Graph-Cut RANSAC

Daniel Barath[12] and Jiri Matas[2]
[1]Machine Perception Research Laboratory, MTA SZTAKI, Budapest, Hungary
[2]Centre for Machine Perception, Czech Technical University, Prague, Czech Republic

Abstract

A novel method for robust estimation, called Graph-Cut RANSAC[1], GC-RANSAC in short, is introduced. To separate inliers and outliers, it runs the graph-cut algorithm in the local optimization (LO) step which is applied when a so-far-the-best model is found. The proposed LO step is conceptually simple, easy to implement, globally optimal and efficient. GC-RANSAC is shown experimentally, both on synthesized tests and real image pairs, to be more geometrically accurate than state-of-the-art methods on a range of problems, e.g. line fitting, homography, affine transformation, fundamental and essential matrix estimation. It runs in real-time for many problems at a speed approximately equal to that of the less accurate alternatives (in milliseconds on standard CPU).

1. Introduction

The RANSAC (RANdom SAmple Consensus) algorithm proposed by Fischler and Bolles [7] 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 [27, 29], wide baseline stereo matching [22, 17, 18], motion segmentation [27], image mosaicing [9], detection of geometric primitives [25], multi-model fitting [31], or for initialization of multi-model fitting algorithms [12, 21]. In brief, the RANSAC approach repeatedly selects random subsets of the input data and fits a model, e.g. a line to two 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.

In the last three decades, many modifications of RANSAC have been proposed. For instance, NAPSAC [20], PROSAC [4] or EVSAC [8] modify the sampling strategy to increase the probability of selecting an all-inlier sample earlier. NAPSAC considers spatial coherence in the sampling of input data points, PROSAC exploits the ordering of the points by their predicted inlier probability, EVSAC uses an estimate of the confidence in each point. Modifications of the model support step has also been proposed. In MLESAC [28] and MSAC [10], the model quality 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 user-defined threshold. The termination of RANSAC is controlled by a manually set confidence value η and the sampling stops when the probability of finding a model with higher support falls below η[2].

Observing that RANSAC requires in practice more samples than theory predicts, Chum et al. [5] 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* model. In the original paper [5], local optimization is implemented as an iterated least squares refitting 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 iterations. 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. [15] that for models with high inlier counts the local optimization step becomes a computational bottleneck due to the iterated least squares model fitting. This is addressed 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 [23]. Nevertheless, the LO proce-

[1] Available at https://github.com/danini/graph-cut-ransac

[2] This interpretation of η holds for the standard cost function only.

dure is ad hoc, complex and needs multiple parameters.

In this paper, we combine two strands of research to obtain a state-of-the-art RANSAC. 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. Spatial coherence, leading to the Potts model [3], has been exploited in many vision problems, for instance, in segmentation [30], multi-model fitting [12, 21] or sampling [20]. In RANSAC techniques, it has only been used to improve efficiency of sampling in NAPSAC [20]. It is computationally prohibitive to formulate the model verification in RANSAC as a graph-cut problem. But when applied as the LO step in [5] just to the *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 achieved in real-time.

The proposed method, called Graph-Cut RANSAC (GC-RANSAC), is a locally optimized RANSAC alternating graph-cut and model re-fitting as the LO step. GC-RANSAC is superior to LO-RANSAC in a number of aspects. First, it is capable of exploiting spatial coherence of inliers and outliers. The LO step is conceptually a simple, easy to implement, globally optimal and computationally efficient graph-cut with only a few intuitive and learnable parameters unlike the ad hoc, iterative and complex LO steps [5]. 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 the 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 outliers.

PEARL [12] introduced pair-wise energy to geometric model fitting. However, it cannot be used for problems solved by RANSAC – in PEARL, the user has to manually set the number of hypotheses tested to the worst-case, i.e. corresponding to the lowest inlier ratio possible. The α-expansion step just in the first iteration of PEARL executes a 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 determined by the RANSAC adaptive termination criterion. Moreover, in GC-RANSAC, applying the local optimization to only the *so-far-the-best* models ensures that the graph-cut is executed only very few times, paying only a small penalty.

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.

2.1. Formulation as Energy Minimization

Suppose that a point set $\mathcal{P} \subseteq \mathbb{R}^n$ $(n > 0)$, 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 top-hat fitness function (1 – close, 0 – far), the implied unary energy is as follows:

$$E_{\{0;1\}}(L) = \sum_{p \in \mathcal{P}} ||L_p||_{\{0;1\}},$$

where

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

Parameter $L \in \{0, 1\}^{|\mathcal{P}|}$ is a labeling, ignored in standard RANSAC, $L_p \in L$ is the label of point $p \in \mathcal{P}$, $|\mathcal{P}|$ is the number of points, and ϵ is the inlier-outlier threshold. Using energy $E_{\{0,1\}}$ we get the same result as RANSAC since it 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 ϵ. This is exactly what RANSAC does.

Since the publication of RANSAC, several papers discussed, e.g. [15], 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 [28]. This improves the accuracy and reduces the sensitivity to threshold ϵ. Unary term E_K exploiting this continuous loss is as follows: $E_\text{K}(L) = \sum_{p \in \mathcal{P}} ||L_p||_\text{K}$, where

$$||L_p||_\text{K} = \begin{cases} 1 - K(\phi(p, \theta), \epsilon) & \text{if } L_p = 1 \\ K(\phi(p, \theta), \epsilon) & \text{if } L_p = 0 \end{cases} \tag{1}$$

and

$$K(\delta, \epsilon) = e^{-\frac{\delta^2}{2\epsilon^2}}, \tag{2}$$

which equals to one if the distance is zero. 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, additional energy terms, e.g. to consider spatial coherence of the points, can be included yet keep the problem *solvable efficiently and globally via the graph-cut algorithm.*

Considering point proximity is a well-known approach for sampling [20] or multi-model fitting [12, 21, 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 term. The problem arises when the data contains significantly more outliers 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 penalties for each neighboring point pair on the basis of their inlier probability. The proposed pair-wise energy term is

$$E_{\mathrm{S}}(L) = \sum_{(p,q)\in\mathcal{A}} \begin{cases} 1 & \text{if } L_p \neq L_q \\ \frac{1}{2}(K_p + K_q) & \text{if } L_p = L_q = 0 \\ 1 - \frac{1}{2}(K_p + K_q) & \text{if } L_p = L_q = 1 \end{cases}, \tag{3}$$

where $K_p = K(\phi(p, \theta), \epsilon)$, $K_q = K(\phi(q, \theta), \epsilon)$ and (p, q) is an edge of neighborhood graph $\mathcal{A}$ between points p and q. In E_{S}, if both points are labeled as outliers 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 $E(L) = E_{\mathrm{K}}(L) + \lambda E_{\mathrm{S}}(L)$, where λ 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.

3. GC-RANSAC

In this section, we include the proposed energy minimization-based local optimization into RANSAC. Benefiting from this approach, the LO step is 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 $\mathcal{A}$ for which we use a sphere with a predefined radius r – this is a parameter of the algorithm – and Fast Approximate Nearest Neighbors algorithm [19]. In Alg. 1, function H is as follows [7]:

$$H(|L^*|, \mu) = \frac{\log(\mu)}{\log(1 - P_I)}, \tag{4}$$

where $P_I = \binom{|L^*|}{m} / \binom{|P|}{m}$. It calculates the required iteration number of RANSAC on the basis of desired probability μ, 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 kth iteration draws a minimal sample using a sampling strategy, e.g. PROSAC [4], then computes the parameters θ_k of the implied model and its support

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

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*, all parameters are updated, i.e. the labeling, model parameters and support, and local optimization is applied if needed. Note that the application criterion of the local optimization step is discussed later.

The proposed local optimization is written in Alg. 2. The main iteration can be considered as a grab-cut-like [24] alternation consisting of two major steps: (i) graph-cut and (ii) 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 add unary (Eq. 1) and binary (Eq. 3) costs, respectively, to the problem graph. Such graph construction is covered in depth in [13] (Section 4). Graph-cut is applied to G determining the optimal labeling L which considers the spatial coherence of the points and their distances from the *so-far-the-best* model. Model parameters θ are computed using a $7m$-sized random subset of the inliers in L, thus speeding up the process, similarly to [15] 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 [15] and this value also suited for us. Finally, the support w of θ 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 performed if it has not been yet applied to the obtained *so-far-the-best* model. Then the model parameters are re-estimated using the whole inlier set similarly to what plain RANSAC does.

Remark: Adding to the local optimization step a RANSAC-like procedure selecting $7m$-size samples is straightforward. In our experiments, it had a high computational overhead without adding significantly to accuracy.

The criterion for applying the LO step was proposed to be: (i) the model is so-far-the-best and (ii) after a user-defined iteration limit, in [15]. However, in our experiments, this approach still spends significant time on optimizing models which are not promising enough. We introduce a simple heuristic for replacing the iteration limit with a data driven strategy which allows to apply LO only a few times without deterioration in accuracy.

Algorithm 1 The GC-RANSAC Algorithm.

```
Input: $\mathcal{P}$ – data points; $r$ – sphere radius, $\epsilon$ – threshold
        $\epsilon_{\text{conf}}$ – LO application threshold, $\mu$ – confidence;
Output: $\theta$ - model parameters; $L$ – labeling

$w^*, n_{LO} \leftarrow 0, 0$.
$\mathcal{A} \leftarrow$ Build neighborhood-graph using $r$.
for k = $1 \rightarrow H(|L^*|, \mu)$ do                ▷ Eq. 4
    $S_k \leftarrow$ Draw a minimal sample.
    $\theta_k \leftarrow$ Estimate a model using $S_k$.
    $w_k \leftarrow$ Compute the support of $\theta_k$.      ▷ Eq. 5
    if $w_k > w^*$ then
        $\theta^*, L^*, w^* \leftarrow \theta_k, L_k, w_k$
        if $\mu_{12} > \epsilon_{conf}$ then                 ▷ Eq. 6
            $\theta_{LO}, L_{LO}, w_{LO} \leftarrow$ Local opt.   ▷ Alg. 2
            $n_{LO} \leftarrow n_{LO} + 1$.
            if $w_{LO} > w^*$ then
                $\theta^*, L^*, w^* \leftarrow \theta_{LO}, L_{LO}, w_{LO}$
if $n_{LO} = 0$ then
    $\theta^*, L^*, w^* \leftarrow$ Local opt.               ▷ Alg. 2
$\theta^* \leftarrow$ least squares model fitting using $L^*$.
```

Algorithm 2 Local optimization.

```
Input: $\mathcal{P}$ – data points, $L^*$ – labeling,
        $w^*$ – support, $\theta^*$ – model;
Output: $L^*_{LO}$ – labeling, $w^*_{LO}$ – support, $\theta^*_{LO}$ – model;

$w^*_{LO}, L^*_{LO}, \theta^*_{LO}, changed \leftarrow w^*, L^*, \theta^*, 1$.
while $changed$ do
    $G \leftarrow$ Build the problem graph.                 ▷ Alg. 3
    $L \leftarrow$ Apply graph-cut to $G$.
    $I_{7m} \leftarrow$ Select a $7m$-sized random inlier set.
    $\theta \leftarrow$ Fit a model using labeling $I_{7m}$.
    $w \leftarrow$ Compute the support of $\theta$.
    $changed \leftarrow 0$.
    if $w > w^*_{LO}$ then
        $\theta^*_{LO}, L^*_{LO}, w^*_{LO}, changed \leftarrow \theta, L, w, 1$.
```

Algorithm 3 Problem Graph Construction.

```
Input: $\mathcal{P}$ – data points, $\mathcal{A}$ – neighborhood-graph
        $\theta$ – model parameters, $\theta^*$ – model;
Output: $G$ – problem graph;

$G \leftarrow$ EmptyGraph().
for $p \in \mathcal{P}$ do
    $c_0, c_1 \leftarrow \text{K}(\phi(p, \theta), 1 - \text{K}(\phi(p, \theta), \epsilon)$
    $G \leftarrow$ AddTerm1($G$, $p$, $c_0$, $c_1$).
for $(p, q) \in \mathcal{A}$ do
    $c_{01}, c_{10} \leftarrow 1, 1$.
    $c_{00} \leftarrow 0.5(\text{K}(\phi(q, \theta) + \text{K}(\phi(p, \theta))$.
    $c_{11} \leftarrow 1 - 0.5(\text{K}(\phi(q, \theta) + \text{K}(\phi(p, \theta))$.
    $G \leftarrow$ AddTerm2($G$, $p$, $q$, $c_{00}$, $c_{01}$, $c_{10}$, $c_{11}$).
```

As it is well-known for RANSAC, the required iteration number k, w.r.t. the inlier ratio η, sample size m and confidence μ, is calculated as $k = \log(1 - \mu)/\log(1 - \eta^m)$. Re-arranging this formula to μ leads to equation $\mu = 1 - 10^{k \log(1 - \eta^m)}$ which determines the confidence of finding the desired model in the kth iteration if the inlier ratio is η.

Suppose that the algorithm finds a new so-far-the-best model with inlier ratio η_2 in the k_2th iteration, whilst the previous best model was found in the k_1th iteration with inlier ratio η_1 ($k_2 > k_1$, $\eta_2 > \eta_1$). The ratio of the confidences μ_{12} in those two models is calculated as follows:

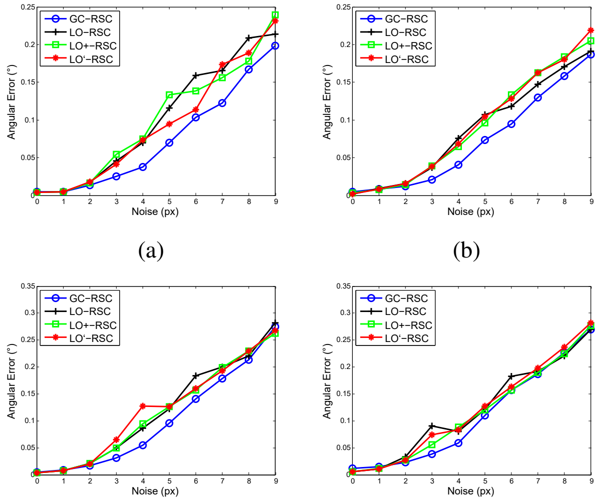
$$\mu_{12} = \frac{\mu_2}{\mu_1} = \frac{1 - 10^{k_2 \log(1 - \eta_2^m)}}{1 - 10^{k_1 \log(1 - \eta_1^m)}}. \tag{6}$$

In experiments, we observed that a model that leads to termination if optimized often shows a significant increase in the confidence. Replacing the parameter blocking LO in the first k iterations, we adopt a criterion $\mu_{12} > \epsilon_{\text{conf}}$, where ϵ_{conf} is a user-defined parameter determining a significant increase.

4. Experimental Results

In this section, GC-RANSAC is validated both on synthesized and publicly available real world data and compared with plain RANSAC [7], LO-RANSAC [5], LO$^+$-RANSAC, LO'-RANSAC [15], and EP-RANSAC [14]. For EP-RANSAC, we tuned the threshold parameter to achieve the lowest mean error and the other parameters were set to the values reported by the authors. Note that the comparison of the processing time with this method is affected by the availability of a Matlab implementation only. All methods apply PROSAC [4] sampling and use MSAC-like truncated quadratic distances with threshold set to $\epsilon = 0.3$ pixels (similarly as in [15]). EP-RANSAC uses inlier maximization strategy since its cost function cannot be replaced straightforwardly. The radius of the sphere to determine neighboring points is 20 pixels and it is applied to the concatenated 4D coordinates of the correspondences. Parameter λ for GC-RANSAC was set to 0.1 and $\epsilon_{\text{conf}} = 0.1$.

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 600×600 window and a random line was generated in its implicit form, sampled at 100 locations and zero-mean Gaussian-noise with σ standard deviation was added to the coordinates. For a straight line, the points were generated using uniform distribution (see Fig. 2a). For a dashed line, 10 knots were put randomly into the window, then the line is sampled at 10 locations with uniform distribution around each knot, at most 10 pixels far (see Fig. 2b). Finally, k outliers were added to the scene. 1000 tests were performed on every noise level.



(a) (b)

(c) (d)

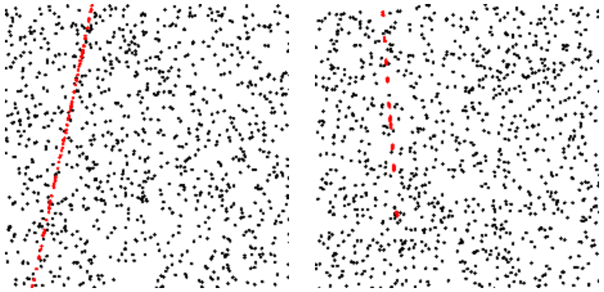
Figure 1: The mean angular error (in degrees) of the obtained 2D lines plotted as the function of noise σ (in pixels). On each noise σ, 1000 runs were performed. The line type and outlier number is (a) straight line, 100%, (b) straight line, 500% (c) dashed line, 100% and (c) dashed line, 500%.

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

Table 1: 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 is reported, thus 15000 runs were performed. For **F**, the mean of 1000 runs on the `AdelaideRMF` dataset is shown.

Fig. 1 shows the mean angular error (in degrees) plotted as the function of the noise σ. The first and second rows report the results of the straight and dashed line cases. For the two columns, 100 and 500 outliers were added, respectively. According to Fig. 1, *GC-RANSAC obtains more accurate lines than the competitor algorithms*.

Estimation of Fundamental Matrix. 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. 3 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 a homography were considered as inliers and others as outliers. In total, the proposed method was tested on 48 image pairs from three publicly available datasets for fundamental matrix estimation. All methods applied the 7-point method [10] to estimate **F**, thus drawing minimal sets of size seven in each RANSAC iteration. For the model re-estimation from a non-minimal sample in the LO step, the normalized 8-point algorithm [11] is used. Note that all fundamental matrices were discarded for which the *oriented* epipolar constraint [6] did not hold.



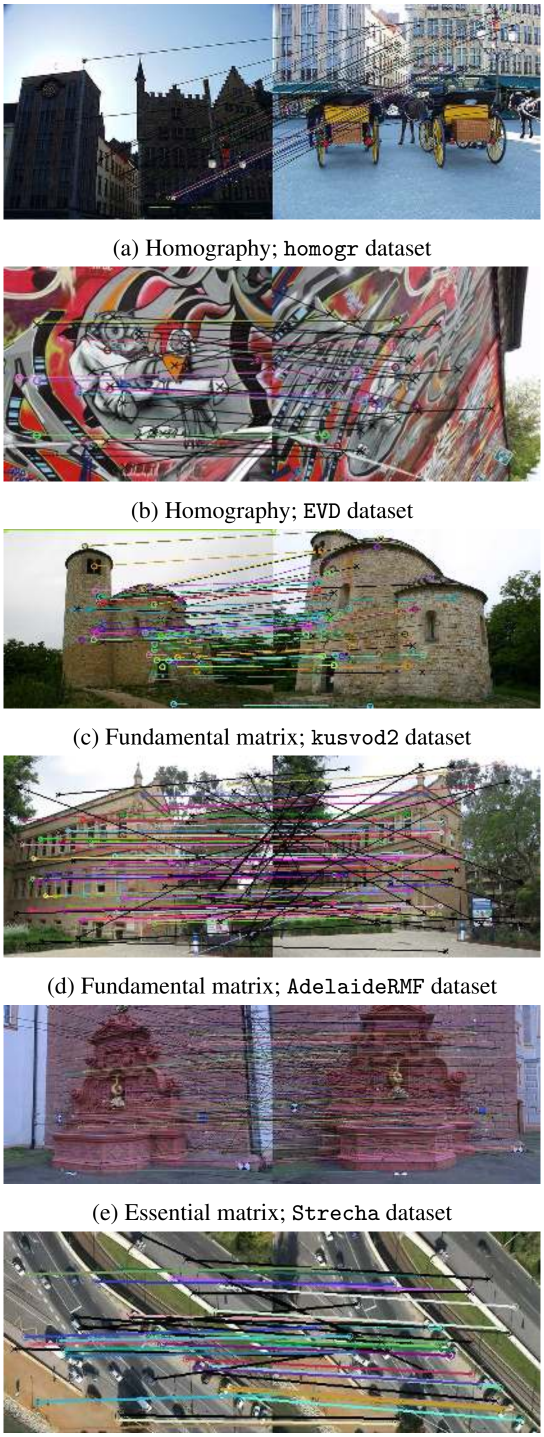
(a) (b)

Figure 2: 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.*

The first three blocks of Table 2, each consisting of four rows, report the quality of the epipolar geometry estimation on each dataset as the average of 1000 runs on every image pair. The first two columns show the name of the tests and the investigated properties: **(1)** LO: the number of applied local optimization steps (graph-cut steps are shown in brackets). **(2)** $\mathcal{E}$ is 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. For essential matrices, it is the mean Sampson distance of the implied fundamental matrix and the correspondences. **(3)** $\mathcal{T}$ is the mean processing time in milliseconds. **(4)** $\mathcal{S}$ is the average number of minimal samples have to be drawn until convergence, basically, the number of RANSAC iterations.

It can be clearly seen that for fundamental matrix estimation GC-RANSAC *always obtains the most accurate model* using fewer samples than the competitive methods.

[3] http://cmp.felk.cvut.cz/data/geometry2view/
[4] http://web.eee.sztaki.hu/ dbarath/
[5] cs.adelaide.edu.au/ hwong/doku.php?id=data



(a) Homography; `homogr` dataset

(b) Homography; `EVD` dataset

(c) Fundamental matrix; `kusvod2` dataset

(d) Fundamental matrix; `AdelaideRMF` dataset

(e) Essential matrix; `Strecha` dataset

(f) Affine transformation; `SZTAKI` dataset

Figure 3: Results of GC-RANSAC on example pairs from each dataset and problem. Correspondences are drawn by lines and circles, outliers by black lines and crosses, every third correspondence is drawn.

Estimation of Homography. In order to test homography estimation we downloaded `homogr`[6] (16 pairs) and `EVD`[7] (15 pairs) datasets (see Fig. 3 for examples). Each consists of image pairs of different sizes from 329×278 up to 1712×1712 with point correspondences and manually selected inliers – correctly matched point pairs. `Homogr` dataset consists of short baseline stereo pairs, whilst the pairs of `EVD` undergo an extreme view change, i.e. wide baseline. All methods apply the normalized four-point algorithm [10] for homography estimation both in the model generation and local optimization steps. Therefore, each minimal sample consists of four correspondences.

The 4th and 5th blocks of Table 2 show the mean results computed using all the image pairs of each dataset. It can be seen that GC-RANSAC obtains the most accurate models for all but one, i.e. `EVD` dataset with time limit, test cases.

Estimation of Essential Matrix. To estimate essential matrices, we used the `strecha` dataset [26] consisting of image sequences of buildings. All image sizes are 3072×2048. The ground truth projection matrices are provided. The methods were applied to all possible image pairs in each sequence. The SIFT detector [16] was used to obtain correspondences. For each image pair, a reference point set with ground truth inliers was obtained by calculating the fundamental matrix from the projection matrices [10]. Correspondences were considered as inliers if the symmetric epipolar distance was smaller than 1.0 pixel. All image pairs with less than 20 inliers found were discarded. In total, 467 image pairs were used in the evaluation.

The results are reported in the 6th block of Table 2. The reason of the high processing time is that the mean inlier ratio is relatively low (27%) and there are many correspondences, 2323, on average. GC-RANSAC obtains the most accurate essential matrices both in the wall-clock time limited and solution confidence above 95% experiments. A significant drop can be seen in accuracy for all methods if a time limit is given.

Estimation of Affine Transformation. The `SZTAKI Earth Observation` dataset[8] [2] (83 image pairs of size 320×240) was used to test affine transformation estimation. The dataset contains images of busy road scenes taken from a balloon. Due to the altitude of the balloon, the image pair relation is well approximated by an affine transformation. Point correspondences were detected by the SIFT detector. For ground truth, 20 inliers were selected manually. Point pairs with distance from the ground truth affine transformation lower than 1.0 pixel were defined as inliers.

[6] http://cmp.felk.cvut.cz/data/geometry2view/

[7] http://cmp.felk.cvut.cz/wbs/

[8] http://mplab.sztaki.hu/remotesensing

The estimation results are shown in the 7th block of Table 2. The reported error is $|\mathbf{A}\mathbf{p}_1 - \mathbf{p}_2|$, where $\mathbf{A}$ is the estimated affine transformation and $\mathbf{p}_k$ is the point in the kth image ($k \in \{1, 2\}$). The methods obtained fairly similar results, however, GC-RANSAC is slightly more accurate. It is marginally slower due to the neighborhood computation. However, it is still faster than real time.

Convergence from a Not-All-Inlier Sample. Table 1 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 σ.

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.

Evaluation of the λ setting. To evaluate the effect of the λ parameter balancing the spatial coherence term, we applied GC-RANSAC to all problems with varying λ. The evaluated values are: (i) $\lambda = 0$, which turns off the spatial coherence term, (ii) $\lambda = 0.1$, (iii) $\lambda = 1$, (iv) $\lambda = 10$, and (v) $\lambda = 100$. Fig. 4a shows the ratio of the geometric errors for $\lambda \neq 0$ and $\lambda = 0$ (in percent). For all investigated non-zero λ values, the error is lower than for $\lambda = 0$. Since $\lambda = 0.1$ led to the most accurate results on average, we chose this setting in the tests.

Evaluation of the criterion for the local optimization. The proposed criterion (Eq. 6) ensuring that local optimization is applied only to the most promising model candidates is tested in this section. We applied GC-RANSAC to all problems combined with the proposed and the standard approaches. The standard technique sets an iteration limit (default value: 50) and the LO procedure is afterwards applied to all models that are so far the best. Fig. 4b reports the ratio of each property (processing time – dark blue, LO – light blue, and GC steps – yellow, geometric error – brown) of the proposed and standard approaches. The new criterion leads to significant improvement in the processing time with no deterioration in accuracy.

Processing Time. Fig. 4c shows the breakdown of the processing times of GC-RANSAC applied to each problem. The time demand of the neighborhood computation (dark blue) linearly depends on the point number. The light blue one is the time demand of the sampling and model fitting step, the yellow and brown bars show the model verification (support computation) and the proposed local optimization step, respectively. The sampling and model fitting part dominates the process.

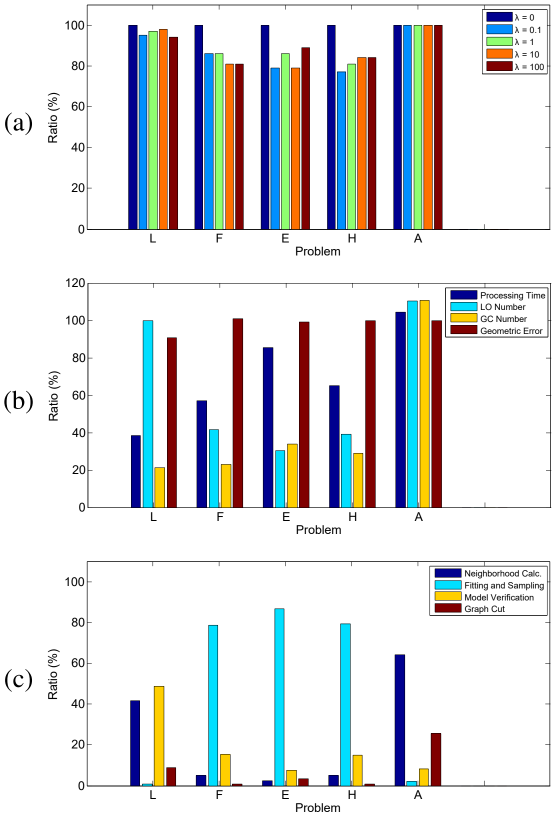


Figure 4: (**a**) The effect of the λ choice weighting the spatial term. The ratio of the geometric error (in percentage) compared to the $\lambda = 0$ case (no spatial coherence) for each problem (**L** – lines, **F** – fundamental matrix, **E** – essential matrix, **H** – homography, **A** – affine transformation). (**b**) The effect of replacing the iteration limit before the first LO applied with the proposed criterion, i.e. the confidence radically increases. The ratios (in percentage) of each property of the proposed and that of standard approaches. (**c**) The breakdown of the processing times in percentage w.r.t. the total runtime. All values were computed as the mean of all tests. *Best viewed in color.*

5. Conclusion

GC-RANSAC was presented. It is more geometrically accurate than state-of-the-art methods. It runs in real-time for many problems at a speed approximately equal to the less accurate alternatives. It is much simpler to implement in a reproducible manner than any of the competitors (RANSAC's with local optimization). Its local optimiza-

			Approx. 60 FPS (or 99% confidence)					Confidence 95%						
			RSC	LO	LO$^+$	LO'	**GC**	RSC	LO	LO$^+$	LO'	EP-RSC	**GC**
kusvod2	**F**, #24	LO	–	2	2	2	1 (3)	–	1	1	1	–	2 (3)
		$\mathcal{E}$	5.01	4.95	4.97	5.02	**4.65**	5.18	5.08	5.03	5.22	7.87	**4.69**
		$\mathcal{T}$	6.2	6.1	6.3	5.9	**4.6**	4.9	5.2	5.1	4.9	439.9	**3.6**
		$\mathcal{S}$	117	96	99	111	**70**	93	76	78	87	–	**53**
Adelaide	**F**, #19	LO	–	2	2	2	1 (3)	–	2	2	3	–	2 (4)
		$\mathcal{E}$	0.55	0.53	0.52	0.55	**0.50**	0.44	0.45	**0.43**	0.44	0.71	**0.43**
		$\mathcal{T}$	14.2	14.8	14.9	**14.1**	18.9	262.7	**194.2**	210.9	237.1	2 121.9	227.1
		$\mathcal{S}$	124	**113**	**113**	122	116	1 363	1 126	1 205	1 305.00	–	**1 115**
Multi-H	**F**, #4	LO	–	1	1	1	1 (3)	–	2	1	2	–	1 (3)
		$\mathcal{E}$	0.35	0.34	0.34	0.34	**0.32**	0.33	0.33	0.33	0.34	0.44	**0.32**
		$\mathcal{T}$	**10.3**	11.5	11.1	**10.3**	14.6	12.8	15.1	14.1	**12.4**	2 371.8	36.0
		$\mathcal{S}$	83	76	76	82	**74**	107	89	90	100	–	**78**
EVD	**H**, #15	LO	–	2	2	2	2 (2)	–	4	4	4	–	3 (6)
		$\mathcal{E}$	1.53	1.63	**1.51**	1.58	1.53	0.96	0.95	0.95	0.96	1.17	**0.92**
		$\mathcal{T}$	16.8	18.3	18.0	**16.8**	19.2	247.3	248.0	251.3	**247.0**	$> 10^4$	249.9
		$\mathcal{S}$	320	**298**	301	318	301	4 303	**4 203**	4 248	4 291	–	4 204
homogr	**H**, #16	LO	–	2	2	2	1 (3)	–	2	2	2	–	1 (4)
		$\mathcal{E}$	0.53	0.53	0.53	0.53	**0.51**	0.50	0.50	0.49	0.50	0.58	**0.47**
		$\mathcal{T}$	7.1	10.4	9.8	**7.1**	7.6	17.1	10.1	9.9	8.5	3 339.7	**7.9**
		$\mathcal{S}$	193	175	175	189	**159**	450	212	214	226	–	**165**
strecha	**E**, #467	LO	–	1	1	1	1 (1)	–	7	7	7	–	7 (7)
		$\mathcal{E}$	11.81	12.34	12.07	12.12	**11.6**	3.03	2.95	2.94	2.87	3.32	**2.83**
		$\mathcal{T}$	11.6	17.3	**17.2**	**17.2**	17.3	3 581.9	3 638.5	3 648.4	3 570.0	$> 10^6$	**3 466.4**
		$\mathcal{S}$	31	**30**	31	31	**30**	3 654	3 646	**3 634**	3 653	–	3 651
SZTAKI	**A**, #52	LO	–	1	1	1	1 (3)	–	1	1	1	–	1 (3)
		$\mathcal{E}$	0.41	0.41	0.41	0.41	**0.40**	0.45	0.46	0.44	0.45	0.48	**0.41**
		$\mathcal{T}$	3.5	**3.2**	**3.2**	**3.2**	10.3	**1.7**	**1.7**	**1.7**	**1.7**	4 718.2	10.2
		$\mathcal{S}$	26	26	26	26	26	**9**	**9**	**9**	**9**	–	**9**

Table 2: Fundamental matrix estimation applied to kusvod2 (24 pairs), AdelaideRMF (19 pairs) and Multi-H (4 pairs) datasets, homography estimation on homogr (16 pairs) and EVD (15 pairs) datasets, essential matrix estimation on the strecha dataset (467 pairs), and affine transformation estimation on the SZTAKI Earth Observation benchmark (52 pairs). Thus the methods were tested on total on 597 image pairs. The datasets, the problem ($\mathbf{F}/\mathbf{H}/\mathbf{E}/\mathbf{A}$), the number of the image pairs ($\#$) and the reported 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 $1/60$ secs (EP-RANSAC is removed since it cannot be applied in real time). For the other columns, there was no time limit but the confidence was set to 95%. Values are the means of 1000 runs. LO is the number of local optimizations and the number of graph-cut runs are shown in brackets. The geometric error ($\mathcal{E}$, in pixels) of the estimated model w.r.t. the manually selected inliers is written in each second row; the mean processing time ($\mathcal{T}$, in milliseconds) and the required number of samples ($\mathcal{S}$) are written in every 3th and 4th rows. The geometric error is the Sampson distance for **F** and **E**, and the projection error for **H** and **A**.

tion step is globally optimal for the so-far-the-best model parameters. We also proposed a criterion for the application of the local optimization step. This criterion leads to a significant improvement in processing time with no deterioration in accuracy. GC-RANSAC can be easily inserted into USAC [23] and be combined with its ”bells and whistles“ like PROSAC sampling, degeneracy testing and fast evaluation with early termination.

Acknowledgement

D. Barath was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00001) and the Hungarian National Research, Development and Innovation Office grant VKSZ 14-1-2015-0072. J. Matas was supported by the Czech Science Foundation Project GACR P103/12/G084.

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] C. Benedek and T. Szirányi. Change detection in optical aerial images by a multilayer conditional mixed markov model. *Transactions on Geoscience and Remote Sensing*, 2009.

[3] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Computer Vision and Pattern Recognition*. IEEE, 1998.

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

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

[6] O. Chum, T. Werner, and J. Matas. Epipolar geometry estimation via RANSAC benefits from the oriented epipolar constraint. In *International Conference on Pattern Recognition*, 2004.

[7] 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.

[8] 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.

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

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

[11] R. I. Hartley. In defense of the eight-point algorithm. *Transactions on Pattern Analysis and Machine Intelligence*, 1997.

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

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

[14] H. Le, T.-J. Chin, and D. Suter. An exact penalty method for locally convergent maximum consensus.

[15] K. Lebeda, J. Matas, and O. Chum. Fixing the locally optimized ransac. In *British Machine Vision Conference*. Citeseer, 2012.

[16] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer vision*. IEEE, 1999.

[17] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 2004.

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

[19] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *International Conference on Computer Vision Theory and Applications*, 2009.

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

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

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

[23] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm. USAC: a universal framework for random sample consensus. *Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[24] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *Transactions on Graphics*. ACM, 2004.

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

[26] C. Strecha, R. Fransens, and L. Van Gool. Wide-baseline stereo from multiple views: a probabilistic account. In *Conference on Computer Vision and Pattern Recognition*. IEEE, 2004.

[27] P. H. S. Torr and D. W. Murray. Outlier detection and motion segmentation. In *Optical Tools for Manufacturing and Advanced Automation*. International Society for Optics and Photonics, 1993.

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

[29] P. H. S. Torr, A. Zisserman, and S. J. Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *Computer Vision and Image Understanding*, 1998.

[30] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In *Conference onComputer Vision and Pattern Recognition*. IEEE, 2004.

[31] M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multiransac algorithm and its application to detect planar homographies. In *International Conference on Image Processing*. IEEE, 2005.