

MAGSAC++, a fast, reliable and accurate robust estimator

Daniel Barath^{1,2}, Jana Noskova¹, Maksym Ivashechkin¹, and Jiri Matas¹

¹ Visual Recognition Group, Department of Cybernetics, Czech Technical University, Prague

² Machine Perception Research Laboratory, MTA SZTAKI, Budapest

barath.daniel@sztaki.mta.hu

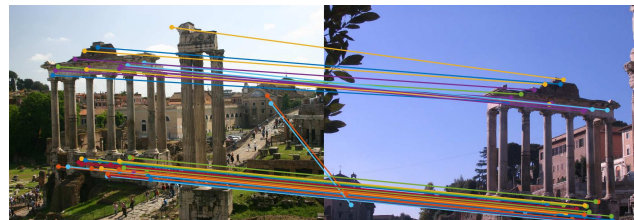
Abstract

A new method for robust estimation, MAGSAC++¹, is proposed. It introduces a new model quality (scoring) function that does not require the inlier-outlier decision, 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. We also propose a new sampler, Progressive NAPSAC, for RANSAC-like robust estimators. Exploiting the fact that nearby points often originate from the same model in real-world data, it finds local structures earlier than global samplers. The progressive transition from local to global sampling does not suffer from the weaknesses of purely localized samplers. On six publicly available real-world datasets for homography and fundamental matrix fitting, MAGSAC++ produces results superior to the state-of-the-art robust methods. It is faster, more geometrically accurate and fails less often.

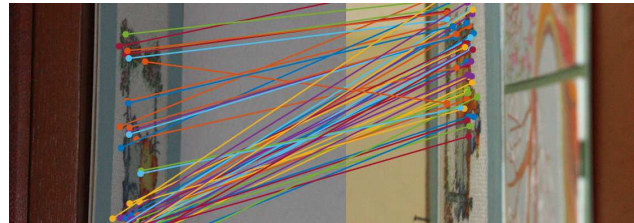
1. Introduction

The RANSAC (RANdom SAMple Consensus) algorithm [5] 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., motion segmentation [27], short baseline stereo [27, 29], wide baseline matching [19, 14, 15], detection of geometric primitives [23], image mosaicing [7], and to perform [32] or initialize multi-model fitting [9, 18]. In brief, RANSAC repeatedly selects minimal random subsets of the input point set and fits a model, e.g., a line to two 2D points or a fundamental matrix to seven 2D point correspondences. Next, the quality of the estimated model is measured, for instance by 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 of all inliers, is returned.

¹<https://github.com/danini/magsac>



(a) Community Photo Collection dataset [30].



(b) ExtremeView dataset [11].



(c) Tanks and Temples dataset [10].

Figure 1. Image pairs where all tested robust estimators (i.e., LMeDS [22], RANSAC [5], MSAC [28], GC-RANSAC [1], MAGSAC [2]) failed, except the proposed MAGSAC++. Inlier correspondences found by MAGSAC++ are drawn by lines.

Since the publication of RANSAC, many modifications have been proposed improving the algorithm. For example, MLESAC [28] estimates the model quality by a maximum likelihood process with all its beneficial properties, albeit under certain assumptions about data distributions. In practice, MLESAC results are often superior to the inlier counting of plain RANSAC, and they are less sensitive to the user-defined inlier-outlier threshold. In MAPSAC [26], the robust estimation is formulated as a process that esti-

mates both the parameters of the data distribution and the quality of the model in terms of maximum a posteriori.

Methods for reducing the dependency on the inlier-outlier threshold include MINPRAN [24] which assumes that the outliers are uniformly distributed and finds the model where the inliers are least likely to have occurred randomly. Moisan et al. [16] proposed a contrario RANSAC, selecting the most likely noise scale for each model. Barath et al. [2] proposed the Marginalizing Sample Consensus method (MAGSAC) marginalizing over the noise σ to eliminate the threshold from the model quality calculation.

The MAGSAC algorithm, besides not requiring a manually set threshold, was reported to be significantly more accurate than other robust estimators on various problems, on a number of datasets. The improved accuracy originates from the new model quality function and σ -consensus model polishing. The quality function marginalizes over the noise scale with the data interpreted as a mixture of uniformly distributed outliers and inliers with residuals having χ^2 -distribution. The σ -consensus algorithm replaces the originally used least-squares (LS) fitting with weighted least-squares (WLS) where the weights are calculated via the marginalization procedure – which requires a number of independent LS estimations on varying sets of points.

Due to the several LS fittings, σ -consensus is slow. In [2], a number of tricks (e.g., preemptive verification; down-sampling of σ values) are proposed to achieve acceptable speed. However, *MAGSAC is often significantly slower* than other robust estimators. In this paper, we propose new quality and model polishing functions, reformulating the problem as an M-estimation with a novel class of M-estimators (a robust kernel) solved by an iteratively reweighted least squares procedure. In each step, the weights are calculated making the same assumptions about the data distributions as in MAGSAC, but, without requiring a number of expensive LS fittings. The proposed MAGSAC++ and σ -consensus++ methods lead to *more accurate results than the original MAGSAC algorithm, often, an order-of-magnitude faster*.

In practice, there are also other ways of speeding up robust estimation. NAPSAC [20] and PROSAC [4] modify the RANSAC sampling strategy to increase the probability of selecting an all-inlier sample early. PROSAC exploits an a priori predicted inlier probability rank of the points and starts the sampling with the most promising ones. PROSAC and other RANSAC-like samplers treat models without considering that inlier points often are in the proximity of each other. This approach is effective when finding a global model with inliers sparsely distributed in the scene, for instance, the rigid motion induced by changing the viewpoint in two-view matching. However, as it is often the case in real-world data, if the model is localized with inlier points close to each other, robust estimation can be

significantly sped up by exploiting this in the sampling.

NAPSAC assumes that inliers are spatially coherent. It draws samples from a hyper-sphere centered at the first, randomly selected, point. If this point is an inlier, the rest of the points sampled in its proximity are more likely to be inliers than the points outside the ball. NAPSAC leads to fast, successful termination in many cases. However, it suffers from a number of issues in practice. First, the models fit to local all-inlier samples are often imprecise due to the bad conditioning the points. Second, in some cases, estimating a model from a localized sample leads to degenerate solutions. For example, when fitting a fundamental matrix by the seven-point algorithm, the correspondences must originate from more than one plane. Therefore, there is a trade-off between near, likely all-inlier, and global, well-conditioned, lower all-inlier probability samples. Third, when the points are sparsely distributed and not spatially coherent, NAPSAC often fails to find the sought model.

We propose in this paper, besides MAGSAC++, the Progressive NAPSAC (P-NAPSAC) sampler which merges the advantages of local and global sampling by drawing samples from gradually growing neighborhoods. Considering that nearby points are more likely to originate from the same geometric model, P-NAPSAC finds local structures earlier than global samplers. In addition, it does not suffer from the weaknesses of purely localized samplers due to progressively blending from local to global sampling, where the blending factor is a function of the input data.

The proposed methods were tested on homography and fundamental matrix fitting on six publicly available real-world datasets. MAGSAC++ combined with P-NAPSAC sampler is superior to state-of-the-art robust estimators in terms of speed, accuracy and failure rate. Example model estimations when all tested robust estimators, except MAGSAC++, failed, are shown in Fig. 1.

2. MAGSAC++

We propose a new quality function and model fitting procedure for MAGSAC [2]. It is shown that the new method can be formulated as an M-estimation solved by the iteratively reweighted least squares (IRLS) algorithm.

The marginalizing sample consensus (MAGSAC) algorithm is based on two assumptions. *First*, the noise level σ is a random variable with density function $f(\sigma)$. Having no prior information, σ is assumed to be uniformly distributed, $\sigma \sim \mathcal{U}(0, \sigma_{\max})$, where σ_{\max} is a user-defined maximum noise scale. *Second*, for a given σ , the residuals of the inliers are described by the trimmed χ -distribution² with n degrees of freedom multiplied by σ with density

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

²The square root of χ^2 -distribution.

for $r < \tau(\sigma)$ and $g(r | \sigma) = 0$ for $r \geq \tau(\sigma)$. Constant $C(n) = (2^{n/2}\Gamma(n/2))^{-1}$ and, for $a > 0$,

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

is the gamma function, n is the dimension of Euclidean space in which the residuals are calculated and $\tau(\sigma)$ is set to a high quantile (e.g., 0.99) of the non-trimmed distribution.

Suppose that we are given input point set \mathcal{P} and model θ estimated from a minimal sample of the data points as in RANSAC. Let $\theta_\sigma = F(I(\theta, \sigma, \mathcal{P}))$ be the model estimated from the inlier set $I(\theta, \sigma, \mathcal{P})$ selected using $\tau(\sigma)$ 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

$$P(p | \theta_\sigma, \sigma) = 2C(n)\sigma^{-n} D^{n-1}(\theta_\sigma, p) \exp\left(\frac{-D^2(\theta_\sigma, p)}{2\sigma^2}\right),$$

if $D(\theta_\sigma, p) \leq \tau(\sigma)$, where $D(\theta_\sigma, p)$ is the point-to-model residual. If $D(\theta_\sigma, p) > \tau(\sigma)$, likelihood $P(p | \theta_\sigma, \sigma)$ is 0.

In MAGSAC, the final model parameters are calculated by weighted least-squares where the weights of the points come from marginalizing the likelihoods over σ . When marginalizing over σ , each $P(p | \theta_\sigma, \sigma)$ 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 paper.

In MAGSAC++, we propose a new approach instead of the original one, requiring several LS fittings when marginalizing over the noise level σ . The proposed algorithm is an iteratively reweighted least squares (IRLS) where the model parameters in the $(i + 1)$ th step are calculated as follows:

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

where the weight of point p is

$$w(D(\theta_i, p)) = \int P(p | \theta_i, \sigma) f(\sigma) d\sigma \quad (2)$$

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

2.1. Weight calculation

The weight function defined in (2) is the marginal density of the inlier residuals as follows:

$$w(r) = \int g(r | \sigma) f(\sigma) d\sigma.$$

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

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

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

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

is the upper incomplete gamma function.

Weight $w(r)$ is positive and decreasing on interval $(0, k\sigma_{\max})$. 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 ([13], chapter 9). Consequently, it converges to a local minimum. This IRLS with $\tau(\sigma) = 3.64\sigma$, where 3.64 is the 0.99 quantile of χ -distribution, will be called σ -consensus++. For problems using point correspondences, $n = 4$. Parameter σ_{\max} is the same user-defined maximum noise level parameter as in MAGSAC, usually, set to a fairly high value, e.g., 10 pixels. 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.

2.2. Model quality function

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

$$Q(\theta, \mathcal{P}) = \frac{1}{L(\theta, \mathcal{P})}, \quad (3)$$

where

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

is a loss function of the M-estimator defined by our weight function $w(r)$. Function $\rho(r) = \int_0^r xw(x)dx$ for $r \in [0, +\infty)$. For $0 \leq r \leq k\sigma_{\max}$,

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

For $r > k\sigma_{\max}$,

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

where

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

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 (3) as quality function and σ -consensus++ for estimating the model parameters.

3. Progressive NAPSAC sampling

We propose a new sampling technique which gradually moves from local to global, assuming initially that localized minimal samples are more likely to be all-inlier. If the assumption does not lead to termination, the process gradually moves towards the randomized sampling of RANSAC.

3.1. N Adjacent Points Sample Consensus

The N Adjacent Points Sample Consensus (NAPSAC) sampling technique [20] builds on the assumption that the points of a model are spatially structured and, thus, sampling from local neighborhoods increases the inlier ratio locally. In brief, the algorithm is as follows:

1. Select an initial point \mathbf{p}_i randomly from all points.
2. Find the set $\mathcal{S}_{i,r}$ of points lying within the hyper-sphere of radius r centered at \mathbf{p}_i .
3. If the number of points in $\mathcal{S}_{i,r}$ is less than the minimal sample size then restart from step 1.
4. Point \mathbf{p}_i and points from $\mathcal{S}_{i,r}$ selected uniformly form the minimal sample.

There are three major issues of local sampling in practice. *First*, models fit to local all-inlier samples are often too imprecise (due to bad conditioning). *Second*, in some cases, estimating a model from a local sample leads to degeneracy. For example, for fundamental matrix fitting, the correspondences must originate from more than one plane. This usually means that the correspondences are beneficial to be far. Thus, purely localized sampling fails. *Third*, when having global structures, e.g., the rigid motion of the background in an image pair, local sampling is much slower than global. We, therefore, propose a transition between local and global sampling progressively blending from one into the other.

3.2. Progressive NAPSAC – P-NAPSAC

In this section, Progressive NAPSAC is proposed combining the strands of NAPSAC-like local sampling and the global sampling of RANSAC. The P-NAPSAC sampler proceeds as follows: the first, location-defining, point in the minimal sample is chosen using the PROSAC strategy. The remaining points, are selected *from a local neighbourhood*, according to their distances. The process samples from the m points nearest to the center defined by the first point in

Algorithm 1 Outline of Progressive NAPSAC.

Input: \mathcal{P} – points; \mathcal{S} – neighborhoods; n – point number
 1: $t_1, \dots, t_n := 0$ ▷ The hit numbers.
 2: $k_1, \dots, k_n := m$ ▷ The neighborhood sizes.

Repeat until termination:

Selection of the first point:

- 3: Let \mathbf{p}_i be a random point. ▷ Selected by PROSAC.
 - 4: $t_i := t_i + 1$ ▷ Increase the hit number.
 - 5: **if** $(t_i \geq T'_{k_i} \wedge k_i < n)$ **then**
 - 6: $k_i := k_i + 1$ ▷ Enlarge the neighborhood.
-

Semi-random sample \mathcal{M}_{i,t_i} of size m :

- 7: **if** $\mathcal{S}_{i,k_i-1} \neq \mathcal{P}$ **then**
 - 8: Put \mathbf{p}_i ; the k_i th nearest neighbor; and $m-2$ random points from \mathcal{S}_{i,k_i-1} into sample \mathcal{M}_{i,t_i} .
 - 9: **else**
 - 10: Select $m-1$ points from \mathcal{P} at random.
-

Increase the hit number of the points from \mathcal{M}_{i,t_i} :

- 11: **for** $\mathbf{p}_j \in \mathcal{M}_{i,t_i} \setminus \mathbf{p}_i$ **do** ▷ For all points in the sample,
- 12: **if** $\mathbf{p}_i \in \mathcal{S}_{j,k_j}$ **then** ▷ if the i th one is close,
- 13: $t_j := t_j + 1$ ▷ increase the hit number.

Model parameter estimation

- 14: Compute model parameters θ from sample \mathcal{M}_{i,t_i} .

Model verification

- 15: Calculate model quality.
-

the minimal sample. The size of the local subset of points is increased data-dependently, as described below. If no quality function is available, the first point is chosen at random similarly as in RANSAC, the other points are selected uniformly from a progressively growing neighbourhood.

In the case of having local models, the samples are more likely to contain solely inliers and, thus, trigger early termination. When the points of the sought model do not form spatially coherent structures, the gradual increment of neighborhoods leads to finding global structures not noticeably later than by using global samplers, e.g., PROSAC.

Growth function and sampling. The design of the growth function defining how fast the neighbourhood grows around a selected point \mathbf{p}_i must find the balance between the strict NAPSAC assumption – entirely localized models – and the RANSAC approach treating every model on a global scale.

Let $\{\mathcal{M}_{i,j}\}_{j=1}^{T(i)} = \{\mathbf{p}_i, \mathbf{p}_{x_{i,j,1}}, \dots, \mathbf{p}_{x_{i,j,m-1}}\}_{j=1}^{T(i)}$ denote the sequence of samples $\mathcal{M}_{i,j} \subset \mathcal{P}^*$ containing point $\mathbf{p}_i \in \mathcal{P}$ and drawn by some sampler (e.g., the uniform one as in RANSAC) where m is the minimal sample size, \mathcal{P}^* is the power set of \mathcal{P} , and $x_{i,j,1}, \dots, x_{i,j,m-1} \in \mathbb{N}^+$ are indices, referring to points in \mathcal{P} . In each $\mathcal{M}_{i,j}$, the points are ordered with respect to their distances from \mathbf{p}_i and indices

j denote the order in which the samples were drawn. The objective is to find a strategy which draws samples consisting of points close to the i th one and, then, samples which contain data points farther from \mathbf{p}_i are drawn progressively.

Since the problem is quite similar to that of PROSAC, the same growth function can be used. Let us define set $\mathcal{S}_{i,k}$ to be the smallest ball centered on \mathbf{p}_i and containing its k nearest neighbours. Let $T_k(i)$ be the number of samples from $\{\mathcal{M}_{i,j}\}_{j=1}^{T(i)}$ which contains \mathbf{p}_i and the other points are from $\mathcal{S}_{i,k}$. For the expected number of $T_k(i)$, holds:

$$E(T_k(i) | T(i)) = T(i) \frac{\binom{k}{m-1}}{\binom{n-1}{m-1}} = T(i) \prod_{j=0}^{m-2} \frac{k-j}{n-1-j},$$

where n is the number of data points. In this case, ratio $E(T_{k+1}(i) | T(i)) / E(T_k(i) | T(i))$ does not depend on i and $E(T_{k+1}(i) | T(i))$ can be recursively defined as

$$E(T_{k+1}(i) | T(i)) = \frac{k+1}{k+2-m} E(T_k(i) | T(i)).$$

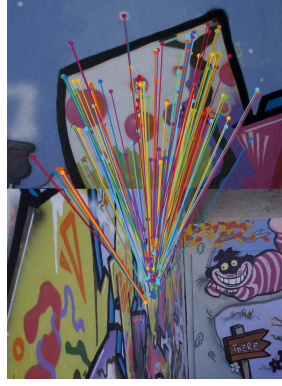
We approximate it by integer function $T'_{k+1} = T'_k + \lceil E(T_{k+1}(i) | T(i)) - E(T_k(i) | T(i)) \rceil$, where $T'_1 = 1$ for all i . Thus, T'_k is i -independent. Growth function $g(t) = \min\{k : T'_k \geq t\}$, i.e., for integer t , $g(t) = k$ where $T'_k = t$. Let t_i be the number of samples including \mathbf{p}_i . For set $\mathcal{S}_{i,g(t_i)}$, t_i is approx. the mean number of samples drawn from $\mathcal{S}_{i,g(t_i)}$ if the random sampler of RANSAC is used.

In the proposed P-NAPSAC sampler, neighbourhood $\mathcal{S}_{i,k}$ of \mathbf{p}_i grows if $g(t_i) = k$, i.e., the number of drawn samples containing the i th point is approximately equal to the mean number of the samples drawn from this neighbourhood by the random sampler.

The t_i th sample \mathcal{M}_{i,t_i} , containing \mathbf{p}_i , is $\mathcal{M}_{i,t_i} = \{\mathbf{p}_i, \mathbf{p}^*(g(t_i))\} \cup \mathcal{M}'_{i,t_i}$, where $\mathcal{M}'_{i,t_i} \subset \mathcal{S}_{i,g(t_i)-1}$ is a set of $|\mathcal{M}'_{i,t_i}| = m-2$ data points, excluding \mathbf{p}_i and $\mathbf{p}^*(g(t_i))$, randomly drawn from $\mathcal{S}_{i,g(t_i)-1}$. Point $\mathbf{p}^*(g(t_i))$ is the $g(t_i)$ -th nearest neighbour of point \mathbf{p}_i .

Growth of the hit number. Given point \mathbf{p}_i , the corresponding t_i is increased in two cases. First, $t_i \leftarrow t_i + 1$ when \mathbf{p}_i is selected to be the center of the hyper-sphere. Second, t_i is increased when \mathbf{p}_l is selected, the neighborhood of \mathbf{p}_l contains \mathbf{p}_i and, also, that of \mathbf{p}_i contains \mathbf{p}_l . Formally, let \mathbf{p}_l be selected as the center of the sphere ($l \neq i \wedge l \in [1, n]$). Let sample $\mathcal{M}_{l,j} = \{\mathbf{p}_l, \mathbf{p}_{x_{l,j,1}}, \dots, \mathbf{p}_{x_{l,j,m-1}}\}$ be selected randomly as the sample in the previously described way. If $i \in \{x_{l,j,1}, \dots, x_{l,j,m-1}\}$ (or equivalently, $\mathbf{p}_i \in \mathcal{M}_{l,j}$) and $\mathbf{p}_l \in \mathcal{S}_{i,g(t_i)}$ then t_i is increased by one.

The sampler (see Alg. 1) can be imagined as a PROSAC sampling defined for every i th point independently, where the sequence of samples for the i th point depends on its neighbors. After the initialization, the first main step is to select \mathbf{p}_i as the center of the sphere and update the corresponding t_i . Then a semi-random sample is drawn consisting of the selected \mathbf{p}_i , its k_i th nearest neighbour and $m-2$



(a) P-NAPSAC made 18 302 iterations in 0.49 secs. PROSAC made 84 831 in 1.76 secs. Scene "There".



(b) P-NAPSAC made 65 842 iterations in 0.84 secs. PROSAC made 99 913 in 1.28 secs. Scene "Vin".

Figure 2. Example image pairs from the EVD dataset for homography estimation. Inlier correspondences are marked by a line segment joining the corresponding points.

random points from \mathcal{S}_{i,k_i-1} (i.e., the points in the sphere around \mathbf{p}_i excluding the farthest one). Based on the random sample, the corresponding t values are updated. Finally, the implied model is estimated, and its quality is measured.

Relaxation of the termination criterion. We observed that, in practice, the termination criterion of RANSAC is conservative and not suitable for finding local structures early. The number of required iterations r of RANSAC is

$$r = \log(1 - \mu) / \log(1 - \eta^m), \quad (4)$$

where m is the size of a minimal sample, μ is the required confidence in the results and η is the inlier ratio. This criterion does not assume that the points of the sought model are spatially coherent, i.e., the probability of selecting an all-inlier sample is higher than η^m . Local structures typically have low inlier ratio. Thus, in the case of low inlier ratio, Eq. 4 leads to too many iterations even if the model is localized and is found early due to the localized sampling.

A simple way of terminating early is to relax the termination criterion. The number of iterations r' for finding a model with $\eta + \gamma$ inlier ratio is

$$r' = \log(1 - \mu) / \log(1 - (\eta + \gamma)^m), \quad (5)$$

where $\gamma \in [0, 1 - \eta]$ is a relaxation parameter.

Fast neighbourhood calculation. Determining the spatial relations of all points is a time consuming operation even by applying approximating algorithms, e.g., the Fast Approximated Nearest Neighbors method [17]. In the sampling of RANSAC-like methods, the primary objective is to find the best sample early and, thus, spending significant time initializing the sampler is not affordable. Therefore,

we propose a multi-layer grid for the neighborhood estimation which we describe for point correspondences. It can be straightforwardly modified considering different input data.

Suppose that we are given two images of size $w_l \times h_l$ ($l \in \{1, 2\}$) and a set of point correspondences $\{(\mathbf{p}_{i,1}, \mathbf{p}_{i,2})\}_{i=1}^n$, where $\mathbf{p}_{i,l} = [u_{i,l}, v_{i,l}]^T$. A 2D point correspondence can be considered as a point in a four-dimensional space. Therefore, the size of a cell in a four-dimensional grid \mathcal{G}_δ constrained by the sizes of the input image is $\frac{w_1}{\delta} \times \frac{h_1}{\delta} \times \frac{w_2}{\delta} \times \frac{h_2}{\delta}$, where δ is parameter determining the number of divisions along an axis. Function $\Sigma(\mathcal{G}_\delta, [u_{i,1}, v_{i,1}, u_{i,2}, v_{i,2}]^T)$ returns the set of correspondences which are in the same 4D cell as the i th one. Thus, $|\Sigma(\mathcal{G}_\delta, \dots)|$ is the cardinality of the neighborhood of a particular point. Having multiple layers means that we are given a sequence of δ s such that: $\delta_1 > \delta_2 > \dots > \delta_d \geq 1$. For each δ , the corresponding \mathcal{G}_{δ_k} grid is constructed. For the i th correspondence during its t_i th selection, the finest layer $\mathcal{G}_{\delta_{\max}}$ is selected which has enough points in the cell in which \mathbf{p}_i is stored. Parameter δ_{\max} is calculated as $\delta_{\max} := \max\{\delta_k : k \in [1, d] \wedge |\mathcal{S}_{i,g(t_i)-1}| \leq |\Sigma(\mathcal{G}_{\delta_k}, \dots)|\}$.

In P-NAPSAC, $d = 5$, $\delta_1 = 16$, $\delta_2 = 8$, $\delta_3 = 4$, $\delta_4 = 2$ and $\delta_5 = 1$. When using hash-maps and an appropriate hashing function, the implied computational complexity of the grid creation is $\mathcal{O}(n)$. For the search, it is $\mathcal{O}(1)$. Note that $\delta_5 = 1$ leads to a grid with a single cell and, therefore, does not require computation.

4. Experimental Results

In this section, we evaluate the accuracy and speed of the two proposed algorithms. First, we test MAGSAC++ on fundamental matrix and homography fitting on six publicly available real-world datasets. Second, we show that Progressive NAPSAC sampling leads to faster robust estimation than the state-of-the-art samplers. Note that *these contributions are orthogonal* and, therefore, can be used together to achieve high performance efficiently – by using MAGSAC++ with P-NAPSAC sampler.

4.1. Evaluating MAGSAC++

Fundamental matrix estimation was evaluated on the benchmark of [3]. The [3] benchmark includes: (1) the TUM dataset [25] consisting of videos of indoor scenes. Each video is of resolution 640×480 . (2) The KITTI dataset [6] consists of consecutive frames of a camera mounted to a moving vehicle. The images are of resolution 1226×370 . Both in KITTI and TUM, the image pairs are short-baseline. (3) The Tanks and Temples (T&T) dataset [10] provides images of real-world objects for image-based reconstruction and, thus, contains mostly wide-baseline pairs. The images are of size from 1080×1920 up to 1080×2048 . (4) The Community Photo Collection (CPC) dataset [30] con-

tains images of various sizes of landmarks collected from Flickr. In the benchmark, 1000 image pairs are selected randomly from each dataset. SIFT [12] correspondences are detected, filtered by the standard SNN ratio test [12] and, finally, used for estimating the epipolar geometry.

The compared methods are RANSAC [5], LMedS [22], MSAC [28], GC-RANSAC [1], MAGSAC [2], and MAGSAC++. All methods used P-NAPSAC sampling, preemptive model validation and degeneracy testing as proposed in USAC [21]. The confidence was set to 0.99. For each method and problem, we chose the threshold maximizing the accuracy. For homography fitting, it is as follows: MSAC and GC-RANSAC (5.0 pixels); RANSAC (3.0 pixels); MAGSAC and MAGSAC++ (σ_{\max} was set considering 50.0 pixels threshold). For fundamental matrix fitting, it is as follows: RANSAC, MSAC and GC-RANSAC (0.75 pixels); MAGSAC and MAGSAC++ (σ_{\max} which 5.0 pixels threshold implies). The used error metric is the symmetric geometric distance [31] (SGD) which compares two fundamental matrices by iteratively generating points on the borders of the images and, then, measuring their epipolar distances. All methods were in C++.

In Fig. 3, the cumulative distribution functions (CDF) of the SGD errors (horizontal) are shown. *MAGSAC++ is the most accurate robust estimator* on CPC, Tanks and Temples and TUM datasets since its curve is always higher than that of the other methods. In KITTI, the image pairs are subsequent frames of a camera mounted to a car, thus, having short baseline. These image pairs are therefore easy and all methods lead to similar accuracy.

In Table 1, the median errors (in pixels), the failure rates (in percentage) and processing times (in milliseconds) are reported. We report the median values to avoid being affected by the failures – which are also shown. A test is considered failure if the error of the estimated model is bigger than the 1% of the image diagonal. The best values are shown in red, the second best ones are in blue. For fundamental matrix fitting (first four datasets), *MAGSAC++ is the best method on three datasets* both in terms of median error and failure rate. On KITTI, all methods have similar accuracy – the difference between the accuracy of the least and most accurate ones is 0.3 pixel. There, MAGSAC++ is the fastest. On the tested datasets, MAGSAC++ is usually as fast as other robust estimators while leading to superior accuracy and failure rate. MAGSAC++ is *always faster* than MAGSAC, e.g., on KITTI by two orders of magnitude.

In the left plot of Fig. 5, the avg. \log_{10} errors over all datasets are plotted as the function of the inlier-outlier threshold. Both *MAGSAC and MAGSAC++ are significantly less sensitive to the threshold* than the other robust estimators. Note that the accuracy of LMedS is the same for all threshold values since it does not require an inlier-outlier threshold to be set.

Method / Dataset	Fundamental matrix (Fig. 3)									Homography (Fig. 4)								
	KITTI [6]			TUM [25]			T&T [10]			CPC [30]			Homogr [11]			EVD [11]		
	ϵ_{med}	λ	t	ϵ_{med}	λ	t	ϵ_{med}	λ	t	ϵ_{med}	λ	t	ϵ_{med}	λ	t	ϵ_{med}	λ	t
MAGSAC++	3.6	2.4	8	3.5	16.4	13	3.9	0.4	142	6.4	7.8	156	1.1	0.0	6	2.6	10.4	173
MAGSAC [2]	3.5	2.8	117	3.7	17.7	18	4.2	0.7	267	7.0	7.8	261	1.3	0.8	32	2.6	12.0	426
GC-RANSAC [1]	3.7	2.3	11	4.1	25.1	11	4.5	2.2	126	7.5	12.1	144	1.1	0.0	25	2.6	18.3	66
RANSAC [5]	3.8	2.7	9	5.4	22.1	11	6.3	2.6	133	16.9	29.5	151	1.1	0.0	26	4.0	26.1	68
LMedS [22]	3.6	2.7	11	4.3	23.9	12	4.9	1.1	166	10.7	17.8	187	1.5	12.5	31	89.9	60.0	82
MSAC [28]	3.8	2.6	10	5.5	36.2	11	7.0	2.2	133	16.5	33.8	153	1.1	0.0	24	3.2	23.7	64

Table 1. The median errors (ϵ_{med} ; in pixels), failure rates (λ ; in percentage) and average processing times (t , in milliseconds) are reported for each method (rows from 4th to 9th) on all tested problems (1st row) and datasets (2nd). The error of fundamental matrices is calculated from the ground truth matrix as the symmetric geometric distance [31] (SGD). For homographies, it is the RMSE re-projection error from ground truth inliers. A test is considered failure if the error is bigger than the 1% of the image diagonal. For each method, the inlier-outlier threshold was set to maximize the accuracy and the confidence to 0.99. The best values in each column are shown by red and the second best ones by blue. Note that all methods, excluding MAGSAC and MAGSAC++, finished with a final LS fitting on all inliers.

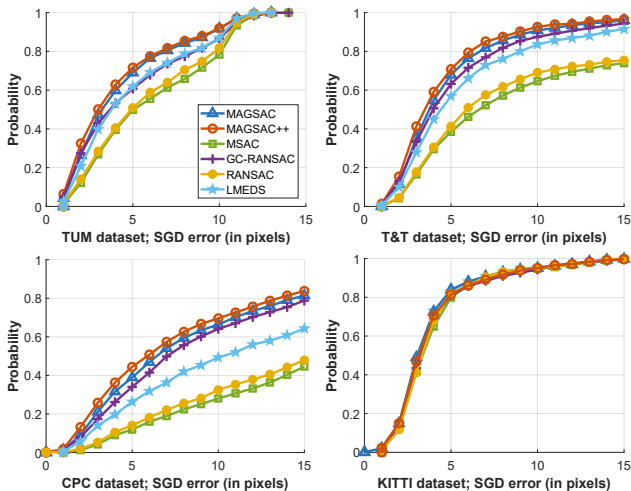


Figure 3. The cumulative distribution functions (CDF) of the SGD errors (horizontal axis) of the estimated fundamental matrices, on datasets CPC, T&T, KITTI and TUM. Being accurate is interpreted by a curve close to the top.

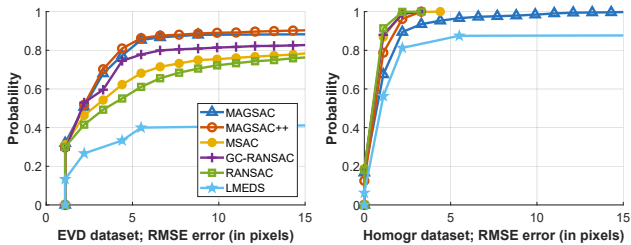


Figure 4. The cumulative distribution functions (CDF) of the RMSE re-projection errors (horizontal axis) of the estimated homographies on datasets EVD and homogr. Being accurate is interpreted by a curve close to the top.

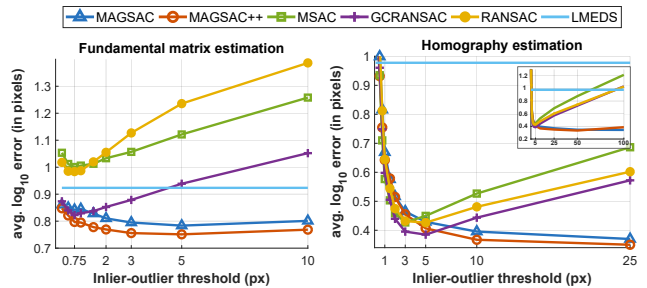


Figure 5. The average \log_{10} errors on the datasets for fundamental matrix (left; SGD error) and homography (right; RMSE re-projection error) fitting plotted as the function of the inlier-outlier threshold (in pixels). In the small plot inside the right one, the threshold goes up to 100 pixels.

For homography estimation, we downloaded homogr (16 pairs) and EVD (15 pairs) datasets [11]. They consist of image pairs of different sizes from 329×278 up to 1712×1712 with point correspondences provided. The homogr dataset contains mostly short baseline stereo images, whilst the pairs of EVD undergo an extreme view change, i.e., wide baseline or extreme zoom. In both datasets, inlier correspondences of the dominant planes are selected manually. All algorithms applied the normalized four-point algorithm [8] for homography estimation and were repeated 100 times on each image pair. To measure the quality of the estimated homographies, we used the RMSE re-projection error calculated from the provided ground truth inliers.

The CDFs of the errors are shown in Fig. 4. On EVD, the MAGSAC++ goes the highest – it is the most accurate method. On homogr, all methods but the original MAGSAC and LMedS have similar accuracy. The last two datasets in Table 1 report the median errors, failure rates and runtimes. On EVD, *MAGSAC++ failed the least often, while having*

the best median accuracy and being 2.5 times faster than MAGSAC. All of the faster methods fail to return the sought model significantly more often. On homog_r, MAGSAC++, GC-RANSAC, RANSAC and MSAC have similar results. MAGSAC++ is the fastest by almost an order of magnitude.

In the right plot of Fig. 5, the avg. \log_{10} errors are plotted as the function of the inlier-outlier threshold (in px). Both *MAGSAC* and *MAGSAC++* are significantly less sensitive to the threshold than the other robust estimators. In the small figure, inside the bigger one, the threshold value goes up to 100 pixels. For MAGSAC and MAGSAC++, parameter σ_{\max} was calculated from the threshold value.

In summary, the experiments showed that MAGSAC++ is more accurate on the tested problems and datasets than all the compared state-of-the-art robust estimators with being significantly faster than the original MAGSAC.

4.2. Evaluating Progressive NAPSAC

In this section, the proposed P-NAPSAC sampler is evaluated on homography and fundamental matrix fitting using the same datasets as in the previous sections. Every tested sampler is combined with MAGSAC++. The compared samplers are the uniform sampler of plain RANSAC [5], NAPSAC [20], PROSAC [4], and the proposed P-NAPSAC. Since both the proposed P-NAPSAC and NAPSAC assumes the inliers to be localized, they used the relaxed termination criterion with $\gamma = 0.1$. Thus, they terminate when the probability of finding a model which leads to at least 0.1 increment in the inlier ratio falls below a threshold. PROSAC used its original termination criterion and the quality function for sorting the correspondences.

Example image pairs are shown in Fig. 2. Inlier correspondences are marked by line segments joining the corresponding points. The numbers of iterations and processing times of PROSAC or P-NAPSAC samplers are reported in the captions. In both cases, P-NAPSAC leads to significantly fewer iterations than PROSAC. The results of the samplers, compared to P-NAPSAC and averaged over all datasets, are shown in Fig. 6a. The number of iterations and, thus, the processing time is the lowest for P-NAPSAC. It is approx. 1.6 times faster than the second best, i.e., PROSAC, while being similarly accurate with the same failure rate.

The relaxed termination criterion was tested by applying P-NAPSAC to all datasets using different γ values. We then measured how each property (i.e., error, failure rate, run-time, and iteration number) changes. Fig. 6b plots the average (over 100 runs on each scene) of the reported properties as the function of γ . The relative values are shown. Thus, for each test, the values are divided by the maximum. For instance, if P-NAPSAC draws 100 iterations when $\gamma = 0$, the number of iterations is divided by 100 for every other γ .

The error and failure ratio slowly increase from approximately 0.8 to 1.0. The trend seems to be close to linear.

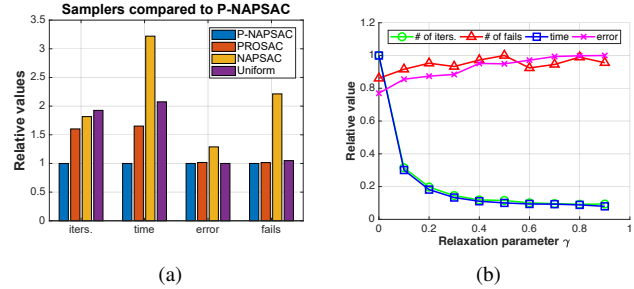


Figure 6. (a) Comparison of samplers to P-NAPSAC (blue bar; divided by the values of P-NAPSAC; all combined with MAGSAC++) on the datasets of Table 1. The reported properties are: the # of iterations; processing time; average error; and failure rate. (b) The relative (i.e., divided by the maximum) error, number of fails, processing time, and number of iterations are plotted as the function of the relaxation parameter γ (from Eq. 5) of the relaxed RANSAC termination criterion.

Simultaneously, the number of iterations and, thus, the processing time decrease dramatically. Around $\gamma = 0.1$ there is significant drop from 1.0 to 0.3. If $\gamma > 0.1$ both values decrease mildly. Therefore, selecting $\gamma = 0.1$ as the relaxation factor does not lead to noticeably worse results but speeds up the procedure significantly.

5. Conclusion

In the paper, two contributions were made. First, we formulate a novel marginalization procedure as an iteratively re-weighted least-squares approach and we introduce a new model quality (scoring) function that does not require the inlier-outlier decision. Second, we propose a new sampler, Progressive NAPSAC, for RANSAC-like robust estimators. Reflecting the fact that nearby points often originate from the same model in real-world data, P-NAPSAC finds local structures earlier than global samplers. The progressive transition from local to global sampling does not suffer from the weaknesses of purely localized samplers.

The two orthogonal improvements are combined with the "bells and whistles" of USAC [21], e.g., pre-emptive verification, degeneracy testing. On six publicly available real-world datasets for homography and fundamental matrix fitting, MAGSAC++ produces results superior to the state-of-the-art robust methods. It is faster, more geometrically accurate and fails less often.

Acknowledgement

This work was supported by the Czech Science Foundation grant GA18-05360S, Czech Technical University student grant SGS17/185/OHK3/3T/13, the Ministry of Education OP VVV project CZ.02.1.01/0.0/0.0/16 019/0000765 Research Center for Informatics, and by grant 2018-1.2.1-NKP-00008.

References

- [1] D. Barath and J. Matas. Graph-Cut RANSAC. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6733–6741, 2018. <https://github.com/danini/graph-cut-ransac>. 1, 6, 7
- [2] D. Barath, J. Noskova, and J. Matas. MAGSAC: marginalizing sample consensus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. <https://github.com/danini/magsac>. 1, 2, 6, 7
- [3] J.-W. Bian, Y.-H. Wu, J. Zhao, Y. Liu, L. Zhang, M.-M. Cheng, and I. Reid. An evaluation of feature matchers for fundamental matrix estimation. *arXiv preprint arXiv:1908.09474*, 2019. <https://jwbian.net/fm-bench>. 6
- [4] O. Chum and J. Matas. Matching with PROSAC-progressive sample consensus. In *Computer Vision and Pattern Recognition*. IEEE, 2005. 2, 8
- [5] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 1, 6, 7, 8
- [6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 6, 7
- [7] D. Ghosh and N. Kaabouch. A survey on image mosaicking techniques. *Journal of Visual Communication and Image Representation*, 2016. 1
- [8] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 7
- [9] H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *International Journal of Computer Vision*, 2012. 1
- [10] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and Temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):78, 2017. 1, 6, 7
- [11] K. Lebeda, J. Matas, and O. Chum. Fixing the locally optimized RANSAC. In *British Machine Vision Conference*. Citeseer, 2012. <http://cmp.felk.cvut.cz/wbs/>. 1, 7
- [12] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer vision*. IEEE, 1999. 6
- [13] R. A. Maronna, R. D. Martin, V. J. Yohai, and M. Salibián-Barrera. *Robust statistics: theory and methods (with R)*. John Wiley & Sons, 2019. 3
- [14] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 2004. 1
- [15] D. Mishkin, J. Matas, and M. Perdoch. MODS: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 2015. 1
- [16] L. Moisan, P. Moulon, and P. Monasse. Automatic homographic registration of a pair of images, with a contrario elimination of outliers. *Image Processing On Line*, 2:56–73, 2012. 2
- [17] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *International Conference on Computer Vision Theory and Applications*, 2009. 5
- [18] T. T. Pham, T.-J. Chin, K. Schindler, and D. Suter. Interacting geometric priors for robust multimodel fitting. *Transactions on Image Processing*, 2014. 1
- [19] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *International Conference on Computer Vision*. IEEE, 1998. 1
- [20] D. R. Myatt, P. Torr, S. Nasuto, J. Bishop, and R. Craddock. NAPSAC: High noise, high dimensional robust estimation - it's in the bag. In *British Machine Vision Conference*, 2002. 2, 4, 8
- [21] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm. USAC: a universal framework for random sample consensus. *Transactions on Pattern Analysis and Machine Intelligence*, 2013. 6, 8
- [22] P. J. Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984. 1, 6, 7
- [23] C. Sminchisescu, D. Metaxas, and S. Dickinson. Incremental model-based estimation using geometric constraints. *Pattern Analysis and Machine Intelligence*, 2005. 1
- [24] C. V. Stewart. Minpran: A new robust estimator for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):925–938, 1995. 2
- [25] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012. 6, 7
- [26] P. H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–61, 2002. 1
- [27] P. H. S. Torr and D. W. Murray. Outlier detection and motion segmentation. In *Optical Tools for Manufacturing and Advanced Automation*. International Society for Optics and Photonics, 1993. 1
- [28] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 2000. 1, 6, 7
- [29] P. H. S. Torr, A. Zisserman, and S. J. Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *Computer Vision and Image Understanding*, 1998. 1
- [30] K. Wilson and N. Snavely. Robust global translations with 1DSfM. In *European Conference on Computer Vision*, pages 61–75. Springer, 2014. 1, 6, 7
- [31] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International journal of computer vision*, 27(2):161–195, 1998. 6, 7
- [32] M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multi-RANSAC algorithm and its application to detect planar homographies. In *International Conference on Image Processing*. IEEE, 2005. 1