# Graph-Cut RANSAC: Local Optimization on Spatially Coherent Structures

Daniel Barath and Jiri Matas

**Abstract**—We propose Graph-Cut RANSAC, GC-RANSAC in short, a new robust geometric model estimation method where the local optimization step is formulated as energy minimization with binary labeling, applying the graph-cut algorithm to select inliers. The minimized energy reflects the assumption that geometric data often form spatially coherent structures – it includes both a unary component representing point-to-model residuals and a binary term promoting spatially coherent inlier-outlier labelling of neighboring points. The proposed local optimization step is conceptually simple, easy to implement, efficient with a globally optimal inlier selection given the model parameters. Graph-Cut RANSAC, equipped with "the bells and whistles" of USAC and MAGSAC++, was tested on a range of problems using a number of publicly available datasets for homography, 6D object pose, fundamental and essential matrix estimation. It is more geometrically accurate than state-of-the-art robust estimators, fails less often and runs faster or with speed similar to less accurate alternatives. The source code is available at https://github.com/danini/graph-cut-ransac.

**Index Terms**—Robust model estimation, RANSAC, local optimization, spatial coherence, energy minimization, graph-cut

## 1 Introduction

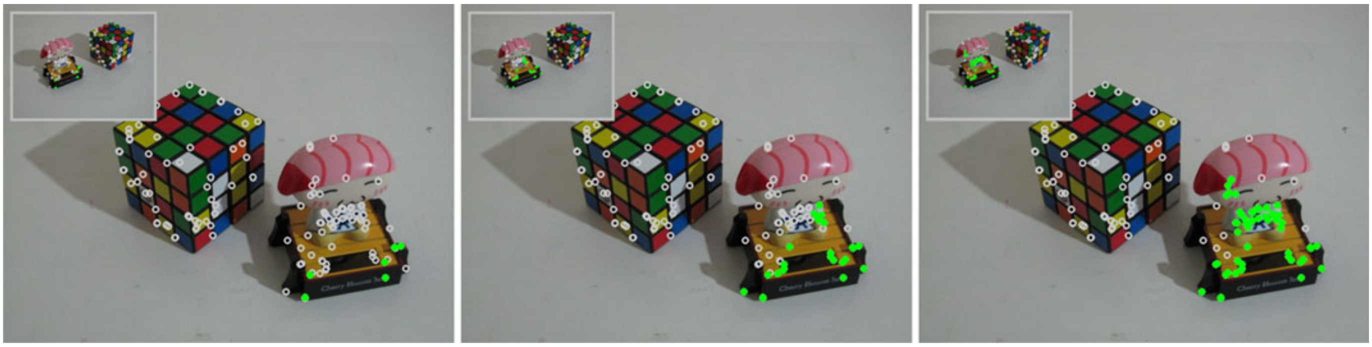
The RANdom SAmple Consensus (RANSAC) algorithm proposed by Fischler and Bolles [1] in 1981 has become the most widely used robust estimator in computer vision. RANSAC and its variants have been successfully applied to a wide range of vision tasks, e.g., motion segmentation [2], short baseline stereo [2], [3], wide baseline matching [4], [5], [6], pose-graph initialization for structure-from-motion pipelines [7], [8], detection of geometric primitives [9], image mosaicing [10], and to perform [11] or initialize multi-model fitting algorithms [12], [13]. In brief, RANSAC repeatedly selects minimal random subsets of the input data points and fits a model, e.g., a line to two 2D points, a fundamental matrix to seven 2D point correspondences, or a 6D pose to three 2D-3D correspondences. Next, the quality of the 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. In this paper, we propose a new local optimization technique for RANSAC considering the fact that real-world data often form spatially coherent structures.

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. NAPSAC [14] assumes that inliers are spatially coherent and therefore it draws samples from a hypersphere 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 [15] assumes that inliers are often "similar" to each other and, therefore, data points can be separated into groups according to their similarities. PROSAC [16] exploits an a priori predicted inlier probability rank of each point and starts the sampling with the most promising ones. Progressively, samples that are less likely to lead to the sought model are drawn. P-NAPSAC [17] merges the advantages of local and global sampling by drawing samples from gradually growing neighborhoods. Gradually, the algorithm changes from the fully localized NAPSAC to the global PROSAC sampling. NG-RANSAC [18] predicts the inlier probability of each point via deep learning.

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 [19], 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 [20]. Given a model to be scored, a randomly selected subset of $d$ points is evaluated. If the inlier ratio within this 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 [21], [22], an optimal randomized model verification

- *Daniel Barath is with the Visual Recognition Group, Department of Cybernetics, Czech Technical University, 166 36 Prague, Czechia, and with the Machine Perception Research Laboratory, SZTAKI, 1111 Budapest, Hungary. E-mail: barath.daniel@sztaki.hu.*
- *Jiri Matas is with the Visual Recognition Group, Department of Cybernetics, Czech Technical University, 166 36 Prague, Czechia. E-mail: matas@cmp.felk.cvut.cz.*

*Manuscript received 13 May 2020; revised 31 Mar. 2021; accepted 31 Mar. 2021. Date of publication 8 Apr. 2021; date of current version 4 Aug. 2022. (Corresponding author: Daniel Barath.) Recommended for acceptance by A. Geiger. Digital Object Identifier no. 10.1109/TPAMI.2021.3071812*



(a) Minimal sample initializing a rigid motion. (b) Inliers by standard thresholding. (c) Inliers by labeling Eq. 8.

Fig. 1. Inlier correspondences (green dots) of a rigid motion model, i.e., a fundamental matrix, initialized by a minimal sample (a). Inliers obtained by (b) standard thresholding of the residual; (c) the proposed graph-cut-based selection considering spatial coherence. All other points are marked by gray circles. The graph-cut-based selection (c) returns more inliers compared to the traditional thresholding (b).

strategy was described. The test is based on Wald's theory of sequential testing [23]. 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.

To improve the accuracy by better modelling the noise in the data, different model quality calculation techniques have been investigated. For instance, MLESAC [24] 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 [25], 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.

There are also methods to reduce the dependency on the user-defined inlier-outlier threshold. For example, MINPRAN [26] assumes that the outliers are distributed uniformly and finds the model where the inliers are least likely to have occurred randomly. Moisan *et al.* [27] proposed a contrario RANSAC, selecting the most likely noise scale for each model candidate. Barath *et al.* [17], [28] proposed the Marginalizing Sample Consensus method (MAGSAC) and its recent improvement (MAGSAC++) marginalizing over the noise scale $\sigma$ to eliminate the threshold from the model quality calculation.

Observing that RANSAC requires in practice more samples than what theory predicts, Chum *et al.* [29] 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. In the original LO-RANSAC paper [29], the local optimization is implemented as an iterated least-squares model re-fitting with a progressively shrinking inlier-outlier threshold inside an inner RANSAC applied only to the inliers of the current model. In the reported experiments, LO-RANSAC is superior to plain RANSAC both in terms of geometric accuracy and number of iterations. It is shown that the number of local optimizations is close to the logarithm of the iteration number and, therefore, it usually does not yield a significant overhead in the processing time. However, Lebeda *et al.* [30] 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 [30], 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.

In this paper, we propose a new local optimization procedure considering that in real-world applications, data points often form spatially coherent structures. In the large body of RANSAC-related literature, the inlier-outlier decision has always been a function of the point-to-model residual, calculated individually for each data point. In practice, both inlier and outlier poinn the remaining casets often are spatially coherent and, therefore, a point near to an outlier or inlier is likely to be, respectively, an outlier or inlier. Spatial coherence, described by, e.g., the Potts model [31], has been exploited in a number of vision problems, for instance, in segmentation [32], multi-model fitting [12], [13], [33], [34], [35], [36] or sampling [14], [17] in RANSAC-like techniques. Directly formalizing the model verification in RANSAC as a graph-cut problem such that it considers spatial coherence is computationally prohibitive. However, when applied as the local optimization step, as in [29], just to each *so-far-the-best* model, the number of graph-cuts is only the logarithm of the number of sampled and verified models, and can be performed efficiently.

The proposed Graph-Cut RANSAC, GC-RANSAC in short, is a locally optimized RANSAC alternating graph-cut and model fitting as the local optimization step. It is superior to original LO-RANSAC in a number of aspects. The *contributions* are:

1. GC-RANSAC is capable of exploiting spatial coherence of points. See Fig. 1 for example. The LO step is conceptually simple, easy to implement, its inlier selection is a globally optimal and efficient graph-cut with only a few intuitive and learnable parameters unlike the ad hoc, iterative and complex LO steps [29].
2. We propose a new energy term which models the spatial coherence of geometric data. Experiments

show that the proposed term is more suitable for geometric robust model estimation than the traditionally used Potts model [31].

3. We combine GC-RANSAC with the bells and whistles of USAC [37] and MAGSAC++ [17]. It is shown experimentally that the proposed algorithm is superior to the state-of-the-art LO-RANSAC variants, included in USAC [37], in terms of accuracy and failure ratio on a wide range of vision problems (i.e., homography, essential and fundamental matrix, and 6D pose estimation).

*Remark*. Isack's and Boykov's PEARL [12] was the first method to introduce spatial coherence to geometric model fitting. However, PEARL cannot be directly used for the problems solved by RANSAC, since the user has to manually set the number of hypotheses tested in the worst-case, i.e., the lowest inlier ratio possible. The $\alpha$-expansion step executes, in the first iteration of PEARL, the graph-cut as many times as the number of hypotheses tested. The number is calculated from the worst-case scenario and is typically orders of magnitude higher than the number of iterations which the adaptive RANSAC termination criterion determines. Moreover, in GC-RANSAC, applying the local optimization to only the *so-far-the-best* models ensures that the graph-cut runs only very few times, paying only a small penalty.

A preliminary version of the GC-RANSAC algorithm was published at CVPR 2018 [38]. This paper extends and improves it by (i) proposing a new spatial coherence model, (ii) adding the USAC components and MAGSAC++ scoring, (iii) and providing a number of new experiments on homography, fundamental matrix, relative and 6D object pose estimation.

## 2 RANSAC Verification Reformulated

The inlier selection of RANSAC is formulated as an energy minimization problem. The novel formulation allows to include additional constraints when selecting the inliers of a given model.

### 2.1 RANSAC as Energy Minimization

To facilitate understanding of the connection to energy minimization, we start by reformulating the original top-hat loss function of RANSAC, see Fig. 2. Then continuous loss functions, e.g., truncated $L_2$, will be considered.

Suppose that we are given a set $\mathcal{P} \subseteq \mathbb{R}^{d_p}$ ($d_p > 0$) of $n$ points and a model represented by parameter vector $\theta \in \mathbb{R}^{d_m}$ ($d_m > 0$), where, respectively, $d_p$ and $d_m$ are the dimensions of a data point and the model. The residual function measuring the point-to-model assignment cost is $\phi : \mathcal{P} \times \mathbb{R}^{d_m} \rightarrow \mathbb{R}^+$. For the standard RANSAC scheme which applies a top-hat fitness function (0 – close, 1 – far), the implied unary energy is as follows: $E_{\{0;1\}}(L) = \sum_{p \in \mathcal{P}} ||L_p||_{\{0;1\}}$, where

$$||L_p||_{\{0;1\}} = \begin{cases} 0 & \text{if } (L_p = 0 \wedge \phi(p,\theta) \leq \epsilon) \vee \\ & \quad (L_p = 1 \wedge \phi(p,\theta) > \epsilon) \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

Parameter $L \in \{0,1\}^n$ is a labeling, ignored in standard RANSAC, $L_p \in L$ is the label of point $p \in \mathcal{P}$, and $\epsilon \in \mathbb{R}^+$ is the user-defined inlier-outlier threshold. Labels 0 and 1 are the inlier and outlier labels, respectively. Solving problem $L^* = \arg\min_L E_{\{0;1\}}(L)$ leads exactly to the RANSAC solution since $E_{\{0;1\}}$ does not penalize only two cases: (i) when $p$ is labeled inlier and it is closer to the model than the threshold, or (ii) when $p$ is labeled outlier and it is farther from the model than $\epsilon$. This is exactly what RANSAC does when selecting the inliers.

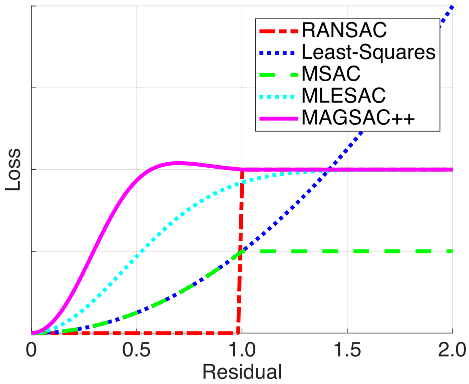


Fig. 2. Example loss functions used for robust model fitting – RANSAC [1], MSAC [25], MLESAC [24], MAGSAC++ [17].

A number of papers discussed [17], [24], [25], [28] the replacement of the $\{0,1\}$ loss with some continuous function $f$, e.g., the truncated $L_2$ loss of MSAC [25], to improve the estimation accuracy. Considering a general robust loss function, the energy term is written as follows: $E_f(L) = \sum_{p \in \mathcal{P}} f(L_p, p)$. For example, when using the MSAC-like truncated $L_2$ loss, $f_{\text{MSAC}}(L_p, p)$ becomes the following:

$$f_{\text{MSAC}}(L_p, p) = \begin{cases} \frac{\phi^2(p,\theta)}{\epsilon^2} & \text{if } (L_p = 0 \wedge \phi(p,\theta) \leq \epsilon), \\ 0 & \text{if } (L_p = 1 \wedge \phi(p,\theta) > \epsilon), \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

### 2.2 MAGSAC++ Loss

To use the state-of-the-art in robust model fitting, we consider the loss function of MAGSAC++ [17] which was designed in a way such that it does not require a strict inlier-outlier decision. The loss function proposed for MAGSAC++ is as follows: $g(\theta, \mathcal{P}) = \sum_{p \in \mathcal{P}} \rho(\phi(p, \theta))$, where function

$$\rho(r) = \int_0^r x w(x) \mathrm{d}x \quad \text{for } r \in [0, +\infty). \tag{3}$$

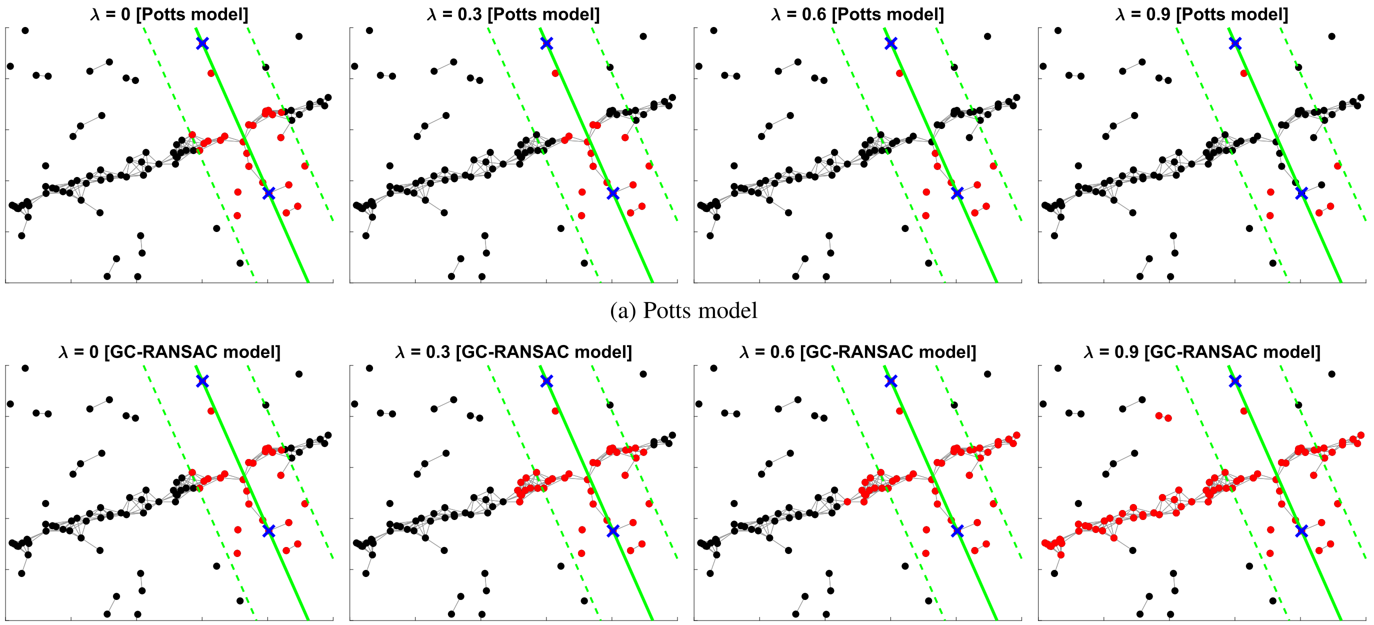
For $0 \leq r \leq k\sigma_{\max}$

$$\rho(r) = \frac{1}{\sigma_{\max}} C(d_p) 2^{\frac{d_p+1}{2}} [\frac{\sigma^2_{\max}}{2} \gamma(\frac{d_p+1}{2}, \frac{r^2}{2\sigma^2_{\max}}) + \\ \frac{r^2}{4} (\Gamma(\frac{d_p-1}{2}, \frac{r^2}{2\sigma^2_{\max}}) - \Gamma(\frac{d_p-1}{2}, \frac{k^2}{2}))],$$

where $\sigma_{\max}$ is a user-defined maximum noise scale, constant $C(d_p) = (2^{d_p/2}\Gamma(d_p/2))^{-1}$ and, for $a > 0$

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

is the gamma function, $d_p$ is the dimension of euclidean space in which the residuals are calculated and $\tau(\sigma)$ is set to



(a) Potts model

(b) GC-RANSAC model

Fig. 3. The effect of spatial coherence weight $\lambda$ on the inlier selection of a 2D line. The inliers (red points) of a line (green) initialized by a minimal sample (blue crosses) are shown. The top row shows the results of a single graph-cut run using different values for $\lambda$ when the Potts model (5) is applied. The bottom one shows the labeling results of a single graph-cut run when using the proposed spatial coherence model (6). The inlier-outlier threshold is shown by green dashed lines. The edges of the neighborhood graph are grey line segments.

a high quantile (e.g., 0.99) of the non-trimmed distribution. For $r > k\sigma_{\max}$

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

where $\gamma(a, x) = \int_0^x t^{a-1} \exp(-t) \mathrm{d}t$ is the lower incomplete gamma function. Weight $w(r)$ in (3) can be calculated efficiently by storing the values of the complete and incomplete gamma functions in a lookup table.

The loss implied by MAGSAC++ given a binary labeling is

$$f_{\mathrm{M++}}(L_p, p) = \begin{cases} g(\theta, p) & \text{if } (L_p = 0 \wedge \phi(p, \theta) \leq \epsilon_{\max}), \\ 0 & \text{if } (L_p = 1 \wedge \phi(p, \theta) > \epsilon_{\max}), \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

where $\epsilon_{\max}$ is the max. threshold which noise scale $\sigma_{\max}$ implies.

### 2.3 Spatial Coherence in RANSAC

In geometric model fitting, real-world data often form spatially coherent structures. This observation inspired a number of approaches, e.g., for sampling [14], [17] in robust methods or multi-model fitting techniques [12], [13], [34], [36]. To the best of our knowledge, there has been no attempt to exploit this property in the local optimization step of RANSAC.

Due to formalizing the inlier selection as an energy minimization via a binary labeling, additional energy terms can be straightforwardly considered. The problem is still *solvable efficiently and globally via the graph-cut algorithm*. To model the point-to-point proximity in the energy, the Potts model [31] usually is a justifiable choice. It is written as follows:

$$E_{\mathrm{Potts}}(L) = \sum_{(p,q) \in \mathcal{E}} \begin{cases} 1 & \text{if } L_p \neq L_q, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where $(p, q) \in \mathcal{E}$ is an edge connecting points $p$ and $q$ in a pre-calculated neighborhood graph $\mathcal{A} = (\mathcal{P}, \mathcal{E})$. When minimizing energy $E_{\mathrm{Potts}}(L)$, the neighboring points are encouraged to have the same label, formalizing the assumption that close points likely belong to the same model.

In our experiments, we saw that the Potts model fails to act as expected, i.e., to spread the inlier label along a structure. An example line fitting is shown in Fig. 3a. Each column shows the results of the binary labeling using different weighting $\lambda \in [0, 1]$ for the spatial coherence term. Due to the outliers being considered similarly structured as the inliers, and the model, the 2D line, being too inaccurate to select the sought inliers, the spatial coherence term forces all points in the structure to be outliers even if their point-to-model residuals are small, i.e., they are close to the line. Other examples are in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3071812.

The expected behaviour is to label all points which are closer than the threshold inliers and, also, points which are in the same spatial structure as the points close to the model. We achieve this behaviour by breaking condition $L_p = L_q$ of (5) down to two cases. The property, of the Potts model, of not penalizing two neighboring points $p, q$ if they both are inliers, $L_p = L_q = 0$, should still be kept. The $L_p = L_q = 1$ case, i.e., when both points are outliers, should depend on the point-to-model residual. Otherwise, the term may force points with small residuals but in the neighborhood of outliers to be labeled outlier. This can be seen in last plot of Fig. 3a, where points close to the line are labeled outliers due to being in a structure consisting mostly of outliers.

The proposed spatial coherence term which fixes the mentioned issues of the Potts model is as follows:

$$E_{\mathrm{GC}}(L) = \sum_{(p,q)\in\mathcal{E}} \begin{cases} 1 & \text{if } L_p \neq L_q, \\ 0 & \text{if } L_p = L_q = 0, \\ 1 - \frac{f(0,p)+f(0,q)}{2} & \text{if } L_p = L_q = 1. \end{cases} \tag{6}$$

When using $E_{\mathrm{GC}}$, the points closer than the threshold are penalized for jointly being labelled outliers. Farther than the inlier-outlier threshold, only the $L_p \neq L_q$ case is penalized, thus, leading to the same effect as the Potts model. This can be imagined as a "bumpy" non-uniform inlier-outlier threshold.

The sub-modularity requires $e_{pq}(0,0) + e_{pq}(1,1) \leq e_{pq}(1,0) + e_{pq}(0,1)$ to hold [39]. For (6), this inequality becomes

$$f(0,p) + f(0,q) \geq -2,$$

since $e_{pq}(0,0) = 0$ and $e_{pq}(1,0) = e_{pq}(0,1) = 1$. The condition holds if $f(0,\cdot) \in [-1,\infty)$, where operator $\cdot$ refers to either $p$ or $q$. This property depends solely on function $g$ from (4). When $g(\theta,\cdot) \in [-1,\infty)$ is true, $f(0,\cdot) \in [-1,\infty)$ holds as well. Both for MSAC and MAGSAC++, function $g(\theta,\cdot) \in [0,1]$. Thus, the implied energy term $E_{\mathrm{GC}}$ is sub-modular.

The bottom row of Fig. 3 shows the effect of the proposed term with different weights, $\lambda \in \{0, 0.3, 0.6, 0.9\}$. Optimizing $E_{\mathrm{GC}}$ leads to the desired effect – the inlier label is spread along the spatial structure while points close to the model do not get affected by the surrounding outliers. Note that this spatial coherence model leads to accurate results on the geometric problems investigated in this paper. However, if different assumptions hold, the energy can be straightforwardly modified and used within GC-RANSAC.

### 2.4 Graph-Cut Energy

The energy $E(L)$ minimized in the proposed Graph-Cut RANSAC is a linear combination of the data (unary) and spatial coherence (pair-wise) terms

$$E(L) = (1-\lambda)E_{\mathrm{M++}}(L) + \lambda E_{\mathrm{GC}}(L), \tag{7}$$

where $\lambda \in \mathbb{R}$ is a parameter balancing the terms. The globally optimal labeling $L^* = \arg\min_L E(L)$ can easily be determined in polynomial time using the graph-cut algorithm [40].

To balance between the energy terms, it is important to have each term normalized. It can be easily seen that $E_{\mathrm{M++}} \leq n$ since robust loss $f_{\mathrm{M++}}(L_p, p) \leq 1$ for each data point $p$. To ensure that $E_{\mathrm{GC}}(L)$ has the same scale, it has to be divided by the number of edges in the neighborhood graph and multiplied by $n$. Therefore, let us define the energy to minimize as follows:

$$\widehat{E}(L) = (1-\lambda)E_{\mathrm{M++}}(L) + \lambda \frac{n}{|\mathcal{E}|} E_{\mathrm{GC}}(L), \tag{8}$$

where $|\mathcal{E}|$ is the number of edges in $\mathcal{A}$. It can be easily seen that $E_{\mathrm{GC}}$ is sub-modular and, thus, $\widehat{E}$ also is.

## 3 Graph-Cut RANSAC (GC-RANSAC)

In this section, the described energy minimization-based inlier selection is used for proposing a new locally optimized RANSAC. Benefiting from the proposed approach, the LO step is conceptually simpler and cleaner than that of the original LO-RANSAC.

### 3.1 Energy-Based Labeling

The construction of problem graph $G$, which is fed into the graph-cut procedure, using unary and pair-wise terms Eqs. (4), (6) is shown in Algorithm 1. Functions AddTerm1 and AddTerm2 add, respectively, unary (4) and binary (6) costs to the problem graph. Such graph construction procedure is covered in depth in [39] (Section 4). The graph-cut algorithm is applied to $G$ determining the globally optimal labeling $L^*$ which considers the spatial coherence of the points and their point-to-model residuals given the current *so-far-the-best* model.

**Algorithm 1.** Problem Graph Construction

```
Input: $\mathcal{P}$ – data points, $\mathcal{A}$ – neighborhood-graph
       $\theta$ – model parameters, $\lambda$ – weight
Output: $G$ – problem graph;
$G \leftarrow$ EmptyGraph().                                ▷ Initialize the problem graph
for $p \in \mathcal{P}$ do                                   ▷ Unary term (4)
    $c_0 \leftarrow (1-\lambda)\rho(\phi(p,\theta))$.                    ▷ Loss of $p$ being outlier
    $c_1 \leftarrow (1-\lambda)(1-\rho(\phi(p,\theta),\epsilon))$.          ▷ Loss of $p$ being inlier
    $G \leftarrow$ AddTerm1($G, p, c_0, c_1$).
for $(p,q) \in \mathcal{A}$ do                               ▷ Binary term (6)
    $c_{01}, c_{10} \leftarrow \lambda, \lambda$.              ▷ Loss of $p, q$ with different labels.
    $c_{00} \leftarrow 0$.                                   ▷ Loss: $p, q$ being inliers.
    $c_{11} \leftarrow \frac{\lambda}{2}\sum_{s\in\{p,q\}}\rho(\phi(s,\theta))$          ▷ Loss: $p, q$ being outliers.
    $G \leftarrow$ AddTerm2($G, p, q, c_{00}, c_{01}, c_{10}, c_{11}$).
```

### 3.2 Graph-Cut in Local Optimization

The original LO step of LO-RANSAC consists of an inner RANSAC, applied locally to the inliers of the current best model, and an iterative model refitting, which uses inliers selected in each step by a progressively shrinking inlier-outlier threshold.

In the proposed GC-RANSAC algorithm, the inner RANSAC is a necessary step. The reason is that the least-squares (LS) fitting, which is applied to all inliers, minimizes the point-to-model residuals, i.e., the unary term. Minimizing this loss on points which are labeled inliers solely due to being in a spatial structure leads to inaccurate results in most of the cases. An intuitive example is shown in the right plot of Fig. 3b, where the sought inliers are found, but, also, points which are outliers of the ground truth model are labeled inliers. In this case, applying LS fitting fails to return the sought model parameters since LS is not robust and, thus, is extremely sensitive to outlying points. Instead, we apply an inner RANSAC to the points labeled inliers. In this case, the configuration of the last plot of Fig. 3b leads to an inner RANSAC applied to a point set with a very high inlier ratio. Consequently, the sought model is found easily in a few iterations.

Each step of the inner RANSAC selects a $7m$-sized sample from the points labeled inliers, where $m$ is the size of a minimal sample, e.g., $m = 4$ for homographies. Parameter

$7m$ was proposed in [30] and works well in our experiments. The LS fitting is always applied to points which are inliers due to the unary term, i.e., their point-to-model residuals are smaller than the inlier-outlier threshold. A detailed explanation of the steps of the proposed local optimization is written in Algorithm 2. Function `ShouldTerminate` is either a fixed iteration number or the standard RANSAC termination criterion. In the experiments, we used a fixed iteration number set to 20 to achieve fast performance.

**Algorithm 2.** GC-RANSAC Local Optimization

```
Input: $\mathcal{P}$ – data points, $L^*$ – labeling,
    $q^*$ – quality, $\theta^*$ – model;
Output: $L^*_{\mathrm{GC}}$ – labeling, $w^*_{\mathrm{GC}}$ – support, $\theta^*_{\mathrm{GC}}$ – model;
$q^*_{\mathrm{GC}}, L^*_{\mathrm{GC}}, \theta^*_{\mathrm{GC}} \leftarrow q^*, L^*, \theta^*$.
terminate, $\leftarrow$ false.
while $\neg$ terminate do
    $G \leftarrow$ ConstructGraph($\mathcal{P}, \mathcal{A}, \theta^*_{\mathrm{GC}}, \lambda$).    ▷ Algorithm 1
    $L \leftarrow$ GraphCut($G$)    ▷ Labeling minimizing (8)
    $\theta, \widehat{L}, q \leftarrow$ RANSAC($L$).    ▷ Inner RANSAC
    if $q > q^*_{\mathrm{GC}}$ then    ▷ If the found model is the new best
        $q^*_{\mathrm{GC}}, \theta^*_{\mathrm{GC}}, L^*_{\mathrm{GC}} \leftarrow q, \theta, \widehat{L}$.    ▷ Update the best model
    else
        terminate $\leftarrow$ true.
    $i \leftarrow i + 1$.
    if ShouldTerminate($\theta^*_{\mathrm{GC}}$, $L^*_{\mathrm{GC}}$, $i$) then
        terminate $\leftarrow$ true.
```

### 3.3 GC-RANSAC

The Graph-Cut RANSAC algorithm is shown in Algorithm 3 in depth. To achieve state-of-the-art results, we combine the proposed graph-cut-based local optimization with the components discussed in USAC [37]. We consider four popular vision problems, i.e., fundamental matrix, homography, 6D object pose (i.e., the P*n*P problem), 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. For 6D object pose estimation, samples are not used where the area of the triangle formed by the three selected points is smaller than a predefined threshold.
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*. 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 and 6D object pose estimation, improper rotation matrices [42], i.e., the ones with negative determinant, are rejected.
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]. For 6D object pose estimation, we assume that the object is in front of the camera and, thus, coordinate $z$ of the translation must be positive.
5. *Sampling*. We use the PROSAC sampler [16]. It requires an a priori determined ordering of the input data points. For point correspondence-based methods, we used the scoring coming from the standard SNN ratio-test [44]. For 6D object pose estimation, the points are ordered by their confidence values provided by deep-learning [45] in the used datasets.
6. *Preemptive model verification*. We use the Sequential Probability Ratio Test [22] (SPRT) to interrupt the model verification if the probability of being better than the current so-far-the-best model falls below a threshold.
7. *Scoring*. We use the scoring of MAGSAC++ [17] to calculate the model quality. Even though MAGSAC ++ does not require a single inlier-outlier threshold, the other components of the algorithm (e.g., local optimization, SPRT, DEGENSAC) do. Therefore, we set the upper bound of the threshold in MAGSAC++ to be $\epsilon_{\max} = 10\epsilon$, where $\epsilon$ is the manually set inlier-outlier threshold.
8. *Final model polishing*. The algorithm finishes with an iteratively re-weighted least-squares model refitting on all inlier points for all problems to polish the final model parameters.

**Algorithm 3.** The GC-RANSAC Algorithm

```
Input: $\mathcal{P}$ – data points; $\epsilon$ – inlier-outlier threshold
    $\mu$ – confidence;
Output: $\theta$ - model parameters; $L$ – labeling
$q^*, \mathcal{A} \leftarrow 0$, BuildNeighborhoodGraph($\mathcal{P}$).
while $\neg$ Terminate() do
    $S \leftarrow$ Sample($\mathcal{P}$).    ▷ PROSAC sampler
    if $\neg$ TestSample($S$) then    ▷ Degen. and cheirality tests
    continue
    $\theta \leftarrow$ EstimateModel($S$)
    if $\neg$ TestModel($\theta$) then    ▷ Degen. and cheirality tests
        continue
    $q, L \leftarrow$ Scoring($\mathcal{P}, \theta, \epsilon$)    ▷ MAGSAC++ and SPRT
    if $q > q^*$ then
        $L_{\mathrm{GC}}, q_{\mathrm{GC}}, \theta_{\mathrm{GC}} \leftarrow$ GC($\mathcal{P}$, $L$, $q$, $\theta$)    ▷ Algorithm 2
        if $q_{\mathrm{GC}} > q \wedge \neg$ TestModel($\theta_{\mathrm{GC}}$) then
            $q^*, \theta^*, L^* \leftarrow q_{\mathrm{GC}}, \theta_{\mathrm{GC}}, L_{\mathrm{GC}}$
        else
            $q^*, \theta^*, L^* \leftarrow q, \theta, L$
$L^*, \theta^* \leftarrow$ ModelPolishing($L^*, \theta^*$).
```

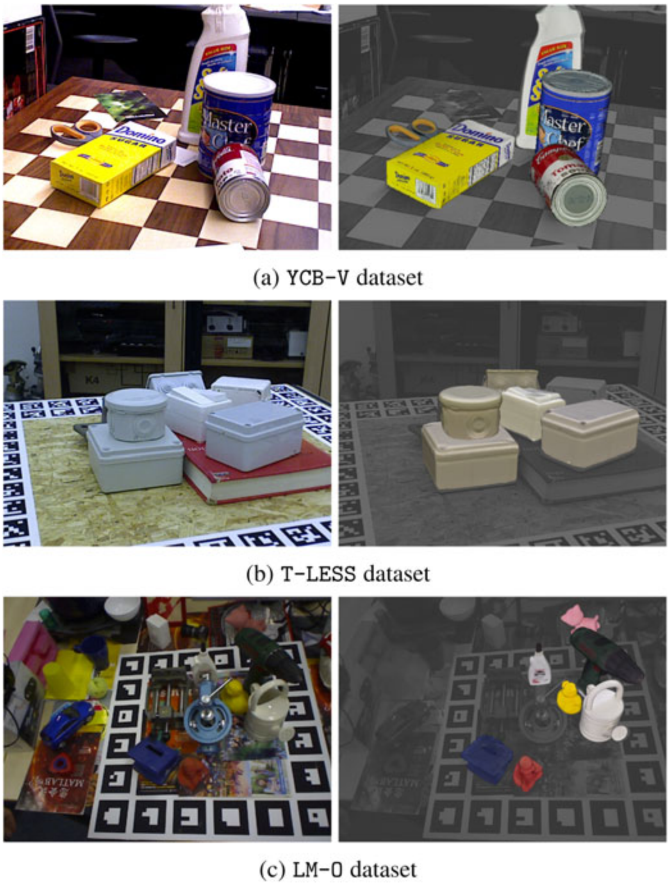
## 4 Experimental Results

We tested Graph-Cut RANSAC on fundamental matrix, relative pose, homography, and 6D object pose estimation using publicly available real-world datasets. The compared methods are GC-RANSAC with MSAC [25] and MAGSAC++ [17] scoring techniques, vanilla RANSAC [1], MSAC [25], and USAC [37]. USAC was applied with local optimization [29]

(a) ExtremeView dataset

(b) Homogr dataset

Fig. 4. Example image pairs from the datasets used for homography estimation evaluation; with inlier correspondence visualization.


(a) YCB-V dataset

(b) T-LESS dataset

(c) LM-O dataset

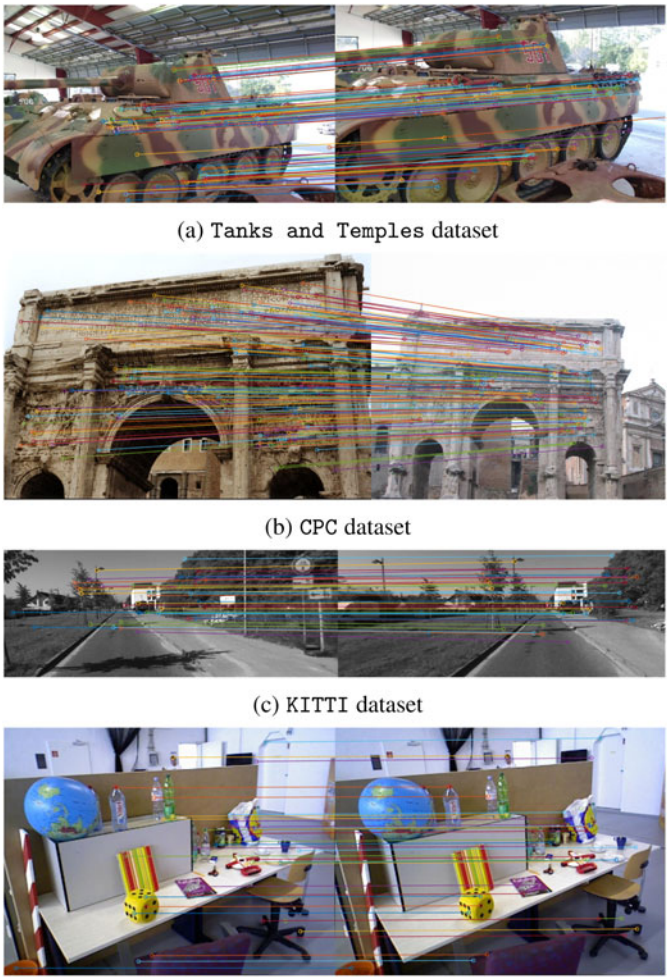
Fig. 5. Example scenes from the datasets used for 6D pose estimation. (*Left*) The input images passed to the EPOS method [45]. EPOS returns a set of 2D-3D correspondences and object masks. (*Right*) The 3D objects rendered using the poses estimated by GC-RANSAC from the predicted 2D-3D correspondences. Courtesy of T. Hodan.

and with the same modules as GC-RANSAC, i.e., SPRT test [22], degeneracy and cheirality tests, MSAC scoring, and PROSAC sampling [16]. Since relative pose and 6D object pose estimation are not included in the available USAC implementation, we copied the corresponding parts from our GC-RANSAC code. Also, we included NG-RANSAC [18] in the comparison for fundamental matrix and relative pose estimation. All compared methods are implemented in C++. The part of NG-RANSAC predicting inlier probabilities is implemented in Python and runs on GPU. Other parts, e.g. the one doing the robust estimation, are in C++. All methods were run on a computer with an Intel Core i7-8700K CPU and two GeForce RTX 2080 Ti GPUs. To provide a neighborhood graph, we used FLANN [46] in the 4D correspondence space using a hypersphere with radius 20 to assign neighbors to points. The distance for FLANN is calculated in the feature space and assigns, on average, 3–4 neighbors to most points. Parameter $\lambda$ was set to 0.975. These values lead to accurate results on all tested problems.

If not stated otherwise, the required confidence in the solution was set to 0.99 and the maximum iteration number to 5000 for all methods. The maximum iteration number is an upper bound for the iteration number – the robust estimation finishes in two cases: (i) by the termination criterion being triggered, (ii) by the iteration number exceeding the maximum iterations. For each method and problem, we chose the threshold maximizing the accuracy. For homography fitting, it was as follows: USAC, MSAC and GC-RANSAC (5.0 pixels); RANSAC (3.0 pixels). For fundamental and essential matrix fitting, it was as follows: USAC, RANSAC, MSAC, NG-RANSAC, and GC-RANSAC (0.75 pixels). For 6D object pose estimation, the threshold was set to 1 pixel. We note that since NG-RANSAC is a deep learning-based sampler and GC-RANSAC is a local optimization technique, they can be straightforwardly combined. We did not include MAGSAC [28] and MAGSAC++ [17] in the comparison since the improvements are orthogonal to that of GC-RANSAC. The algorithms, indeed, can be combined more than just taking the MAGSAC++ scoring function. However, that is out of this paper's scope. Comparing the methods would give the *false* message that they are competitors.

### 4.1 Fundamental Matrix Estimation

Fundamental matrix estimation is evaluated on the benchmark of [47]. The benchmark includes scenes from datasets TUM, KITTI, Tanks and Temples, and Community Photo Collection. TUM [48] consists of videos of indoor scenes. Each video is of resolution $640 \times 480$. KITTI [49] consists of consecutive frames of a camera mounted to a moving vehicle. The images are of resolution $1226 \times 370$. Both in KITTI and TUM, the image pairs are short-baseline and, thus, the epipolar geometry estimation is relatively easy, usually, with high inlier ratio. Tanks and Temples (T&T) [50] provides images of real-world objects for image-based reconstruction and, thus, contains mostly wide-baseline pairs. The images are of size from $1080 \times 1920$ up to $1080 \times 2048$. Community Photo Collection (CPC) [51] contains images of various sizes of landmarks collected from Flickr. The benchmark defines 1000 randomly selected image pairs from each dataset. SIFT [44] correspondences are detected, filtered by the SNN ratio test [44] and, finally, used for estimating the epipolar geometry. The used error metric is the symmetric geometric distance [52] (SGD) in pixels which compares two fundamental matrices by iteratively generating points on the borders of the



(a) Tanks and Temples dataset

(b) CPC dataset

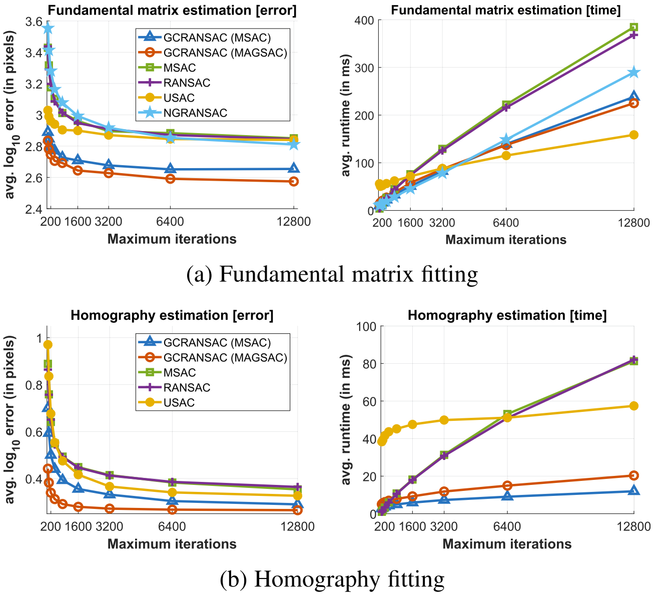
(c) KITTI dataset

(d) TUM dataset

Fig. 6. Example image pairs from the datasets used for epipolar geometry estimation; with inlier correspondence visualization.

images and, then, measuring their epipolar distances. Example image pairs from the datasets are shown in Fig. 6.

In Fig. 8, the cumulative distribution functions (CDF) of the SGD errors (horizontal; in pixels) are shown. The probability (vertical axis) is plotted as the function of the error (horizontal). For all datasets, GC-RANSAC is the most geometrically accurate method no matter if MSAC or MAGSAC++ scoring is used. The best performance is achieved by using MAGSAC++ scoring.



(a) Fundamental matrix fitting

(b) Homography fitting

Fig. 7. *Maximum iteration number study.* The avg. $\log_{10}$ error (px) and the run-time (ms) on manually selected inliers are plotted as the function of the max. iteration number. The confidence was set to 0.99.

The failure ratio (in percentage) and the average and median errors are reported in Table 1. A test is considered failure if the error of the estimated model is greater than the 1 percent of the image diagonal. The average values are calculated from the successful tests. The best values are shown in red, the second best ones are in blue. On three out of the four datasets, GC-RANSAC with MAGSAC++ scoring is superior to the competitor algorithms in terms of failure ratio, average and median errors. On Tanks and Temples dataset, GC-RANSAC with MSAC scoring leads to the best accuracy by a small margin, while MAGSAC++ scoring leads to significantly lower failure rate. NG-RANSAC performs competitively on the datasets TUM and KITTI. On Tanks and Temples and CPC it performs poorly, worse than any tested method. This behaviour is probably related to the properties of data it was trained on. GC-RANSAC with MAGSAC++ scoring outperforms NG-RANSAC in all cases.

In Fig. 7a, the $\log_{10}$ SGD errors (left plot) and the processing times (right; in milliseconds) are plotted as the function of the maximum iteration number. For these tests, the confidence was set to 0.99. GC-RANSAC leads to the most accurate results and it is the least sensitive one to the maximum iteration number. MAGSAC++ scoring leads to better accuracy than MSAC while having similar processing time. In Fig. 9a, the $\log_{10}$ SGD errors (left plot) and the processing times (right) are plotted as the function of the required confidence. For these tests, the maximum iteration number was set to 1000000. We excluded NG-RANSAC from this test

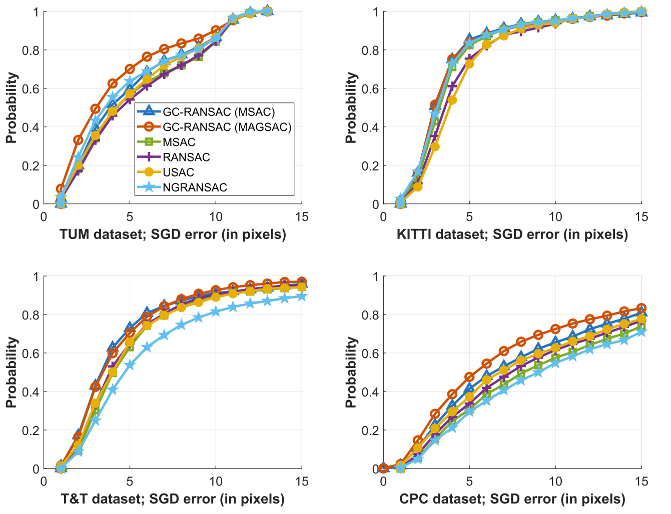


Fig. 8. *Fundamental matrix fitting.* The cumulative distribution functions of the SGD errors (in pixels) on four datasets, each consisting of 1000 image pairs. Being accurate is interpreted as a curve close to the top-left corner. The confidence and maximum iteration number were set to 0.99 and 5000, respectively.

TABLE 1
The Errors and Failure Ratios (in Percentage) are Reported for All Methods (1st Row) on All Problems (1st col.) and Datasets (2nd col.)

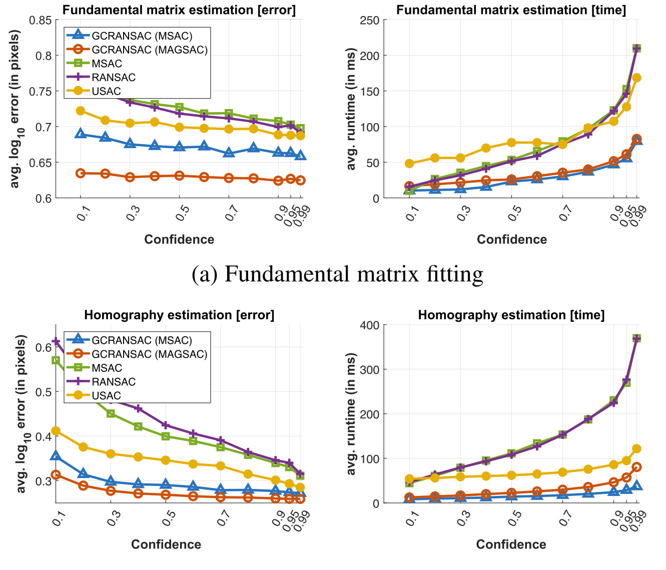
| | | | **GC-RANSAC (MSAC)** | **GC-RANSAC (MAGSAC++)** | MSAC [25] | RANSAC [1] | USAC [37] | NGRANSAC [18] |
|---|---|---|---|---|---|---|---|---|
| **Fundamental matrix** Figs. 8, 7a, 9a | TUM | $\epsilon_{avg}$ (px) | 5.42 | **4.95** | 5.81 | 5.91 | 5.60 | **5.26** |
| | | $\epsilon_{med}$ (px) | 4.35 | **3.73** | 4.68 | 4.97 | 4.63 | **3.96** |
| | | $f$ (%) | **10.30** | **8.52** | 12.00 | 13.39 | 10.40 | 10.40 |
| | KITTI | $\epsilon_{avg}$ (px) | **4.20** | **4.17** | 4.37 | 4.80 | 4.93 | 4.26 |
| | | $\epsilon_{med}$ (px) | **3.49** | **3.44** | 3.69 | 4.00 | 4.31 | 3.59 |
| | | $f$ (%) | **0.00** | **0.00** | **0.20** | 0.70 | **0.00** | **0.00** |
| | T&T | $\epsilon_{avg}$ (px) | **5.15** | **5.20** | 5.83 | 5.63 | 5.79 | 7.03 |
| | | $\epsilon_{med}$ (px) | **3.81** | **3.83** | 4.53 | 4.27 | 4.50 | 5.16 |
| | | $f$ (%) | 3.73 | **0.40** | 8.81 | 7.30 | **0.00** | **0.00** |
| | CPC | $\epsilon_{avg}$ (px) | **8.85** | **8.23** | 10.20 | 9.66 | 9.30 | 10.58 |
| | | $\epsilon_{med}$ (px) | **6.82** | **6.26** | 8.74 | 7.90 | 7.29 | 9.54 |
| | | $f$ (%) | 7.34 | **1.00** | 11.54 | 12.61 | 2.01 | **2.00** |
| **Relative pose** Figs. 14, 10 | TUM | $\epsilon_R$ (°) | 0.76 | **0.58** | 0.77 | 0.73 | 0.82 | **0.62** |
| | | $\epsilon_t$ (°) | 22.71 | **18.63** | 23.78 | 23.40 | 25.68 | **20.03** |
| | | $f$ (%) | 20.70 | **14.00** | 20.00 | 20.60 | 21.40 | **15.60** |
| | KITTI | $\epsilon_R$ (°) | **0.10** | **0.09** | 0.11 | 0.11 | 0.11 | 0.11 |
| | | $\epsilon_t$ (°) | 3.57 | 3.53 | **3.47** | 3.54 | 3.50 | **3.39** |
| | | $f$ (%) | 3.20 | 3.20 | **2.70** | 3.00 | 3.00 | **2.60** |
| | T&T | $\epsilon_R$ (°) | **4.20** | **4.37** | 4.77 | 5.05 | 5.00 | 4.49 |
| | | $\epsilon_t$ (°) | **4.36** | **4.62** | 5.10 | 4.96 | 5.10 | 4.82 |
| | | $f$ (%) | 2.50 | **1.30** | 2.80 | 3.10 | 2.80 | **2.40** |
| | CPC | $\epsilon_R$ (°) | 8.57 | **7.57** | 9.77 | 9.46 | 10.82 | **6.20** |
| | | $\epsilon_t$ (°) | 13.12 | **11.90** | 15.69 | 14.68 | 14.77 | **10.19** |
| | | $f$ (%) | 11.10 | **9.30** | 12.80 | 11.40 | 13.70 | **6.50** |
| **Homography** Figs. 11, 7b, 9b | EVD | $\epsilon_{avg}$ (px) | **2.70** | **2.49** | 3.38 | 3.40 | 2.79 | – |
| | | $\epsilon_{med}$ (px) | **2.34** | **2.30** | 3.55 | 3.67 | 2.58 | – |
| | | $f$ (%) | **16.13** | **7.24** | 18.53 | 17.40 | 19.47 | – |
| | homogr | $\epsilon_{avg}$ (px) | **1.13** | **1.17** | 1.34 | 1.33 | 1.55 | – |
| | | $\epsilon_{med}$ (px) | 1.12 | 1.12 | **1.09** | **1.11** | 1.59 | – |
| | | $f$ (%) | **0.12** | **0.00** | **0.00** | **0.00** | **0.12** | – |
| **6D object pose** Figs. 12, 13 | LM-O | $\epsilon_R$ (°) | **8.53** | **8.54** | 9.35 | 9.36 | 9.17 | – |
| | | $\epsilon_t$ (mm) | **40.06** | **41.80** | 41.90 | 44.16 | 41.25 | – |
| | | $f$ (%) | **20.49** | **17.44** | 33.74 | 35.53 | 27.57 | – |
| | YCB-V | $\epsilon_R$ (°) | 4.90 | **4.42** | 5.09 | 5.06 | **4.77** | – |
| | | $\epsilon_t$ (mm) | 24.70 | **21.82** | 25.04 | 24.83 | **22.89** | – |
| | | $f$ (%) | **17.08** | **15.03** | 21.14 | 21.33 | 17.84 | – |
| | T-LESS | $\epsilon_R$ (°) | **5.48** | **4.87** | 6.02 | 5.92 | 6.07 | – |
| | | $\epsilon_t$ (mm) | 19.80 | **17.72** | **19.76** | 34.23 | 20.67 | – |
| | | $f$ (%) | **38.45** | **36.53** | 44.15 | 43.74 | 44.46 | – |
| **All** | | $\epsilon_{avg}$ | **9.42** | **8.83** | 10.07 | 10.81 | 10.03 | – |
| | | $f$ (%) | **11.46** | **8.78** | 14.29 | 14.46 | 12.37 | – |
| | | $t$ (ms) | **81.82** | **93.12** | 131.96 | 120.73 | 107.84 | 728 (+ 1341) |

*For homography and fundamental matrix fitting, the avg. and median pixel errors are shown besides the failure rate. For relative and 6D object pose estimation, the avg. rotation (in degrees) and translation (in millimeters) errors are shown. The errors were calculated from the successful tests. The last three rows reports the average error, failure ratio and processing time (in milliseconds) over all datasets. The inlier-outlier thresholds were set to maximize the accuracy. The confidence was 0.99 and the maximum iteration number 5000. The best values in each column are shown by red and the second best ones by blue. The plus time demand of NG-RANSAC is the time of the model loading.*

since it has no confidence parameter. It can be seen that GC-RANSAC leads to the most accurate results and it is the least sensitive method to the confidence parameter. The processing time implied by the two tested scoring techniques is similar.

## 4.2 Homography Estimation

For homography estimation, we downloaded homogr (16 pairs) and EVD (15 pairs) datasets [30]. They consist of image pairs of different sizes from $329 \times 278$ up to $1712 \times 1712$ with point correspondences provided. The homogr



(a) Fundamental matrix fitting

(b) Homography fitting

Fig. 9. *Confidence study.* The avg. $\log_{10}$ error (in pixels) and the run-time (in milliseconds) are plotted as the function of the required confidence. The max. iteration number was 1000000.

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 [53] for homography estimation and were repeated 1000 times on each image pair. To measure the quality of the estimated homographies, we used the RMSE re-projection error (in pixels) calculated from the provided ground truth inliers in the reference image. Example image pairs are shown in Fig. 4.

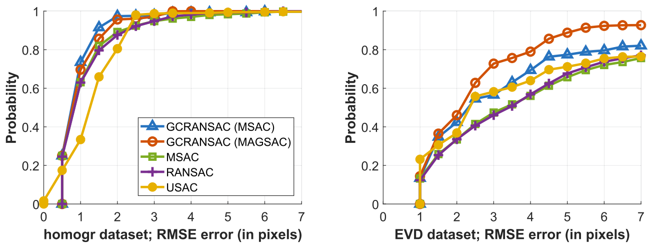


Fig. 11. *Homography fitting.* The cumulative distribution functions of the re-projection errors (in pixels) on two datasets. Being accurate is interpreted as a curve close to the top-left corner. The confidence and maximum iteration number were set to 0.99 and 5000, respectively.

The CDFs of the errors are in Fig. 11. On homogr, GC-RANSAC with MSAC scoring is slightly more accurate than the second best algorithm, i.e., GC-RANSAC with MAGSAC++. On EVD, the GC-RANSAC with MAGSAC++ goes the highest – it is the most accurate method. The avg. and median errors and the failure ratio are reported in Table 1. For GC-RANSAC, the avg. and median errors are fairly similar for both scoring techniques with a max. of $0.12$ px difference. On EVD, MAGSAC++ scoring leads to a significant improvement in the failure ratio ($\sim 6\%$).

The effect of changing the maximum iteration number and required confidence is shown, respectively, in Figs. 7b and 9b. It can be seen that GC-RANSAC with MAGSAC++ scoring is the least sensitive to these two tested parameters and leads to the most accurate results. The processing time is marginally higher than that of the fastest methods, i.e., GC-RANSAC with MSAC scoring.

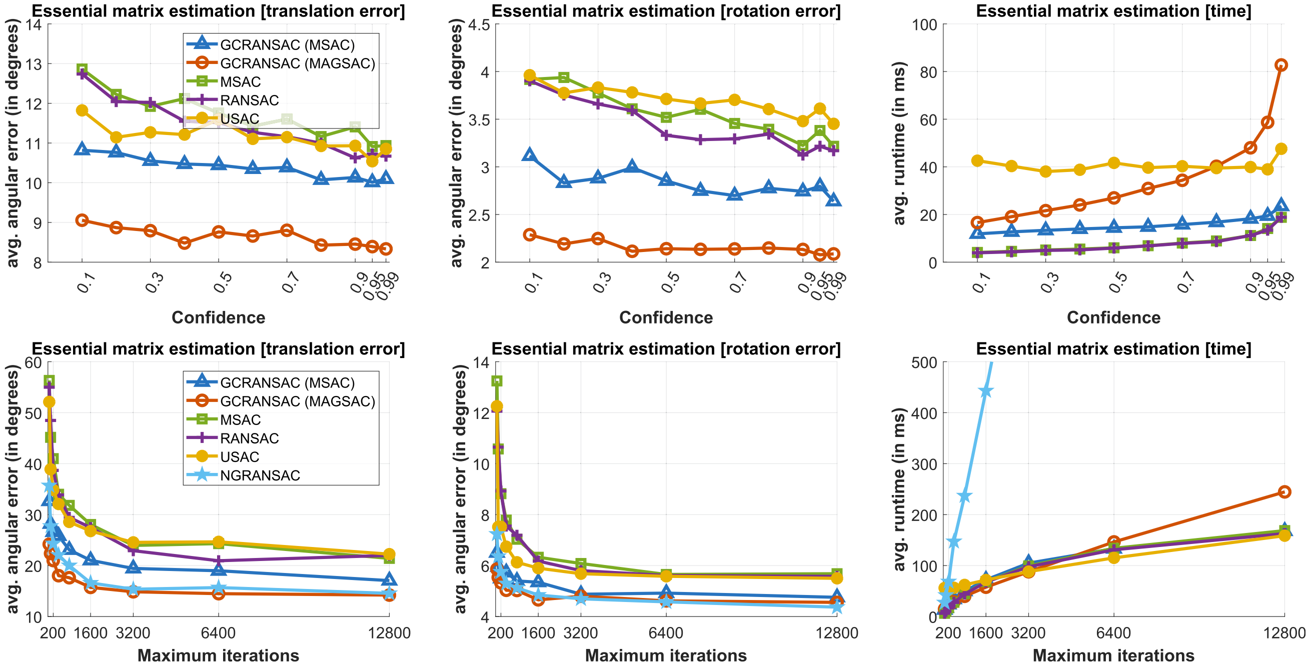


Fig. 10. *Relative pose fitting with varying parameters.* The average translation (left; in degrees) and rotation (middle; in degrees) errors and the processing time (right; in ms) are plotted as the function of the confidence (top) and maximum iteration number (bottom). The reported values are the average errors over 4000 scenes from datasets TUM, KITTI, T&T, and CPC. The compared methods are the proposed Graph-Cut RANSAC combined with MSAC [24] and MAGSAC++ [17] scoring techniques, MSAC [24], RANSAC [1], USAC [37], and NG-RANSAC [18]. In the bottom-right plot, the time of NG-RANSAC goes up to 3.4 seconds. In addition, NG-RANSAC model loading takes 1.4 seconds on average.

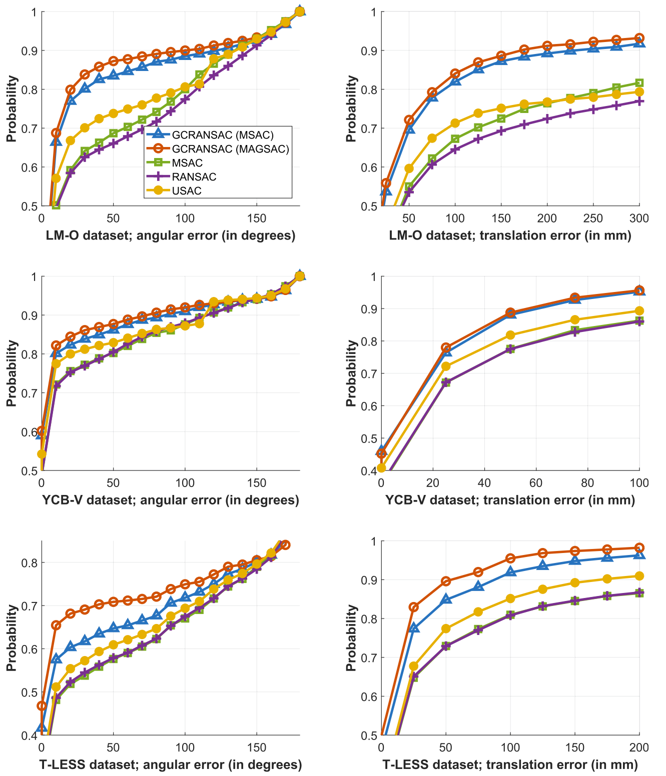


Fig. 12. *6D object pose estimation.* The cumulative distribution functions of the rotation (left column; in degrees) and translation (right; in millimeters) errors on three datasets (rows) are shown. Being accurate is interpreted as a curve close to the top-left corner. The confidence and max. iteration number were set to 0.99 and 5000, respectively.

### 4.3 Relative Pose Estimation

The relative pose, i.e., essential matrix, estimation is tested on the same datasets – `TUM`, `KITTI`, `Tanks and Temples`, and `Community Photo Collection` – as what are used for fundamental matrix estimation since the intrinsic camera matrices and the ground truth relative poses are provided for all scenes.

The reported rotation and translation errors are measured in degrees (°). The rotation error is calculated as follows: $\epsilon_{\mathrm{R}} = \cos^{-1}\left(\frac{1}{2}\left(\mathrm{tr}(\hat{\mathbf{R}}\mathbf{R}^{\mathrm{T}}) - 1\right)\right)$, where $\hat{\mathbf{R}}$ is the measured and $\mathbf{R}$ is the ground truth rotation matrix. Translation error $\epsilon_{\mathrm{t}}$ is the angular difference between the estimated $\hat{\mathbf{t}}$ and ground truth translations $\mathbf{t}$.

The CDFs of the rotation (Fig. 14a) and translation (Fig. 14b) errors are shown in Fig. 14. It can be seen that, GC-RANSAC obtains the most accurate rotations and translations. NG-RANSAC leads to similar accuracy.

The failure ratio and avg. rotation and translation errors are in Table 1. An estimation is considered failure if the errors are greater than $45^{\circ}$. Note that using different threshold does not change the ordering of the methods. The most accurate results are obtained by GC-RANSAC and NG-RANSAC which have similar accuracy. However, it can be seen that GC-RANSAC is *two orders of magnitude* faster. The effect of varying the confidence (top row) and maximum iteration number (bottom) is shown in Fig. 10. The average translation (left) and rotation (middle) errors and the processing time (right) are plotted as the function of the tested parameter. The most accurate results are achieved by GC-RANSAC with MAGSAC++ scoring and NG-RANSAC.

### 4.4 6D Object Pose Estimation

The experiments were conducted on three datasets: T-LESS [54], YCB-V [55], LM-O [56]. The datasets include color 3D object models and RGB-D images of VGA resolution with ground-truth 6D object poses. LM-O contains 200

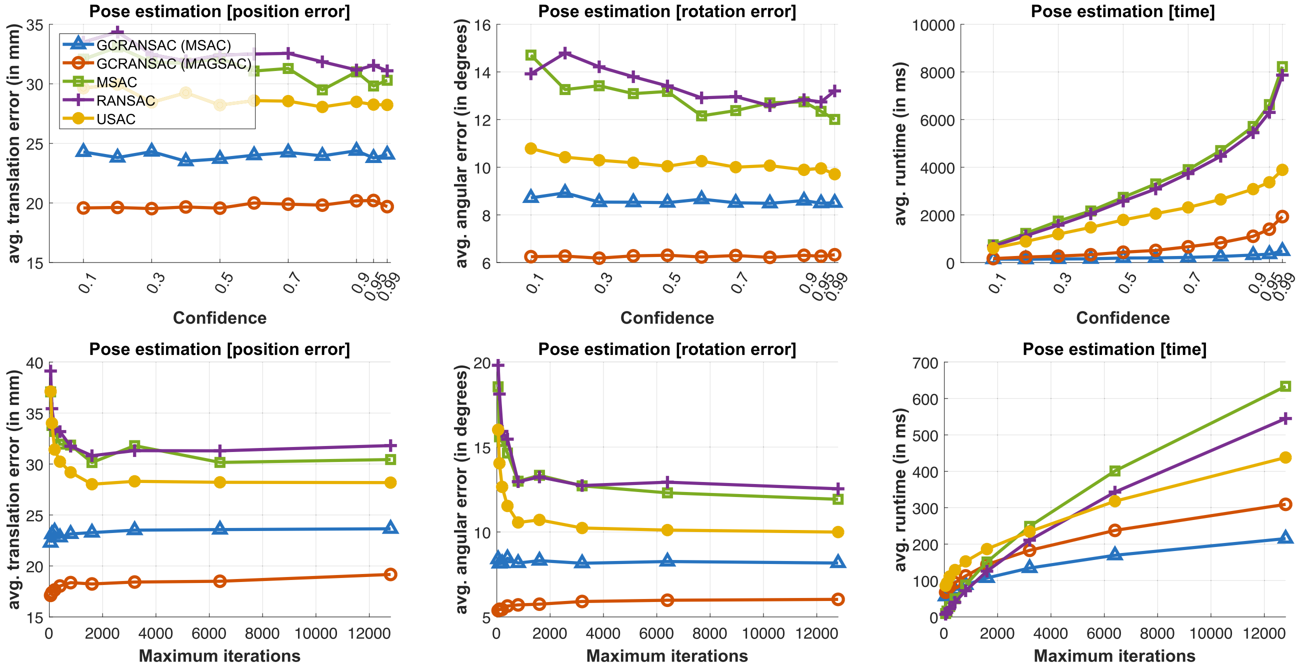
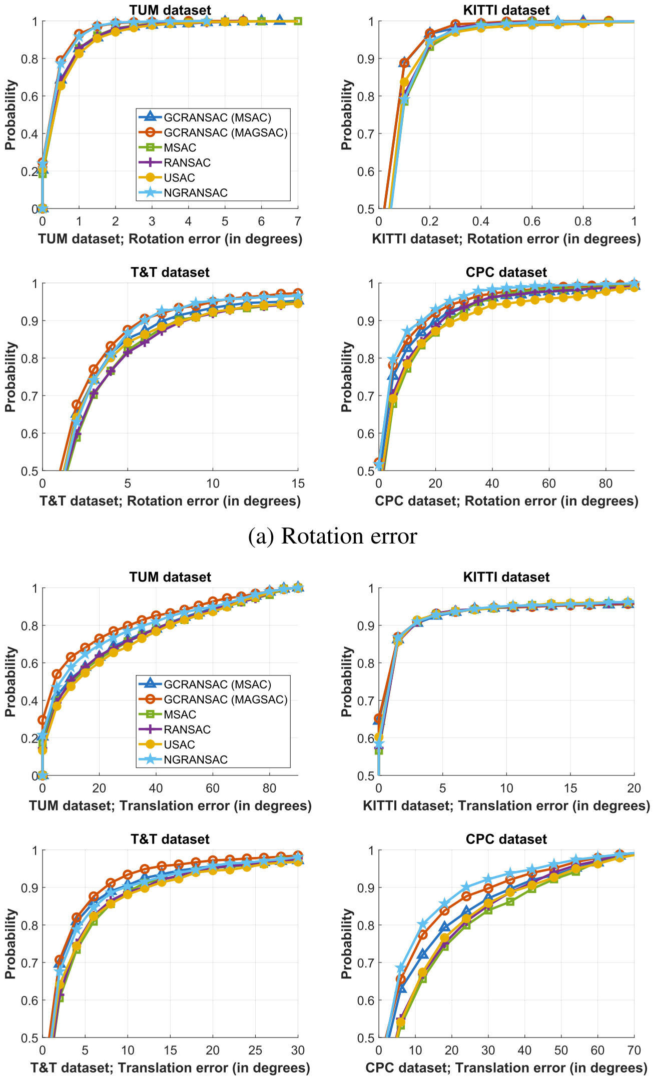


Fig. 13. *6D object pose fitting with varying parameters.* The average translation (left; in millimeters) and rotation (middle; in degrees) errors and the processing time (right; in milliseconds) are plotted as the function of the confidence (top) and maximum iteration number (bottom) The reported values are the average errors on datasets `LM-O`, `YCB-V` and `T-LESS`. The compared methods are the proposed Graph-Cut RANSAC combined with MSAC [24] and MAGSAC++ [17] scoring techniques, MSAC [24], RANSAC [1], and USAC [37].



(a) Rotation error

(b) Translation error

Fig. 14. *Relative pose fitting*. The cumulative distribution functions of the rotation and translation errors (both in degrees) on four datasets are shown. Being accurate is interpreted as a curve close to the top-left corner. The confidence and maximum iteration number were set to 0.99 and 5000, respectively.

test images with the ground truth for eight, mostly textureless objects from LM [57] captured in a cluttered scene under various levels of occlusion. YCB-V includes 21 objects, which are both textured and texture-less, and 900 test images showing the objects with occasional occlusions and limited clutter. T-LESS contains 30 objects with no significant texture or discriminative color, and with symmetries and mutual similarities in shape and/or size. It includes 1000 test images from 20 scenes with varying complexity, including challenging scenes with multiple instances of several objects and with a high amount of clutter and occlusion. To get 2D-3D correspondences, we applied the EPOS method [45]. The tested robust estimators were applied to the obtained correspondences and the 6D pose was compared to the ground truth one. Example images from the datasets are shown in Fig. 5.

The reported errors for rotation and translation both were measured in degrees (°). The rotation errors were calculated similarly as for relative pose estimation. The translation errors were in millimeters (mm) measured as follows: $\epsilon_{\mathrm{t}} = \sqrt{(\hat{\mathbf{t}} - \mathbf{t})^{\mathrm{T}}(\hat{\mathbf{t}} - \mathbf{t})}$, where $\hat{\mathbf{t}}$ is the estimated and $\mathbf{t}$ is the ground truth translation vector.

The CDFs of the rotation (left column) and translation errors (right) are shown in Fig. 12. It can be seen that, on all tested datasets, GC-RANSAC leads to the most accurate results both in terms of rotation and translation accuracy. The failure ratio and average rotation and translation errors are put in Table 1. An estimation is considered failure if the rotation has greater than $45^\circ$ angular error. Note that using different threshold does not change the ordering of the methods. It can be seen that GC-RANSAC leads always to the most accurate results with the lowest failure ratio. On YCV-V and T-LESS, MAGSAC++ scoring leads to the most accurate results both in terms accuracy and failure rate. On LM-O, MSAC scoring leads to the most accurate results by a small margin while being the second best in terms of failure ratio.

In the top row of Fig. 13, the average translation (left) and rotation (middle) errors and the processing times (right) are plotted as the function of the required confidence. For these tests, the maximum iteration number was set to 1000000. It can be seen that GC-RANSAC leads to the most accurate results and it is the least sensitive one to the confidence set. GC-RANSAC with MAGSAC++ scoring is the fastest method.

In the bottom row of Fig. 13, the average translation (left) and rotation (middle) errors and the processing times (right) are plotted as the function of the maximum iteration number. For these tests, the confidence was set to 0.99. It can be seen that GC-RANSAC leads to the most accurate results and it is the least sensitive one to the maximum iteration number. GC-RANSAC with MSAC scoring is the fastest method being slightly faster than MAGSAC++ scoring.

## 4.5 Effect of Spatial Coherence Weight

In Fig. 15, the effect of parameter $\lambda$ is shown. For each problem, the relative average error, i.e., divided by the maximum average error, is plotted as the function of $\lambda \in [0, 1]$. For relative and 6D object pose fitting, the rotation (R) and translation (t) errors are shown by different curves. For all problems, the most accurate results are obtained when $\lambda$ is relatively high. Interestingly, the most notable improvement is achieved for homography fitting, while the gain on the other tested problems is around $10 - 15$ percent. In all experiments, $\lambda$ is set to 0.975 since that leads to accurate results on all problems and all datasets. It is however straightforward to tune $\lambda$ whenever GC-RANSAC is applied to reflect spatial coherence properties of data in a particular domain.

## 4.6 Summary of the Experiments

The average errors and failure ratios on all datasets and problems are shown in the last two rows of Table 1. On average, Graph-Cut RANSAC leads to the most accurate results and the fastest robust estimation on all datasets on four vision problems. On 10 out of the 14 datasets, it fails to find the sought model parameters the least often. In the remaining cases, it fails only marginally more often than the best method. While NG-RANSAC shows comparable accuracy on relative pose fitting, its processing time is two orders of magnitude higher than that of GC-RANSAC. Using MAGSAC++ scoring inside GC-RANSAC leads to a significant improvement, in

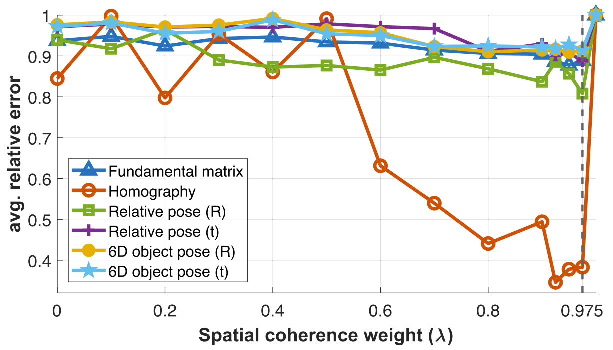


Fig. 15. The relative average errors, i.e., divided by the maximum avg., are plotted as the function of $\lambda$. The values are calculated from all datasets. The vertical dashed line denotes the $\lambda$ value, i.e., 0.975, where the average error summed over all problems is minimal.

terms of accuracy and failure rate, for almost all datasets. For all the tested datasets and problems, setting the required confidence in the solution to 0.99 and the maximum iteration number to $\sim$3000 leads to the most accurate results. We included the results of [58], [59] in the supplementary material, available online. Additional experiments on different datasets can be found in [60].

## 5 Conclusion

We presented the Graph-Cut RANSAC algorithm which combines the strands of robust model fitting and energy minimization. GC-RANSAC is capable of modelling spatially coherent point distributions, and exploits this property in a local optimization procedure. It is more geometrically accurate than state-of-the-art robust estimation. It runs in real-time for many problems at a speed similar to its less accurate alternatives. It is much simpler to implement in a reproducible manner than many of the competitors (RANSACs with local optimization). The inlier selection in the local optimization, given the so-far-the-best model, is globally optimal. Two new parameters are introduced in GC-RANSAC, the neighborhood size and weight $\lambda$, which are easy to set. If $\lambda = 0$ or the neighborhood size is too small, the algorithm acts as a well-implemented LO-RANSAC. Otherwise, if $\lambda \in (0, 1)$ and a reasonable neighborhood size is used, the results are superior to LO-RANSAC and USAC. On the tested problems and datasets, $\lambda = 0.975$ leads to the best performance with a neighborhood size assigning 3–4 neighbors, on average, to each point.

The C++ and Python implementations of Graph-Cut RANSAC are available at https://github.com/danini/graph-cut-ransacincluding all components tested in the paper and examples for homography, fundamental matrix, relative and 6D object pose estimation.

## Acknowledgments

This work was supported by the Czech Science Foundation under Grant GA18-05360S, the Ministry of Education OP VVV Project CZ.02.1.01/0.0/0.0/16_019/0000765 Research Center for Informatics, and the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems and Artificial Intelligence National Laboratory Programs.

## 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, pp. 381–395, 1981.

[2] P. H. S. Torr and D. W. Murray, “Outlier detection and motion segmentation,” in *Proc. SPIE Opt. Tools Manuf. Adv. Autom.*, 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, pp. 312–333, 1998.

[4] P. Pritchett and A. Zisserman, “Wide baseline stereo matching,” in *Proc. 6th 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, pp. 761–767, 2004.

[6] D. Mishkin, J. Matas, and M. Perdoch, “MODS: Fast and robust method for two-view matching,” *Comput. Vis. Image Understanding*, vol. 141, pp. 81–93, 2015.

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

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

[9] 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.

[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] 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. 1–10.

[15] K. Ni, H. Jin, and F. Dellaert, “GroupSAC: Efficient consensus in the presence of groupings,” in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 2193–2200.

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

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

[18] E. Brachmann and C. Rother, “Neural-guided RANSAC: Learning where to sample model hypotheses,” in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 4322–4331. [Online]. Available: https://github.com/vislearn/ngransac

[19] O. Chum and J. Matas, “Randomized RANSAC with Tdd test,” in *Proc. Brit. Mach. Vis. Conf.*, 2002, pp. 448–457.

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

[21] J. Matas and O. Chum, “Randomized RANSAC with sequential probability ratio test,” in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, 2005, pp. 1727–1732.

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

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

[24] P. H. S. Torr and A. Zisserman, “MLESAC: A new robust estimator with application to estimating image geometry,” *Comput. Vis. Image Understanding*, vol. 78, pp. 138–156, 2000.

[25] 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.

[26] 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.

[27] L. Moisan, P. Moulon, and P. Monasse, “Automatic homographic registration of a pair of images, with a contrario elimination of outliers,” *Image Process. On Line*, vol. 2, pp. 56–73, 2012.

[28] D. Barath, J. Noskova, and J. Matas, “MAGSAC: Marginalizing sample consensus,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10189–10197.

[29] O. Chum, J. Matas, and J. Kittler, “Locally optimized RANSAC,” in *Proc. Joint Pattern Recognit. Symp.*, 2003, pp. 236–243.

[30] K. Lebeda, J. Matas, and O. Chum, “Fixing the locally optimized RANSAC,” in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–11.

[31] Y. Boykov, O. Veksler, and R. Zabih, “Markov random fields with efficient approximations,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1998, pp. 648–655.

[32] R. Zabih and V. Kolmogorov, “Spatially coherent clustering using graph cuts,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. II–II.

[33] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, “Fast approximate energy minimization with label costs,” *Int. J. Comput. Vis.*, vol. 96, pp. 1–27, 2012.

[34] D. Barath, J. Matas, and L. Hajder, “Multi-H: Efficient recovery of tangent planes in stereo images,” in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 13.1–13.13.

[35] D. Barath and J. Matas, “Multi-class model fitting by energy minimization and mode-seeking,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 221–236.

[36] D. Baráth and J. Matas, “Progressive-X: Efficient, anytime, multi-model fitting algorithm,” in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 3780–3788.

[37] 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.

[38] D. Barath and J. Matas, “Graph-cut RANSAC,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6733–6741.

[39] V. Kolmogorov and R. Zabin, “What energy functions can be minimized via graph cuts?,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.

[40] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[41] O. Chum, T. Werner, and J. Matas, “Two-view geometry estimation unaffected by a dominant plane,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 772–779.

[42] H. Haber, (2011). Three-dimensional proper and improper rotation matrices. University of California, Santa Cruz Physics 116A Lecture Notes.

[43] O. Chum, T. Werner, and J. Matas, “Epipolar geometry estimation via RANSAC benefits from the oriented epipolar constraint,” in *Proc. 17th Int. Conf. Pattern Recognit.*, 2004, pp. 112–115.

[44] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, 1999, pp. 1150–1157.

[45] T. Hodan, D. Barath, and J. Matas, “EPOS: Estimating 6D pose of objects with symmetries,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11700–11709.

[46] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” in *Proc. 4th Int. Conf. Comput. Vis. Theory Appl.*, 2009, pp. 331–340.

[47] J.-W. Bian *et al.* “An evaluation of feature matchers for fundamental matrix estimation,” in *Proc. Brit. Mach. Vis. Conf.*, 2019. [Online]. Available: https://jwbian.net/fm-bench

[48] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[49] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[50] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, “Tanks and temples: Benchmarking large-scale scene reconstruction,” *ACM Trans. Graph.*, vol. 36, 2017, Art. no. 78.

[51] K. Wilson and N. Snavely, “Robust global translations with 1DSfM,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 61–75.

[52] Z. Zhang, “Determining the Epipolar geometry and its uncertainty: A review,” *Int. J. Comput. Vis.*, vol. 27, no. 2, pp. 161–195, 1998.

[53] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[54] T. Hodaň, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, “T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2017, pp. 880–888.

[55] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, “PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes,” in *Proc. Robot. Sci. Syst.*, 2018, doi: 10.15607/RSS.2018.XIV.019.

[56] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, “Learning 6D object pose estimation using 3D object coordinates,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 536–551.

[57] S. Hinterstoisser *et al.*“Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes,” in *Proc. Asian Conf. Comput. Vis.*, 2012, pp. 548–562.

[58] T.-J. Chin, P. Purkait, A. Eriksson, and D. Suter, “Efficient globally optimal consensus maximisation with tree search,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2413–2421.

[59] Z. Cai, T.-J. Chin, and V. Koltun, “Consensus maximization tree search revisited,” in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 1637–1645.

[60] J. Matas and O. Chum, “Randomized RANSAC with T(d,d) test” in *Proc. Brit. Mach. Vis. Conf.*, 2002, pp. 43.1–43.10, doi: 10.5244/C.16.43.

**Daniel Barath** received the PhD degree from the Eotvos Lorand University, in 2019. He was 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 (MTA SZTAKI), Budapest, Hungary. He is currently at ETH Zurich. His research interest include robust estimation and minimal methods in computer vision.

**Jiri Matas** is currently a professor at the Center for Machine Perception, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic. He has authored or coauthored more than 250 papers in the area of 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.