Adaptive Reordering Sampler with Neurally Guided MAGSAC

Tong Wei[1], Jiří Matas[1], and Daniel Barath[2]

[1] Visual Recognition Group, FEE, Czech Technical University in Prague
[2] Computer Vision and Geometry Group, ETH Zurich

{weitong, matas}@fel.cvut.cz, danielbela.barath@inf.ethz.ch

Abstract

We propose a new sampler for robust estimators that always selects the sample with the highest probability of consisting only of inliers. After every unsuccessful iteration, the inlier probabilities are updated in a principled way via a Bayesian approach. The probabilities obtained by the deep network are used as prior (so-called neural guidance) inside the sampler. Moreover, we introduce a new loss that exploits, in a geometrically justifiable manner, the orientation and scale that can be estimated for any type of feature, e.g., SIFT or SuperPoint, to estimate two-view geometry. The new loss helps to learn higher-order information about the underlying scene geometry. Benefiting from the new sampler and the proposed loss, we combine the neural guidance with the state-of-the-art MAGSAC++. Adaptive Reordering Sampler with Neurally Guided MAGSAC (ARS-MAGSAC) is superior to the state-of-the-art in terms of accuracy and run-time on the PhotoTourism and KITTI datasets for essential and fundamental matrix estimation. The code and trained models are available at https://github.com/weitong8591/ars_magsac.

1. Introduction

Robust estimation of two-view geometry is a fundamental problem in numerous computer vision applications, *e.g.*, wide baseline matching [39, 33, 35], multi-model fitting [24, 38, 9], initial pose recovery of Structure-from-Motion [46, 47] and Simultaneous Localization and Mapping (SLAM) [36] pipelines. RANSAC (RANdom SAmple Consensus) [19], and its recent variants [40, 11, 10, 25, 4] have been widely applied in practice due to efficiency, simplicity, and accuracy that makes them appealing in real-world scenarios. In brief, RANSAC works in iterations by selecting a subset of data points, estimating the model (*e.g.*, relative pose of a camera pair), and measuring its quality as the number of points consistent with it (*i.e.*, its inliers).

Since the publication of RANSAC, several modifications have been proposed to improve its accuracy and speed, fo-

cusing on specific components of the original algorithm. One common way to increase the accuracy is considering realistic noise distributions in model scoring, rather than relying on inlier counting that essentially assumes uniform noise. MLESAC [54] uses a maximum likelihood procedure to reason about the quality of each minimal sample model. While MLESAC can achieve better accuracy than standard inlier counting, it can be computationally expensive compared to the original RANSAC. MSAC [53] proposes the use of truncated L_2 loss, which has been shown to be equivalent to the maximum likelihood-based approaches, but it still heavily relies on a manually set threshold. More recently, MAGSAC [11] and MAGSAC++ [10] have been proposed to alleviate the dependence on manual threshold selection by marginalizing over an acceptable range of noise scales. As shown in the recent survey [32], MAGSAC++ is currently the most accurate RANSAC variant as in [25].

Improving the sampling procedure is another way to enhance the performance of RANSAC by selecting a good sample early and triggering the termination criterion. Several samplers have been proposed, each with its own assumptions and limitations. The NAPSAC [52] sampler assumes that inliers are spatially coherent and, thus, it draws samples from a hyper-sphere centered at the first, randomly selected, location-defining point. The GroupSAC algorithm [37] assumes that inliers are often "similar" and, thus, can be separated into groups. PROSAC [16] exploits an a priori predicted inlier probability rank of each point and starts the sampling with the most promising ones. Progressively, samples that are less likely to lead to the sought model are drawn. More recently, [15] proposes a sampler to exploit prior knowledge of inlier probabilities, *e.g.*, from a deep network. The sampler selects minimal samples according to the probability, assuming that it follows a categorical distribution over the discrete set of observations.

Recently, several algorithms have been proposed for robust relative pose estimation using neural networks. The first paper on the topic, Context Normalization Networks (PointCN) [60] proposes the use of PointNet (MLP) with batch normalization as a context mechanism. Attentive
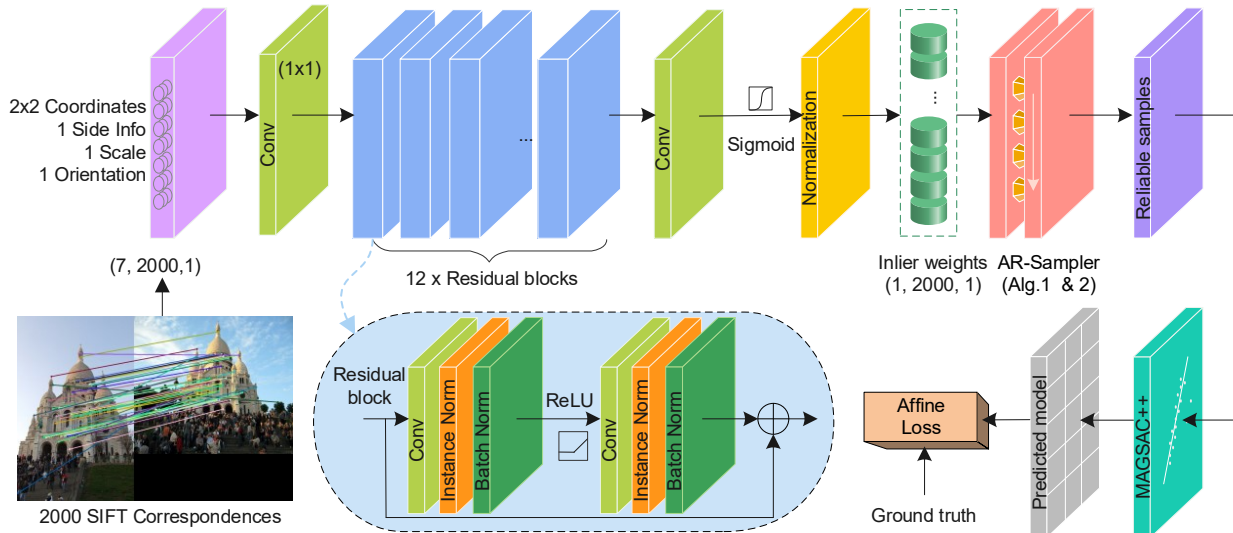
Figure 1. ARS-MAGSAC. The point coordinates, SIFT orientations and scales and other information (*e.g.*, SNN ratio [30]) are fed into the network. The predicted inlier probabilities used as priors are updated by the proposed AR-Sampler (see Sec. 5.4). The estimated model, *e.g.* relative pose, is used to calculate the loss that combines the proposed Affine Loss (see Sec. 3.2) and other ones.

Context Normalization Networks [50] introduces a special architectural block for the task. The Deep Fundamental Matrix Estimation (DFE) [41] uses differentiable iteratively re-weighted least-squares with predicted weights. The OANet algorithm [61] introduces differentiable pooling and un-pooling blocks for correspondence filtering. Neural Guided RANSAC (NG-RANSAC) [15] uses a PointCN-like architecture with a different training objective, and the predicted correspondence scores are exploited inside RANSAC by using a guided sampling method that helps to find accurate models early. More recently, CLNet [62] introduces several algorithmic and architectural improvements to remove gross outliers with iterative pruning. These techniques provide alternatives for tentative correspondence pre-filtering and weighting. While these methods have shown promising results, they do not substitute standard robust estimation, as demonstrated in the RANSAC tutorial [5].

This paper makes three main contributions. *First*, we propose a new sampler tailored for neurally guided robust estimators always selecting the sample with the highest probability of consisting only of inliers. It exploits the inlier probabilities obtained by deep networks as prior knowledge and updates them in a principled way via a Bayesian approach. *Second*, we propose a new loss that incorporates the orientation and scale from off-the-shelf feature detectors, *e.g.*, SIFT [31], in a geometrically meaningful manner directly into the training. Additionally, we show that this extra information can be obtained for any features, *e.g.* SuperPoint [17], allowing to seamlessly integrate the proposed loss with recent detectors. The loss allows learning higher-order information about the underlying scene geometry and improves the robustness of the estimation in challenging en-

vironments. *Third*, as a technical contribution, we combine two state-of-the-art algorithms, *i.e.*, MAGSAC++ [10] and NG-RANSAC [15], to improve the relative pose estimation accuracy on a wide range of scenes.

2. Adaptive Re-ordering Sampler

In this section, we describe the proposed new sampler that always selects the sample with the highest probability of containing only inliers. This probability is updated adaptively according to the success or failure of the current minimal sample in robust estimation. The new sampler will be called **AR-Sampler** in the remainder of the paper.

Let us assume that we are given point correspondences $p_{i_1^t}$, $p_{i_2^t}$, ..., $p_{i_n^t} \in \mathcal{P}$ with inlier probabilities $\mu_{i_1^t}$, $\mu_{i_2^t}$, ..., $\mu_{i_n^t} \in [0, 1]$ such that $\mu_{i_1^t} \geq \mu_{i_2^t} \geq \cdots \geq \mu_{i_n^t}$, where $i_1^t, \ldots, i_n^t \in [1, n]$ are indices in the tth RANSAC iteration ensuring that the points are ordered by the inlier probabilities in a descending order. The probability of sample $S = (p_{j_1}, p_{j_1}, \ldots, p_{j_m}) \in \mathcal{P}^{\times}$ consisting only of inliers is calculated as $\mu_S = \prod_{k=1}^{m} \mu_{j_k}$, assuming independence, where m is the sample size, *e.g.*, $m = 5$ for essential matrix estimation. The independence assumption is incorrect for samples that include any subset of points that have previously been tested in another minimal sample, but it turns out to be a tractable and useful approximation. Consequently, the *globally* optimal sampler maximizes the sample probability of $S_t^* = (p_{i_1^t}, p_{i_1^t}, \ldots, p_{i_m^t})$ in the tth iteration to increase the probability of finding the sought model early.

Every unsuccessful RANSAC iteration reduces the inlier probability of the points in the minimal sample. This stems from the fact that in the case of having an all-inlier sample that is good enough to find the sought model, RANSAC ter-

Algorithm 1 Probability Update.

Input: $p_1, \ldots, p_n$ – points; $\mu_1, \ldots, \mu_n$ – probabilities
$\quad\quad$ S – minimal sample; $n_1, \ldots, n_p$ – usage numbers
$\quad\quad$ $(a_1, b_1), \ldots, (a_n, b_n)$ – initial distribution params
Output: $\mu'_1, \ldots, \mu'_n$ – updated inlier probabilities

1: **for** $i \in [1, n]$ **do**
2: $\quad$ **if** $p_i \in S$ **then** $\quad \triangleright$ Decrease μ for all sampled points
3: $\quad\quad$ $a'_i \leftarrow a_i; b'_i \leftarrow b_i + n_i$
4: $\quad\quad$ $\mu'_i \leftarrow a'_i/(a'_i + b'_i)$
5: $\quad$ **else** $\quad\quad \triangleright$ Other points have the same μ as before
6: $\quad\quad$ $\mu'_i \leftarrow \mu_i$

Algorithm 2 Adaptive Re-ordering Sampler.

Input: $p_1, \ldots, p_n$ – points; $\mu_1, \ldots, \mu_n$ – inlier probs.
$\quad\quad$ m – sample size; $n_1, \ldots, n_p$ – usage numbers
$\quad\quad$ $(a_1, b_1), \ldots, (a_n, b_n)$ – initial distribution param.
Output: S^* – minimal sample

1: $i_1, \ldots, i_n \leftarrow \text{reorder}(\mu_1, \ldots, \mu_n)$ $\triangleright$ By the inlier prob.
2: $S^* \leftarrow \{p_{i_j} \mid j \in [1, m]\}$
3: **for** $p_{i_j} \in S^*$ **do**
4: $\quad$ $n_{i_j} \leftarrow n_{i_j} + 1$
5: $\quad$ $\mu_{i_j} \leftarrow \text{Update}(a_{i_j}, b_{i_j}, \mu_{i_j}, n_{i_j})$ $\quad\quad \triangleright$ Algorithm 1

minates.[1] The sample that does not trigger the termination is not all-inlier (failure). Therefore, of all the possible inlier-outlier configurations in the sample, the "all points are inliers" is ruled out, and, consequently, the inlier probabilities of the sample points decrease, typically very modestly. In order to model this in a principled way, we update the probabilities using the Bayesian approach after each RANSAC iteration. We note that the Bayesian approach ignores the dependencies between points that appeared in a sample. As prior knowledge, we can either consider the output of the deep network or even the point ordering that the SNN ratio [30] implies. In each update, only the points from the current sample are considered, and, thus, the probability of other points remains unchanged in the $t + 1$-th iteration as we did not gather additional information about them.

The probability of point **p** being inlier in the tth iteration follows the Bernoulli distribution. Consequently, the number of times point **p** is being selected in an outlier-contaminated sample when selected n_p times follows the binomial distribution with parameters $\mu_p(n_p)$ and n_p. The usual conjugate prior for a binomial distribution is a beta distribution with prior hyper-parameters $a(n_p)$ and $b(n_p)$,

---

with expectation $a(n_p)/(a(n_p) + b(n_p))$, variance

$$v = \frac{a(n_p)b(n_p)}{(a(n_p) + b(n_p))^2(a(n_p) + b(n_p) + 1)},$$

and posterior hyper-parameters $a(n_p)$ and $b(n_p)$. The posterior distribution parameters are $a(n_p + 1) = a(n_p)$, $b(n_p+1) = b(n_p)+1$. Since $a(n_p)$ is constant, we will simply write a in the rest of the paper. The best estimator for $\mu_p(n_p + 1)$ using a quadratic loss function is an expectation of the posterior distribution. Consequently,

$$\mu_p(n_p + 1) = \frac{a}{a + b(n_p + 1)}. \quad (1)$$

For each point **p**, the initial parameters of the beta distribution a and $b(1)$ are set using the predicted inlier ratio $\mu_p(1) = \mu_p^1$. We assume that the inlier probability prediction provides the expectation of the prior beta distribution and with the same mean precision for all points. Thus, the variance v of all these initial beta distributions is equal and can be learned in advance. Given the learned variance,

$$\mu_p(1) = \frac{a}{a + b(1)}, v = \frac{ab(1)}{(a + b(1))^2(a + b(1) + 1)} \quad (2)$$

leads to

$$a = \frac{(\mu_p(1))^2(1 - \mu_p(1))}{v} - \mu_p(1), \; b(1) = a\frac{1 - \mu_p(1)}{\mu_p(1)}.$$

Parameters a and $b(1)$ are calculated prior to the robust estimation. The probability update and the sampler are shown, respectively, in Algorithms 1 and 2. Both methods contain only a few calculations and, thus, are very efficient, shown in Sec. 5.4. This is expected from a sampler in a RANSAC-like estimator where it runs in every iteration, often thousands of times. Note that we found that the sampler works better with probabilities shuffled by adding a small random number ϵ. Setting ϵ so it is uniformly distributed in-between $\pm 5e^{-4}$ works well in all our experiments.

3. Scale and Orientation Loss

We propose a new loss function considering that, in most of the two-view cases, we apply feature detectors that provide more information about the underlying scene geometry than simply the point coordinates. For instance, ORB [45] features contain the orientation of the image patches centered on the detected points in the two images. In addition to the feature orientation, the SIFT [30] and SURF [13] detectors return a uniform scaling parameter. Even the full affine warping of the patch can be recovered when using affine-covariant feature detectors, *e.g.*, Hessian-Affine [34] or MODS [35]. Moreover, even the most recent features, *e.g.* SuperPoint [17], can be equipped with orientation and scale by applying the Self-Scale-Ori [29] method.
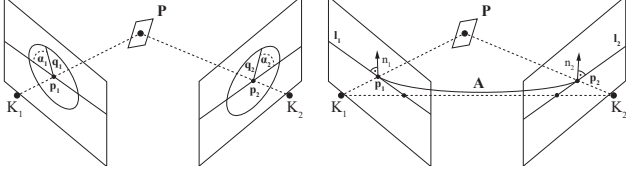
Figure 2. **(Left)** Visualization of the orientation- and scale-covariant features. Point **P** and the surrounding patch projected into cameras $\mathbf{K}_1$ and $\mathbf{K}_2$. The rotation of the feature in the ith image is $\alpha_i \in [0, 2\pi)$ with size $q_i \in \mathbb{R}$, $i \in \{1, 2\}$. **(Right)** The geometric interpretation of the relations of local affine transformations and the epipolar geometry (Eq. (3); proposed in [6]). The normal $\mathbf{n}_1$ of epipolar line $\mathbf{l}_1$ is mapped by affinity $\mathbf{A} \in \mathbb{R}^{2\times2}$ into the normal $\mathbf{n}_2$ of epipolar line $\mathbf{l}_2$.

This additional information that such features provide has not yet been exploited in a geometrically meaningful manner to minimize the training loss. Recent deep networks, *e.g.*, [15], use SNN ratios as side information added to the feature vectors or for pre-filtering correspondences.

3.1. Affine Epipolar Error

In order to interpret fully or partially affine-covariant features, we adopt the definition from [6] and the affine transformation model from [3]. We consider an affine correspondence (AC) a triplet: $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{A})$, where $\mathbf{p}_1 = [u_1, \ v_1, \ 1]^\mathrm{T}$ and $\mathbf{p}_2 = [u_2, \ v_2, \ 1]^\mathrm{T}$ are a corresponding homogeneous point pair in the two images, and

$$\mathbf{A} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} q_u & w \\ 0 & q_v \end{bmatrix}, \quad (3)$$

is a 2×2 linear transformation describing the relationship of the infinitesimal image patches centered on points $\mathbf{p}_1$ and $\mathbf{p}_2$, where α is rotation, q_u and q_v are the scales along the axes, and w is the shear parameter. Formally, $\mathbf{A}$ is defined as the first-order Taylor-approximation of the 3D $\rightarrow$ 2D projection functions. For perspective cameras, $\mathbf{A}$ is the first-order approximation of the related 3×3 homography.

The relationship of affine correspondences and epipolar geometry is shown in [42, 6], with [6] providing a geometrically interpretable definition of the constraint as

$$\mathbf{A}^{-\mathrm{T}}\mathbf{n}_1 = -\mathbf{n}_2 \quad (4)$$

where $\mathbf{n}_1 = (\mathbf{F}^\mathrm{T}\mathbf{p}_2)_{[1:2]}$ and $\mathbf{n}_2 = (\mathbf{F}\mathbf{p}_1)_{[1:2]}$ are the normals of the epipolar lines in the two images, and lower-index $\mathbf{v}_{[1:2]}$ selects the first two coordinates of a vector $\mathbf{v}$, as shown in the right plot of Fig. 2.

While constraint (4) is originally formulated as two linear equations in [6] to simplify the estimation, it can be rewritten to two geometrically meaningful constraints that we can use in the loss function. First, (4) implies that $\mathbf{A}^{-\mathrm{T}}$ rotates the normal in the first image to its corresponding pair in the second one as $(\mathbf{A}^{-\mathrm{T}}\mathbf{n}_1) \times \mathbf{n}_2 = 0$, where the angle between $\mathbf{A}^{-\mathrm{T}}\mathbf{n}_1$ and $\mathbf{n}_2$ can be used as an error for an

estimated fundamental matrix $\widehat{\mathbf{F}}$ as follows:

$$f(\mathbf{A}, \widehat{\mathbf{F}}, \mathbf{p}_1, \mathbf{p}_2) = \cos^{-1} \frac{\left(\mathbf{A}^{-\mathrm{T}}(\widehat{\mathbf{F}}^\mathrm{T}\mathbf{p}_2)_{[1:2]}\right)(\widehat{\mathbf{F}}\mathbf{p}_1)_{[1:2]}}{\left|\mathbf{A}^{-\mathrm{T}}(\widehat{\mathbf{F}}^\mathrm{T}\mathbf{p}_2)_{[1:2]}\right|\left|(\widehat{\mathbf{F}}\mathbf{p}_1)_{[1:2]}\right|}. \quad (5)$$

Second, (4) implies that the scale change is

$$\sqrt{\det \mathbf{A}} = \frac{|\mathbf{n}_2|}{|\mathbf{n}_1|} = \frac{\left|(\mathbf{F}\mathbf{p}_1)_{[1:2]}\right|}{\left|(\mathbf{F}^\mathrm{T}\mathbf{p}_2)_{[1:2]}\right|} \quad (6)$$

providing another geometrically meaningful error as

$$g(\mathbf{A}, \widehat{\mathbf{F}}, \mathbf{p}_1, \mathbf{p}_2) = \sqrt{\det \mathbf{A}} - \frac{\left|(\widehat{\mathbf{F}}\mathbf{p}_1)_{[1:2]}\right|}{\left|(\widehat{\mathbf{F}}^\mathrm{T}\mathbf{p}_2)_{[1:2]}\right|}. \quad (7)$$

Overall, these errors are used to measure the quality of the epipolar geometry given an affine correspondence.

3.2. Affine Loss Function

In practice, we are usually given partially affine-covariant features, *e.g.*, with orientation and scale, that do not allow using (5) and (7) directly. To define a justifiable loss, we first approximate the local affine frame $\widehat{\mathbf{A}}$ using the rotations α_1, α_2 and scales q_1, q_2 from the features via the affine transformation model in (3) assuming that shear $w = 0$, rotation $\alpha = \alpha_2 - \alpha_1$, and $q_u = q_v = q_2/q_1$ is a uniform scaling along the axes similarly as in [3], see the left plot of Fig. 2. It is important to note that directly using $\widehat{\mathbf{A}}$ to measure the error of the prediction is still not viable since $\widehat{\mathbf{A}}$ is only an approximation and, thus, (5) and (7) are not zero even if the ground truth fundamental matrix is used. We, thus, define the orientation and scale losses respectively as

$$L_{\mathrm{ori}}(\dots) = \left| f(\widehat{\mathbf{A}}, \widehat{\mathbf{F}}, \mathbf{p}_1, \mathbf{p}_2) - f(\widehat{\mathbf{A}}, \mathbf{F}, \mathbf{p}_1, \mathbf{p}_2) \right|,$$

$$L_{\mathrm{scale}}(\dots) = \left| g(\widehat{\mathbf{A}}, \widehat{\mathbf{F}}, \mathbf{p}_1, \mathbf{p}_2) - g(\widehat{\mathbf{A}}, \mathbf{F}, \mathbf{p}_1, \mathbf{p}_2) \right|,$$

where $\mathbf{F}$ is the ground truth fundamental matrix used as a target for the network and $\widehat{\mathbf{F}}$ is the prediction. Measuring the error in this way allows ignoring the approximative nature of $\widehat{\mathbf{A}}$. The final loss minimized is

$$L(\mathbf{F}, \widehat{\mathbf{F}}, \mathcal{P}) = \sum_{(\mathbf{p}_1, \mathbf{p}_2, \widehat{\mathbf{A}}) \in \mathcal{P}} w_{\mathrm{ori}} L_{\mathrm{ori}}(\mathbf{F}, \widehat{\mathbf{F}}, \widehat{\mathbf{A}}, \mathbf{p}_1, \mathbf{p}_2) +$$

$$w_{\mathrm{scale}} L_{\mathrm{scale}}(\mathbf{F}, \widehat{\mathbf{F}}, \widehat{\mathbf{A}}, \mathbf{p}_1, \mathbf{p}_2) + \cdots$$

where $\mathcal{P} = \{(\mathbf{p}_1, \mathbf{p}_2, \alpha_1, \alpha_2, q_1, q_2) \mid \mathbf{p}_1, \mathbf{p}_2 \in \mathbb{R}^2 \wedge \alpha_1, \alpha_2 \in [0, 2\pi) \wedge q_1, q_2 \in \mathbb{R}^+\}$ is the set of correspondences, w_{ori} and w_{scale} are weighting parameters, and $\cdots$ represents other metrics, *e.g.*, epipolar or pose error, or inlier ratio. To propagate the gradient, the training objective $L(w)$ is defined as the minimization of the expected task

loss, similarly as in [15]. Since integrating over all hypotheses to calculate the expectation is infeasible, the gradients for the categorical distribution over the discrete set of observations are approximated by drawing K samples as

$$\frac{\partial}{\partial w} L(w) \approx \frac{1}{K} \sum_{k=1}^{K} [l(\hat{h}) \frac{\partial}{\partial w} \log p(H_k; w)], \qquad (8)$$

where K is the number of samples used for gradient approximation. The task loss is $l(\hat{h})$ with $\hat{h}$ as the robust solver, and $p(H_k; w)$ is the learned distribution of the kth sample.

4. Neurally Guided MAGSAC

We combine NG-RANSAC [15] and MAGSAC++ [10] with the proposed algprithms. The pipeline is visualized in Fig. 1. Even though we will describe it focusing on epipolar geometry estimation, ARS-MAGSAC is *general*.

MAGSAC++ formulates the robust estimation problem as an iteratively re-weighted least-squares (IRLS) approach. Both the model quality calculation and inlier selection are done without making strict inlier-outlier decisions. The model parameters θ_{i+1} in the $(i + 1)$th step of the IRLS are calculated from the point-to-model residual function, $D(\theta_i, \mathbf{p})$, where $\mathbf{p}$ is a point from the input sets, as $\theta_{i+1} = \arg\min_\theta \sum_{\mathbf{p} \in \mathcal{P}} w(D(\theta_i, \mathbf{p})) D^2(\theta, \mathbf{p})$, where the weight of point $\mathbf{p}$ is determined by marginalizing over the noise scale σ as $w(D(\theta_i, \mathbf{p})) = \int_0^{+\infty} P(\mathbf{p} \mid \theta_i, \sigma) f(\sigma) d\sigma$ and $\theta_0 = \theta$, *i.e.*, the initial model from the minimal sample.

In order to improve MAGSAC++ using recent neural network-based techniques, we adopt the Neural Guided RANSAC (NG-RANSAC) architecture [15][2]. The NG-RANSAC algorithm predicts the probability of each point correspondence being inlier and uses a weighted sampling approach to incorporate this information in the robust estimation procedure. Due to the neural network and the robust estimator being loosely connected in such a manner, we can replace RANSAC with MAGSAC++ with all its bells and whistles and retrain the network. While training the weights with sparse correspondences end-to-end, the inlier masks and selected samples are used to update the gradients of the neurons and generate point probabilities as weights for the consequent epochs. We use additional side information as well, namely, the scale and orientation of each SIFT feature.

5. Experimental Results

We evaluate the accuracy and speed of ARS-MAGSAC and the impact of each individual improvement proposed in this paper, *e.g.*, AR-Sampler and affine loss. The compared methods are the OpenCV RANSAC [19] and LMEDS [43],

---

the implementations provided by the authors of GC-RANSAC [8], MAGSAC [11] and MAGSAC++ [10], NG-RANSAC [15], OANet [61], EAS [18], and CLNet [62]. We re-trained NG-RANSAC, OANet and CLNet using the same data as ARS-MAGSAC, explained in Sec. 5.1 and 5.3. Also, we will show their results with the provided models trained on significantly more image pairs than what we use for training ARS-MAGSAC. For fair comparison, we also run MAGSAC++ in the end of OANet and CLNet. All the experiments were conducted on Ubuntu 20.04 with GTX 3090Ti, OpenCV 4.5/3.4, and PyTorch 1.7.1.

Technical details. We use RootSIFT [2] features to improve the robust estimation accuracy and help the deep network to learn accurate weights. RootSIFT is a strategy normalizing the SIFT [30] descriptors, thus, helping the feature matcher to find good tentative correspondences. When training, we provide the network with the feature scales and orientations as a learned side information. Also, we do SNN ratio [30] filtering on the correspondences as a preliminary step. In the SNN test, the correspondences are discarded if the distance between the first and the second nearest neighbors is larger than a manually set threshold, which works well as we set to 0.8 in all of our experiments.

5.1. Essential Matrix Estimation

To test the essential matrix estimation of the proposed algorithm, we downloaded 13 scenes from the CVPR IMW 2020 PhotoTourism challenge [48]. These scenes were also used in the CVPR tutorial *RANSAC in 2020* [1] to compare robust estimators. The dataset contains tentative correspondences formed by mutual nearest neighbors matching Root-SIFT descriptors, ground-truth intrinsic camera parameters and relative poses. We use scene St. Peter's Square, consisting of 4950 image pairs, for training ARS-MAGSAC, retraining NG-RANSAC, CLNet, and OANet, and tuning the hyper-parameters of other compared methods. For testing, we use 1000 randomly chosen image pairs from each of the 12 scenes. Thus, the methods are tested on a total of 12 000 image pairs. For maximum reproducibility, we will provide these pairs together with the source code.

Before applying end-to-end training, we initialize our model by minimizing the Kullback–Leibler divergence [56] of the prediction and the target distribution using a 1000-epoch-long initial training process. To our experiments, this procedure improves the convergence speed of the end-to-end training. For the main experiments, ARS-MAGSAC is trained on RootSIFT correspondences for 10 epochs, with inlier-outlier threshold upper bound set to 0.75 pixel, Adam optimizer [26], batch size of 32, and 10^{-5} learning rate.

In each iteration of the training process, the pre-filtered correspondences are re-ordered according to the predicted weights, and MAGSAC++ with AR-Sampler is applied to estimate the $\mathbf{E}$ matrix. The pose error is calculated as the

Dataset / Method	LMEDS [44]	RSC [19]	GC-RSC [8]	MSC [11]	MSC++ [10]	EAS [18]	OANet [61] + MSC++	CLNet [62] + MSC++	NG-RSC [15]	ARS-MAGSAC
Avg. time (ms) ↓	**26.7**	88.1	175.1	239.4	113.4	325.8	49.1	57.5	79.8	33.9
Buckingham P.	0.19	0.20	0.20	0.27	0.26	0.13	0.19	0.27	0.28	**0.33**
Brandenburg G.	0.34	0.42	0.48	0.53	0.54	0.38	0.49	0.59	0.55	**0.61**
Colosseum E.	0.25	0.25	0.27	0.32	0.31	0.19	0.29	**0.38**	0.32	0.36
Grand Place B.	0.14	0.14	0.17	0.22	0.21	0.10	0.19	0.24	0.22	**0.32**
Notre Dame F.	0.24	0.27	0.38	0.40	0.41	0.24	0.38	**0.51**	0.34	0.49
Palace of W.	0.19	0.31	0.36	0.37	0.37	0.25	0.22	0.33	0.38	**0.43**
Pantheon E.	0.49	0.41	0.48	0.62	0.62	0.33	0.52	0.65	0.62	**0.72**
Prague Old T.	0.10	0.11	0.12	0.16	0.16	0.07	0.10	**0.20**	0.17	**0.20**
Sacre Coeur	0.52	0.64	0.68	0.71	0.71	0.65	0.58	0.69	0.63	**0.75**
Taj Mahal	0.36	0.48	0.52	0.52	0.55	0.47	0.48	0.62	0.55	**0.67**
Trevi Fountain	0.28	0.29	0.30	0.37	0.35	0.22	0.33	**0.48**	0.38	0.43
Westminster A.	0.46	0.36	0.49	0.51	0.51	0.33	0.43	0.54	0.49	**0.70**
All	0.30	0.32	0.37	0.42	0.42	0.28	0.35	0.46	0.41	**0.50**

(Left vertical label: Relative Pose AUC@10° ←)

Table 1. Essential matrix estimation on the PhotoTourism dataset [48]. We report the AUC scores [60] thresholded at $10°$ (higher is better) calculated from the pose error, *i.e.*, the maximum of the relative rotation and translation errors in degrees. The first row shows the average run-times (ms). The last one reports the scores averaged over all scenes. For RANSAC, GC-RANSAC, MAGSAC and MAGSAC++, we use the threshold as in [12]. Also, we tuned the threshold for EAS manually. We trained OANet, CLNet, NG-RANSAC, and ARS-MAGSAC on the same datasets. The results with the pre-trained models provided by the authors are in Tab. 2.

maximum of the rotation $\epsilon_{\widehat{\mathbf{R}}} = (180/\pi) \cos^{-1}((\text{tr}(\widehat{\mathbf{R}}\mathbf{R}^{\mathrm{T}}) - 1)/2)$, and the translation errors $\epsilon_{\hat{\mathbf{t}}} = (180/\pi) \cos^{-1} \frac{\mathbf{t}^{\mathrm{T}}\hat{\mathbf{t}}}{|\mathbf{t}||\hat{\mathbf{t}}|}$, in degrees, where $\widehat{\mathbf{R}} \in \mathrm{SO}(3)$ is the 3D rotation and $\hat{\mathbf{t}} \in \mathbb{R}^3$ is the translation, both decomposed from the estimated $\widehat{\mathbf{E}}$. Note that we use the angular translation error since the length of $\hat{\mathbf{t}}$ can not be recovered from two views [21]. Also, note that the scale and orientations of the features have to be normalized together with the point correspondences by the intrinsic camera matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ as proposed in [7]. The rotation remains unchanged. The scale is normalized by f_2/f_1, where f_i is the focal length of the ith camera.

We adopted the neural network from [15] and [60], a commonly used neural network for geometry data, which comprises 12 residual blocks that connect information from different layers and several multi-layer perceptions (MLPs). Each block is constructed by two linear layers, a batch normalization layer, and a ReLU activation function [22]. Besides, the global context is included by adding the instance normalization [55] layer into each block. The inlier probabilities of the matches are mapped by a Sigmoid function. Finally, MAGSAC++ with the proposed AR-Sampler estimates $\mathbf{E}$ with its iteration number fixed to 1000.

The training objective is defined as the minimization of the expected task loss [15]. We approximate the gradients for the categorical distribution over the discrete set of observations, shown in Eq. 8. The number of times each correspondence was selected in a minimal sample is back-propagated and used to update the weights and contribute to distribution learning for the next iteration.

PhotoTourism [48], RootSIFT [2]. To measure the accuracy of the estimated essential matrices, we decompose them to rotation and translation and calculate the pose error. Finally, we calculate the AUC scores at $5°$, $10°$ and $20°$ from the pose errors as the area under the recall

curves [15]. The AUC@$10°$ scores on each scene from the PhotoTourism dataset [48] are reported in Tab. 1. Also, we show the run-time (in milliseconds) and the AUC scores averaged over all scenes. The proposed ARS-MAGSAC is superior to the state-of-the-art on *all* Scenes, *i.e.* EAS, MAGSAC++, CLNet and NG-RANSAC, by a large margin both in terms of run-time and accuracy. Its average score is higher by 4 AUC points than that of the second most accurate method (*i.e.*, CLNet with MAGSAC++). The only faster method is LMEDS that has the second lowest accuracy. The left plot of Fig. 3 shows the cumulative distribution functions (CDF) of the pose errors of the estimated $\mathbf{E}$ matrices on all tested scenes. Being accurate is indicated by a curve close to the top-left corner. ARS-MAGSAC is significantly more accurate than the other compared methods.

Threshold	OANet [61]	CLNet [62]	NG-RANSAC [15]	ARS-MAGSAC
@$5°$	0.38	0.41	0.38	**0.47**
@$10°$	0.44	0.48	0.43	**0.50**
@$20°$	0.51	**0.56**	0.49	0.54

Table 2. AUC scores [60] of essential matrix estimation using the pre-trained models provided by the authors of NG-RANSAC, OANet and CLNet. CLNet and OANet were trained on 541 184 image pairs from the YFCC [51] dataset. NG-RANSAC was trained on 10 000 pairs from scene St. Peter's Square of Photo-Tourism dataset [48]. ARS-MAGSAC was trained on 4950 pairs.

Furthermore, Tab. 2 shows the performance comparison of ARS-MAGSAC with OANet, CLNet and NG-RANSAC when using the pre-trained models that their authors provide. CLNet and OANet was trained on 541 184 image pairs from the YFCC [51] dataset. NG-RANSAC was trained on a total of 10 000 pairs from the same scene as what we use for ARS-MAGSAC. As a reminder, ARS-MAGSAC was trained on 4950 image pairs in total. Even in this unfair comparison, the proposed method leads to the most accurate results in the AUC@$5°$ and AUC@$10°$ cases by a large
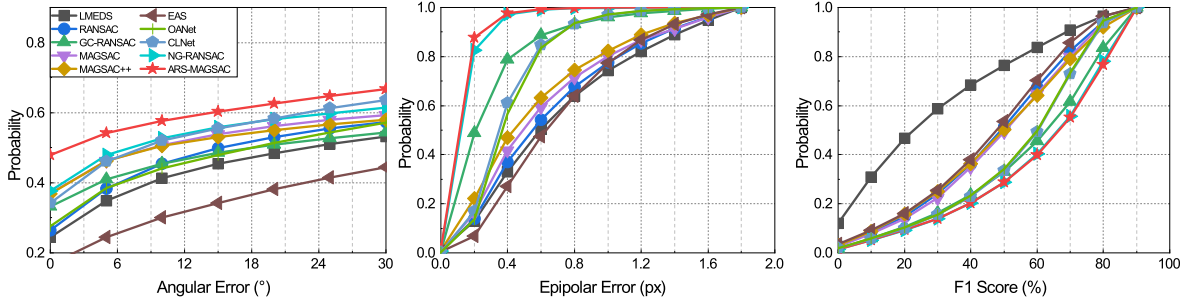
Figure 3. The cumulative distribution functions (CDF) of the angular errors (left) for **E** estimation; epipolar errors (middle; in pixels) and F1 score (right; in percentages) for **F** estimation. Essential matrix estimation was tested on 12 000 image pairs from the PhotoTourism [48] dataset. Fundamental matrix estimation is tested on 9690 pairs from the KITTI [20] dataset. We use the thresholds as in [12] for the traditional algorithms. We trained OANet, CLNet, NG-RANSAC, and ARS-MAGSAC on the same datasets. In the left two plots, being close to the top-left corner indicates accurate results. In the right one (F1 score), the bottom-right corner is preferable.

Method	Loss	AUC@5° ↑	AUC@10° ↑	AUC@20° ↑	Run-time (ms) ↓
MAGSAC++	-	0.378	0.427	0.480	134.33
ARS-MAGSAC	Pose	0.375	0.433	0.499	91.83
ARS-MAGSAC	Affine	**0.385**	**0.442**	**0.509**	**85.83**

Table 3. AUC scores and avg. run-times of MAGSAC++ [10] and the proposed ARS-MAGSAC on SuperPoint features [17] on 12 test scenes from the PhotoTourism dataset. Two versions of ARS-MAGSAC are shown, trained with the standard pose loss and the proposed affine loss on orientations and scales obtained by [59].

Threshold	Pose loss	Self-supervised loss	Proposed loss
AUC@5° ↑	0.38	0.41	**0.47**
AUC@10° ↑	0.43	0.45	**0.50**
AUC@20° ↑	0.48	0.49	**0.54**

Table 4. AUC scores of essential matrix estimation using different loss functions for training ARS-MAGSAC on the RootSIFT features of the PhotoTourism benchmark.

5.2. Ablation Studies

In the left plot of Fig. 4, we show the accuracy gained from each component of the algorithm. We show the AUC scores and their std. at 5°, 10° and 20° averaged over 12 scenes. The proposed affine loss plays an important role in improved accuracy. Also, it confirms that the widely used techniques, *e.g.*, SNN filtering, RootSIFT, initial training, are important steps to achieve state-of-the-art results.

In the right plot of Fig. 4, we show the results of different samplers used within ARS-MAGSAC. The compared samplers are the uniform one from [19], the NG-RANSAC sampler [15], PROSAC [16], and the proposed AR-Sampler. It can be seen that the proposed AR-Sampler leads to the best accuracy. Interestingly, PROSAC significantly outperforms NG-RANSAC, which is just marginally more accurate than the uniform sampler when used inside ARS-MAGSAC.

The average AUC scores of essential matrix estimation when using the pose error as loss (Pose loss), outlier ratio as loss (Self-supervised loss), and the proposed one (when training ARS-MAGSAC) on the PhotoTourism dataset are shown in Tab. 4. The proposed loss combining pose and affine losses leads to the most accurate results.

Figure 4. AUC scores at 5°, 10° and 20° of **E** estimation on the PhotoTourism dataset [48]. *Left*: the impact of each component. *Right*: AUC scores testing with the Uniform [19], NG-RANSAC sampler [15], PROSAC [16] and the proposed AR-Sampler.

margin – in the AUC@5° case, it is better than CLNet (*i.e.*, the second best) by 6 AUC points. This clearly shows that ARS-MAGSAC generalizes better than the state-of-the-art learning-based robust estimation approaches and is able to learn the underlying scene geometry better.

PhotoTourism [48], SuperPoint [17]. To demonstrate that any features can be made rotation and scale covariant by a post-processing step, we extracted SuperPoint features from the PhotoTourism dataset and used mutual NN matching to get correspondences. We estimated the feature orientations and scales by the recent Self-Scale-Ori method [59]. We then trained two models, one with the proposed affine loss on the extracted orientations and scales, and one with the pose loss. The **E** estimation results of these models and MAGSAC++ averaged over the 12 scenes of PhotoTourism are shown in Tab. 3. ARS-MAGSAC trained with affine loss is superior to training with only pose, showing that the proposed loss works with the state-of-the-art features as well.

Additionally, Tab. 5 reports the AUC scores and the run-time of essential matrix estimation using our proposed method, or combined with other networks, *e.g.*, the model proposed by CLNet [62], replacing residual blocks with densely connected blocks among different layers, *etc*. The proposed ARS-MAGSAC shows the best accuracies and marginally worse run-time than the best. Our architecture generalizes better than other networks, as simple as possible. The dense connections learn from the given correspon-
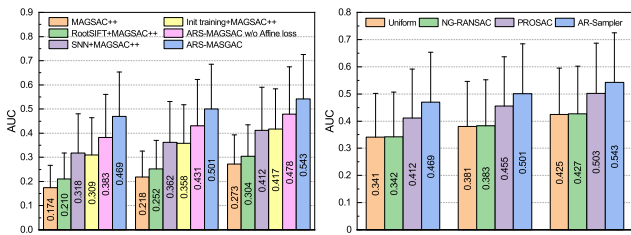
dences as well as CLNet, but with low efficiency.

Threshold	ARS-MAGSAC + GNN layer [27]	ARS-MAGSAC + CLNet model [62]	ARS-MAGSAC + Dense blocks [23]	ARS-MAGSAC
AUC@5°	0.37	0.41	0.41	**0.47**
AUC@10°	0.41	0.45	0.45	**0.50**
AUC@20°	0.46	0.50	0.50	**0.54**
Run-time (ms)	**22.4**	84.4	127.8	31.4

Table 5. AUC scores of essential matrix estimation using ARS-MAGSAC with different networks on PhotoTourism.

5.3. Fundamental Matrix Estimation

We test ARS-MAGSAC for **F** estimation on the KITTI benchmark [20]. As in [60, 15], Sequences "00-05" and "06-10" are regarded as the training and testing sets, respectively. The KITTI dataset consists of consecutive frames of high-resolution cameras rigidly mounted to a moving vehicle in a mid-size city, rural areas and highways [20]. The images are of size 1226×370. Correspondences are detected between subsequent images. In total, we use $14\,130$ image pairs for training, and another 9060 for testing.

Method	LMEDS [44]	RSC [19]	GC-RSC [8]	MSC [11]	MSC++ [10]	EAS [18]	OANet [61]	CLNet [62]	NG-RSC [15]	ARS-MSC
F1 score (%)↑	38.55	56.83	66.90	57.80	60.65	55.16	64.10	64.47	69.50	**69.93**
AUC@10° ↑	0.45	0.78	0.91	0.76	0.83	0.83	0.94	0.95	0.92	**0.97**
Error (px) ↓	3.15	0.98	0.42	0.84	0.75	0.88	0.57	0.54	0.41	**0.29**
Run-time (ms) ↓	20	32	56	233	413	310	17	13	18	12

Table 6. The F1 score, AUC score thresholded at $10°$, and median symmetric epipolar error (in pixels) of fundamental matrix estimation on 9690 images pairs from the KITTI benchmark [20].

Fundamental matrix estimation runs on the same architecture as what we described in Sec. 5.1. In this case, neither the point coordinates nor the orientations and scales are normalized. In contrast to **E** estimation, we do not apply initial training as it does not improve the accuracy here.

KITTI [20], RootSIFT [2]. Tab. 6 reports the average run-time in milliseconds, the median symmetric epipolar error in pixels, the AUC and F1 scores of the estimated **F** matrices. ARS-MAGSAC leads to the highest accuracy in all metrics. Interestingly, while the F1 score is only marginally higher than that of NG-RANSAC, the AUC score is better by 5%. This implies that the F1 score is not in perfect agreement with the actual camera pose error captured in the AUC score. The run-time of ARS-MAGSAC is the lowest, being 33% faster than NG-RANSAC. These timings exclude the prediction time which is at most $1 - 2$ milliseconds. Moreover, Fig. 3 shows the CDFs of the epipolar errors (middle) and the F1 scores (right) on the 9060 image pairs. ARS-MAGSAC leads to the lowest errors and highest F1 scores.

PhotoTourism [48], RootSIFT [2]. We compare **F** matrix estimation on the RootSIFT features of the same scenes as we test for **E**. The F1 scores and the run-time are shown in Table 7. The proposed ARS-MAGSAC leads to the best results on all but one scene, where it is the second best. Comparable rum-time is achieved among the lowest ones.

Method	LMEDS [44]	RSC [19]	GC-RSC [8]	MSC [11]	MSC++ [10]	EAS [18]	OANet [61]	CLNet [62]	NG-RSC [15]	ARS-MSC
AUC@10° ↑	35.00	40.10	43.41	42.68	42.46	35.30	36.91	40.67	43.66	**47.76**
Run-time (ms) ↓	**21.00**	30.67	73.25	281.30	318.13	325.83	**21.00**	34.83	25.85	31.35

Table 7. Fundamental matrix estimation on the RootSIFT features of the PhotoTourism dataset [48]. Average run-times (ms) in the *first* row, F1 scores on each scene and the average in the *end*. For RANSAC, GC-RANSAC, MAGSAC and MAGSAC++, we use the threshold as in [12]. We trained OANet, CLNet, NG-RANSAC on the same datasets as we use to train ARS-MAGSAC.

PhotoTourism [48], SuperPoint [17]. In addition, we train and test ARS-MAGSAC for **F** matrix estimation on learning-based features on the PhotoTourism dataset. The coordinates are detected by SuperPoint [17] and matched by the mutual nearest neighbor matcher. The feature orientations and scales are obtained by the state-of-the-art Self-Scale-Ori method [59]. We trained on these features both with the pose error and our proposed affine loss that learns from the scales and orientations. Tab. 8 demonstrates that the proposed method works accurately with SuperPoint features as well. The best performance is achieved with the proposed affine loss.

Method	Loss	F1 score (%) ↑	med. epi. error (px) ↓	run-time (ms) ↓
MAGSAC++ [10]	-	26.65	4.43	180.50
ARS-MAGSAC	Pose	29.45	4.30	23.92
ARS-MAGSAC	Affine	**29.72**	**4.13**	**22.83**

Table 8. Fundamental matrix estimation on the SuperPoint features of the PhotoTourism dataset [48]. We show the trained models with two different losses, compared with MAGSAC++.

Comparison to [14], RootSIFT. We compare the proposed ARS-MAGSAC with running LMEDS [44] as a post-processing step on MAGSAC++ [10] or GC-RANSAC [8] as proposed in [14] for **F** matrix estimation. We use the dataset from [14] and also PhotoTourism. The F1 scores are reported in Table 9. ARS-MAGSAC is the most accurate on all but one scene (*i.e.*, TUM [49]. On TUM, all methods lead to similar results and the differences are small.

Dataset	ARS-MSC	MSC++ [10]	+ LMEDS [44]	GC-RSC [8]	+ LMEDS
CPC [58]	**27.40**	25.18	25.32	24.56	25.19
KITTI [20]	**69.93**	60.65	69.45	66.90	69.50
T&T [28]	**14.04**	13.58	13.61	12.58	12.82
TUM [49]	9.17	9.20	9.20	**9.24**	9.21
PhotoTourism	**47.76**	42.46	42.29	43.41	42.66

Table 9. F1 scores (in percentages) of **F** estimation on the dataset from [14] using the proposed ARS-MAGSAC, and MAGSAC++ or GC-RANSAC followed by LMEDS as proposed in [14].
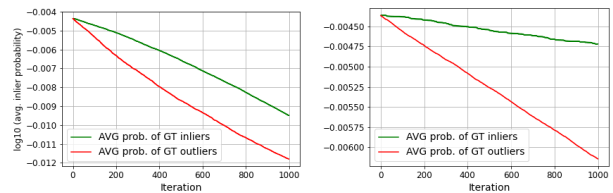

Figure 5. Avg. inlier probabilities of the GT inliers (green) and outliers (red) over iterations as updated by the proposed AR-Sampler. (*Left*) **E** matrix estimation. (*Right*) Absolute pose estimation.
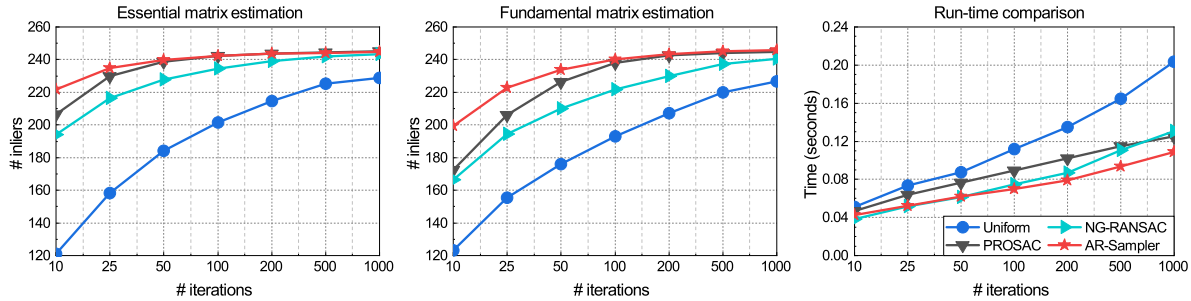
Figure 6. *Left*, *middle*: the number of inliers (vertical axis) of the best model found within a given number of iterations (horizontal axis) by MAGSAC++ [10] when combined with the Uniform [19] (blue), NG-RANSAC [15] (green), PROSAC [16] (black) and the proposed AR-Sampler (red). Average over 4950 image pairs from scene Sacre Coeur. *Right*: the run-time, in seconds, versus the iteration number. The SNN ratio [30] provides the inlier probabilities here, without using any learning algorithms.

5.4. Sampler Comparison

Decreasing Speed Comparison. To give a more nuanced understanding of the proposed sampler, we ran essential matrix estimation on a randomly selected image pair and recorded the updated inlier probabilities throughout the iterations. These probabilities are then averaged independently for the ground-truth (GT) inliers (green curve) and outliers (red) and plotted in the *left* plot of Fig. 5. Additionally, to test the proposed AR-Sampler on a completely different problem, we ran the P1AC [57] single-point solver, estimating the absolute pose of a single query image (*right* plot). In both cases, the probabilities of the outliers w.r.t. the ground truth reduce faster than that of the inliers, demonstrating that the proposed sampler works as intended.

Sampler Comparison with SNN Ratio. We test the proposed sampler on the 4950 image pairs from scene Sacre Coeur when using the second nearest neighbor (SNN) ratio to order the points according to the inlier probabilities. To our experiments, considering the SNN ratio directly as prior inlier probability does not lead to an improvement compared to PROSAC. However, exploiting the point ranks implied by the SNN ratio works well. Assume that we are given n points $\mathbf{p}_{i_1}, \ldots, \mathbf{p}_{i_n}$ ordered by their SNN ratios $s_{i_1}, \ldots, s_{i_n}$. Thus, $s_{i_1} \leq s_{i_2} \leq \cdots \leq s_{i_n}$. We calculate the prior probability of the i_jth point as $\mu_{i_j} = 1 - (j-1)/(n-1)$, $j \in [1, n]$. Consequently, the first point, has 11 as prior probability when ordered by the SNN ratio. Conversely, the last one is assigned zero.

The inlier numbers and run-times of the original MAGSAC++ when used together with the uniform [19], PROSAC [16], NG-RANSAC samplers [15], and the proposed one are shown in Fig. 6. The horizontal axis is the max. iteration number which is a strict upper bound on the iteration number that is controlled by the RANSAC confidence parameter. The curve of the proposed sampler starts from a higher inlier number, both for **E** and **F** estimation,

than that of the others, *i.e*., it leads to finding good samples earlier than the other methods. As expected all methods converge to similar results after many iterations. Due to being extremely efficient, AR-Sampler leads to the fastest robust estimation, as shown in the right plot of Fig. 6.

Run-time of the Re-ordering Procedure. In the proposed adaptive re-ordering sampler, we use the priority queue implemented in the standard C++ library based on a heap structure to efficiently update the probabilities. The average run-time of the update is 32 microseconds in case of **E** matrix estimation ($m = 5$). For comparison, the PROSAC [16] update costs 97 microseconds on average.

6. Conclusion

We propose ARS-MAGSAC, a novel algorithm for robust relative pose estimation that achieves state-of-the-art accuracy while maintaining comparable or better processing times than its less accurate alternatives. ARS-MAGSAC runs in real-time on most of the tested problems, making it highly practical for computer vision applications. Additionally, we introduce a new loss that leverages additional geometric information, such as feature orientation and scale, improving the robustness in challenging environments. For many features, this extra information is available *for free*. For other ones, it can be easily extracted by, *e.g*., the Self-Scale-Ori method [29]. Furthermore, our proposed AR-Sampler outperforms traditional samplers, both when using predicted weights or SNN ratios as inlier probabilities. We also demonstrate that ARS-MAGSAC generalizes better than state-of-the-art learning-based approaches.

References

[1] Ransac tutorial 2020 data. https://github.com/ducha-aiki/ransac-tutorial-2020-data. 5

[2] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 5, 6, 8

[3] Daniel Barath. Five-point fundamental matrix estimation for uncalibrated cameras. In *CVPR*, 2018. 4

[4] Daniel Barath, Luca Cavalli, and Marc Pollefeys. Learning to find good models in ransac. In *CVPR*, 2022. 1

[5] D. Barath, T-J. Chin, Chum Ondřej., D. Mishkin, R. Ranftl, and J. Matas. RANSAC in 2020 tutorial. In *CVPR*, 2020. 2

[6] Daniel Barath and Levente Hajder. Efficient recovery of essential matrix from two affine correspondences. *IEEE Transactions on Image Processing*, 2018. 4

[7] Daniel Barath and Zuzana Kukelova. Homography from two orientation-and scale-covariant features. In *ICCV*, 2019. 6

[8] Daniel Barath and Jiří Matas. Graph-cut RANSAC. In *CVPR*, 2018. 5, 6, 8

[9] Daniel Barath and Jirı Matas. Multi-class model fitting by energy minimization and mode-seeking. In *ECCV*, 2018. 1

[10] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jirı Matas. Magsac++, a fast, reliable and accurate robust estimator. In *CVPR*, 2020. 1, 2, 5, 6, 7, 8, 9

[11] Daniel Barath, Jana Noskova, and Jiří Matas. MAGSAC: marginalizing sample consensus. In *CVPR*, 2019. 1, 5, 6, 8

[12] Daniel Barath, Jana Noskova, and Jirı Matas. Marginalizing sample consensus. *TPAMI*, 2021. 6, 7, 8

[13] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006. 3

[14] Jia-Wang Bian, Yu-Huan Wu, Ji Zhao, Yun Liu, Le Zhang, Ming-Ming Cheng, and Ian Reid. An evaluation of feature matchers for fundamental matrix estimation. In *BMVC*, 2019. 8

[15] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *ICCV*, 2019. 1, 2, 4, 5, 6, 7, 8, 9

[16] Ondřej Chum and Jiří Matas. Matching with PROSAC-progressive sample consensus. In *CVPR*, 2005. 1, 7, 9

[17] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR*, 2018. 2, 3, 7, 8

[18] Aoxiang Fan, Jiayi Ma, Xingyu Jiang, and Haibin Ling. Efficient deterministic search with robust loss functions for geometric model fitting. *TPAMI*, 2021. 5, 6, 8

[19] 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*, 1981. 1, 5, 6, 7, 8, 9

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 7, 8

[21] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 6

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 6

[23] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 8

[24] Hossam Isack and Yuri Boykov. Energy-based geometric multi-model fitting. *IJCV*, 2012. 1

[25] Maksym Ivashechkin, Daniel Barath, and Jiří Matas. VSAC: Efficient and accurate estimator for H and F. In *ICCV*, 2021. 1

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[27] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 8

[28] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 2017. 8

[29] Jongmin Lee, Byungjin Kim, and Minsu Cho. Self-supervised equivariant learning for oriented keypoint detection. In *CVPR*, 2022. 3, 9

[30] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 2, 3, 5, 9

[31] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2

[32] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image matching from handcrafted to deep features: A survey. *IJCV*, 2021. 1

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

[34] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jirı Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *IJCV*, 2005. 3

[35] Dmytro Mishkin, Jirı Matas, and Michal Perdoch. MODS: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 2015. 1, 3

[36] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 2015. 1

[37] Kai Ni, Hailin Jin, and Frank Dellaert. GroupSAC: Efficient consensus in the presence of groupings. In *ICCV*. IEEE, 2009. 1

[38] Trung Thanh Pham, Tat-Jun Chin, Konrad Schindler, and David Suter. Interacting geometric priors for robust multimodel fitting. *IEEE Transactions on Image Processing*, 2014. 1

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

[40] Rahul Raguram, Ondřej Chum, Marc Pollefeys, Jirı Matas, and Jan-Michael Frahm. USAC: a universal framework for random sample consensus. *TPAMI*, 2013. 1

[41] Rene Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *ECCV*, 2018. 2

[42] Carolina Raposo and Joao P Barreto. Theory and practice of structure-from-motion using affine correspondences. In *CVPR*, 2016. 4

[43] Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 1984. 5

[44] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*. 2005. 6, 8

[45] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2011. 3

[46] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1

[47] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1

[48] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3D. In *SIGGRAPH*. 2006. 5, 6, 7, 8

[49] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012. 8

[50] Weiwei Sun, Wei Jiang, Andrea Tagliasacchi, Eduard Trulls, and Kwang Moo Yi. Attentive context normalization for robust permutation-equivariant learning. In *CVPR*, 2020. 2

[51] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 2016. 6

[52] Philip Hilaire Torr, Slawomir J Nasuto, and John Mark Bishop. NAPSAC: High noise, high dimensional robust estimation-it's in the bag. In *BMVC*, 2002. 1

[53] P. H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *IJCV*, 2002. 1

[54] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding (CVIU)*, 2000. 1

[55] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 6

[56] Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 2014. 5

[57] Jonathan Ventura, Zuzana Kukelova, Torsten Sattler, and Daniel Barath. P1AC: Revisiting absolute pose from a single affine correspondence. In *ICCV*, 2023. 9

[58] Kyle Wilson and Noah Snavely. Robust global translations with 1DSfM. In *ECCV*, 2014. 8

[59] Sanghyun Woo, Dahun KIm, Joon-Young Lee, and In So Kweon. Global context and geometric priors for effective non-local self-attention. In *BMVC*, 2021. 7, 8

[60] Kwang Moo Yi*, Eduard Trulls*, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *CVPR*, 2018. 1, 6, 8

[61] 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. 2, 5, 6, 8

[62] Chen Zhao, Yixiao Ge, Feng Zhu, Rui Zhao, Hongsheng Li, and Mathieu Salzmann. Progressive correspondence pruning by consensus learning. In *ICCV*, 2021. 2, 5, 6, 7, 8