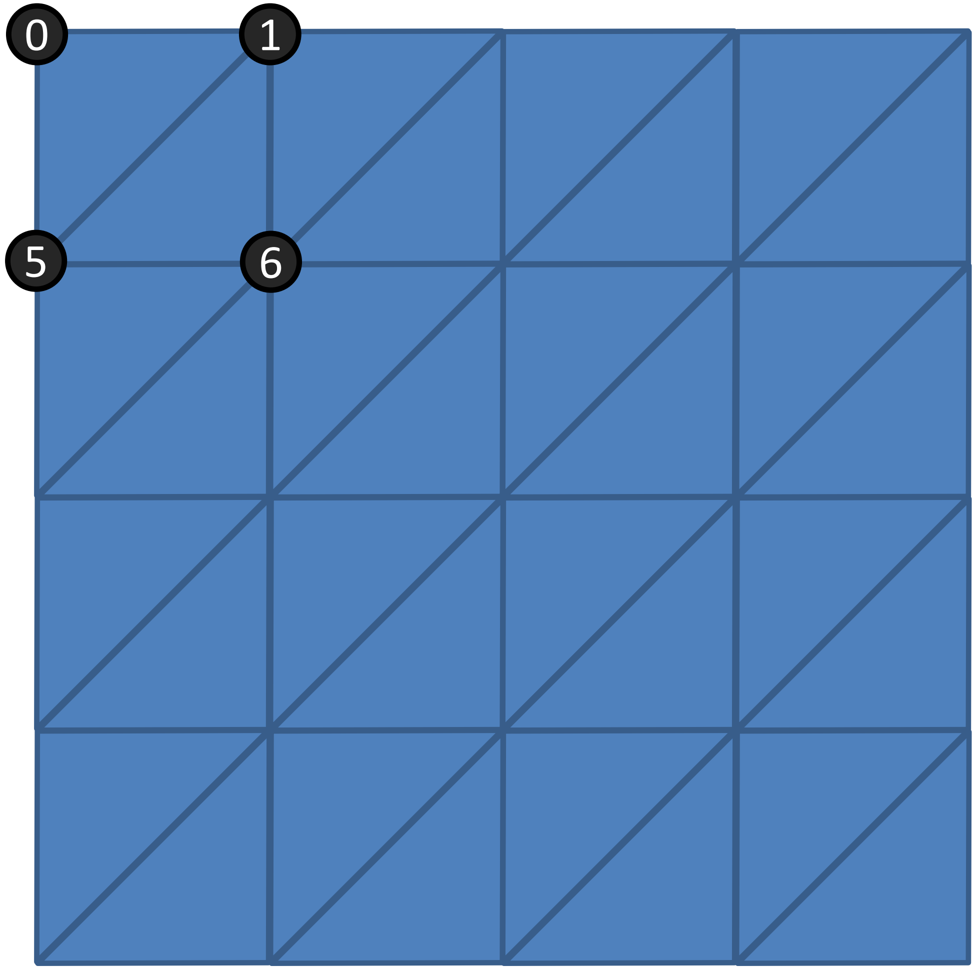


Task 3) Mesh Output

- Write OFF file
  - Simple text-based format
  - Vertices/Points:
    - Position
    - Color
  - Faces
    - Indices of vertices

```
COFF
# numVertices numFaces numEdges
4 2 0
# list of vertices
# X Y Z R G B A
0.0 1.0 0.0 255 255 255 255
0.0 0.0 0.0 255 255 255 255
1.0 0.0 0.0 255 255 255 255
1.0 1.0 0.0 255 255 255 255
# list of faces
# nVerticesPerFace idx0 idx1 idx2 ...
3 0 1 2
3 0 2 3
```

Task 3) Mesh Structure



Ensure consistent orientation of the triangles! (Usually counter-clockwise)

Example:

First triangle:	0-5-1
Second triangle:	5-6-1

Visual Studio 2017 Community

- https://www.visualstudio.com/de/downloads/

- Known issues:
  - fatal error LNK1104: cannot open file ‘gdi32.lib‘
    - https://stackoverflow.com/questions/33599723/fatal-error-lnk1104-cannot-open-file-gdi32-lib