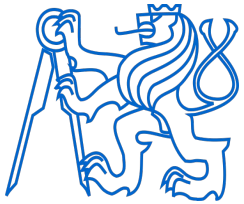
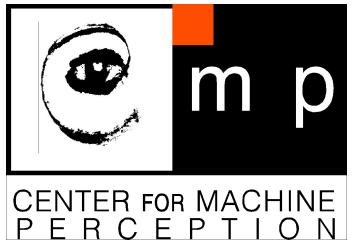




Benchmarking Robust Estimation Methods

Dmytro Mishkin, Czech Technical University in Prague

Deep matchers: RANSAC sucks!

Is that so?

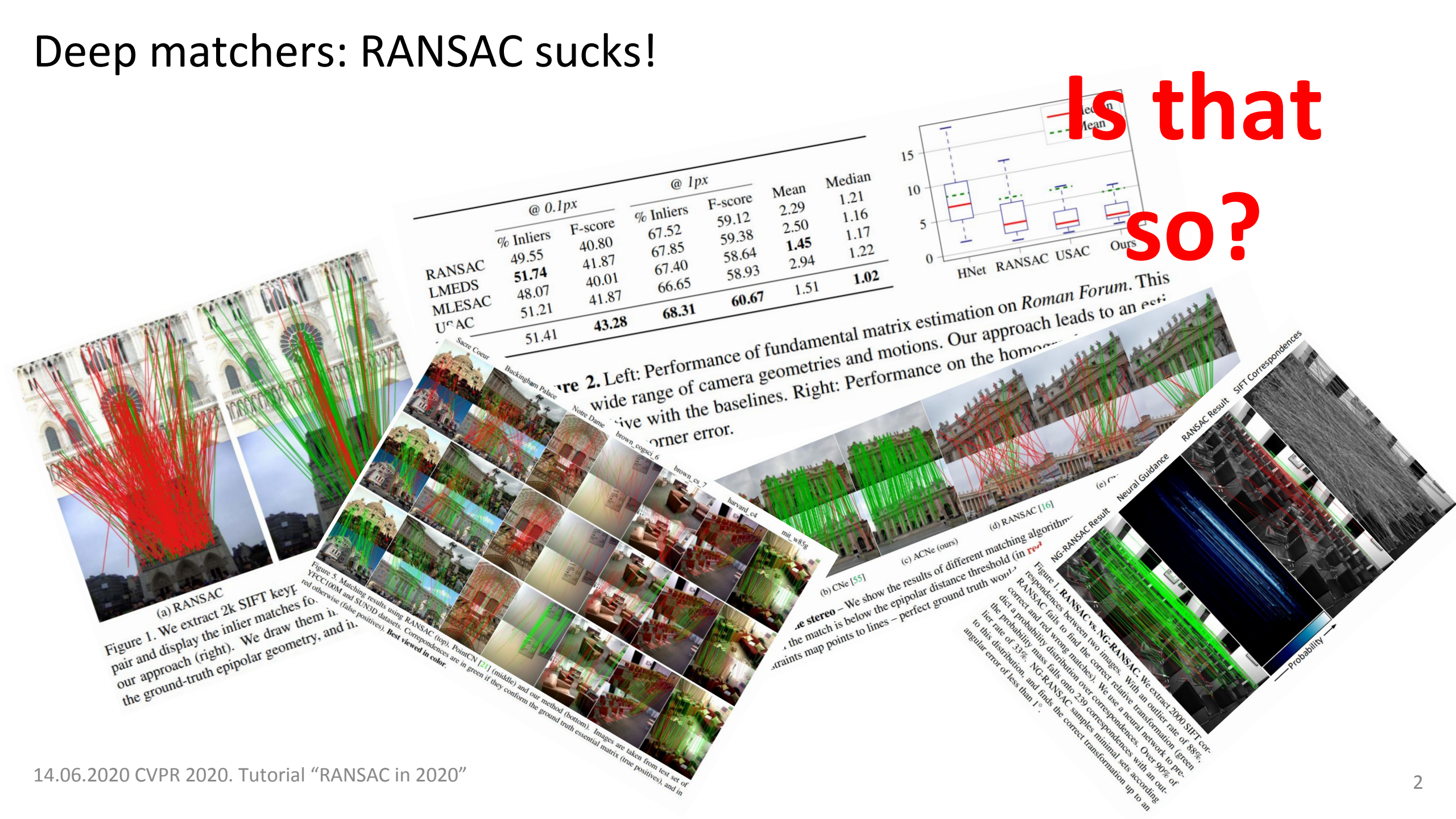


Figure 1. We extract 2k SIFT keyp
pair and display the inlier matches fo
our approach (right). We draw them in
the ground-truth epipolar geometry, and in

Figure 2. Left: Performance of fundamental matrix estimation on *Roman Forum*. This
wide range of camera geometries and motions. Our approach leads to an est
ive with the baselines. Right: Performance on the homog
orner error.

(a) RANSAC (b) CNe [55] (c) ACNe (ours) (d) RANSAC [16]

Figure 5. Matching results using RANSAC (top), PointCN [21] (middle) and our method (bottom). Images are taken from test set of YFCC100M and SUN3D datasets. Correspondences are in green if they conform the ground truth essential matrix (true positives), and in red otherwise (false positives). **Best viewed in color.**

ue stereo – We show the results of different matching algorithm
the match is below the epipolar distance threshold (in red
straints map points to lines – perfect ground truth would

Figure 1. **RANSAC vs. NG-RANSAC.** We extract 2000 SIFT correspondences between two images. With an outlier rate of 88%, RANSAC fails to find the correct relative transformation (green correct and red wrong matches). We use a neural network to predict a probability distribution over correspondences. Over 90% of the probability mass falls onto 239 correspondences with an outlier rate of 33%. NG-RANSAC samples minimal sets according to this distribution, and finds the correct transformation up to an angular error of less than 1°.

We would like to benchmark the following problems

- Fundamental & essential matrices
- Homography
- PnP
- Rigid point cloud registration
- Multi-instance estimation
- Multi-model estimation

2 month before tutorial….

We would like to benchmark the following problems

- Fundamental & essential matrices
- Homography
- PnP
- Rigid point cloud registration

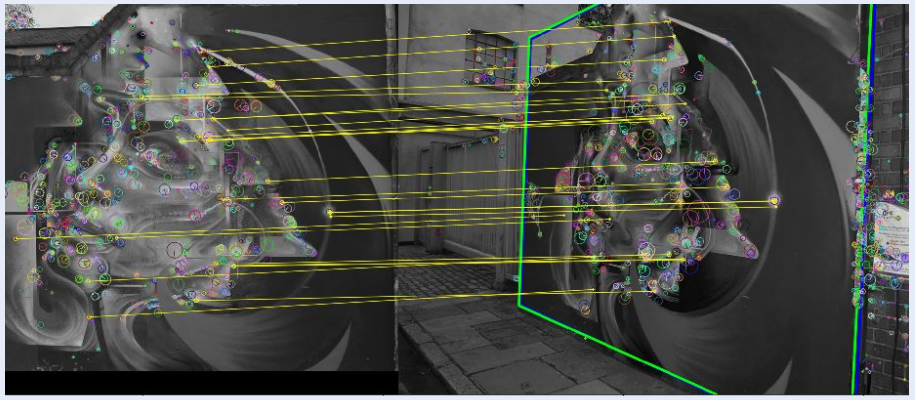
1 month before tutorial….

We have the results for

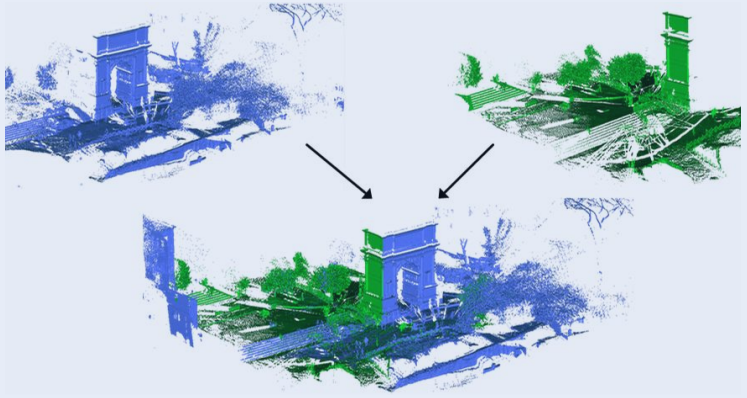
1 day before tutorial…

Fundamental
&
Essential matrices

Homography



Rigid point cloud registration



Why the robust estimation benchmark is hard?

- There are few datasets for RANSAC-like problems and existing ones are small
  - That is why one cannot train/evaluate deep learning method on existing datasets
- There is no common evaluation protocol
  - What is the metric? Inlier ratio? Number of “correct” inliers? % of successes?
- There is no training/validation/test split.
  - **Common practice - tune the methods on the same dataset, or even image pair, it is tested on**.
- There are often no fixed set of correspondences to use.
  - One can easily make almost any method to outperform the rest because of differences in preprocessing

RANSAC 2020: benchmark collection

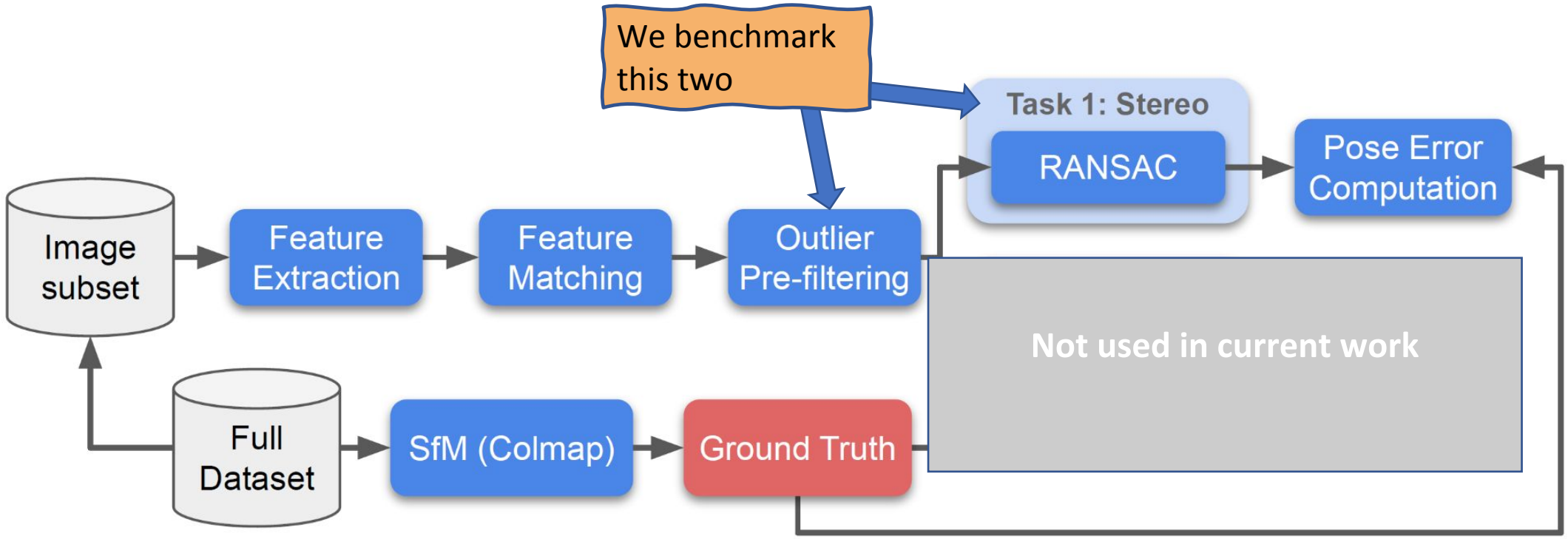
- We haven’t been able to solve all the mentioned problem in the current evaluation.
- But we believe that we did a good step in the right direction.
- Input data and evaluation scripts (not documented well yet) are online:

  https://github.com/ducha-aiki/ransac-tutorial-2020-data

- Let’s start!

Fundamental and essential estimators

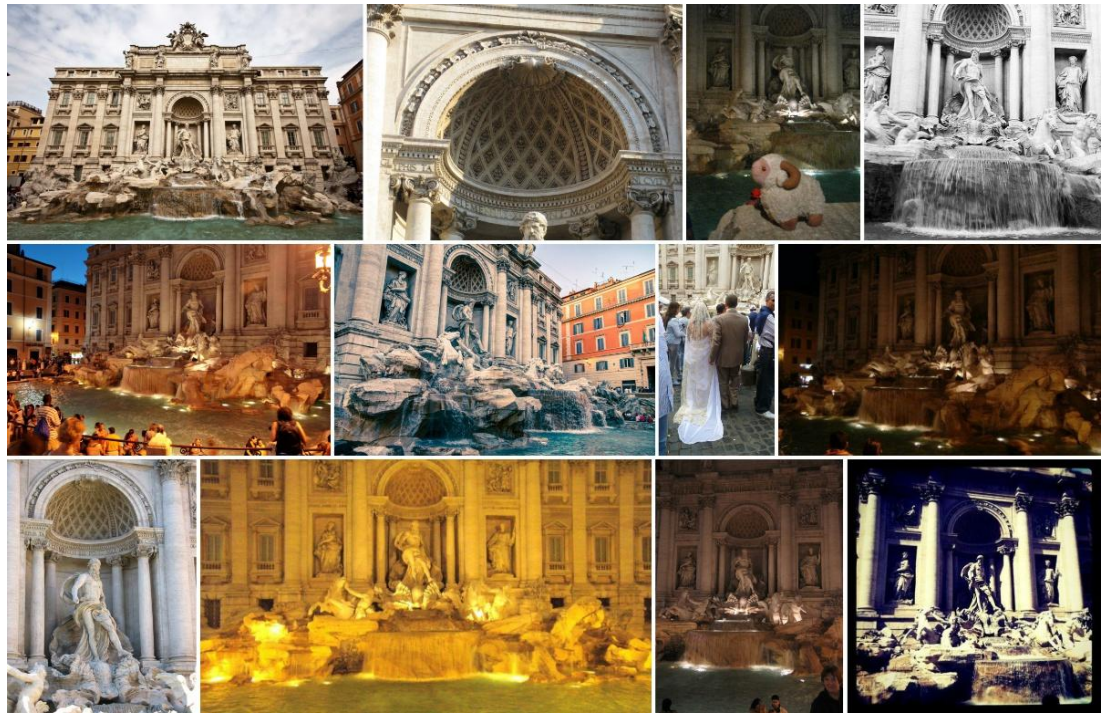
Modular WBS pipeline to benchmark everything



- 15 training image sets
- 10 test image sets, 100 images each
- Metric: pose accuracy

Image Matching across Wide Baselines: From Paper to Practice.
Jin et.al, arXiv 2020

The Phototourism Dataset



- 30k images from YCC100M dataset, in 26 scenes
- “Ground truth” established by COLMAP reconstruction
- The basis of Image Matching Competitions 2019 & 2020

The Phototourism Dataset

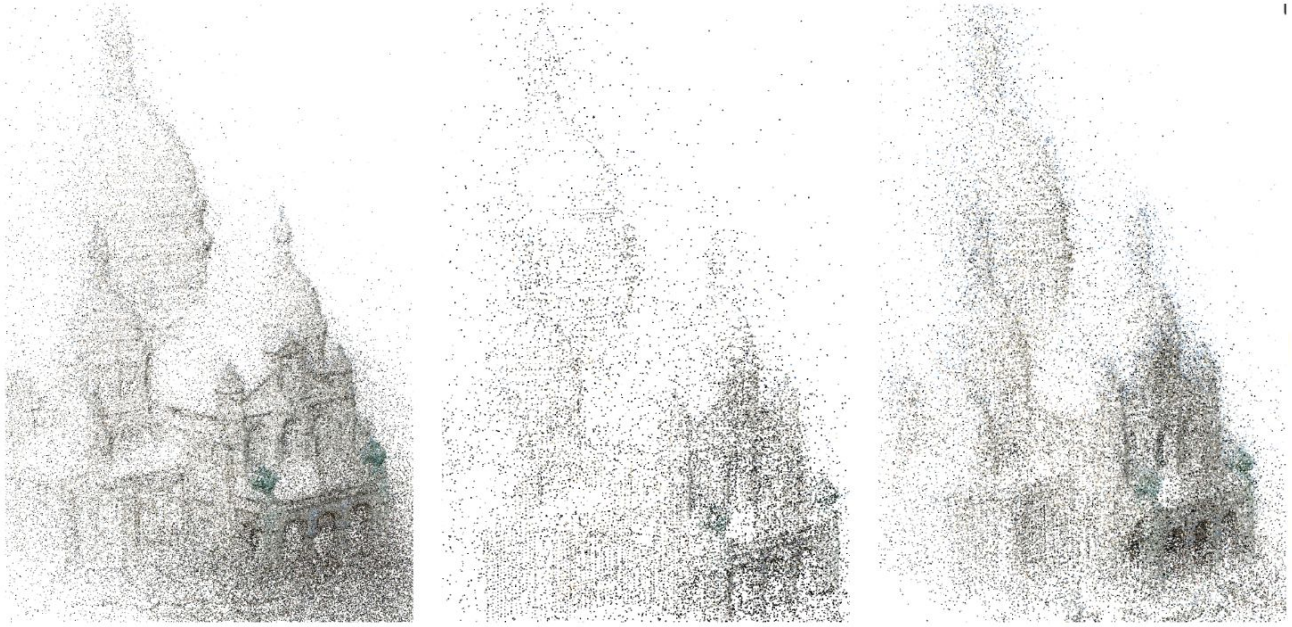
Training sets

Name	Images	3D points
"Brandenburg Gate"	1363	100040
"Buckingham Palace"	1676	234052
"Colosseum Exterior"	2063	259807
"Grand Place Brussels"	1083	229788
"Hagia Sophia Interior"	888	235541
"Notre Dame Front Facade"	3765	488895
"Palace of Westminster"	983	115868
"Pantheon Exterior"	1401	166923
"Prague Old Town Square"	2316	558600
"Reichstag"	75	17823
"Sacre Coeur" (SC)	1179	140659
"St. Peter's Square" (SPS)	2504	232329
"Taj Mahal"	1312	94121
"Temple Nara Japan"	904	92131
"Trevi Fountain"	3191	580673
"Westminster Abbey"	1061	198222
Total	25.7k	3.6M

Test sets

Name	Images	3D points
"British Museum" (BM)	660	73569
"Florence Cathedral Side" (FCS)	108	44143
"Lincoln Memorial Statue" (LMS)	850	58661
"London (Tower) Bridge" (LB)	629	72235
"Milan Cathedral" (MC)	124	33905
"Mount Rushmore" (MR)	138	45350
"Piazza San Marco" (PSM)	249	95895
"Sagrada Familia" (SF)	401	120723
"St. Paul's Cathedral" (SPC)	615	98872
Total	3774	643k

Can we trust Colmap “Ground truth”?



(a) SIFT (b) SuperPoint (c) R2D2

Yes, we can!

Feature used	Number of images			
	100 vs. all	200 vs. all	400 vs. all	800 vs. all
SIFT [55]	0.46° / 0.13°	0.42° / 0.11°	0.32° / 0.08°	0.39° / 0.08°
SuperPoint [34]	2.09° / 1.57°	2.09° / 1.54°	1.87° / 1.21°	2.53° / 0.53°
R2D2 [80]	0.41° / 0.14°	0.29° / 0.09°	0.28° / 0.09°	0.21° / 0.06°

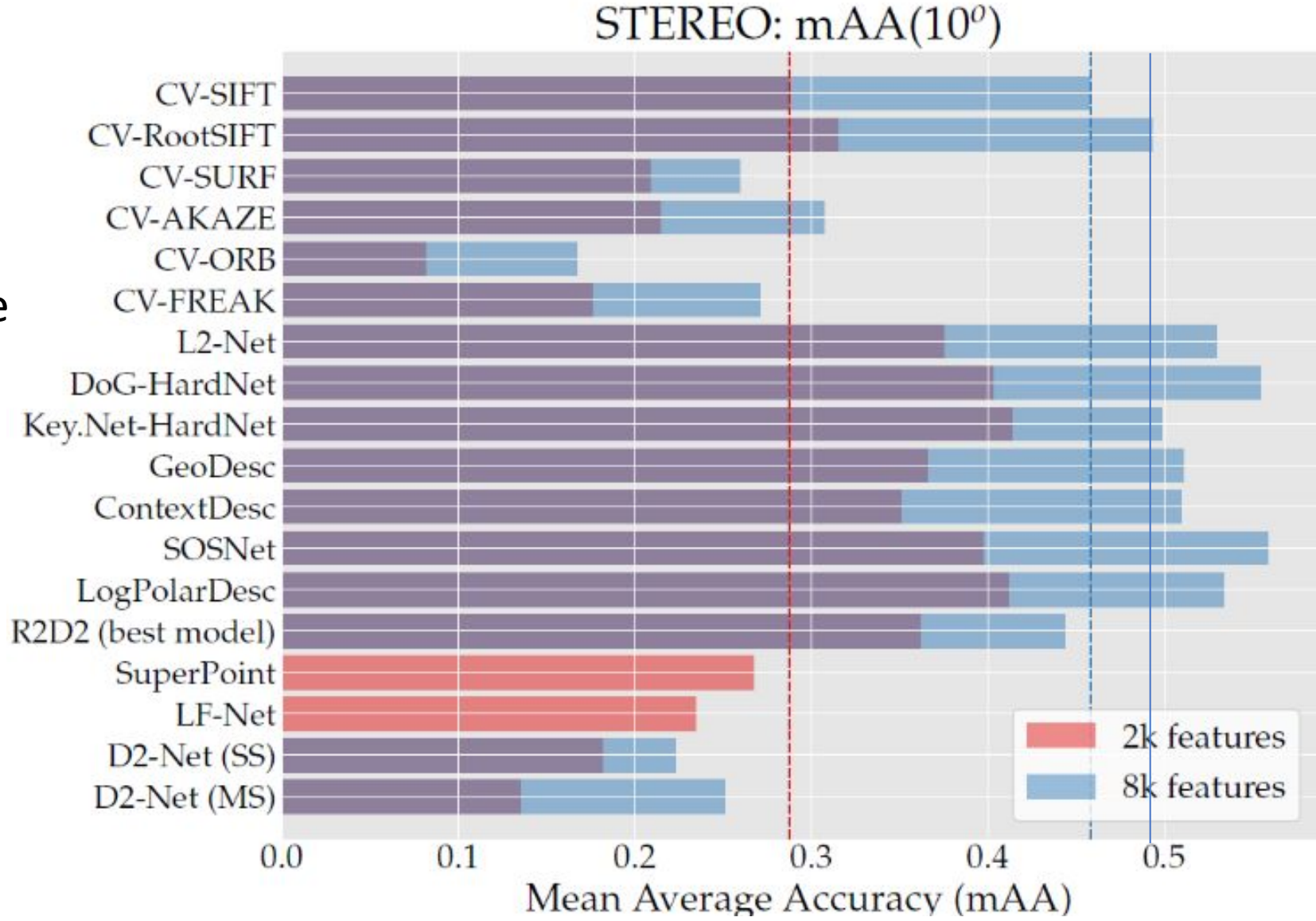
Table 3 Pose convergence in SfM. We report the mean/median of the difference (in degrees) between the poses extracted with the full set of 1179 images for “Sacre Coeur”, and different subsets of it, for three local feature methods – to keep the results comparable we only look at the 100 images in common across all subsets. We report the maximum among the angular difference between rotation matrices and translation vectors. The estimated poses are stable, with as low as 100 images.

Metric computation

1. RANSAC $\longrightarrow$ F or E
   a. Essential matric E from F: $E = K_1^T F\ K_2$
2. Camera pose (R,t) from OpenCV function cv2.recoverPose(E, inliers)
3. Decompose (R,t) into rotation and translation components, keep only rotation, get the angular error
4. Threshold angular error for set of thresholds and get accuracy per threshold
5. Calculate mAA @ 10

Local features comparison on PhotoTourism

- Local feature: RootSIFT.
  - RootSIFT is a very strong baseline
    #9 among all methods
  - 13.1% relative less than the best
- Matching: mutual nearest neighbour
- Additional info: SNN Lowe ratio



https://vision.uvic.ca/image-matching-challenge/leaderboard/

Training and test correspondences: the same for all methods

- Training data: 100k image pairs per training sequence
  - Total: 1.6M image pairs, 500-3000 correspondences each.
- Test data: 4950 image pairs per test sequence
- The training data and the test data (without GT) was sent to deep learning method authors.

Evaluated methods: classical F

Baselines

- scikit-image: 8pt vanilla RANSAC + final least squares fitting
- OpenCV RANSAC: 7pt vanilla RANSAC + Levenberg-Marquardt final fitting
- OpenCV LMedS: least medians of squares
- LO-RANSAC: 7pt LO-RANSAC + final least squares fitting

Advanced methods

- DEGENSAC: Two-view Geometry Estimation Unaffected by a Dominant Plane
- GC-RANSAC: Graph-Cut RANSAC
- MAGSAC: marginalizing sample consensus
- MAGSAC++, a fast, reliable and accurate robust estimator
- USAC: A Universal Framework for Random Sample Consensus
- AC RANSAC: Automatic Homographic Registration of a Pair of Images with A Contrario Elimination of Outliers
- GC-RANSAC + DEGENSAC
- GC-RANSAC + DEGENSAC + MAGSAC++

Classical methods, which are not included

- Affine RANSACs (2AC, 3AC)
- Colmap RANSAC
- Guided matching methods

Evaluated methods: learned F

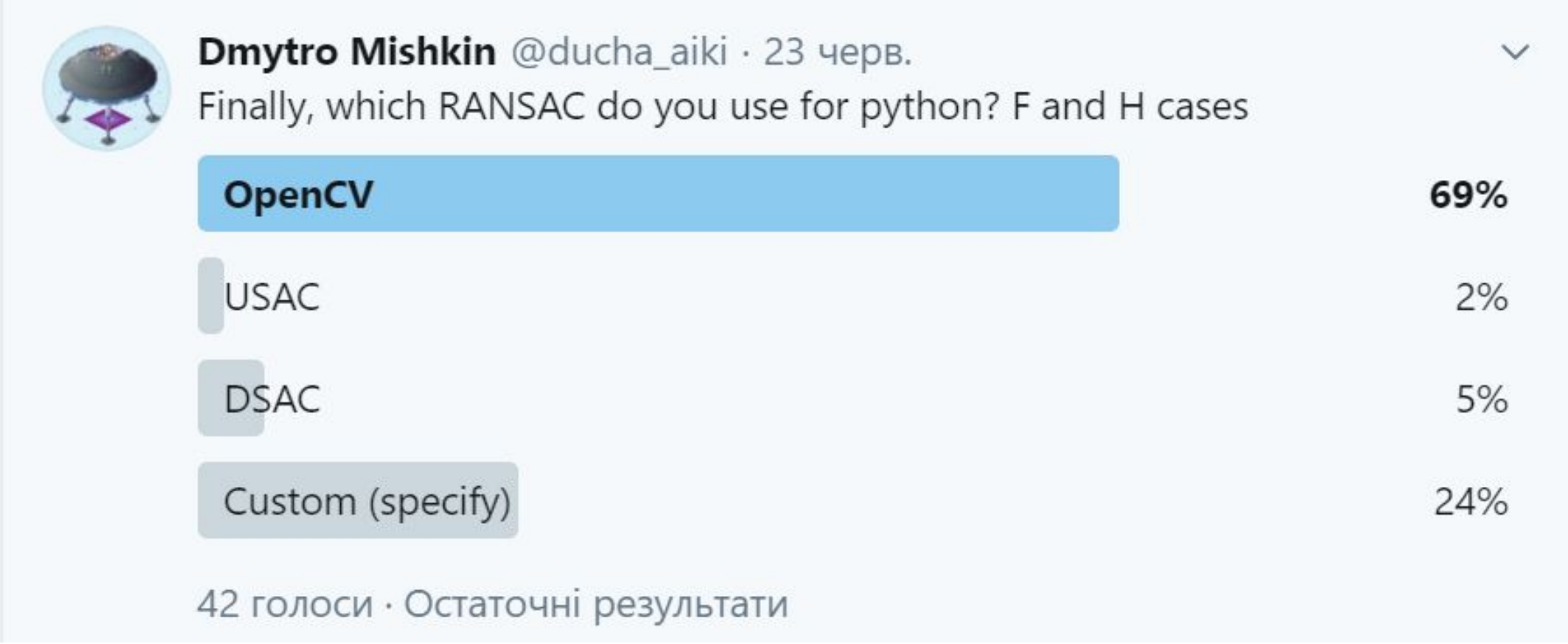
All methods evaluated are using only correspondences (x,y) (x,y) and the match confidence.

- DFE: Deep Fundamental Matrix Estimation
- CNe: Learning to Find Good Correspondences
- ACNe: Attentive Context Normalization for Robust Permutation-Equivariant Learning
- NM-Net: Mining Reliable Neighbors for Robust Feature Correspondences (v2)
- OANet: Learning Two-View Correspondences and Geometry Using Order-Aware Network (v2)

Not included:

- SuperGlue, as it takes much richer input, that the most of the methods in our study. SuperGlue uses all raw keypoints and descriptors from both images.
- NG-RANSAC (CVPR 2020) - lack of time, hope to add later
- 6D Conv (CVPR 2020) - lack of time, hope to add later
- Eigen-Free training (ECCV 0218) - author doesn't respond

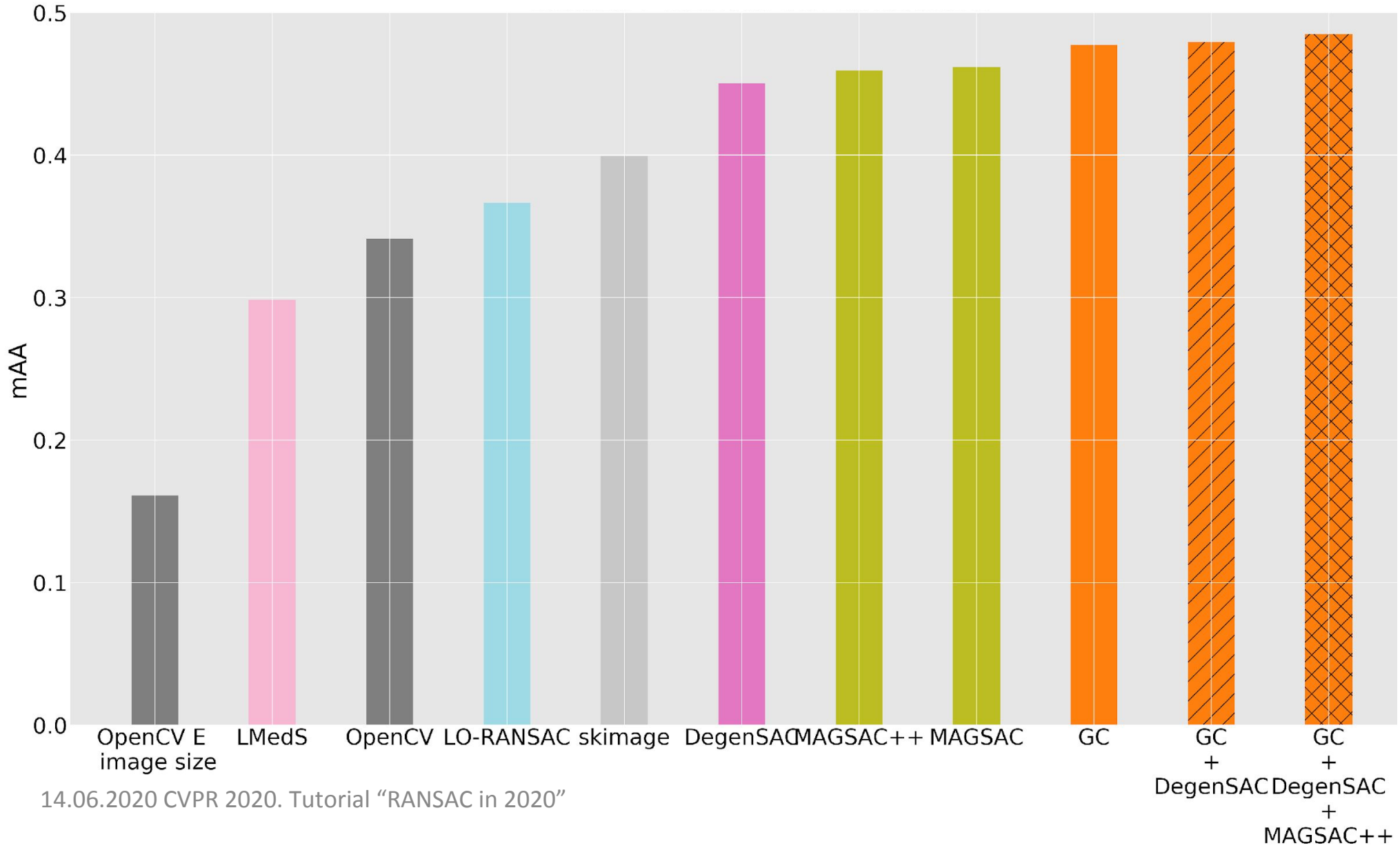
Is OpenCV RANSAC is a way to go?



OpenCV functions:
cv2.findHomography()
cv2.findFundamentalMatrix()

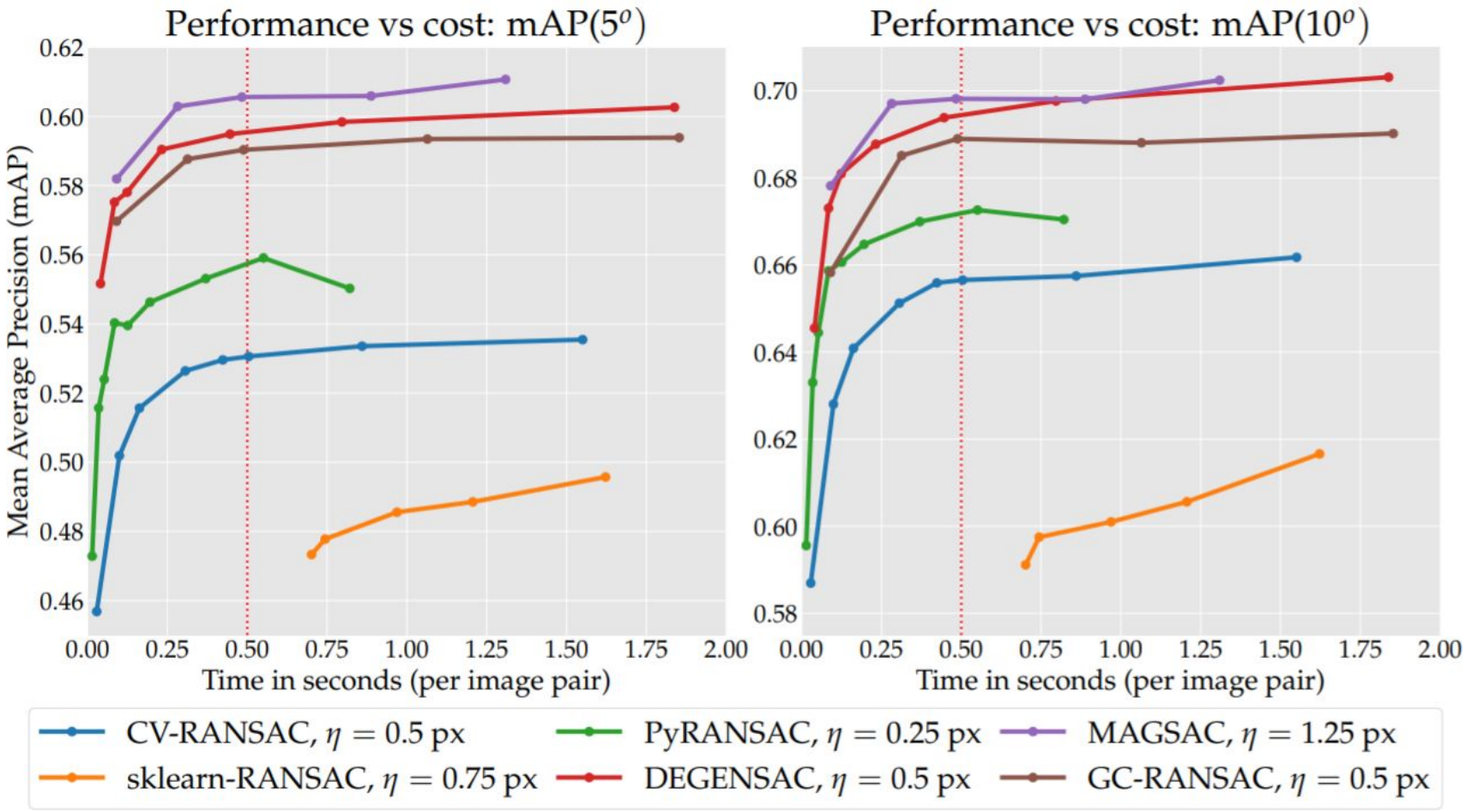
https://twitter.com/ducha_aiki/status/1142847831516037120

Classical F methods, 1k iterations



- Methods are sorted by accuracy
- sk-image RANSAC is orders of magnitude slower than the rest
- OpenCV is the least precise RANSAC

Classical F methods in Jin et.al 2020



Jin et.al 2020:

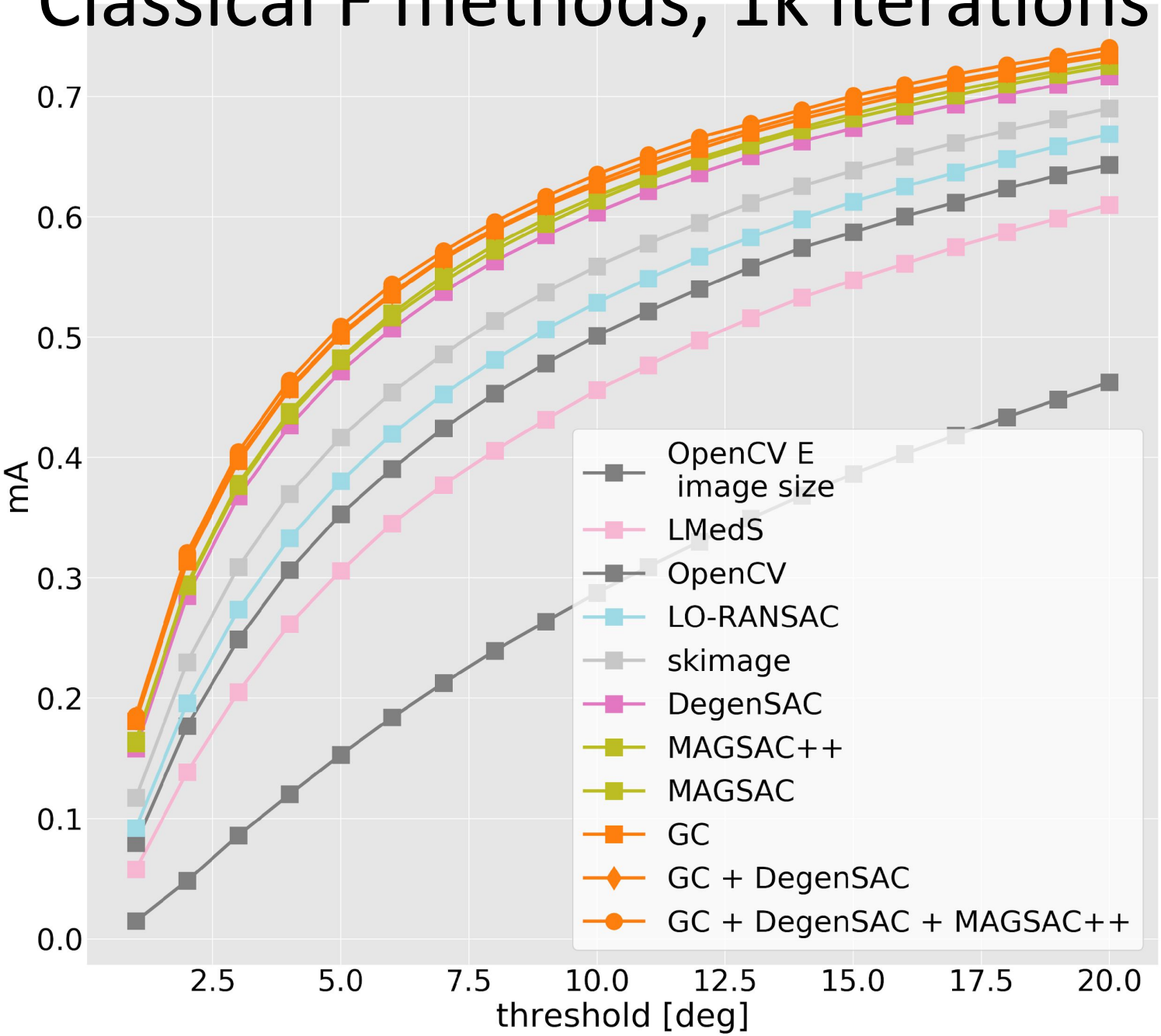
- Feature: SIFT, 8k points
- Vary maxIters, measure time.
- Advanced methods (MAGSAC, GC, DegenSAC) are better for both per second and per iteration

This tutorial:

- Benchmark was run on 4 different machines.
- Instead we fix the number of iterations for 1k, 10k, 100k, 1M
- We tune all parameters of the methods e.g. “spatial coherence term” for GC-RANSAC, which improves its results significantly

MAGSAC and MAGSAC++ github.com/danini/magsac (CVPR 2019 & CVPR 2020)
DEGENSAC github.com/ducha-aiki/pydegensac Chum et. al CVPR 2005
GC-RANSAC github.com/danini/graph-cut-ransac Barath and Matas. Graph-cut RANSAC. CVPR 2018

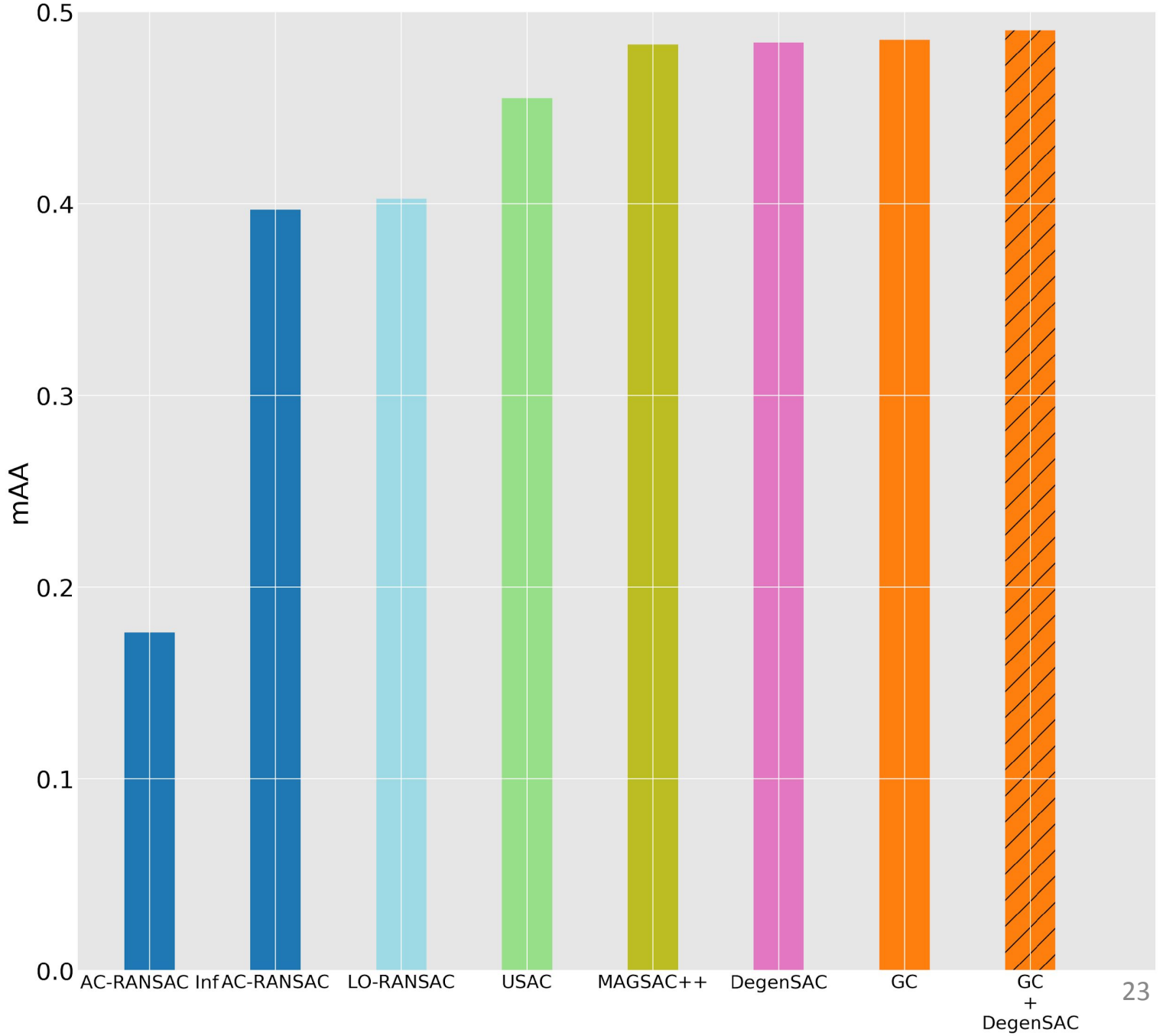
Classical F methods, 1k iterations



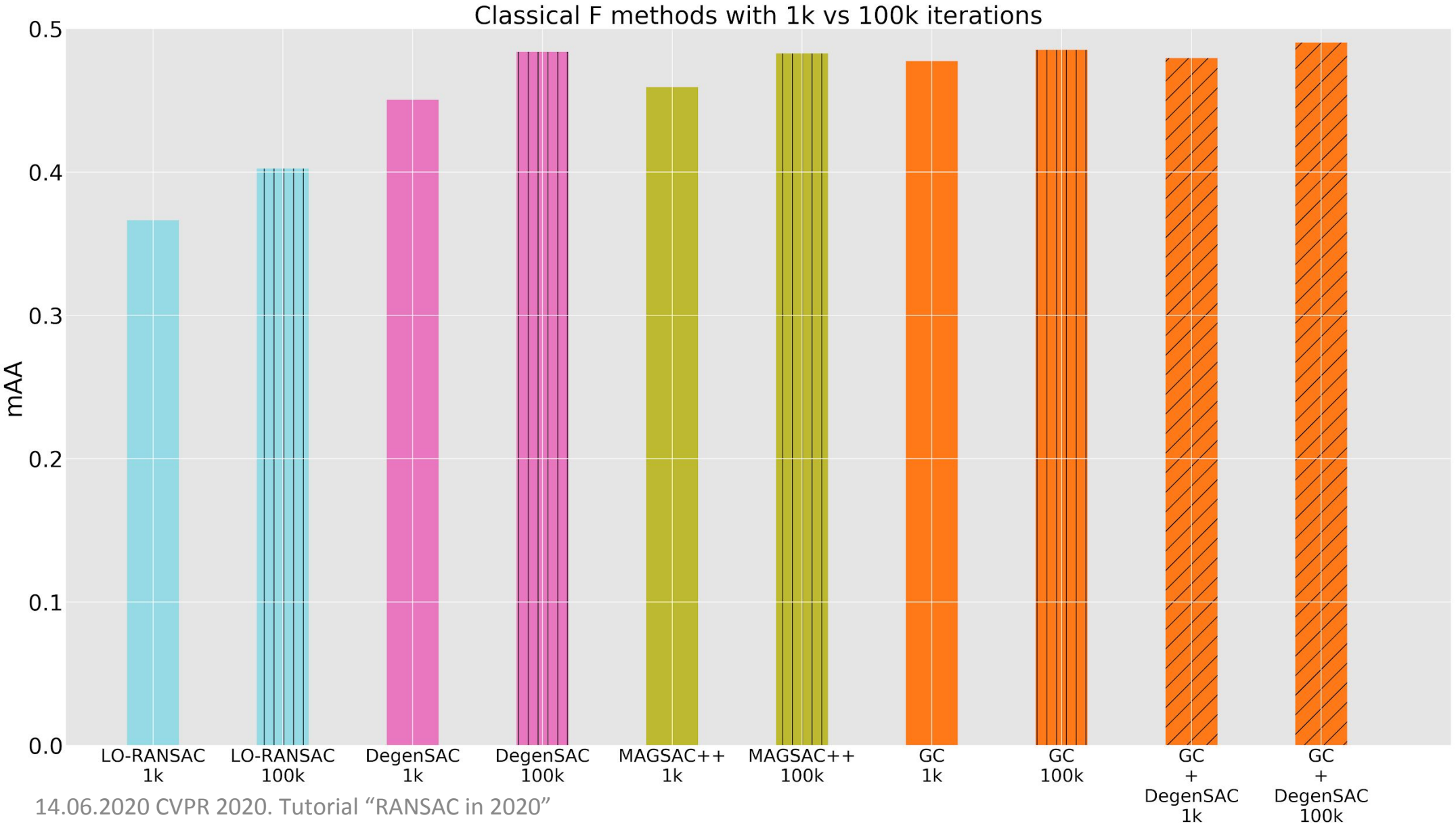
- Method ranking is consistent for different accuracy thresholds.
- In Jin et.al. GC-RANSAC < DEGENSAC, because of lacking of non-standard parameter tuning
- OpenCV E & F had hardcoded. maxIters = 1000. Now fixed, but not for pip install (yet?)

Classical F methods 100k iterations

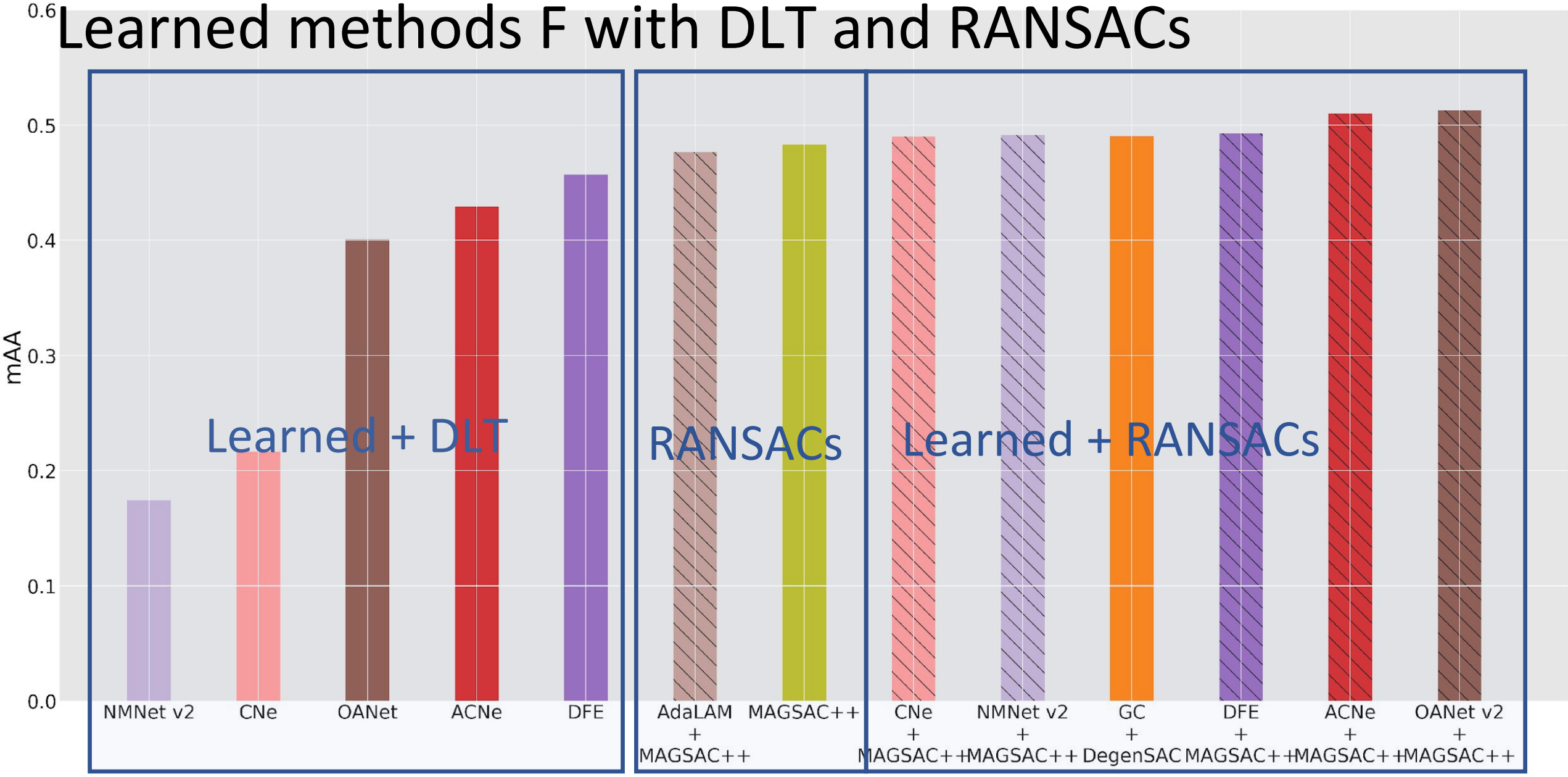
- A-Contrario RANSAC needs tuning as well as other RANSACs. Without, it performs very poorly
- The same is true for MAGSAC (untuned is not shown here, but it is bad)
- USAC is worse than standalone DegenSAC, probably because of implementation issues



Classical F: 1k iterations vs 100k iterations



Learned methods F with DLT and RANSACs



F-summary: general results

- We haven’t exaustively tried to pair all RANSACs with all deep methods
- Least squares (DLT) is not good enough for model estimation after deep methods.
- Degenerate solutions are the big problem
- “We are parameter-free” (AC-RANSAC, MAGSAC) claims are not true
- May be:
  - use iterated re-weighted least squares on top of deep correspondences
  - but also inside (small) RANSAC loop to ensure robustness

F-summary: recommendations

- Deep learned methods are not replacements for the RANSAC, they are replacement for the hand-crafted correspondence pre-filtering like Lowe SNN ratio.
- Performance of different RANSACs varies significantly, and all the methods have to be tuned to perform well
- Don’t use OpenCV or sk-image F-RANSACs, use GC-RANSAC, MAGSAC or DEGENSAC (all available with python bindings)
- Implementation matters (see USAC fail)

Evaluated methods: classical E

Baselines

- scikit-image: 8pt vanilla RANSAC + final least squares fitting
- OpenCV RANSAC: 5pt vanilla RANSAC + LM final fitting
- OpenCV LMeds: least medians

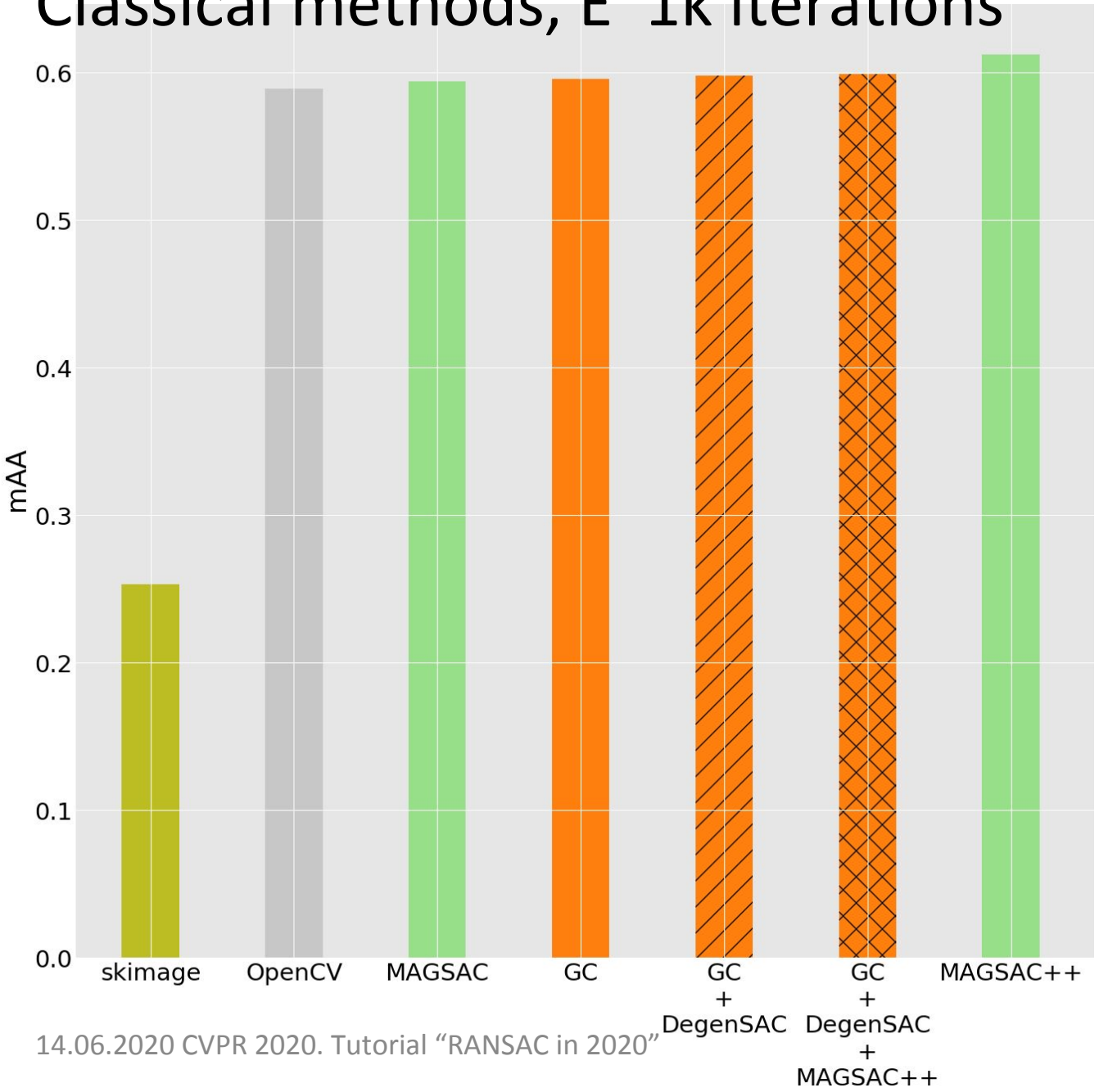
Advanced methods

- GC-RANSAC: Graph-Cut RANSAC
- MAGSAC: marginalizing sample consensus
- MAGSAC++, a fast, reliable and accurate robust estimator
- USAC: A Universal Framework for Random Sample Consensus
- AC RANSAC: Automatic Homographic Registration of a Pair of Images with A Contrario Elimination of Outliers
- GC-RANSAC + DEGENSAC
- GC-RANSAC + DEGENSAC + MAGSAC++

Evaluated methods: learned E

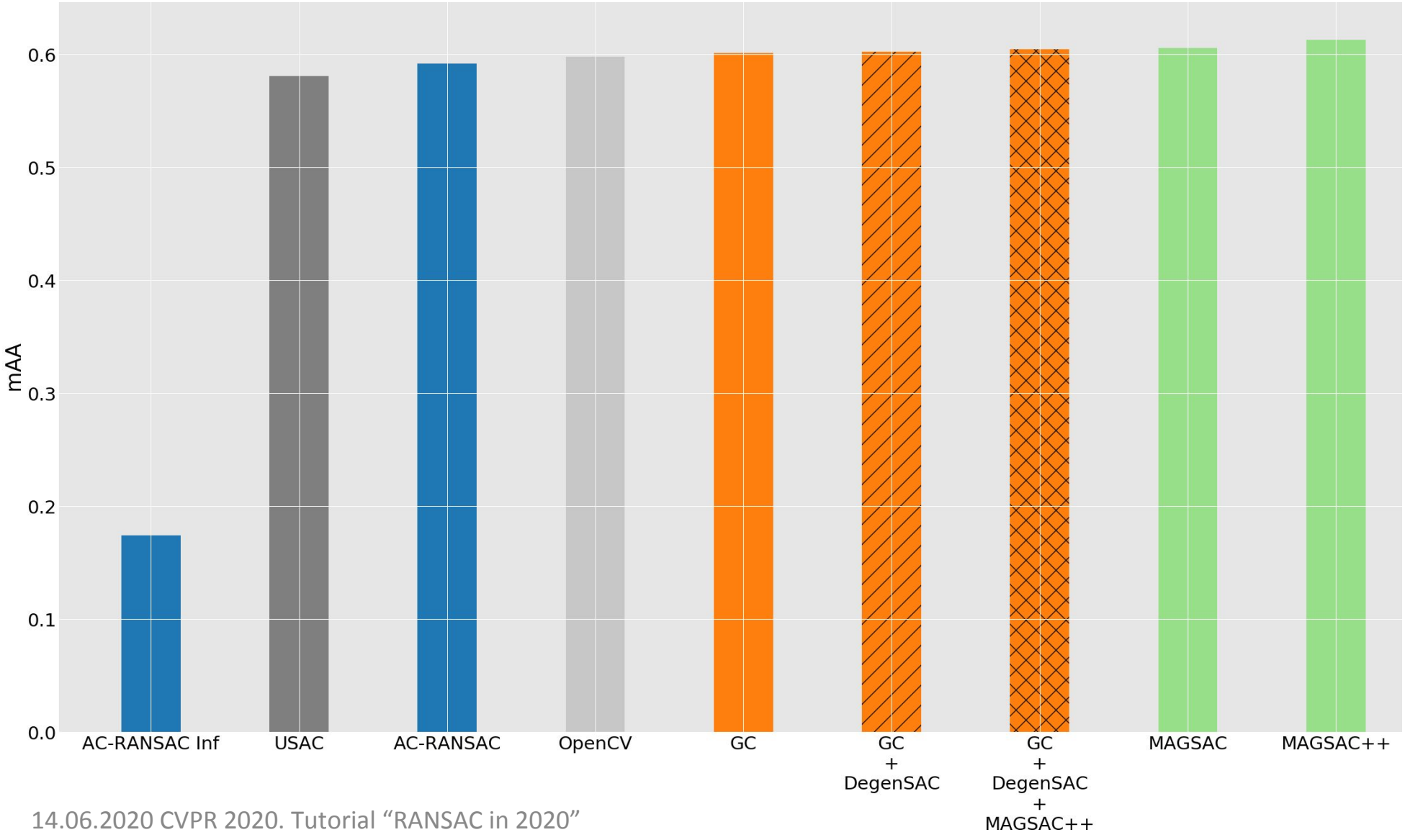
- CNe: Learning to Find Good Correspondences
- ACNe: Attentive Context Normalization for Robust Permutation-Equivariant Learning
- NM-Net: Mining Reliable Neighbors for Robust Feature Correspondences (v2)
- OANet: Learning Two-View Correspondences and Geometry Using Order-Aware Network (v2)

Classical methods, E 1k iterations



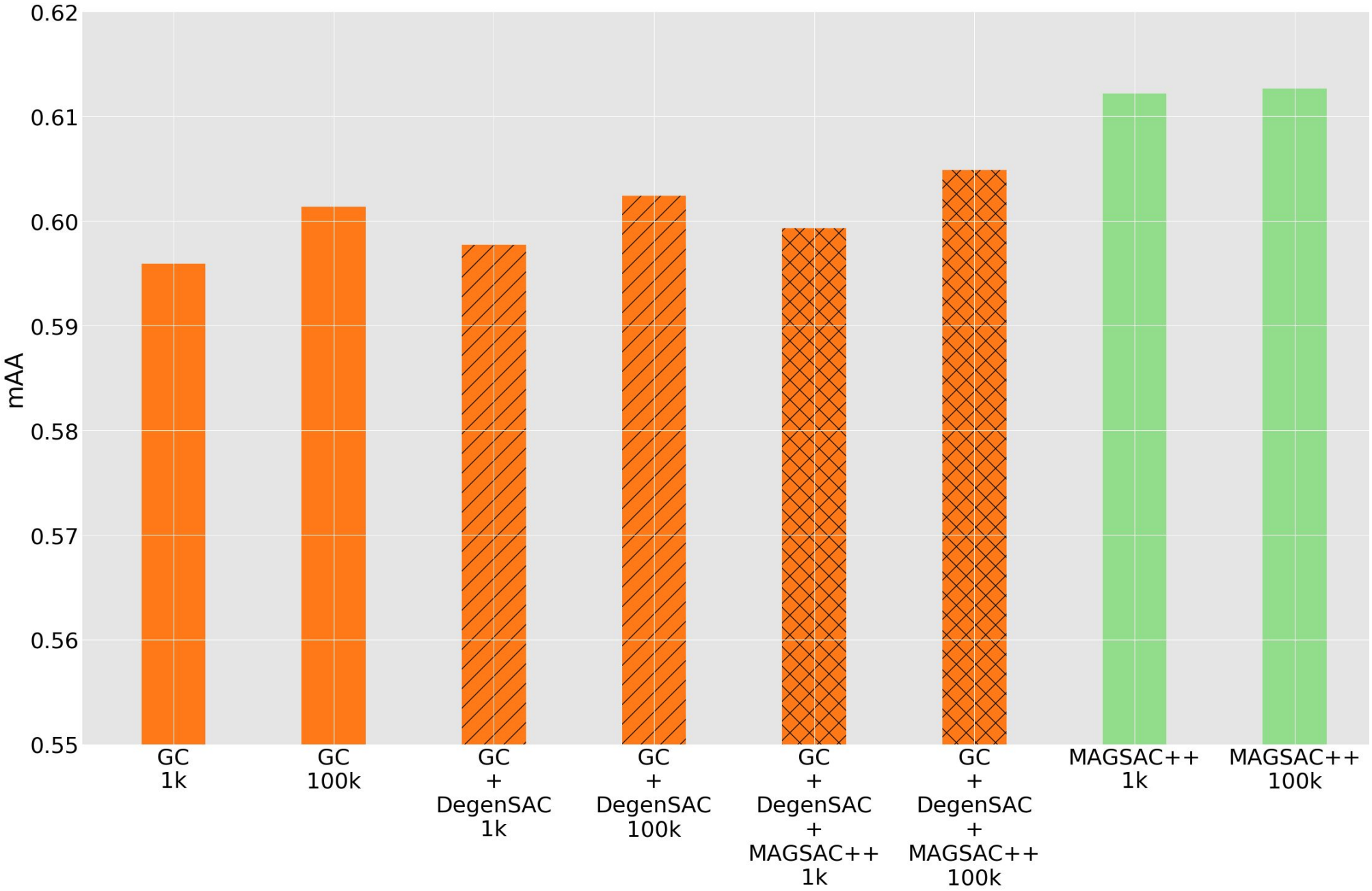
- skimage 8pt solver is significantly worse and orders of magniture slower than 5 pt solvers in the rest of methods
- Unlike F, there is no big difference in performance among methods
- OpenCV E works quite good

Classical methods, E 100k iterations



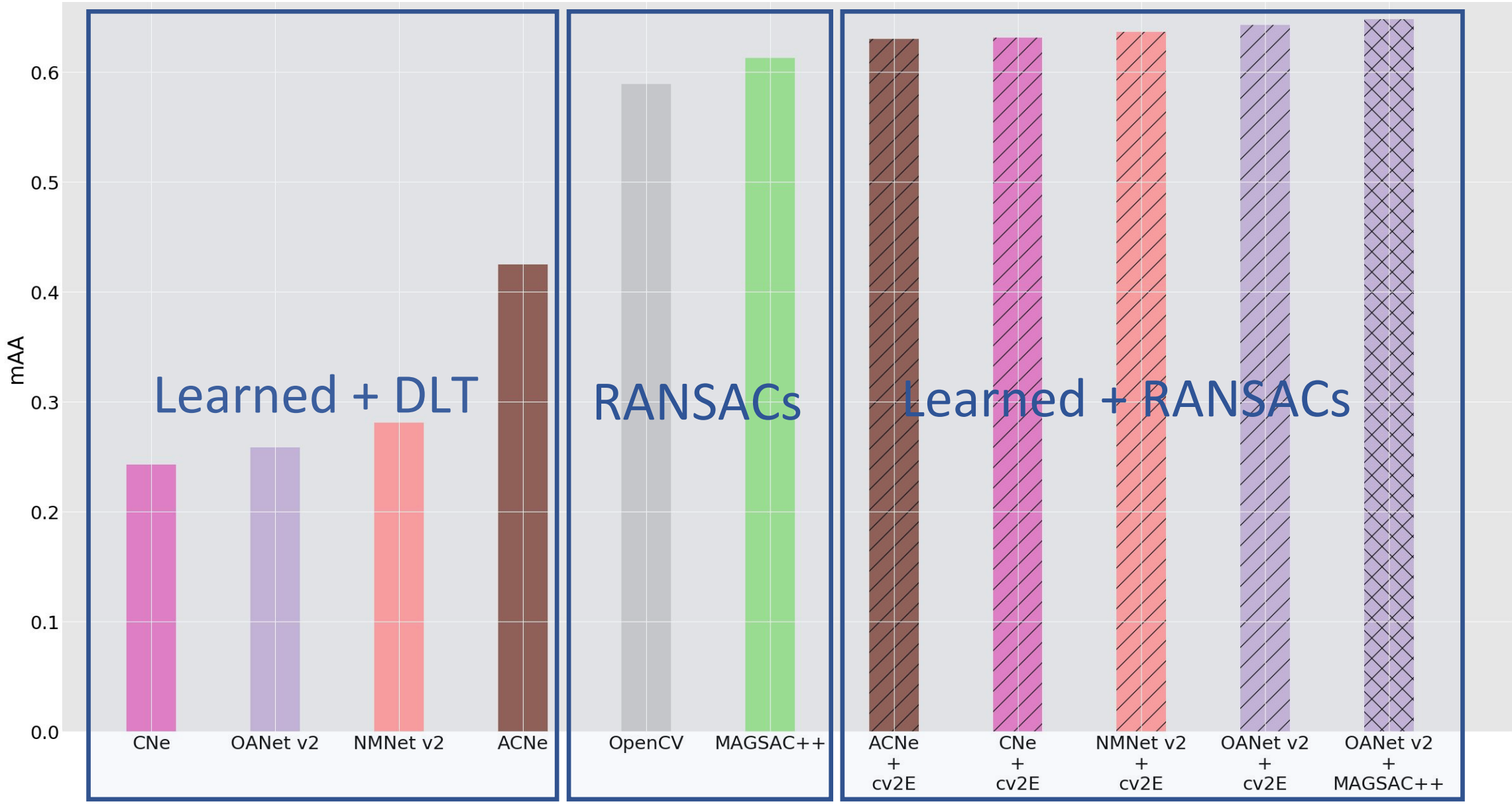
- A-Contrario RANSAC needs tuning as well as other RANSACs. Without, it performs very poorly
- USAC is worse than even OpenCV

Classical F: 1k iterations vs 100k iterations



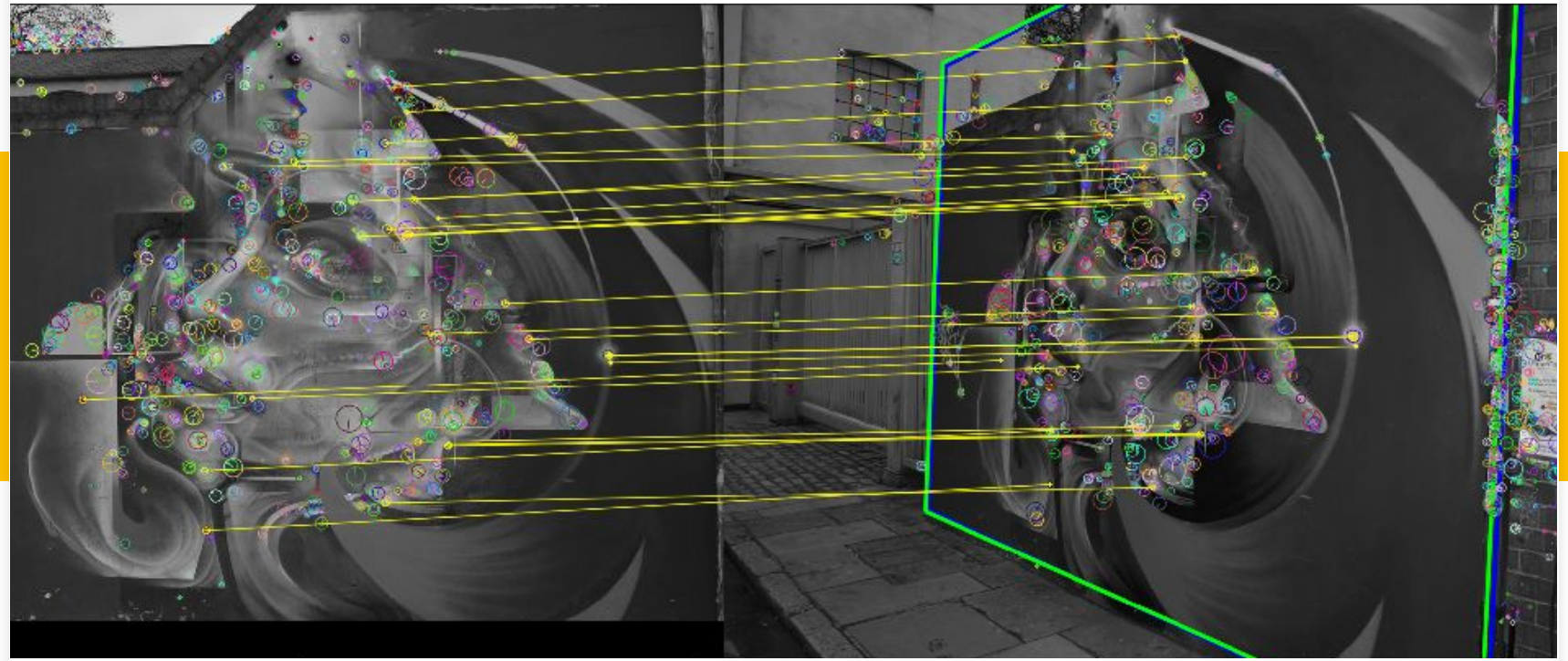
The benefit of having 100x more iterations is quite small

Learned methods E



E-summary

- Deep learned methods are not replacements for the RANSAC, they are replacement for the hand-crafted correspondence pre-filtering like Lowe SNN ratio.
- Performance of different RANSACs for E estimation varies, but not as significant, as for F
- Don’t use sk-image E-RANSAC, instead use MAGSAC, GC-RANSAC or OpenCV

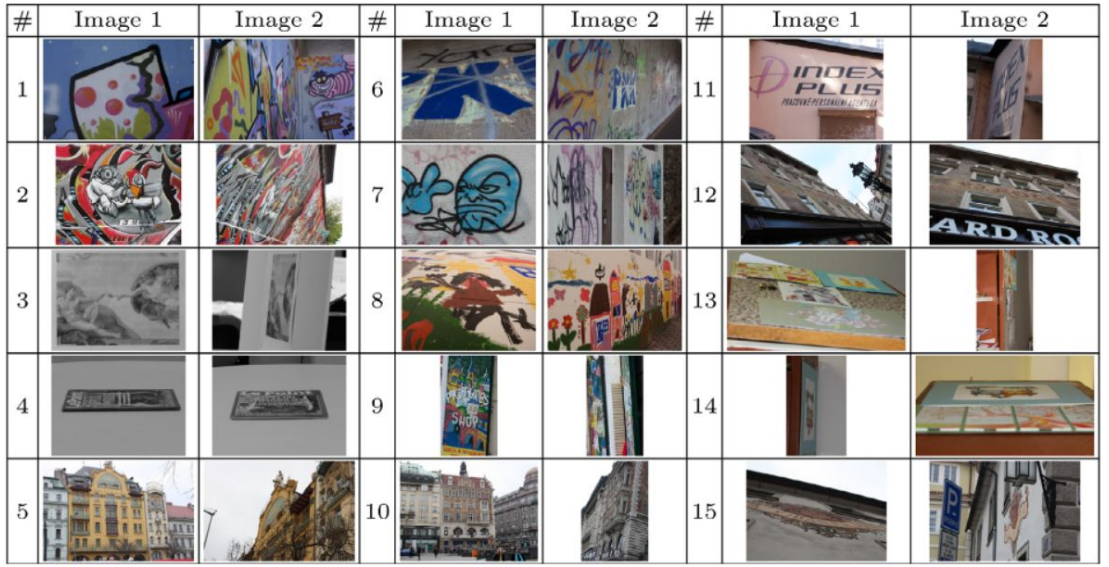


Homography estimators

Datasets: Hpatches-Sequences Viewpoints & EVD

- Problems:
  - There is no large-scale and diverse real-world homography dataset
  - If we are going to synthetic training data (by augmentation), then the best method is the one with closest statistics to the test set
- Our (temporary) solution:
  - Use HPatches Sequences (57x5 pairs) and Extreme View Dataset (15 pairs) with random validation/test splits 50/50, without training set

HPatches: A benchmark and evaluation of handcrafted and learned local descriptors
Vassileios Balntas, Karel Lenc, Andrea Vedaldi and Krystian Mikolajczyk, CVPR 2017.

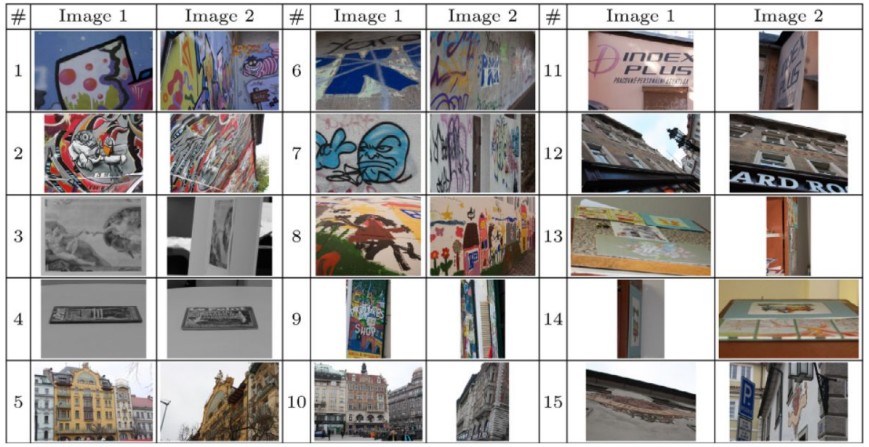


MODS: Fast and Robust Method for Two-View Matching.
D. Mishkin and J.Matas, and M.Perdoch. CVIU 2015

Input data

- HPatches sequences: RootSIFT features
  - Matching: mutual nearest neighbour
  - Additional info: SNN Lowe ratio

- EVD: MSER-RootSIFT and Hessian-Affine-RootSIFT MODS-generated correspondences with view synthesis.
  - Matching: uni-directional nearest neighbor
  - Additional info: SNN Lowe ratio



Metric: mAA@20 px reprojection error

The image mask is reprojected to the 2nd image
and back to get the commonly visible area.

Mean reprojection error is averaged over the visible area and then thresholded to get the mAA. Thresholds are from 1 to 20px in logspace

Evaluated methods: classical H

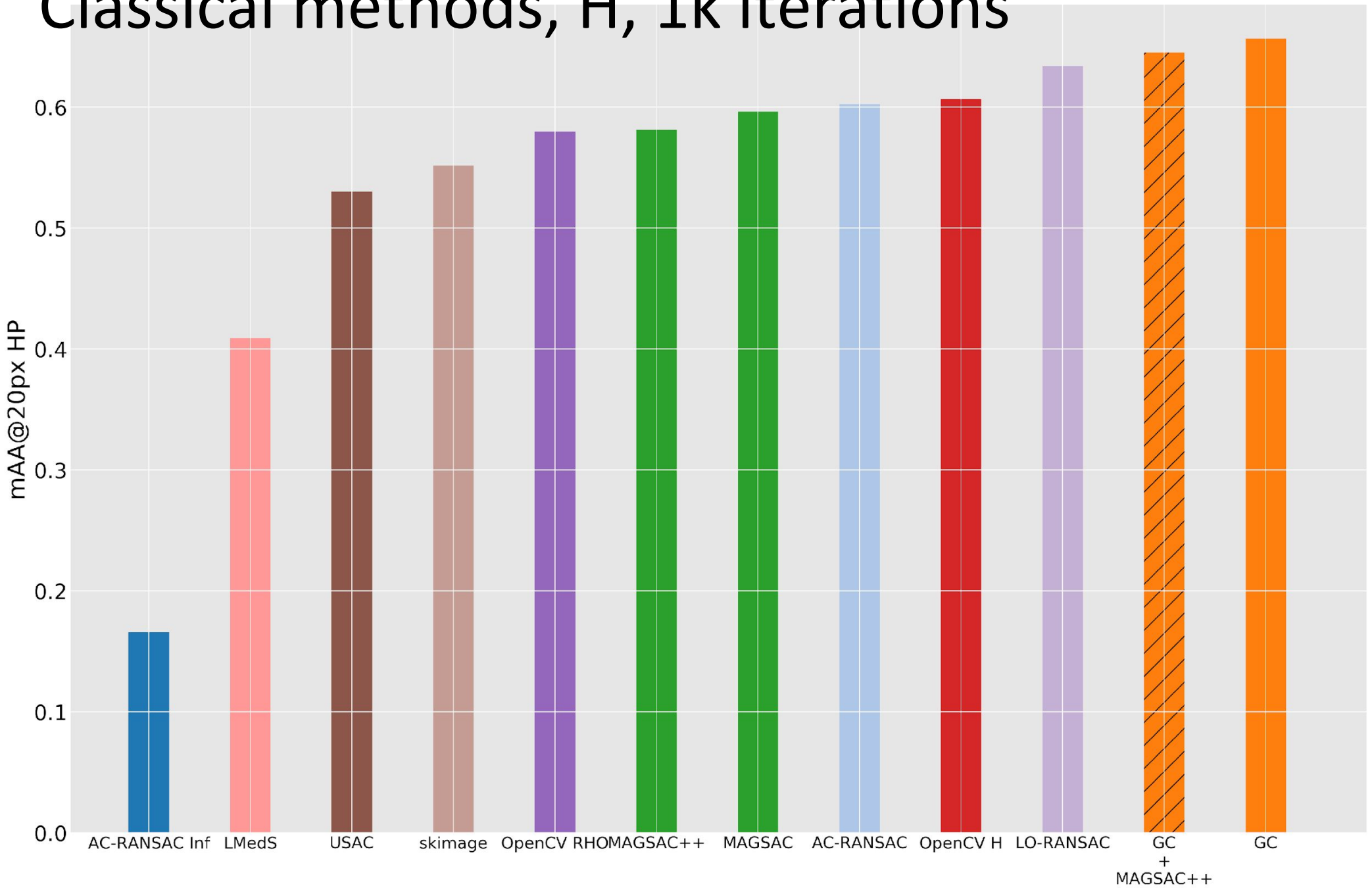
Baselines

- scikit-image: 4pt vanilla RANSAC + final least squares fitting
- OpenCV RANSAC: 4pt vanilla RANSAC + LM final fitting
- OpenCV LMeds: least medians
- OpenCV RHO: Fast Target Recognition on Mobile Devices: Revisiting Gaussian Elimination for the the Estimation of Planar Homographies
- PyRANSAC: 4pt LO-RANSAC + final least squares fitting

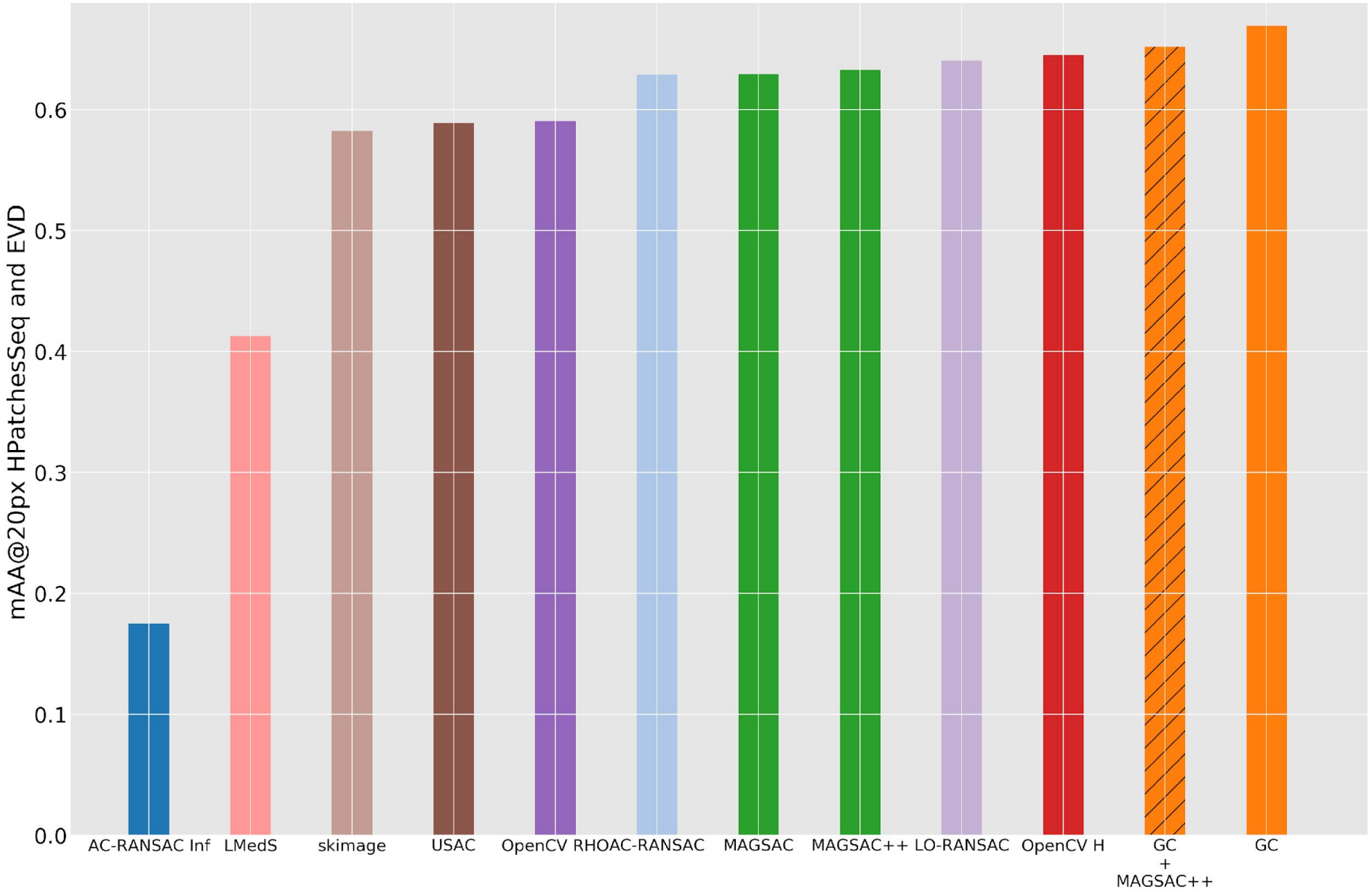
Advanced methods

- GC-RANSAC: Graph-Cut RANSAC
- MAGSAC: marginalizing sample consensus
- MAGSAC++ MAGSAC++, a fast, reliable and accurate robust estimator
- USAC: A Universal Framework for Random Sample Consensus
- AC RANSAC: Automatic Homographic Registration of a Pair of Images with A Contrario Elimination of Outliers
- GC-RANSAC + MAGSAC++

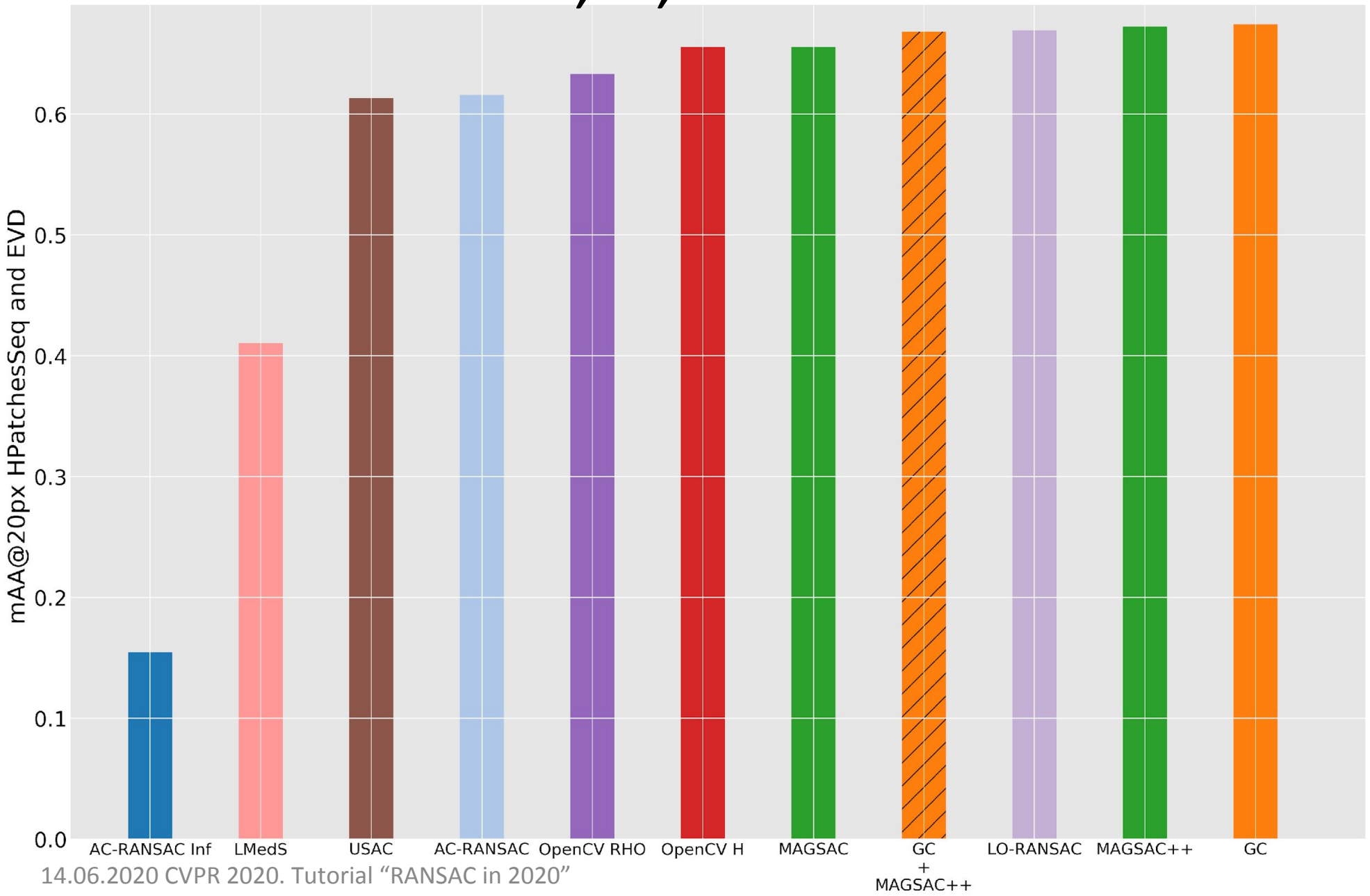
Classical methods, H, 1k iterations



Classical methods, H, 10k iterations

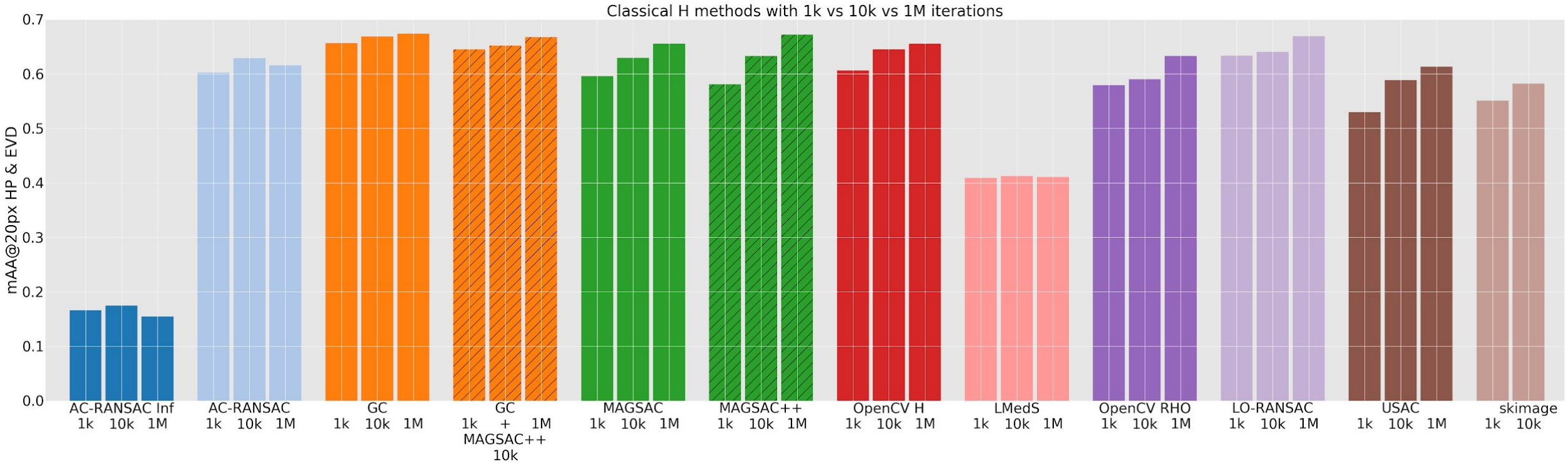


Classical methods, H, 1M iterations



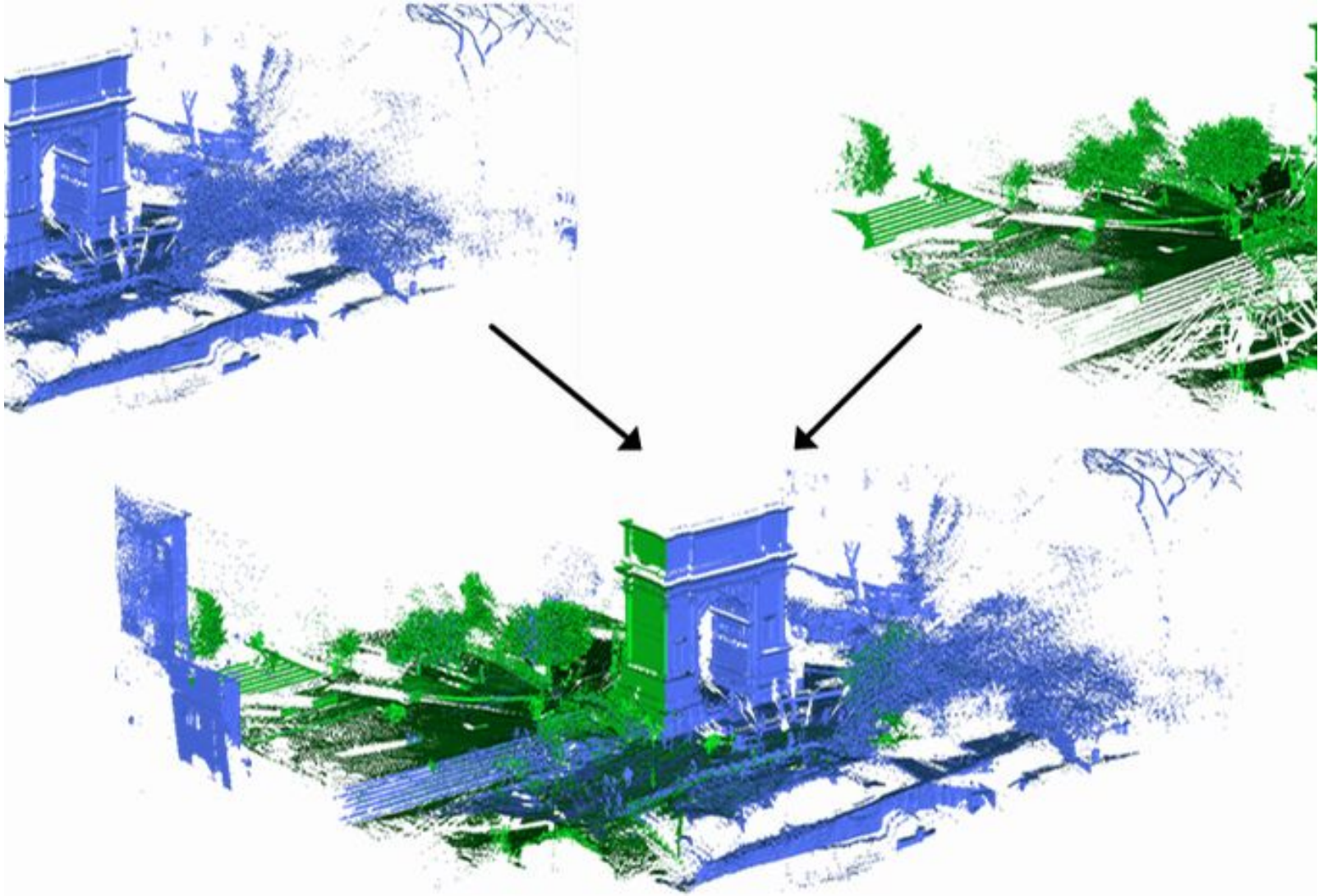
Hpatches-Sequences Viewpoints & EVD
1k vs 10k vs 1M iterations for H

It is better to use more advances RANSAC, than go for the 1M iterations



H-summary

- Performance of different RANSACs varies significantly, and all the methods have to be tuned to perform well
- Don’t use USAC, sk-image H-RANSACs, OpenCV RHO.
- Use GC-RANSAC, LO-RANSAC, MAGSAC or OpenCV H



Rigid PCL registration estimators

Point Cloud Registration. Datasets:

- **RGB-D Dataset 7-Scenes + RGBD Object datasets: 4 scenes, median of 10 runs**
- **ETHZ IGP datasets: 22 scenes, 1 run (large scale, slow)**

  - Due to the small size of the datasets and their different statistics, there is no val/test split.
  - We have run algorithms with “reasonable defaults”, but…take results with a grain of salt.
  - If you would like to do a proper benchmark for the 3D point cloud registration, please contact us.

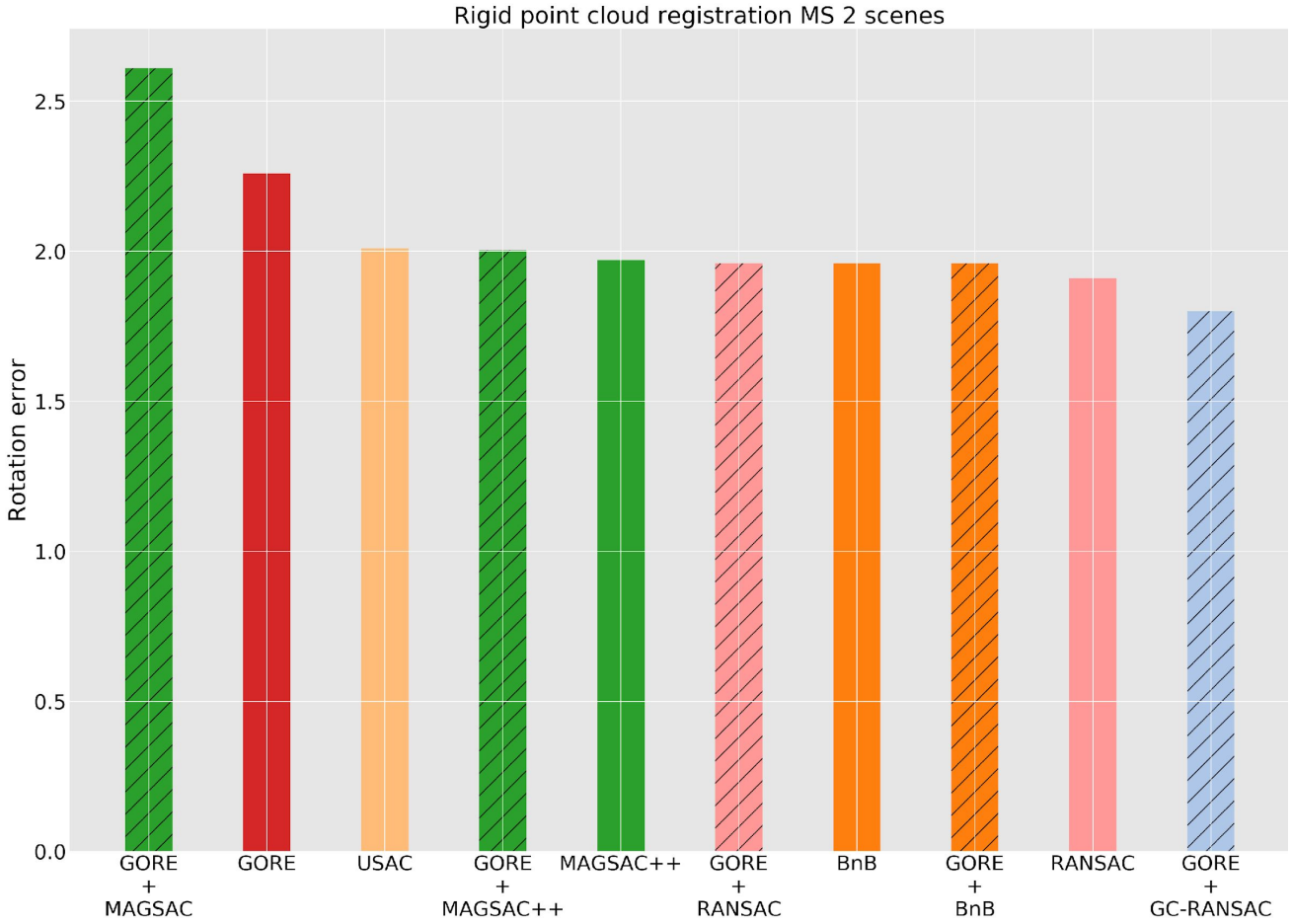
Point Cloud Registration. Metrics

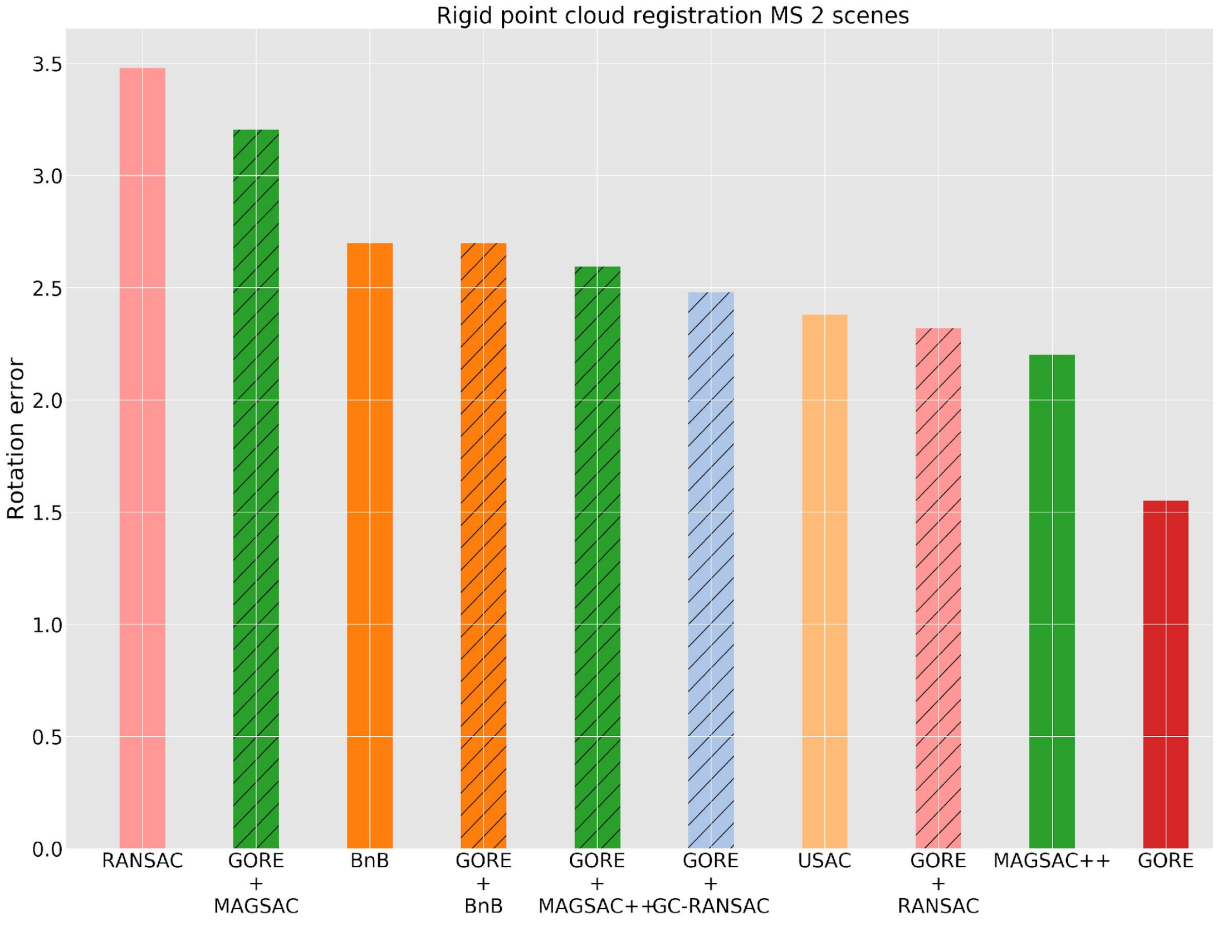
- Mean rotation error [deg]
- Mean translation error [m]
- Time [s]

Point Cloud Registration. Evaluated methods

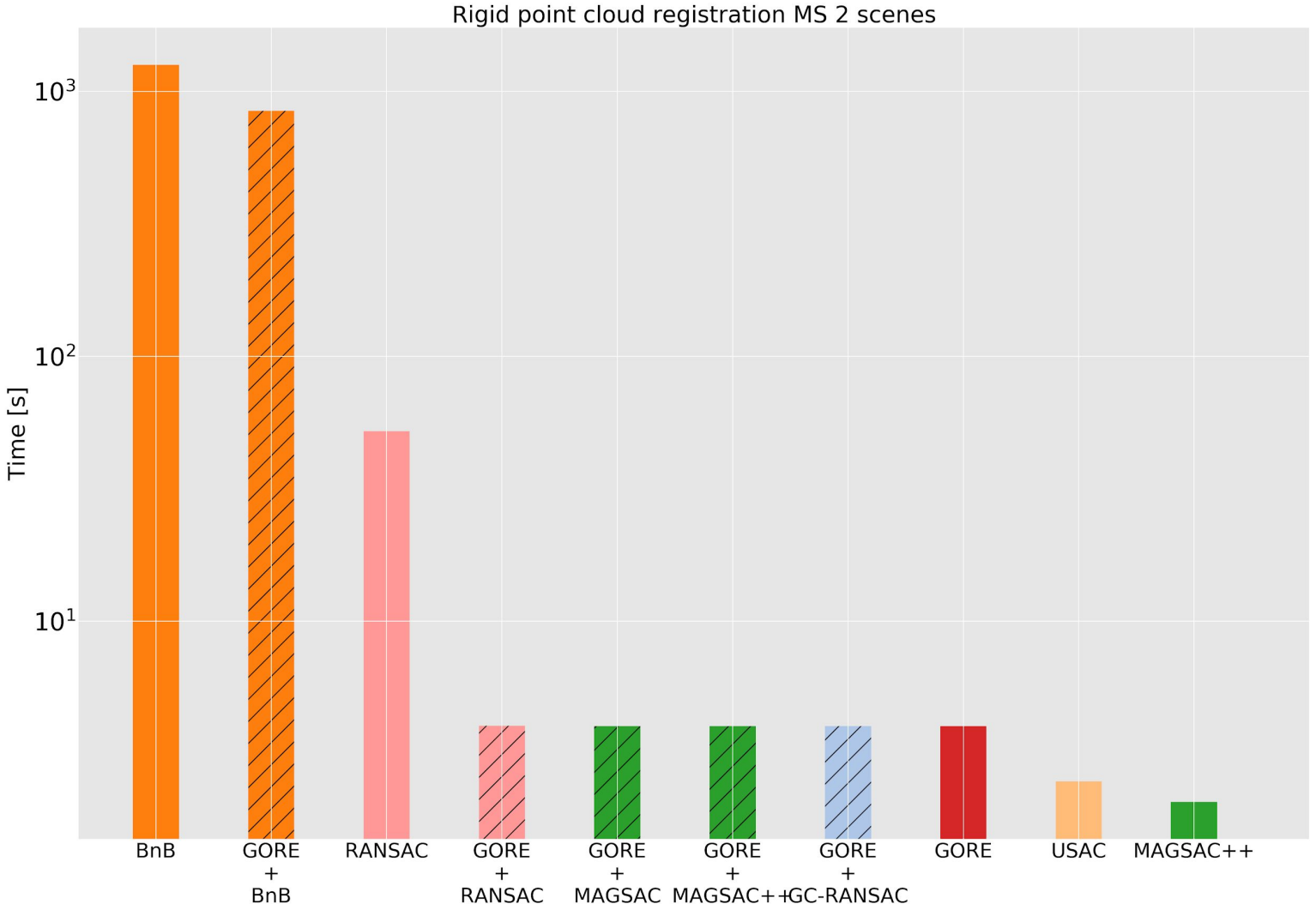
- GORE: Guaranteed Outlier Removal for Point Cloud Registration with Correspondences
- GC-RANSAC: Graph-Cut RANSAC
- MAGSAC: marginalizing sample consensus
- MAGSAC++, a fast, reliable and accurate robust estimator
- USAC: A Universal Framework for Random Sample Consensus
- GC-RANSAC + MAGSAC++
- GORE + (*SACs)
- TEASER: Fast and Certifiable Point Cloud Registration
- Branch and Bound (BnB)

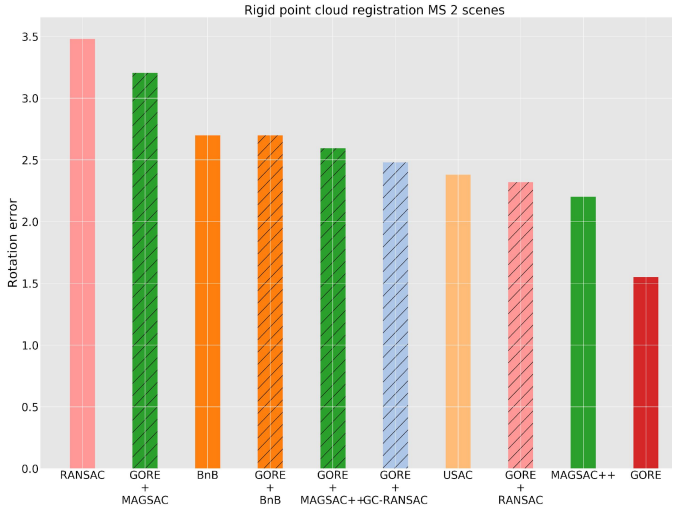
Point Cloud Registration. Test on MS Scenes

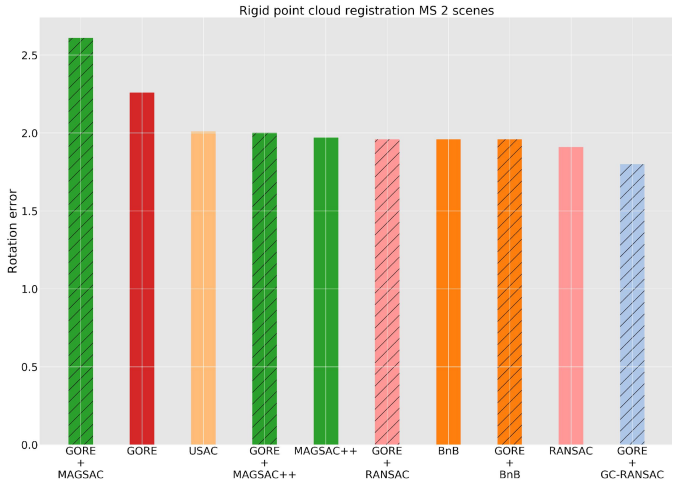




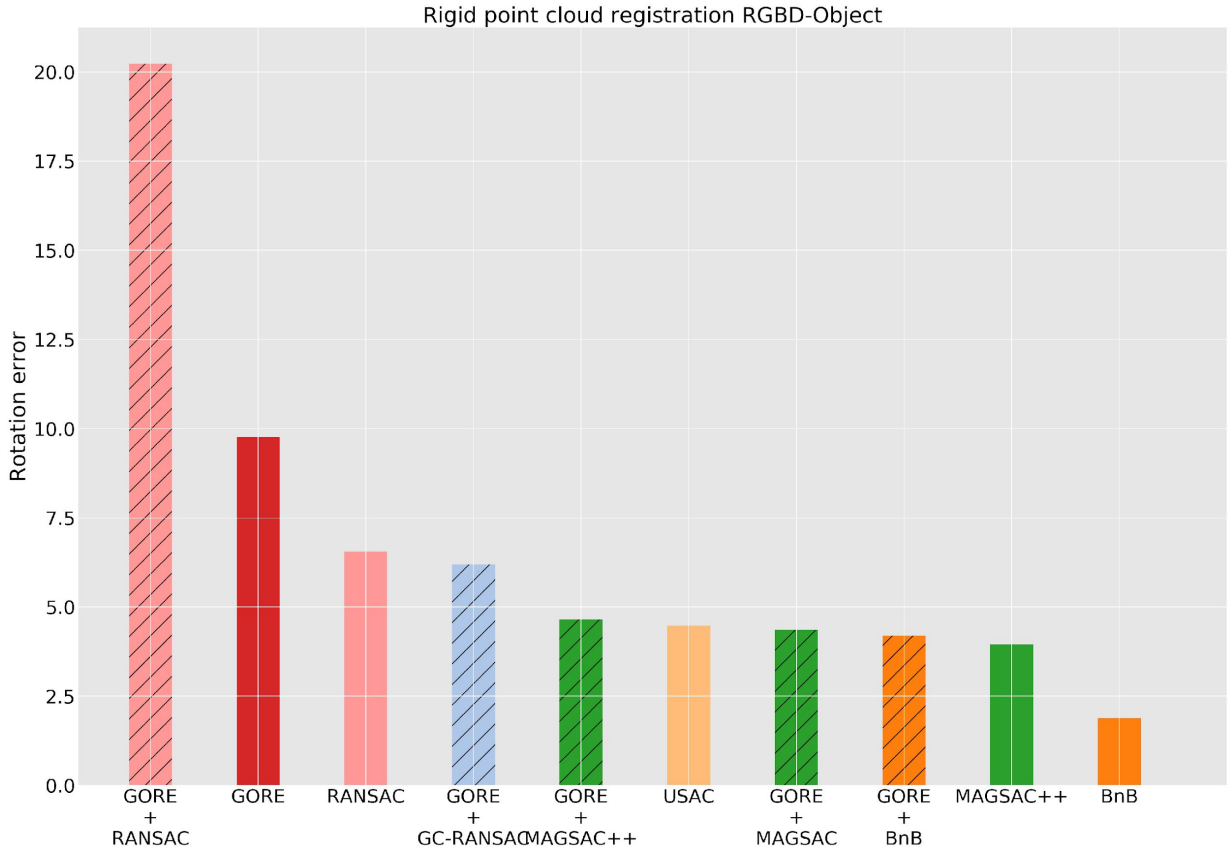
Point Cloud Registration. Test on MS Scenes: Runtime

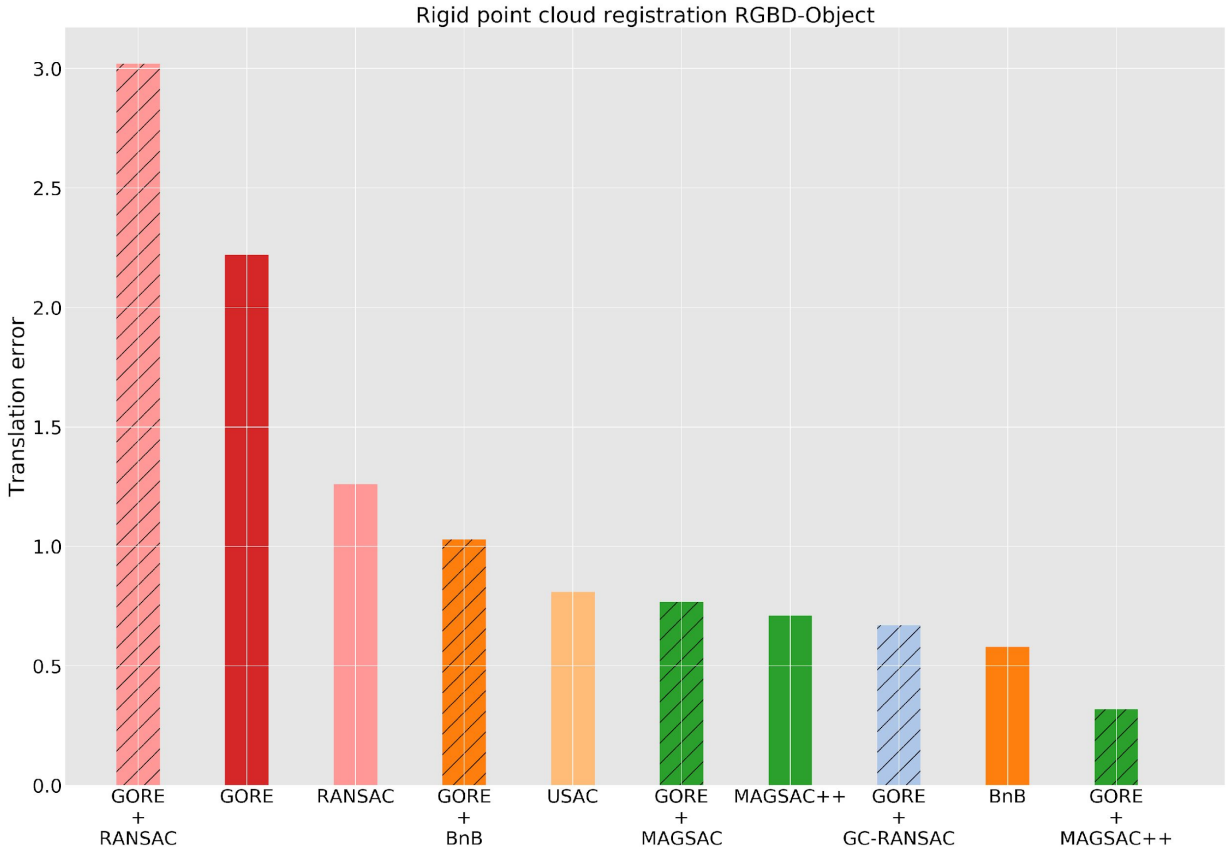




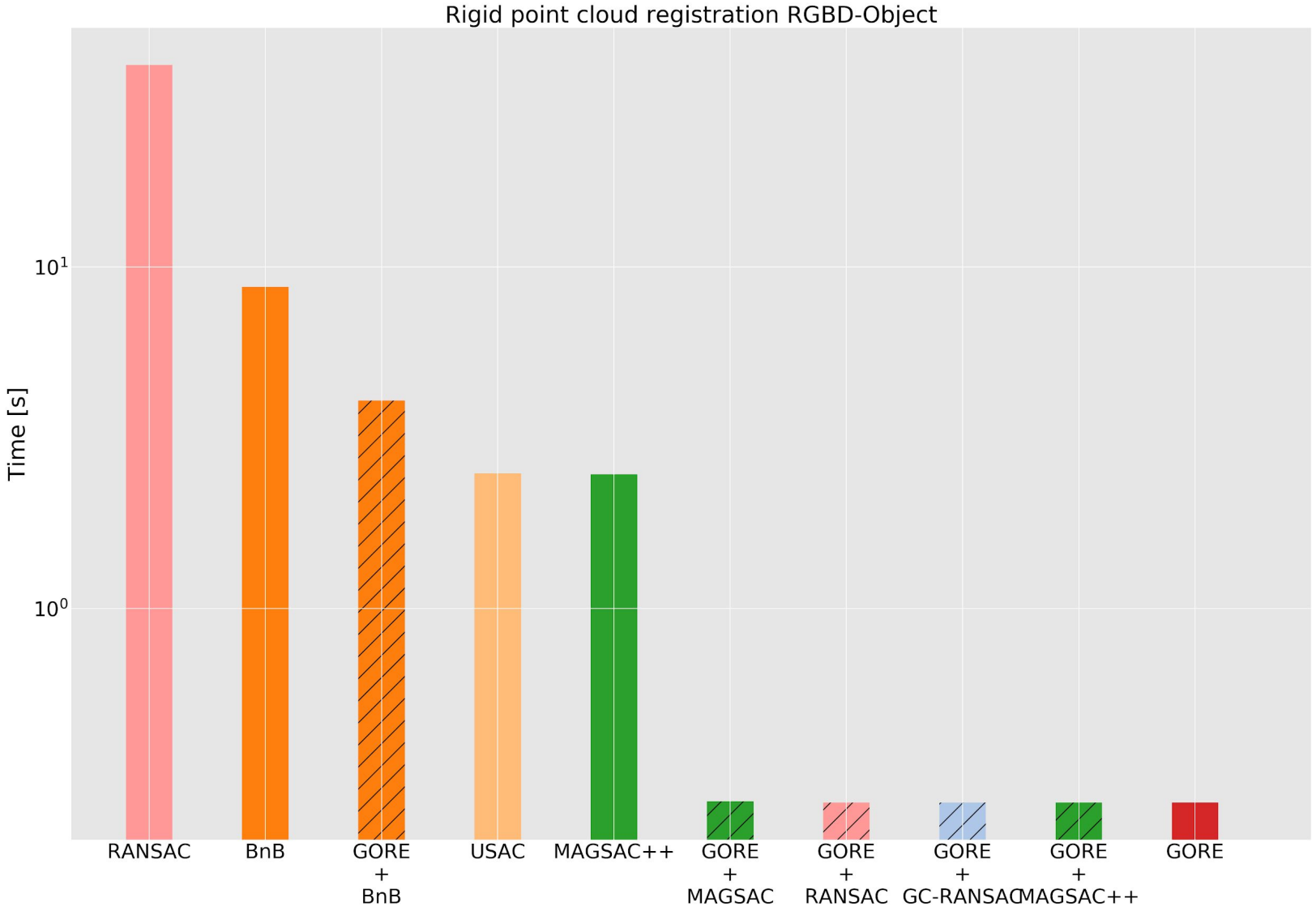


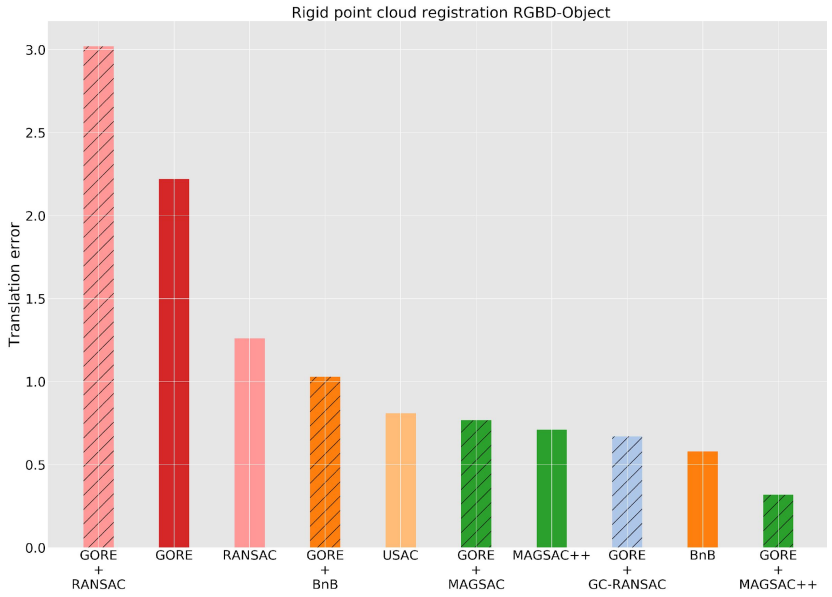
Point Cloud Registration. Test on RGBD Objects

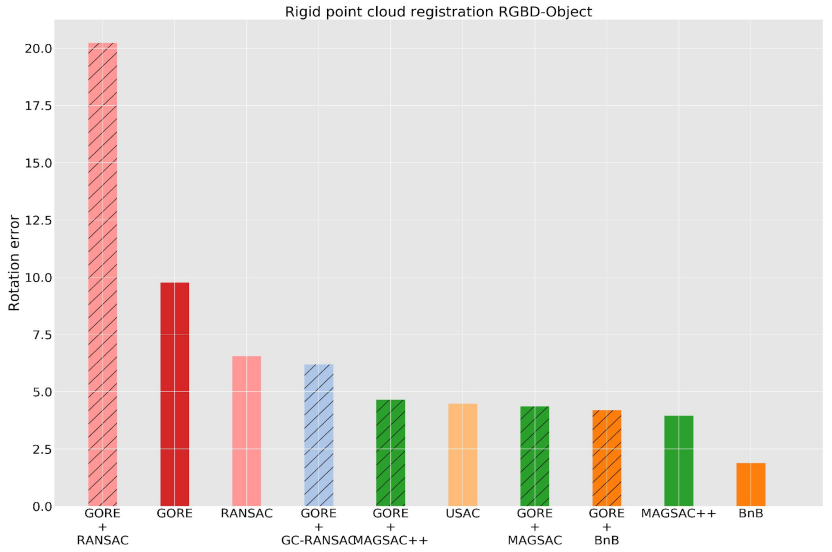




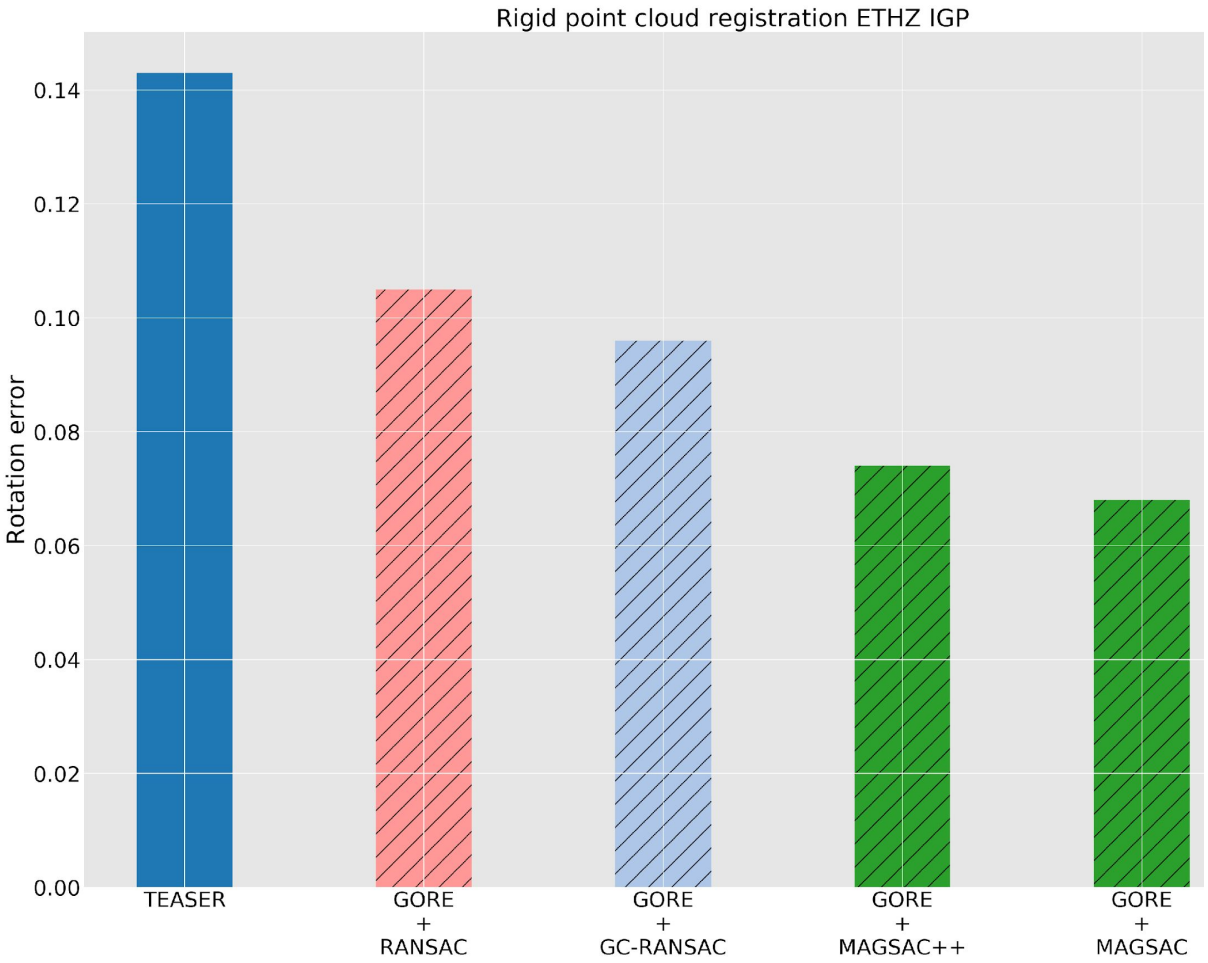
Point Cloud Registration. Test on RGBD Objects: runtime

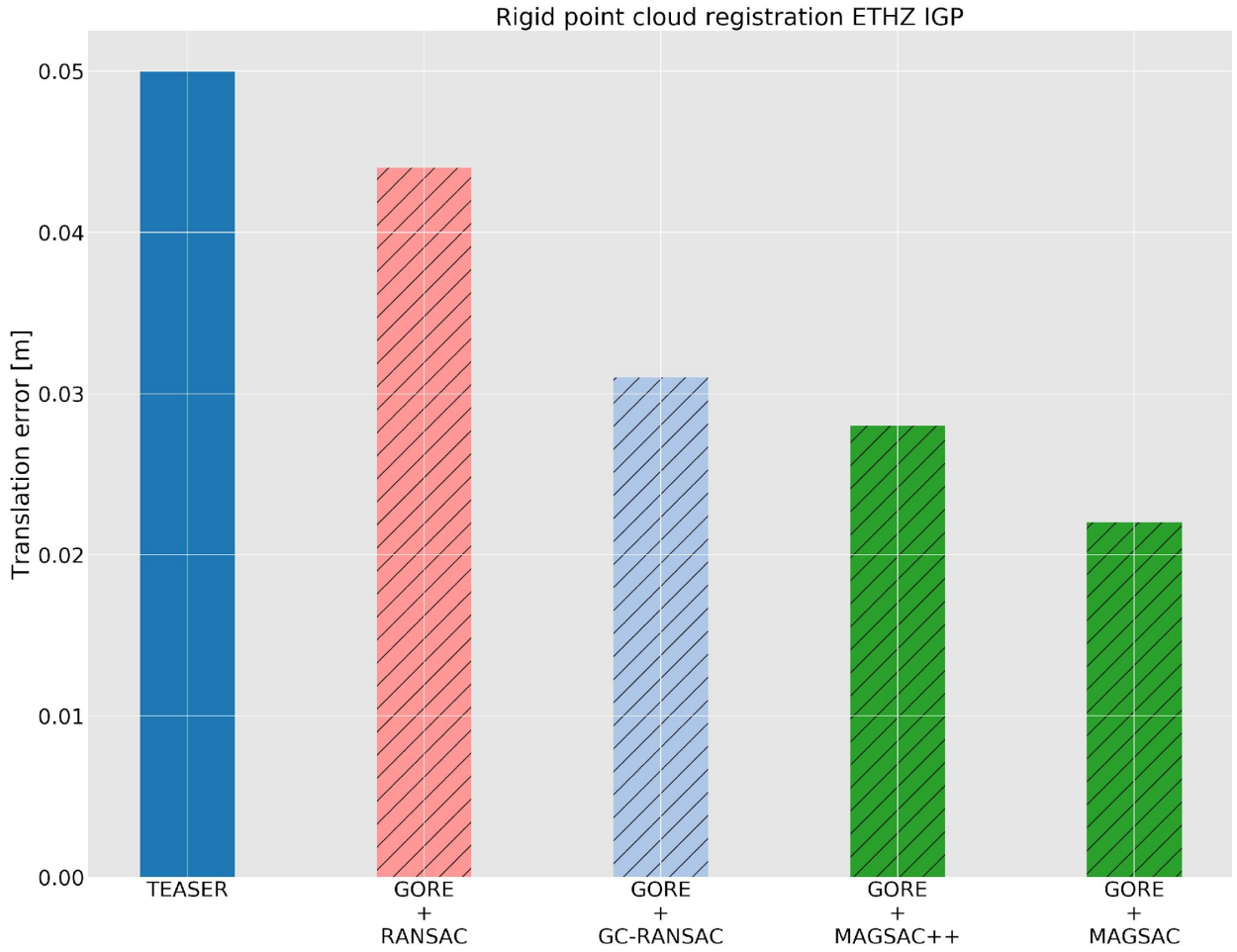






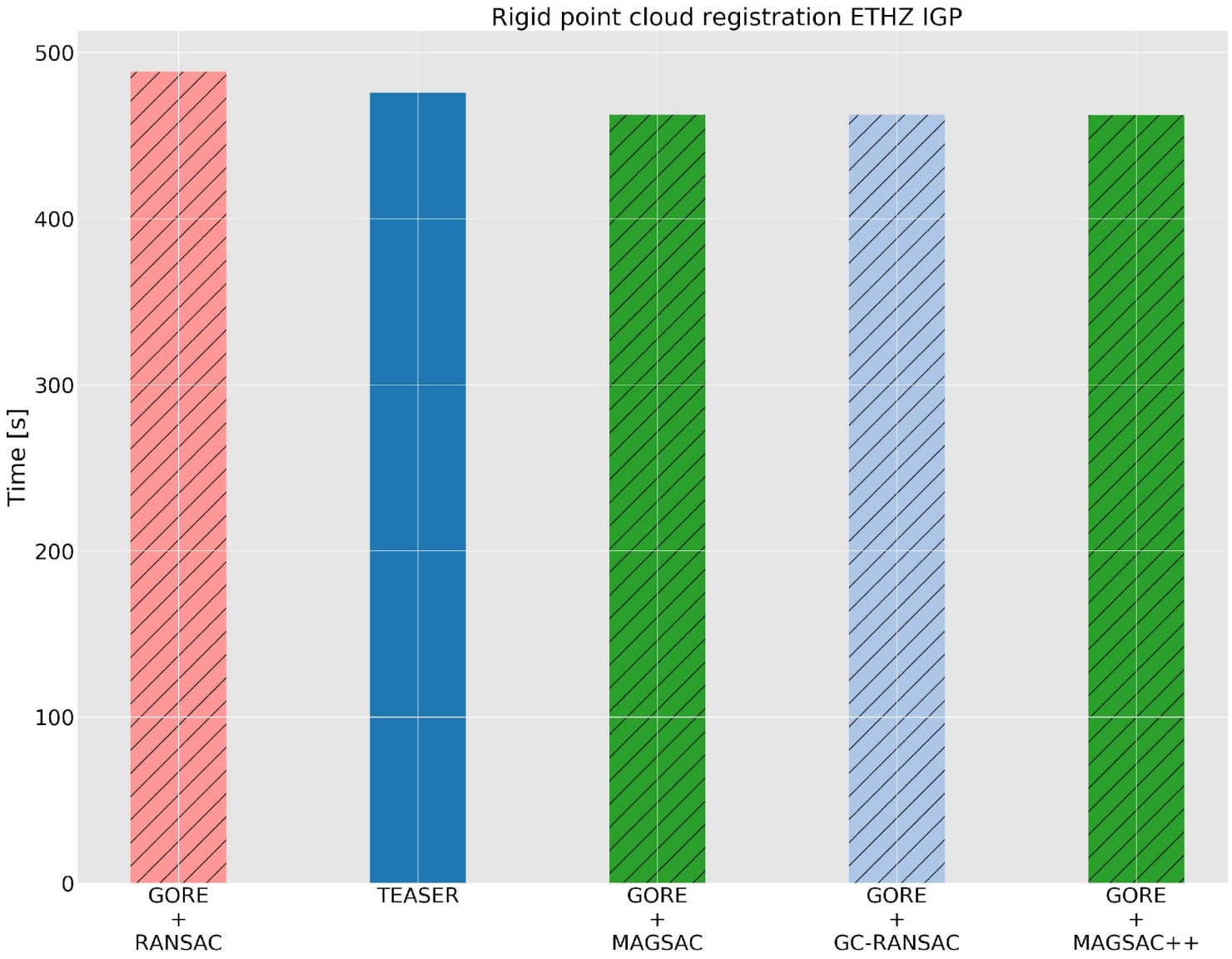
Point Cloud Registration. Test on ETHZ IGP

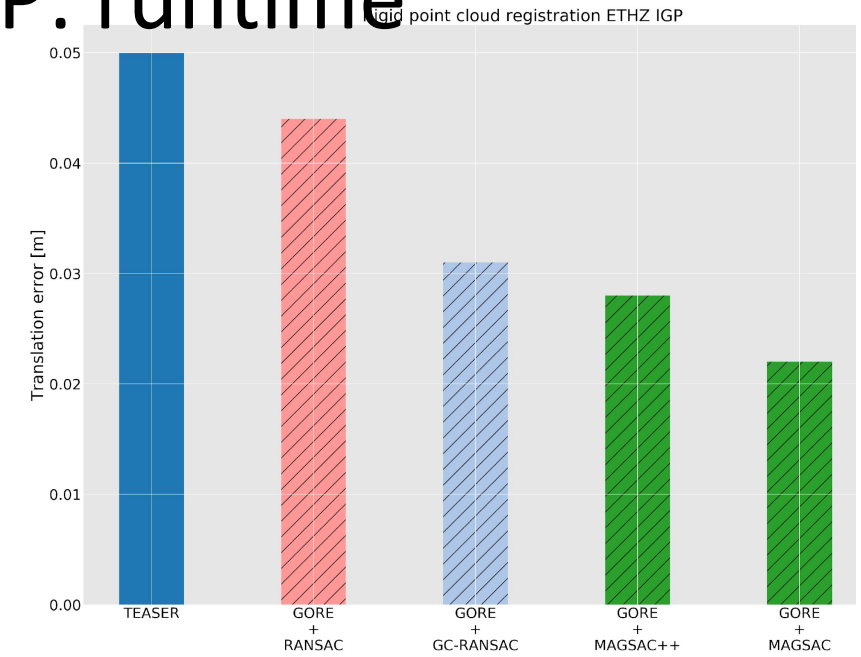


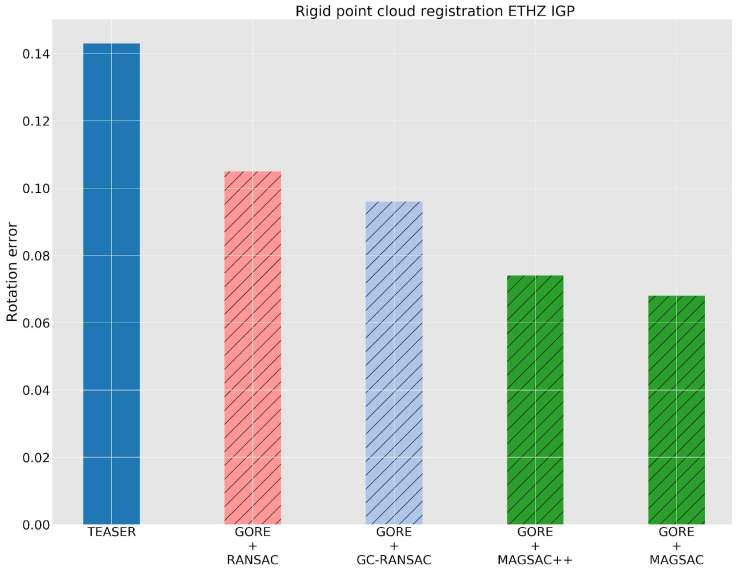


RANSAC does not work here without GORE due to low inlier ratio

Point Cloud Registration. Test on ETHZ IGP: runtime







Rigid conclusions

- There is no cheap pre-filtering method like Lowe`s SNN ratio check here
- For some problems you need to use GORE to get the answer at all.
- Good place to put deep learning methods as pre-filters

Overall conclusions

- One still cannot do (yet?) robust estimation without RANSAC
- For further progress of learned methods we need new large scale datasets for training and evaluation
- For any method – we need a protocol to evaluation and train/val/test splits
- Implementation matters a lot.
- Use GC-RANSAC + deep pre-filtering methods, if available.

Thank you for the attention. Questions?