A Large-Scale Homography Benchmark

Daniel Barath[1], Dmytro Mishkin[2,5], Michal Polic[2,3], Wolfgang Förstner[4], Jiri Matas[2]
[1]Computer Vision and Geometry Group, ETH Zurich, Switzerland,
[2]VRG, Faculty of Electrical Engineering, CTU in Prague, Czech Republic,
[3]CIIRC, CTU in Prague, Czech Republic, [4]University Bonn, Germany, [5]HOVER Inc.

Abstract

We present a large-scale dataset of Planes in 3D, Pi3D, of roughly 1000 planes observed in 10 000 images from the 1DSfM dataset, and HEB, a large-scale homography estimation benchmark leveraging Pi3D. The applications of the Pi3D dataset are diverse, e.g. training or evaluating monocular depth, surface normal estimation and image matching algorithms. The HEB dataset consists of 226 260 homographies and includes roughly 4M correspondences. The homographies link images that often undergo significant viewpoint and illumination changes. As applications of HEB, we perform a rigorous evaluation of a wide range of robust estimators and deep learning-based correspondence filtering methods, establishing the current state-of-the-art in robust homography estimation. We also evaluate the uncertainty of the SIFT orientations and scales w.r.t. the ground truth coming from the underlying homographies and provide codes for comparing uncertainty of custom detectors. The dataset is available at https://github.com/danini/homography-benchmark.

1. Introduction

The planar homography is a projective mapping between images of co-planar 3D points. The homography induced by a plane is unique up to a scale and has eight degrees-of-freedom (DoF). It encodes the intrinsic and extrinsic camera parameters and the parameters of the underlying 3D plane.

The homography plays an important role in the geometry of multiple views [30] with hundreds of papers published in the last few decades about its theory and applications. Estimating planar homographies from image pairs is an important task in computer vision with a number of applications. For instance, monocular SLAM systems [55, 63, 70] rely on homographies when detecting pure rotational camera movements, planar scenes, and scenes with far objects. As a homography induced by a plane at infinity represents rotation-only camera motion, it is one of the most important tools for stitching images [1, 15]. The generated images cover a larger field-of-view and are useful in various applications, *e.g.* image-based localization [3], SLAM [34, 39], autonomous driving [65], sport broadcasting [17], surveillance [68], and augmented and virtual reality [33, 44]. Homographies play an important role in calibration [18, 73], metric rectification [21, 41], augmented reality [58, 76], optical flow based on piece-wise planar scene modeling [67], video stabilization [28, 77], and incremental [56] and global [48, 62] Structure-from-Motion.

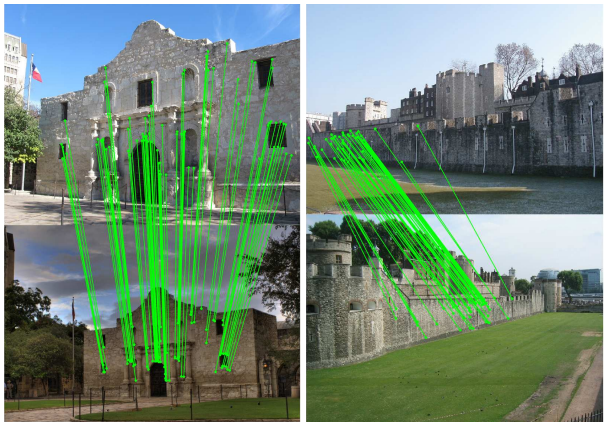

Figure 1. Example image pairs and homographies with their inlier correspondences shown, from the proposed Homography Estimation Benchmark (HEB) dataset. Outliers are not drawn.

The traditional approach of finding homographies in image pairs consists of two main stages. First, similarly as in most algorithms working with pairs, feature points are detected and matched [15, 35, 43, 54, 57]. They are then often filtered by the widely-used second nearest neighbors (SNN) ratio [42, 43] or by deep learned filtering methods [51, 61, 69, 75], to remove gross outliers and, therefore, improve the robust estimation procedure that follows. The found tentative point correspondences are contaminated by various sources of noise due to, *e.g.*, measurement and quantization, and a large proportion of them are still outliers – correspondences inconsistent with the sought model manifold. Consequently, some form of robust estimation

(a) Homogr Dataset (b) HPatches Dataset (c) ExtremeView Dataset (d) Proposed HEB Dataset

Figure 2. Typical image pairs (a-c) from widely used datasets for homography estimator benchmarking and (d) from HEB.

has to be applied to find a set of inliers and to estimate the parameters of the sought homography. In practice, either a randomized RANSAC-like [24] robust estimator or an iteratively re-weighted least squares fitting [31] is applied.

The number of datasets on which recent homography and, in general, robust estimation papers evaluate their algorithms is severely limited. The Homogr dataset [38] consists only of a few image pairs with relatively small baselines and, thus, high inlier ratios. Given that recent robust estimators, *e.g.* [6], report lower than 0.5 pixel average reprojection errors on the provided manually labeled correspondences, it is safe to say that this dataset is solved. The HPatches dataset [4] consists of a few hundreds of image pairs, all looking at an almost completely planar scene, with either significant illumination or viewpoint (mostly in tilt angle) changes. While [4] is a useful tool for evaluating local feature detector and image matching methods, it is very easy for robust estimators [5]. The ExtremeView (EVD) dataset [46] poses a significantly more challenging problem for homography estimation than the previous two. The images undergo extreme view-point changes, therefore making both the feature matching and robust estimation tasks especially challenging. However, EVD consists only of 15 image pairs, severely limiting its benchmarking power.

Besides the data part, a good benchmark has well-defined parameter tuning (training) and evaluation protocols and training-test set split. Otherwise, as it happens in other fields, the seemingly rapid progress might be an artifact of tuning the algorithms on the test data, or an artifact of the flawed evaluation procedure [12, 27, 49].

In short, there are no available large-scale benchmarks with ground truth (GT) homographies that allow evaluating new algorithms on standard internet photos, *i.e.*, ones not necessarily looking at completely planar scenes.

As the *first contribution*, we create a large-scale dataset of 1046 large Planes in 3D (Pi3D) from a standard landmark dataset [66]. We use the scenes from the 1DSfM dataset as input and find 3D planes in the reconstructions. *Second*, we use the Pi3D dataset to find image pairs with estimatable homographies and create a large-scale homography benchmark (HEB) containing a total of $226\,260$ homographies that can be considered GT when testing new algorithms (see Fig. 1 for examples). A large proportion of the image pairs capture significant viewpoint and illumination changes. The homographies typically have low inlier ratio, thus making the robust estimation task challenging. *Third*, we compare a wide range of robust estimators, including recent ones based on neural networks, establishing the current state-of-the-art in robust homography estimation. As the *forth* contribution, we demonstrate that the dataset can be used to evaluate the uncertainty of partially or fully affine covariant features detectors [43, 47]. While we show it on DoG features [42], the homographies can be leveraged similarly for the comparison with other detectors.

Existing Datasets. The datasets traditionally used for evaluating homography estimators are the following. The **Homogr** dataset [38] consists of 16 image pairs with GT homographies. The GT comes from (also provided) hand-labeled correspondences, which later were optimized to improve the localization accuracy. There is no train-test split, nor a benchmark protocol. The **ExtremeView** dataset [46] consists of 15 image pairs, taken under extreme viewpoint change, together with GT homographies and correspondences. The homographies are derived from hand-labeled correspondences that stem from multiple local feature detectors paired with an affine view synthesis procedure [46] and RootSIFT descriptor [2] matching. There is no train-test split, nor a benchmark protocol. The **HPatches** dataset [4] was introduced in form of local patches for benchmarking descriptors and metric learning methods,

Dataset	# image pairs	train-test split	camera pose	scene type	baseline	illumination change	inlier ratio
Homogr [38]	16	✗	✗	buildings	short/medium	✗	high
ExtremeView [46]	15	✗	✗	walls	large	✗	low
HPatches [4]	$(59 + 57) \times 5$	✓	✗	walls	short/medium	✗ + ✓	high
HEB	226 260	✓	✓	landmark photos	diverse	✓	low

Table 1. Comparison of the existing and the proposed HEB homography estimation datasets.

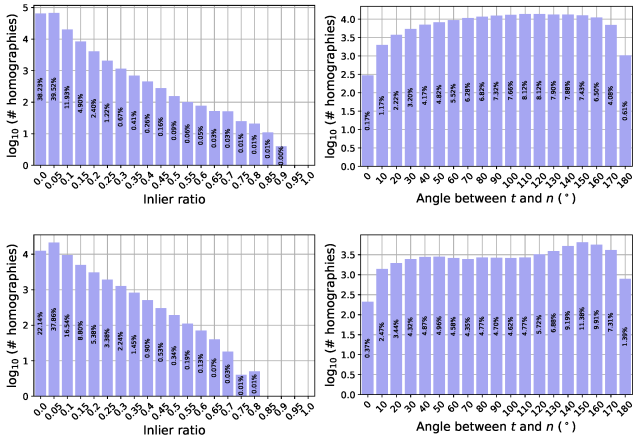

Figure 3. **HEB properties**: test (top; 169 654 pairs) and training (bottom; 56 593 pairs) splits. Percentages are written inside the bars. When calculating the angle between the translation and plane normal, the sign of the normal is set so it looks towards the camera.

later extended with images and homographies. It consists of 57 image sextuplets with significant illumination but negligible viewpoint changes and 59 ones with viewpoint, but no illumination changes. The viewpoint difference mostly consist of tilt (perspective change) in the horizontal direction and some shift – no big rotation or scale changes. The GT was obtained from manually annotated correspondences for the initial model estimation and polished by minimizing MSE of image pixel re-projections. There is no official protocol, nor standard correspondences for homography evaluation – every paper uses slightly different evaluations, but there is an official train-test split.

To conclude, there is no difficult-enough, large-scale dataset with train-test split and benchmark protocol for evaluating robust homography estimation. Table 1 summarizes the properties of each publicly available dataset and, also, that of the proposed one. Typical image pairs from the datasets are shown in the first three columns of Fig. 2.

2. Planes in 3D Dataset

The Planes in 3D Dataset is based on images from the 1DSfM dataset [66]. The objective of this section is to create a large-scale dataset of 3D planes in scenes consisting of thousands of real-world photos. 1DSfM consists of 13 scenes of landmarks with photos of varying sizes collected from the internet. It provides 2-view matches with epipolar geometries and a reference reconstruction from incremental SfM (with Bundler [59, 60]) for measuring error. Instead, we reconstructed the scenes with COLMAP [56], providing more accurate reconstruction [36]. Incremental SfM (*e.g.*, COLMAP) results are often considered GT, *e.g.* in IMC [35], as they are the best which we can get from internet images. We manually checked all reconstructions ensuring that only those scenes are used where COLMAP returned an accurate and coherent reconstruction. We, thus, excluded Gendarmenmarkt and Trafalgar.

We considered several options (*e.g.*, IMC [35] and MegaDepth [40]) before deciding to use 1DSfM. We chose it since, nowadays, it is rarely used in computer vision, likely, due to the attached Bundler reconstruction (we replaced it with COLMAP). Thus, introducing it back to the community is preferable to keep the variety of commonly used datasets, and not overfitting to IMC, which is only twice bigger than 1DSfM.

Let us introduce the concept of "estimatable homographies". An "estimatable homography" is a homography that links two views of a real 3D planar surface; it is consistent with the camera motion; and it is estimatable from its GT correspondences by the standard normalized DLT algorithm [30]. We keep only those planes in the Pi3D dataset that imply at least a single estimatable homography – planes that are visible and estimatable in at least an image pair. The steps of the pipeline finding such planes and homographies:

1. COLMAP reconstructs the scene from the images.
2. Multiple 3D planes are detected in the COLMAP point cloud reconstruction.
3. For each 3D plane, all image pairs where the plane is visible are selected.
4. A homography is estimated from each 3D plane in each image pair, where it is visible, using the camera parameters, *i.e.*, the poses and intrinsic matrices.
5. A homography is rejected if it can not be estimated from only the assigned GT point correspondences, without the camera parameters, accurately.

Multiple Planes in the Reconstruction. The first step of the pipeline is to find 3D planes that can be used when finding planar regions in image pairs. For this purpose, we use the Progressive-X$^+$ algorithm [10]. To ensure that only dominant planes are found in the reconstruction, we use the following parameters: $n_{\text{min}} = 5000$ and $\epsilon_{\text{T}} = 0.1$. Parameter n_{min} is the number of inliers a plane needs to be considered as a dominant one. Parameter ϵ_{T} is the threshold for the pair-wise Tanimoto similarity of the plane consensus vectors. Briefly, the Tanimoto similarity measures how

similar two planes are in terms of their support. These parameters lead to plane segmentations with keeping only the dominant structures and suppressing small details.

Recovering Absolute Scale. The COLMAP reconstruction is scaleless, *i.e.*, the metric size of the scene is unknown. This is why prior work, *e.g.* [35], use angle-based metrics to compare camera translations recovered by image matching algorithms. Instead, we manually added the scale to the reconstructions. 3D points were re-projected on the images and a manual annotator picked those which are easily identifiable and far enough from each other, *e.g.*, the facade edges of the largest building. We then measured the distance with the ruler tool of Google Maps [26]. The ratio between these two gives the scaling coefficient to the 3D reconstruction. This procedure is repeated several times and the coefficients are averaged to get the final scale. The standard deviation of the manually picked absolute scales is approximately 16 cm, implying that the recovered scales are accurate.

Visible 3D Planes. First, we iterate through all possible image pairs (I_i, I_j), $i, j \in [0, p)$, from the COLMAP reconstruction of the scene, where $p = \binom{n}{2}$ and $n \in \mathbb{N}$ is the number of images. For each pair, we collect the planes that have more than ten 3D points visible in both views according to COLMAP depth maps. Second, we detect SIFT features [43] as implemented in OpenCV [14] with RootSIFT [2] descriptors. In each image, at most 8000 keypoints are detected and matched. We combine mutual nearest neighbor check to establish tentative point correspondences, as it is recommended in [35]. The SNN ratio is stored, but no correspondences are filtered out, because different robust estimators, either deep or traditional, may prefer different ratios to achieve their best performance.

Relative poses are calculated as $\mathbf{R} = \mathbf{R}_2\mathbf{R}_1^{\mathrm{T}}$ and $\mathbf{t} = \mathbf{t}_2 - \mathbf{R}_2\mathbf{R}_1^{\mathrm{T}}\mathbf{t}_1$, where $\mathbf{R}_1, \mathbf{R}_2 \in \mathrm{SO}(3)$ are the absolute rotations and $\mathbf{t}_1, \mathbf{t}_2 \in \mathbb{R}^3$ are the translations from the reconstruction. The parameters of the normalized homography implied by the plane are calculated as follows: $\mathbf{H} = \mathbf{R} - (\mathbf{t}\mathbf{n}^{\mathrm{T}})/d$, where $\mathbf{n} \in \mathbb{R}^3$ is the plane normal and d is its intercept. Correspondences are considered inliers if the re-projection error is less than ϵ pixels given homography $\mathbf{H}$. Homographies with fewer than 10 inliers are rejected. To make sure that the GT homography can be recovered from its inliers and they are not in a degenerate configuration, we estimate homography $\mathbf{H}'$ by the normalized DLT algorithm from the inliers. It is decomposed to rotation $\mathbf{R}'$ and translation $\mathbf{t}'$ by the standard procedure [45]. We reject homography $\mathbf{H}$ if either $\epsilon_{\mathbf{R}'} > 3°$ or $\epsilon_{\mathbf{t}'} > 3°$, where

$$\epsilon_{\mathbf{R}'} = (180/\pi) \arccos\left(\left(\mathrm{tr}\left(\mathbf{R}'\mathbf{R}^{\mathrm{T}}\right) - 1\right)/2\right) \quad (1)$$

is the rotation error and

$$\epsilon_{\mathbf{t}'} = (180/\pi) \arccos(\mathbf{t}^{\mathrm{T}}\mathbf{t}')(|\mathbf{t}||\mathbf{t}'|) \quad (2)$$

is the angular translation error in degrees [35]. This ensures

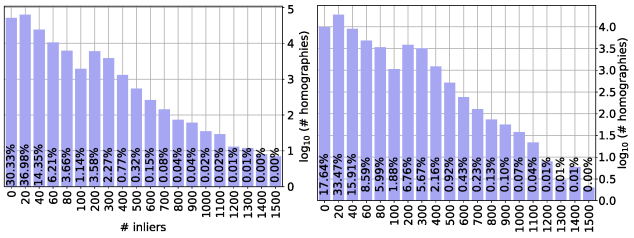

Figure 4. Inlier number distribution in the training (left) and test (right) set of the HEB dataset.

that the homography is consistent with the scene geometry and it can be recovered from the correspondences.

Finally, we keep only a single estimatable homography for each test case since the purpose of the benchmark is to compare robust estimators, *e.g.* RANSAC, that find only a single model. Thus, an image pair with k homographies is split into k test scenes. Each of them is generated by removing the inliers of the other estimatable homographies. Note that we keep those correspondences that are shared between the current homography and any other one.

3. Homography Evaluation Benchmark

The tentative correspondences are obtained from the mutually nearest RootSIFT matches minus the inliers of the other planes in the image pairs. The full input information, available to the methods is a set of N correspondences $\{C_i\}_{i=1}^{N}$, each consisting of $(x_i, y_i, \phi_i, s_i, x_i', y_i', \phi_i', s_i', \text{SNN ratio})$, where x_i, $y_i \in \mathbb{R}$ are the point coordinates, $\phi \in [0, 2\pi)$ is the SIFT feature orientation, $s \in \mathbb{R}$ is the scale, and SNN ratio is Lowe ratio [43] and $'$ denotes the second image.

The dataset is split into two disjoint parts. The training set contains two scenes – Alamo and NYC Library. The test set contains the remaining nine scenes. While the training set might not be large enough to allow training models from scratch, it allows to set the parameters of models and traditional algorithms, such as inlier-outlier threshold.

In Figures 3 and 4, properties of the HEB dataset are visualized. The left plots of Fig. 3 report the $\log_{10}$ number of homographies (vertical axis) having a particular inlier ratio (horizontal). The figures clearly demonstrate that the benchmark is extremely challenging since approximately the 80% of the homographies in the dataset have at most 0.1 inlier ratio. The training set shows similar statistics with marginally fewer cases with high inlier ratio.

The plot in the right of Fig. 3 shows histograms of the angle between the translations $\mathbf{t}$ and plane normals $\mathbf{n}$. The $0°$ case can be interpreted as a camera moving backwards from the plane. When the angle is $90°$, the camera moves sideways. At $180°$, the camera moves towards the observed plane. It can be seen that all possible directions are well-covered both in the test and training sets.

In Fig. 4, the inlier numbers are shown. In 30% of the

Figure 5. Comparison of **H** quality metrics. Results averaged over all datasets: (a), (b) average median number of inliers versus mAA of the pose error and mAA of the pixel re-projection error. While more inliers often imply better accuracy, it is not always the case, and methods may have different accuracy with similar numbers of inliers. In plots (c), (d), the mAA of the pose versus mAA of the re-projection errors and the mAA of rotation-only component (used in IMC [35]) are shown. LMEDS and LSQ are omitted here. The re-projection error is a good proxy for pose accuracy with two exceptions. (d) While the pose accuracy and scale-less rotation-only mAA [35] are well-correlated, the method ranking is significantly affected by the metric.

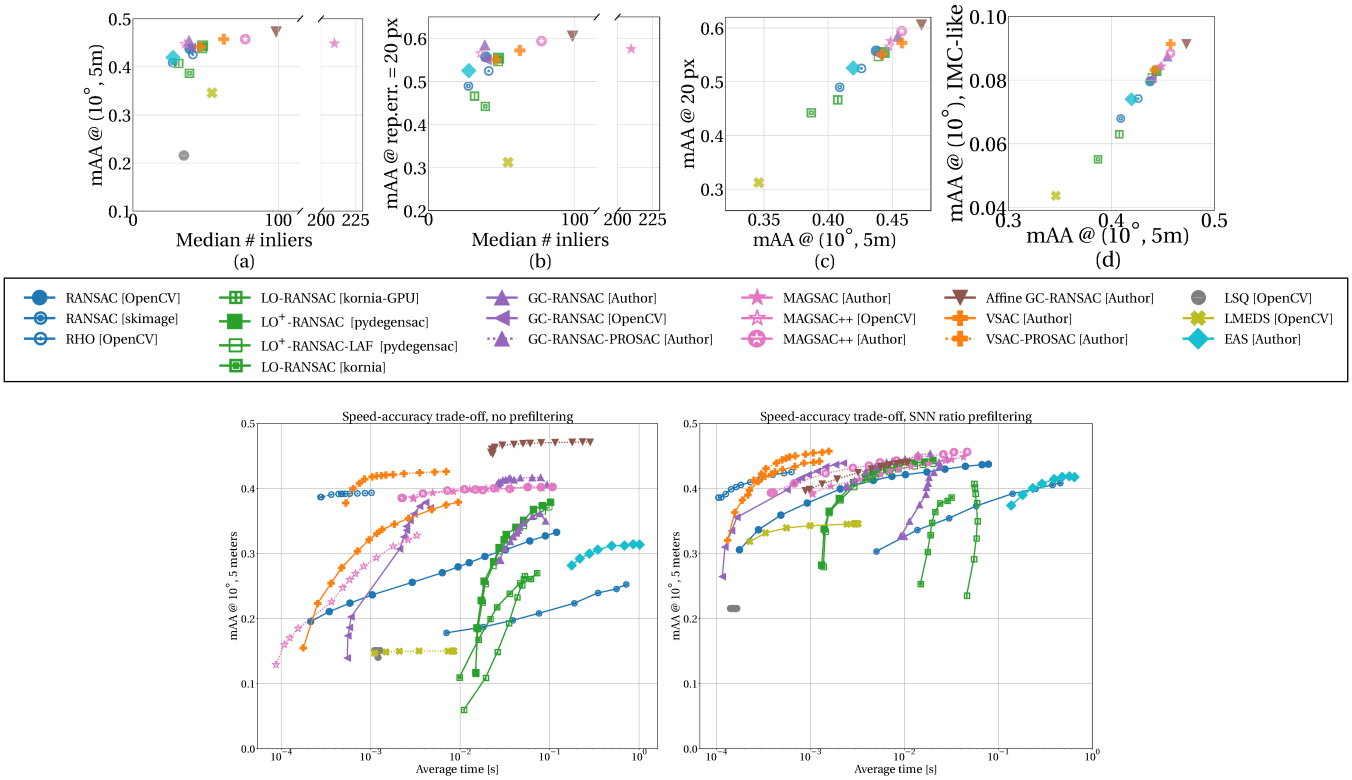

Figure 6. Speed-accuracy comparison of homography estimators on HEB test set, average over images and scenes. The max. number of iterations was varied from 10^1 to 10^3. Note the logarithmic scale of the time axis. Left – no prefiltering except mutual nearest neighbor check. Right – mutual nearest neighbor and SNN (Lowe) ratio check. PROSAC sorting is indicated by a dashed line.

cases, the homographies have fewer than 20 inliers, making the robust estimation challenging, especially when the outlier number is high. It is important to note that the success, in practice, depends more on the inlier number than the inlier ratio. This is caused by the fact the outliers often tend to form spatially coherent structures misleading the estimator if the inliers are sparsely distributed in the scene [32]. The majority of the homographies have fewer than 50 inliers. The same distribution holds for the training set.

4. Experimental Protocol

Our evaluation protocol is largely influenced by the Image Matching Benchmark [35]. However, we made several important changes, described below.

Metrics. We compute a range of per-pair metrics from one of the following three groups.

(i) Pose-based: Eqs. (1), (2) and absolute translation error:

$$\epsilon_{\mathbf{t}'_{abs}} = |\mathbf{t} - \mathbf{t}'|_2. \tag{3}$$

(ii) Ground truth correspondences-based: re-projection error of the GT correspondences with estimated homography:

$$\epsilon_{\text{repr}} = |\mathbf{x} - \mathcal{H}(\mathbf{x}')|_2. \tag{4}$$

the homography operator $\mathcal{H}$ transforming the non-homogeneous image coordinates $\mathbf{x}'$.

(iii) Self-supervised: number of inliers, run-time.

The per-homography metrics are accumulated into scene-metrics by the (a) mean, (b) median and (c) calculating mean average accuracy (mAA) with thresholds: from 1° to 10° for angular metrics, from 0.1 m to 5 meters for absolute translation error Eq. (3) and from 1 to 20 pixels for re-projection error Eq. (4). The thresholds resemble the ones used in the visual localization literature [74].

Since the scale can not be recovered from an **E** or **H** matrix [30], we assign the GT absolute scale to the estimated translation $\mathbf{t}'$. There is an important difference in measuring the absolute translation and purely angular errors in Eq. (2) as done in IMC [35]. For short baseline, *e.g.*, a few centime-

ters, the camera position noise largely affects the translation angle. Thus, Eq. (2) distorts the evaluation by returning large errors even when the camera barely moves in the real world. We select the averages of the rotation and translation mAA scores to be our main metric.

Metrics comparison. We plot the angular pose accuracy vs. metric pose accuracy in Fig. 5 (right). They are mostly in agreement, except for a few methods, *e.g.*, EAS [23] and Affine GC-RANSAC [9]. The mAA of the re-proj. error is also in agreement with the mAA of the pose error (Fig. 5; 3rd) with some exceptions, *e.g.*, LO$^+$-RANSAC.

The number of inliers (Fig. 5, two left graphs) greatly depends not only on image resolution, but also on the inlier threshold and particulars of each algorithm – MAGSAC outputs many more inliers, while having similar pose accuracy to other methods, while the LMEDS pose is much worse with the same number of inliers as the rest.

Training and Test Protocols. One of the drawbacks of the existing homography estimation datasets is the lack of tuning and test protocols. We propose the following procedure for fair evaluation. The main principle is as follows: one should not not make more than one or two evaluation runs on the test set. That it why all the hyper-parameters of the algorithms are fixed when running on the test set. The tuning and learning are done on the training set, which has similar, but not equal properties and no overlap in terms of content with the test set. We tune all the hyper-parameters with grid search for simplicity.

Training protocol. We fix number of iterations to 1000 for all methods. With each method, grid search is performed on the training set to determine the optimal combination of the hyper-parameters, such as inlier-outlier threshold θ, the SNN ratio threshold and other algorithm-specific parameters, such as the spatial weight of GC-RANSAC. Note that, unlike IMC [35], inlier-outlier and SNN thresholds are tuned jointly and not consequently – we found that it leads to slightly better hyper-parameters.

We tested the robust estimators on correspondences filtered by the predicted score of recent deep learning models. After obtaining the scores, we post-processed them in one of the two ways: (a) thresholding the scores at θ and removing tentative correspondences below it; and (b) sorting the correspondences by their score and keeping the top K best. Both θ and K were found by running grid search on the training set similarly as for other hyper-parameters.

Test protocol. After fixing all hyper-parameters, we run the algorithms on the test set, varying their maximum number of iterations from 10 to 10 000 (to 1000 for methods significantly slower than the rest, *i.e.*, scikit-image RANSAC, EAS and kornia-CPU) to obtain a time-accuracy plot. The algorithm terminates after its iteration number reaches the maximum. Note that, unlike in IMC [35], such experiments are performed on the test, not training set.

Methods for Homography Estimation. We give a brief overview of algorithms that we compare on HEB. Note that we consider it important to compare not just the algorithms as published in their respective papers but, also, their available implementations. Even though it might seem unfair to compare a method implemented in Python to C++ codes, the main objective is to provide useful guidelines for users on which algorithms and implementations to use in practice.

Traditional Algorithms. In all tested methods, the normalized DLT algorithm runs both on minimal and non-minimal samples. We found that the implementation is as important as the algorithm itself, thus, we define a method by its name and the library in which it is implemented.

We compare the OpenCV implementations of RANSAC [24], LMEDS [53], LSQ, RHO [11], MAGSAC++ [7], and Graph-Cut RANSAC [6]. The RANSAC implementation as in the scikit-image library [64]. Unlike OpenCV RANSAC, which is implemented in optimized C++ code, scikit-image is implemented in pure Python with the help of numpy [29]. LO-RANSAC [20] as implemented in the PyTorch [50]-based kornia library [52]. LO-RANSAC$^+$ [38] implemented in the pydegensac library with and without local affine frame (LAF) check [46]. The Graph-Cut RANSAC, MAGSAC [8], MAGSAC++ and VSAC [32] algorithms implemented by the authors. While MAGSAC and MAGSAC++ uses the PROSAC sampler [19] as default, we run GC-RANSAC and VSAC with and without PROSAC. We also evaluate the deterministic EAS algorithm [23] provided by the authors. EAS is implemented in pure Python using the numpy [29] package.

Also, we apply the affine correspondence-based GC-RANSAC [9] with its implementation provided by the authors. Since our benchmark does not have affine correspondences, we approximate them using SIFT features. Given rotations $\alpha_1, \alpha_2 \in [0, 2\pi]$ and scales s_1, s_2 in the two images for a correspondence, the affine transformation is calculated as $\mathbf{A} = \mathbf{J}_2\mathbf{J}_1^{-1}$, where $\mathbf{J}_i = \mathbf{R}_i\mathbf{S}_i$, matrix $\mathbf{R}_i$ is the 2D rotation by α_i degrees, and $\mathbf{S}_i$ is the 2D scale matrix uniformly scaling by s_i along the axes, $i \in [1, 2]$.

Deep prefiltering. The standard two-view matching pipeline with SIFT or other local features uses the SNN test [43] to filter out unreliable matches before running RANSAC [13, 22, 35]. Recently, it was shown [51, 69] that using a neural network for correspondence prefiltering might provide benefits over the SNN ratio test.

We evaluated how using models [13,16,51,61,69,72,75] for correspondence prefiltering for uncalibrated epipolar geometry help in homography estimation. For our study, we took pre-trained models, provided by the authors of each paper and use them for scoring the tentative correspondences. We emphasize that we neither trained, nor fine-tuned them for the homography estimation task, so their performance

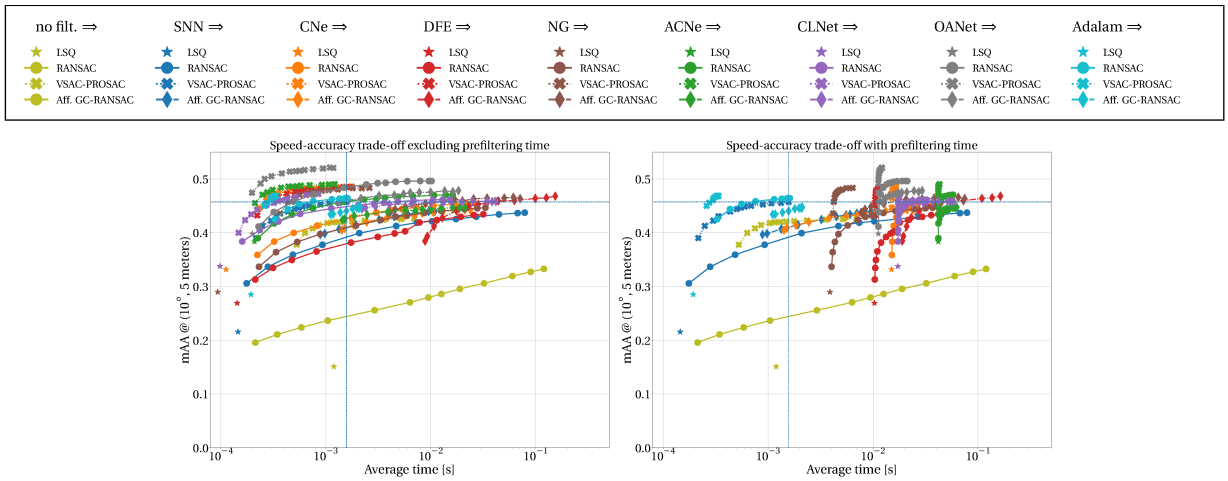

Figure 7. Speed-accuracy comparison of the classical SNN ratio and deep prefiltering on OpenCV LSQ and RANSAC algorithms, Affine GC-RANSAC [9] (best accuracy w/o prefiltering) and VSAC-PROSAC [32] (best accuracy with SNN prefiltering). Left – RANSAC time ($\log_{10}$ scale), right – time of RANSAC and the deep prefiltering. The best result of SNN ratio is marked by blue dashed lines.

is sub-optimal. The reason why we did not take the pre-trained models for homographies is that authors do not provide them. Unless stated otherwise, all the pre-trained models we used, were trained on a subset [69] of YCC100M dataset for fundamental matrix estimation.

5. Experiments

Traditional Methods. The pose errors are shown in Fig. 6.

No-prefiltering. This is the setup, where the difference between methods is the most pronounced. The most accurate method in all metrics is Affine GC-RANSAC that exploits the orientation and scale of SIFT features and, thus, reduces the combinatorial complexity of the problem. The second most important feature is PROSAC sampling, which improves the results of VSAC and GC-RANSAC by up to 10 percentage points. The optimized implementation matters a lot in terms of speed – python-based skimage RANSAC, EAS and kornia-CPU are up to 1-3 orders of magnitude slower than the other RANSACs. Kornia-GPU is on par in terms of speed with OpenCV RANSAC or pydegensac LO$^+$-RANSAC, but is worse in terms of accuracy. Even with the same language (C++), the speed and even the accuracy of different implementations of GC-RANSAC and MAGSAC++ vary significantly.

Prefiltering with SNN ratio. With optimal SNN ratio filtering, the difference between methods becomes smaller and most of the advanced RANSACs show similar accuracy, *e.g.*, LO$^+$ and GC-RANSAC. For most methods, the best SNN threshold is 0.6, which is stricter than the widely used 0.8. We believe that it is due to HEB having small inlier ratios, hence requiring aggressive filtering. The RHO algorithm is still the leader in top-speed part, outperformed by VSAC-PROSAC with increasing time budget. Affine GC-RANSAC is the one which benefits from correspondence prefiltering the least, both in terms of speed and accuracy.

As expected, LSQ fitting and LMEDS yield inaccurate results in all cases due to the high inlier ratio in the dataset. Interestingly, the recently proposed EAS algorithm [23] leads to highly inaccurate results both in the SNN-filtered and unfiltered cases. It is also surprising that affine GC-RANSAC [9] with using approximated affine correspondences only (from the SIFT orientations and scales) is the top-performing method in the unfiltered case and is among the best ones when SNN filtering is applied. This highlights the importance of using higher-order features to reduce the sample size in RANSAC. Due to the small sample size, the combinatorics of the problem is reduced, thus improving randomized RANSAC-like robust estimation.

Deep prefiltering. Results are shown in Fig. 7. The top row shows the combined pose error, while the bottom one shows the errors either in the rotation or in the translation. The best deep prefiltering methods provide an accuracy boost to advanced RANSACs of the similar magnitude, as switching from the no-filtering to SNN ratio filtering. However, not all methods are equal: there is a clear distinction between earlier methods like DFE, CNE and NG, and later models like OANet, ACNe and CLNet. The latter ones use specialized architectures, while DFE, CNE and NG are based on batch-normalized MLPs. OANet provides the best results, it is also the only model among the leaders which uses side information – SNN ratio – as an input. It is also interesting that the vanilla OpenCV RANSAC with OANet or CLNet prefiltering performs similarly to VSAC + SNN ratio in terms of accuracy. LSQ with deep filtering performs similarly to RANSAC with SNN-ratio filtering and better than RANSAC without prefiltering at all.

Finally, we show the time-accuracy plot in Fig. 7 (right) when the deep prefiltering (on NVIDIA V100 GPU) time is taken into account. It is at least 5-10 ms per image pair

for the fastest methods (NG, DFE and CLNet), which potentially is a limitation for real-time applications, especially when running on a smart device without GPU.

An application: uncertainty of SIFT keypoints. The uncertainty of popular detectors and their implementations is unknown or incomparable, *e.g.*, only referring to a certain resolution. Our goal is to determine bias and variance of angular, scale, and positional transformations of SIFT keypoints $\{C_i\}_{i=1}^N$ and – if possible – compare it to previous results. This may be a motivation to use the scaled rotation as an approximation for the local affine transformation.

The positional uncertainty of SIFT keypoints is known to be approximately 1/3 pixel (see [25] p.681, [37] Tab.6). The standard deviations (STD) of the keypoints depend on the detector scales (see [25] p.681, [71] Eq.(15)). We are not aware of investigations into the uncertainty of the directions and scales. The SIFT detector (in OpenCV) uses an orientation histogram with 36 bins of 10 degrees. Assuming an average STD of less than three times the rounding error $10^\circ/\sqrt{12} \approx 2.89^\circ$, the average STD of $\alpha_i = \phi'_i - \phi_i$ is approx. 12°, the factor three taking care of other model errors. This large uncertainty may be useful in cases where the rotation between keypoints is large.

While the reference scale ratios easily can be determined from a local reference affinity $\widetilde{\mathbf{A}}_i$, derived from $\widetilde{\mathbf{H}}_i$, the reference rotations $\widetilde{\alpha}_i$ requires care. There are two approaches to obtain reference rotations: (1) comparing direction vectors $\mathbf{d}(\phi'_i)$ in the second image with the transformed direction $\mathbf{d}(\phi_i)$ in the first image, and (2) deriving a local rotation from the reference affinity matrix $\widetilde{\mathbf{A}}_i$ and compare it to α_i.

We apply following approach: approximate the projective transformation by a local affinity $\widetilde{\mathbf{A}}_i \in \mathbb{R}^{2\times2}$, and, decompose $\widetilde{\mathbf{A}}_i$ into reference scale ratio $\widetilde{r}_i$, rotation angle $\widetilde{\alpha}_i$, and two shears $\widetilde{\mathbf{p}}_i \in \mathbb{R}^2$. We investigated QR, SVD and an exponential decompositions, namely decomposing the exponent $\widetilde{\mathbf{B}_i}$ of $\widetilde{\mathbf{A}_i} = \exp(\widetilde{\mathbf{B}_i})$ additively (see supplement). We evaluate the differences $\Delta\alpha_i = \widetilde{\alpha}_i - \alpha_i$ between observed and reference angles. The bias $\mathbb{E}(\Delta\alpha_i)$, *i.e.* the mean of $\Delta\alpha_i$ and the STD $\sigma_{\Delta\alpha_i} = \sqrt{\mathbb{D}(\Delta\alpha_i)}$ of the rotation differences $\Delta\alpha$, for the OpenCV SIFT detector empirically lead to an estimated STD of the rotation $\hat{\sigma}_{\Delta\alpha_i} = 11.8^\circ$, which is close to the above mentioned expectation.

Each of the three approaches leads to different reference rotations $\widetilde{\alpha}_i$. Rotation $\widetilde{\alpha}_i$ is effected by the shears $\widetilde{\mathbf{p}}_i$ in $\widetilde{\mathbf{A}}_i$. If the shears are small, all three methods yield similar rotations. The magnitude $|\widetilde{\mathbf{p}}_i|^2$ of the shears can be approximated by the condition number $\text{cond}(\widetilde{\mathbf{A}}_i)$. To evaluate the rotations α_i of the keypoint pairs, we restrict the samples to those with condition number < 1.5, which for image pairs in normal pose roughly is equivalent to slopes of the scene plane below 25°, see Suppl. 4.2. Moreover, we show the comparison of angular residuals between $\mathbf{d}'_i = [\cos(\phi'_i)\ \sin(\phi'_i)]^\mathrm{T}]$ and the one obtained by affinely transformed $\mathbf{d}_i = [\cos(\phi_i)\ \sin(\phi_i)]^\mathrm{T}]$, *i.e.* with $\widetilde{\mathbf{A}}_i\mathbf{d}_i$. The average deviations are similar to those obtained with the decomposition methods, see the details in the suppl. material.

The scale ratio $r_i = s'_i/s_i$ of a keypoint pair and its ratio $\Delta r_i = r_i/\widetilde{r}_i$ to the reference ratio $\widetilde{r}_i$ should lead to $\mathbb{E}(\Delta r_i) = 1$. Further, we use a weighted log-ratio, measured as $\rho_i = \log(\Delta r_i)/\widetilde{r}_i$ which should follow $\mathbb{E}(\rho_i) = 0$, and takes into account the intuition, that larger scales are less accurate. The OpenCV implementation of the SIFT detector empirically leads to $\hat{\sigma}_{\rho_i} = 0.51$ (see the suppl. material). Obviously, the scales from the detector may on average deviate by a factor $1.6 \approx \exp(0.51)$ in both directions.

The positional residual of each keypoint pair is characterized by the mean reprojection error $\epsilon_{x_i} = \sqrt{(|\mathbf{x}'_i - \widetilde{\mathcal{H}}(\mathbf{x}_i)|_2^2 + |\mathbf{x}_i - \widetilde{\mathcal{H}}^{-1}(\mathbf{x}'_i)|_2^2)/8}$, the factor 8 guaranteeing that ϵ_{x_i} can be compared to the expected uncertainty of the coordinates. For the OpenCV SIFT detector, we empirically obtain a positional uncertainty of ϵ_{x_i} as $\hat{\sigma}_x \approx 0.67$ pixels. The STD is a factor two larger, than expected, which might result from accepting small outliers.

6. Conclusion

A large-scale dataset containing roughly 1000 planes (Pi3D) in reconstructions of landmarks, and a homography estimation benchmark (HEB) is presented. The applications of the Pi3D and HEB datasets are diverse, *e.g.*, training or evaluating monocular depth, surface normal estimation and image matching. As one possible application, we performed a rigorous evaluation of a wide range of robust estimators and deep learning-based correspondence filtering methods, establishing the current state-of-the-art in robust homography estimation. The top accuracy is achieved by combining VSAC [32] with OANet [72]. In the GPU-less case, a viable option is to use VSAC [32], OpenCV RHO [11] or Affine GC-RANSAC with SNN test, depending on the time budget. We also show that PROSAC – a well-known, but often ignored sampling scheme accelerates RANSAC by an order of magnitude. Exploiting feature orientation and scale has clear benefits in Affine GC-RANSAC and it can be used in other approaches as well, *e.g.*, VSAC.

As another application, we show that having a large number of homographies allows for analyzing the noise in partially or fully affine-covariant features. As an example, we evaluate DoG features. To the best of our knowledge, we are the first ones to investigate the actual noise in the orientation and scaling components of such features.

Acknowledgedment. The work was funded by EU H2020 ARtwin No. 856994, EU H2020 SPRING No. 87124, by OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics", and by the ETH Postdoc fellowship.

References

[1] Ebtsam Adel, Mohammed Elmogy, and Hazem Elbakry. Image stitching based on feature extraction techniques: a survey. *International Journal of Computer Applications*, 99(6):1–8, 2014. 1

[2] Relja Arandjelovic and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Conference on Computer Vision and Pattern Recognition*, pages 2911–2918, 2012. 2, 4

[3] Clemens Arth, Manfred Klopschitz, Gerhard Reitmayr, and Dieter Schmalstieg. Real-time self-localization from panoramic images on mobile devices. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 37–46, 2011. 1

[4] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3

[5] D. Barath, T-J. Chin, O. Chum, D. Mishkin, R. Ranftl, and J. Matas. RANSAC in 2020 tutorial. In *Conference on Computer Vision and Pattern Recognition*, 2020. 2

[6] Daniel Barath and Jiří Matas. Graph-cut RANSAC. In *Conference on Computer Vision and Pattern Recognition*, pages 6733–6741, 2018. 2, 6

[7] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *Conference on Computer Vision and Pattern Recognition*, pages 1304–1312, 2020. 6

[8] Daniel Barath, Jana Noskova, and Jiri Matas. Marginalizing sample consensus. *IEEE TPAMI*, 2021. 6

[9] Daniel Barath, Michal Polic, Wolfgang Förstner, Torsten Sattler, Tomas Pajdla, and Zuzana Kukelova. Making affine correspondences work in camera geometry computation. In *European Conference on Computer Vision*, pages 723–740. Springer, 2020. 6, 7

[10] Daniel Barath, Denys Rozumny, Ivan Eichhardt, Levente Hajder, and Jiri Matas. Progressive-X+: Clustering in the consensus space. *arXiv preprint arXiv:2103.13875*, 2021. 3

[11] Hamid Bazargani, Olexa Bilaniuk, and Robert Laganiere. A fast and robust homography scheme for real-time planar target detection. *Journal of Real-Time Image Processing*, 15(4):739–758, 2018. 6, 8

[12] Eric Brachmann, Martin Humenberger, Carsten Rother, and Torsten Sattler. On the limits of pseudo ground truth in visual camera re-localisation. In *ICCV*, 2021. 2

[13] Eric Brachmann and Carsten Rother. Neural- Guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 2019. 6

[14] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 4

[15] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007. 1

[16] Luca Cavalli, Viktor Larsson, Martin Ralf Oswald, Torsten Sattler, and Marc Pollefeys. Handcrafted outlier detection revisited. In *ECCV*, 2020. 6

[17] Jianhui Chen, Fangrui Zhu, and James J Little. A two-point method for PTZ camera calibration in sports. In *WACV*, pages 287–295. IEEE, 2018. 1

[18] Zhou Chuan, Tan Da Long, Zhu Feng, and Dong Zai Li. A planar homography estimation method for camera calibration. In *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No. 03EX694)*, volume 1, pages 424–429. IEEE, 2003. 1

[19] Ondrej Chum and Jiri Matas. Matching with PROSAC-progressive sample consensus. In *Conference on Computer Vision and Pattern Recognition*, volume 1, pages 220–226. IEEE, 2005. 6

[20] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized ransac. In Bernd Michaelis and Gerald Krell, editors, *Pattern Recognition*, pages 236–243, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. 6

[21] Robert T Collins and J Ross Beveridge. Matching perspective views of coplanar structures using projective unwarping and similarity matching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 240–245. IEEE, 1993. 1

[22] Ufuk Efe, Kutalmis Gokalp Ince, and Aydin Alatan. Effect of parameter optimization on classical and learning-based image matching methods. In *ICCV Workshop*, October 2021. 6

[23] Aoxiang Fan, Jiayi Ma, Xingyu Jiang, and Haibin Ling. Efficient deterministic search with robust loss functions for geometric model fitting. *IEEE TPAMI*, 2021. 6, 7

[24] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2, 6

[25] Wolfgang Förstner and Bernhard P. Wrobel. *Photogrammetric computer vision*. Springer, 2016. 8

[26] Google. Google maps. http://maps.google.com/. 4

[27] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline, 2021. 2

[28] Matthias Grundmann, Vivek Kwatra, Daniel Castro, and Irfan Essa. Calibration-free rolling shutter removal. In *2012 IEEE international conference on computational photography (ICCP)*, pages 1–8. IEEE, 2012. 1

[29] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020. 6

[30] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 3, 5

[31] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977. 2

[32] Maksym Ivashechkin, Daniel Barath, and Jiri Matas. VSAC: Efficient and accurate estimator for h and f. In *ICCV*, 2021. 5, 6, 7, 8

[33] Manish Jethwa, Andrew Zisserman, and Andrew W Fitzgibbon. Real-time panoramic mosaics and augmented reality. In *BMVC*, pages 1–11, 1998. 1

[34] Shunping Ji, Zijie Qin, Jie Shan, and Meng Lu. Panoramic slam from a multiple fisheye camera rig. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159:169–183, 2020. 1

[35] Yuhe Jin, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 2020. 1, 3, 4, 5, 6

[36] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 3

[37] Thomas Läbe, Timo Dickscheid, and Wolfgang Förstner. On the Quality of Automatic Relative Orientation Procedures. In *ISPRS Archives*, volume XXXVII Part B3b, pages 37–42, 2008. 8

[38] Karel Lebeda, Ondřej Chum, and Jiří Matas. Fixing the locally optimized ransac. In *British Machine Vision Conference*, 2012. 2, 3, 6

[39] Thomas Lemaire and Simon Lacroix. Slam with panoramic vision. *Journal of Field Robotics*, 24(1-2):91–111, 2007. 1

[40] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 3

[41] David Liebowitz and Andrew Zisserman. Metric rectification for perspective images of planes. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, pages 482–488. IEEE, 1998. 1

[42] David Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*. IEEE, 1999. 1, 2

[43] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1, 2, 4, 6

[44] Andrew MacQuarrie and Anthony Steed. Cinematic virtual reality: Evaluating the effect of display type on the viewing experience for panoramic video. In *2017 IEEE Virtual Reality (VR)*, pages 45–54. IEEE, 2017. 1

[45] Ezio Malis and Manuel Vargas. *Deeper understanding of the homography decomposition for vision-based control*. PhD thesis, INRIA, 2007. 4

[46] Dmytro Mishkin, Jiri Matas, and Michal Perdoch. Mods: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 2015. 2, 3, 6

[47] D. Mishkin, F. Radenovic, and J. Matas. Repeatability is Not Enough: Learning Affine Regions via Discriminability. In *ECCV*, 2018. 2

[48] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016. 1

[49] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *ECCV*, 2020. 2

[50] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[51] Rene Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *The European Conference on Computer Vision (ECCV)*, 2018. 1, 6

[52] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *Winter Conference on Applications of Computer Vision*, 2020. 6

[53] Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984. 6

[54] Ethan Rublee, Vincent Rabaud, Kurt Konolidge, and Gary Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *Internation Conference on Computer Vision*, 2011. 1

[55] Muhamad Risqi U Saputra, Andrew Markham, and Niki Trigoni. Visual slam and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–36, 2018. 1

[56] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 1, 3

[57] Johannes Lutz Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision*, 2016. 1

[58] Gilles Simon, Andrew W Fitzgibbon, and Andrew Zisserman. Markerless tracking using planar structures in the scene. In *Proceedings IEEE and ACM international symposium on augmented reality (ISAR 2000)*, pages 120–128. IEEE, 2000. 1

[59] Noah Snavely, Steve Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM TOG*, volume 25, pages 835–846. ACM, 2006. 3

[60] Noah Snavely, Steve Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008. 3

[61] Weiwei Sun, Wei Jiang, Andrea Tagliasacchi, Eduard Trulls, and Kwang Moo Yi. Attentive context normalization for ro-

bust permutation-equivariant learning. In *CVPR*, 2020. 1, 6

[62] Chris Sweeney. Theia multiview geometry library. http://theia-sfm.org. 1

[63] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):1–11, 2017. 1

[64] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014. 6

[65] Lang Wang, Wen Yu, and Bao Li. Multi-scenes image stitching based on autonomous driving. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2020. 1

[66] K. Wilson and N. Snavely. Robust Global Translations with 1DSfM. In *European Conference on Computer Vision*, pages 61–75, 2014. 2, 3

[67] Jiaolong Yang and Hongdong Li. Dense, accurate optical flow estimation with piecewise parametric model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1019–1027, 2015. 1

[68] Tao Yang, Zhi Li, Fangbing Zhang, Bolin Xie, Jing Li, and Linfeng Liu. Panoramic UAV surveillance and recycling system based on structure-free camera array. *IEEE Access*, 7:25763–25778, 2019. 1

[69] Kwang Moo Yi*, Eduard Trulls*, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *CVPR*, 2018. 1, 6, 7

[70] Georges Younes, Daniel Asmar, Elie Shammas, and John Zelek. Keyframe-based monocular slam: design, survey, and future directions. *Robotics and Autonomous Systems*, 98:67–88, 2017. 1

[71] Bernhard Zeisl, Pierre Fite Georgel, Florian Schweiger, Eckehard Steinbach, and Nassir Navab. Estimation of Location Uncertainty for Scale Invariant Feature Points. In *Proc. BMVC*, pages 57.1–57.12, 2009. doi:10.5244/C.23.57. 8

[72] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. *ICCV*, 2019. 6, 8

[73] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000. 1

[74] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *International Journal of Computer Vision*, 129(4):821–844, 2021. 5

[75] Chen Zhao, Yixiao Ge, Feng Zhu, Rui Zhao, Hongsheng Li, and Mathieu Salzmann. Progressive correspondence pruning by consensus learning. In *ICCV*, 2021. 1, 6

[76] Zihan Zhou, Hailin Jin, and Yi Ma. Robust plane-based structure from motion. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1482–1489. IEEE, 2012. 1

[77] Zihan Zhou, Hailin Jin, and Yi Ma. Plane-based content preserving warps for video stabilization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2299–2306, 2013. 1