Module: 3D Vision

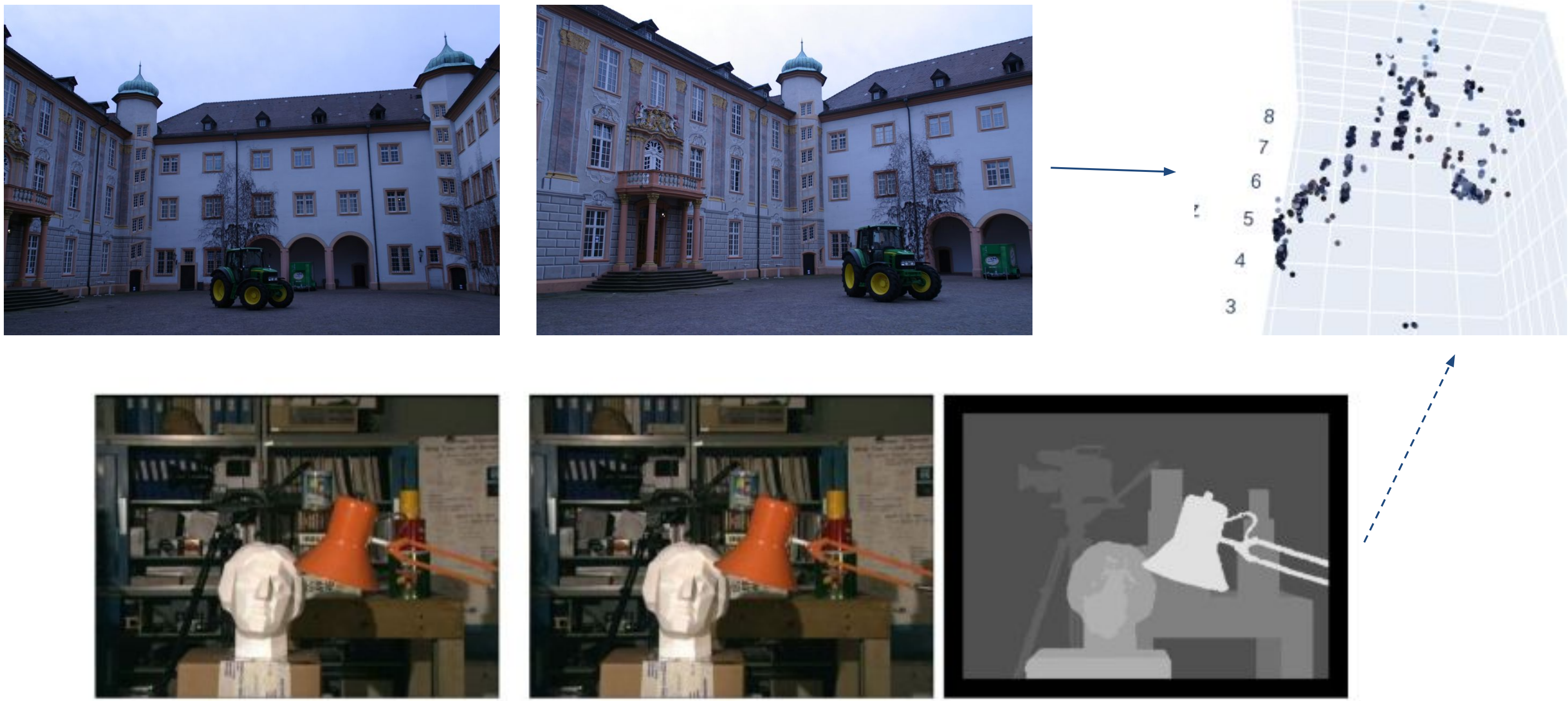
Project: 3D recovery of urban scenes (Session 4)

Original Lab: Pedro Cavestany

Modifications: Daniel Ordoñez and Marc Perez (marc.perez.quintana@upc.edu)

Goal

3D Reconstruction from two images with known internal parameters



Mandatory Tasks

- 1. Triangulation with the DLT method **(2.0)**
- 2. Reconstruction from two views:
  - 2.1 Estimate the image matches **(0.5)**
  - 2.2 Estimate the Fundamental Matrix **(0.5)**
  - 2.3 Estimate the Essential Matrix **(1.0)**
  - 2.4 Estimate the Camera Matrices from the Essential Matrix **(1.0)**
  - 2.5 3D Visualization **(0)**
  - 2.6 Reprojection Error **(0.5)**
- 3. Depth map computation using local methods
  - 3.1 Vectorization (hint) **(0)**
  - 3.2 Disparity between a pair of rectified images **(1.5)**
  - 3.3 Sum of Squared Differences Cost **(0.5)**
  - 3.4 Normalized Cross Correlation Cost **(0.5)**
  - 3.5 Apply to facade images **(0.5)**
  - 3.6 Adaptive support weights **(1.5)**

Optional Tasks

- Optional 1: Depth map computation with plane sweeping **(+1.5)**
- Optional 2: New view synthesis using view morphing **(+1.5)**
- Optional 3: Depth map fusion review **(+2.0)**
- Optional 4: Depth map to 3D point cloud **(+1.0)**

- Organized notebook/report and good programming practices and mathematical notation **(+1.0)**

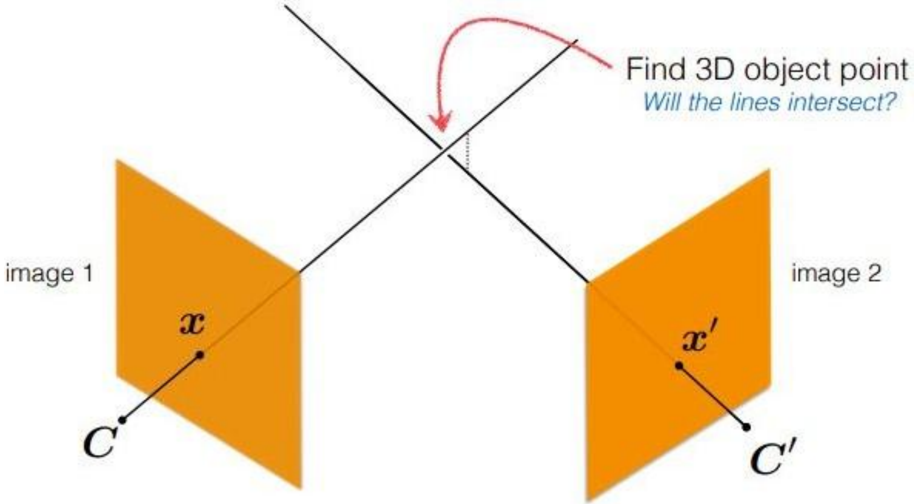
Assignment

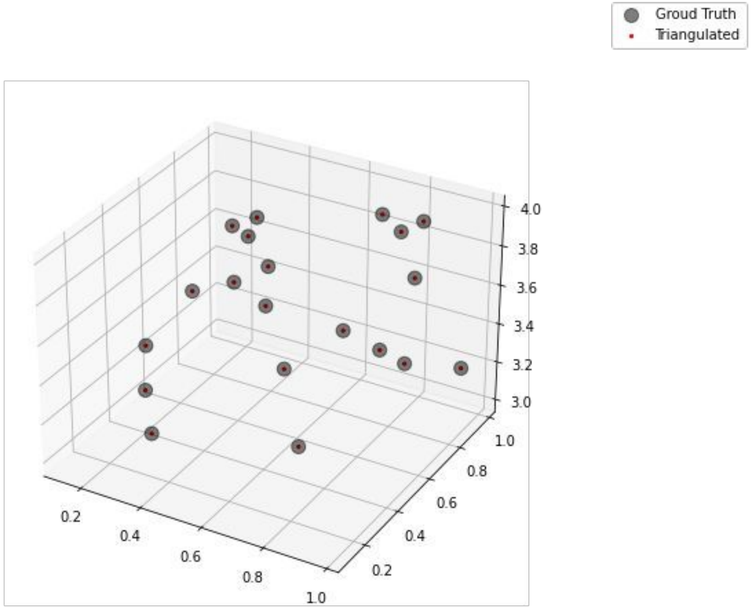
- Code is provided in python in a jupyter notebook.
- Auxiliary functions and algorithms are provided on additional modules.
- Deliver before 11AM of next wednesday, February 1st.

Deliverables

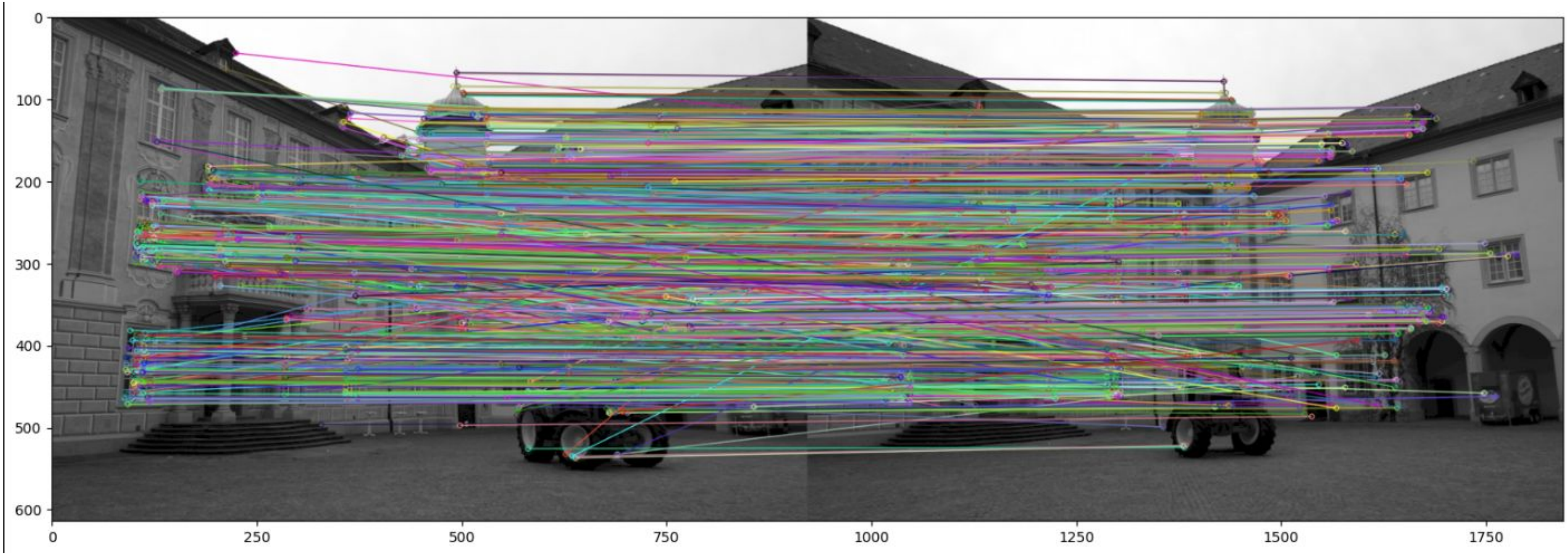
- **Jupyter notebook:** ready to run.
    - Document your code and decisions on markdown.
    - Be clear of what information is assumed/required for each algorithm/operation.
    - Understand the equations do not just reproduce them from the slides.
- **Report:**
    - Short report.
    - In depth analysis.
    - Do not paste code in report. I am interested in analysis and justification.
    - Problems and comments.
    - You can use the notebook as a report **IF, AND ONLY IF,** you format the notebook appropriately.

1. Triangulation with the DLT method

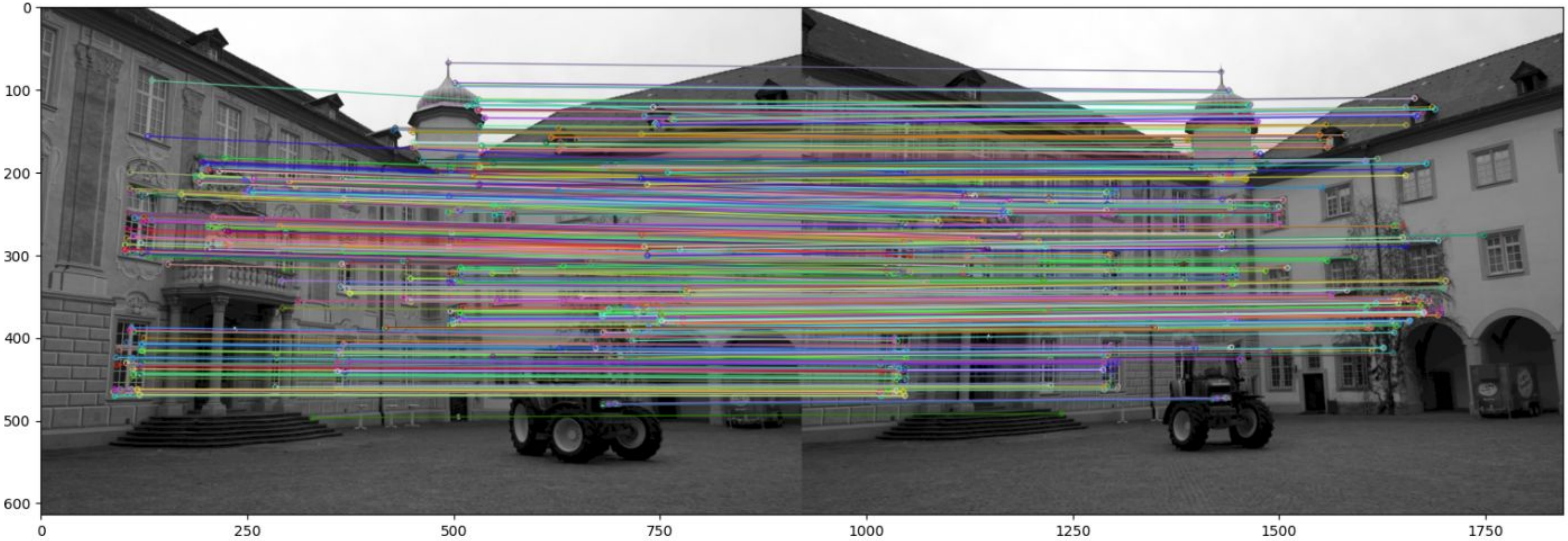




2. Reconstruction from two views
2.1 Estimate the image matches



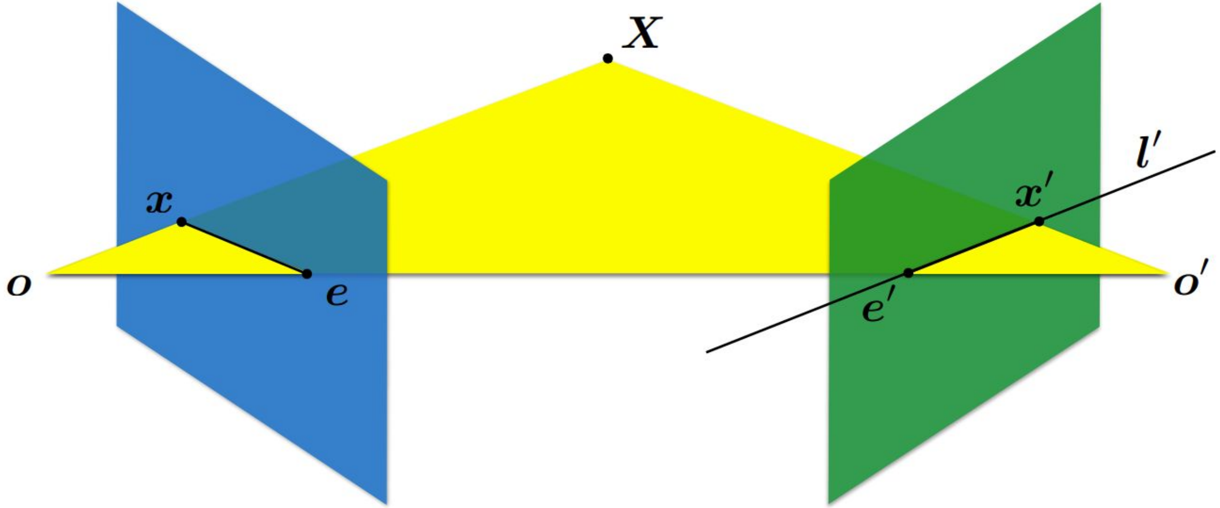
2. Reconstruction from two views
2.2 Estimate the Fundamental Matrix



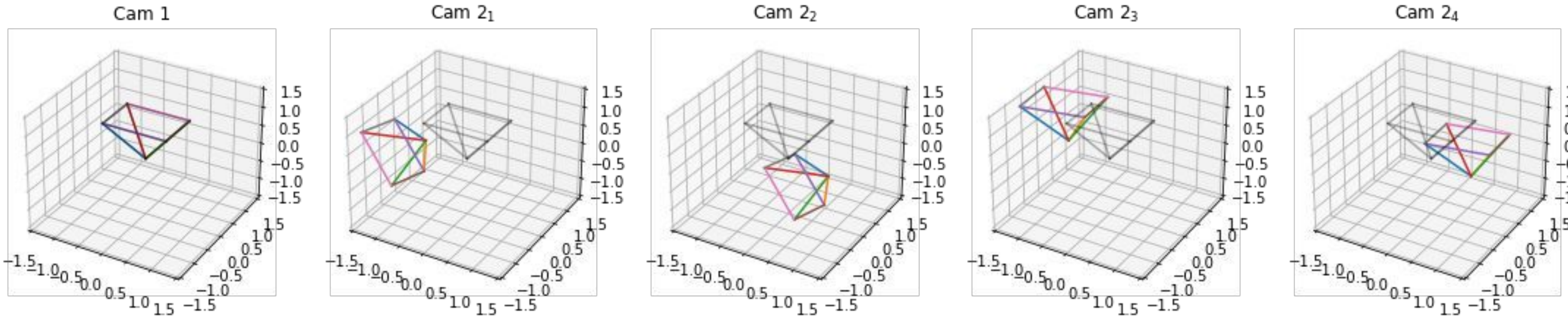
2. Reconstruction from two views

2.3 Estimate the Essential Matrix

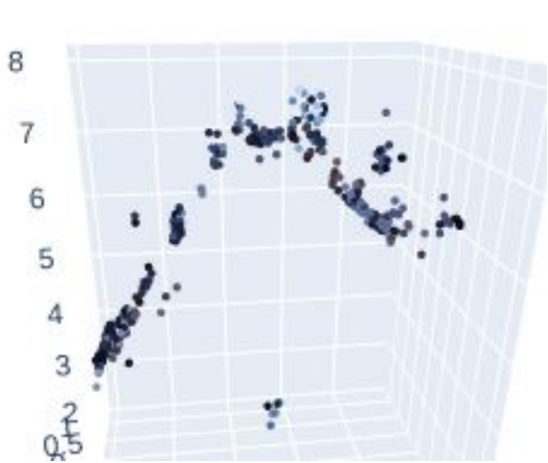
$$\boxed{\mathbf{E}\boldsymbol{x} = \boldsymbol{l}'}$$



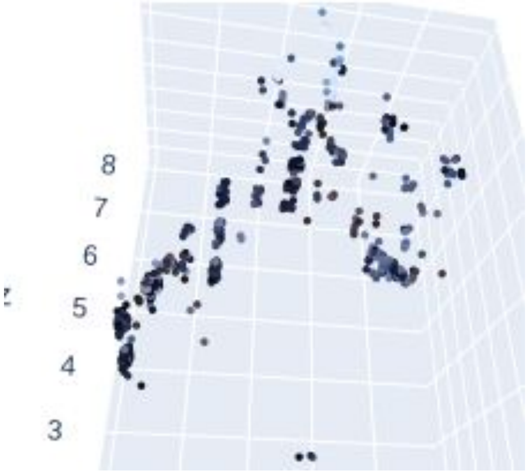
2. Reconstruction from two views
2.4 Estimate the Camera Matrices from the Essential Matrix



2. Reconstruction from two views
2.5 3D Visualization



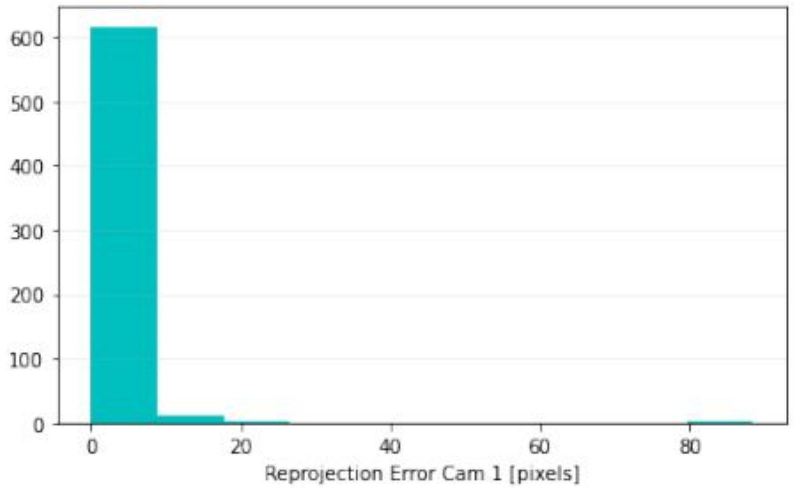
Top view

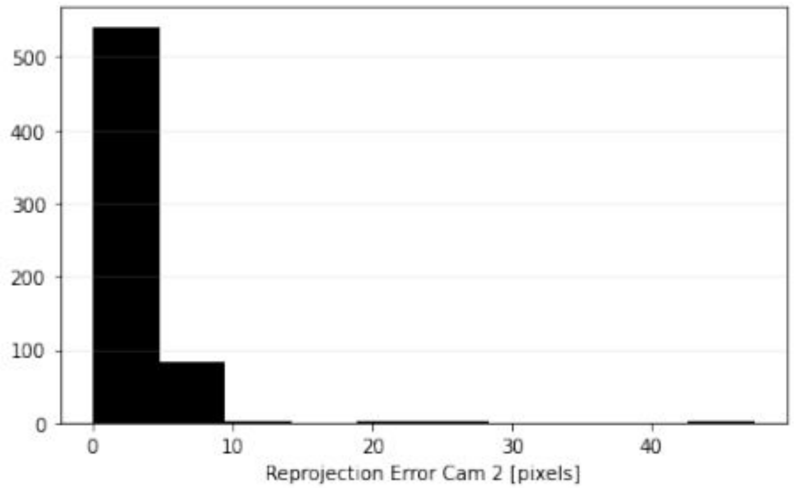


Front view

2. Reconstruction from two views
2.5 3D Visualization: Keypoints

2. Reconstruction from two views
2.6 Reprojection error





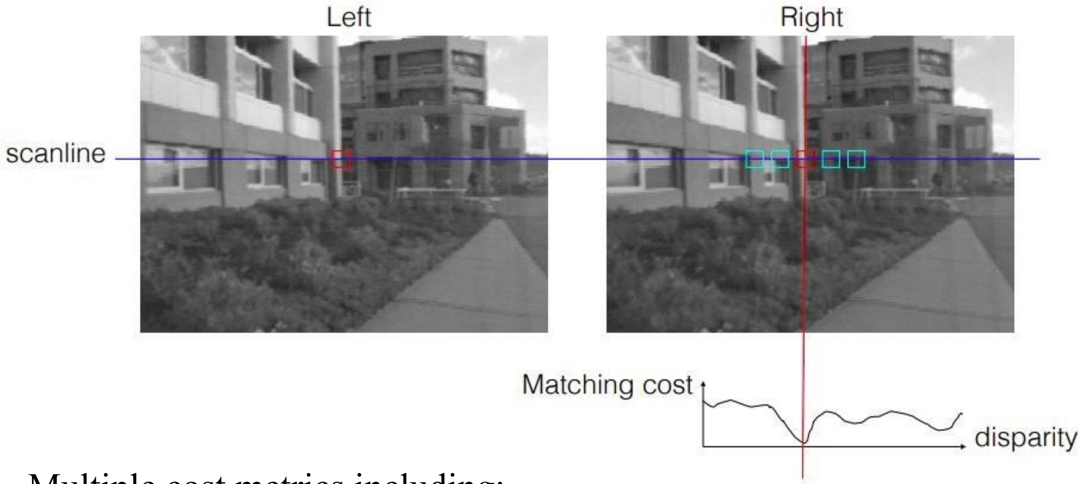
3. Depth map computation using local methods
3.1 Vectorization (hint)

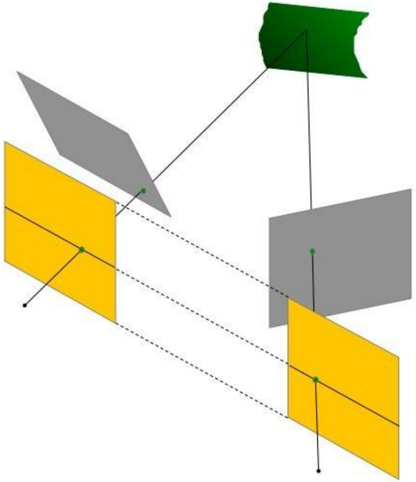
Last year, some implementations took **hours** to run!

Naive method result:	-0.36277495047128566
Parallelized method result:	-0.36277495047128583
Vectorized method result:	-0.36277495047128555

Naive method time:	0.594348669052124 seconds
Parallelized method time:	0.265704870223999 seconds
Vectorized method time:	**0.0051238536834716 seconds** **x100 Speedup**

3. Depth map computation using local methods
3.3, 3.4, 3.6





Multiple cost metrics including:

- 3.3 Sum of Squared Differences
- 3.4 Normalized Cross Correlation
- 3.6 Adaptive support weights

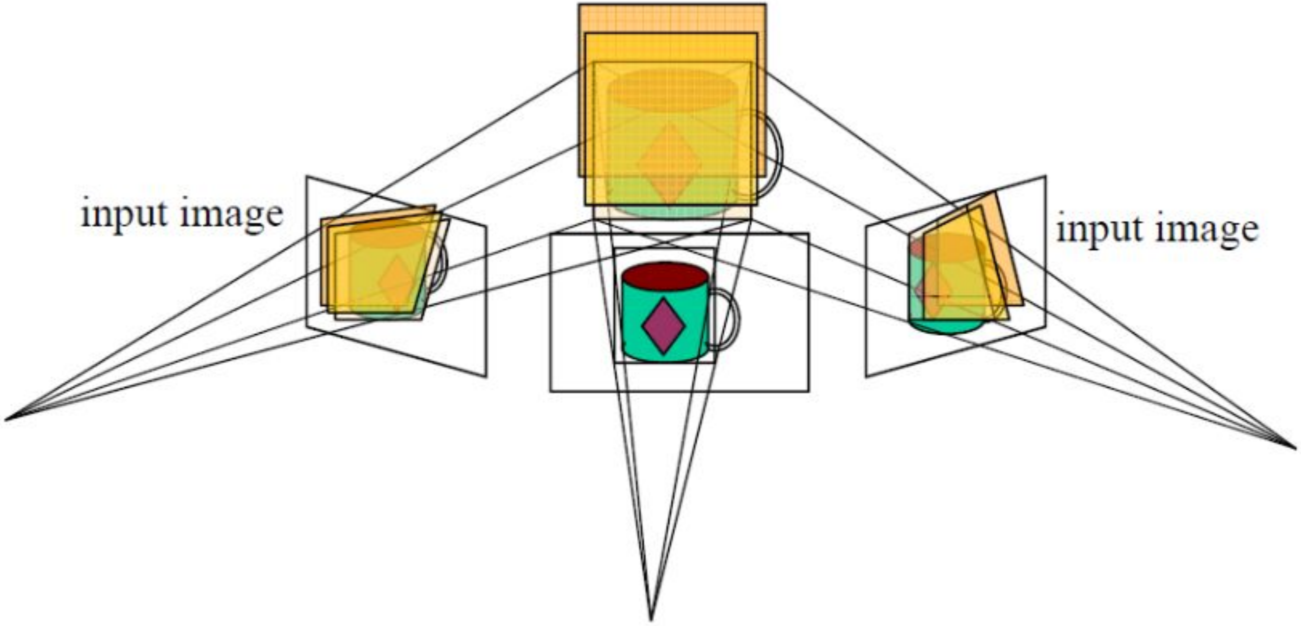
3. Depth map computation using local methods
Apply to Middlebury images

(a) Reference image. (b) Target image. (c) Ground-truth disparity map.

Focal length and baseline of the camera configuration is available at:
https://vision.middlebury.edu/stereo/data/scenes2014/

3. Depth map computation using local methods
3.5 Apply to facade images

Optional Points
Optional 1: Depth map computation with plane sweeping



Optional Points
Optional 2: New view synthesis using view morphing

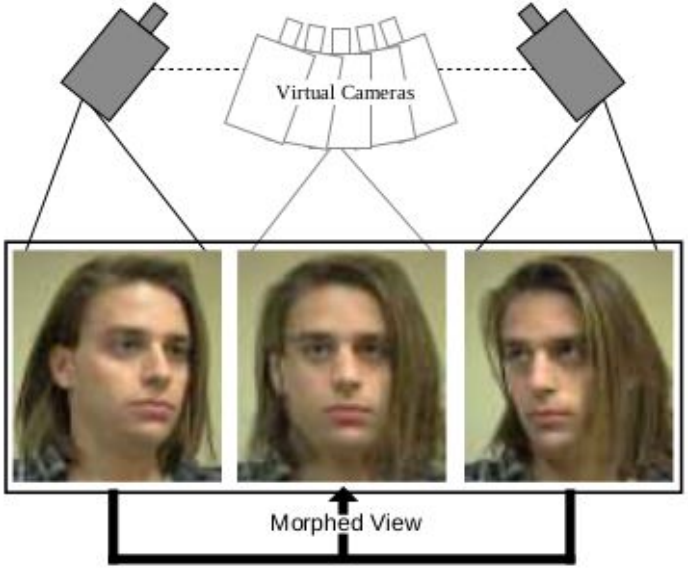


Figure 1: View morphing between two images of an object taken from two different viewpoints produces the illusion of physically moving a virtual camera.

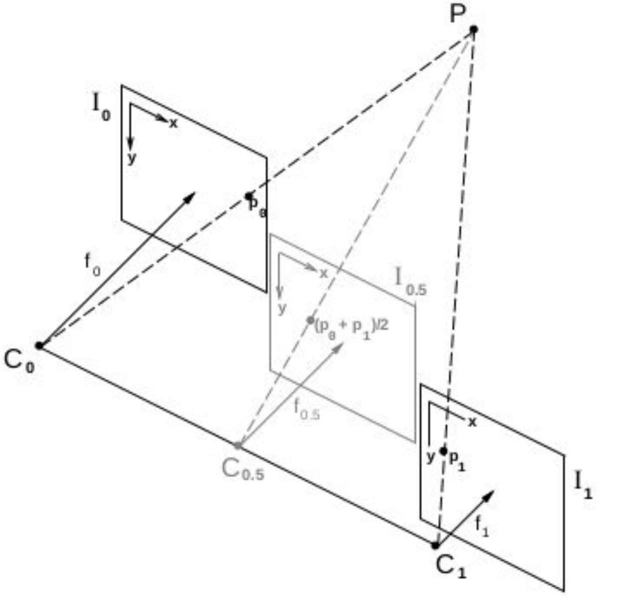
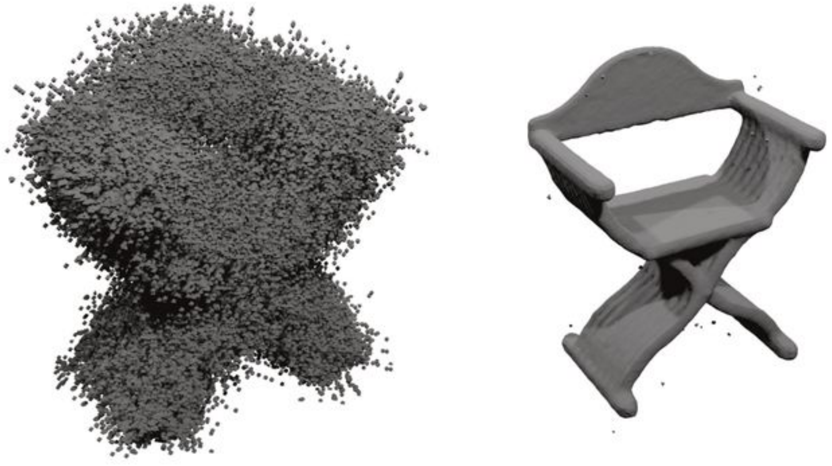


Figure 3: Morphing Parallel Views. Linear interpolation of corresponding pixels in parallel views with image planes $\mathcal{I}_0$ and $\mathcal{I}_1$ creates image $\mathcal{I}_{0.5}$, representing another parallel view of the same scene.

Optional Points
Optional 3: Depth map fusion review

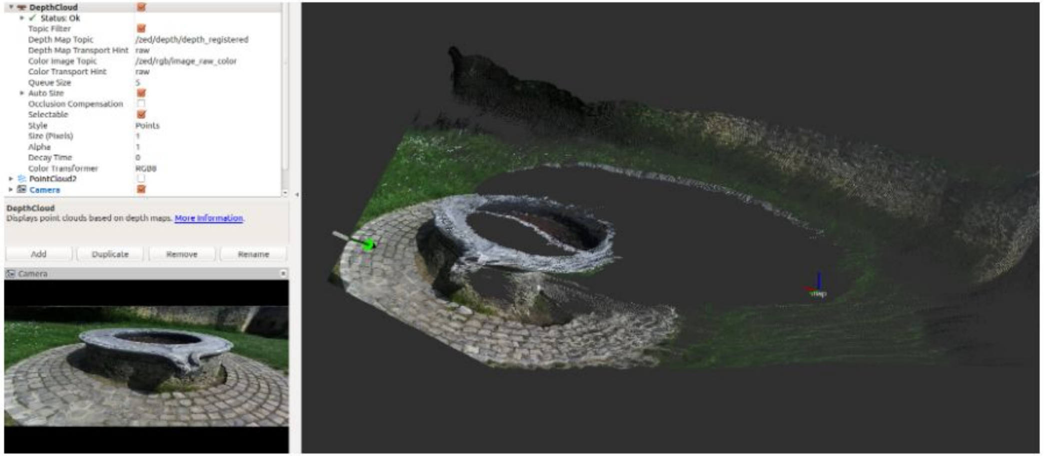
TSDF Fusion | **RoutedFusion**

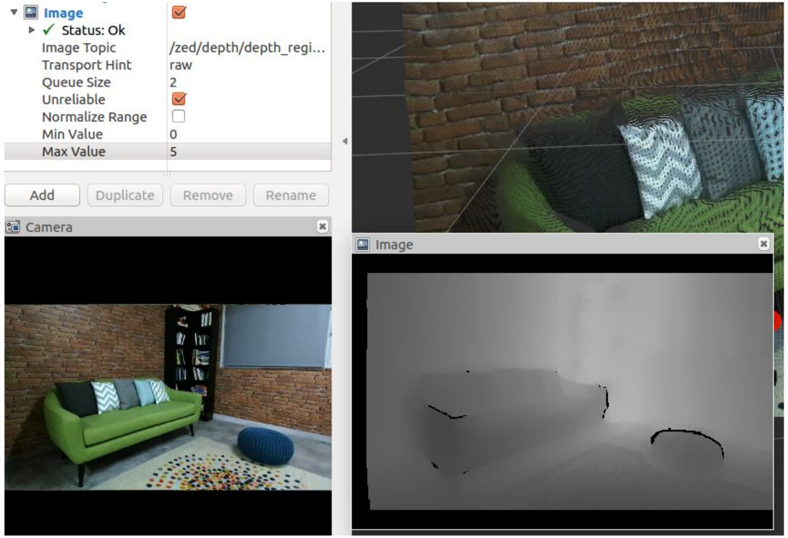


1996 | 2020

Optional Points
Optional 4: Depth map to 3D point cloud

- Create 3D point clouds from the depth maps on the Middlebury images. Both for the ground truth depth maps and for you estimates. Compare the results. What is the effect of noise in the depth map on the point cloud?
- Do the same on the facade images. Compare with the results from task 2.





https://www.stereolabs.com/docs/ros/depth-sensing/