FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

HOVER

# Image Matching Challenge 2019 — 2024

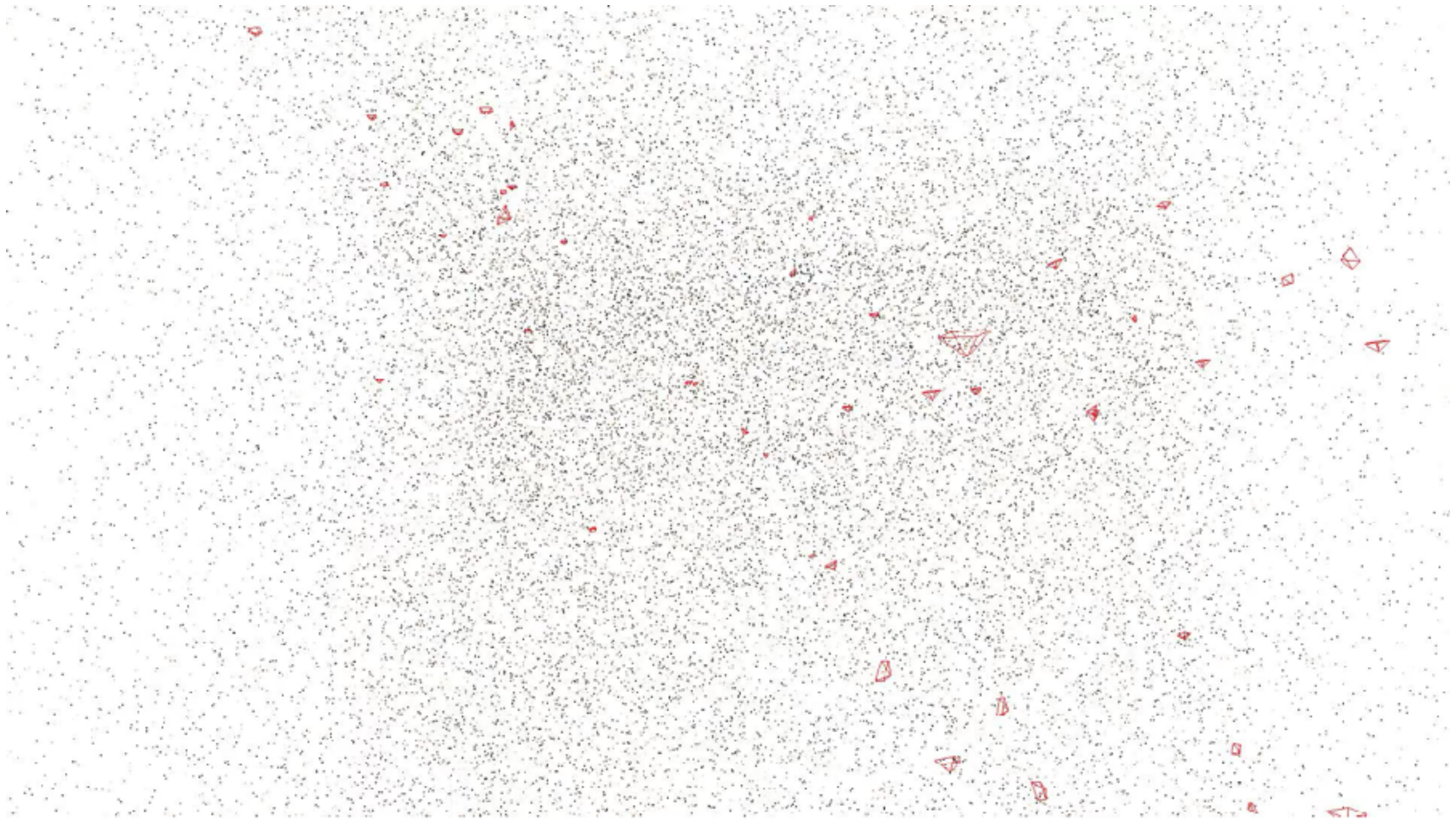
## How to move the goalposts: coming up with yet unsolved challenges for SfM

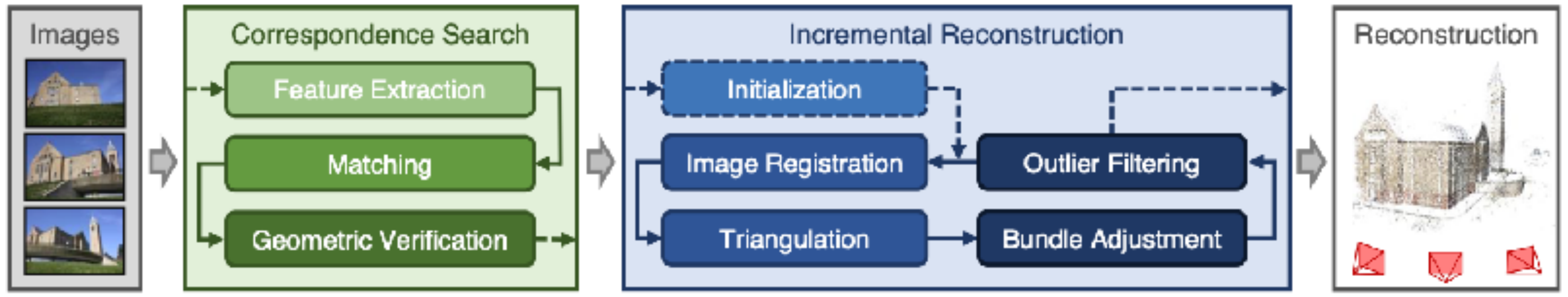
**Dmytro Mishkin, Faculty of Electrical Engineering, CTU in Prague / HOVER Inc.**

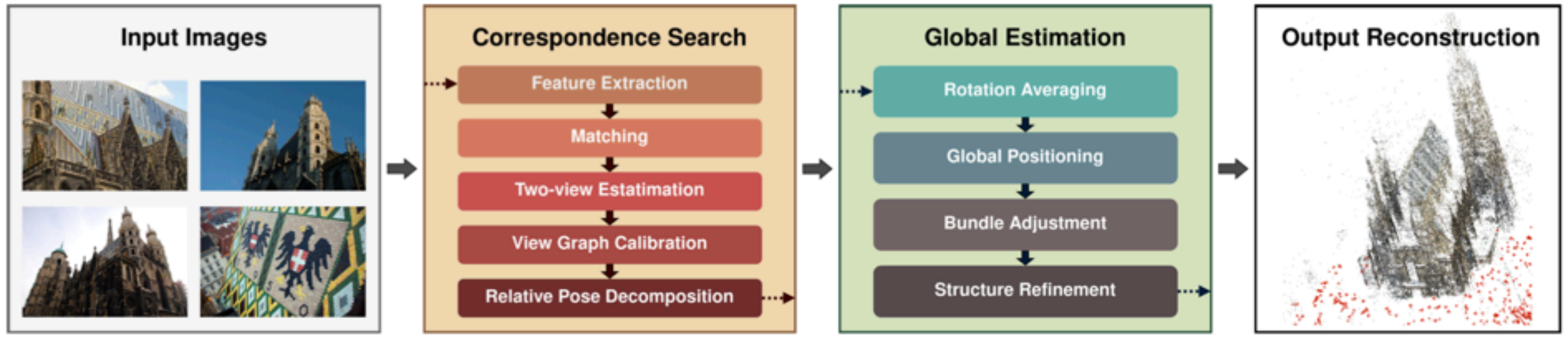
Presented research is supported by OP VVV funded project CZ.02.1.01/0.0/0.0/16.019/0000765 ``Research Center for Informatics''

# A slide about how cool SfM is

## We all know it, right?





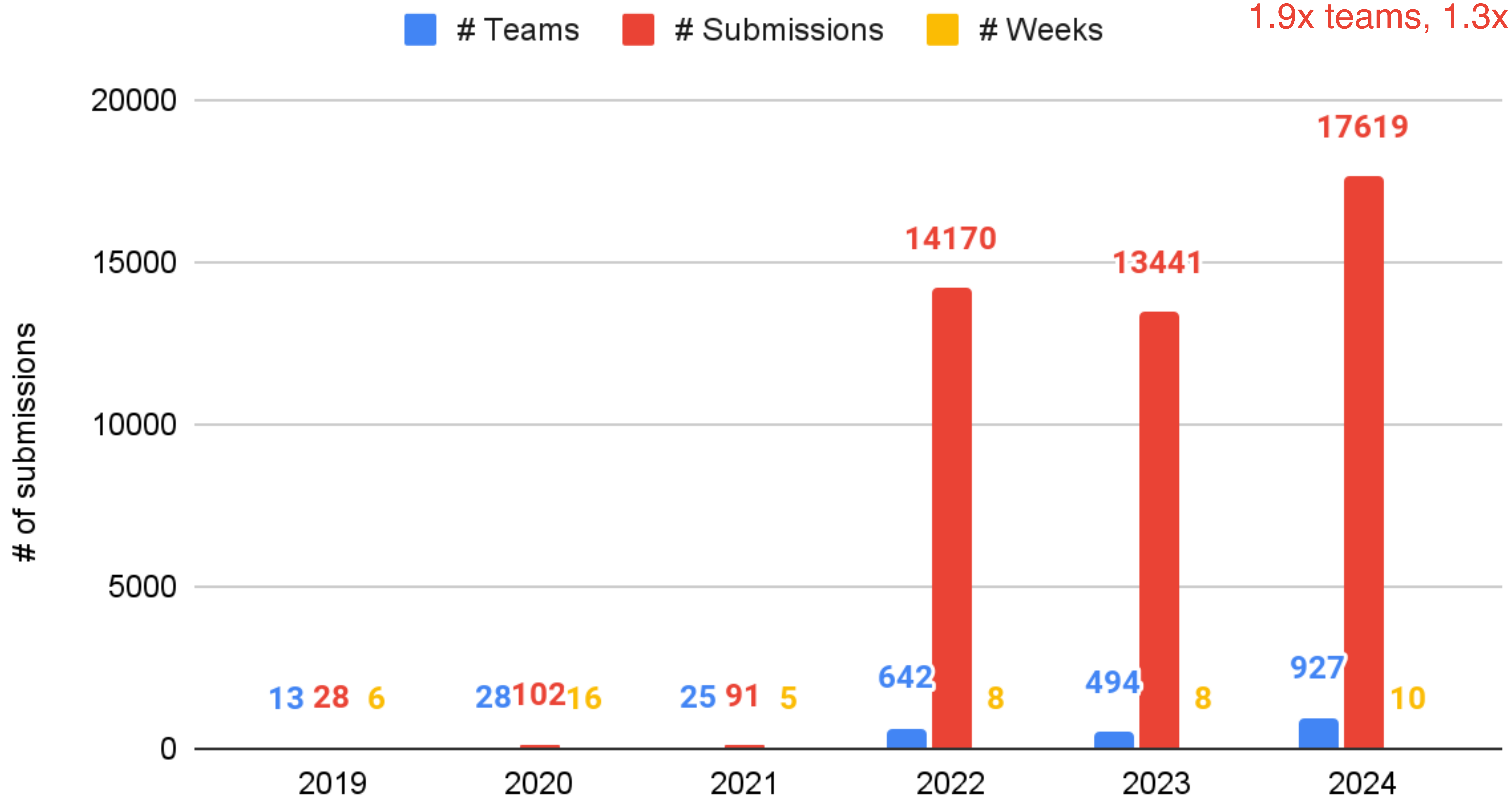




https://demuc.de/papers/schoenberger2016sfm.pdf

https://lpanaf.github.io/eccv24_glomap/

# A slide about how cool IMC is

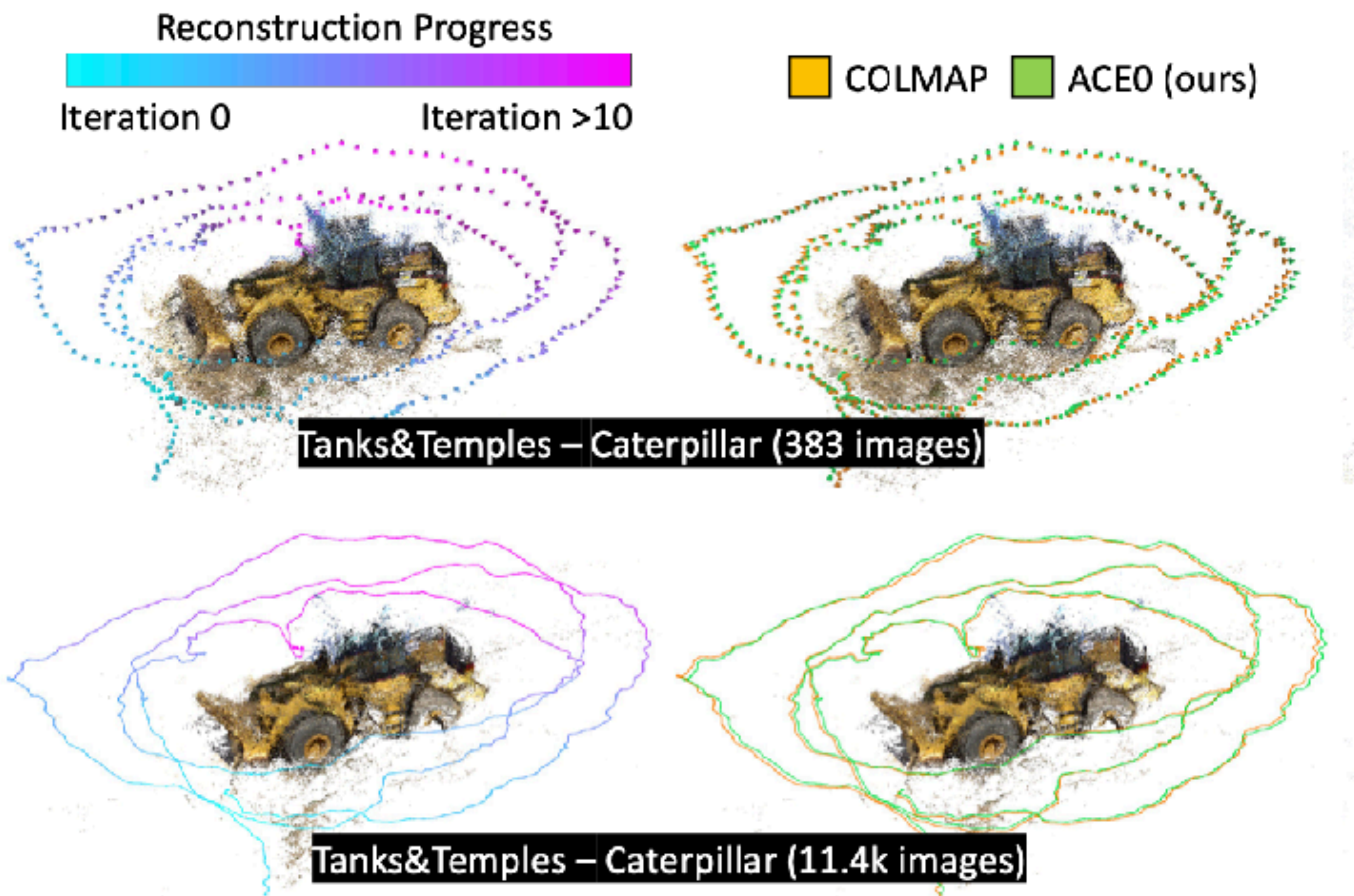
**2021 → 2022**
25x teams, 150x submissions
**2023 →2024**
1.9x teams, 1.3x submissions



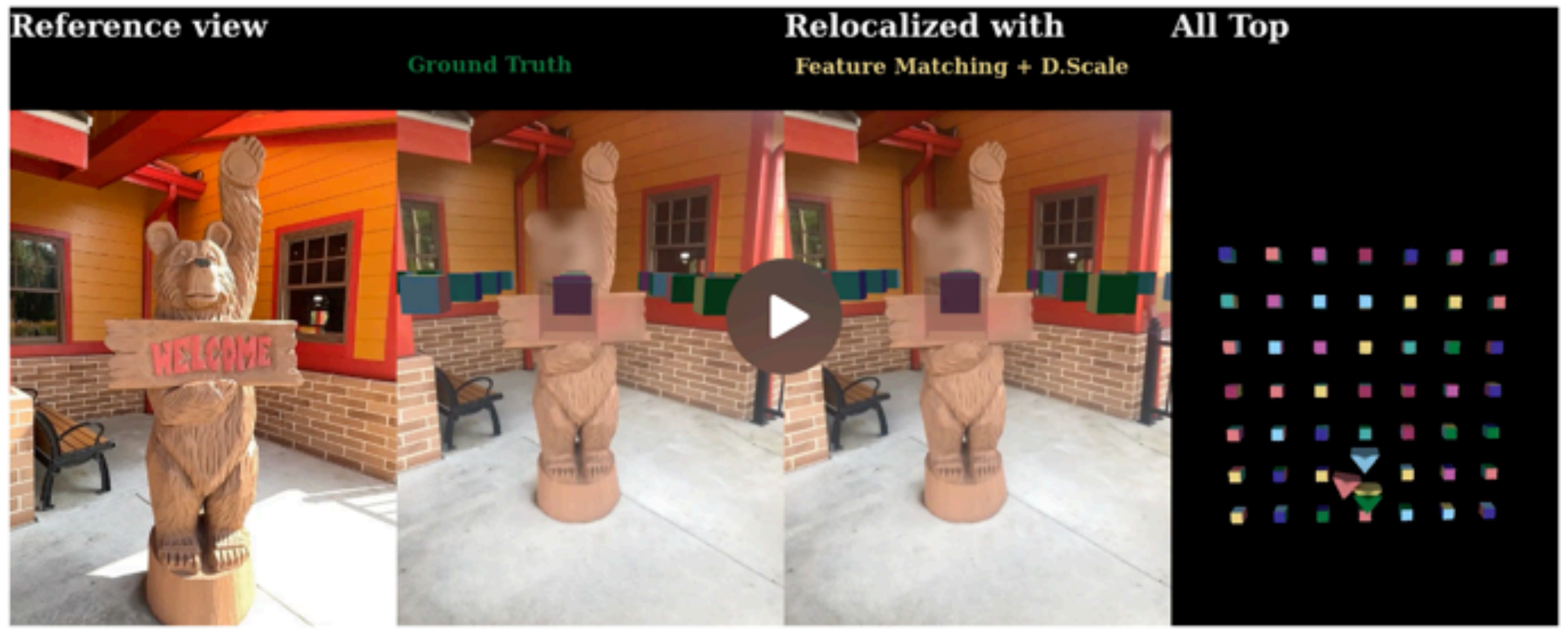
# How do we benchmark image matching in 2024?

## as a part of SfM typically

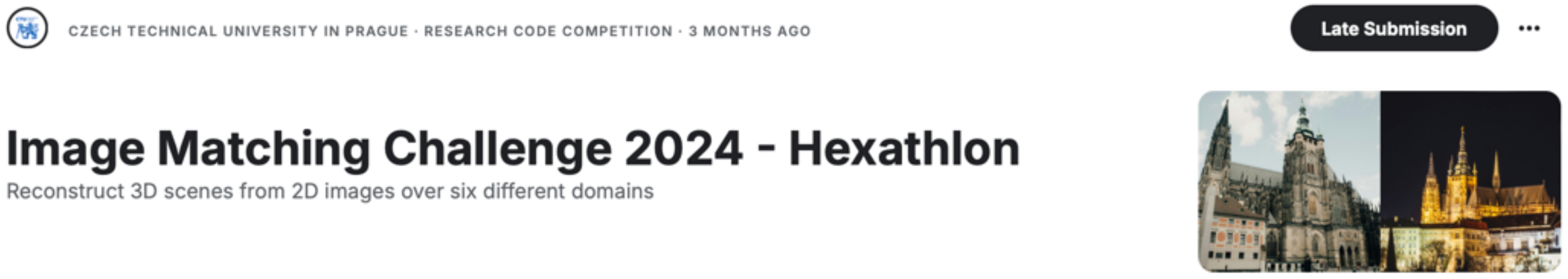
- Downstream metric:
  - Pose accuracy one way or another
    - Photo-consistency via NERF/GS
- Suitable for any method




https://github.com/nianticlabs/acezero




https://research.nianticlabs.com/mapfree-reloc-benchmark



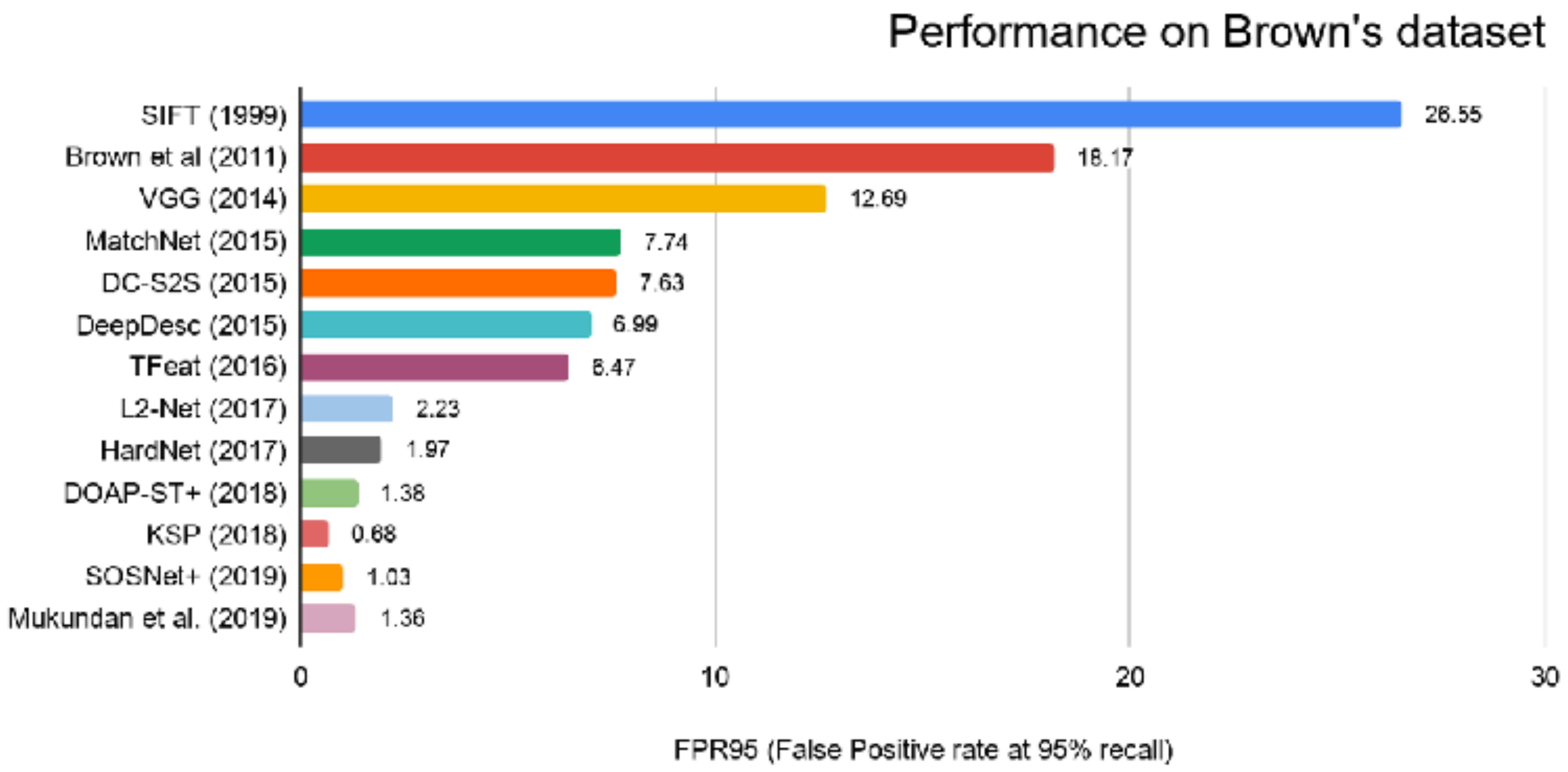

https://www.kaggle.com/competitions/image-matching-challenge-2024/leaderboard

https://www.visuallocalization.net/benchmark/

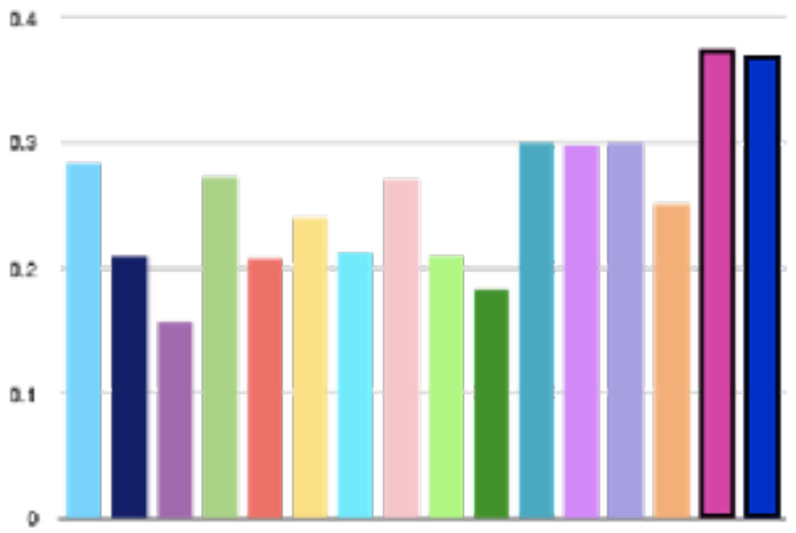
It was not always like this

# Before 2019:

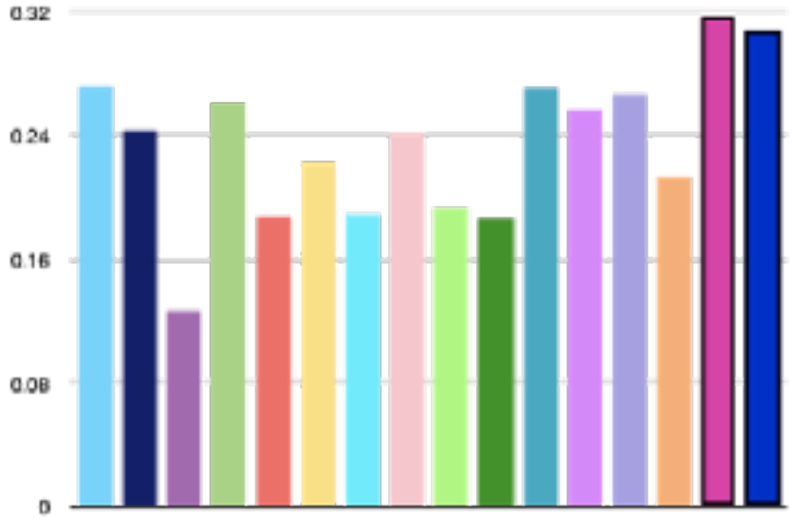
## Bunch of metrics per some component



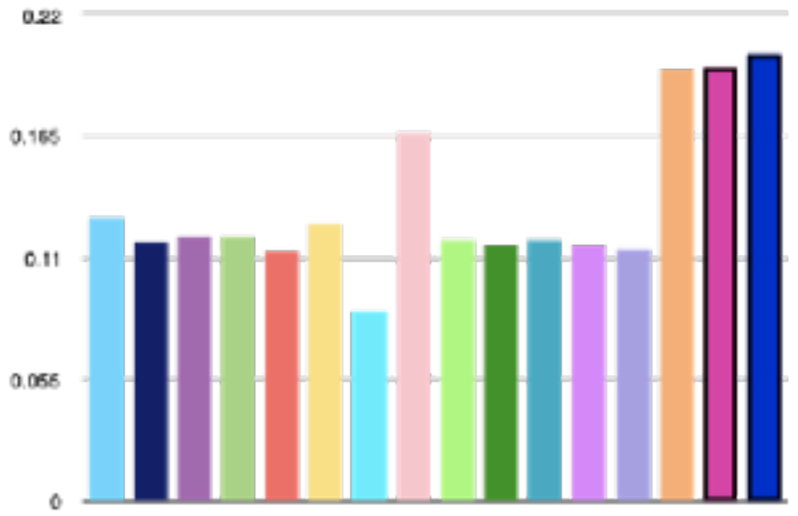

- Brown dataset FPR@95@ recall (who said we need 95% recall?)
- “Average matching score” on DTU/Stretcha
- HPatches — mean average precision for patch classification/retrieval/matching
- (used in HardNet) - mean average precision @ *image retrieval on Oxford 5k* with Bag-of-Words
- Schönberger et al., CVPR’17 — number of registered image and 3D points .
- RANSAC? What is RANSAC?



Avg. matching score on ‘Strecha’



Avg. matching score on ‘DTU’

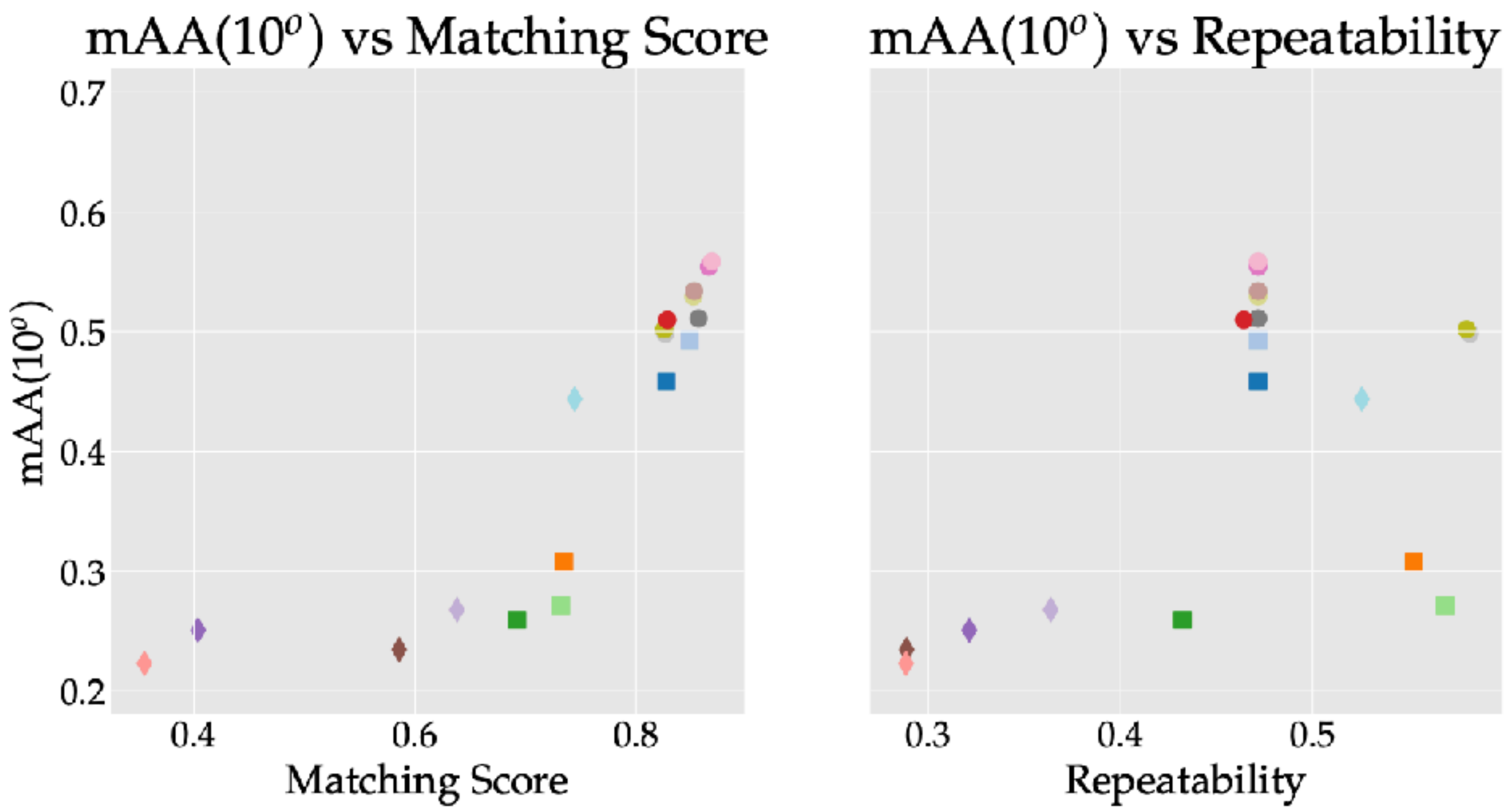


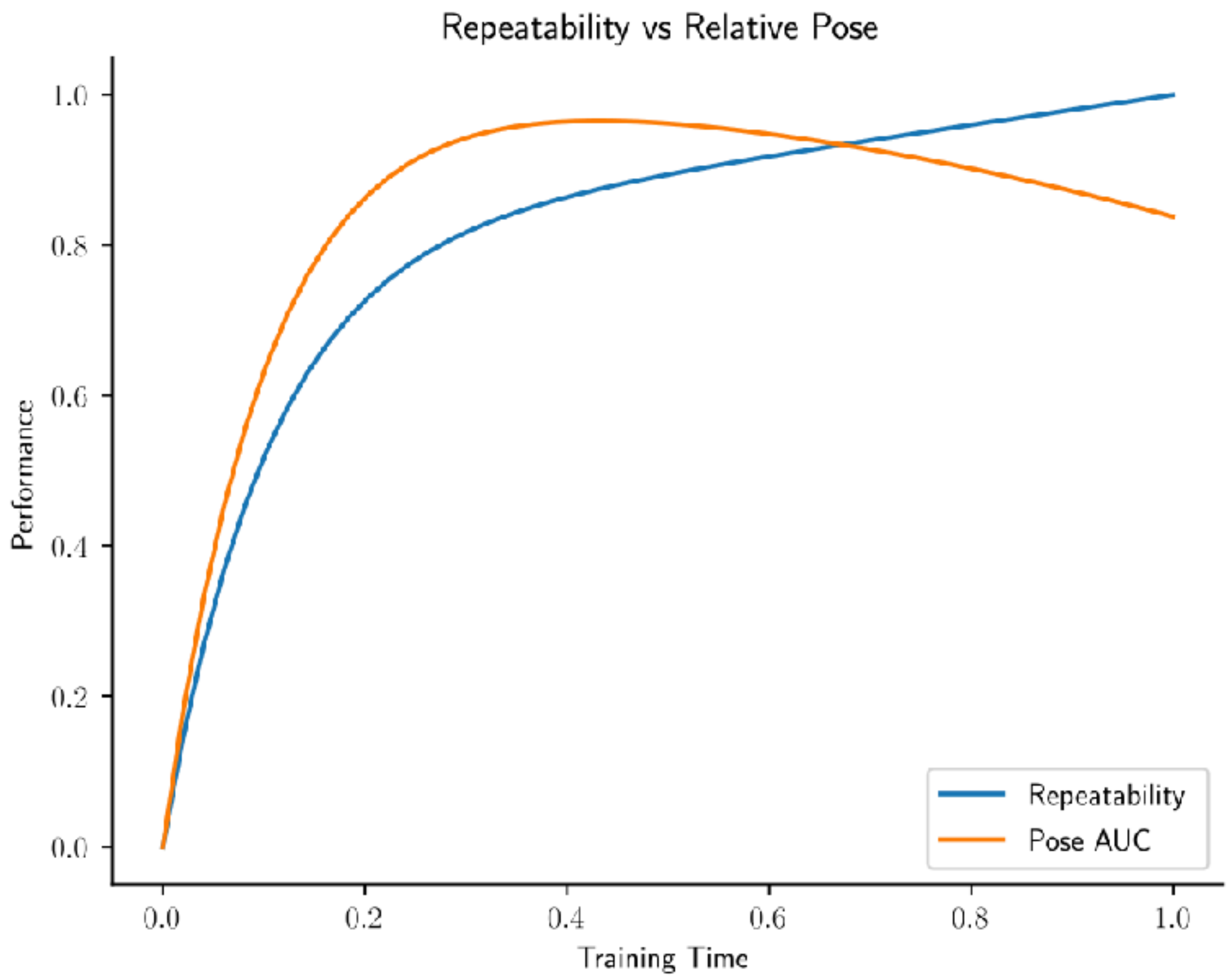
Avg. matching score on ‘Webcam’

# Why downstream metrics?

## Because per-component metrics do not predict the final outcome

Even for the single method — as shown in DeDoDe v2 paper




Matching Across Wide Baselines: From Paper to Practice

DeDoDe v2: Analyzing and Improving the DeDoDe Keypoint Detector

# Image Matching Challenge 2019


**Vassileios Balntas** Scape Technologies | **Vincent Lepetit** U. Bordeaux | **Johannes Schönberger** Microsoft | **Eduard Trulls** Google | **Kwang Moo Yi** U. Victoria


Still available at https://image-matching-workshop.github.io/leaderboard/

# Image Matching Challenge 2019: first attempt

✓ Phototourism data: viewpoint, sensors, illumination, motion blur, occlusions, etc

✓ Large-scale: ~30k images

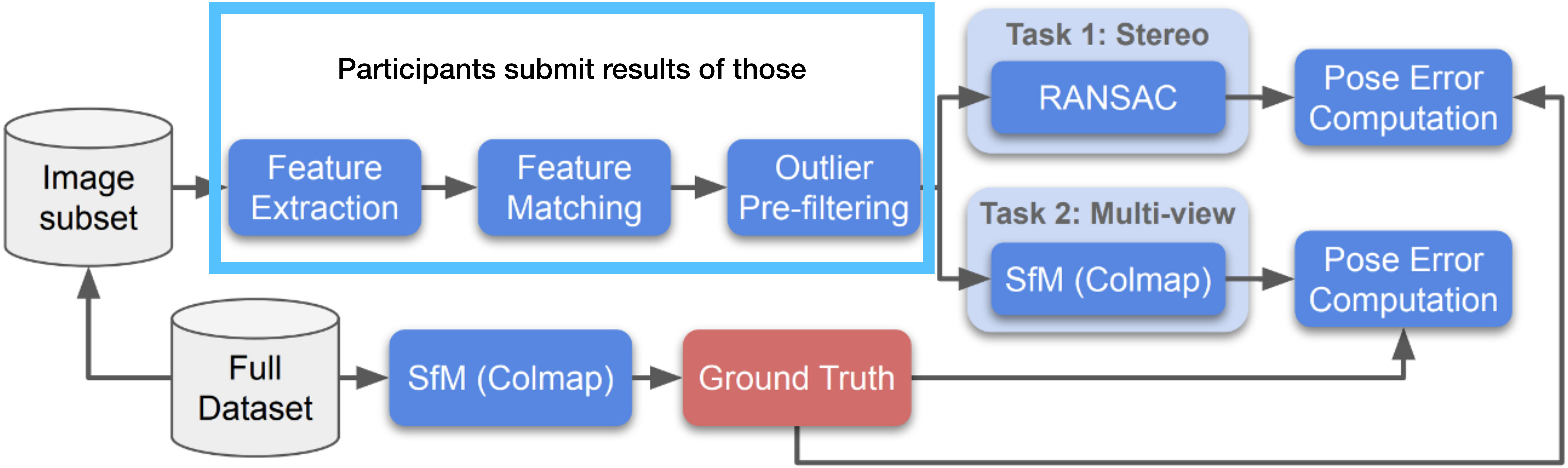
✓ Downstream metric: camera pose accuracy

✓ 2 tasks: SfM and stereo

- "Quasi" ground truth data is generated by performing SfM with COLMAP with all images.
- Assumption: Images registered in COLMAP are accurate given enough images ~1000s

# IMC benchmark idea

## 2019 was more “Local feature quality evaluation”



# Metric: what is mAA@10°?

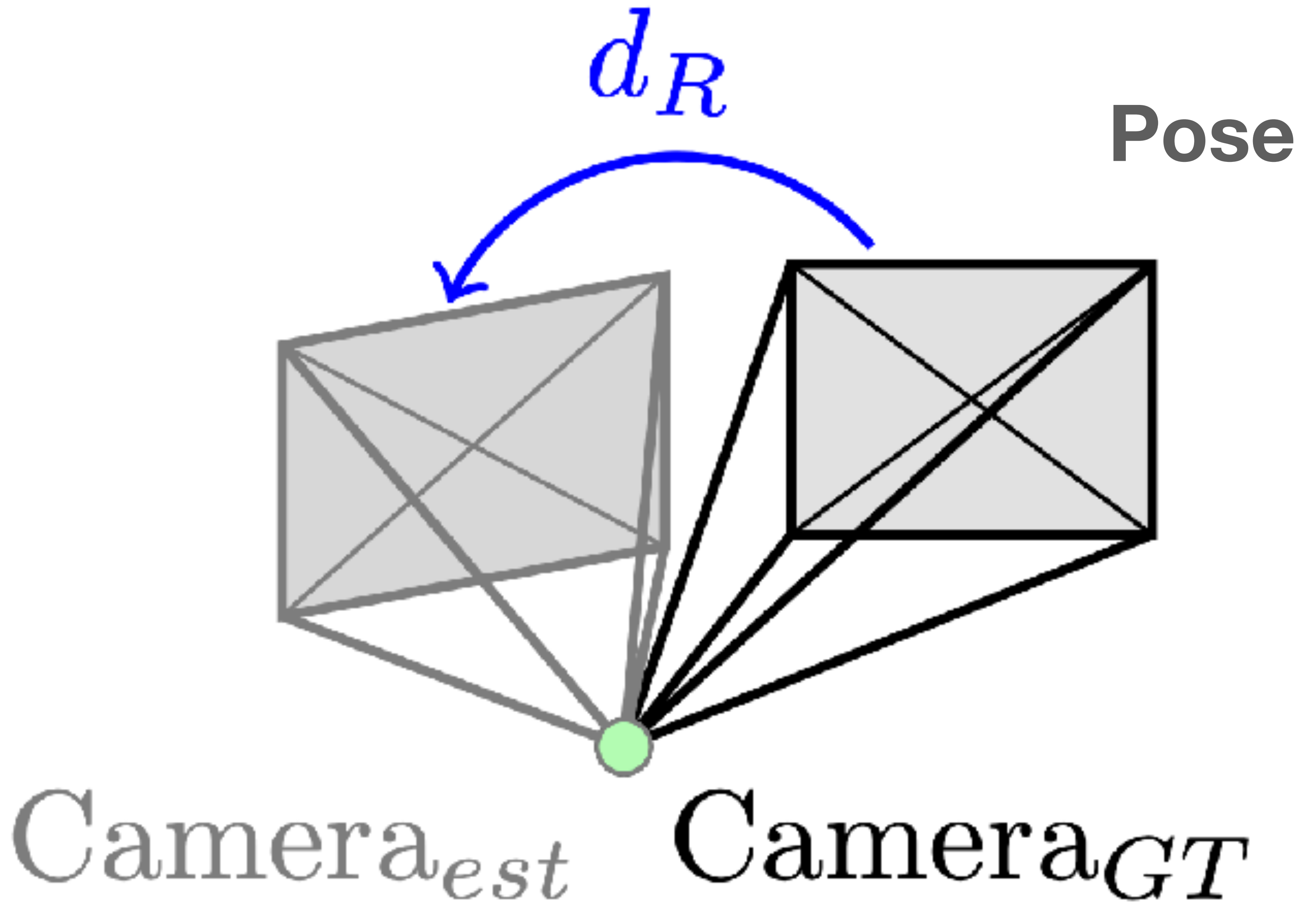
The mean Average Accuracy is a robust aggregate of the pose error of all image pairs and scenes.
If an error is below a threshold, the camera pair is considered correct,

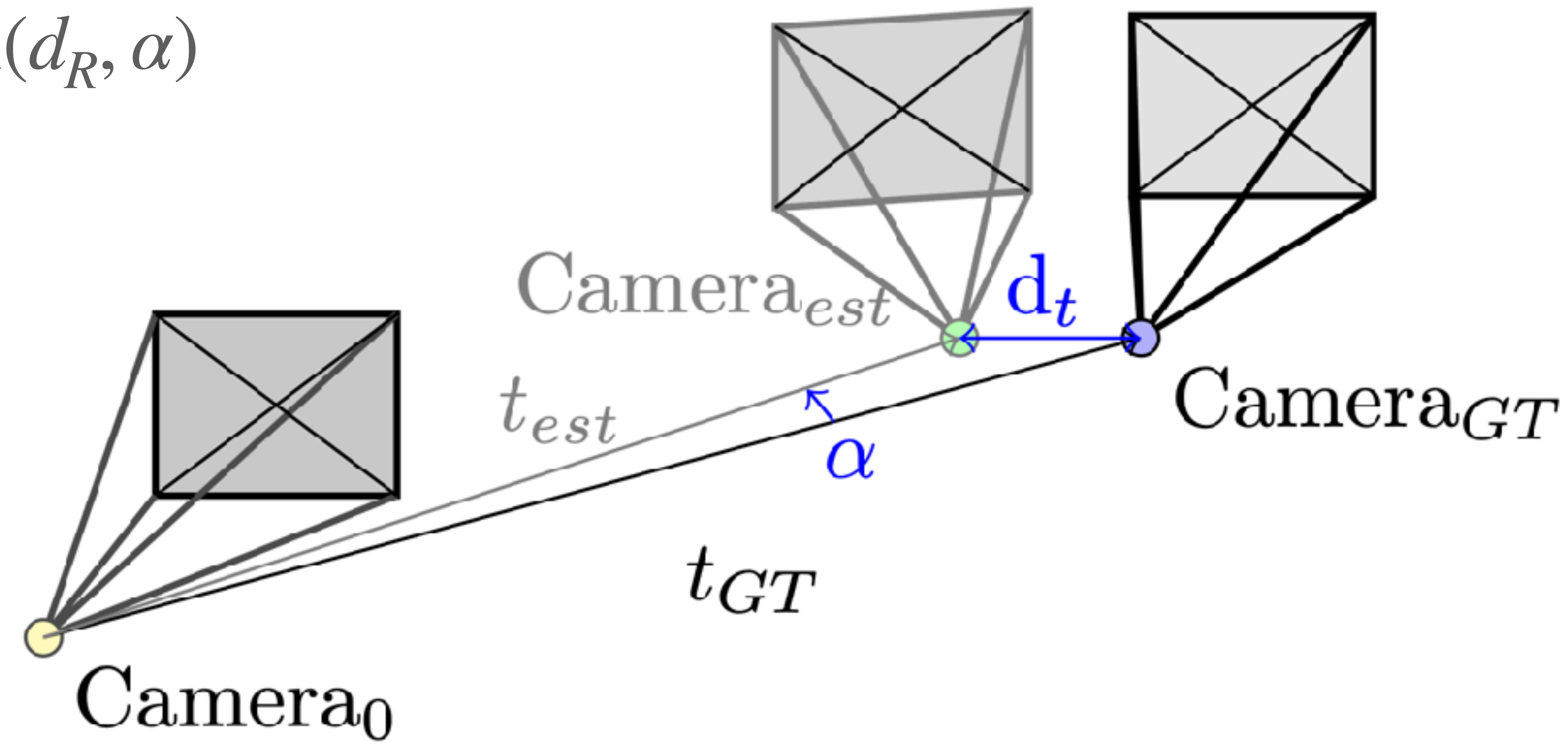
The pose error has two components: **rotation** and **translation**.
Because the scale of the estimate and ground truth are both unknown, we rely exclusively on angles

**Rotation error** $d_R$ :
angle, which aligns GT and estimated camera.

**Translation *angular* error** $\alpha$ :
angle between direction to GT and estimated camera.

**Pose Error =** $\max(d_R, \alpha)$



There were some problems as well

# Image Matching Challenge 2019: first attempt

## Submission: tentative correspondences only

- Stereo best mAP15: 8%
- SfM best mAP15: 73%

Why? Seems that something is wrong?

Yes! Under evaluation framework, stereo pose estimation was done badly:

- RANSAC is not tuned
- No Lowe's ratio test for SIFT

### [P1] Phototourism dataset — Stereo task

Performance in stereo matching, averaged over all the test sequences.

- Click here for a breakdown by sequence

Show 10 entries Search:

Stereo — averaged over all sequences

| Method | Date | Type | #kp | MS | $mAP^{5°}$ | $mAP^{10°}$ | $mAP^{15°}$ | $mAP^{20°}$ | $mAP^{25°}$ |
|---|---|---|---|---|---|---|---|---|---|
| SIFT + ContextDesc + Inlier Classification V2 kp:8000, match:custom | 19-05-28 | F/M | 7515.2 | 0.3533 | 0.0016 | 0.0217 | 0.0823 | 0.1818 | 0.2963 |

### [P2] Phototourism dataset — Multi-view task

Performance in SfM reconstruction, averaged over all the test sequences.

- Click here for a breakdown by sequence
- Click here for a breakdown by subset size

Show 10 entries Search:

MVS — averaged over all sequences

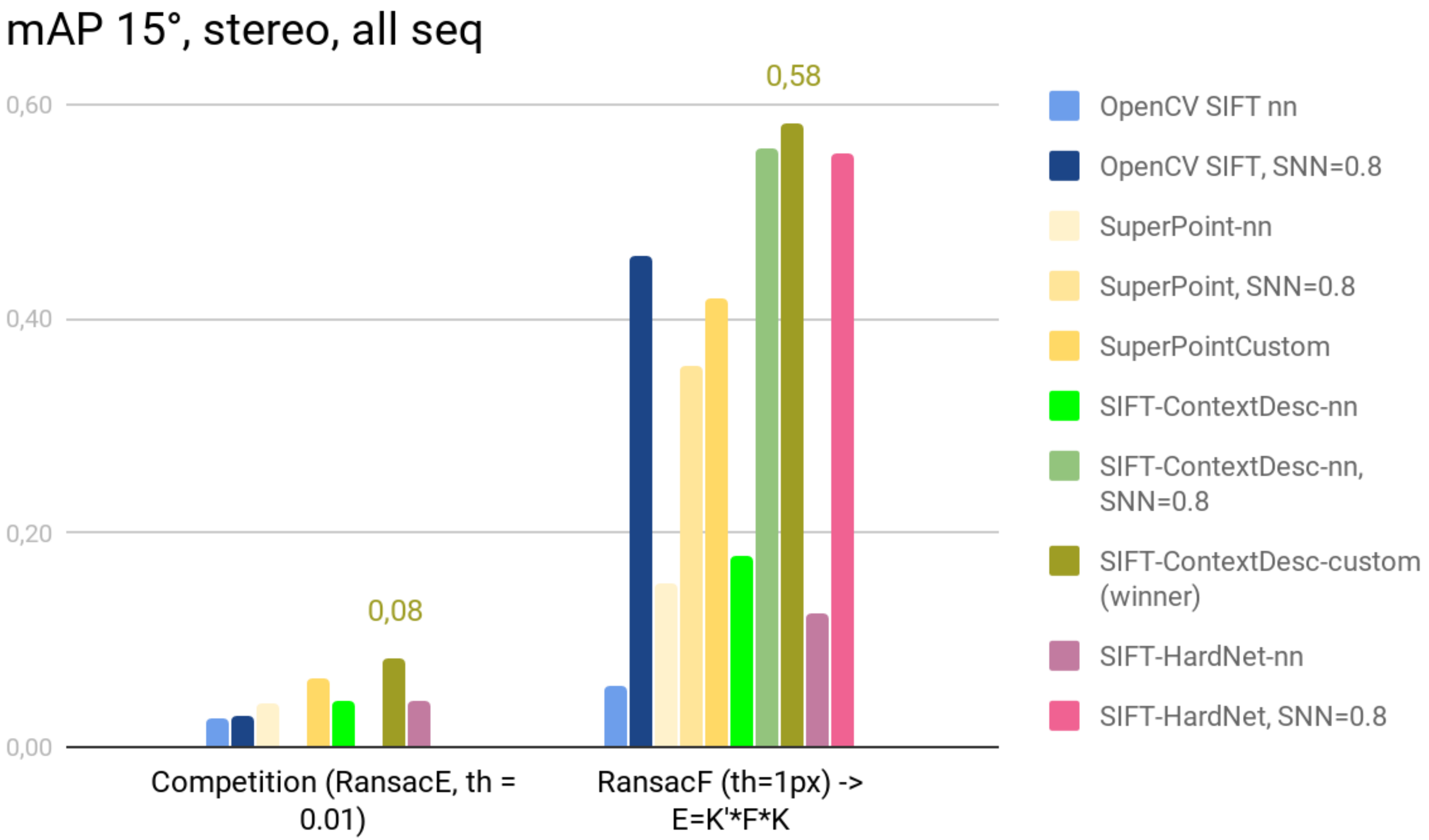
| Method | Date | Type | Ims (%) | #Pts | SR | TL | $mAP^{5°}$ | $mAP^{10°}$ | $mAP^{15°}$ | $mAP^{20°}$ | $mAP^{25°}$ | ATE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SIFT + ContextDesc + Inlier Classification V2 kp:8000, match:custom | 19-05-28 | F/M | 50.5 | 6125.0 | 97.5 | 3.44 | 0.5755 | 0.6030 | 0.7309 | 0.7750 | 0.8006 | |

https://image-matching-workshop.github.io/leaderboard/

# Image Matching Challenge 2019: first attempt

## Results after tuning RANSAC and SNN-ratio

- Results change drastically after tuning
- SIFT is strong
- SIFT detector (DoG) + learned patch descriptor +outlier rejection is a winner
- Winner didn't change after RANSAC tuning, but its margin did



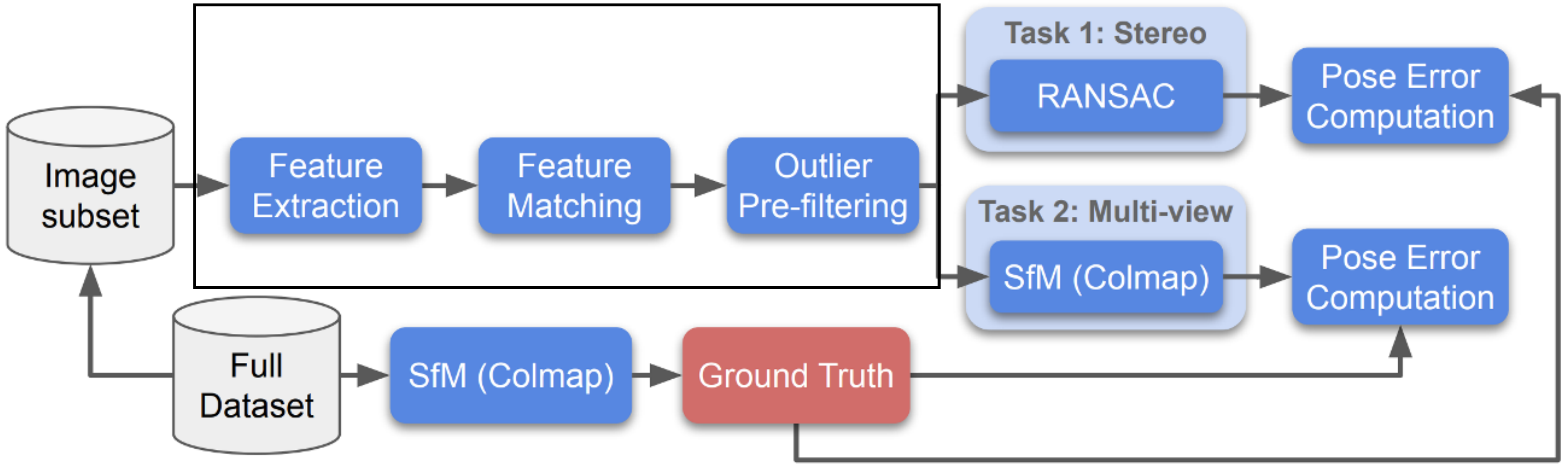
# Image Matching Challenge 2020

Vassileios Balntas, Scape Technologies
Vincent Lepetit, University of Bordeaux
Jiri Matas, Czech Technical University
Dmytro Mishkin, Czech Technical University
Johannes Schönberger, Microsoft
Eduard Trulls, Google
Kwang Moo Yi, University of Victoria
Yuhe Jin, University of British Columbia

Still available at https://www.cs.ubc.ca/research/image-matching-challenge/2020/leaderboard/

# IMC 2020: evaluation convergence

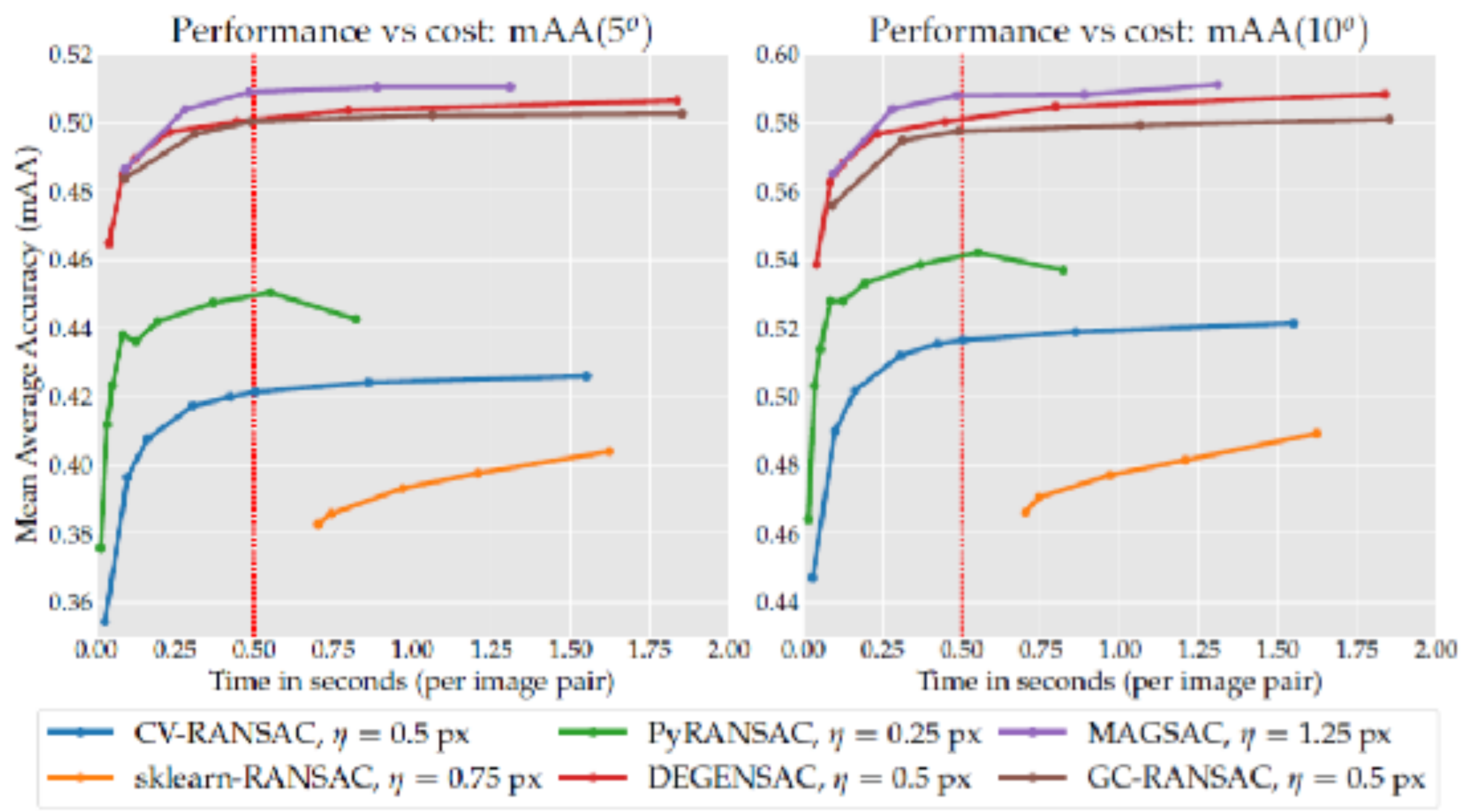
- Empirical evidence that you can create "ground truth" with 1000s of SfM data, which is not biased towards used local features
- Provide the codebase https://github.com/ubc-vision/image-matching-benchmark/
  - Also baselines repo https://github.com/ubc-vision/image-matching-benchmark-baselines
- Establish RANSAC-tuning protocol
- You give features & matches
  → we do reconstruction & results
- Publish a paper

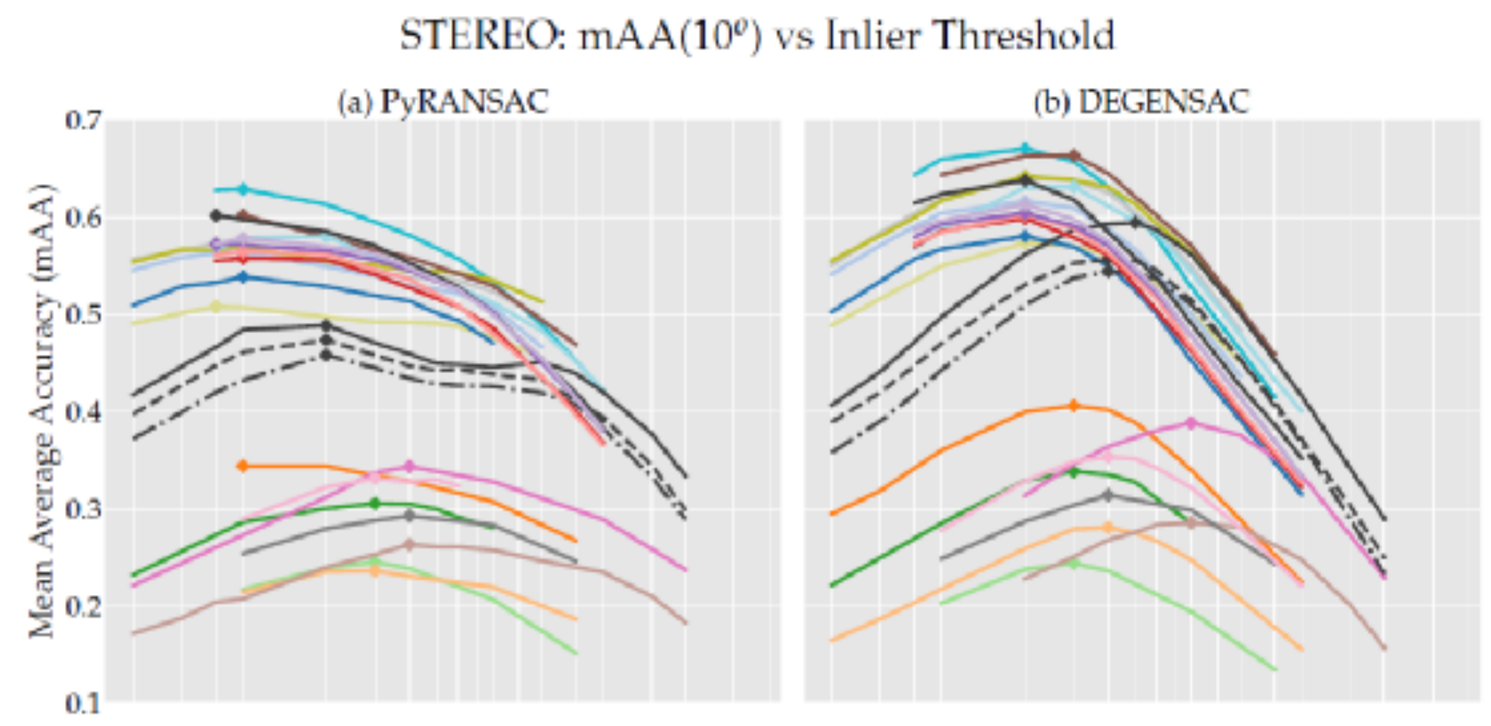



https://arxiv.org/abs/2003.01587

# IMC 2020 paper: messages to community

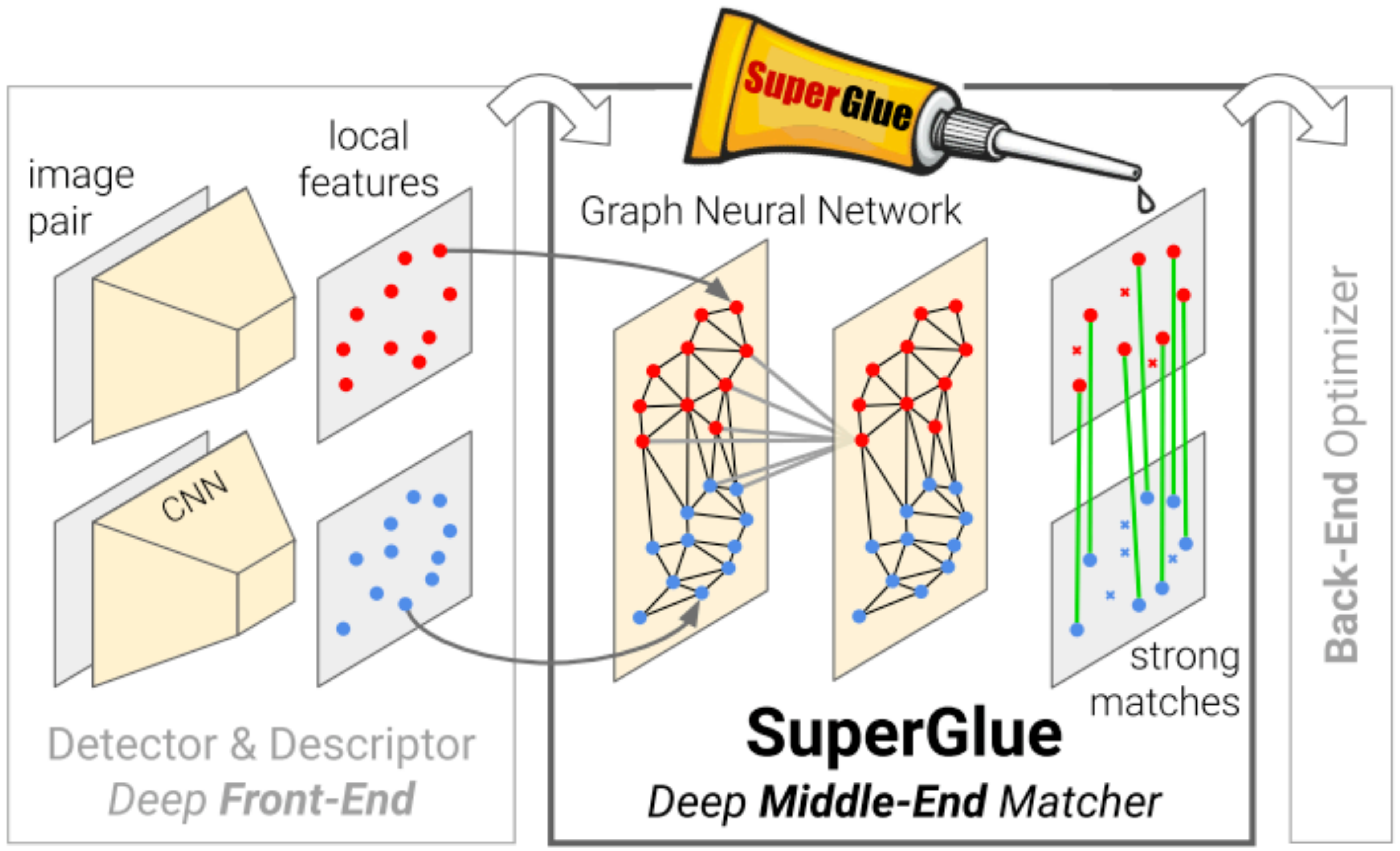
- RANSAC implementation matters
  - Use PoseLib, USAC_MAGSAC or pydegensac
- You have to tune RANSAC threshold
- Correspondence filtering matters (Lowe's SNN ratio!)
- SIFT is still great
- All components should be tuned together




https://arxiv.org/abs/2003.01587

# Findings from IMC 2020




- SuperGlue dominated the field
  - Not only in 2020, but until LightGlue in 2023
- Tons of features (8k) with decent outlier rejection are good enough for PhotoTourism (DoG + HardNet + AdaLAM / OANet)
- DISK appeared!

https://www.cs.ubc.ca/research/image-matching-challenge/2020/leaderboard/

SuperGlue: Learning Feature Matching with Graph Neural Networks

There were some problems as well

# IMC 2020 issues

- Evaluation is very compute-intensive — up to a day of compute for tuning and getting results for 8k submission
  - **100 CPU-years** per 2020 competition on Compute Canada
- Hard (impossible?) to evaluate detectorless (optical flow) methods.
- Even worse — pose regression methods
- PhotoTourism is a limited domain, and looks like saturated
- People don’t like to not having access to GT (sorry, not changing that)

https://arxiv.org/abs/2003.01587

# Image Matching Challenge 2021

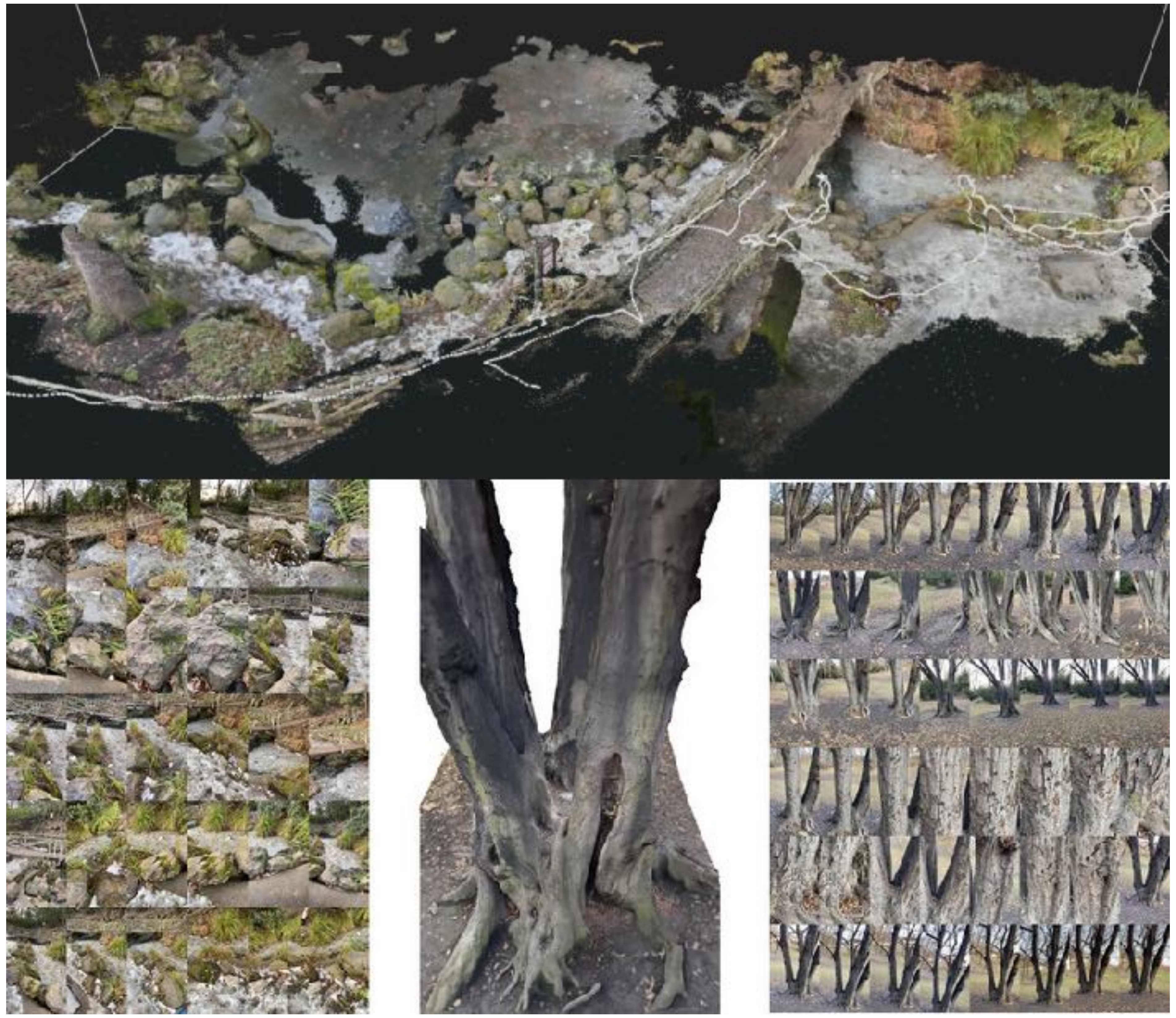
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Vassileios Balntas | Vincent Lepetit | Jiri Matas | Dmytro Mishkin | Johannes Schönberger | Eduard Trulls | Kwang Moo Yi | **Yuhe Jin** |
| Scape Technologies | University of Bordeaux | Czech Technical University | Czech Technical University | Microsoft | Google | University of Victoria | University of British Columbia |

Still available at https://www.cs.ubc.ca/research/image-matching-challenge/2021/leaderboard/

# IMC 2021

## Dataset expansion + tutorials

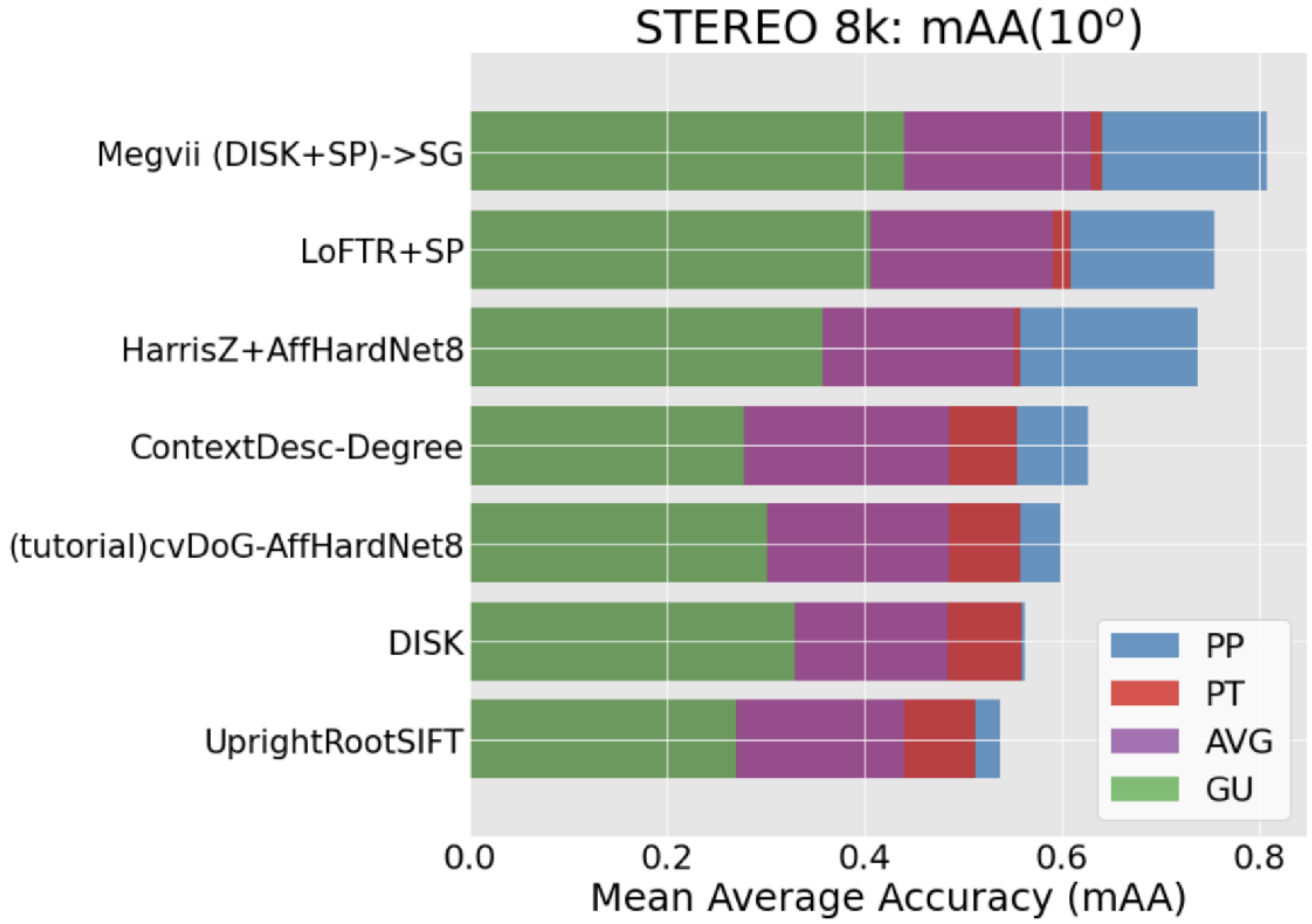
- Google Urban dataset — lower quality mobile photos, close-ups, etc
- PragueParks — more “nature” scenes
- Tutorials [features], [matchers]



Mountain View

Bangkok

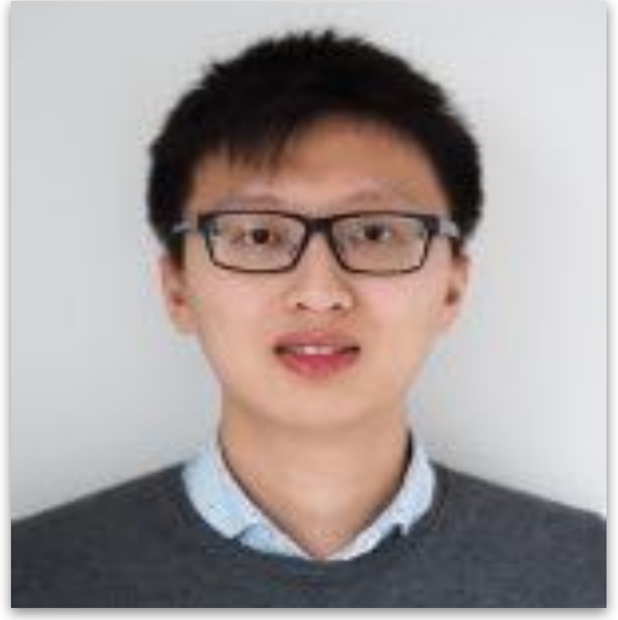
# New data better shows difference between methods




Methods, that perform the same on the PhotoTourism (red), perform very different on other datasets GoogleUrban and PragueParks

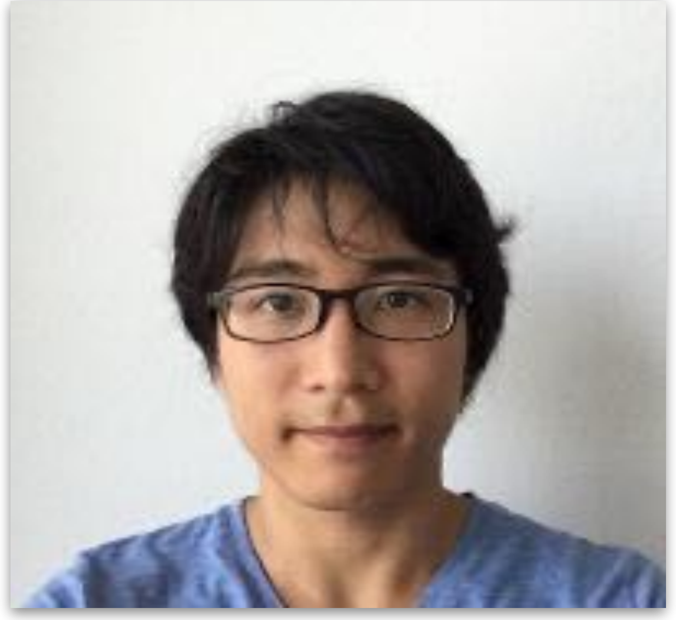
People managed to add LoFTR, by “snapping” to closest SuperPoints.

# Image Matching Challenge 2022

**Yuhe Jin**
Univ. British Columbia

**Eduard Trulls**
Google



**Kwang Moo Yi**
Univ. British Columbia

**Jiri Matas**
CTU Prague

**Dmytro Mishkin**
CTU Prague/HOVER Inc.


Still available at https://www.kaggle.com/competitions/image-matching-challenge-2022

# IMC2022: Before the competition

We were looking for:

- allowing dense (detector-less) methods
- fully hidden test set — Google Urban dataset was super hard to release (and we had to delete it since then)
- safe way to run docker images w/o tons of infra work

Kaggle was satisfying all the conditions, and Eduard Trulls worked with Kaggle to make it happen.

Huge Kaggle community as bonus

# IMC2022: Before the competition

Drawbacks:

- installing colmap was infeasible —stereo only.
- Metrics has to be implemented by us in C# (not anymore since 2024)
- Training set was phototourism, but test set was Google Urban-like.

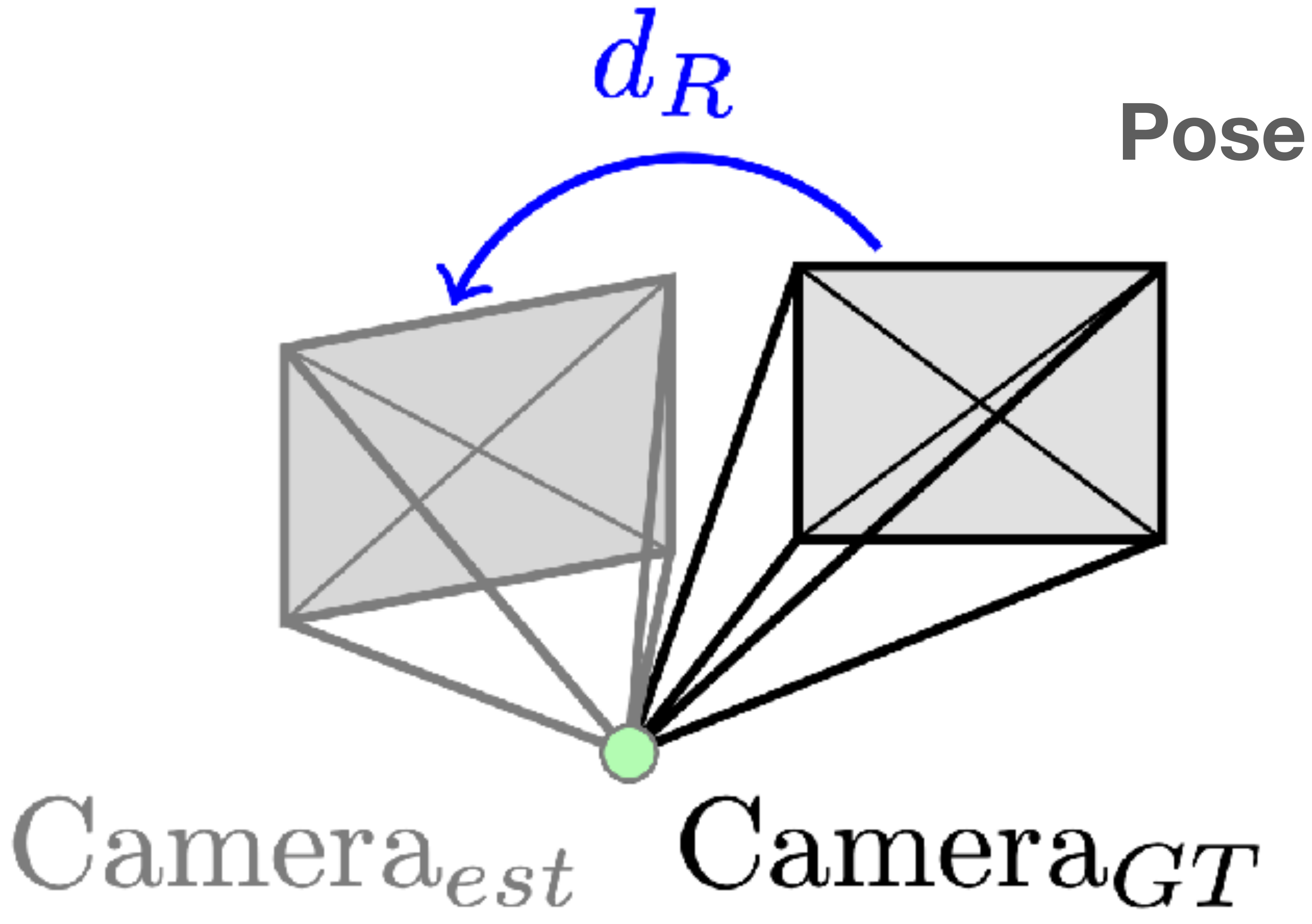
# 2022 Metric change: semi-metric translation

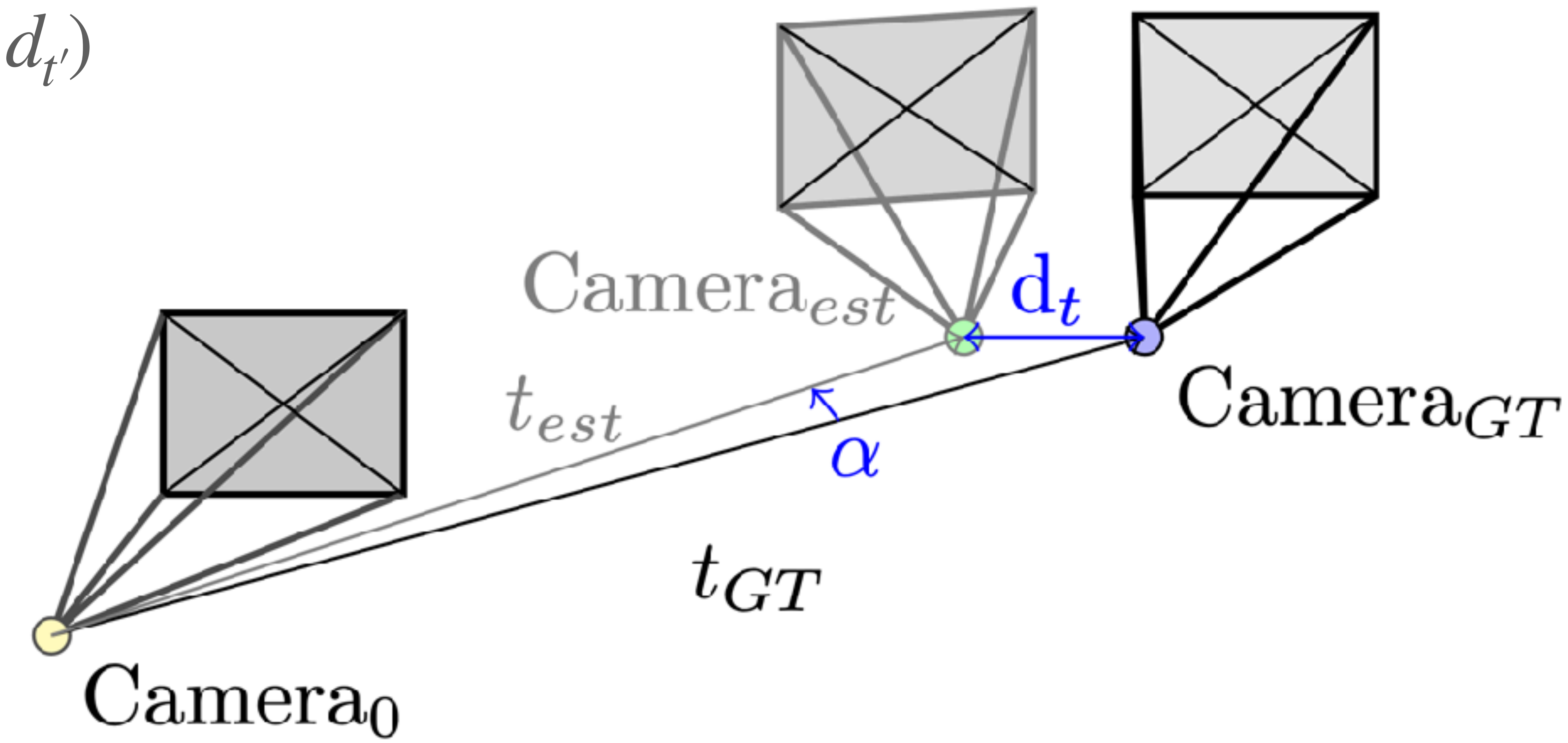
Now we annotated the scale for all our datasets, so we know ground truth translation $t_{GT}$ in meters.
However, we still cannot estimate true translation error $d_t$, because we don't have scale for submission.
So we "grant GT scale" to the submission and calculate $d_{t'} = |\, t_{GT} - t_{est}\frac{|t_{GT}|}{|t_{est}|} \,|$

**Rotation error** $d_R$ :
angle, which aligns GT and estimated camera.

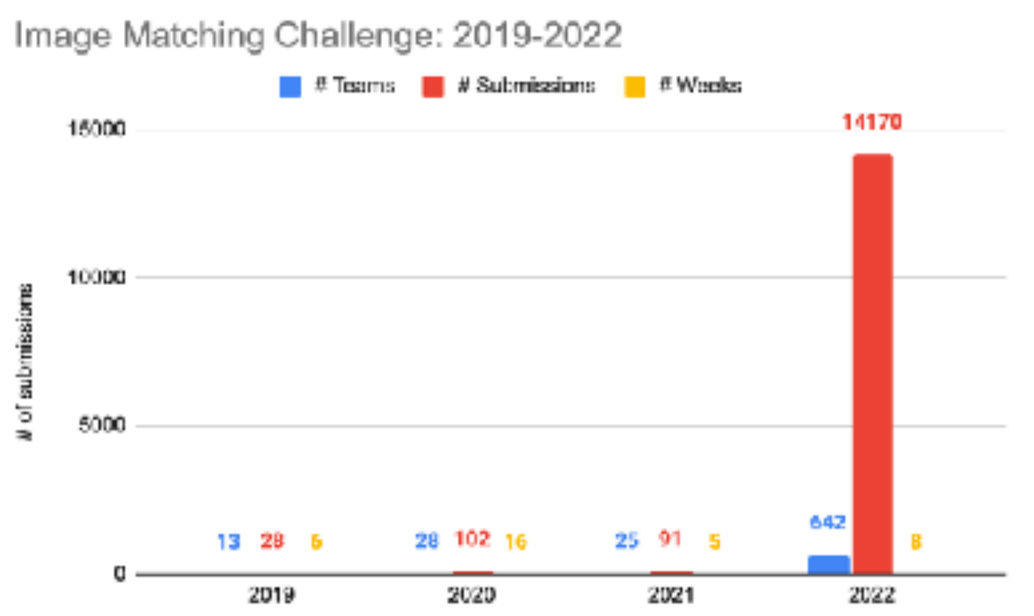
**Translation *semi-metric* error** $d_{t'}$ :
distance estimate between GT and estimated camera

**Pose Error =** $(d_R, d_{t'})$



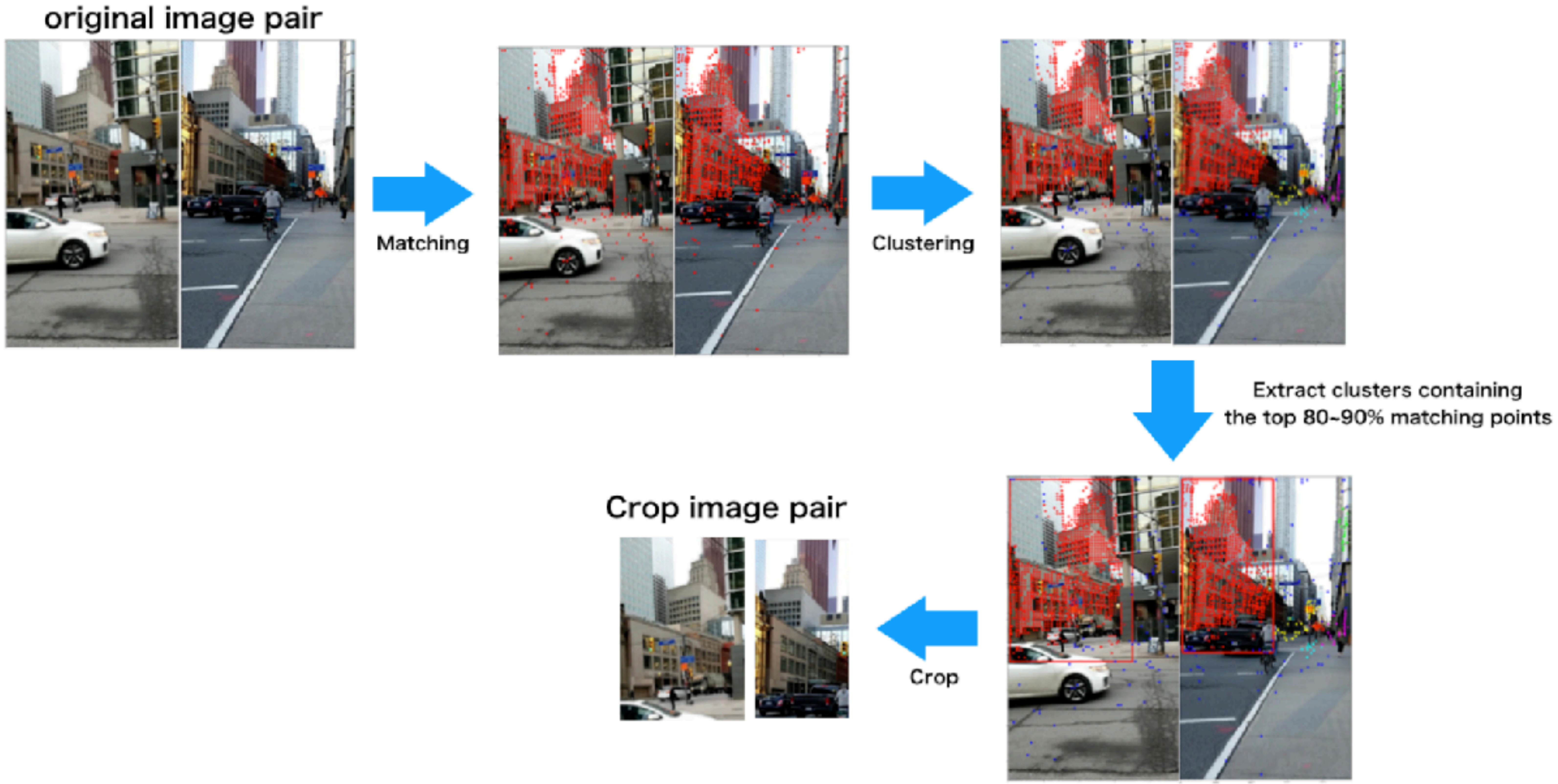



# IMC2022: Results




## New component in the image matching pipeline?

- zoom-in and refine are keys for stereo matching
- people tuned the baselines to extreme, which is good
- 25x more teams, 150x more submission
- LoFTR + SuperGlue
- Does it transfer to the SfM?




Papers with after-IMC22 ideas:

- [MKPC], [ASpanFormer]

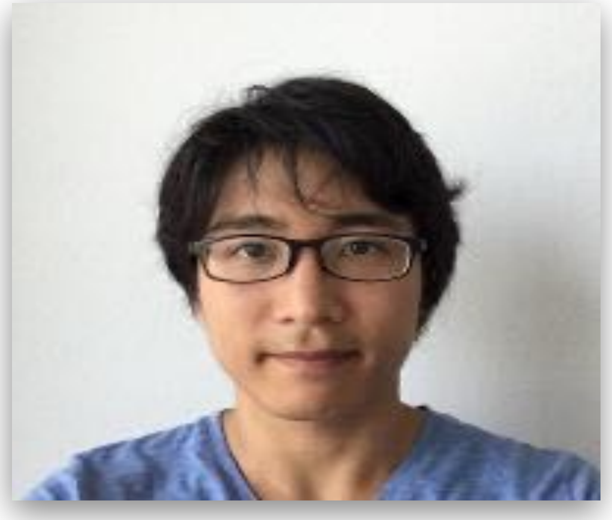
https://ducha-aiki.github.io/wide-baseline-stereo-blog/2022/07/05/IMC2022-Recap.html

https://www.kaggle.com/competitions/image-matching-challenge-2022/discussion/329131

**Fabio Bellavia**
Univ. Palermo

**Weiwei Sun**
U. British Columbia

**Eduard Trulls**
Google

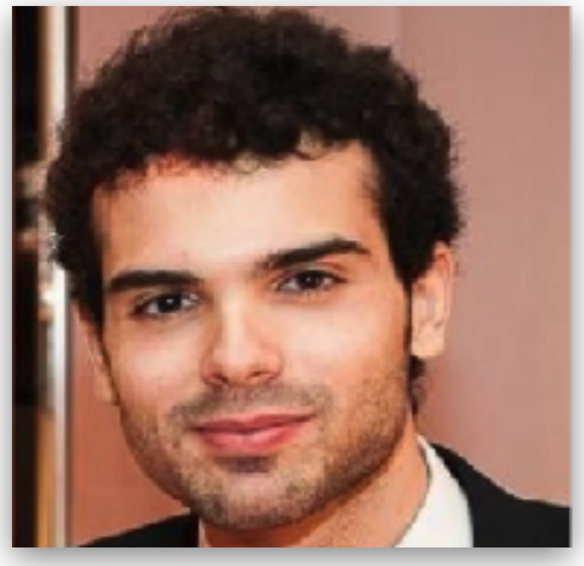


**Kwang Moo Yi**
U. British Columbia

# Image Matching Challenge 2023

**Jiri Matas**
CTU Prague

**Dmytro Mishkin**
CTU Prague/HOVER Inc.



**Luca Morelli**
Univ. Trento/BFK

**Fabio Remondino**
Bruno Kessler Foundation

Still available at https://www.kaggle.com/competitions/image-matching-challenge-2023

# IMC2023

## 2021 and 2022, but better

- SfM track: installed pycolmap and kornia to kaggle
- More datasets!
  - Urban day/night
  - Haiper NERF-like capture
  - Heritage: also with UAV
- $50k minimal prize requirement from Kaggle
  - Thanks to Haiper and Google for sponsorship

# IMC2023

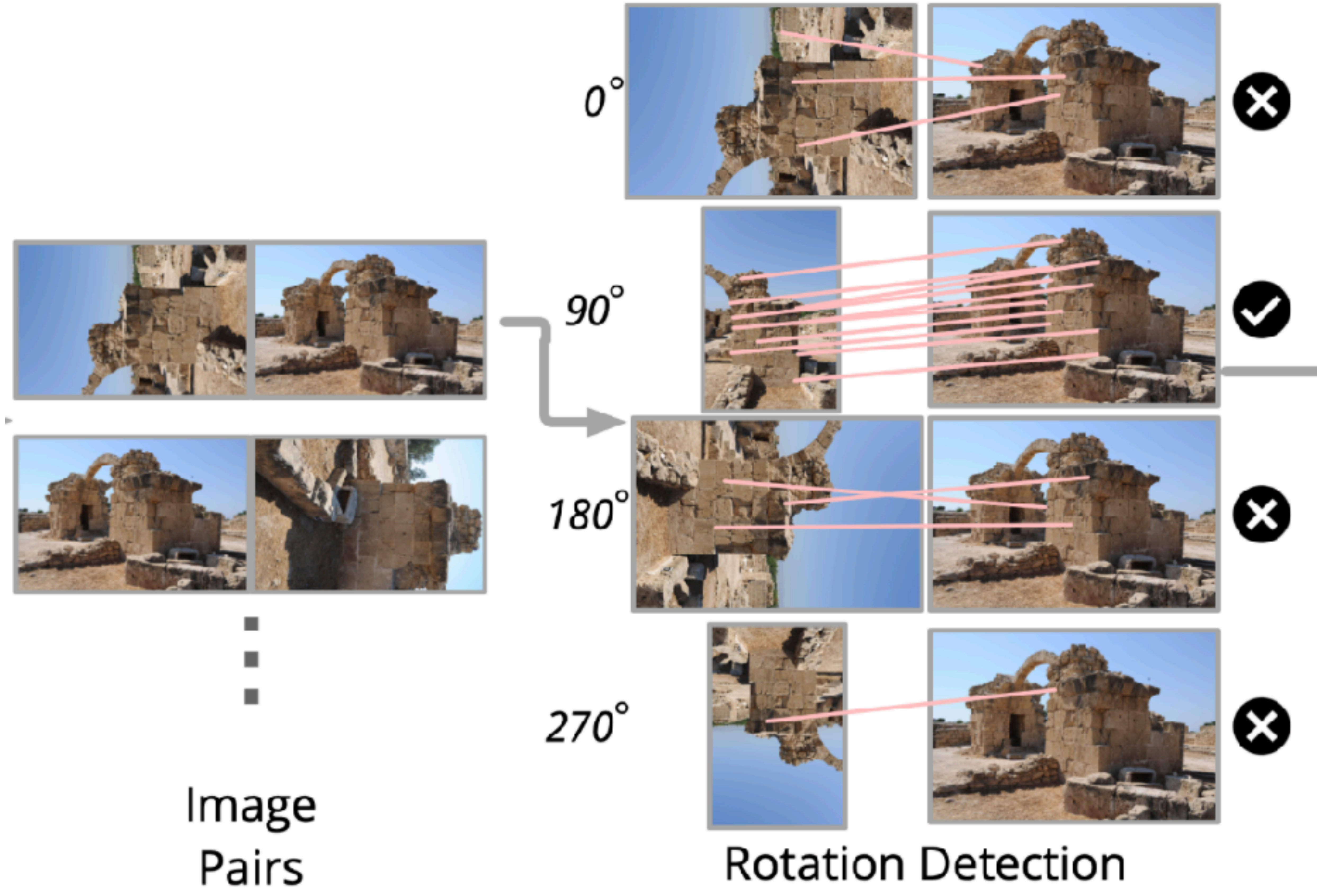
## Results

- Even harder to debug SfM on Kaggle
- Not many academics took part

Technical results:

- Brute-force rotation estimation for SuperGlue/LoFTR
- Detector-free SfM appeared
- LightGlue appeared!
- Handcrafted off-the-shelf kornia local feature got 5th place!

https://ducha-aiki.github.io/wide-baseline-stereo-blog/2023/07/05/IMC2023-Recap.html




https://www.kaggle.com/competitions/image-matching-challenge-2023/discussion/417407

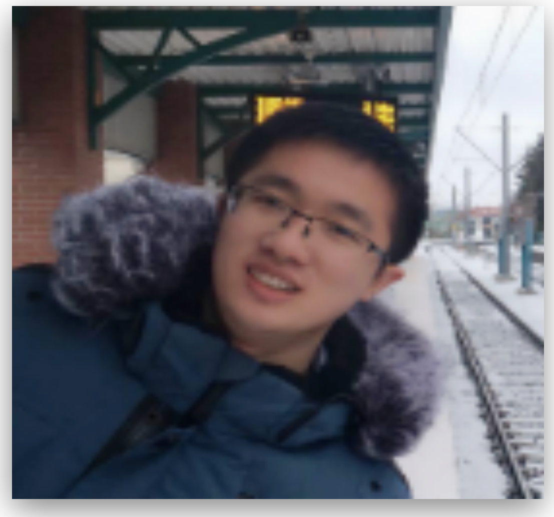
# Why do we do benchmarking?

- Understand the state-of-the-art. Many people stop here.
- Measure the progress of the field
- Find open questions
- Direct the research in a certain way
    - Can be area, practices, etc.

We spent 2019 — 2023 trying to understand the SoTA and measure the progress.

We also all the time advocated for downstream methods and proper RANSACs, and that seems to be successful.

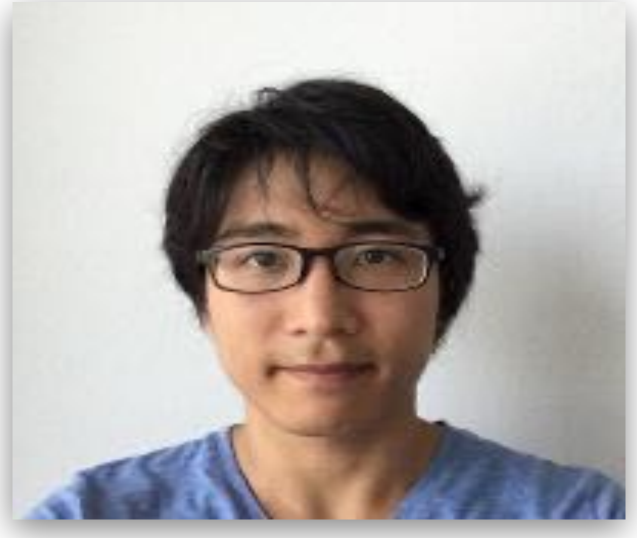
Now we are trying to direct Image Matching/SfM into more diverse areas.



**Weiwei Sun**
U. British Columbia

**Amy Tabb**
USDA-ARS-AFRS

**Eduard Trulls**
Google



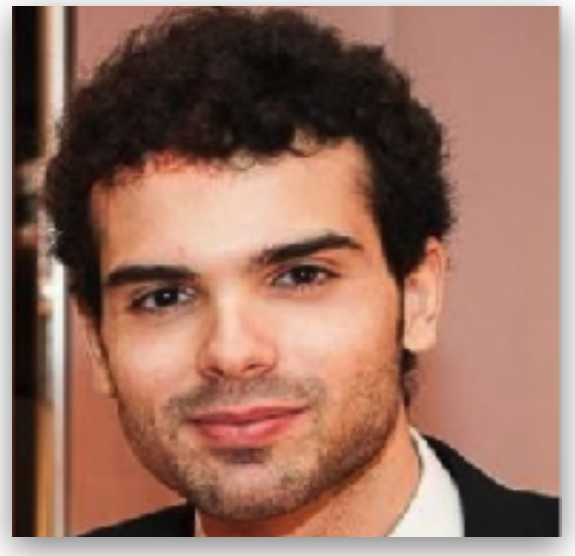
**Kwang Moo Yi**
U. British Columbia

# Image Matching Challenge 2024

**Fabio Bellavia**
Univ. Palermo

**Jiri Matas**
CTU Prague

**Dmytro Mishkin**
CTU Prague/HOVER Inc.



**Luca Morelli**
Univ. Trento/BFK

**Fabio Remondino**
Bruno Kessler Foundation

Still available at https://www.cs.ubc.ca/research/image-matching-challenge/2021/leaderboard/

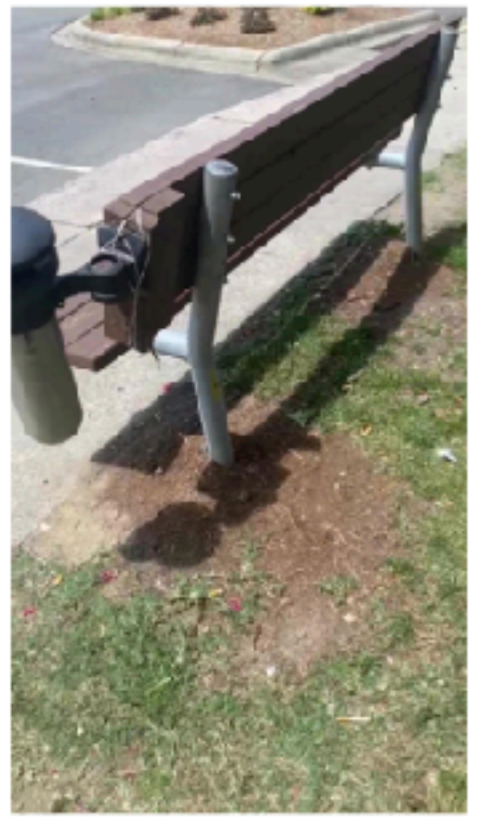
# IMC2024: before the start

- Standard two-view matching is mostly solved
- So we need to try new things
  - either SfM-based
  - or super hard images

IMC 2022

| Method ↓ mAA → | @10 ↑ |
|---|---|
| SiLK [21] | 68.6 |
| SP [14]+SuperGlue [41] | 72.4 |
| LoFTR [44] CVPR'21 | 78.3 |
| MatchFormer [55] ACCV'22 | 78.3 |
| QuadTree [46] ICLR'22 | 81.7 |
| ASpanFormer [12] ECCV'22 | 83.8 |
| DKM [17] CVPR'23 | 83.1 |
| **RoMa** | **88.0** |

RoMA

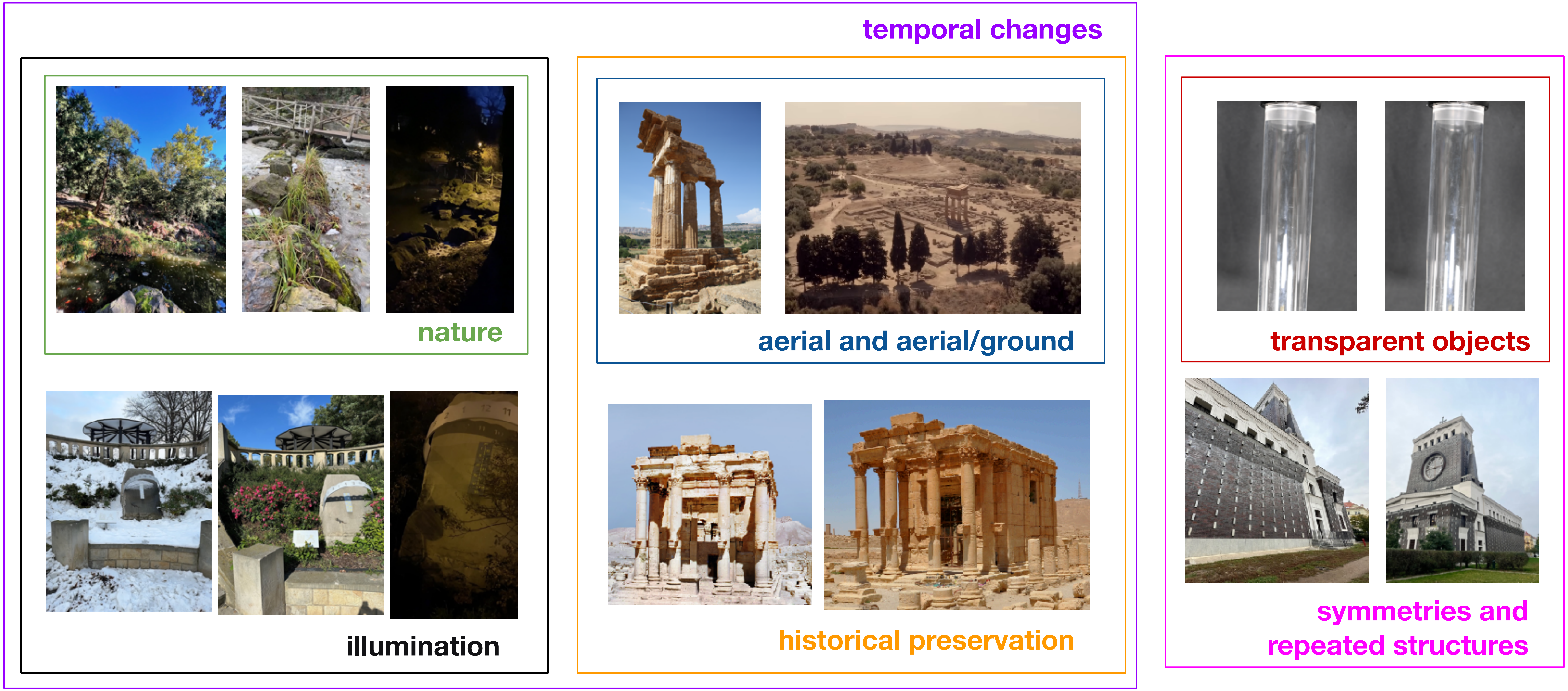


Dust3r

RoMa: Robust Dense Feature Matching

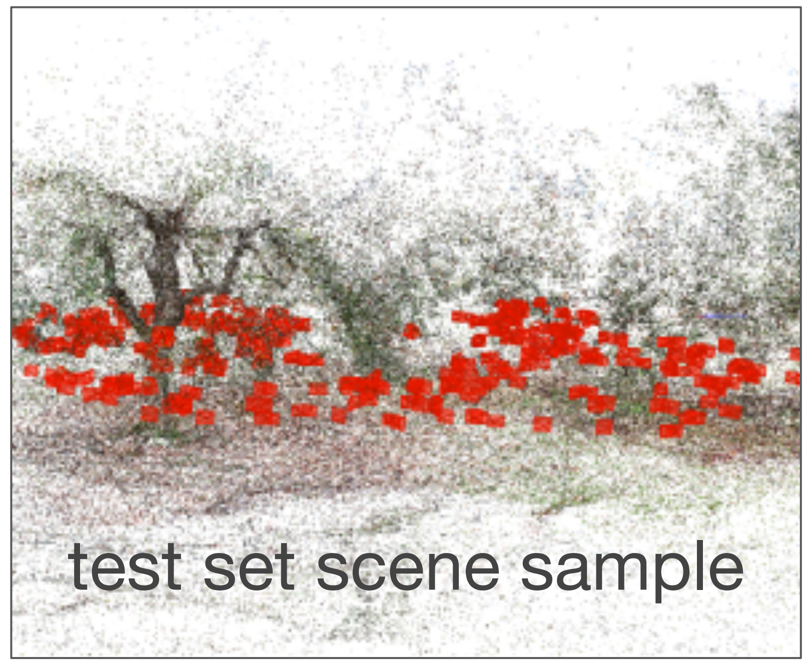
DUSt3R: Geometric 3D Vision Made Easy

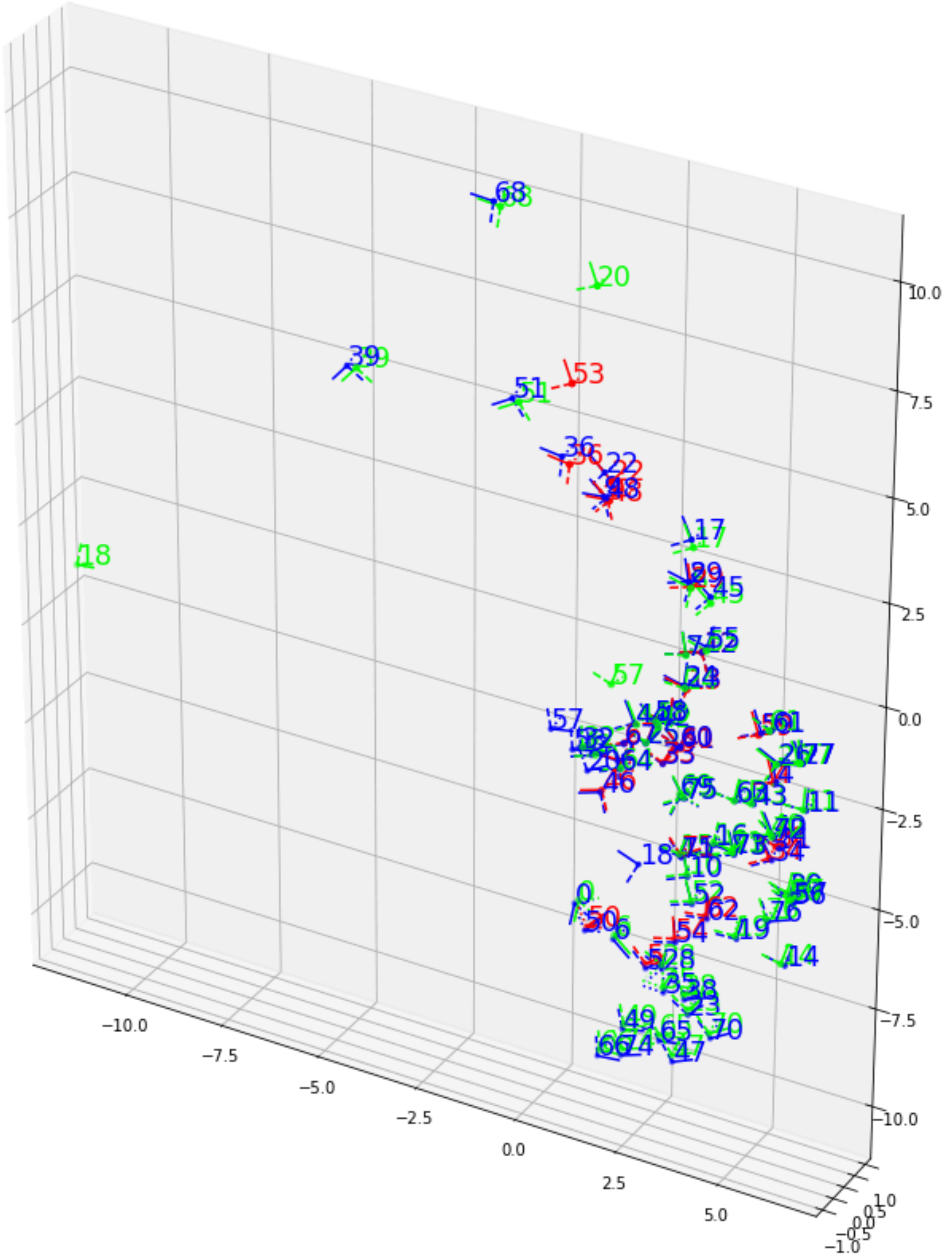
# IMC2024: Hexathlon



# IMC2024: registration-based metric

- Cameras are aligned with exhaustive RANSAC-like registration with list of triplets for Horn-based solver.
- Metric is mAA on purely translation error between GT and solution



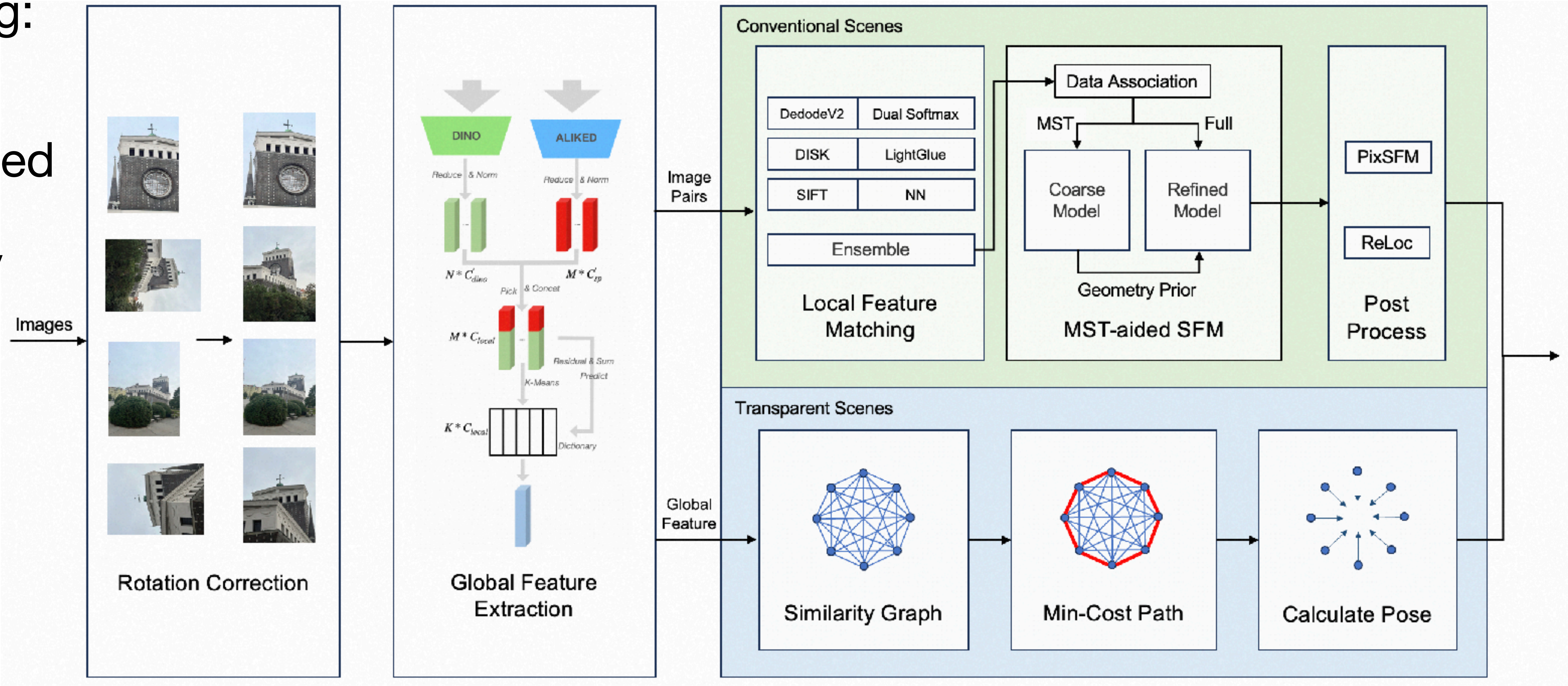



best submission (public/private split) aligned with ground-truth

# IMC2024: results

Good thing:

Winners actually tried to solve covisibility graph first




https://www.kaggle.com/competitions/image-matching-challenge-2024/discussion/510499

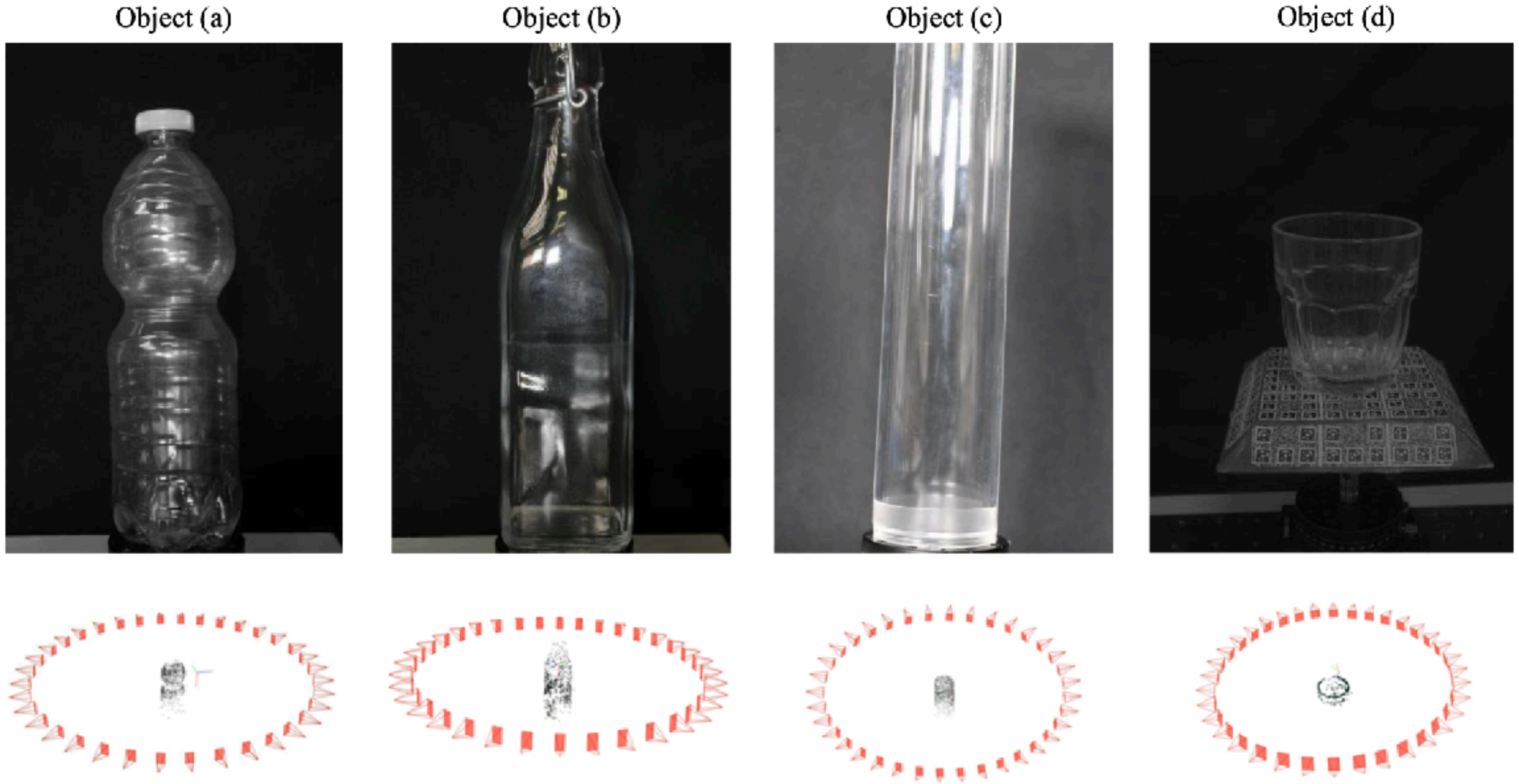
# IMC2024: our failures

## Any shortcuts will be exploited on Kaggle

Issue #1: transparent objects were shoot on turntable. Moreover, we leaked some objects.

We thought that scrambling image order made the task hard enough.

It turns out, that local order is recoverable, and that is enough together with hardcoding positions



# IMC2024: our failures

## Any shortcuts will be exploited on Kaggle

Issue #2: It is extremely hard to debug on Kaggle

Issue #3: People also like to see the images from the test set.

- This is bad from a benchmark fairness point of view
- But it is good for the community to know what to work on
- Having bigger training/validation set would help, but that is hard to get

# Challenges on Kaggle

Good:

- Fair benchmark with a hidden test set
- challenge the over-complicated methods vs crowd-source tuned baselines
- solving non fancy and “unpublishable” things
- “free” compute for participants
- Kaggle may provide $50k prizes via their academic program

Bad:

- cannot do long term leaderboard
- debugging
- high entry threshold for non-python stuff
- single metric only
- $50k minimum prize fund

# Why participate?

- You got a fair result of your method on a challenging problem
  - But it is hard to be sota, so maybe bad for convincing R2.
- But if you beat Kaggle, then you are really good!
  - Some things are not obvious before you try
  - E.g., running on GPU-poor machines in real scenario
    - VGGSfM (2024) has to be significantly optimized to be even able to run on IMC2024
- Prizes are big :)

# IMC Summary (research and practical)

- Downstream metrics are the way to go for image matching
- Tune all the components of the system (best by crowd-sourcing on Kaggle)
- Two-view matching is solved for many cases, while the SfM is not
- SfM Scalability is under-explored (and we don’t do BoW retrieval anymore)
- Diversity in the datasets is essential
- Proper metrics are hard
- Do multi-year benchmarks

# Image Matching Challenge 2025

- Probably will be on Kaggle (we are in discussing it)
- No transparent objects
- Improve metric a little bit more
- Add another practical difficulty (secret for now)

**Thank you for your attention!**