Politecnico di Torino - M.Sc in Data Science and Engineering

AI-Driven Feature Detection, Matching, and Efficient Model Deployment for Space Applications

AIKO

INFINITE WAYS TO AUTONOMY

Supervisors
Prof. Daniele Jahier Pagliari
Dr. Alessio Burrello
Dr. Armando La Rocca
Dr. Matteo Stoisa

Candidate
Alessandro Ficca (319084)

Objectives

- ➔ Compare classical algorithms and state-of-art Deep Learning models in **Feature Detection and Matching (FDM)** for **Space** applications
- ➔ Apply **Knowledge Distillation** to replace Attention layers in Transformer-based models

AGENDA

- ➔ Feature Detection and Matching for space applications
- ➔ AI approaches for Feature Detection and Matching
- ➔ Comparison: Classical vs AI methods
- ➔ Knowledge Distillation

FDM for space applications

What is Feature Detection and Matching?

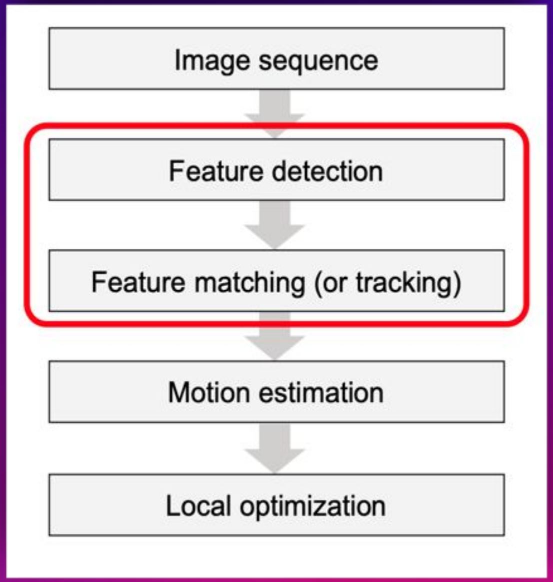
***FDM** is the process of identifying similar **interest points** between two images of the same scene captured under different conditions*

Why do we need keypoints?

Keypoints are fundamental for motion estimation

Many applications use local features tracking

- 3D object reconstruction
- Image registration
- Visual Simultaneous Localization and Mapping (V-SLAM)



Why FDM in space?

FDM can be used for **Optical Navigation (ON)** in space applications

Optical Navigation: *Use of on-board cameras to capture visual information of a target and perform relative navigation*

ON is relevant in space

➔ Enables autonomous navigation
➔ Substitutes traditional active navigation sensors
➔ Improves navigation precision

Space missions:

- Active Debris Removal
- On-Orbit Servicing
- Rendezvous and Proximity Operations
- Deep Space exploration

Space Datasets

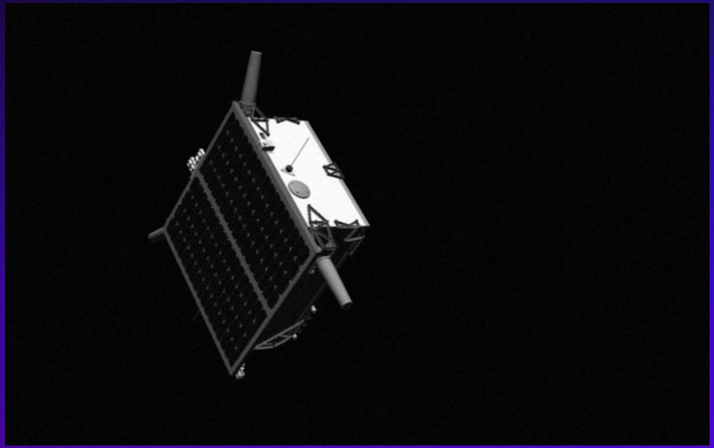
Space-related datasets for Satellite Pose Estimation

(Estimate position and orientation of a satellite target respect to the camera)

➔ **SPEED** [1]

➔ **SPEED+** [2]

➔ **SPEED-UE-Cube** [3]



TANGO spacecraft

3U CubeSat spacecraft

[1] Mark Kisantal et al. “Satellite Pose Estimation Challenge: Dataset, Competition Design and Results”.
[2] Tae Ha Park et al. “SPEED+: Next-Generation Dataset for Spacecraft Pose Estimation across Domain Gap”.
[3] Zahra Ahmed et al. “SPEED-UE-Cube: A Machine Learning Dataset for Autonomous, Vision-Based Spacecraft Navigation”.

Feature Detection and Matching through AI techniques

Classical FDM Algorithms

➔ Classical approaches in FDM: **SIFT** [4], **ORB** [5] and **AKAZE** [6]

➔ Traditional pipeline: Feature **Detection**, **Description** and **Matching**

➔ Algorithmic approach and mathematical operations:

- Gaussian and non-linear diffusion filtering, local image gradient
- Search of local extrema
- Fixed descriptor generation

➔ Matching:

- Descriptors similarity
- K-Nearest Neighbours with ratio test

[4] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints
[5] Ethan Rublee et al. “ORB: An efficient alternative to SIFT or SURF”
[6] Pablo F. Alcantarilla et al. “Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces”

AI FDM Methods

- Deep Learning
- **SuperPoint** [7]
  - CNN-based
  - Simultaneous feature extraction and description
- **LightGlue** [8]
  - Attention-based Graph Neural Network (keypoints as “nodes”)
  - Feature matching only
- **LoFTR** [9]
  - Transformer-based
  - No feature detection
  - directly produces matches from dense grid of the images

Used together

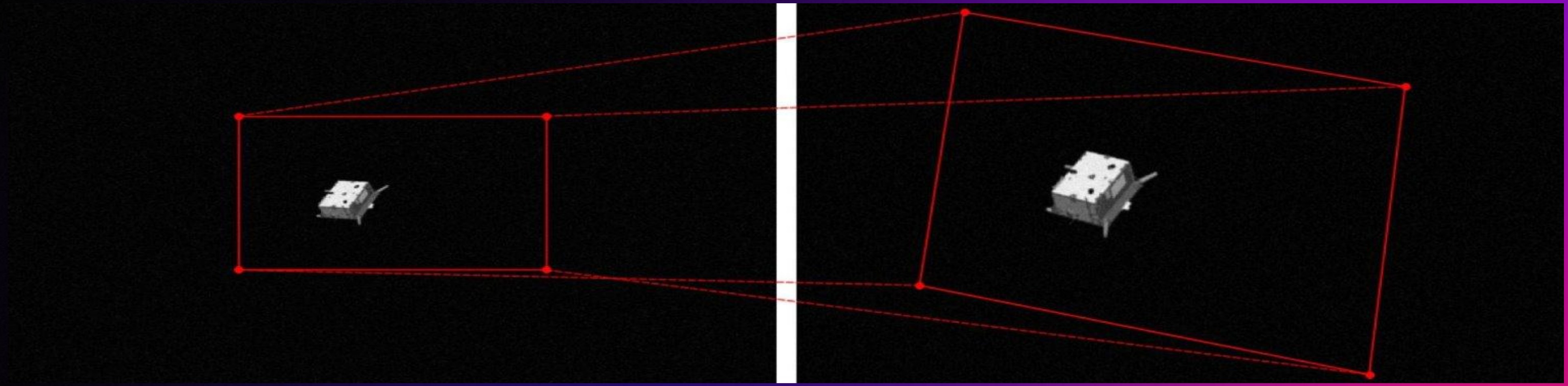
[7] Daniel DeTone et al. SuperPoint: Self-Supervised Interest Point Detection and Description. 2018
[8] Philipp Lindenberger et al. LightGlue: Local Feature Matching at Light Speed. 2023
[9] Jiaming Sun et al. LoFTR: Detector-Free Local Feature Matching with Transformers. 2021

Data Preparation and Evaluation metrics

Homography Estimation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = H \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

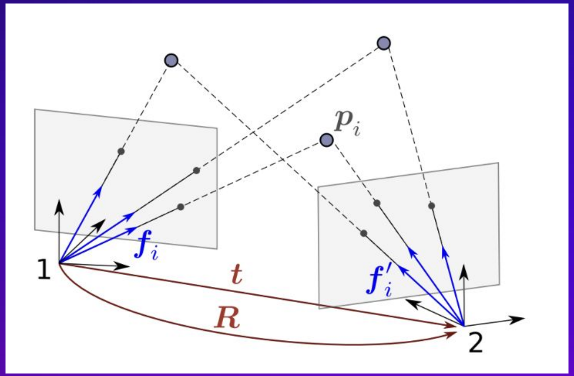
- ➔ Apply a random geometric transformation
  - ◆ Mix of rotation, translation, scaling, perspective distortion
- ➔ **Problem:** Estimate the ***homography matrix H*** between two images

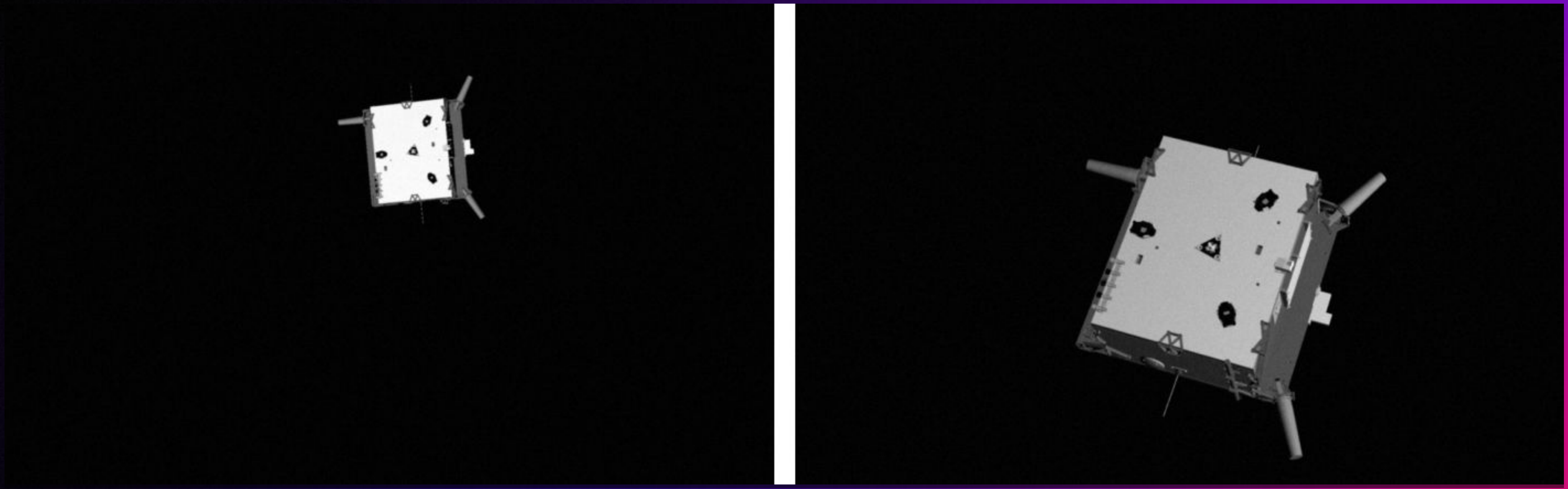


Relative camera pose estimation

➔ **Pose:** Position ***t*** and orientation ***R*** of 2° camera respect to 1° camera

➔ **Problem:** Estimate the camera pose between two frames

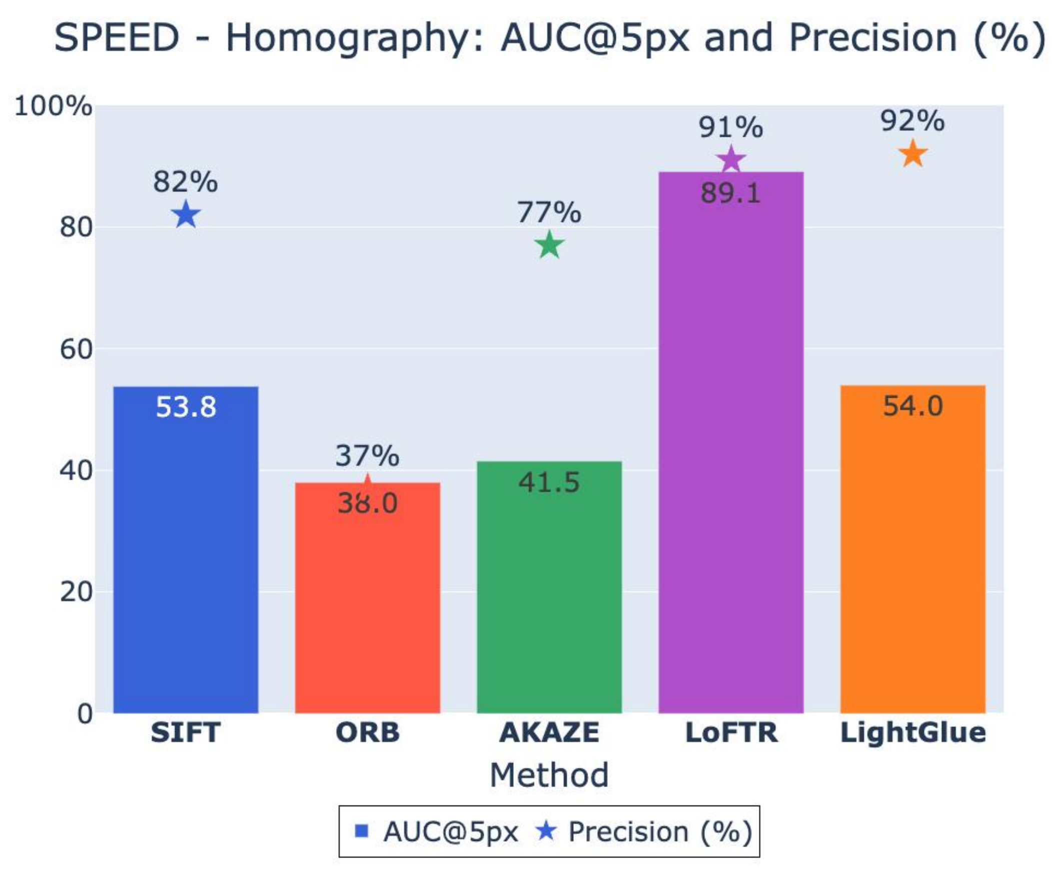




Comparison:
Classical vs AI methods

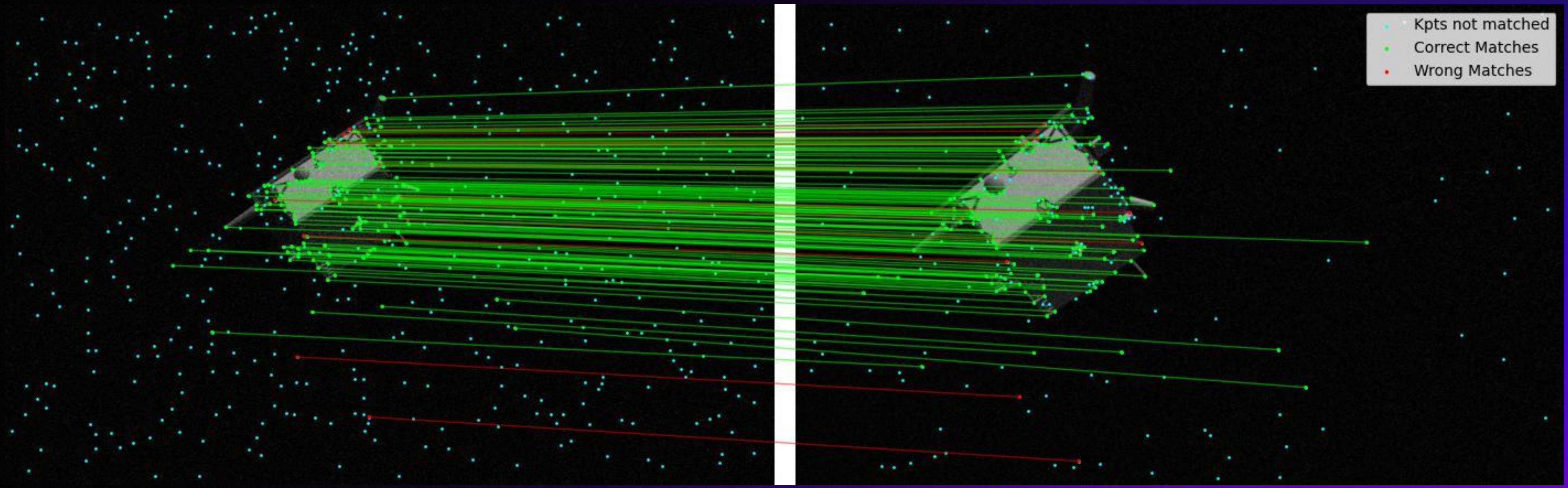
Homography estimation results

- **SPEED** Datasets
- **Mean reprojection error:** distance in pixels between corners image projected with the GT and estimated Homography
- **AUC** * at 5 pixel threshold
- **Precision:** Percentage of correctly predicted matches compared to GT correspondences

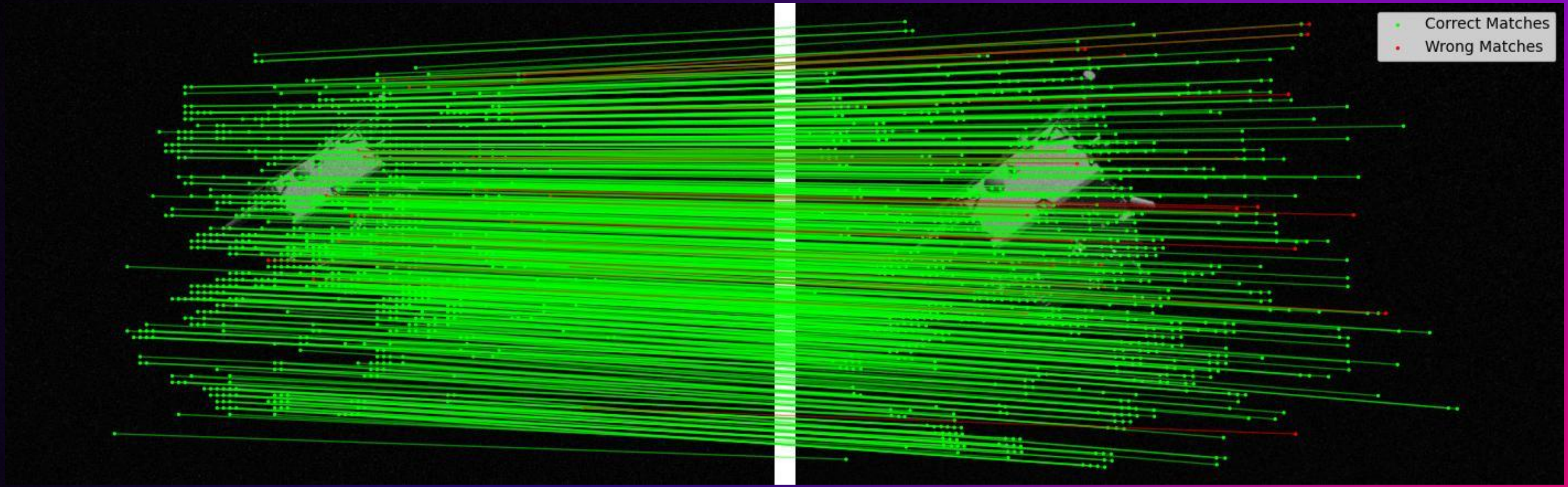


* **AUC:** Area under the cumulative curve of the error up to a specific threshold

LightGlue

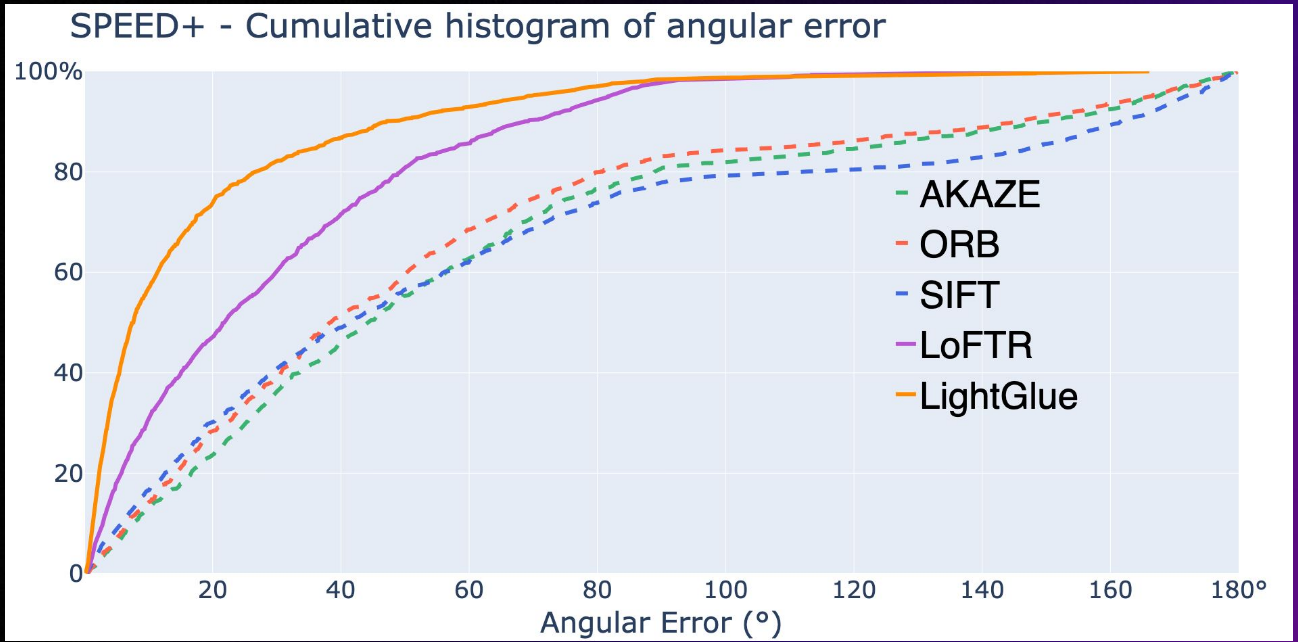


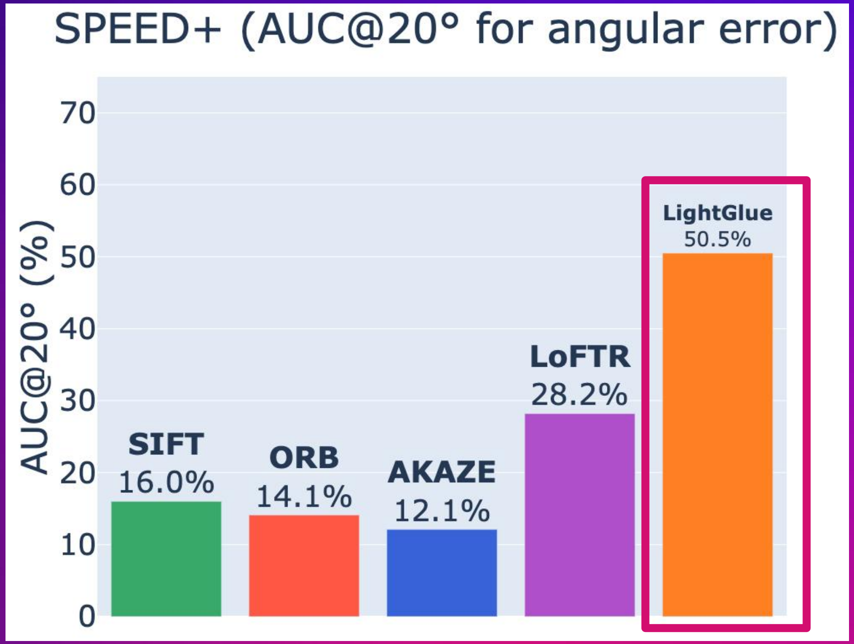
LoFTR



Relative pose estimation results

- ➔ **SPEED+** Datasets
- ➔ Metric: **Angular error** (maximum angular deviations [°] between rotation and translation error)





Knowledge Distillation

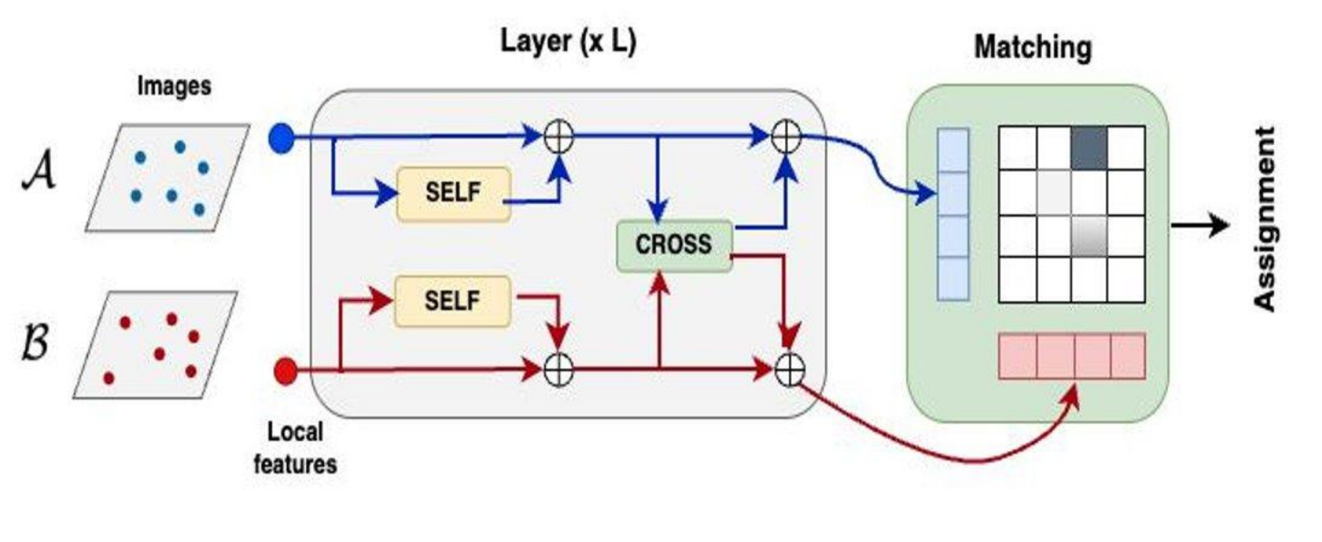
Knowledge Distillation *trains a Deep Learning model under the supervision of a different pre-trained model to effectively transfer knowledge*

Student Architecture

- ➔ Teacher model: **LightGlue**
- ➔ Student replaces the Attention layers
- ➔ Enable the deployment on space-grade hardware

Self-Attention
⬇
2D convolutional layers [10]

Cross-Attention
⬇
Cross-Graph learning module [11]

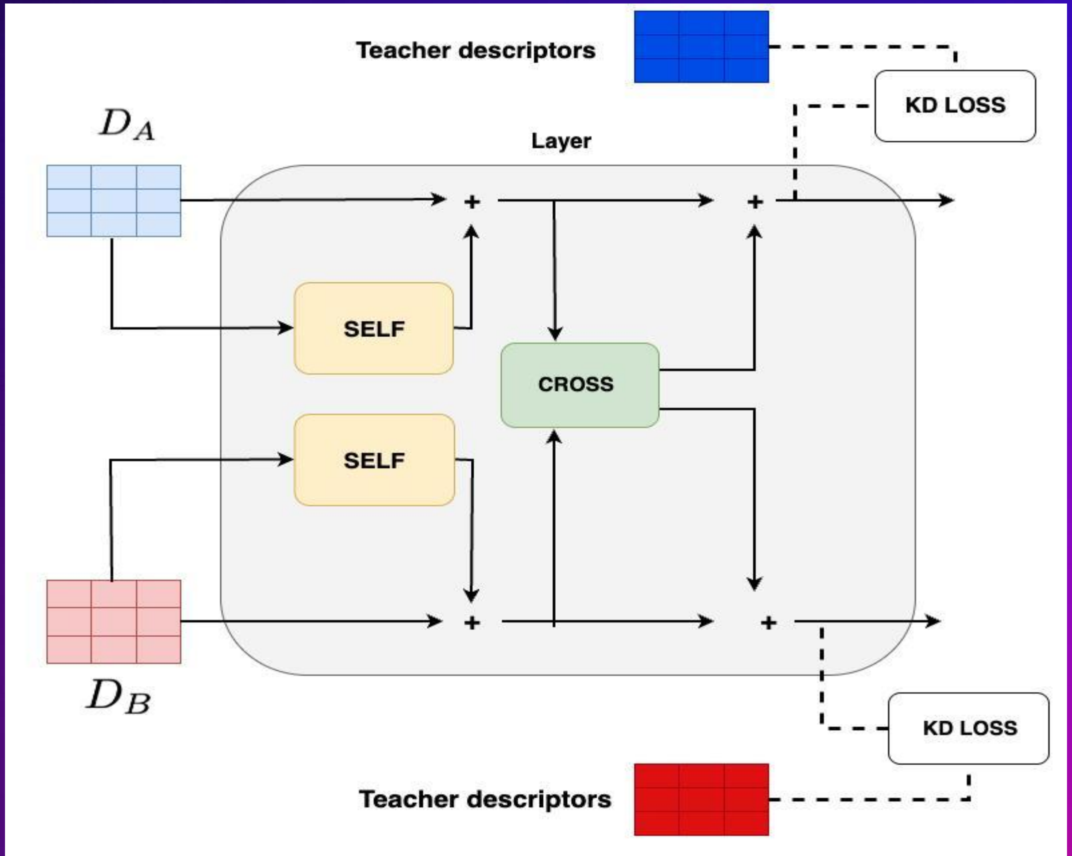


Overview student architecture

[10] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s.
[11] Bo Jiang, Pengfei Sun, Jin Tang, and Bin Luo. GLMNet: Graph Learning-Matching Networks for Feature Matching

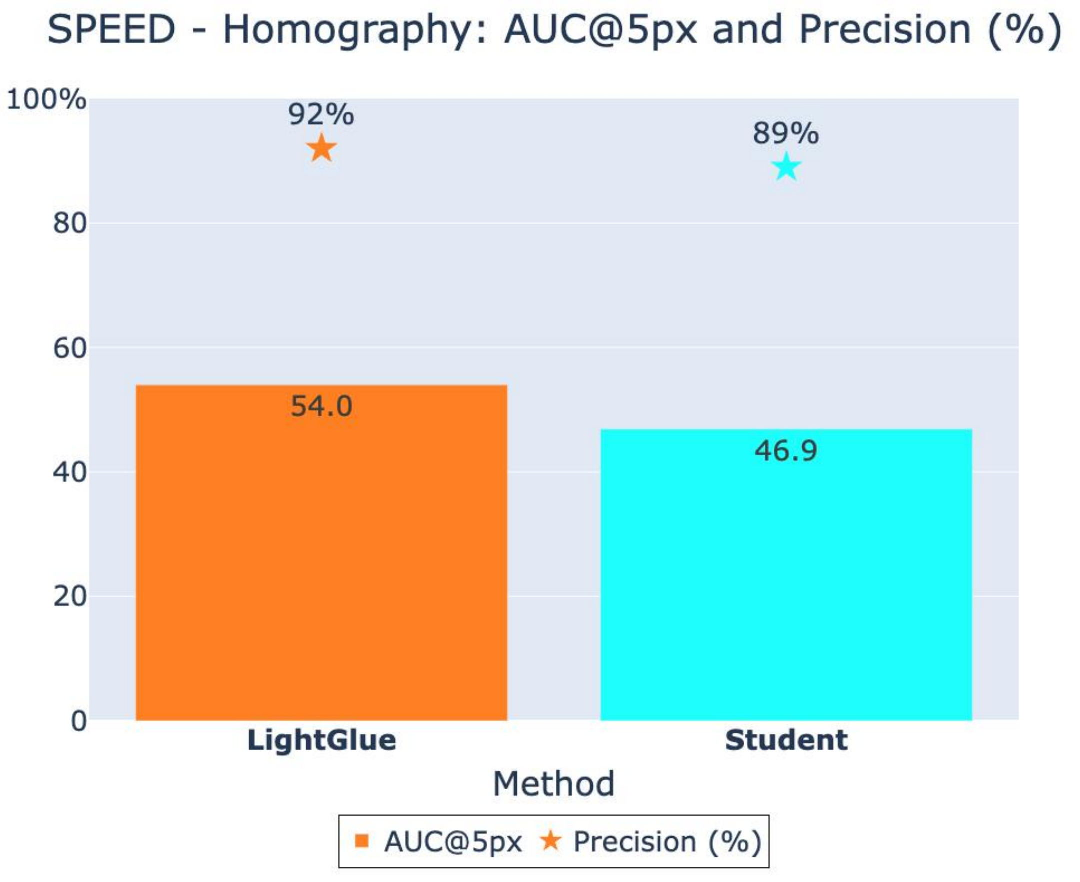
Training

- ➔ **Feature-based** Offline Distillation Strategy
- ➔ KD loss**: Mean Square Error**
- ➔ Training dataset: **SPEED**
- ➔ Pre-train:
  - ◆ Synthetic homographies ***H***
  - ◆ GT supervision from ***H*** + KD loss
- ➔ Fine-tuning:
  - ◆ Image pairs based on camera transformation
  - ◆ Pseudo GT (LightGlue predictions) + KD loss



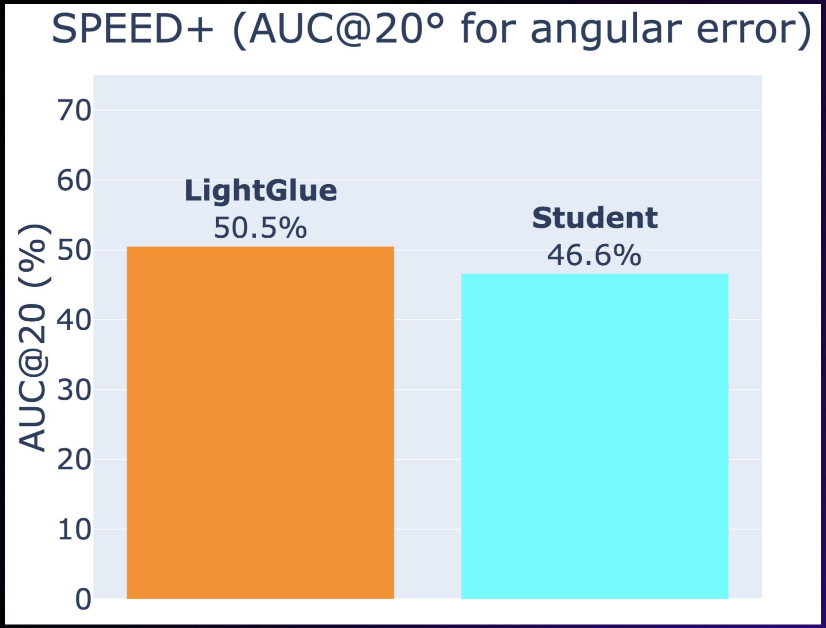
KD results

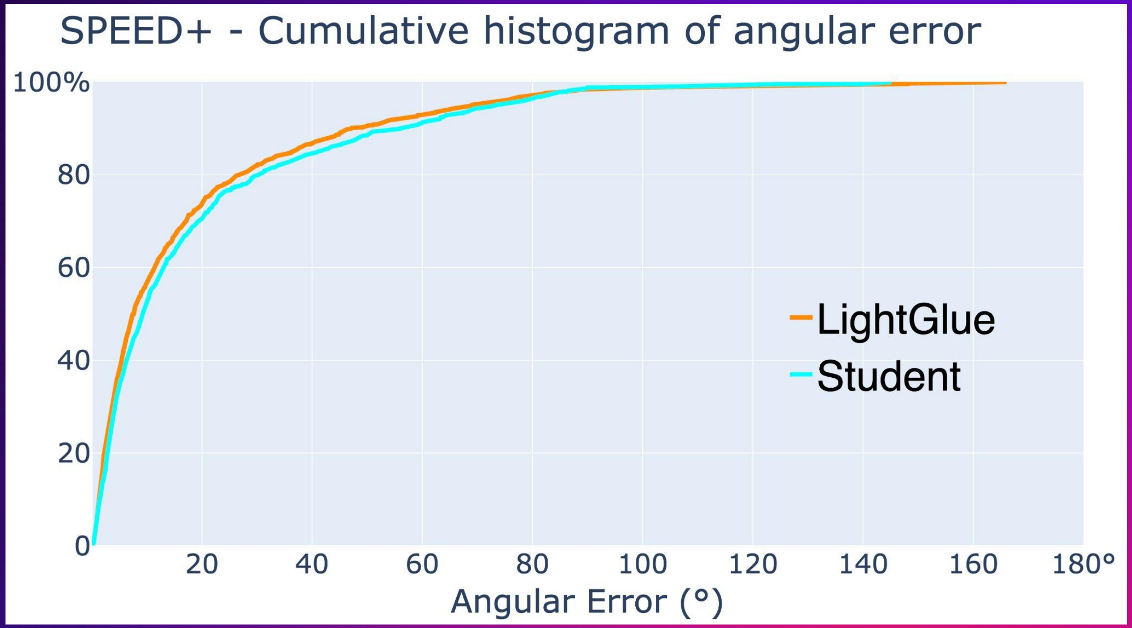
- **Homography** estimation
- SPEED dataset
- Gap teacher-student
  - Precision: **-3 %**
  - AUC@5px: **-7.1%**



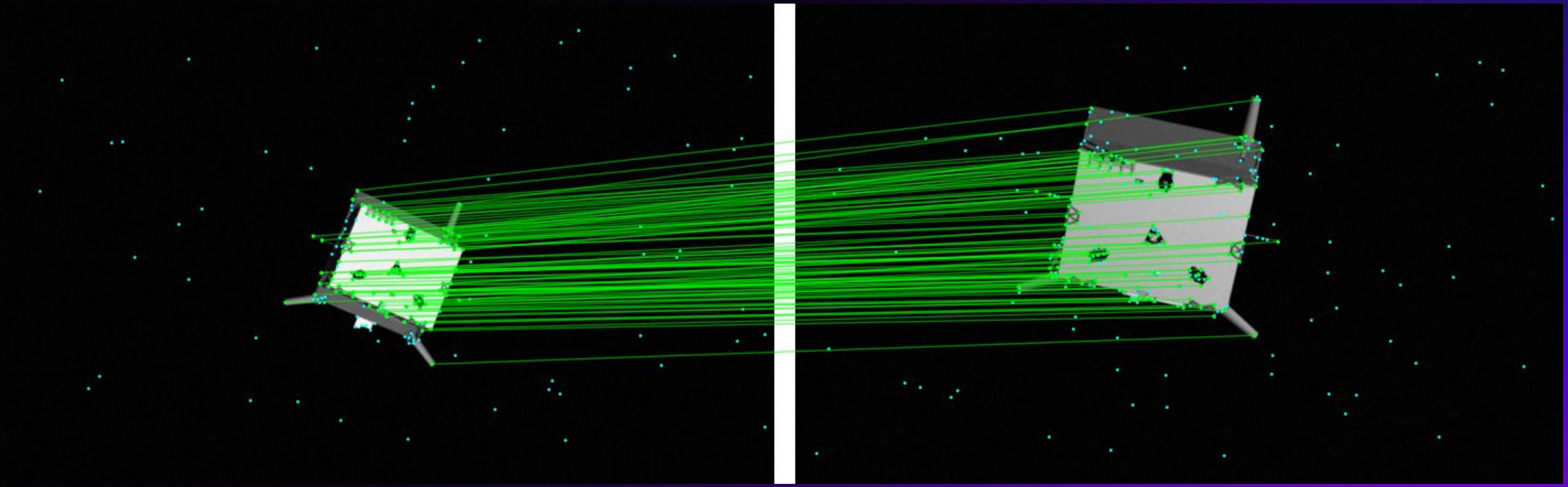
KD results

- ➔ **Relative camera pose** estimation
- ➔ SPEED+ dataset
- ➔ Gap teacher-student: **-3.9%** (AUC@20°)

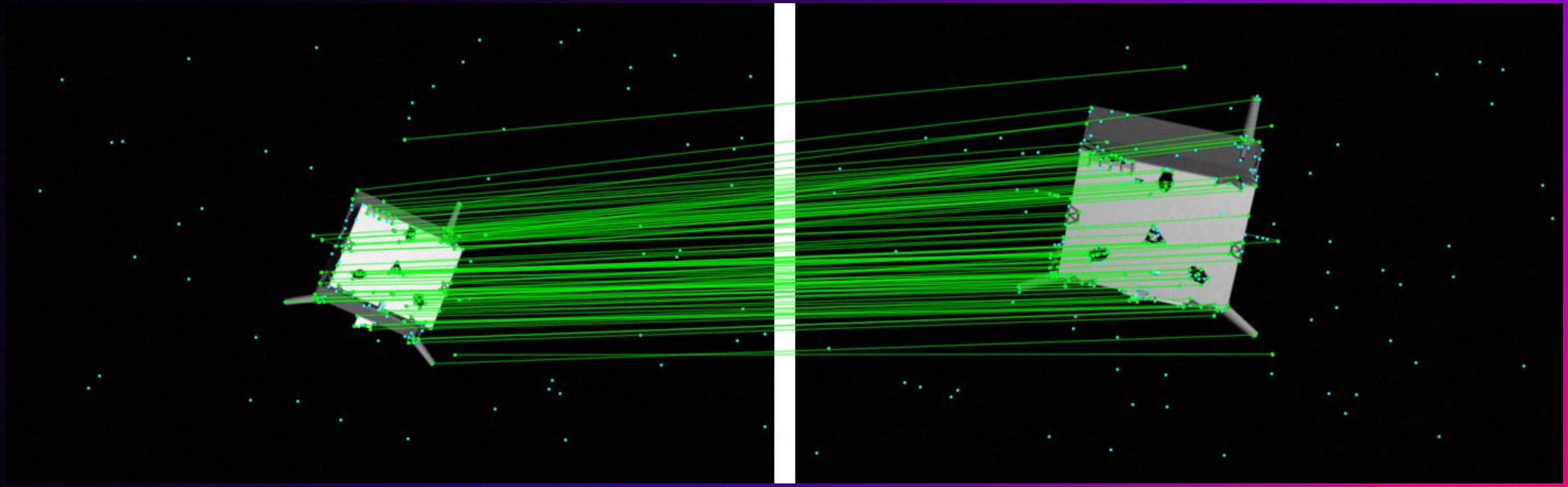




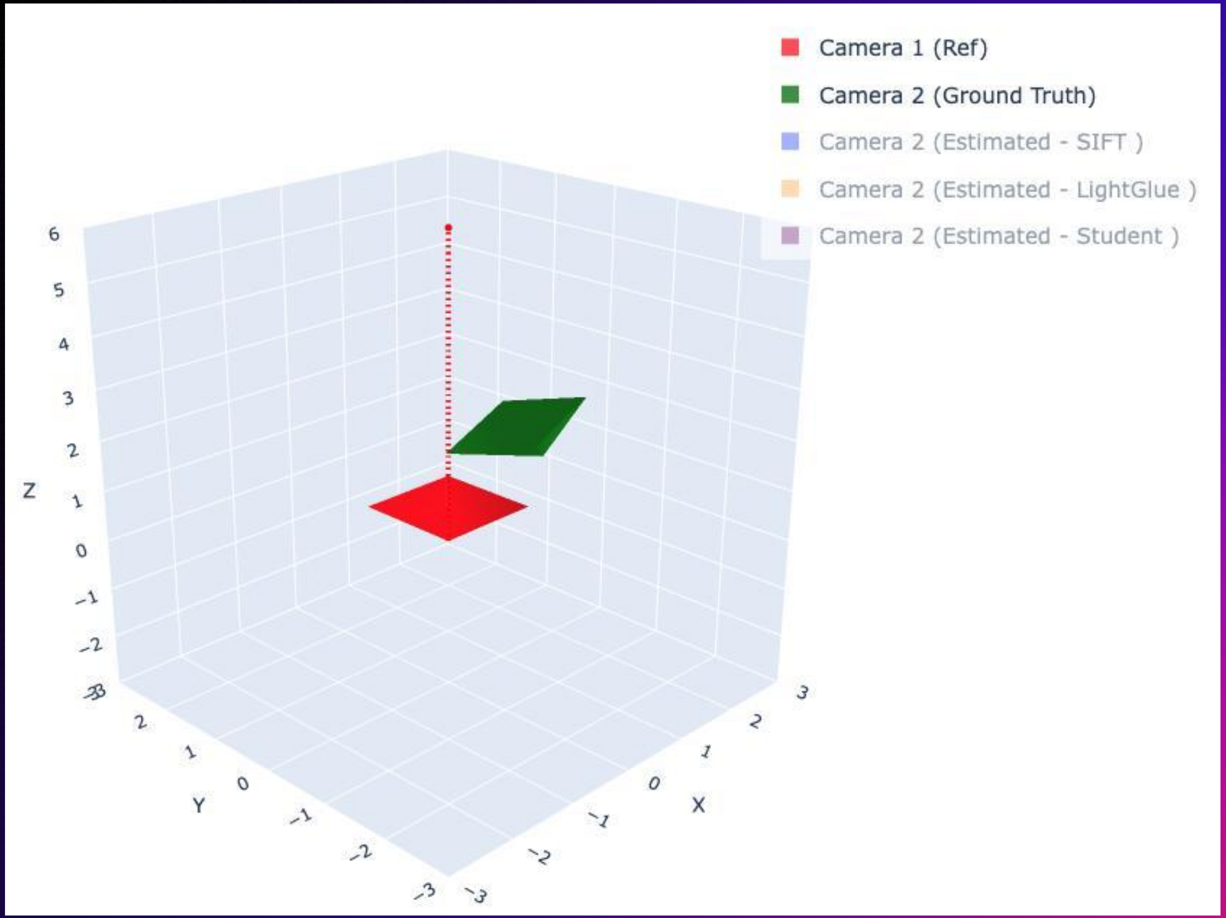
LightGlue



Student



Scene reconstruction from images

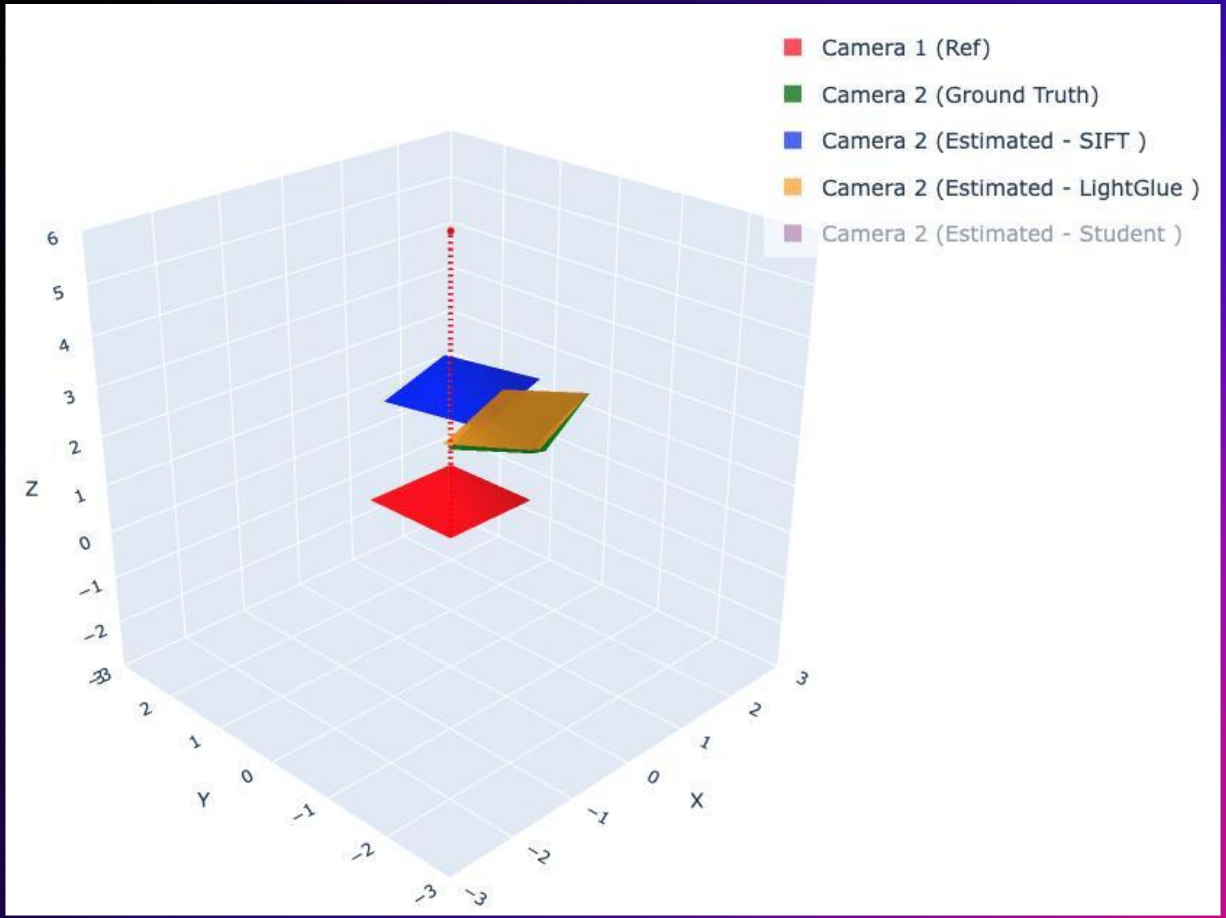


Scene reconstruction from images

Angular errors
(compared to GT pose):

SIFT: 54.6°

LightGlue: 5.9°



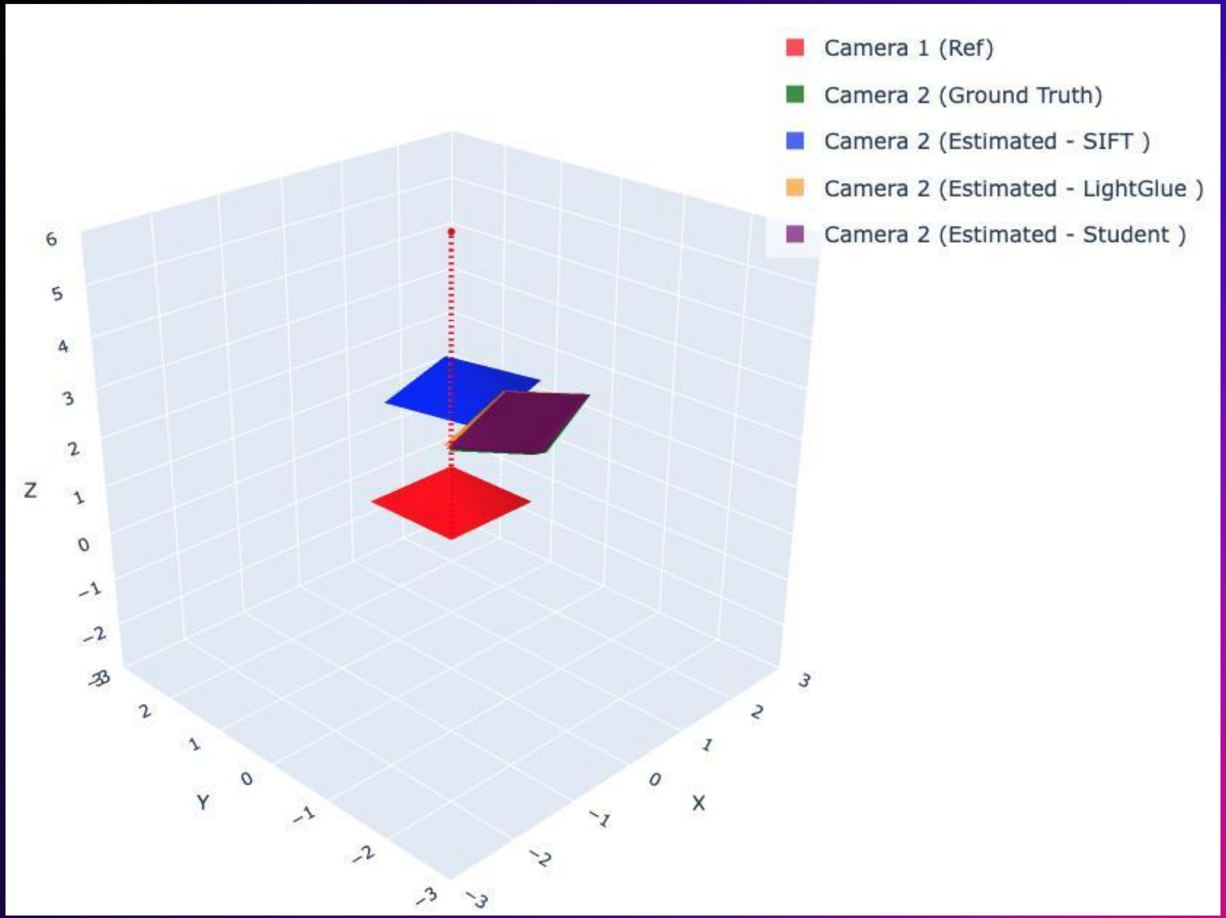
Scene reconstruction from images

Angular errors:

SIFT: 54.6° (vs GT)

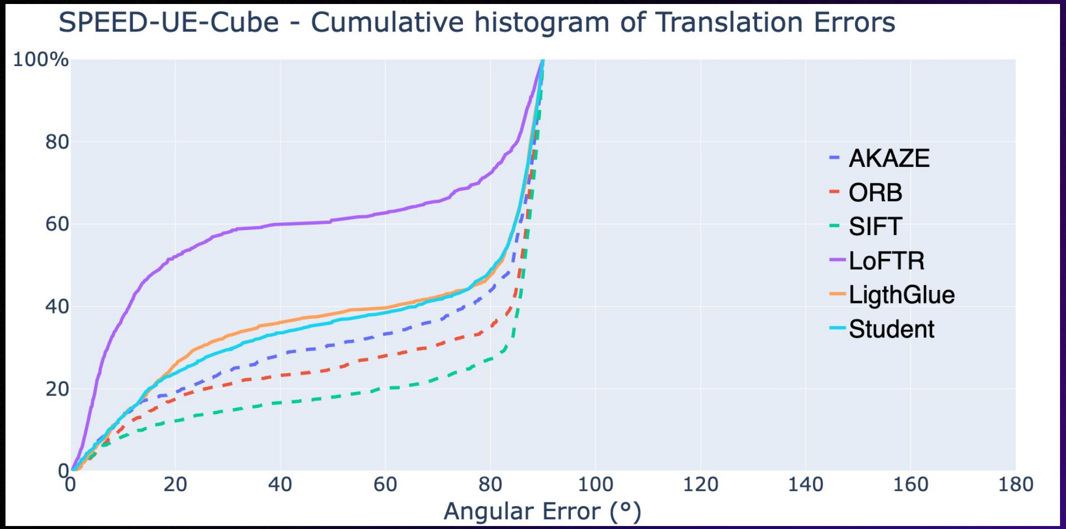
LightGlue: 5.9° (vs GT)

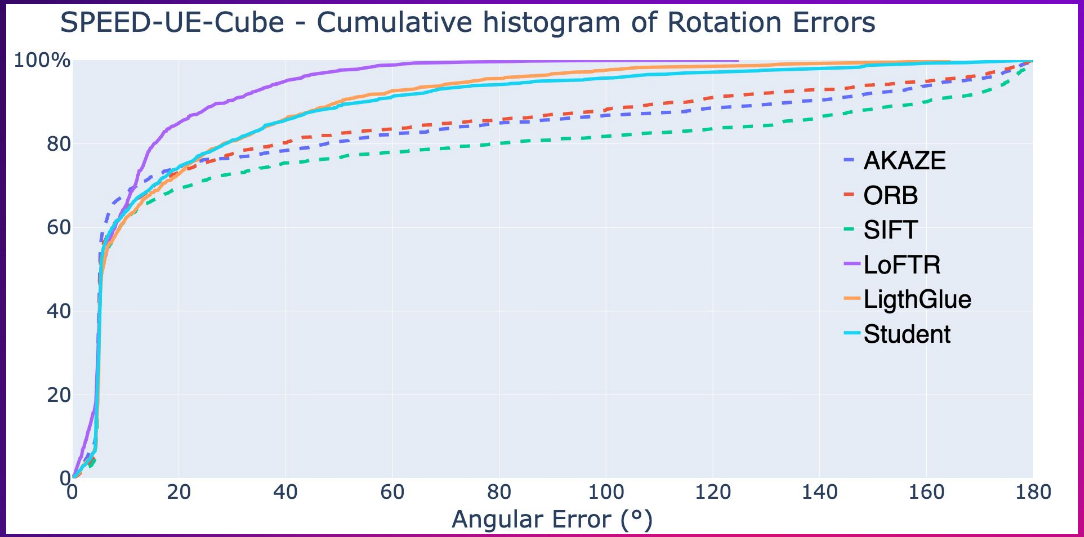
<u>Student:</u> 1.8° (vs GT)



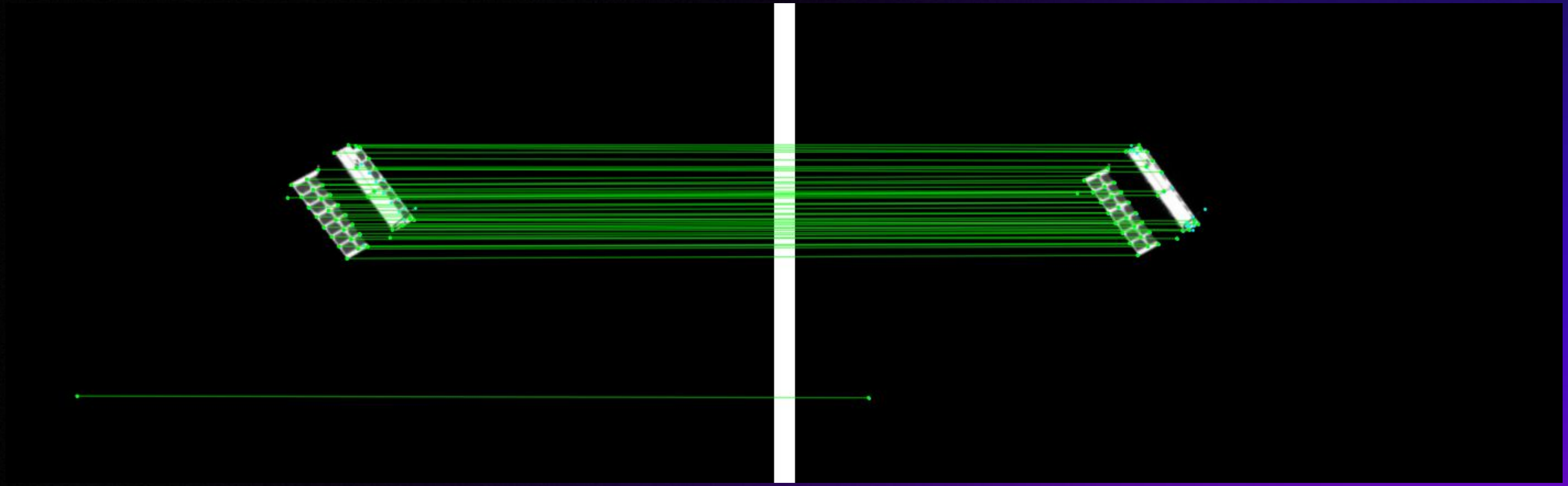
Relative pose estimation on trajectory

- ➔ **SPEED-UE-Cube** Datasets
- ➔ Trajectory set: sequential frames

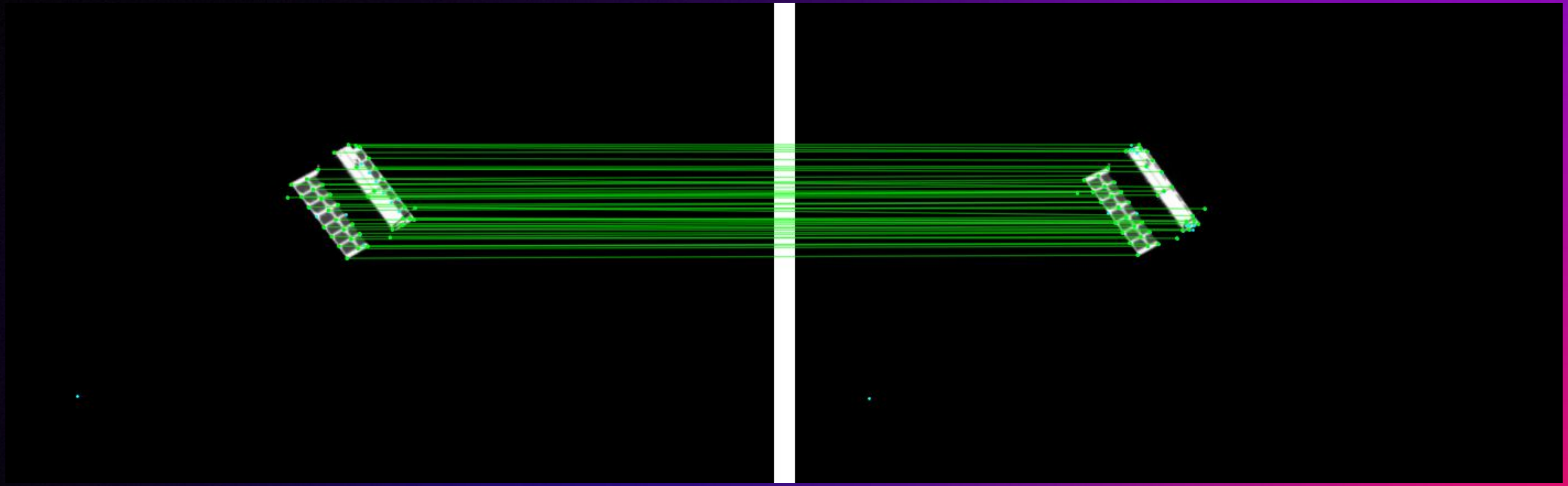




LightGlue

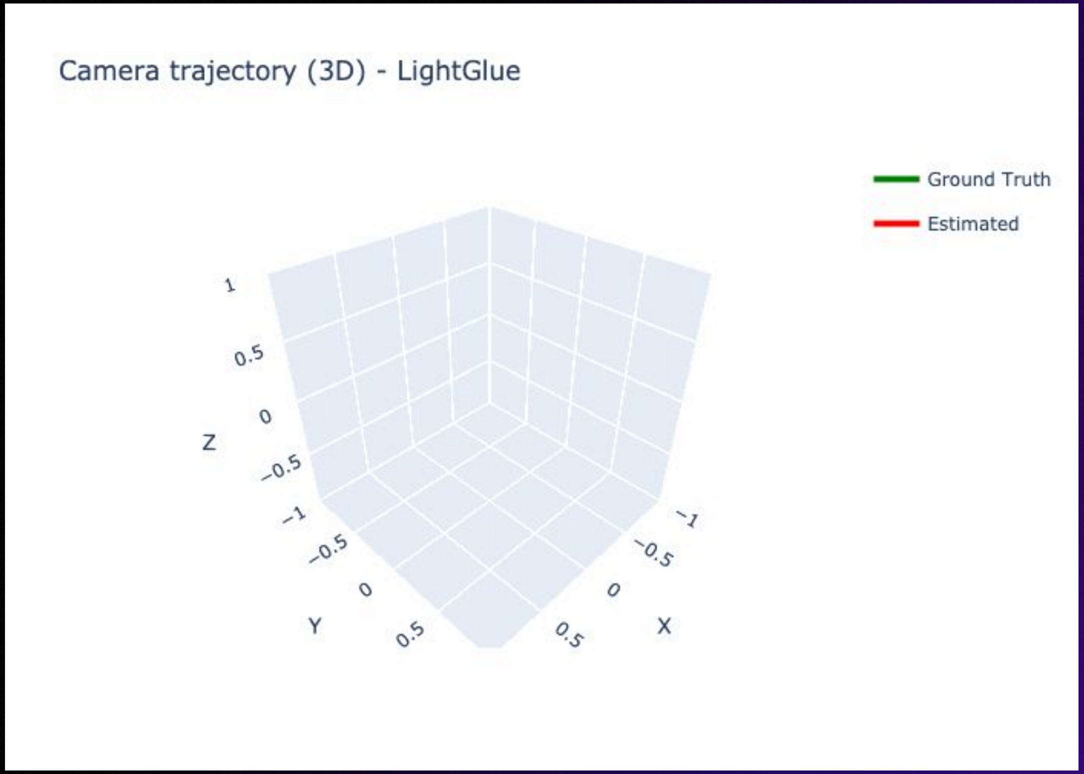


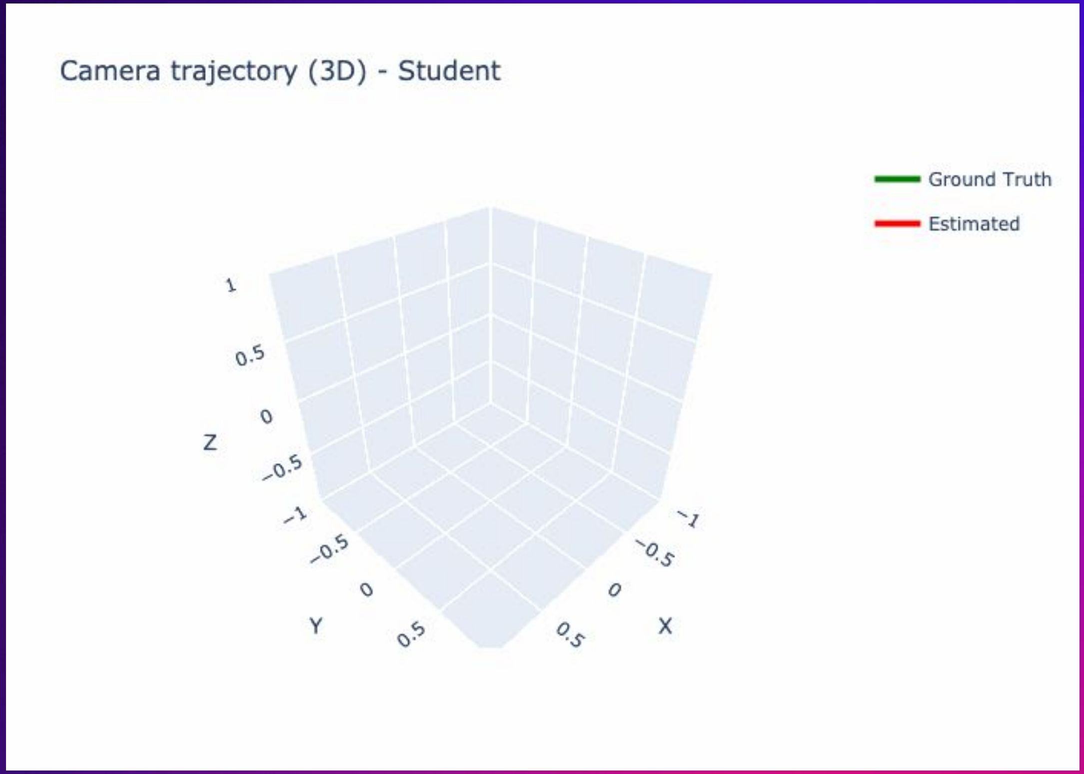
Student



Camera trajectory simulation

- Reconstruction of the trajectory from the images with **LightGlue** and **Student** model and GT trajectory.
- Position of camera at frame *i* is computed from GT position at frame *i-1* to avoid the accumulation of the error during the trajectory





Conclusions

- ➔ AI methods outperform classical algorithms in FDM
- ➔ AI models are able to specialize from generic visual context to more challenging space scenarios
- ➔ KD is effective to transfer knowledge in this task to a student model with different operations and fully-supervised by the teacher

Next Steps

- ➔ Apply optimization techniques to improve inference performance and dimensions of Student model
- ➔ Test deployment on space-grade hardware
- ➔ Integrate FDM in Optical Navigation systems

Thanks for your attention!

AIKO

INFINITE WAYS TO AUTONOMY

www.aikospace.com _info@aikospace.com

FOLLOW US ON

Appendix

Metrics

➔ **Mean reprojection error:** distance in pixels between corners image projected with the GT and estimated Homography

$$\text{reprojection error} = \frac{1}{4}\sum_{j=1}^{4} \|\mathbf{c}'_j - \hat{\mathbf{c}}'_j\|$$

➔ **Angular deviations:** Deviations between the GT relative pose and the estimated relative pose

$$\theta_t = arccos(\frac{t \cdot \hat{t}}{||t|| \cdot ||\hat{t}||}) \qquad \theta_r = arccos(\frac{trace(R^T \cdot \hat{R}) - 1}{2})$$

Student architecture

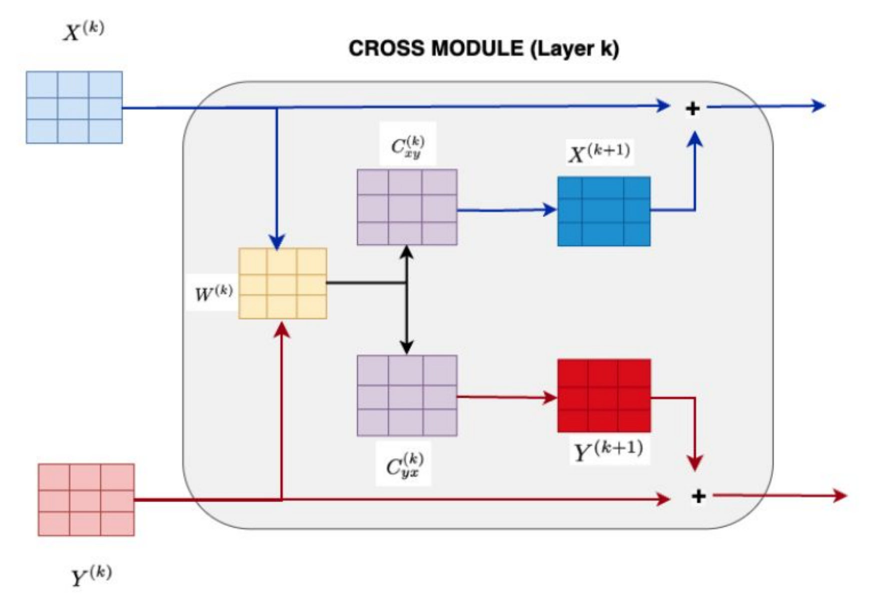
- ➔ Inputs: $X \in \mathcal{R}^{N \times d}, Y \in \mathcal{R}^{M \times d}$
- ➔ Self module: **2D convolutional layers**
- ➔ Cross module: **Cross-graph learning**

$$C_{xy}^{(k)}(i,j) = \exp\left(\frac{\left(X^{(k)^T} W^{(k)} Y^{(k)} \right)_{ij}}{\delta} \right) \in \mathbb{R}^{m \times n}$$

$$X^{(k+1)} = \left[C_{xy}^{(k)} Y^{(k)} \| X^{(k)} \right] \Theta_{xy}^{(k)}$$

$$Y^{(k+1)} = \left[C_{yx}^{(k)} X^{(k)} \| Y^{(k)} \right] \Theta_{yx}^{(k)}$$

Operation	Kernel Size	Input Channels	Output Channels
Pointwise Convolution	1×1	1	32
	Convolutional Blocks (3x)		
Depthwise Convolution	7×7	32	32
Layer Normalization	-	-	-
Pointwise Convolution	1×1	32	128
ReLU Activation	-	-	-
Pointwise Convolution	1×1	128	32



Computational performance

- ➔ Model parameters and size
- ➔ Average inference time (image pairs/sec)

Model	Parameters (M)	Model Size (MB)
SuperPoint	1.30	4.96
LightGlue	**11.85**	**45.21**
Student	**11.56**	**44.12**
LoFTR	28.35	108.13
SuperPoint + LightGlue	**13.15**	**50.17**
SuperPoint + Student	**12.86**	**49.08**

Model	Average Inference Speed (pairs/sec)
SuperPoint + LightGlue	**30.43**
SuperPoint + Student	**7.10**
LoFTR	10.23