Graph-Cut RANSAC

Daniel Barath
Machine Perception Research Laboratory
MTA SZTAKI, Budapest, Hungary
barath.daniel@sztaki.mta.hu

Jiri Matas
Centre for Machine Perception, Department of Cybernetics
Czech Technical University, Prague, Czech Republic
matas@cmp.felk.cvut.cz

Abstract

A novel method, called Graph Cut RANSAC, GC-RANSAC in short, is presented. To separate inliers and outliers, it runs the graph cut algorithm in the local optimization (LO) step which is applied after a so-far-the-best model is found. The proposed LO step is conceptually simple, easy to implement, globally optimal and efficient.

Experiments show that GC-RANSAC outperforms LO-RANSAC and its state-of-the-art variants in terms of both accuracy and the required number of iterations for line, homography and fundamental matrix estimation on standard public datasets. GC-RANSAC is very efficient, its processing time for hundreds of input points is approximately 1 – 10 milliseconds, depending on the inlier-outlier ratio.

1. Introduction

The RANSAC (RANdom SAmple Consensus) algorithm proposed by Fischler and Bolles [5] in 1981 has become the most widely used robust estimator in computer vision. RANSAC and similar hypothesize-and-verify approaches have been successfully applied to many vision tasks, e.g. to short baseline stereo [20, 22], wide baseline stereo matching [16, 12, 13], motion segmentation [20], image mosaicing [7], detection of geometric primitives [19], multi-model fitting [24], or for initialization of multi-model fitting algorithms [9, 15].

In brief, the RANSAC approach repeatedly selects random subsets of the input data and fits a model to them, e.g. a line to two 2D points or a fundamental matrix to seven point correspondences. In the second step, the model support, i.e. the number of inliers, is obtained. The model with the highest support, polished e.g. by a least-squares fit on inliers, is returned.

Figure 1: The proposed graph-cut based local optimization converging from a “not-all-inlier” sample, i.e. it is contaminated by an outlier, to the desired model. (a) The input data points, (b) RANSAC-like sampling and model fitting, (c) computation of model support, e.g. counting the inliers, (d) considering spatial proximity by graph-cut, (e-f) iterated local optimization using least-squares fitting and graph-cut. Best viewed in color.

In the last three decades, many modification of RANSAC have been proposed. For instance, NAPSAC [14], PROSAC [3] or EVSAC [6] modify the sampling strategy to increase the probability of selecting an all-inlier sample earlier. NAPSAC considers spatial coherence of the input data points, PROSAC exploits the ordering of the points by their inlier probability, EVSAC uses an estimate of confidence in each point. The model support computation step had also been discussed in several papers, e.g. MLESAC [21] and MSAC [8]. The model is estimated by a maximum-likelihood process, albeit under certain assumptions, with all its beneficial properties. In practice, MLESAC results
are often superior to the inlier counting of plain RANSAC and less sensitive to the used-defined threshold. The termination of RANSAC is controlled by a manually set confidence value \( \eta \) and the sampling stops when the probability of finding a model with higher support falls below \( \eta \).

Observing that in practice RANSAC requires more samples than theory predicts, Chum et al. [4] 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. Chum et al. [4] 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 model. In the original paper [4], local optimization is implemented as an iterated least squares re-fitting with a shrinking inlier-outlier threshold inside an inner RANSAC applied only to the inliers of the current model. In the reported experiments, LO-RANSAC outperforms standard RANSAC in both accuracy and the required number of interations. The number of LO runs is close to the logarithm of the number of verifications, and it does not create a significant overhead in the processing time in most of the cases tested.

However, it was shown by Lebeda et al. [11] that for models with high inlier counts the local optimization step becomes a computational bottleneck of the process due to the iterated least-squares model fitting. This is fixed by using a \( 7m \)-sized subset of the inliers in each LO step, where \( m \) is the size of a minimum sample; the factor of 7 was set by exhaustive experimentation. The idea of local optimization has been included in state-of-the-art RANSAC approaches like USAC [17]. Nevertheless, the LO procedure remains ad hoc, complex and requires multiple parameters.

In this paper, we combine two strands of research to obtain a state-of-the-art RANSAC. So far, in the large body of RANSAC-related literature, the inlier-outlier decision has always been a function of the distance to the model, done individually for each data point. Yet both inliers and outliers are spatially coherent, a point near an outlier or inlier is more likely to be an outlier or inlier respectively, see Fig. 1. Spatial coherence, leading to the Potts-model [2], has been exploited in many vision problems, e.g. in segmentation [23], multi-model fitting [9, 15] or sampling [14]. It has probably been always considered computationally prohibitive to formulate model verification in RANSAC as a graph-cut problem. But when applied as the LO-step in [4] just on the so-far-the-best model, the number of graph-cut is only the logarithm of the number sampled and verified models, and can be achieved in real-time.

The novel method, called Graph Cut RANSAC (GC-RANSAC) is simply an LO-RANSAC with graph-cut as local optimization. GC-RANSAC is superior to LO-RANSAC in a number of aspects. First, as mentioned above, it is capable to model spatial coherence of inliers and outliers. Second, the LO step is conceptually simple, easy to implement\(^2\), globally optimal and computationally efficient graph cut with only a few intuitive and learnable parameters unlike the ad hoc, iterative and complex LO steps [4]. Third, we show experimentally that GC-RANSAC outperforms LO-RANSAC and its recent variants in both accuracy and the required number of iterations on a wide range of publicly available datasets. On many problems, it is faster than the competitors in terms of wall-clock time. Finally, we were surprised to observe that GC-RANSAC terminates before the theoretically expected number of iterations. The reason is that the local optimization that takes spatial proximity into account is often capable of converging to a “good” model even when starting from a sample that is not all-inlier, i.e. it contains an outlier or outliers.

2. Local Optimization and Spatial Coherence

In this section, we formulate the inlier selection of RANSAC as an energy minimization considering point-to-point proximity. The proposed local optimization is seen as an iterative energy minimization of a binary labeling (outlier – 0 and inlier – 1). For the sake of simplicity, we start from the original RANSAC scheme and then formulate the maximum-likelihood estimation as an energy minimization. The term considering the spatial coherence will be included into the energy. Finally, we propose a technique to set the parameter balancing the energy terms automatically on the basis of the input.

2.1. Formulation as Energy Minimization

We assume a point set \( P \subseteq \mathbb{R}^m \) \((n > 0)\), a model represented by a parameter vector \( \theta \in \mathbb{R}^m \) \((m > 0)\) and a distance function \( \phi : \mathcal{P} \times \mathbb{R}^m \rightarrow \mathbb{R} \) measuring the point-to-model assignment cost are given.

For the standard RANSAC scheme which applies a tophat fitness function \((1 – close, 0 – far)\), the implied unary energy is:

\[
E_{\{0,1\}}(L) = \sum_{p \in P} ||L_p||_{\{0,1\}},
\]

where

\[
||L_p||_{\{0,1\}} = \begin{cases} 
0 & \text{if} \ (L_p = 1 \land \phi(p, \theta) < \epsilon) \lor \ (L_p = 0 \land \phi(p, \theta) \geq \epsilon) \\
1 & \text{otherwise.}
\end{cases}
\]

Parameter \( L \in \{0,1\}^{|P|} \) is a labeling, ignored in standard RANSAC, \( L_p \in L \) is the label of point \( p \in \mathcal{P}, |P| \) is the number of points, and \( \epsilon \) is the inlier-outlier threshold. Using

\(^1\)This interpretation of \( \eta \) holds for the standard cost function only.

\(^2\)We will make the optimized C++ source of GC-RANSAC publicly available after publication.
energy $E_{\{0,1\}}$ we get the same result as RANSAC since it does not penalize only two cases: (1) when $p$ is labeled inlier and it is closer to the model than the threshold, or (2) when $p$ is labeled outlier and it is farther from the model than $\epsilon$. This is exactly what RANSAC does.

Since the publication of RANSAC, several papers discussed, e.g. [11], replacing the $\{0,1\}$ loss with a kernel function $K : \mathbb{R} \times \mathbb{R} \rightarrow [0,1]$, e.g. the Gaussian-kernel. Such choice is close to maximum likelihood estimation as proposed in MLESAC [21]. This improves the accuracy reducing the sensitivity on threshold $\epsilon$ proposed in MLESAC [21]. Such choice is close to maximum likelihood estimation as proposed in MLESAC [21].

In GC-RANSAC, we use $E_K$ as the unary energy term in the graph-cut based verification.

### 2.2. Spatial Coherence

Benefiting from a binary labeling energy minimization, we are able to include additional energy terms, i.e. consider spatial coherence of the points, yet keep the problem solvable efficiently and globally via the standard graph-cut algorithm.

Considering point proximity is a well-known approach for sampling [14] or multi-model fitting [9, 15, 1]. To the best of our knowledge, there is no paper exploiting it in the local optimization step of methods like LO-RANSAC. Applying the Potts-model which penalizes all neighbors having different labels would be a justifiable choice to be the pair-wise energy. The problem arises when the data contains significantly more outliers, probably close to desired model, than inliers. In that case, penalizing differently labeled neighbors using the same penalty for all classes many times leads to the domination of outliers forcing all inliers to be labeled outlier. To overcome this problem, we modified the Potts-model to use different penalty for each neighboring point pair on the basis of their distances. The proposed pair-wise energy term is

$$E_S(L) = \sum_{(p,q) \in A} \begin{cases} 1 & \text{if } L_p \neq L_q \\ \frac{1}{2} (K_p + K_q) & \text{if } L_p = L_q = 0 \\ 1 - \frac{1}{2} (K_p + K_q) & \text{if } L_p = L_q = 1 \end{cases}$$

where $K_p = K(\phi(p, \theta), \epsilon)$, $K_q = K(\phi(q, \theta), \epsilon)$ and $(p,q)$ is a edge of neighborhood graph $A$ between points $p$ and $q$. In $E_S$, for both points labeled outlier the penalty is $\frac{1}{2} (K_p + K_q)$ thus “rewarding” label 0 if the neighboring points are far from the model. The penalty of considering a point as inlier is $1 - \frac{1}{2} (K_p + K_q)$ which rewards the label if the points are close to the model.

The proposed overall energy measuring the fitness of points to a model and considering spatial coherence is $E(L) = E_K(L) + \lambda E_S(L)$, where $\lambda$ is a parameter balancing the terms. The globally optimal labeling $L^* = \arg\min_L E(L)$ can easily be determined in polynomial time using graph-cut algorithm.

#### 2.3. Automatic $\lambda$ Selection

On different types of data, the scales of the two energy terms could be significantly different, thus it is necessary to set $\lambda$ automatically on the basis of the input to avoid manual parameter setting problem-by-problem. Choosing $K$ to be the Gaussian-kernel leads to the following inequality

$$E_K(L) = \sum_{p \in P} ||L_p||_K \leq \omega \sum_{p \in P} 1 = \omega |P|,$$

where $|P|$ is the number of data points, $\omega$ is the current estimate of the inlier ratio and $0 \leq ||L_p||_K \leq 1$ for every $p$. An upper limit can also be determined for $E_S$ as

$$E_S(L) \leq \sum_{(p,q) \in A} 1,$$

which is basically the edge number in the neighborhood graph. Finally,

$$\lambda = \frac{\omega |P|}{(\sum_{(p,q) \in A} 1)} , \quad \omega = \frac{|I^*|}{|P|}$$

is an appropriate choice to balance the energies, where $I^*$ is the current set of inliers. If a new so-far-the-best model is found, parameters $\lambda$ and $\omega$ are updated using the new inlier number.

### 3. GC-RANSAC

In this section, we include the proposed energy minimization-based local optimization into RANSAC. Benefiting from this new approach, the LO step is getting simpler and cleaner than that of LO-RANSAC.

The main algorithm is shown in Alg. 1. The first step is the determination of neighborhood graph $A$ for which we use a sphere with a predefined radius $r$ – this is a parameter of the algorithm. In Alg. 1, function $H$ is as follows [5]:

$$H(|L^*|, \mu) = \frac{\log(\mu)}{\log(1 - P_I)}$$

where $P_I = \frac{(L^*^c)}{C}$, it calculates the required iteration number of RANSAC on the basis of desired probability $\mu$. 

```python
In [5]:
```
the size of the required minimal point set \( m \) and the inlier number \( |L^*| \) regarding to the current so-far-the-best model. Note that norm \(|\cdot|\) applied to the labeling counts the inliers.

Every \( k \)-th iteration draws a minimal sample using a sampling strategy, then computes the parameters \( \theta_k \) of the implied model and its support

\[
w_k = \sum_{p \in \mathcal{P}} K(\phi(p, \theta_k), \epsilon) \tag{5}\]

w.r.t. the data points, where function \( K \) is a Gaussian-kernel as proposed in Eq. 2. If \( w_k \) is higher than that of the so-far-the-best model \( w^* \), this model is considered the new so-far-the-best, local optimization is applied. Note that in the proposed algorithm, local optimization is applied only if sufficient number of RANSAC iterations have been performed, i.e. after the first 50 iterations, as it is proposed in [11]. However, local optimization is applied to the so-far-the-best model at least once.

The proposed local optimization is written in Alg. 2. The main iteration can be considered as a grab-cut-like [18] alternation consisting of two major steps: (1) graph-cut and (2) model re-fitting. The construction of problem graph \( G \) using unary and pair-wise terms Eqs. 1, 3 is shown in Alg. 3. Functions AddTerm1 and AddTerm2 are discussed by [10] in depth. Graph-cut is applied to \( G \) determining the optimal labeling \( \tilde{L} \) which considers the spatial coherence of the points and their distances from the so-far-the-best model. Model parameters \( \theta \) are computed using a \( 7m \)-sized random subset of the inliers in \( L \), thus speeding up the process, similarly to [11] does, where \( m \) is the size of a minimal sample, e.g. \( m = 2 \) for lines. Note that \( 7m \) is set by exhaustive experimentation in [11] and this value also suited for us. Finally, the support \( w \) of \( \theta \) is computed and the so-far-the-best model is updated if the new one has higher support, otherwise the process terminates. After the main algorithm, a local optimization step is applied if it is not performed at least once during the algorithm, and the parameters of the obtained so-far-the-best model is re-estimated using the whole inlier set similarly to plain RANSAC does.

Remark: Including a RANSAC-like procedure into the local optimization step which repeats the \( 7m \)-size sample selection would be a straightforward step, however, it yields high computational overhead without significant superiority in accuracy according to our experience. Even so, if real-time capability is not required it is an important step towards the maximum accuracy.

4. Experimental Results

In this section, GC-RANSAC is validated both on synthesized and real world data and compared with plain RANSAC [5], LO-RANSAC [4], LO*-RANSAC and LO'*-RANSAC [11]. The parameter setting is reported in Table 1. To make the comparison fair, all methods, apply

Algorithm 1 The GC-RANSAC Algorithm.

**Input:** \( \mathcal{P} \) – data points; \( r \) – sphere radius, \( \epsilon \) – threshold \( k_{\text{min}} \) – iterations before LO, \( \mu \) – confidence;

**Output:** \( \theta \) - model parameters; \( L \) – labeling

1: \( w^*, n_{LO} \leftarrow 0, 0. \)
2: \( \mathcal{A} \leftarrow \text{Build neighborhood-graph using } r. \)
3: for \( k = 1 \rightarrow H(|L^*|, \mu) \) do
4: \( S_k \leftarrow \text{Draw a minimal sample.} \)
5: \( \theta_k \leftarrow \text{Estimate a model using } S_k. \)
6: \( w_k \leftarrow \text{Compute the support of } \theta_k. \) \( \triangleright \) Eq. 5
7: if \( w_k > w^* \) then
8: \( \theta^*, L^*, w^* \leftarrow \theta_k, L_k, w_k \)
9: if \( k \geq k_{\text{min}} \) then
10: \( \theta_{LO}, L_{LO}, w_{LO} \leftarrow \text{Local opt.} \) \( \triangleright \) Alg. 2
11: \( n_{LO} \leftarrow n_{LO} + 1. \)
12: if \( w_{LO} > w^* \) then
13: \( \theta^*, L^*, w^* \leftarrow \theta_{LO}, L_{LO}, w_{LO} \)
14: if \( n_{LO} = 0 \) then
15: \( \theta^*, L^*, w^* \leftarrow \text{Local opt.} \) \( \triangleright \) Alg. 2
16: \( \theta^* \leftarrow \text{Least-squares model fitting using } L^*. \)

Algorithm 2 Local optimization.

**Input:** \( \mathcal{P} \) – data points, \( L^* \) – labeling;

**Output:** \( L_{LO}^* \) – labeling, \( w_{LO}^* \) – support, \( \theta_{LO}^* \) – model;

1: \( w_{LO}^*, L_{LO}^*, \theta_{LO}^*, \text{changed} \leftarrow w^*, L^*, \theta^*, 1. \)
2: while \( \text{changed} \) do
3: \( G \leftarrow \text{Build the problem graph.} \) \( \triangleright \) Alg. 3
4: \( L \leftarrow \text{Apply graph-cut to } G. \)
5: \( I_{7m} \leftarrow \text{Select a } 7m\text{-sized random inlier set.} \)
6: \( \theta \leftarrow \text{Fit a model using labeling } I_{7m}. \)
7: \( w \leftarrow \text{Compute the support of } \theta. \)
8: \( \text{changed} \leftarrow 0. \)
9: if \( w > w_{LO}^* \) then
10: \( \theta_{LO}^*, L_{LO}^*, w_{LO}^*, \text{changed} \leftarrow \theta, L, w, 1. \)

Algorithm 3 Problem Graph Construction.

**Input:** \( \mathcal{P} \) – data points, \( \mathcal{A} \) – neighborhood-graph

**Output:** \( G \) – problem graph;

1: \( G \leftarrow \text{EmptyGraph}.() \)
2: for \( p \in \mathcal{P} \) do
3: \( c_0, c_1 \leftarrow K(\phi(p, \theta), 1 - K(\phi(p, \theta), \epsilon) \)
4: \( G \leftarrow \text{AddTerm1}(G, p, c_0, c_1). \)
5: for \( (p, q) \in \mathcal{A} \) do
6: \( c_{01}, c_{10} \leftarrow 1, 1. \)
7: \( c_{00} \leftarrow 0.5(K(\phi(q, \theta) + K(\phi(p, \theta))) \)
8: \( c_{11} \leftarrow 1 - 0.5(K(\phi(q, \theta) + K(\phi(p, \theta))). \)
9: \( G \leftarrow \text{AddTerm2}(G, p, q, c_{00}, c_{01}, c_{10}, c_{11}). \)
uniform sampling and encapsulates the point-to-model distance, e.g. re-projection error for homographies, with a Gaussian-kernel using $\epsilon = 0.31$, which is set by exhaustive experimentation. The radius of the sphere to determine neighboring points is 20 pixels. Parameter $\lambda$ for GC-RANSAC is estimated using Eq. 4 adaptively during the algorithm.

**Synthetic Tests on 2D Lines.** To compare GC-RANSAC with the state-of-the-art in a fully controlled environment, we chose two simple tests: detection of a 2D straight or dashed line. For each trial, a $600 \times 600$ window and a random line was generated in its implicit form, sampled at 100 locations and zero-mean Gaussian-noise with $\sigma$ spread was added to the coordinates. For a straight line, the points were generated using uniform distribution (see Fig. 3a). For a dashed line, 10 knots were put randomly into the scene, then the line is sampled at 10 locations with uniform distribution around each knot, at most 10 pixels far (see Fig. 3b). Then $k$ outliers were added to the scene. 1000 tests were performed on every noise level.

Fig. 2 shows the mean angular error (in degrees) plotted as the function of the noise $\sigma$. The first and second rows report the results of the straight and dashed line cases, respectively. For the two columns, 100 and 500 outliers were added, respectively. According to Fig. 2, GC-RANSAC obtains more accurate lines than the competitor algorithms. Fig. 4 shows an example where GC-RANSAC (blue line) finds the desired model even for unrealistically high outlier level (5000 outliers and $\sigma = 9.0$ pixels).

**Estimation of Epipolar Geometry.** Evaluating the performance of GC-RANSAC on fundamental matrix estimation we used kusvod2 (24 pairs)\(^3\), Multi-H\(^4\) (5 pairs), and AdelaideRMF\(^5\) (19 pairs) datasets (see Fig. 5 for examples). Kusvod2 consists of 24 image pairs of different sizes with point correspondences and fundamental matrices estimated using manually selected inliers. AdelaideRMF and Multi-H consist a total of 24 image pairs with point correspondences, each assigned manually to a homography (or the outlier class). For them, all points which are assigned to

---

\(^{3}\)http://cmp.felk.cvut.cz/data/geometry2view/
\(^{4}\)http://web.eee.sztaki.hu/~dbarath/
\(^{5}\)https://cs.adelaide.edu.au/~hwong/doku.php?id=data

---

**Table 1: Setting for line, homography and fundamental matrix fitting.** Outlier threshold ($\epsilon$), radius of the sphere for proximity computation ($r$), weight of the pair-wise term ($\lambda$).

|   | $\epsilon$ | $r$ | $\lambda$ |
|---|---|---|---|
| L | 0.31 | 20 px | Eq. 4 |

**Table 2: Percentage of “not-all-inlier” minimal samples leading to the correct solution during line (L) and fundamental matrix (F) fitting.** For lines, the average over 1000 runs on three different outlier percentage (100%, 500%, 1000%) and noise levels 0.0 – 9.0 px, thus 15000 runs were performed. For fundamental matrices, the mean of 1000 runs on the AdelaideRMF dataset is shown.

|   | LO | LO$^+$ | LO$'$ | GC |
|---|---|---|---|---|
| L | 6% | 5% | 4% | 15% |
| F | 29% | 30% | 24% | 32% |

---

**Figure 2:** The mean angular error (in degrees) of the obtained 2D lines plotted as the function of noise $\sigma$ (in pixels). On each noise level, 1000 runs were performed. The line type and outlier number is (a) straight line, 100%, (b) straight line, 500% (c) dashed line, 100% and (c) dashed line, 500%. *Best viewed in color.*

**Figure 3:** An example input for (a) straight and (b) dashed lines. The 1000 black points are outliers, the 100 red ones are inliers. *Best viewed in color.*
always obtains the most accurate model for all but one, i.e. EVD dataset consists of short baseline stereo pairs, while the pairs of EVD undergo an extreme view change, i.e. wide baseline. All methods apply the normalized four-point algorithm \[8\] for homography estimation. Therefore, each minimal sample consists of four correspondences.

The two blocks on the bottom of Fig. 3 report the results on two selected image pairs. The last two blocks of Fig. 4 show the mean results computed using all the image pairs of each dataset. It can be seen that GC-RANSAC obtains the most accurate model for all but one, i.e. EVD dataset with time limit, test cases.

**Convergence from a Not-All-Inlier Sample.** Table 2 reports the frequencies when a “not-all-inlier” sample led to the correct model. For lines \((L)\), it is computed using 1000 runs on each outlier (100, 500 and 1000) and noise level (from 0.0 up to 9.0 pixels). Thus 15000 runs were performed. A minimal sample is counted as a “not-all-inlier” if it contains at least one point farther from the ground truth model than the ground truth noise \(\sigma\).

Table 3: Fundamental matrix (blocks 1–2) and homography (blocks 3–4) estimation on selected image pairs. The names of the tests and the investigated properties are shown in the first two columns. The remaining ones report the results with confidence 95\%. Values are computed as the means of 1000 runs. LO reports the number of local optimization steps and the numbers in the brackets are the numbers of graph-cut steps. The geometric error (in pixels) of the estimated model w.r.t. the manually annotated inliers is written in each second row; the mean processing time (in milliseconds) and the required number of samples are also reported in every 3-th and 4-th rows. The geometric error is the Sampson distance for fundamental matrices and re-projection error for homographies.

| F | PLAIN | LO | LO+ | LO' | GC |
|---|-------|----|-----|-----|----|
| Error (px) | 9.23 | 9.52 | 9.21 | 9.44 | 5.00 |
| Time (ms) | 1.55 | 1.43 | 1.30 | 1.21 | 1.57 |
| Samples | 44.00 | 45.00 | 44.00 | 43.00 | 43.00 |

| H | LO | – | 2 | 2 | 2 | 1 (3) |
|---|----|---|---|---|---|-----|
| Error (px) | 0.34 | 0.34 | 0.33 | 0.38 | 0.29 |
| Time (ms) | 64.88 | 63.72 | 65.93 | 57.00 | 38.55 |
| Samples | 1610.00 | 1299.00 | 1330.00 | 1419.00 | 913.00 |

a homography considered as inliers and others as outliers. On total, the proposed method is tested on 48 image pairs from three publicly available datasets for fundamental matrix estimation. All methods apply the 7-point method \[8\] to estimate the fundamental matrix, thus drawing minimal sets of size seven in each RANSAC iteration.

The first two blocks of Table 3 each consisting of four rows report the quality of the epipolar geometry estimation on two selected image pairs computed as the mean of 1000 runs. The first two columns show the name of the tests and the investigated properties of each method. These properties are: (1) LO: the number of applied local optimization steps (graph-cut steps are shown in brackets). (2) Error (px): the geometric error (in pixels) of the obtained model w.r.t. the manually annotated inliers. For fundamental matrices and homographies it is defined as the average Sampson distance and re-projection error, respectively. (3) Time (ms): the mean processing time in milliseconds. (4) Samples: the average number of minimal samples have to be drawn until convergence, basically, the number of RANSAC iterations.

The first three blocks of Table 4 report the mean results for fundamental matrix estimation obtained on each dataset using its all image pairs as the average of 1000 runs. The structure of the table is similar to that of Table 3. It can be clearly seen that for fundamental matrix estimation GC-RANSAC always obtains the most accurate model using less samples than the competitive methods.

**Estimation of Homography.** In order to test homography estimation we downloaded homogr\[6\] (16 pairs) and EVD\[7\] (15 pairs) datasets (see Fig. 6 for examples). Each consists of image pairs of different sizes from 329 \(\times\) 278 up to 1712 \(\times\) 1712 with point correspondences and manually selected inliers – correct point pairs. Homog dataset consists of short baseline stereo pairs, while the pairs of EVD undergo an extreme view change, i.e. wide baseline. All methods apply the normalized four-point algorithm \[8\] for homography estimation. Therefore, each minimal sample consists of four correspondences.

Figure 4: 100 inliers and 5000 outliers. (a) The ground truth line. (b) The obtained line of each method: blue (GC-RANSAC), red (LO-RANSAC), light green (LO\(^+\)-RANSAC), and dark green (LO\(^-\)-RANSAC). The percentage of finding the desired line on 1000 runs: GC-RANSAC 98\%, LO-RANSAC 92\%, LO\(^+\)-RANSAC 94\%, LO\(^-\)-RANSAC 95\%. Best viewed in color.

http://cmp.felk.cvut.cz/data/geometry2view/

http://cmp.felk.cvut.cz/wbs/
For fundamental matrices (F), the frequencies of success from a “not-all-inlier” sample are computed as the mean of 1000 runs on all pairs of the AdelaideRMF dataset. In this dataset, all inliers are labeled manually, thus it is easy to check whether a sample point is inlier or not. A fundamental matrix is considered correct if the mean Sampson distance from all inliers is at most 1.2 times that of the ground truth fundamental matrix (computed using only the inliers).

**Processing Time.** Fig. 7 shows the mean processing times of GC-RANSAC on all line, fundamental matrix and homography fitting tests. The yellow bar reports the time demand of the neighborhood computation – this linearly depends on the point number. The green one is the time demand of the sampling and model fitting step, the red and claret bars show the model verification (support computation) and the proposed local optimization step, respectively. It can be seen that the sampling and model fitting part dominates the process, however, this can be moderated by changing the sampling technique to e.g. PROSAC [3].

**5. Conclusion**

A novel method, called GC-RANSAC was proposed. It runs iteratively the graph cut algorithm in the local optimization (LO) step which is applied after a so-far-the-best model is found. The proposed LO step is conceptually simple, easy to implement, globally optimal and efficient.

We showed experimentally that GC-RANSAC outperforms LO-RANSAC and its state-of-the-art variants in terms of both accuracy and the required number of iterations for line, homography and fundamental matrix estimation on

Figure 5: Fundamental matrix estimation. Results of GC-RANSAC on example pairs from (a) kusvod2, (b) AdelaideRMF, (c) Multi-H datasets. Correspondences are drawn by lines and circles, outliers by black lines and crosses, every third correspondence is drawn. Best viewed in color.

Figure 6: Homography estimation. Results of GC-RANSAC on example pairs from (a) homogr and (b) EVD datasets. Correspondences are drawn by lines and circles, outliers by black lines and crosses, every third correspondence is drawn. Best viewed in color.
Table 4: Fundamental matrix estimation applied to kusvod2 (24 pairs), AdelaideRMF (19 pairs) and Multi-H (4 pairs) datasets, and homography estimation on homogr (16 pairs) and EVD (15 pairs) datasets. The datasets, the problem (F/H), the number of the image pairs (#) and the investigated properties are shown in the first three columns. The next five report the results at 99% confidence with a time limit set to 60 FPS, i.e. the run is interrupted after 1/60 sec. For the remaining columns, there was no time limit but the confidence is set to 95%. Values are the means of 1000 runs. LO is the number of local optimization steps and the numbers of graph-cuts runs are shown in brackets. The geometric error (in pixels) of the estimated model w.r.t. the manually annotated inliers is written in each second row; the mean processing time (in milliseconds) and the required number of samples are reported in every 3-th and 4-th rows. The geometric error is the Sampson distance for fundamental matrices and re-projection error for homographies.

| Dataset   | Problem | # pairs | Neighb. Computation | Model Fitting and Sampling | Model Verification | Graph Cut |
|-----------|---------|---------|----------------------|---------------------------|-------------------|------------|
| kusvod2   | F       | 24      |                      |                           |                   |            |
| Adelaide  | F       | 19      |                      |                           |                   |            |
| Multi-H   | F       | 4       |                      |                           |                   |            |
| EVD       | H       | 15      |                      |                           |                   |            |
| homogr    | H       | 16      |                      |                           |                   |            |
| av. #78   |         |         |                      |                           |                   |            |

| Min. 60 FPS with 99% confidence | Confidence 95% |
|--------------------------------|----------------|
| Time (ms) | Error (px) | Samples | Time (ms) | Error (px) | Samples | Time (ms) | Error (px) | Samples | Time (ms) | Error (px) | Samples |
| F. #24    | LO  | 1       | 1.34     | 113.00   | 113.00   | 122.00   | 116.00   | 70.00    | 93.00    | 76.00     | 78.00   | 87.00   | 53.00   |
|           | LO' | 1       | 1.34     | 113.00   | 113.00   | 122.00   | 116.00   | 70.00    | 93.00    | 76.00     | 78.00   | 87.00   | 53.00   |
|           | LO+ | 1       | 1.34     | 113.00   | 113.00   | 122.00   | 116.00   | 70.00    | 93.00    | 76.00     | 78.00   | 87.00   | 53.00   |
|           | GC  | 1       | 1.34     | 113.00   | 113.00   | 122.00   | 116.00   | 70.00    | 93.00    | 76.00     | 78.00   | 87.00   | 53.00   |

Figure 7: The breakdown of the processing times in milliseconds. Computed as the mean of all tests. Best viewed in color.

standard public datasets.

Benefiting from the capability of modeling spatial coherence, we experimentally verified GC-RANSAC relatively often converges to a “good” model even when starting from a sample that is not all-inlier. GC-RANSAC is very efficient, its processing time for hundreds of input points is approximately 1 – 10 milliseconds, depending on the inlier-outlier ratio.

We will make the optimized C++ source available after publication.

References

[1] D. Barath, J. Matas, and L. Hajder. Multi-H: Efficient recovery of tangent planes in stereo images. In British Machine Vision Conference, 2016.

[2] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In Computer Vision and Pat-
tern Recognition. IEEE, 1998.

[3] O. Chum and J. Matas. Matching with prosac-progressive sample consensus. In Computer Vision and Pattern Recognition. IEEE, 2005.

[4] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In Joint Pattern Recognition Symposium. Springer, 2003.

[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.

[6] V. Fragoso, P. Sen, S. Rodriguez, and M. Turk. Evsac: accelerating hypotheses generation by modeling matching scores with extreme value theory. In International Conference on Computer Vision, 2013.

[7] D. Ghosh and N. Kaabouch. A survey on image mosaicing techniques. Journal of Visual Communication and Image Representation, 2016.

[8] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003.

[9] H. Isack and Y. Boykov. Energy-based geometric multimodel fitting. International Journal of Computer Vision, 2012.

[10] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? Pattern Analysis and Machine Intelligence, 2004.

[11] K. Lebeda, J. Matas, and O. Chum. Fixing the locally optimized ransac–full experimental evaluation. In British Machine Vision Conference. Citeseer, 2012.

[12] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. Image and vision computing, 2004.

[13] D. Mishkin, J. Matas, and M. Perdoch. MODS: Fast and robust method for two-view matching. Computer Vision and Image Understanding, 2015.

[14] D. Nasuto and J. M. B. R. Craddock. Napsac: High noise, high dimensional robust estimation–its in the bag. 2002.

[15] T. T. Pham, T.-J. Chin, K. Schindler, and D. Suter. Interacting geometric priors for robust multimodel fitting. Transactions on Image Processing, 2014.

[16] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In International Conference on Computer Vision. IEEE, 1998.

[17] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm. Usac: a universal framework for random sample consensus. IEEE transactions on pattern analysis and machine intelligence, 35(8):2022–2038, 2013.

[18] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In ACM transactions on graphics (TOG). ACM, 2004.

[19] C. Sminchisescu, D. Metaxas, and S. Dickinson. Incremental model-based estimation using geometric constraints. Pattern Analysis and Machine Intelligence, 2005.

[20] 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.

[21] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding, 2000.

[22] 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.

[23] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, pages II–II. IEEE, 2004.

[24] 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.