Ground Truth for Evaluating 3D Reconstruction of Jet Engines

1 - Problem

- Measurements of a **jet engine** can be efficiently made by the use of a **3D reconstruction** from a **borescope video** of the engine.
- Ideally evaluate 3D reconstruction with **ground truth**.
- However, there is no ground data available, so only qualitative evaluations can be done.

How to quantitatively evaluate 3D reconstruction of jet engines with ground truth?

How should ground truth data be created?

How should the difference between two 3D models be measured?

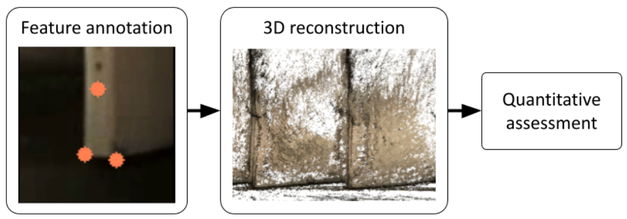

Figure 1: The pipeline for quantitative assessment.

2 - Background

Feature localization

- Ground truth data is more accurate than the results of the algorithms under test.
- In [1], **human-generated features** are used as ground truth for the evaluation of feature detection algorithms.
- **Graphical user interface** for annotating the features.
- Exclusively using human-generated features might not be optimal, since extrema in local geometries can be detected more easily with **algorithmic feature detection**.

Quantitative 3D model comparison

- The **Wasserstein distance** [2] can be used to compare 3D point clouds that represent 3D models.
- It computes the cost of transporting the weight from all points of one point cloud to those of the other point cloud.

3 - Experiments & Results

Feature matching tool *(Figure 2)*

- GUI for manual feature annotation and matching.
- Frames of the input video are visualized.
- User can annotate and delete features by clicking.

Laptop reconstruction *(Figure 3)*

- All frames of a video of a laptop annotated with 16 features.
- The shape of the laptop is clearly observable in the 3D reconstruction.

Optical flow

- Lucas-Kanade optical flow does not track the features within the frames sufficiently.

Linear interpolation *(Figure 4)*

- Deriving feature locations with interpolation.
- If the interpolation interval is between frame 1 and 6 (n = 5), these frames are annotated and the feature coordinates of the intermediate frames are derived.

Combining SuperGlue with interpolation *(Figure 5)*

- SuperGlue is used for feature detection and matching instead of manual annotation.
- Interpolation is overlapped and performed for multiple intervals.
- The resulting 3D model has more quality and less noise than the model reconstructed with only using SuperGlue (Figure 6).

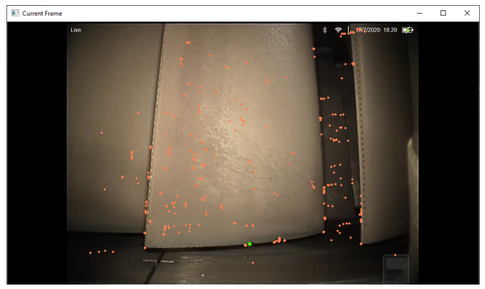

Figure 2: The feature matching tool.

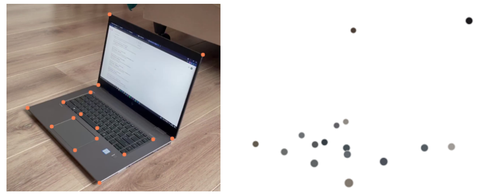

Figure 3: SfM 3D reconstruction of a manually annotated laptop.

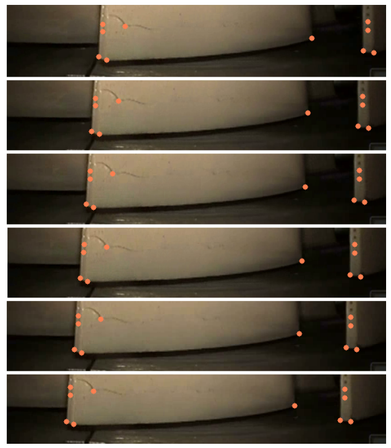

Figure 4: Linear interpolation with n = 5.

Figure 5: 3D model from experiment above (SfM).

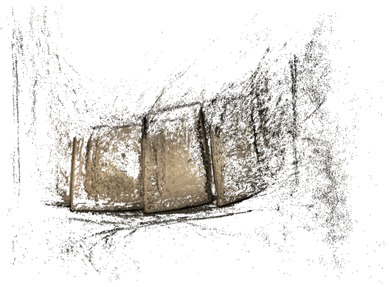

Figure 6: 3D model with SuperGlue features (SfM).

Wasserstein distance

- The Wasserstein distance seems to result in expected distances when comparing 3D point clouds.
- Differences in point locations give higher distances, while differences in order of points do not.
- However, only small point clouds are experimented with because of the high computational cost.

4 - Conclusions

- Features for the ground truth 3D models can partially be localized with a combination of algorithmic feature detection and interpolation with n being not too high.
- Filtering out noisy feature matches and manually adding missing matches is also necessary.
- The 3D reconstruction out of the resulting set of feature localizations and matches can then be created.
- If the ground truth model is more accurate than the results of the 3D reconstruction methods that will be evaluated, it can be utilized for assessment.
- It might be interesting to conduct further research on the performance of the Wasserstein distance for the comparison of jet engine 3D reconstructions.

Research Project (CSE3000)

Author: Devin Lieuw A Soe (4933028)

Email: D.L.LieuwASoe@student.tudelft.nl

Supervisor 1: Jan van Gemert

Supervisor 2: Burak Yildiz

TU Delft

[1] H. Dutagaci, C. P. Cheung, and A. Godil, "Evaluation of 3d interest point detection techniques via humangenerated ground truth," The Visual Computer, vol. 28, pp. 901–917, 2012.

[2] K. Kawano, S. Koide, and T. Kutsuna, Learning Wasserstein Isometric Embedding for Point Clouds, 2020, pp. 473–482.