Learning to Find Good Models in RANSAC

Daniel Barath, Luca Cavalli, Marc Pollefeys
Computer Vision and Geometry Group, Department of Computer Science, ETH Zürich
dbarath@ethz.ch

Abstract

We propose the Model Quality Network, MQ-Net in short, for predicting the quality, e.g. the pose error of essential matrices, of models generated inside RANSAC. It replaces the traditionally used scoring techniques, e.g., inlier counting of RANSAC, truncated loss of MSAC, and the marginalization-based loss of MAGSAC++. Moreover, Minimal samples Filtering Network (MF-Net) is proposed for the early rejection of minimal samples that likely lead to degenerate models or to ones that are inconsistent with the scene geometry, e.g., due to the chirality constraint. We show on 54450 image pairs from public real-world datasets that the proposed MQ-Net leads to results superior to the state-of-the-art in terms of accuracy by a large margin. The proposed MF-Net accelerates the fundamental matrix estimation by five times and significantly reduces the essential matrix estimation time while slightly improving accuracy as well. Also, we show experimentally that consensus maximization, i.e. inlier counting, is not an inherently good measure of the model quality for relative pose estimation. The code is at `https://github.com/danini/learning-good-models-in-ransac`.

1. Introduction

The RANSAC (RANdom SAmple Consensus) algorithm proposed by Fischler and Bolles [12] in 1981 has become the most widely used robust estimator in computer vision. RANSAC and its variants have been successfully applied to a wide range of vision tasks, *e.g.*, short baseline stereo [40, 42], wide baseline matching [23, 24, 28], motion segmentation [40], detection of geometric primitives [35], pose-graph initialization for both incremental and global structure-from-motion pipelines [3, 33, 34], image mosaicing [14], and to perform [2, 19, 46], or initialize general multi-model fitting algorithms [17, 27].

In brief, RANSAC repeatedly selects, typically, minimal subsets of the data points and fits a model, *e.g.*, a 3D plane to three points, an essential matrix to five 2D point correspondences, or a 6D pose to three 2D-3D correspondences. The quality of the model is measured as the cardinality of its support, *i.e.*, the number of inlier data points. Finally, the model with the highest quality, polished, *e.g.*, by least-squares fitting or numerical optimization on all inliers, is returned. In this paper, first, we focus on improving the RANSAC scoring by a learning-based approach. Second, we accelerate the robust estimation by learning to reject minimal samples that likely lead to degenerate solutions.

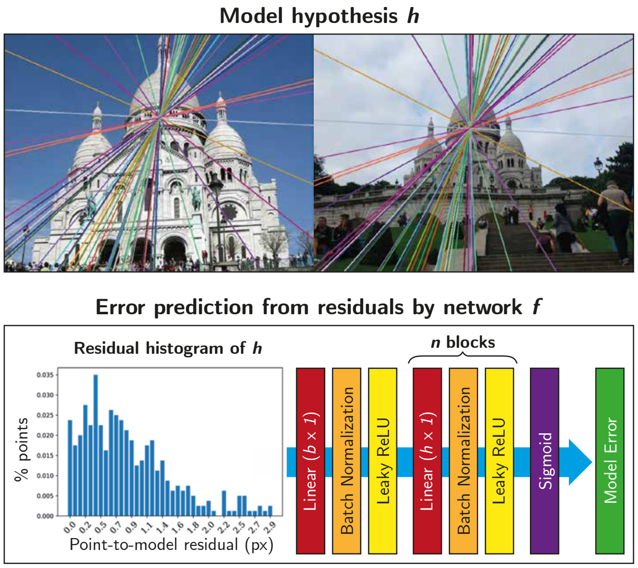

Figure 1. **MQ-Net**: quality prediction of model hypotheses.

Since the publication of RANSAC, a number of modifications have been proposed replacing components of the original algorithm. To improve the accuracy by better modelling the noise in the data, different model quality calculation techniques have been investigated. For instance, MLESAC [41] estimates the quality by a maximum likelihood procedure with all its beneficial properties, albeit under certain assumptions about point distributions. In practice, MLESAC results often are superior to the inlier counting of plain RANSAC, and they are less sensitive to the manually set inlier-outlier threshold. In MSAC [41], the loss is formulated as a truncated quadratic error by assigning constant loss to the outliers (*i.e.*, points with residuals larger than the inlier-outlier threshold) and a quadratic one to the inliers. In MAPSAC [39], the estimation is formulated as a process that estimates both the parameters of the data distribution and the model quality in terms of maximum a posteriori. In the recently proposed MAGSAC++ [6], the model quality calculation is formulated as a marginalization over a range of noise scales. The inlier residuals are assumed to have χ^2 distribution. This allows MAGSAC++ to be significantly less sensitive to the inlier-outlier threshold than other robust estimators. According to a recent survey [22], MAGSAC++ is currently the most accurate robust estimator.

Finessing to interpret inlier and outlier distributions, traditional scoring techniques usually consider the data as a mixture with the outliers being uniformly distributed in the scene. However, this assumption is rarely satisfied in real scenes, where the outliers tend to form spatially coherent structures invalidating the assumption of uniformity and misleading the scoring function [18]. While consensus maximization is an actively researched area in computer vision [20, 29, 38], maximizing the inlier number does not necessarily lead to finding the sought model parameters [41]. To demonstrate this, Fig. 2 shows the Sampson distance distributions of relative poses estimated from real image pairs from [36]. Each curve shows the average residual distribution of $10\,000$ poses whose errors fall within the interval shown in the legend. For instance, the green curve shows the distribution calculated from poses with errors in-between $[1^\circ, 5^\circ]$. Its value at 0.75 is approx. 0.04%. Consequently, the 0.04% of the points has 0.75 Sampson distance for poses with such errors. Basically, the area under the curve is the inlier ratio. Notice that almost perfect models (red curve) have, on average, fewer inliers than the ones that are reasonably accurate but not perfect (green). This suggests that inlier maximization is not an inherently good measure of the model quality, at least, when estimating relative poses and using Sampson distance.

To better model the inlier distributions, we propose a new scoring technique that is trained to predict the model quality from point-to-model residuals without making explicit assumptions about the actual distributions. To build residual histograms that are then learned, we use a reasonably large inlier-outlier threshold that works on a wide range of scenes without further hyper-parameter tuning. Since this threshold is too large for selecting a final set of inliers, we also propose a data-driven strategy for inlier selection. The method straightforwardly replaces the scoring function in modern RANSAC frameworks, *e.g.*, VSAC [18].

In modern RANSACs, the model estimation often continues with degeneracy and chirality testing to reject models that are incompatible with the scene geometry, *e.g.*, as in DEGENSAC [11]. Some models, *e.g.* homography, allow to perform checks directly on minimal samples, thus, accelerating the robust estimator significantly by skipping both the model estimation and quality calculation if the sample does not pass a test. Besides the speed-up, rejecting degenerate models improves the accuracy as well since such models often have high inlier counts [8, 18]. For epipolar geometry estimation, there are, however, no such checks that can be applied to minimal samples before the fundamental or essential matrix is estimated.

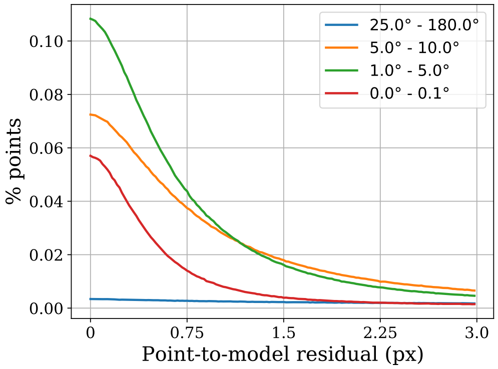

Figure 2. **Consensus maximization does not favor the best model**. Average residual distributions over $10\,000$ image pairs. Each curve shows the residual distribution of relative poses with error falling in the interval shown in the name of the curve. The vertical axis shows the percentage of points that has a particular point-to-model residual (Sampson distance; horizontal axis).

We propose a network to predict the probability of a minimal sample leading to a degenerate model when estimating the epipolar geometry. We train an extremely light-weight network that is invariant to both the point and image ordering. It efficiently rejects minimal samples prior to the model estimation, thus, leading to a significant speed-up while slightly improving the accuracy as well. The training data is naturally synthesized from explicit post-model degeneracy checks on the available training images.

The algorithms are tested both on fundamental and essential matrix estimation on $54\,450$ image pairs from the PhotoTourism dataset. The proposed scoring and sample filtering techniques, together, improve the accuracy by a large margin compared to the state-of-the-art (*e.g.*, the median error is the half of the MAGSAC++ error) while running faster or at a comparable speed, in *real-time*.

2. Model Quality Network

In this section, we describe Model Quality Network (MQ-Net) proposed for learning the model error from a histogram built from the point-to-model residuals of points closer than a fairly large $\epsilon_{\max}$ threshold, *e.g.*, 3 pixels for fundamental or essential matrix estimation.

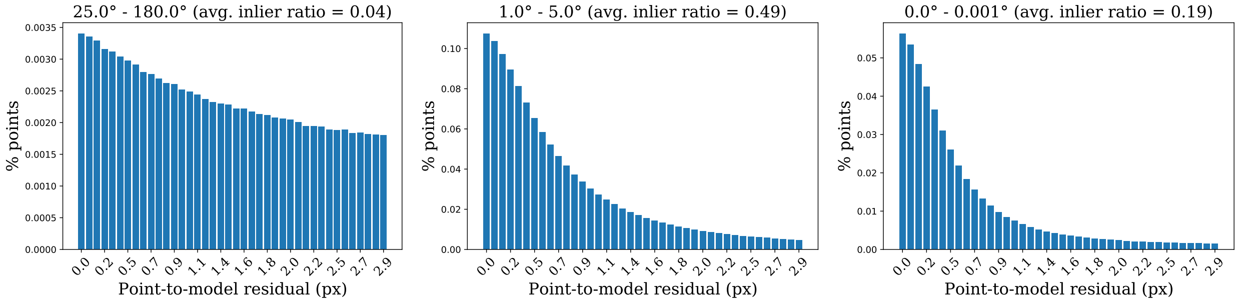

Figure 3. **Histograms and inlier ratios for bad, good, and perfect relative poses.** Average residual histograms of 10 000 instances of relative poses where the maximum of the rotation and translation errors is in-between (left) $25^\circ - 180^\circ$, (middle) $1^\circ - 5^\circ$, and (right) $0^\circ - 0.001^\circ$. The average inlier ratio of the models is written in the title. Notice that the inlier ratio is higher for the moderately accurate poses in the middle plot than for the almost perfect ones in the right one.

2.1. Residual Histogram

State-of-the-art algorithms try modelling the noise in the point-to-model residuals as having Gaussian [39, 41] or χ^2 [4, 5] distribution. We, however, found that assuming the inlier or outlier residual distribution to follow a particular model in real-world scenes is unnatural and, thus, necessarily leads to sub-optimal solutions. Fig. 2 shows the residual (*i.e.*, Sampson error) distributions averaged over 10 000 relative poses from the PhotoTourism dataset. We used only those relative poses for a particular curve, where the mean of the rotation and translation errors (compared to a ground truth; in degrees) is in the range shown in the legend.

Instead of guessing the actual distribution, we learn it from the point-to-model residuals. We create residual histograms with $h \in \mathbb{N}$ bins only using residuals that are smaller than a fairly large inlier-outlier threshold ϵ_{max}. Parameter $\epsilon_{\text{max}} \in \mathbb{R}^+$ can be considered as a threshold upper-bound as in [6]. Given a model $\theta \in \mathbb{R}^d$ ($d \in \mathbb{N}$) estimated, *e.g.*, from a minimal sample inside RANSAC, the value in the i-th bin, $i \in [0, h)$, of the histogram is calculated as

$$b_i = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \left[\!\!\left[\frac{i\epsilon_{\text{max}}}{h} \leq R(p, \theta) < \frac{(i+1)\epsilon_{\text{max}}}{h} \right]\!\!\right],$$

where $R : \mathcal{P} \times \mathbb{R}^d \to \mathbb{R}^+$ is a residual function, and $[\![.]\!]$ is the Iverson-bracket that is one if the condition inside holds and zero otherwise. The value of b_i basically is the proportion of points with residuals falling in interval

$$I_i = \left[\frac{i\epsilon_{\text{max}}}{h}, \frac{(i+1)\epsilon_{\text{max}}}{h} \right).$$

Example histograms of relative poses are shown in Fig. 3. The histograms are averaged over 10 000 problem instances. The right plot shows the histograms of poses with approximately zero error – these we generated directly from the ground truth COLMAP [32] reconstructions. In the middle one, the poses are reasonably accurate but not perfect. Their error is in-between 1° and 5°. The left one shows the histograms of inaccurate poses. The average inlier ratio is in the title. While inaccurate poses are easy to be differentiated from the accurate ones, it is interesting that the average inlier ratio of reasonably good models is higher than that of the almost perfect ones. This means that scoring techniques based purely on the inlier ratio, *e.g*. RANSAC and MSAC, *fail* to find the most accurate relative poses by nature.

2.2. Data Generation

In order to generate training and validation data, we first load each image pair with a known ground truth relative pose. We detect 8000 SIFT keypoints in both images in order to have a reasonably dense point cloud reconstruction and precise camera poses [43]. We combine mutual nearest neighbor check with standard distance ratio test [21] to establish tentative correspondences, as recommended in [43].

To be able to learn how the residuals of accurate poses look, we calculate the histogram of the ground truth one and store it with a prediction target of zero. We generate 10 poses with perfect rotation and translation vector rotated by a random rotation matrix. Also, 10 poses are added with perfect translation and rotation matrix multiplied by a random rotation. Finally, 100 relative poses are generated by drawing minimal samples uniformly randomly, estimating the implied models, calculating their errors w.r.t. the ground truth pose, and storing their residual histograms. The prediction target is always the average of the translation and rotation errors, w.r.t. to the ground truth pose, divided by 180°. Thus, it is normalized into interval $[0, 1]$. A total of 121 samples are generated from each image pair. We found that learning rotation and translation errors separately is, generally, less effective than learning a unified score. This is expected since, due to the nature of projective geometry [15], the error in the translation and rotation can not be disentangled from the point-to-model residuals.

2.3. Training Loss and Network

Due to being able to normalize the target between 0 and 1, we can consider the problem as binary classification, where 0 is an accurate and 1 is an inaccurate model. This assumption implies that the conditional density $p_\gamma(y \mid x)$ is the Bernoulli distribution as follows:

$$p_\gamma(y \mid x) = \begin{cases} f_\gamma(x), & \text{if } y = 1; \\ 1 - f_\gamma(x), & \text{otherwise}, \end{cases}$$

where $f_\gamma(x)$ is a point estimate, γ is the model parameters, x is the input, and we are given a set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{M}$ of training data generated as explained in the previous section.

In order to train the network by minimizing the negative log likelihood, we can use the Binary Cross Entropy (BCE) loss that is written as follows:

$$-\log p_\gamma(y \mid x) = -\left[y \log f_\gamma(x) + (1-y)\log(1 - f_\gamma(x))\right].$$

However, since the final objective is to discriminate the best model among the most accurate ones, learning the histograms of accurate models is significantly more important than learning inaccurate ones. Therefore, we modify the loss function to be

$$-\log p_\gamma(y \mid x, w) = -(1 - f_\gamma(x))^w \\ \left[y \log f_\gamma(x) + (1-y)\log(1 - f_\gamma(x))\right].$$

where w is a weighting parameter [9]. Weighting by the target error as $(1-f_\gamma(x))^w$ assigns high loss to histograms that resemble the histograms of accurate models. This allows the network to better discriminate between accurate models while diminishing the importance of inaccurate ones. This is motivated by the fact that it is negligible from the estimation standpoint if the error, for example, in the estimated rotation matrix is 80° or 100°. In contrast, assigning high score to a rotation with 1° error and lower score to one with 10° error is of extreme importance.

We use a fairly simple network that allows the proposed technique to be fast. To do so, we use $n_l \in \mathbb{N}$ linear layers of size s_l, each followed by a 1D batch normalization, a leaky ReLU and a dropout layer, see Fig. 1. Due to considering the problem as binary classification, we choose the sigmoid function as the last layer. To our tests, $w = 4$, $n_l = 5$ and $s_l = 1024$ lead to accurate results while being fast.

Interestingly, we found that a mixture of our learned score with the traditional inlier ratio outperforms both scores taken alone, suggesting that each score contributes discriminative information that the other does not provide. Therefore, the final model score is calculated as follows:

$$S(\theta, \mathcal{P}) = \alpha \frac{\sum_{p \in \mathcal{P}} [\![R(p, \theta) < \epsilon_{\text{max}}]\!]}{|\mathcal{P}|} + (1 - \alpha) f_\gamma(\theta),$$

where $S : \mathbb{R}^d \times \mathcal{P}^\times \to \mathbb{R}$ is the scoring function, $\alpha \in [0, 1]$ is a weighting parameter, and $f_\gamma(\theta)$ is the error prediction of the network given model θ. We use $\alpha = 0.5$ in all our experiments, thus, taking the average of the predicted score and the actual inlier ratio.

3. Final Model Polishing

The proposed learning-based scoring technique uses a fairly wide inlier-outlier threshold ϵ_{max} when calculating the residual histograms. This threshold is too wide in practice and, thus, it is not well suited to select the inliers of the model that scored the best by the proposed technique. This is quite problematic from the RANSAC standpoint, which always finishes with either a least-squares fitting or a numerical optimization on the final set of inliers. Therefore, we propose the following strategy to determine a set of inliers that can be used for the model re-estimation.

Suppose that we are given an initial set of inliers $\mathcal{I} = \{p \mid R(p, \theta) < \epsilon_{\text{max}} \ \wedge \ p \in \mathcal{P}\}$, point-to-model residual function $R : \mathcal{P} \times \mathbb{R}^d \to \mathbb{R}$, scoring function $S'(: \mathbb{R}^d \times \mathcal{P}^\times \to \mathbb{R}) = f_\gamma(\theta)$ and model fitting function $F : \mathcal{P}^\times \to \mathbb{R}^d$ that estimates the model parameters $\theta \in \mathbb{R}^d$ from a set of point correspondences, where $\mathcal{P}^\times$ is the power set of $\mathcal{P}$. Note that scoring S' only uses the prediction of the network and is not combined with the inlier ratio. We assume that maximum threshold ϵ_{max} is wide enough to accommodate the true threshold $0 \leq \epsilon^* \leq \epsilon_{\text{max}}$ (*i.e.*, implied by the noise scale σ). Consequently, the task is to solve

$$\epsilon^* = \arg \max_{\epsilon \in [0, \epsilon_{\text{max}}]} S'(F(\mathcal{I}_\epsilon), \mathcal{P}) \tag{1}$$

where $\mathcal{I}_\epsilon = \{p \mid p \in \mathcal{I} \wedge R(p, \theta) \leq \epsilon\} \subseteq \mathcal{I}$.

Let us recognize that the set of candidate values for ϵ^* leading to different $\mathcal{I}_\epsilon$ is finite. This threshold set coincides with the set of point-to-model residuals within interval $[0, \epsilon_{\text{max}}]$. Let us increasingly order the residuals of the points from $\mathcal{I}$ as $0 = r_1 = \cdots = r_m \leq r_{m+1} \leq \cdots \leq \epsilon^* \leq \cdots \leq r_{|\mathcal{I}|} \leq \epsilon_{\text{max}}$, where m is the minimal sample size. Threshold ϵ^* maximizing $S'(F(\mathcal{I}_\epsilon))$ is found by progressively increasing the threshold value, where $\epsilon_0 = r_{m+1}$, $\epsilon_1 = r_{m+2}, \cdots, \epsilon_{|I|-m} = r_{|I|}$ and, thus, adding the points one-by-one to the final inlier set. The best threshold ϵ^* is the one where the predicted score of model $\theta^* = F(\mathcal{I}_{\epsilon^*})$ from the implied inlier set $\mathcal{I}_{\epsilon^*}$ maximizes the learned score $S'(F(\mathcal{I}_{\epsilon^*}))$. The algorithm is shown in Alg.1.

We make two important notes. First, scoring function S' must not increase monotonically together with the size of the inlier set, *e.g.*, as in the inlier counting of plain RANSAC. Otherwise, the best value for ϵ^* will always be ϵ_{max}. This is the sole reason why we use S' instead of S. Second, the estimation procedure might be time-consuming if we include the points one-by-one to the current inlier set. Therefore, it is preferred to divide the residual set into k

Algorithm 1 Final Model Fitting.

```
Input: $\mathcal{I}$ – initial inliers; $\mathcal{P}$ – points; $m$ – sample size
       $r_1 \leq \dots \leq r_{|\mathcal{I}|}$ – inlier residuals; $\delta$ – step size
Output: $\theta^*$ – model parameters; $\mathcal{I}^*$ – inliers

$k \leftarrow m + 1, s^* \leftarrow 0, \mathcal{I}^* \leftarrow \varnothing$
while $k \leq |\mathcal{I}|$ do
    $\mathcal{I}_k \leftarrow \{p \mid R(p, \theta) \leq r_k \ \wedge \ p \in \mathcal{I}\}$
    $\theta_k \leftarrow F(\mathcal{I}_k)$                    ▷ Model estimation
    $s \leftarrow S(\theta_k, \mathcal{P})$                   ▷ Score calculation
    if $s > s^*$ then                       ▷ New so-far-the-best model
        $s^* \leftarrow s, \mathcal{I}^* \leftarrow \mathcal{I}_k$
    $k \leftarrow k + \delta$                          ▷ Increase the sample size
```

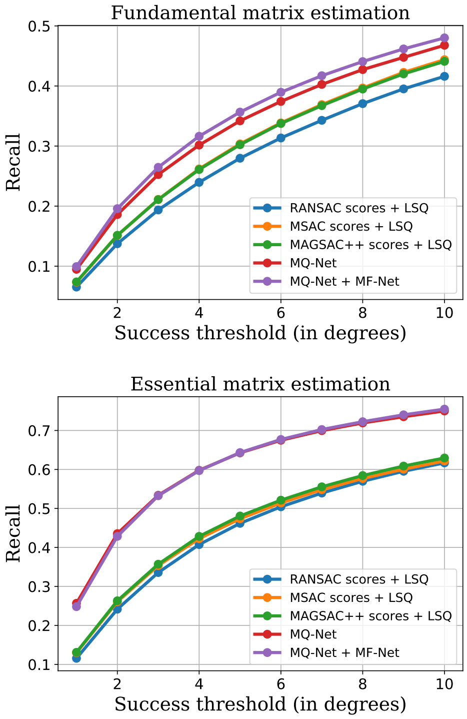

Figure 4. **Recall curves** showing the recall (vertical axis) of *plain* RANSAC with RANSAC, MSAC and MAGSAC++ scorings, followed by least-squares fitting on the final set of inliers; the recall of RANSAC with the proposed MQ-Net with and without filtering minimal samples using MF-Net.

groups and do k instead of $|\mathcal{I}| - m$ estimations. We will use $k = 10$ in the experiments. In the algorithm, there is no other threshold used besides ϵ_{max}.

4. Minimal Sample Filtering Network

In most of the estimation tasks, minimal samples often lead to degenerate models or to ones implying an impossible underlying scene. For instance, such a case is when a homography represents a plane that flips between the two views, *i.e.*, the second camera sees it from the back. For homographies, this case can be identified by simply checking the minimal sample. However, there is no such solution that finds degenerate configurations prior to epipolar geometry estimation. Prior art like DEGENSAC [11] and QDEGSAC [13] always need to perform the expensive epipolar geometry estimation to recognize degenerate minimal samples. In practice, this means that the models are estimated and the quality is calculated unnecessarily before identifying degenerate situations.

In this section we propose MF-Net (Minimal sample Filtering Network), a network that predicts the probability of a minimal sample leading to a degenerate model. This is fundamentally different from prior works [7,25,31,45] that train networks for outlier rejection: while they encode the full context of the available correspondences to filter the ones which do not conform with the camera motion, we instead score a minimal sample to predict its degeneracy independently of the underlying motion. This leads to the possibility, and requirement, of using an extremely lightweight model without learning motion priors which would hinder the generalization across datasets.

We define MF-Net as $MF_\omega : \mathbb{R}^{4m} \rightarrow [0, 1]$, a parametric function with parameters ω that maps a minimal sample $x \in \mathbb{R}^{4m}$ with m correspondences to the probability of its degeneracy. Although an analytically precise solution is available for this problem, it requires an expensive model estimation and quality calculation, which we wish to substitute with a much cheaper probabilistic solution. MF-Net needs to obey both the invariance to the ordering of correspondences and to the ordering of images by architecture, independently of the learned parameters ω. We take inspiration from PointNet [30] and achieve point ordering invariance by processing each correspondence independently with shared MLPs, and sharing information across correspondences with global max pooling operations. Image ordering, on the other hand, only spans two possibilities, so we run our backbone on both combinations and max pool features before predicting the final degeneracy score. The architecture is shown in the supplementary material.

We train MF-Net to classify minimal samples as degenerate or valid, and produce training data by running the classical degeneracy test on random minimal samples from real image correspondences. We use the binary cross-entropy loss and balance classes with double inverse frequency weighting so that the output probabilities represent the parameters of Bernoulli distributions independent of the frequency of degenerate samples in the training set [44]. We aim at achieving a speed-up without noticeable loss in the accuracy, so the optimal filtering threshold on the network confidence is not a trivial choice. We thus tune the threshold on a subset of the training set by observing the resulting accuracy and speed-up, as shown in Section 5.3 and Fig. 5.

For essential matrices, we train the network to recognize the following cases. (1) The minimal solver returns at least a single real solution. (2) The essential matrix has at least $m + 1$ inliers, where $m = 5$ is the minimal sample size. (3) There is at least a single pose, decomposed from the estimated essential matrix, which triangulates all correspondences from the minimal sample in front of both cameras.

For fundamental matrices, we check the minimal sample for H-degeneracy [11] together with the same criteria as what are used for essential matrices.

5. Experiments

In this section, we compare the proposed deep learning-based scoring technique to the original RANSAC inlier counting [12], the truncated quadratic loss of the widely used MSAC [41] and the state-of-the-art MAGSAC++ [6] methods. To do so, we implemented plain RANSAC with a final model re-estimation step applied to all found inliers. We then replaced the scoring function of this RANSAC with MSAC, MAGSAC++ and the proposed one. For RANSAC, MSAC and MAGSAC++, we applied LSQ fitting followed by the Levenberg-Marquardt [26] numerical optimization to estimate the final model parameters from all found inliers. For the proposed scoring, we run the algorithm proposed in Section 3. The confidence is set to 0.999 and the maximum iteration number to 10 000. The proposed learning-based approaches work with 2048-sized batches. The generated histograms consist of 100 bins when estimating essential matrices and 225 bins for fundamental matrices. The matches are filtered by using an SNN ratio threshold of 0.9. All methods use the PROSAC sampler [10] on correspondences ordered by the SNN ratio.

For testing the methods, we use the problems and datasets from CVPR tutorial *RANSAC in 2020* [1]. The data are from the CVPR IMW 2020 PhotoTourism challenge. Correspondences are obtained using RootSIFT features and mutual nearest neighbour matching. For calculating the accuracy, we use all scenes from the test set, each containing 4950 image pairs. Thus, the accuracy is calculated on a total of 54 450 image pairs. The proposed learning-based techniques are trained and validated on the provided training set. For RANSAC, MSAC and MAGSAC++, we use the hyper-parameters tuned in [1].

We test three versions of MQ-Net depending on the problem it was trained on. We train it on fundamental matrix (**F**), essential matrix estimation (**E**) and on both problems simultaneously (**E** & **F**). In the **E** & **F** case, the same network runs for both fundamental and essential matrix estimation. Differently, MF-Net is either trained on fundamental matrix or on essential matrix.

	mAA@10° ↑		Median (°) ↓		Time (ms) ↓	
Model scoring	**R**	**t**	$\epsilon_{\mathbf{R}}$	$\epsilon_{\mathbf{t}}$	AVG	MED
RANSAC [12]	0.64	0.32	2.19	11.20	**1.12**	**2.10**
MSAC [41]	0.63	0.31	2.39	12.14	**1.35**	3.88
MAGSAC++ [6]	0.64	0.31	2.33	12.25	**1.35**	3.95
MQ-Net (E)	0.62	0.29	2.29	15.09	9.73	4.74
MQ-Net (F)	0.66	0.34	1.83	10.98	8.54	3.82
MQ-Net (E & F)	**0.70**	**0.35**	**1.67**	**10.45**	8.42	3.75
MQ-Net + MF-Net	**0.70**	**0.36**	**1.63**	**10.20**	1.76	**1.65**

Table 1. **Fundamental matrix estimation**. The reported values are the rotation and translation mAA@10° scores; median errors ($\epsilon_{\mathbf{R}}$ and $\epsilon_{\mathbf{t}}$) in degrees; and the run-times in milliseconds. MQ-Net (**E**) and (**F**) are trained, respectively, on essential and fundamental matrix estimation. MQ-Net (**E & F**) is trained on both problems. The last row shows the results with filtering by MF-Net.

5.1. Fundamental Matrix Estimation

To estimate fundamental matrices, we use the 7-point algorithm [15] as minimal solver and the normalized 8-point [16] one for estimating from a non-minimal sample.

Table 1 reports the rotation and translation mean Average Accuracy (mAA) at 10°; the median errors ($\epsilon_{\mathbf{R}}$ and $\epsilon_{\mathbf{t}}$) in degrees, and the run-times (t) in milliseconds of the entire robust estimation. The mAA score is calculated as the area under the recall curve cropped at 10°. All three variants of MQ-Net lead to a significantly improved accuracy compared to the traditional techniques. The best results are achieved by the network trained on both problems simultaneously. The median rotation error is the 72% of the MAGSAC++ error. The median translation error is decreased by 2.05 degrees. MF-Net is able to accelerate the method by five times while improving the accuracy as well. MQ-Net combined with MF-Net is both *faster* and *more accurate* than the traditional methods.

The recall curves are shown in Fig. 4 (top). The success threshold (horizontal axis; in degrees) defines the error upper bound for a relative pose to be considered as accurate. The error is calculated as the maximum of the rotation and translation errors. The vertical axis shows the ratio of poses considered accurate by using a particular success threshold.

5.2. Essential Matrix Estimation

For estimating essential matrices, we use the 5-point algorithm [37] as minimal solver. In the final model polishing stage, we optimize the pose with the Levenberg-Marquardt numerical optimization [26] minimizing the pose error.

Table 2 reports the rotation and translation mAA@10° scores, median errors ($\epsilon_{\mathbf{R}}$ and $\epsilon_{\mathbf{t}}$) in degrees, and the run-times (t) in milliseconds of the entire robust estimation procedure. All three variants of the proposed algorithm lead to better accuracy than the traditional techniques. Again, the best results are achieved by the network trained on both

	mAA@10° ↑		Median (°) ↓		Time (ms) ↓	
Model scoring	**R**	**t**	$\epsilon_{\mathbf{R}}$	$\epsilon_{\mathbf{t}}$	AVG	MED
RANSAC [12]	0.70	0.46	1.76	5.41	**1.64**	**1.61**
MSAC [41]	0.71	0.47	1.67	5.21	**1.94**	**2.73**
MAGSAC++ [6]	0.71	0.47	1.64	5.03	1.96	2.69
MQ-Net (E)	0.76	0.61	0.99	2.56	7.43	5.94
MQ-Net (F)	0.76	0.61	0.98	2.51	6.62	5.43
MQ-Net (E & F)	**0.78**	**0.62**	**0.94**	**2.40**	5.38	3.75
MQ-Net + MF-Net	**0.79**	**0.62**	**0.91**	**2.34**	4.33	3.35

Table 2. **Essential matrix estimation**. The reported values are the rotation and translation mAA@10° scores; median errors ($\epsilon_{\mathbf{R}}$ and $\epsilon_{\mathbf{t}}$) in degrees; and the run-times in milliseconds. MQ-Net (**E**) and (**F**) are trained, respectively, on essential and fundamental matrix estimation. MQ-Net (**E & F**) is trained on both problems. The last row shows the results with filtering by MF-Net.

problems. Both median errors and mAA@10° are improved by a *large margin*. The median error of the proposed algorithm is the half of the error of the traditional methods. MF-Net accelerates the robust estimation significantly while also improving accuracy. We argue that early sample rejection plays a larger role in harder problems with a higher ratio of outlier minimal samples, which is the case for fundamental matrix compared to essential matrix. We substantiate this claim with further experiments in the supplementary, where we observed speedups up to an order of magnitude in harder settings.

The recall curves are in the bottom of Fig. 4. The success threshold (horizontal axis; in degrees) defines what error is accepted as a success. The error is the maximum of the rotation and translation errors. The vertical axis is the ratio of poses that are considered successful using the a particular success threshold.

5.3. Sample Rejection

We tune the confidence threshold ϵ_{conf} of MF-Net on the 4950 image pairs of scene Notre Dame Front Facade from the training set of [1]. We compare the traditional algorithm that runs chirality and degeneracy checks after the model is estimated; and the proposed one that first runs MF-Net, estimates the model from the samples that survived and, finally, applies the traditional checks to the estimated models. Confidence threshold $\epsilon_{\text{conf}} \in [0, 1]$ is used to reject samples where the predicted confidence is smaller than ϵ_{conf}. The traditional algorithm with no deep filtering runs if $\epsilon_{\text{conf}} = 0$. For this experiment, we did not filter by the SNN ratio.

Fig. 5 shows the ratios of the results of the proposed and traditional techniques as a function of ϵ_{conf}. The shown properties are the rotation ($\epsilon_{\mathbf{R}}$) and translation ($\epsilon_{\mathbf{t}}$) errors and run-times. The vertical lines are placed so the filtering leads to the best accuracy. For essential matrices, setting the threshold to 0.8 leads to almost an *order-of-magnitude* speed-up. The accuracy is improved by 20% on average.

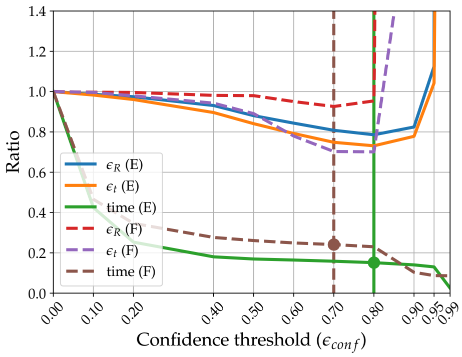

Figure 5. **MF-Net threshold influence**. The rotation ($\epsilon_{\mathbf{R}}$) and translation ($\epsilon_{\mathbf{t}}$) errors and the run-time divided by the $\epsilon_{\text{conf}} = 0$ case and plotted as a function of filtering confidence threshold ϵ_{conf} for essential (E) and fundamental (F) matrix estimation. Case $\epsilon_{\text{conf}} = 0$ means that no deep filtering is applied. The vertical lines are placed so the threshold leads to the best accuracy.

For fundamental matrices, the threshold leading to the most accurate results is 0.7 which reduces the run-time to almost one fifth. The accuracy is improved by 17% on average.

5.4. Ablation Study: Histogram Size

We test the proposed scoring approach with histograms of different sizes. The mean and median rotation and translation errors and run-times of essential matrix estimation are reported in Table 3. We tested networks trained either on essential or fundamental matrix estimation and, also, on both problems simultaneously. The histogram sizes are in the first column. The mAA score increases together and the median error inverse proportionally with the histogram size – the denser the histograms, the lower the error. The best results are obtained by histograms consisting of 2500 bins. Note that, while it is possible to run with bigger histograms, it is both memory and time consuming.

We run the same test for fundamental matrix estimation. The mAA scores, median rotation and translation errors and run-times are reported in Table 4. Again, we tested MQ-Net trained either on **E** or **F** estimation and, also, on both problems simultaneously. Interestingly, the histogram size has an opposite effect on the results when compared with essential matrix estimation. The smaller the histogram, the better the results. The best results are obtained by either 100 or 225 bins. In the tests, we chose 225 since its average results are marginally more accurate than using 100 bins.

5.5. Ablation Study: Model Re-estimation

The results on **E** and **F** estimation are shown, respectively, in Tables 5 and 6. Four strategies are tested: no final model polishing; using all inliers closer than $\mathcal{I}_{\epsilon_{\text{max}}}$; using the algorithm proposed in Section 3 without grouping the residuals ($\mathcal{I}_{\epsilon_{\text{no}}}$) and with 10 groups ($\mathcal{I}_{\epsilon_{10}}$).

	mAA@10° ↑		Median (°) ↓		Time (ms) ↓	
trained on **E**	**R**	**t**	$\epsilon_{\mathbf{R}}$	$\epsilon_{\mathbf{t}}$	AVG	MED
100	0.75	0.60	1.05	2.65	**3.52**	**2.30**
225	0.76	0.60	1.03	2.69	**3.63**	**2.38**
625	0.76	**0.61**	1.00	2.61	3.75	2.47
1600	0.76	**0.61**	0.99	2.56	4.79	2.59
2500	0.76	**0.61**	1.00	2.62	4.17	2.76
trained on **F**						
100	0.75	0.60	1.01	2.60	4.36	3.01
225	0.75	0.60	1.02	2.59	4.50	3.03
625	0.75	**0.61**	1.01	2.58	4.54	3.13
1600	0.75	**0.61**	0.99	2.56	5.07	3.76
2500	0.76	**0.61**	0.98	2.51	5.40	3.76
trained on **E & F**						
100	**0.77**	**0.62**	**0.95**	**2.39**	4.35	3.01
225	**0.77**	**0.62**	0.96	2.44	4.49	3.03
625	**0.77**	**0.62**	0.96	2.47	4.49	3.11
1600	**0.77**	**0.62**	**0.95**	2.45	5.07	3.51
2500	**0.78**	**0.62**	**0.94**	**2.40**	5.38	3.75

Table 3. **Histogram size for E estimation**. The rotation and translation mAA@10° scores; median errors ($\epsilon_{\mathbf{R}}$ and $\epsilon_{\mathbf{t}}$; in °); and the run-times (in ms) are plotted as a function of the histogram size.

	mAA@10° ↑		Median (°) ↓		Time (ms) ↓	
trained on **E**	**R**	**t**	$\epsilon_{\mathbf{R}}$	$\epsilon_{\mathbf{t}}$	AVG	MED
100	0.62	0.29	2.29	15.09	**5.94**	**2.67**
225	0.61	0.27	2.44	15.57	5.96	2.72
625	0.59	0.26	2.58	16.16	6.09	2.73
1600	0.56	0.23	3.35	17.35	6.38	2.86
2500	0.52	0.21	4.09	18.34	6.78	2.99
trained on **F**						
100	0.66	**0.34**	1.83	10.98	8.09	**2.44**
225	0.66	**0.34**	1.84	11.22	8.45	3.75
625	0.66	0.32	1.86	11.77	8.56	3.90
1600	0.63	0.32	2.06	12.09	8.69	3.98
2500	0.65	0.30	1.96	12.09	9.17	4.20
trained on **E & F**						
100	**0.70**	**0.35**	**1.69**	**10.29**	8.08	3.66
225	**0.70**	**0.35**	**1.67**	**10.45**	8.42	3.75
625	**0.69**	**0.34**	1.76	10.68	8.12	3.72
1600	0.64	0.29	2.06	10.99	**5.94**	3.98
2500	0.64	0.29	2.01	11.14	9.16	4.20

Table 4. **Histogram size for F estimation**. The rotation and translation mAA@10° scores; median errors ($\epsilon_{\mathbf{R}}$ and $\epsilon_{\mathbf{t}}$; in °); and the run-times (in ms) are plotted as a function of the histogram size.

	mAA@10° ↑		Median (°) ↓		Time (ms) ↓	
LSQ	**R**	**t**	$\epsilon_{\mathbf{R}}$	$\epsilon_{\mathbf{t}}$	AVG	MED
w/o	0.55	0.23	3.57	16.78	**6.83**	**3.13**
$\mathcal{I}_{\epsilon_{\max}}$	0.67	0.31	2.18	11.35	**7.04**	**3.21**
$\mathcal{I}^*_{\text{no}}$	**0.74**	**0.38**	**1.33**	**9.03**	28.49	7.79
$\mathcal{I}^*_{10}$	**0.70**	**0.36**	**1.63**	**10.20**	8.42	3.75

Table 5. **Model re-estimation on fundamental matrices**. Rotation and translation mAA@10° scores; median errors ($\epsilon_{\mathbf{R}}$ and $\epsilon_{\mathbf{t}}$; in degrees); and the run-times (in ms) of fundamental matrix estimation with different final re-fitting strategies: no polishing (w/o); re-fitting on all inlier closer than the max. threshold ($\mathcal{I}_{\epsilon_{\max}}$); re-fitting with the proposed technique without grouping the residuals ($\mathcal{I}^*_{\text{no}}$); proposed method with 10 groups ($\mathcal{I}^*_{10}$).

	mAA@10° ↑		Median (°) ↓		Time (ms) ↓	
LSQ	**R**	**t**	$\epsilon_{\mathbf{R}}$	$\epsilon_{\mathbf{t}}$	AVG	MED
w/o	0.66	0.45	2.22	5.47	**4.25**	**2.17**
$\mathcal{I}_{\epsilon_{\max}}$	0.16	0.07	30.09	34.29	**4.25**	**2.17**
$\mathcal{I}^*_{\text{no}}$	**0.79**	**0.62**	**0.99**	**2.45**	27.97	10.38
$\mathcal{I}^*_{10}$	**0.78**	**0.62**	**0.91**	**2.34**	5.38	3.73

Table 6. **Model re-estimation on essential matrices**. The rotation and translation mAA@10° scores; median errors ($\epsilon_{\mathbf{R}}$ and $\epsilon_{\mathbf{t}}$; in degrees); and the run-times (in ms) of essential matrix estimation with different final re-fitting strategies: no polishing (w/o); re-fitting on all inlier closer than the max. threshold ($\mathcal{I}_{\epsilon_{\max}}$); re-fitting with the proposed technique without grouping the residuals ($\mathcal{I}^*_{\text{no}}$); proposed method with 10 groups ($\mathcal{I}^*_{10}$).

Without re-fitting, the results are purely the accuracy of the minimal sample models scored the best by the proposed technique without any LSQ re-estimating the model parameters. When the models are re-estimated from all inliers with lower than $\epsilon_{\max}$ residuals, the results are, as expected, inaccurate. This justifies the need for an adaptive inlier selection strategy. Applying the proposed strategy without grouping the residuals is extremely accurate but five times slower than the other variants. Grouping the residuals into 10 groups and, thus, doing only 10 non-minimal model estimations leads to similar accuracy while being fast.

6. Conclusion

We propose two new learning-based approaches MQ-net and MF-Net to improve the robust estimation accuracy by learning to find models with small errors, and to speed it up by rejecting minimal samples early. MQ-Net, together with a new adaptive model re-estimation strategy and MF-Net, leads to results superior to the state-of-the-art by a large margin while running faster than its less accurate alternatives. MQ-Net using a single model trained jointly on essential and fundamental matrix estimation leads to the most accurate results on both problems on thousands of image pairs. The algorithms can be straightforwardly plugged into state-of-the-art RANSAC pipelines, *e.g.*, VSAC [18]. Moreover, we demonstrate an interesting property of such robust estimation problems: consensus maximization does not necessarily lead to the most accurate relative poses.

Acknowledgments. This work was supported by the ETH Zurich Postdoctoral Fellowship and the Google Focused Research Award.

References

[1] D. Barath, T-J. Chin, O. Chum, D. Mishkin, R. Ranftl, and J. Matas. RANSAC in 2020 tutorial. In *CVPR*, 2020. 6, 7

[2] D. Barath and J. Matas. Progressive-X: Efficient, anytime, multi-model fitting algorithm. In *ICCV*, October 2019. 1

[3] D. Barath, D. Mishkin, I. Eichhardt, I. Shipachev, and J. Matas. Efficient initial pose-graph generation for global SfM. In *CVPR*, pages 14546–14555, 2021. 1

[4] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *CVPR*, 2020. 3

[5] D. Barath, J. Noskova, and J. Matas. MAGSAC: marginalizing sample consensus. In *CVPR*, 2019. https://github.com/danini/magsac. 3

[6] D. Barath, J. Noskova, and J. Matas. Marginalizing sample consensus. *IEEE TPAMI*, 2021. 2, 3, 6, 7

[7] E. Brachmann and C. Rother. Neural-guided RANSAC: Learning where to sample model hypotheses. In *CVPR*, pages 4322–4331, 2019. 5

[8] M. Bujnak, Zu. Kukelova, and T. Pajdla. Robust focal length estimation by voting in multi-view scene reconstruction. In *ACCV*, pages 13–24. Springer, 2009. 2

[9] Nontawat Charoenphakdee, Jayakorn Vongkulbhisal, Nuttapong Chairatanakul, and Masashi Sugiyama. On focal loss for class-posterior probability estimation: A theoretical perspective. In *Conference on Computer Vision and Pattern Recognition*, pages 5202–5211, 2021. 4

[10] O. Chum and J. Matas. Matching with PROSAC-progressive sample consensus. In *CVPR*. IEEE, 2005. 6

[11] O. Chum, T. Werner, and J. Matas. Two-view geometry estimation unaffected by a dominant plane. In *CVPR*. IEEE, 2005. 2, 5, 6

[12] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 1, 6, 7

[13] J-M Frahm and Marc Pollefeys. Ransac for (quasi-) degenerate data (qdegsac). In *CVPR*, volume 1, pages 453–460. IEEE, 2006. 5

[14] D. Ghosh and N. Kaabouch. A survey on image mosaicking techniques. *Journal of Visual Communication and Image Representation*, 2016. 1

[15] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 3, 6

[16] R. I. Hartley. In defense of the eight-point algorithm. *TPAMI*, 1997. 6

[17] H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *IJCV*, 2012. 1

[18] M. Ivashechkin, D. Barath, and J. Matas. VSAC: Efficient and accurate estimator for h and f. *ICCV*, 2021. 2, 8

[19] F. Kluger, E. Brachmann, H. Ackermann, C. Rother, M. Y. Yang, and B. Rosenhahn. CONSAC: Robust multi-model fitting by conditional sample consensus. In *CVPR*, pages 4634–4643, 2020. 1

[20] H. M. Le, T-J. Chin, A. Eriksson, T-T. Do, and D. Suter. Deterministic approximate methods for maximum consensus robust fitting. *TPAMI*, 2019. 2

[21] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*. IEEE, 1999. 3

[22] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan. Image matching from handcrafted to deep features: A survey. *IJCV*, 129(1):23–79, 2021. 2

[23] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *IVC*, 2004. 1

[24] D. Mishkin, J. Matas, and M. Perdoch. MODS: Fast and robust method for two-view matching. *CVIU*, 2015. 1

[25] K. Moo Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua. Learning to find good correspondences. In *CVPR*, pages 2666–2674, 2018. 5

[26] Jorge J. Moré. The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978. 6

[27] T. T. Pham, T-J. Chin, K. Schindler, and D. Suter. Interacting geometric priors for robust multimodel fitting. *TIP*, 2014. 1

[28] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *ICCV*. IEEE, 1998. 1

[29] T. Probst, D. P. Paudel, A. Chhatkuli, and L. V. Gool. Unsupervised learning of consensus maximization for 3d vision problems. In *CVPR*, pages 929–938, 2019. 2

[30] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 5

[31] R. Ranftl and V. Koltun. Deep fundamental matrix estimation. In *ECCV*, pages 284–299, 2018. 5

[32] J. Schonberger and J-M. Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 3

[33] J. Schönberger, E. Zheng, M. Pollefeys, and J-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1

[34] Johannes Lutz Schönberger and J-M. Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1

[35] C. Sminchisescu, D. Metaxas, and S. Dickinson. Incremental model-based estimation using geometric constraints. *TPAMI*, 2005. 1

[36] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006. 2

[37] H. Stewénius, D. Nistér, F. Kahl, and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. *IVC*, 26(7):871–877, 2008. 6

[38] R. Tennakoon, D. Suter, E. Zhang, T-J. Chin, and A. Bab-Hadiashar. Consensus maximisation using influences of monotone boolean functions. In *CVPR*, pages 2866–2875, 2021. 2

[39] P. H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *IJCV*, 2002. 2, 3

[40] P. H. S. Torr and D. W. Murray. Outlier detection and motion segmentation. In *Optical Tools for Manufacturing and Advanced Automation*. International Society for Optics and Photonics, 1993. 1

[41] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *CVIU*, 2000. 1, 2, 3, 6, 7

[42] P. H. S. Torr, A. Zisserman, and S. J. Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *CVIU*, 1998. 1

[43] E. Trulls, Y. Jun, K. Yi, D. Mishkin, J. Matas, and P. Fua. Image matching challenge. In *CVPR*, 2020. 3

[44] B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *International Conference on Data Mining*, pages 435–442. IEEE, 2003. 5

[45] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao. Learning two-view correspondences and geometry using order-aware network. In *CVPR*, pages 5845–5854, 2019. 5

[46] M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multiRANSAC algorithm and its application to detect planar homographies. In *ICIP*. IEEE, 2005. 1