Marginalizing Sample Consensus

Daniel Barath, Jana Noskova, and Jiri Matas

Abstract—A new method for robust estimation, MAGSAC++, is proposed. It introduces a new model quality (scoring) function that does not make inlier-outlier decisions, and a novel marginalization procedure formulated as an M-estimation with a novel class of M-estimators (a robust kernel) solved by an iteratively re-weighted least squares procedure. Instead of the inlier-outlier threshold, it requires only its loose upper bound which can be chosen from a significantly wider range. Also, we propose a new termination criterion and a technique for selecting a set of inliers in a data-driven manner as a post-processing step after the robust estimation finishes. On a number of publicly available real-world datasets for homography, fundamental matrix fitting and relative pose, MAGSAC++ produces results superior to the state-of-the-art robust methods. It is more geometrically accurate, fails fewer times, and it is often faster. It is shown that MAGSAC++ is significantly less sensitive to the setting of the threshold upper bound than the other state-of-the-art algorithms to the inlier-outlier threshold. Therefore, it is easier to be applied to unseen problems and scenes without acquiring information by hand about the setting of the inlier-outlier threshold. The source code and examples both in C++ and Python are available at https://github.com/danini/magsac.

Index Terms—Robust model estimation, RANSAC, noise scale, M-estimator, marginalization

1 Introduction

The RANdom SAmple Consensus (RANSAC) algorithm proposed by Fischler and Bolles [1] 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 [2], [3], wide baseline matching [4], [5], [6], motion segmentation [2], detection of geometric primitives [7], pose-graph initialization for structure-from-motion pipelines [8], [9], image mosaicing [10], and to perform [11] or initialize multi-model fitting algorithms [12], [13]. In brief, RANSAC repeatedly selects random subsets of the input data points, typically minimal, and fits a model, e.g., a 2D line to two points, a fundamental matrix to seven 2D point correspondences, or a 6D pose to three 2D-3D correspondences. The quality of the model is then measured, for instance, 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.

We propose a new robust loss, a randomized RANSAC-like robust estimator (MAGSAC++) and a termination criterion which eliminate the need for a hand-picked inlier-outlier threshold by marginalizing over a range of noise scales when determining the model quality and the inlier probabilities of data points.

Since the introduction of RANSAC, a number of modifications have been proposed replacing the components of the original algorithm. For instance, improving the sampler impacts the speed of the robust estimation procedure via selecting a good sample early and, thus, triggering the termination criterion. The NAPSAC [17] sampler assumes that inliers are spatially coherent and, therefore, it draws samples from a hyper-sphere centered at the first, randomly selected, location-defining point. If this point is an inlier, the points sampled in its proximity are more likely to be inliers than the ones outside the ball. While NAPSAC exploits the observation that inliers tend to be "closer" to each other than outliers, the GroupSAC algorithm [18] assumes that inliers are often "similar" to each other and, therefore, data points can be separated into groups according to their similarities. PROSAC [19] 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. P-NAPSAC [20] merges the advantages of local and global sampling by drawing samples from progressively growing neighborhoods. Gradually, the algorithm changes from the fully localized NAPSAC to the global PROSAC sampling.

Regarding speeding up the robust estimation process, one way of avoiding unnecessary calculations is via termination of verification of models which are unlikely to be more accurate than the current so-far-the-best. There has been a number of preemptive model verification strategies proposed. For example, when using the $T_{d,d}$ test [21], the model verification is first performed on d randomly selected points (where $d \ll n$). The remaining $n - d$ ones are evaluated only if the first d points are all inliers to the verified model. The test was extended by the so-called bail-out test [22]. Given a model to be scored, a randomly selected subset of d points is evaluated. If the inlier ratio within this

- *Daniel Barath is with the Visual Recognition Group, Department of Cybernetics, Czech Technical University, Prague and the Machine Perception Research Laboratory, SZTAKI, Budapest and also with the Computer Vision and Geometry Group, Department of Computer Science, ETH Zürich, 8092 Zürich, Switzerland. E-mail: dbarath@inf.ethz.ch.*
- *Jana Noskova and Jiri Matas are with the Visual Recognition Group, Department of Cybernetics, Czech Technical University, 166 36 Prague, Czechia. E-mail: Jana.Noskova@cvut.cz, matas@cmp.felk.cvut.cz.*

Manuscript received 15 Jan. 2021; revised 19 July 2021; accepted 25 July 2021. Date of publication 10 Aug. 2021; date of current version 3 Oct. 2022. (Corresponding author: Daniel Barath.)
Recommended for acceptance by P. Mordohai.
Digital Object Identifier no. 10.1109/TPAMI.2021.3103562

subset is significantly smaller than the current best inlier ratio, it is unlikely that the model will yield a larger consensus set than the current maximum and, thus, is discarded. In [23], [24], an optimal randomized model verification strategy was described. The test is based on Wald's theory of sequential testing [25]. Wald's SPRT test is a solution of a constrained optimization problem, where the user supplies acceptable probabilities for errors of the first type (rejecting a good model) and the second type (accepting a bad model) and the resulting optimal test is a trade-off between the time to decision and the errors committed.

Observing that RANSAC requires in practice more samples than what theory predicts, Chum *et al.* [26] identified a problem that not all all-inlier samples are "good", i.e., lead to a model accurate enough to distinguish all inliers, e.g., due to poor conditioning of the selected random all-inlier sample. They address the problem by introducing the locally optimized RANSAC (LO-RANSAC) that augments the original approach with a local optimization step applied to the *so-far-the-best* models. Lebeda *et al.* [14] showed that, for models with many inliers, the local optimization becomes a computational bottleneck due to the iterated least-squares model fitting where the processing time is a function of the number of used points. In [14], it is proposed to consider only a subset of the inliers in the local optimization. Only the final model polishing process is applied to the whole inlier set.

To improve the accuracy by better modelling the noise in the data, different model quality calculation techniques have been investigated. For instance, MLESAC [27] estimates the model quality by a maximum likelihood procedure with all its beneficial properties, albeit under certain assumptions about data point distributions. In practice, MLESAC results are often superior to the inlier counting of plain RANSAC, and they are less sensitive to the manually set inlier-outlier threshold. In MAPSAC [28], the robust estimation is formulated as a process that estimates both the parameters of the data distribution and the quality of the model in terms of maximum a posteriori.

All of the above-mentioned scoring strategies require a manually selected inlier-outlier threshold. Selecting a suitable threshold requires the user to acquire knowledge about the problem and the actual scene, restricting the out-of-the-box applicability of such algorithms. While there are commonly used threshold values for a number of problems, e.g., 2-3 pixels for homography estimation, they rarely lead to highly accurate solutions. Addressing this issue, the dependency on the user-defined inlier-outlier threshold is reduced by its adaptive selection during the model parameter estimation. The MINPRAN [29] algorithm, proposed in 1995, assumes that the outliers are distributed uniformly in the image. For each tested model, MINPRAN tests a number of candidate thresholds and chooses the one with inliers the least likely to have occurred randomly. Moisan *et al.* [30] proposed a contrario RANSAC, AC-RANSAC in short, which follows an approach similar to MINPRAN, but the minimized probability models the consistency of data points with an unknown rigid model. In [31], the best threshold is selected using the Likelihood Ratio Test. While MINPRAN and AC-RANSAC are shown to achieve accurate results, they obtain their solutions using a single adaptively selected threshold. This approach can fail when the background model does not follow the assumed distribution, e.g., the outliers are structured, and it ignores the additional information that other candidate thresholds provide. Also, testing multiple thresholds for each minimal sample model often leads to a deterioration in the processing time. The RECON [32] algorithm assumes that the noisy observations of the sought model have a large amount of common inliers with similar point-to-model residuals. Finding multiple models with similar inlier sets is interpreted as finding the sought model. The RANSAAC [33] algorithm follows a different strategy to eliminate the threshold from the model fitting procedure. RANSAAC estimates models from randomly selected minimal samples similarly as RANSAC. It then converts the models to sets of 2D points, and combines multiple models by averaging the point coordinates used for representing them. Finally, the model is fitted to the averaged point coordinates. Besides the number of drawbacks of RANSAAC, e.g., non-robust model-to-points conversion, it is shown by the authors that it only works inside a local optimization process after a reasonably good model is found. Thus, the inlier-outlier threshold is still required.

As the *main contribution* of this paper, we propose an approach, σ-consensus++, that eliminates the need for a precise user-defined noise scale σ when estimating the model parameters in a robust manner. Instead of σ, only a loose upper bound $\sigma_{\max}$ is required defining the range of possible threshold values. The σ-consensus++ algorithm is in fact a new M-estimator (a robust kernel), solved by an iteratively re-weighted least squares procedure. This M-estimator marginalizes over the range of noise scales. As *minor contributions*, we propose a new termination criterion which does not require a σ value. Considering the fact that some applications, e.g., structure-from-motion [34], need to know inliers, we propose a way to adaptively determine the set of inliers after the robust estimation finishes. The inliers are selected by thresholding, such that the model to which they lead after least-squares fitting is similar to the model determined by the robust estimation procedure applied without inlier-outlier decisions done.

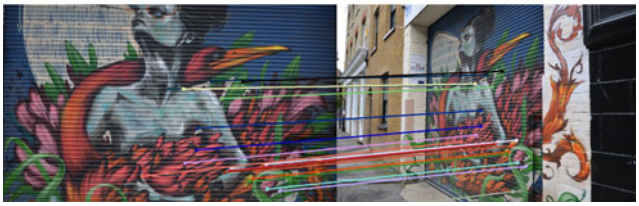
Preliminary versions of MAGSAC++ with σ-consensus++ were published at CVPR 2019 [35] and CVPR 2020 [20]. This paper extends and improves them by (i) combining their "bells and whistles", (ii) proposing a termination criterion applicable for MAGSAC++, (iii) proposing an inlier selection technique after the robust process is applied, (iv) and providing a number of new experiments on homography, fundamental matrix and relative pose estimation. Example results are shown in Fig. 1.

2 Notation and Preliminaries

In this paper, the set of input data points is denoted $\mathcal{P} = \{p \mid p \in \mathbb{R}^{\nu}, \nu \in \mathbb{N}_{>0}\}$, where ν is the dimension, e.g., $\nu = 2$ for 2D points and $\nu = 4$ for point correspondences. The inlier set is $\mathcal{I} \subseteq \mathcal{P}$. The model to fit is represented by its parameter vector $\theta \in \boldsymbol{\Theta}$, where $\boldsymbol{\Theta} = \{\theta \mid \theta \in \mathbb{R}^{d}, d \in \mathbb{N}_{>0}\}$ is the manifold, e.g., of all possible 2D lines, and $d = 2$ is the dimension of the model (angle and offset). Fitting function $F : \mathcal{D} \rightarrow \boldsymbol{\Theta}$, where $\mathcal{D} \subset \mathcal{P}^{*}$ and $|\mathcal{D}| \geq m$, calculates the model

(a) Homography; `ExtremeView` dataset [14]

(b) Epipolar geometry; `IMW2020` dataset (`St. Paul's Cathedral`) [15]



(c) Homography; `HPatches` dataset [16]

Fig. 1. Example image pairs from the datasets used for testing the robust estimators. The inliers of MAGSAC++, selected adaptively by the proposed procedure, are visualized.

parameters from $n \geq m$ data points, where $\mathcal{P}^* = \exp\mathcal{P}$ is the power set of $\mathcal{P}$ and $m \in \mathbb{N}_{>0}$ is the minimum point number for fitting a model, e.g., $m = 2$ for lines. Note that F is a combined function applying different estimators based on the input point set. For instance, for $\mathcal{P}' \in \mathcal{P}^*$

$$F(\mathcal{P}') = \begin{cases} \text{MinimalSolver}(\mathcal{P}') & \text{if } |\mathcal{P}'| = m, \\ \text{LSQ}(\mathcal{P}') & \text{otherwise.} \end{cases} \tag{1}$$

Function $R : \Theta \times \mathcal{P} \rightarrow \mathbb{R}^+$ calculates the point-to-model residual. Function $I : \Theta \times \mathbb{R}^+ \times \mathcal{P}^* \rightarrow \mathcal{P}^*$ selects the set of inliers given model θ and noise standard deviation σ. We assume that the inlier-outlier threshold is calculated from the noise σ as $\tau(\sigma) = k\sigma$, where k is some constant. For instance, for the original RANSAC approach, $I_{\text{RANSAC}}(\theta, \sigma, \mathcal{P}) = \{p \in \mathcal{P} \mid R(\theta, p) < \tau(\sigma)\}$ and $\tau(\sigma) = \sigma$. The model quality function, measuring how much the actual model interprets the scene, is $Q : \Theta \times \mathbb{R}^+ \times \mathcal{P}^* \rightarrow \mathbb{R}^+$. Higher quality is interpreted as better model. Let $\{R(\theta, p_i)\}_{i=1}^n$ be the point-to-model residuals, ordered increasingly, such that $0 \leq R(\theta, p_1) < R(\theta, p_2) < \cdots < R(\theta, p_n)$. For RANSAC, $Q_{\text{RANSAC}}(\theta, \sigma, \mathcal{P}) = |I(\theta, \sigma, \mathcal{P})|$ and for MSAC, it is

$$Q_{\text{MSAC}}(\theta, \sigma, \mathcal{P}) = |I(\theta, \sigma, \mathcal{P})| - \frac{1}{\tau(\sigma)^2} \sum_{i=1}^{|I(\theta,\sigma,\mathcal{P})|} R^2(\theta, p_i).$$

3 MAGSAC

First, we describe the idea and design choices of the original MAGSAC [35] approach in brief. We will also discuss its merits and drawbacks.

Symbols used in this paper	
$\mathcal{P} = \{p \mid p \in \mathbb{R}^\nu, \nu \in \mathbb{N}_{>0}\}$	- Set of data points
$\mathcal{P}^*$	- Power set of $\mathcal{P}$
$\sigma \in \mathbb{R}^+$	- Noise standard deviation
$\sigma_{\max} \in \mathbb{R}^+$	- Noise std. upper bound
$\tau(\sigma)$	- Inlier-outlier threshold
$\Theta = \{\theta \mid \theta \in \mathbb{R}^d, d \in \mathbb{N}_{>0}\}$	- Model manifold
$R : \Theta \times \mathcal{P} \rightarrow \mathbb{R}^+$	- Point-to-model residual
$F : \mathcal{P}^* \rightarrow \Theta$	- Model estimator function
$I : \Theta \times \mathbb{R}^+ \times \mathcal{P}^* \rightarrow \mathcal{P}^*$	- Inlier selector function
$Q : \Theta \times \mathbb{R}^+ \times \mathcal{P}^* \rightarrow \mathbb{R}^+$	- Model quality function

3.1 Marginalizing Sample Consensus

Idea. In the original marginalizing sample consensus (MAGSAC) algorithm [35], the model quality is defined by marginalizing over the noise scale σ as follows:

$$Q^*(\theta, \mathcal{P}) = \int_0^{+\infty} Q(\theta, \sigma, \mathcal{P}) f(\sigma) \mathrm{d}\sigma,$$

where the noise σ is a random variable with density function $f(\sigma)$, $Q : \Theta \times \mathbb{R}^+ \times \mathcal{P}^* \rightarrow \mathbb{R}^+$ is a quality function, e.g., the inlier counting of RANSAC, which depends on an input model $\theta \in \Theta$, the inlier-outlier threshold $\tau(\sigma)$, and the set $\mathcal{P}$ of n data points.

Having no prior information, σ is assumed to be uniformly distributed within range $(0, \sigma_{\max})$, where $\sigma_{\max}$ is an upper bound for the noise scale ($\sigma_{\max} > 0$). Considering this assumption, the quality calculation becomes

$$Q^*(\theta, \mathcal{P}) = \frac{1}{\sigma_{\max}} \int_0^{\sigma_{\max}} Q(\theta, \sigma, \mathcal{P}) \mathrm{d}\sigma. \tag{2}$$

For instance, using the inlier counting of plain RANSAC $Q_{\text{RANSAC}}(\theta, \sigma, \mathcal{P})$, where $\tau(\sigma) = \sigma$ is the inlier-outlier threshold, we get marginalized quality function

$$Q^*_{\text{RANSAC}}(\theta, \mathcal{P}) = |I(\theta, \sigma_{\max}, \mathcal{P})| - \frac{1}{\sigma_{\max}} \sum_{i=1}^{|I(\theta,\sigma_{\max},\mathcal{P})|} R(\theta, p_i).$$

Data Interpretation and Design Choices. In MAGSAC, the choice of the marginalized quality function Q is motivated by the assumption that the residuals are calculated as the square root of a sum of squared normally distributed variables. Typically, the residuals of the inliers are calculated as the euclidean-distance from model θ in some ν-dimensional space (e.g., the re-projection error). In the case of assuming the distances along each axis of this ν-dimensional space to be independent and normally distributed with the same variance σ^2, value $(\text{residuals})^2/\sigma^2$ has χ^2- distribution with ν degrees of freedom. For a given σ, the residuals of the inliers are described by the trimmed χ-distribution[1] with ν degrees of freedom multiplied by σ with density

$$g(r \mid \sigma) = 2C(\nu)\sigma^{-\nu}\exp(-r^2/2\sigma^2)r^{\nu-1},$$

for $r < \tau(\sigma)$ and $g(r \mid \sigma) = 0$ for $r \geq \tau(\sigma)$. The normalizing constant $C(\nu) = (2^{\nu/2}\Gamma(\nu/2)\alpha)^{-1}$ and, for $a > 0$

1. The square root of χ^2-distribution.

$$\Gamma(a) = \int_0^{+\infty} t^{a-1}\exp(-t)\mathrm{d}t,$$

is the gamma function, ν is the dimension of the euclidean space in which the residuals are calculated and $\tau(\sigma)$ is set to α-quantile (e.g., $\alpha = 0.99$) of the non-trimmed distribution.

Note: the idea of model quality marginalization is general and independent of the choice of the noise distribution, here χ^2.

Model Polishing. The last step of RANSAC-like algorithms is the re-fitting of the model to all inliers. However, due to MAGSAC not making a strict inlier-outlier decision, the standard model polishing step is not directly applicable. Therefore, the σ-consensus algorithm was proposed which, first, assigns an inlier weight to each point and, finally, applies weighted least-squares fitting.

Suppose an input point set $\mathcal{P}$ and model θ estimated from a minimal sample as in RANSAC. Let $\theta_\sigma = F(I(\theta, \sigma, \mathcal{P}))$ be the model estimated from the inlier set

$$I(\theta, \sigma, \mathcal{P}) = \{p \mid p \in \mathcal{P} \wedge R(\theta, p) \; < \; \tau(\sigma)\}, \tag{3}$$

selected using threshold $\tau(\sigma)$ around the input model θ. Scalar $\tau(\sigma)$ is the threshold which σ implies; function F estimates the model parameters from a set of data points; function I returns the set of data points for which the point-to-model residuals are smaller than $\tau(\sigma)$.

For each possible σ value, the likelihood of point $p \in \mathcal{P}$ being inlier is calculated as

$$\mathrm{P}(p \mid \theta_\sigma, \sigma) = 2C(\nu)\sigma^{-\nu}R^{\nu-1}(\theta_\sigma, p)\,\exp\left(\frac{-R^2(\theta_\sigma, p)}{2\sigma^2}\right),$$

if $R(\theta_\sigma, p) \leq \tau(\sigma)$, where $R(\theta_\sigma, p)$ is the point-to-model residual. If $R(\theta_\sigma, p) > \tau(\sigma)$, likelihood $\mathrm{P}(p \mid \theta_\sigma, \sigma)$ is 0. For each point p, likelihood $\mathrm{P}(p \mid \theta_\sigma, \sigma)$ is marginalized over σ and the obtained probability is used as an inlier weight in the final weighted least-squares fitting. The objective function $Q(\theta, \sigma, \mathcal{P})$ is the log-likelihood with inlier density $g(r \mid \sigma)$ and outliers assumed uniformly distributed.

Issues.There are two main issues with the MAGSAC approach, a practical and a theoretical one. In practice, the procedure of marginalizing $\mathrm{P}(p \mid \theta_\sigma, \sigma)$ over σ calculates $\mathrm{P}(p \mid \theta_\sigma, \sigma)$ a number of times with different σ values. Each calculation requires to select the set of inliers and obtain θ_σ by LS fitting on them. *This step is time consuming* even with the number of speedups proposed in the original paper [35]. The theoretical issue is that the objective function does not have its maximum at zero. Consequently, in the case of having perfect data, i.e., no noise, MAGSAC fails to return the sought model parameters. As a minor issue, both the quality function and the likelihood can only be calculated approximately for non piece-wise constant objective functions, e.g., χ^2-based or truncated L_2 loss. The exact calculation can only be done for the RANSAC-like inlier counting.

4 MAGSAC++

The MAGSAC++ algorithm is proposed here via reformulating the previously described MAGSAC problem as an iteratively re-weighted least-squares (IRLS) approach. To do so, a new model quality function and a procedure to polish the model parameters without making strict inlier-outlier decisions and doing a number of LS fittings are proposed.

The proposed MAGSAC++ is based on an iteratively reweighted least squares (IRLS) approach where the model parameters in the $(i+1)$th step are calculated as follows:

$$\theta_{i+1} = \arg\min_\theta \sum_{p \in \mathcal{P}} w(R(\theta_i, p))R^2(\theta, p), \tag{4}$$

where the weight of point p is

$$w(R(\theta_i, p)) = \int_0^{+\infty} \mathrm{P}(p \mid \theta_i, \sigma)f(\sigma)\mathrm{d}\sigma, \tag{5}$$

and $\theta_0 = \theta$, i.e., the initial model from the minimal sample.

Data Interpretation and Design Choices.Similarly as in MAGSAC, the inlier residuals are euclidean-distances of points assumed to be corrupted by Gaussian noise and, thus, have χ-distribution. The noise standard deviation σ is assumed to be uniformly distributed within $(0, \sigma_{\max})$. However, we make no assumptions about the outlier distributions. Note that the proposed quality and inlier weight functions can be modified straightforwardly when considering differently distributed inliers.

4.1 Inlier Weight Calculation

The weight function defined in (5) is the marginal density of the inlier residuals as follows:

$$w(r) = \int_0^{+\infty} g(r \mid \sigma)f(\sigma)\mathrm{d}\sigma. \tag{6}$$

Let $\tau(\sigma) = k\sigma$ be the chosen quantile of the χ-distribution. For residual $0 \leq r \leq k\sigma_{\max}$

$$w(r) = \frac{1}{\sigma_{\max}}\int_{r/k}^{\sigma_{\max}} g(r \mid \sigma)\mathrm{d}\sigma = \frac{1}{\sigma_{\max}}C(\nu)2^{\frac{\nu-1}{2}} \left(\Gamma\left(\frac{\nu-1}{2}, \frac{r^2}{2\sigma_{\max}^2}\right) - \Gamma\left(\frac{\nu-1}{2}, \frac{k^2}{2}\right)\right)$$

and, for $r > k\sigma_{\max}$, weight $w(r) = 0$. Function

$$\Gamma(a, x) = \int_x^{+\infty} t^{a-1}\exp(-t)\mathrm{d}t,$$

is the upper incomplete gamma function. Due to the design choices, weight $w(r)$ is positive and decreasing on interval $[0,\ \tau(\sigma_{\max})]$. Thus there is a ρ-function of an M-estimator which is minimized by IRLS using $w(r)$ and each iteration guarantees a non-increase in its loss function (chapter 9 of [36]). Consequently, it converges to a local minimum. If different noise distribution is assumed, this property does not necessarily hold. In those cases, a different algorithm should be used to solve the problem, e.g., Levenberg-Marquardt optimization [37].

IRLS (4) where $w(r)$ is defined by (6) with $\tau(\sigma) = 3.64\sigma$, where 3.64 is the 0.99 quantile of the χ-distribution with $\nu = 4$, will be called σ-consensus++ for problems using point correspondences. Parameter $\sigma_{\max}$ is the same user-defined maximum noise level parameter as in MAGSAC, usually, set to a fairly high value, e.g., 10 pixels for homography fitting. The σ-consensus++ algorithm is applied for fitting to a

non-minimal sample and, also, as a post-processing to improve the output of any robust estimator.

4.2 Model Quality Function

In order to select the model interpreting the data, a quality function has to be defined. Let

$$Q_{\mathrm{M++}}(\theta,\mathcal{P}) = n - \frac{1}{\rho(k\sigma_{\max})} L(\theta,\mathcal{P}) = |I(\theta,\sigma_{\max},\mathcal{P})| - \frac{1}{\rho(k\sigma_{\max})} \sum_{i=1}^{|I(\theta,\sigma_{\max},\mathcal{P})|} \rho(R(\theta,p_i)),$$

where

$$L(\theta,\mathcal{P}) = \sum_{p\in\mathcal{P}} \rho(R(\theta,p)), \tag{7}$$

is a loss function of the M-estimator defined by our weight function $w(r)$. Function

$$\rho(r) = \int_0^r xw(x)\mathrm{d}x = \int_0^{+\infty} \left(\int_0^r xg(x \mid \sigma)\mathrm{d}x \right) f(\sigma)\mathrm{d}\sigma,$$

for $r \in [0, +\infty)$. For any point p with residual r, the loss function is the mean of the residual values lower then r of a random variable with χ-distribution, i.e., the assumed distribution of the inlier residuals. Thus, the ρ-function is some type of a reasonable distance. It can be formulated in the same way for each σ and then marginalized over σ as in MAGSAC.

Due to assuming that the σ values are uniformly distributed within range $[0, \sigma_{\max}]$ for $0 \leq r \leq \tau(\sigma_{\max})$

$$\rho(r) = \frac{1}{\sigma_{\max}} \int_0^{\sigma_{\max}} \left[C(\nu) 2^{\frac{\nu+1}{2}} \sigma \gamma\left(\frac{\nu+1}{2}, \frac{r^2}{2\sigma^2} \right) - \frac{r^2}{2} g(k\sigma_{\max}|\sigma) \right] \mathrm{d}\sigma$$

and the integral can be removed as follows:

$$\rho(r) = \frac{1}{\sigma_{\max}} C(\nu) 2^{\frac{\nu+1}{2}} \left[\frac{\sigma_{\max}^2}{2} \gamma\left(\frac{\nu+1}{2}, \frac{r^2}{2\sigma_{\max}^2} \right) + \frac{r^2}{4} \left(\Gamma\left(\frac{\nu-1}{2}, \frac{r^2}{2\sigma_{\max}^2} \right) - \Gamma\left(\frac{\nu-1}{2}, \frac{k^2}{2} \right) \right) \right].$$

For $r > \tau(\sigma_{\max})$

$$\rho(r) = \rho(k\sigma_{\max}) = \sigma_{\max} C(\nu) 2^{\frac{\nu-1}{2}} \gamma\left(\frac{\nu+1}{2}, \frac{k^2}{2} \right),$$

where

$$\gamma(a,x) = \int_0^x t^{a-1}\exp(-t)\mathrm{d}t,$$

is the lower incomplete gamma function. Weight $w(r)$ can be calculated precisely or approximately as in MAGSAC. However, the precise calculation can be done very fast by *storing the values* of the complete and incomplete gamma functions *in a lookup table*. Then the weight and quality calculation becomes merely a few operations per point. MAGSAC++ algorithm uses (7) as quality function and σ-consensus++ for estimating the model parameters.

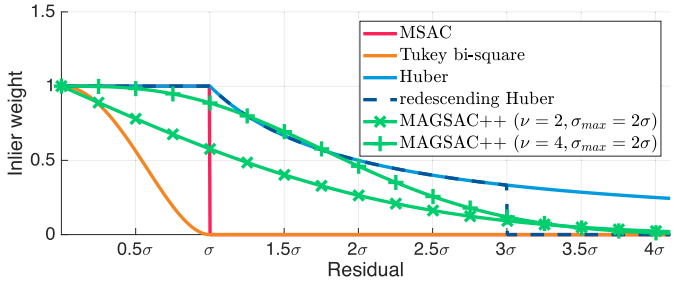


Fig. 2. Weighting functions for robust fitting. For MAGSAC++, we use $\sigma_{\max} = 2\sigma$ as an example and degrees-of-freedom $\nu = 2$ (e.g., 2D line fitting) and 4 (e.g., problems with point correspondences).

Function $w(r)$ is visualized in Fig. 2 together with other weightings which are often used for robust model fitting.

4.3 Termination Criterion

The number of inliers during the robust estimation is unknown due to not making strict inlier-outlier decisions. It is thus not possible to apply the standard termination criterion of RANSAC [38]

$$k(\theta,\sigma,\mathcal{P}) = \frac{\ln(1-\mu)}{\ln\left(1 - \left(\frac{|I(\theta,\sigma,\mathcal{P})|}{|\mathcal{P}|}\right)^m\right)}, \tag{8}$$

where k is the iteration number, μ is a manually set confidence in the results (typical values are 0.95 or 0.99), m is the size of the minimal sample needed for the estimation, and $|I(\theta,\sigma,\mathcal{P})|$ is the inlier number of the so-far-the-best model.

In order to determine k without using a particular value for σ, it is a straightforward choice to marginalize over the noise scale σ. Let us assume that the points are ordered by their residuals as $0 = \tau(\sigma_0) \leq R(\theta,p_1) = \tau(\sigma_1) \leq R(\theta,p_2) = \tau(\sigma_2) \leq \cdots \leq R(\theta,p_k) = \tau(\sigma_k) \leq \tau(\sigma_{\max}) < R(\theta,p_{k+1}) = \tau(\sigma_{k+1}) \leq \cdots \leq R(\theta,p_n) = \tau(\sigma_n)$. The iteration number is calculated as

$$k^*(\theta,\mathcal{P}) = \frac{1}{\sigma_{\max}} \int_0^{\sigma_{\max}} k(\theta,\sigma,\mathcal{P})d\sigma = \tag{9}$$

$$\frac{1}{\sigma_{\max}} \int_0^{\sigma_{\max}} \frac{\ln(1-\mu)}{\ln\left(1 - \left(\frac{|I(\theta,\sigma,\mathcal{P})|}{|\mathcal{P}|}\right)^m\right)} d\sigma. \tag{10}$$

Due to the fact that function $|I(\theta,\sigma,\mathcal{P})|$, measuring the number of inliers given a noise scale σ, is piece-wise constant, and that is the only part of (10) depending on σ, the integral can be replaced by a weighted summation. It is as follows:

$$k^*(\theta,\mathcal{P}) = \frac{1}{\sigma_{\max}} \sum_{i=1}^{k} \frac{(\sigma_i - \sigma_{i-1}) \ln(1-\mu)}{\ln\left(1 - \left(\frac{|I(\theta,\sigma_{i-1},\mathcal{P})|}{|\mathcal{P}|}\right)^m\right)}. \tag{11}$$

The function is, however, problematic when there are no points with zero residual. In that case, the denominator becomes $\ln(1) = 0$ and the iteration number ∞. We, thus, shift the inlier number by one and introduce a slight and artificial approximation as

$$k^*(\theta,\mathcal{P}) \approx \frac{1}{\sigma_{\max}} \sum_{i=1}^{k} \frac{(\sigma_i - \sigma_{i-1}) \ln(1-\mu)}{\ln\left(1 - \left(\frac{i}{|\mathcal{P}|}\right)^m\right)}. \tag{12}$$

Thus the number of iterations required for MAGSAC++ is calculated during the procedure and updated whenever a new so-far-the-best model is found, similarly as in RANSAC.

5 Inlier Selection

For some applications, the knowledge of what is inlier and outlier is a requirement. For instance, in structure-from-motion algorithms, the inlier correspondences are triangulated in 3D after the relative pose estimation and used for the reconstruction. Given the estimated model parameters θ after applying MAGSAC++, the objective is to find a reasonable set of inliers without introducing new parameters, e.g., a threshold. The *idea* is to return the set of points on which a least-squares fitting leads to a model which is similar to the one determined by the robust estimator. The problem is formalized as follows:

$$\mathcal{I}^* = \underset{\mathcal{I} \subseteq \mathcal{P}}{\arg\min} |F(\mathcal{I}) - \theta|, \tag{13}$$

where function F estimates the model parameters from a set of data points, and norm $|.|$ is some distance function defined over the model manifold. Note that this formulation allows to consider as inliers points with large point-to-model residuals. Besides, the problem introduced in (13) is NP-hard. Therefore, we weaken (13) by assuming that there exists a noise scale σ^* and, thus, an inlier-outlier threshold $\tau(\sigma^*)$ such that the points with residuals smaller than $\tau(\sigma^*)$ are the elements of $\mathcal{I}^*$. Consequently, it is enough to find σ^*. The problem becomes the following:

$$\sigma^* = \underset{\sigma \in \Sigma}{\arg\min} |F(I(\theta, \sigma, \mathcal{P})) - \theta|, \tag{14}$$

where $\Sigma = \{\sigma_i\}_{i=1}^{k} \subset [0, \sigma_{\max}]$ as introduced above (10). Note that it is straightforward to see that there are no other threshold values leading to different sets of inliers [29].

In the algorithm, we define the model-to-model distance as the sum of L^1 point-to-model residual distances as follows:

$$|\theta_1 - \theta_2| = \sum_{p \in \mathcal{P}} |R(\theta_1, p) - R(\theta_2, p)|. \tag{15}$$

Since the sought model should be of the same distance from both the inliers and outliers as the initial one, distance $|\theta_1 - \theta_2|$ can measured on all points without differentiating inliers and outliers. Since we measure the L^1 residual differences, outlier points with large residuals do not have higher impact on the model-to-model distance than inliers with small residuals. Also, distance $|\theta_1 - \theta_2|$ is enough to be measured only on a subset of points to speed up the procedure when needed. The pseudo-code of the algorithm is shown in Algorithm 3. Parameter $n_{\min}$ is the minimum number of points required to return, depending on the current application. If there is no requirement, $n_{\min} = m$, where m is the minimal sample size. Note that for models which are estimated from a larger-than-minimal sample by using SVD decomposition, e.g., fundamental/essential matrix, homography, using an incremental version of SVD, e.g., [39], speeds up the procedure significantly when a large number of points falls closer than $\sigma_{\max}$. Also, the procedure is straightforwardly parallelizable on multiple CPU cores.

Algorithm 1. The MAGSAC++ Algorithm

Input: $\mathcal{P}$ – data points; $\epsilon_{\max}$ – max. threshold
μ – confidence;
Output: θ^* – model parameters; $\mathcal{I}^*$ – inliers (*optional*)
$q^* \leftarrow 0$.
while $\neg$ `Terminate`(μ, q^*) **do** ▷ Section 4.3
  $S \leftarrow$ `Sample`$(\mathcal{P})$. ▷ *default*: P-NAPSAC sampler [20]
  if $\neg$ `TestSample`(S) **then** ▷ Degen. and cheirality tests
    `continue`
  $\theta \leftarrow$ `EstimateModel`(S)
  if $\neg$ `TestModel`(θ) **then** ▷ Degen. and cheirality tests
    `continue`
  $\theta' \leftarrow \sigma$-`consensus++`$(\mathcal{P}, \theta, \tau^{-1}(\epsilon_{\max}))$ ▷ Algorithm 2
  if $\neg$ `TestModel`(θ') **then** ▷ Degen. and cheirality tests
    `continue`
  $q \leftarrow$ `Scoring`$(\mathcal{P}, \theta', \tau^{-1}(\epsilon_{\max}))$ ▷ Eq. (7)
  if $q > q^*$ **then**
    $q^*, \theta^* \leftarrow q, \theta'$
$\mathcal{I}^* \leftarrow$ `SelectInliers`$(\theta^*, \mathcal{P})$ ▷ Section 5 (*optional*)

Algorithm 2. The σ-Consensus++ Algorithm

Input: $\mathcal{P}$ – data points; $\sigma_{\max}$ – max. noise scale
θ – initial model;
Output: θ^* - model parameters
$\theta_0, i \leftarrow \theta, 0$.
repeat
  $\{r_j\}_{j=1}^{|\mathcal{P}|} \leftarrow \{R(\theta_i, p) \mid p \in \mathcal{P}\}$
  $\{\widehat{r}_j\}_{j=1}^{|\mathcal{P}|} \leftarrow$ `Sort`$(\{r_j\}_{j=1}^{|\mathcal{P}|})$
  $\{w_j\}_{j=1}^{|\mathcal{P}|} \leftarrow \{w(\widehat{r}_j)\}_{j=1}^{|\mathcal{P}|}$ ▷ Eq. (6)
  $\theta_{i+1} \leftarrow$ `WLS`$(\mathcal{P}, \{w_j\}_{j=1}^{|\mathcal{P}|})$ ▷ Weighted least-squares
  if $\neg$ `TestModel`(θ_{i+1}) **then** ▷ Degen. and cheir. tests
    `break`
  $i \leftarrow i + 1$
until `Terminate`$(\theta_{i-1}, \theta_i, i)$
$\theta^* \leftarrow \theta_i$

6 Algorithmic Choices

To achieve state-of-the-art results, we combine the proposed MAGSAC++ with the components discussed in USAC [40]. We consider three popular vision problems, i.e., fundamental matrix, homography and relative pose (i.e., essential matrix) estimation. The included components for each problem are as follows: 1. *Sample degeneracy*. The degeneracy tests of minimal samples are for rejecting clearly bad samples to avoid the sometimes expensive model estimation. For homographies, samples consisting of collinear points are rejected. 2. *Sample cheirality*. The test is for rejecting samples based on the assumption that both of the cameras observing a 3D surface must be on its same side. For homography fitting, we check if the ordering of the four point correspondences – along their convex hulls – in both images are the same. If not, the sample is rejected. 3. *Model degeneracy*.

TABLE 1
The Average Processing Times (in Milliseconds) and Errors (in Pixels) in the Estimated Homographies (H), Fundamental (F) and Essential (E) Matrices Using Different Methods for Solving the Linear Systems in Their Solvers When Estimating the Model Parameters From a Minimal (m) or a Larger-Than-Minimal ($> m$) Sample. Each test is repeated 100 000 times. The size of the larger-than-minimal sample is selected uniformly randomly from range $[m+1, 1000]$. For error calculation, the re-projection was used for homographies, and Sampson-distance for fundamental and essential matrices. The tested methods solving linear systems are the ones implemented in the Eigen library.

	Average processing time (milliseconds)						Average error (pixels)					
	H		F		E		H		F		E	
	m	$> m$	m	$> m$	m	$> m$	m	$> m$	m	$> m$	m	$> m$
LLT	**0.002**	–	–	–	–	–	10^{-8}	–	–	–	–	–
LDLT	0.003	–	–	–	–	–	10^{-8}	–	–	–	–	–
PartialPivLU	0.003	–	–	–	–	–	$\mathbf{10^{-11}}$	–	–	–	–	–
FullPivLU	0.003	–	**0.011**	–	**0.060**	–	$\mathbf{10^{-11}}$	–	$\mathbf{10^{-12}}$	–	$\mathbf{10^{-14}}$	–
HouseholderQR	0.005	0.099	0.014	0.028	0.067	0.081	$\mathbf{10^{-11}}$	$\mathbf{10^{-7}}$	10^{-9}	$\mathbf{10^{-7}}$	10^{-12}	10^{-6}
ColPivHouseholderQR	0.006	**0.085**	0.015	0.027	0.069	0.077	10^{-10}	$\mathbf{10^{-7}}$	10^{-10}	$\mathbf{10^{-7}}$	10^{-13}	$\mathbf{10^{-8}}$
FullPivHouseholderQR	0.006	0.103	0.014	**0.026**	0.066	**0.075**	$\mathbf{10^{-11}}$	$\mathbf{10^{-7}}$	$\mathbf{10^{-12}}$	$\mathbf{10^{-7}}$	$\mathbf{10^{-14}}$	10^{-3}
JacobiSVD	0.023	22.356	0.028	0.039	0.079	0.088	10^{-6}	10^{-6}	10^{-4}	10^{-6}	10^{-13}	10^{-7}
BDCSVD	0.024	27.954	0.028	0.040	0.080	0.089	10^{-6}	10^{-6}	10^{-4}	10^{-6}	10^{-13}	10^{-7}

The purpose of this test is to reject models early to avoid verifying them unnecessarily. For fundamental matrices, DEGENSAC [41] is applied to determine if the epipolar geometry is affected by a dominant plane. For relative pose estimation, improper rotation matrices [42], i.e., the ones with negative determinant, are rejected. We observed that, for epipolar geometry estimation, symmetric epipolar distance tends to be more robust to degenerate models. In contrast, Sampson distance leads to higher accuracy – when using Sampson distance some degenerate models have lots of inliers. Therefore, we use Sampson distance as residual function when estimating fundamental and essential matrices and reject all models where the inlier number is significantly lower with symmetric epipolar distance. In practice, we found that a model can be rejected if it does not have at least half as many inliers with symmetric epipolar distance as with Sampson distance. 4. *Model cheirality*. The test is for rejecting models considering that the cameras must be on the same side of the observed surface. For fundamental and essential matrix estimation, we apply the oriented epipolar constraint [43]. 5. *Sampling*. We use the P-NAPSAC sampler [20]. It requires an a priori determined ordering of the input data points for its PROSAC [19] part. We used the scoring coming from the ratio-test [44]. The neighborhoods were determined by a multi-layer grid as proposed in [20] to minimize the computational overhead. 6. *Solvers*. One of the most time-sensitive parts of RANSAC-like robust estimation is the solver estimating the model parameter from a minimal or larger-than-minimal sample. It is time-sensitive since it runs *at least* once in every iteration. In many popular vision problems, e.g., homography estimation, the solution includes homogeneous or inhomogeneous linear systems. We thus tested the ways of solving such systems by the algorithms implemented in the Eigen library and chose the actual solvers in our MAGSAC++ implementation accordingly. Homographies are estimated by the standard normalized 4PT algorithm [38]. In the minimal case, the correspondences were not normalized since the system is not over-determined – the solution is exact. For fundamental matrices, the 7PT algorithm [38] runs to estimate from a minimal sample. In the over-determined case, we applied the normalized 8PT algorithm [45]. Essential matrices are estimated by the solver of Stewenius *et al.* [46]. When selecting the actual method applied to solve a linear system, our strategy was the following.

Table 1 reports the accuracy in pixels and processing time in milliseconds of methods solving the linear systems in the solvers for homography, fundamental and essential matrix estimation. Each test is repeated 100 000 times on randomly generated point correspondences. In each test, the size of the larger-than-minimal sample is selected uniformly randomly from range $[m+1, 1000]$, where m is the sample size.

In the minimal case, we chose the fastest methods from Table 1 since the accuracy is not crucial – the model is always improved later on more inliers. Also, this solver runs the most times. For fitting homographies to minimal samples, we solve the normal equations of the implied linear system via the Cholesky decomposition (LLT in the table). For estimating fundamental matrices, the null-space from the coefficient matrix is calculated by the LU decomposition with complete pivoting since that is one of the fastest solutions when we are given a 7×8 coefficent matrix (FullPivLU). For essential matrices, we chose the LU decomposition with complete pivoting (FullPivLU).

In the over-determined case, we selected the methods leading to the lowest errors. If there are multiple ones leading to the same error, the fastest one is applied. For fitting homographies, we apply the QR decomposition with column pivoting (ColPivHouseholderQR) – all tested types of QR decomposition lead to similarly low error, but column pivoting is the fastest. For estimating fundamental matrices, the null-space from the coefficient matrix is calculated by the QR decomposition with full pivoting (FullPivHouseholderQR). For essential matrices, we chose the QR decomposition with column pivoting (ColPivHouseholderQR).

The pseudo-code of MAGSAC++ and σ-consensus++ are shown in Algorithms 1 and 2, respectively. In the algorithm,

`TestSample` refers to the degeneracy and cheirality checks applied to minimal samples. Function `TestModel` is the degeneracy and cheirality checks applied to the estimated models.

Algorithm 3. Inlier Selection

Input: $\mathcal{P}$ – data points; θ – initial model
$n_{\min}$ – min. # of required points ▷ *default*: sample size
Output: $\mathcal{I}^*$ – inliers
1: $\{r_j\}_{j=1}^{k} \leftarrow \texttt{Sort}(\{R(\theta, p) \mid p \in \mathcal{P} \wedge R(\theta, p) \leq \tau(\sigma_{\max})\})$
2: $\epsilon^* \leftarrow \infty$.
3: **for** $i = n_{\min} \ldots k$ **do**
4: $\quad \theta' \leftarrow \texttt{LS}(\{r_j\}_{j=1}^{i})$ ▷ Least-squares fitting
5: $\quad \epsilon \leftarrow |\theta - \theta'|$ ▷ Eq. (15)
6: $\quad$ **if** $\epsilon < \epsilon^*$ **then**
7: $\quad\quad \epsilon^*, \mathcal{I}^* \leftarrow \epsilon, \{r_j\}_{j=1}^{i}$

7 Experiments

For testing the proposed methods, we used the problems and datasets from CVPR tutorial *RANSAC in 2020* [47]. The datasets and codes used are available at https://github.com/ducha-aiki/ransac-tutorial-2020-data. The hyper-parameters of all compared methods were tuned on the provided training set to maximize the accuracy. The reported errors were then calculated on the set which was not used when setting the hyper-parameters.

The error metric used is the mean Average Accuracy (mAA). This metric was originally introduced in [48], where it was called mean Average Precision (mAP). Later, Jin *et al.* [49] argued that "accuracy" is the correct terminology, due to simply evaluating how many of the predicted poses are accurate, as determined by thresholding the acceptance threshold, i.e., the threshold which decides if a particular result is accurate or not.

In order to determine which method is the least sensitive to the setting of either σ or $\sigma_{\max}$, we also measure the insensitivity to the inlier-outlier threshold (or upper limit in the case of MAGSAC, MAGSAC++ and AC-RANSAC). The methods were run multiple times using different threshold values from $t_1, \ldots, t_n$. For fundamental matrix and relative pose estimation, $t_{1..8} = (0.1, 0.25, 0.5, 1.0, 1.5, 3.0, 5.0, 10.0)$. For homography estimation, the following threshold values are used $t^{\mathrm{H}}_{1..12} = (0.1, 0.25, 0.5, 1.0, 1.5, 3.0, 5.0, 10.0, 25.0, 50.0, 75.0, 100.0)$. For each run, we calculated the mAA score of the results. The insensitivity of a method is measured as the weighted average of the mAA scores as follows:

$$\frac{\sum_{i=1}^{n}(t_i - t_{i-1})\mathrm{mAA}(t_i)}{\sum_{i=1}^{n}(t_i - t_{i-1})} = \frac{1}{t_n}\sum_{i=1}^{n}(t_i - t_{i-1})\mathrm{mAA}(t_i), \tag{16}$$

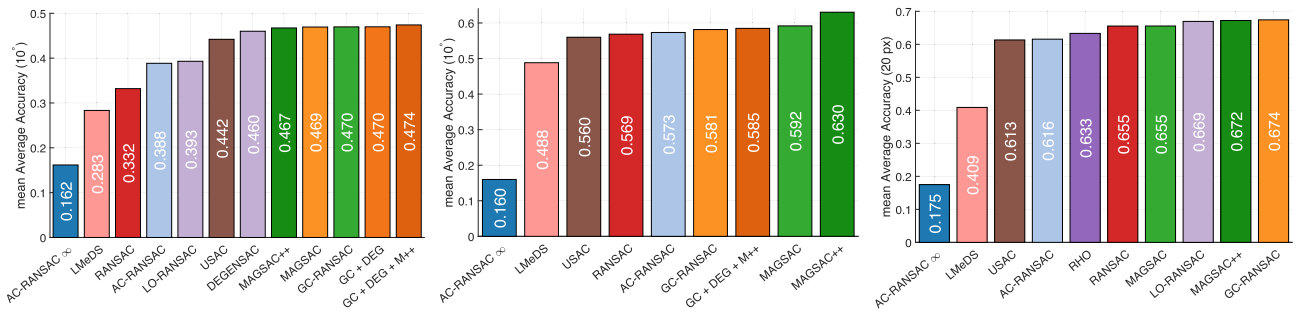
where $t_0 = 0$ and $\mathrm{mAA}(t_i)$ is the mAA score of a method after running it with threshold t_i. Formula (16) approximates the area under the mAA curve when plotted as the function of the inlier-outlier threshold used for the estimation.

In the rest of the paper, we call (16) the *insensitivity* measure. Note that measuring purely the insensitivity without including the accuracy of a method would require normalizing (16) by the maximum mAA value. We avoid this to make the insensitivity scores interpretable on their own. For example, (16) equals to 1 only if the method returns the perfect solution independently of the threshold.

7.1 Fundamental Matrix Estimation

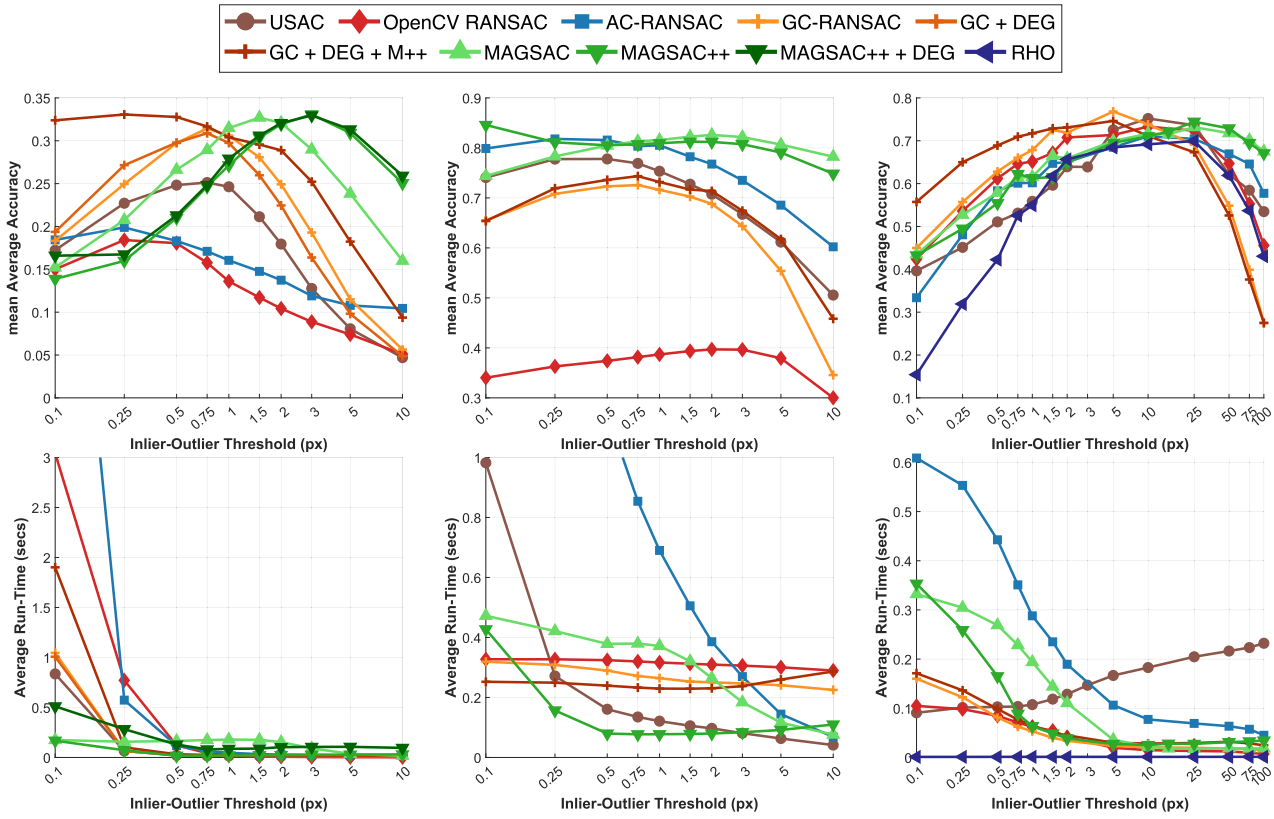
The methods compared for fundamental matrix estimation are OpenCV RANSAC [1], OpenCV LMedS [50], LO-RANSAC [26], LO-RANSAC + DEGENSAC [41], GC-RANSAC [51], GC-RANSAC + DEGENSAC, USAC [40], AC-RANSAC [30], MAGSAC, MAGSAC++, and GC-RANSAC with MAGSAC++ quality function and DEGENSAC. AC-RANSAC is a method setting the threshold adaptively. We tested two settings, i.e., with (AC-RANSAC) and without (AC-RANSAC ∞) an upper bound on threshold. The upper bound was tuned on the test set similarly as the parameters of the other tested methods.

The data are from the CVPR IMW 2020 PhotoTourism challenge. Correspondences were obtained using RootSIFT features and mutual nearest neighbour matching. We used all scenes from the test set, i.e., Sacre Coeur, St Peters Square, Brandenburg Gate, Buckingham Palace, Colosseum Exterior, Grand Place Brussels, Notre Dame Front Facade, Palace of Westminster, Pantheon Exterior, Prague Old Town Square, Taj Mahal, Temple Nara Japan, Trevi Fountain, Westminster Abbey. From the validation set, we used only scene British Museum to tune the hyper-parameters of the methods. Each scene contains 4950 image pairs. The reported accuracy is



(a) Fundamental matrix estimation (b) Relative pose estimation (c) Homography estimation

Fig. 3. The mean Average Accuracy of the tested robust estimators on fundamental matrix, relative pose and homography estimation. For each problem, the methods are ordered according to their scores. We used all scenes from the test set of the CVPR IMW 2020 PhotoTourism challenge. For F and E estimation, the methods were tested on a total of 54450 image pairs. Abbrevations used: OpenCV RANSAC (RANSAC), GC-RANSAC + DEGENSAC (GC + DEG), GC-RANSAC + DEGENSAC + MAGSAC++ scoring (GC + DEG + M++). *Higher value is better.*



(a) Fundamental matrix estimation (b) Relative pose estimation (c) Homography estimation

Fig. 4. The mean Average Accuracy (top row; higher is better) and average processing time (bottom; in seconds; lower is better) plotted as the function of the inlier-outlier threshold (or its upper limit; horizontal axis) parameter. For fundamental matrix and relative pose estimation, only scene British Museum was used. Homographies were estimated from both the EVD and HPatches datasets. The threshold (horizontal axis) is shown on a *logarithmic scale* – the right half of the plots covers a significantly larger area than the left one.

calculated on the total of 54450 image pairs from the test set using the parameters tuned on scene British Museum.

The results on the test set are shown in Fig. 3a. It can be seen that MAGSAC, MAGSAC++, GC-RANSAC, GC-RANSAC + DEGENSAC, and GC+RANSAC + DEGENSAC with MAGSAC++ quality function leads to similar accuracy. The maximum mAA difference between their results is 0.007. The most accurate results are obtained by GC-RANSAC with DEGENSAC and the proposed MAGSAC++ quality function. The other methods which do not need to a set a single threshold value, i.e., AC-RANSAC and LMeDS, are significantly less accurate. AC-RANSAC when applied without an upper bound (AC-RANSAC ∞) fails to return reasonable solutions in most of the cases. With an upper bound, it is more accurate than the RANSAC implemented in OpenCV.

The first row of Fig. 4a plots the mAA scores on scene British Museum as the function of the inlier-outlier threshold used for the estimation. We chose this scene since it is the first one in the validation set when the scene names are ordered alphabetically. All methods expect for MAGSAC and MAGSAC++, have a similar trend, i.e., their results increase slightly in the beginning while the threshold approaches its optimal value – for example, 0.75 px for USAC. Then their accuracy starts dropping dramatically. The trend of MAGSAC and MAGSAC++ is different. If the maximum threshold is set to a too low value, e.g., < 1 px, the results are inaccurate as it is expected. Between 1 and 10 pixels, the results are reasonably stable. This range is much wider than for the other methods which are only stable in-between $0.5 - 1.5$ pixels. Graph-Cut RANSAC with DEGENSAC and the proposed MAGSAC++ scoring shows an interesting trend, since it leads to almost constant mAA score in-between $0.1 - 1.5$ px threshold, then it starts deteriorating, however, less significantly than most of the other methods. The second row of Fig. 4a shows the processing time as the function of the threshold. It can be seen that MAGSAC++ is faster than MAGSAC as it is expected. It leads to similar processing time to its other less accurate alternatives.

TABLE 2
The Insensitivity (16) to the Inlier-Outlier Threshold (or its Upper Bound) is Shown on Fundamental Matrix (F), Essential Matrix (E), and Homography (H) Fitting. The best values are shown in red, the second best ones are in blue.

	F	E	H	AVG
OpenCV RANSAC	0.076	0.342	0.358	0.259
OpenCV RHO	–	–	0.329	–
USAC	0.096	0.590	0.452	0.379
GC + DEG	0.113	–	–	–
AC-RANSAC	0.118	0.670	0.421	0.403
GC-RANSAC	0.125	0.489	0.261	0.292
GC (+ DEG) + M++	0.170	0.564	0.275	0.336
MAGSAC	0.215	**0.797**	**0.519**	0.510
MAGSAC++	0.273	0.776	0.514	**0.521**
MAGSAC++ + DEG	**0.279**	–	–	–

The first column of Table 2 reports the threshold-insensitivity score on scene British Museum calculated as proposed in (16). MAGSAC++ combined with DEGENSAC yields the highest score and, thus, that method is the least sensitive to the setting of the inlier-outlier threshold – it is the easier to be used when applying robust estimation to a yet unseen scene.

7.2 Essential Matrix Estimation

The methods compared on relative pose (i.e., essential matrix) estimation are OpenCV RANSAC [1], OpenCV LMedS [50], LO-RANSAC [26], GC-RANSAC [51], USAC [40], AC-RANSAC [30], MAGSAC, MAGSAC++, and GC-RANSAC with MAGSAC++ quality function. DEGENSAC is not included in these tests since it is for recovering the fundamental matrix from scenes with dominant planar structures. For the five-point algorithm [46], planar scenes are not degenerate. Since the datasets used for fundamental matrix estimation contain the intrinsic camera parameters as well, we used the same scenes.

Fig. 3b shows that the most accurate essential matrices are clearly obtained by MAGSAC++ which achieves $\approx 4\%$ higher mAA score than the second best MAGSAC. The other methods which do not need to a set a single threshold value, i.e., AC-RANSAC and LMeDS, are significantly less accurate, however, they are better than for fundamental matrix estimation. AC-RANSAC without an upper bound (AC-RANSAC ∞) fails to return reasonable solutions in most of the cases. With an upper bound, it is more accurate than OpenCV RANSAC and USAC.

The top row of Fig. 4b shows similar trend as for fundamental matrix estimation. All methods but MAGSAC and MAGSAC++ have a clear "best" threshold. If it is exceeded, their accuracy deteriorates dramatically. The results of MAGSAC and MAGSAC++ are *almost constant* throughout the range of thresholds. Interestingly, MAGSAC++ is the most accurate when the threshold upper bound is set to a small value, e.g., 0.1. Its results are just slightly less accurate for other threshold values. AC-RANSAC performs better here than for fundamental matrix estimation. The processing times are shown in the bottom row of Fig. 4b. MAGSAC++ is significantly faster for most of the threshold values than the other robust estimators. While AC-RANSAC leads to reasonable accuracy, it is significantly slower than the other methods.

7.3 Homography Estimation

For homography estimation, we used the `Extreme-View` [14] (`EVD`) and `HPatches` [16] datasets partitioned into test and validation sets as done in [47]. They consist of image pairs of different sizes from 329×278 up to 1712×1712 with point correspondences provided. The pairs of `EVD` undergo an extreme view change, i.e., wide baseline or extreme zoom. The `HPatches` scenes are extracted from a number of image sequences, where each sequence contains images of some planar object, e.g., a painting or a wall with graffiti. Since the datasets contain significantly fewer images then the ones used for epipolar geometry estimation, we repeated every method 100 times on each image pair. Besides the methods used for epipolar geometry estimation, we included the RHO [52] method implemented in OpenCV. The validation set was used to tune the hyper-parameters of the methods. The accuracy is measured on the test set.

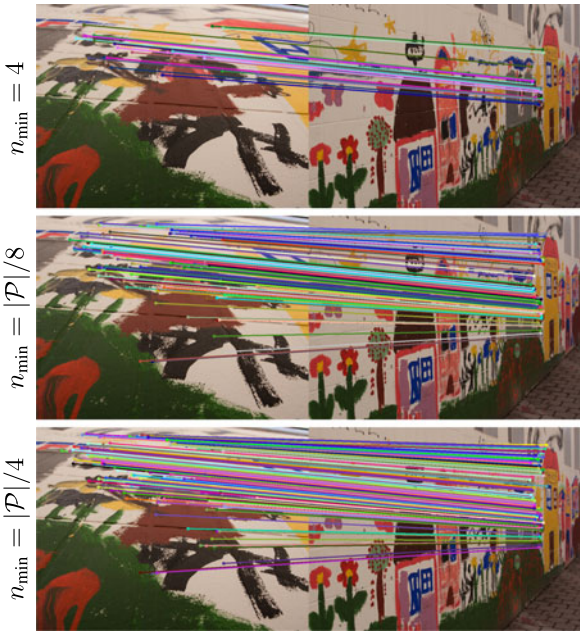


Fig. 5. The inliers of the estimated homography selected by the proposed adaptive strategy with varying the parameter $n_{\min}$ which controls the minimum number of required inliers.

It can be seen in Fig. 3c that the most accurate results are estimated by GC-RANSAC, MAGSAC++ and LO-RANSAC with a marginal difference of 0.002 – 0.005 in their mAA scores. The same trend can be observed for AC-RANSAC and LMeDS as before. AC-RANSAC with its threshold upper bound tuned works reasonably well. LMeDS fails to return accurate results. The mAA scores on the test set using varying threshold values are shown in the top row of Fig. 4c. Since the methods do not seem to be as sensitive to the inlier-outlier threshold as when fitting epipolar geometry, we tested a much wider range 0.1 – 100 than previously. The performance of MAGSAC and MAGSAC++ is very stable if $\sigma_{\max}$ is chosen from interval $[5, 100]$, where the accuracy difference is small. They achieve their maximum accuracy at $\sigma_{\max} = 25$, however, the accuracy drops only marginally for higher values. The bottom plot of Fig. 4c shows the processing time in seconds as the function of the inlier-outlier threshold. If the threshold is set to a small value (≤ 1) all methods, except RHO, gets slow. However, if $\tau(\sigma)$ or $\tau(\sigma_{\max})$ is greater than 3 the proposed MAGSAC and MAGSAC++ is similarly fast as the other methods running at real-time speed. While RHO is extremely fast for all settings, it is reasonably accurate only for a narrow range of thresholds, where all the other methods are similarly fast.

7.4 Iteratively Re-Weighted Least-Squares on 2D Lines

We compare the proposed iterative re-weighting strategy without the other components of MAGSAC++. To do so, we generated 100 2D points stemming from a 2D line and outliers. The outliers were generated uniformly randomly within a window of size 1000×1000. A 2D line passing through the middle of the window is generated with a random normal.

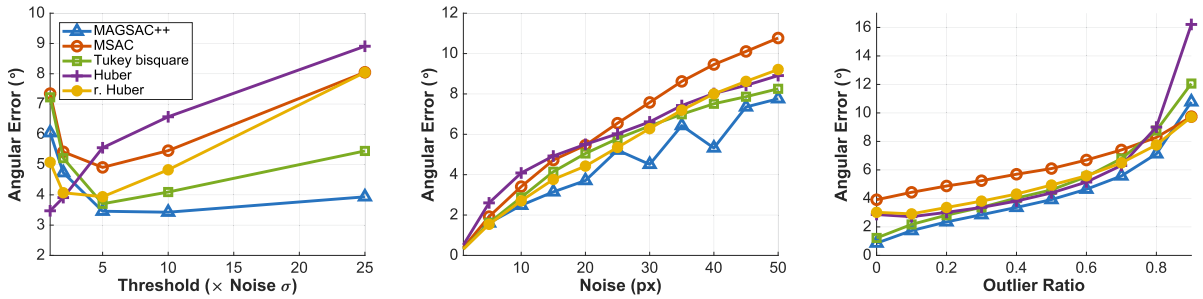


Fig. 6. The average results of iteratively re-weighted least-squares fitting using different robust weights (i.e., the proposed MAGSAC++, MSAC, Tukey bisquare, Huber and redescending Huber weights) when fitting 2D lines. The methods were repeated 10000 times using each parameter setting. (*Left*) The angular error, in degrees, of the estimated lines are plotted as the function of inlier-outlier threshold multiplier. The actual threshold is calculated by multiplying the noise σ by the values shown on the horizontal axis. (*Middle*) The angular error is plotted as the function of the noise σ added to the point coordinates. (*Right*) The angular error is plotted as the function of the outlier ratio.

Points were sampled from the line uniformly randomly and, then, zero-mean Gaussian-noise was added to their coordinates. We tested the following parameters: noise $\sigma \in \{0, 5, \ldots, 50\}$; outlier ratio $\mu \in \{0.0, 0.1, \ldots, 0.9\}$; threshold multiplier $\tau \in \{1, 2, 5, 10, 25\}$. The actual inlier-outlier threshold is calculated by multiplying τ with the noise scale σ. For each configuration, 10000 tests were run.

Fig. 6 plots the average angular errors (in degrees) as the function of the tested parameters. The compared robust weighting techniques are the proposed MAGSAC++; MSAC, assigning weight 1 if the point is closer than the threshold and, otherwise 0; Tukey bi-square weighting; Huber weights and re-descending Huber weights. It be seen that the MAGSAC++ weights guide the IRLS more successfully than the other compared techniques. Thus, the final errors of MAGSAC++ are smaller if threshold is set reasonably large. Also, it is the least sensitive to over-estimating the threshold value – its results are just slightly affected even if the actual threshold is 25 times the noise scale. Note that the offset errors of the estimated lines show a similar trend.

7.5 Inlier Selection

To test the proposed inlier selection, we generated a synthetic scene similarly as in the previous section. We compared the proposed technique with MINPRAN [29] and a contrario RANSAC [30]. We measured the average model error (15), in pixels, and the number of returned inliers. All algorithms got the ground truth line as input to select the inliers. Each test was repeated 10000 times. The results are shown in Fig. 7. The average model accuracy (left) and the number of inliers returned (right) of the compared adaptive threshold selection techniques are plotted as the function of the image noise, in pixels. From the left plot, it can be seen that the proposed technique returns inlier sets which lead to significantly more similar models, to the input one, than the other algorithms. The average model error of the proposed method for inlier ratio 0.1 is lower than the error of the other method for inlier ratio 0.9. For the fair comparison, it is important to note that MINPRAN and AC-RANSAC solve a different problem, i.e., selecting the noise scale which minimizes the randomness of the points which fall closer than the threshold. Their objective function is designed to select both the model and noise scale together. In our case, the input model is accurate and, therefore, we only need a set of inliers leading to a similarly accurate model.

From the right plot of Fig. 7, it can be seen that the proposed inlier selection usually returns fewer points than the other methods if the inlier ratio is higher than 0.1. The number of points that suffices depends on a particular

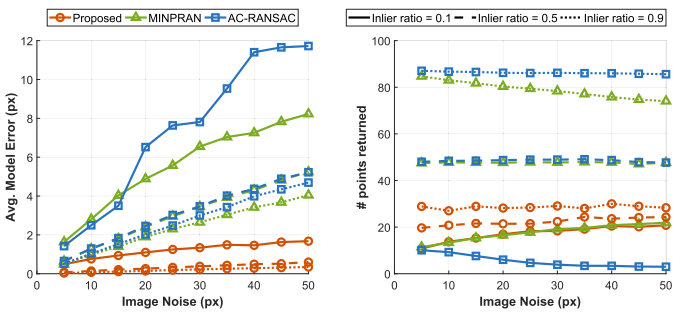


Fig. 7. The avg. model error (left) and the number of returned inliers (right) of adaptive threshold selection techniques are plotted as the function of the image noise (in pixel). Synthetic scene: points from a 2D line with zero-mean Gaussian-noise and uniformly distributed outliers (in total, 100 points), 10000 runs on each setting.

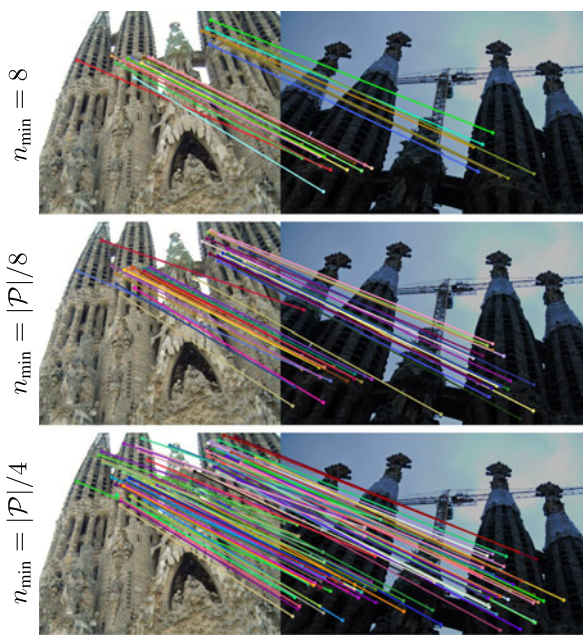


Fig. 8. The inliers of the estimated fundamental matrix selected by the proposed adaptive strategy with varying the parameter $n_{\min}$ which controls the minimum number of required inliers.

application where the proposed method is used. For example, for doing a cheirality check after decomposing an essential matrix, a few correspondences are usually enough, while a scene reconstruction might need many points. Setting the minimum number of points required to $n_{\min}$ is straightforward by initially including the $n_{\min}$ points with the lowest residuals. The algorithm starts adding new points from the $(n_{\min}+1)$th closest one. The upper bound of $n_{\min}$ is the number of points with residuals smaller than $\tau(\sigma_{\max})$.

Example scenes showing the proposed adaptive inlier selection with different values for $n_{\min}$ in the cases of homography and fundamental matrix estimation are shown in Figs. 5 and 8, respectively. Three different values are tested for $n_{\min}$ which are m (4 for homographies; 8 for fundamental matrices), $|\mathcal{P}|/8$ and $|\mathcal{P}|/4$. In these examples, all the selected inliers are correct. Moreover, a reasonable number of inliers are returned even when $n_{\min} = m$. Note that even if the ground truth inlier number is lower than, e.g., $|\mathcal{P}|/4$, the algorithm is guaranteed to return the inliers which lead to an as similar model as possible to the input one.

8 Conclusion

We formulate a novel marginalization procedure as an iteratively re-weighted least-squares (IRLS) approach. We introduce a new model quality (scoring) function, that is increased by this IRLS approach, and a termination criterion for RANSAC-like robust estimation that does not require a crisp inlier-outlier decision. Also, a new method for adaptive inlier selection is proposed assuming that an accurate model is known. Combining the proposed techniques, the "bells and whistles" of USAC [40], e.g., pre-emptive verification, degeneracy testing, and a number of technical improvements, we propose MAGSAC++.

To the experiments, MAGSAC++ leads to the most accurate relative pose estimation. When all methods are tested using their "best" inlier-outlier thresholds, the most accurate fundamental matrices are obtained by combining the proposed quality function with GC-RANSAC [51]. For homography estimation, MAGSAC++ is the second most accurate method with only marginally higher errors than first one, i.e., GC-RANSAC. In practice, this "best" threshold is usually unknown. In those cases, both MAGSAC and MAGSAC++ are significantly less sensitive to the setting of the noise scale or its upper limit than the other state-of-the-art robust estimators. The source code and examples implemented both in C++ and Python are available at https://github.com/danini/magsacand in OpenCV.

Acknowledgments

This work was supported in part by the Czech Science Foundation under Grant GA18-05360S, in part by the Ministry of Education OP VVV Project CZ.02.1.01/0.0/0.0/16 019/0000765 Research Center for Informatics, in part by the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems National Laboratory Program, and in part by the Artificial Intelligence National Laboratory Program.

References

[1] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[2] P. H. S. Torr and D. W. Murray, "Outlier detection and motion segmentation," in *Optical Tools for Manufacturing and Advanced Automation*. Bellingham, WA, USA: Int. Soc. Opt. Photon., 1993, pp. 432–443.

[3] P. H. S. Torr, A. Zisserman, and S. J. Maybank, "Robust detection of degenerate configurations while estimating the fundamental matrix," *Comput. Vis. Image Understanding*, vol. 71, no. 3, pp. 312–333, 1998.

[4] P. Pritchett and A. Zisserman, "Wide baseline stereo matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1998, pp. 754–760.

[5] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.

[6] D. Mishkin, J. Matas, and M. Perdoch, "MODS: Fast and robust method for two-view matching," 2015, *arXiv:1503.02619*.

[7] C. Sminchisescu, D. Metaxas, and S. Dickinson, "Incremental model-based estimation using geometric constraints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 727–738, May 2005.

[8] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.

[9] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *Eur. Conf. Comput. Vis.*, 2016, pp. 501–518.

[10] D. Ghosh and N. Kaabouch, "A survey on image mosaicking techniques," *J. Vis. Commun. Image Representation*, vol. 34, pp. 1–11, 2016.

[11] M. Zuliani, C. S. Kenney, and B. S. Manjunath, "The multiRANSAC algorithm and its application to detect planar homographies," in *Proc. IEEE Int. Conf. Image Process.*, 2005, pp. III–153.

[12] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting," *Int. J. Comput. Vis.*, vol. 97, pp. 123–147, 2012.

[13] T. T. Pham, T.-J. Chin, K. Schindler, and D. Suter, "Interacting geometric priors for robust multimodel fitting," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4601–4610, Oct. 2014.

[14] K. Lebeda, J. Matas, and O. Chum, "Fixing the locally optimized RANSAC," in *Proc. Brit. Mach. Vis. Conf.* 2012, pp. 95.1–95.11.

[15] E Trulls, Y. Jun, K. Yi, D. Mishkin, J. Matas, and P. Fua, "Image matching challenge," 2020. [Online]. Available: http://cmp.felk.cvut.cz/cvpr2020-ransac-tutorial/

[16] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3852–3861.

[17] P. H. Torr, S. J. Nasuto, and J. M. Bishop, "NAPSAC: High noise, high dimensional robust estimation-it's in the bag," in *Proc. Brit. Mach. Vis. Conf.*, 2002, pp. 458–467.

[18] K. Ni, H. Jin, and F. Dellaert, "GroupSAC: Efficient consensus in the presence of groupings," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 2193–2200.

[19] O. Chum and J. Matas, "Matching with PROSAC-progressive sample consensus," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 220–226.

[20] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, "MAGSAC++, a fast, reliable and accurate robust estimator," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1301–1309.

[21] O. Chum and J. Matas, "Randomized RANSAC with TDD test," in *Proc. Brit. Mach. Vis. Conf.*, 2002, vol. 2, pp. 448–457.

[22] D. P. Capel, "An effective bail-out test for RANSAC consensus scoring," in *Proc. Brit. Mach. Vis. Conf.*, 2005, pp. 78.1–78.10.

[23] J. Matas and O. Chum, "Randomized RANSAC with sequential probability ratio test," in *Proc. 10th IEEE Int. Conf. Comput. Vis.* Volume 1, 2005, vol. 2, pp. 1727–1732.

[24] O. Chum and J. Matas, "Optimal randomized RANSAC," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 8, pp. 1472–1482, Aug. 2008.

[25] A. Wald, *Sequential Analysis*. Chelmsford, MA, USA: Courier Corporation, 2004.

[26] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *Proc. Joint Pattern Recognit. Symp.*, 2003, pp. 236–243.

[27] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Comput. Vis. Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.
[28] P. H. S. Torr, "Bayesian model estimation and selection for epipolar geometry and generic manifold fitting," *Int. J. Comput. Vis.*, vol. 50, pp. 35–61, 2002.
[29] C. V. Stewart, "MINPRAN: A new robust estimator for computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 10, pp. 925–938, Oct. 1995.
[30] L. Moisan, P. Moulon, and P. Monasse, "Automatic homographic registration of a pair of images, with a contrario elimination of outliers," *Image Process. Line*, vol. 2, pp. 56–73, 2012.
[31] A. Cohen and C. Zach, "The likelihood-ratio test and efficient robust estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2282–2290.
[32] R. Raguram and J.-M. Frahm, "Recon: Scale-adaptive robust estimation via residual consensus," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1299–1306.
[33] M. Rais, G. Facciolo, E. Meinhardt-Llopis , J.-M. Morel, A. Buades, and B. Coll, "Accurate motion estimation through random sample aggregated consensus," 2017, *arXiv: 1701.05268*.
[34] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.
[35] D. Barath, J. Noskova, and J. Matas, "MAGSAC: marginalizing sample consensus," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10189–10197.
[36] R. A. Maronna, R. D. Martin, V. J. Yohai, and M. Salibián-Barrera, *Robust Statistics: Theory and Methods (with R)*. Hoboken, NJ, USA: Wiley, 2019.
[37] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis*. Berlin, Germany: Springer, 1978, pp. 105–116.
[38] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[39] C. G. Baker, K. A. Gallivan, and P. Van Dooren , "Low-rank incremental methods for computing dominant singular subspaces," *Linear Algebra Appl.*, vol. 436, no. 8, pp. 2866–2888, 2012.
[40] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, "USAC: A universal framework for random sample consensus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 2022–2038, Aug. 2013.
[41] O. Chum, T. Werner, and J. Matas, "Two-view geometry estimation unaffected by a dominant plane," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 772–779.
[42] H. Haber. Three-dimensional proper and improper rotation matrices. 2011. [Online]. Available: http://scipp.ucsc.edu/haber/ph116A/rotation_11.pdf
[43] O. Chum, T. Werner, and J. Matas, "Epipolar geometry estimation via RANSAC benefits from the oriented epipolar constraint," in *Proc. Int. Conf. Pattern Recognit.*, 2004, pp. 112–115.
[44] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. Comput. Vis.*, 1999, pp. 1150–1157.
[45] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 580–593, Jun. 1997.
[46] H. Stewénius, D. Nistér, F. Kahl, and F. Schaffalitzky, "A minimal solution for relative pose with unknown focal length," *Image Vis. Comput.*, vol. 26, no. 7, pp. 871–877, 2008.
[47] D. Barath, T.-J. Chin, O. Chum, D. Mishkin, R. Ranftl, and J. Matas, "RANSAC in 2020 tutorial," 2020. [Online]. Available: http://cmp.felk.cvut.cz/cvpr2020-ransac-tutorial/
[48] K. Moo Yi , E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to find good correspondences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2666–2674.
[49] Y. Jin *et al.*, "Image matching across wide baselines: From paper to practice," 2020, *arXiv:2003.01587*.
[50] P. J. Rousseeuw, "Least median of squares regression," *J. Amer. Statist. Assoc.*, vol. 79, no. 388, pp. 871–880, 1984.
[51] D. Barath and J. Matas, "Graph-cut RANSAC," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6733–6741.
[52] O. Bilaniuk, H. Bazargani, and R. Laganiere, "Fast target recognition on mobile devices: revisiting gaussian elimination for the estimation of planar homographies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 119–125.

Daniel Barath was born in Budapest in 1989. He received the PhD defense degree from Eotvos Lorand University in 2019. He is a member of the Visual Recognition Group, FEE, Czech Technical University, Prague, Czech Republic, and the Machine Perception Research Laboratory, Institute for Computer Science and Control, Budapest, Hungary. His research interests include robust model estimation and minimal methods in computer vision.

Jana Noskova received the master's degree in probability and mathematical statistics and the doctoral degree in robust statistics from Charles University, Prague, in 1996. She is currently an assistant professor with the Department of Mathematics and a researcher with Visual Recognition Group, Department of Cybernetics, Czech Technical University, Prague.

Jiri Matas is currently a professor with the Center for Machine Perception, Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic. He has authored or coauthored more than 250 papers in computer vision and machine learning. His research interests include object recognition, image retrieval, tracking, sequential pattern recognition, invariant feature detection, and Hough transform and RANSAC-type optimization.