USACv20: robust essential, fundamental and homography matrix estimation

Maksym Ivashechkin[1], Daniel Barath[12], and Jiri Matas[1]

[1] Centre for Machine Perception, Czech Technical University in Prague, Czech Republic

[2] Machine Perception Research Laboratory, MTA SZTAKI, Budapest, Hungary

{ivashmak, matas}@cmp.felk.cvut.cz   barath.daniel@sztaki.mta.hu

Abstract. *We review the most recent RANSAC-like hypothesize-and-verify robust estimators. The best performing ones are combined to create a state-of-the-art version of the Universal Sample Consensus (USAC) algorithm. A recent objective is to implement a modular and optimized framework, making future RANSAC modules easy to be included. The proposed method, USACv20, is tested on eight publicly available real-world datasets, estimating homographies, fundamental and essential matrices. On average, USACv20 leads to the most geometrically accurate models and it is the fastest in comparison to the state-of-the-art robust estimators. All reported properties improved performance of original USAC algorithm significantly. The pipeline will be made available after publication.*

1. Introduction

The RANdom SAmple Consensus (RANSAC) algorithm [12] has been one of the most widely used robust estimators in computer vision. RANSAC and many of its variants have been successfully applied to a wide range of vision tasks, for instance, short baseline stereo [37, 39], motion segmentation [37], detection of geometric primitives [31], wide baseline matching [27, 21, 22], in structure-from-motion [1, 40, 30] (SfM) or simultaneous localization and mapping [11, 23] (SLAM) pipelines, image mosaicing [14], and to perform [41] or initialize multi-model fitting [16, 26].

In this paper, we review some of the most recent RANSAC modifications, combine them together and propose a state-of-the-art variant of the Universal Sample Consensus [28] (USAC) algorithm. Also, an important objective is to make the implemented modular and optimized C++ framework publicly avail-
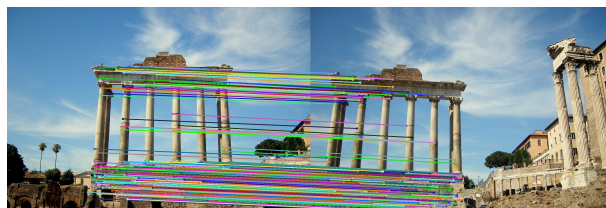
(a) `Community Photo Collection dataset` [40].
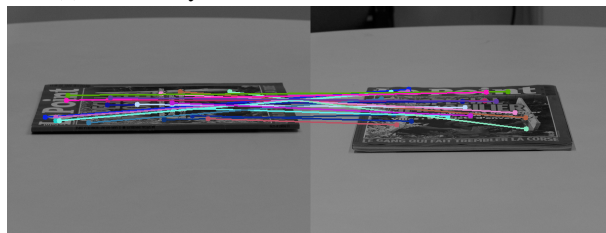
(b) `ExtremeView dataset` [22].

(c) `Tanks and Temples dataset` [17].

(d) `Piccadilly dataset` [40].

Figure 1. Example image pairs where USACv20 has lower error to ground truth inliers than OpenCV RANSAC and USAC [28] estimators.

able, therefore, making future RANSAC modules easy to be combined with the proposed USACv20.

In short, the RANSAC approach repeatedly creates minimal sets of randomly selected points and fits a model to them, e.g., a circle to three 2D points

or a homography to four 2D point correspondences. Next, the quality of the estimated model is measured, for example, by the cardinality of its support, i.e., the number of data points closer than a manually set inlier-outlier threshold. Finally, the model with the highest score, polished, e.g., by least squares fitting of all inliers, is returned.

Scoring function. Many modifications have been proposed since the publication of RANSAC, improving the components of the algorithm. For instance, in MAPSAC [36], 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. MLESAC [38] estimates the model quality by a maximum likelihood process with all its beneficial properties, albeit under certain assumptions about data distributions. In practice, MLESAC results are often superior to the inlier counting of plain RANSAC, and are less sensitive to the inlier-outlier threshold defined manually.

Local Optimization. Observing that RANSAC requires in practice more samples than theory predicts, Chum et al. [8, 18] 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 addressed the problem by introducing the locally optimized RANSAC that augments the original approach with a local optimization step applied to the *so-far-the-best* model. This approach had been further improved in Graph-Cut RANSAC [3] considering the fact that real-world data often form spatially coherent structures. Graph-Cut RANSAC exploits the proximity of the points in the local optimization step, leading to results superior to LO-RANSAC in terms of geometric accuracy.

Sampling Strategies. Samplers NAPSAC [24] and PROSAC [6] modify the RANSAC sampling strategy to increase the probability of selecting an all-inlier sample early. PROSAC exploits an a priori predicted inlier probability rank of the points and starts the sampling with the most promising ones. PROSAC and other RANSAC-like samplers treat models without considering that inlier points often are in the proximity of each other. This approach is effective when finding a global model with inliers sparsely distributed in the scene, for instance, the rigid motion induced by changing the viewpoint in two-view matching. However, as it is often the case in real-world data, if the model is localized with in-

lier points close to each other, robust estimation can be significantly sped up by exploiting this in the sampling. NAPSAC assumes that inliers are spatially coherent. It draws samples from a hyper-sphere centered at the first, randomly selected, point. If this point is an inlier, the rest of the points sampled in its proximity are more likely to be inliers than the points outside the ball. Progressive NAPSAC [2] was proposed to combine NAPSAC-like localized sampling with PROSAC by drawing minimal samples from gradually growing neighborhoods.

Optimizing Model Verification. One of the most successful improvement for speeding up the verification is the optimal randomized model verification strategy [20, 7] (WaldSAC) based on Wald's theory of sequential decision making. When the level of outlier contamination is known a priori, the WaldSAC strategy is provably optimal. In practice, however, inlier ratios have to be estimated during the evaluation process and WaldSAC adjusted to the current so-far-the-best model. The performance of the SPRT test is not significantly affected by the imperfect estimation of these parameters.

Termination criterion. There were a number of different termination criteria proposed for RANSAC-like hypothesize-and-verify methods. The original criterion is based on the assumption that the inliers are noise-free. The number of iterations required is calculated from the inlier ratio and the number of points needed for the model estimation. This criterion was then relaxed by Progressive NAPSAC [2] by terminating if the probability of finding a model which has significantly more inliers than the previous best falls below a threshold. In [6], another criterion was proposed. The PROSAC algorithm terminates if the number of inliers satisfies the following conditions: (i) non-randomness – the probability that i^* out of n data points are by chance inliers to an arbitrary incorrect model is smaller than a threshold; (ii) maximality – the probability that a solution with more than i^* inliers exists and was not found after k samples is smaller than μ_0.

2. USACv20

The structure of the proposed framework is summarized in Algorithm 1. The standard RANSAC loop is executed between lines 2: and 27:. The implementation is modular, and each step of the algorithm allows a range of options.

In the version of USACv20 evaluated in the paper,

Algorithm 1 USACv20.

Input: $\mathcal{P}$ – points; η – confidence, t – maximum iterations, $\mathcal{T}$ – termination, ...

Output: $\hat{\theta}^*$ – the best found model

1: $\varepsilon^* \leftarrow \infty$
2: **while** ! *terminate* $(\mathcal{T}, \eta, t)$ **do**
3:      $\mathcal{S} \leftarrow$ *sampling* $(\mathcal{P})$
4:      **if** ! *validate_sample* $(\mathcal{S})$ **then**
5:          ***continue***
6:      $\hat{\Theta} \leftarrow$ *estimate* $(\mathcal{S})$
7:      **for** $\hat{\theta} \in \hat{\Theta}$ **do**
8:          **if** ! *validate_model* $(\hat{\theta}, \mathcal{S})$ **then**
9:              ***continue***
10:          **if** ! *preemptive_verification*($\hat{\theta}$) **then**
11:              ***continue***
12:          $\varepsilon \leftarrow$ *model_quality*($\hat{\theta}$)
13:          **if** $\varepsilon^* \prec \varepsilon$ **then**
14:              $\hat{\theta}' \leftarrow$ *recover_if_degenerate* $(\hat{\theta}, \mathcal{S})$
15:              **if** $\hat{\theta}' =$ NULL **then**
16:                  ***continue***
17:              $\varepsilon' \leftarrow$ *model_quality* $(\hat{\theta}')$
18:              **if** $\varepsilon^* \prec \varepsilon'$ **then**
19:                  $\hat{\theta}_{LO} \leftarrow$ *local_optimization* $(\hat{\theta}')$
20:                  $\hat{\theta}_{LO} \leftarrow$ *recover* $(\hat{\theta}_{LO})$
21:                  **if** $\hat{\theta}_{LO} \neq$ NULL **then**
22:                      $\varepsilon_{LO} \leftarrow$ *model_quality* $(\hat{\theta}_{LO})$
23:                      **if** $\varepsilon' \prec \varepsilon_{LO}$ **then**
24:                          $\hat{\theta}', \varepsilon' \leftarrow \hat{\theta}_{LO}, \varepsilon_{LO}$
25:              $\hat{\theta}^*, \varepsilon^* \leftarrow \hat{\theta}', \varepsilon'$
26:              $\mathcal{T} \leftarrow$ *update* $(\hat{\theta}^*, \mathcal{I}_{\hat{\theta}^*})$
27: $\hat{\theta}^* \leftarrow$ *polish_final* $(\hat{\theta}^*)$

---

the chosen sampling method is Progressive NAP-SAC, alg. 1, line 3. Other samplers are described in section 2.2. The pre-emptive model verification is SPRT, alg. 1, line 10. Other options could be none verification or $T_{d,d}$ test, see section 2.4. The termination condition, alg. 1, line 2 is combination of SPRT and P-NAPSAC since P-NAPSAC and SPRT are used. The measured quality of model is MSAC (sum of truncated errors), alg. 1, line 12. The MSAC quality could be also replaced by MLESAC or MAGSAC quality, see section 2.3. The local optimization step is done in the line 19 by graph-cut-based local optimization. Other modifications of local optimization are in the section 2.1.

The degeneracy of model (e.g., validation of epipolar oriented constraint [9]) is done in the alg. 1, line 8 and after finding so-far-the-best model in the line 14 (e.g., planarity of fundamental matrix [10]). In the end the output model is polished by least squares fitting on all inliers, alg. 1, line 27.

2.1. Local optimization

The options for local optimization are listed below. The one chosen in USACv20 is written in bold.

LO-RANSAC [8]	Refine each so-far-the-best model by an inner RANSAC.
FLO-RANSAC [18]	Improvement of LO-RANSAC.
Graph-Cut RANSAC [3]	Spatial coherence is considered when doing the inner RANSAC.
σ-consensus [4]	A part of the MAGSAC algorithm marginalizing over the noise-scale.

We chose Graph-Cut RANSAC since it is more accurate than LO-RANSAC and FLO-RANSAC and significantly faster than the σ-consensus which requires a number of least-squares fittings.

2.2. Sampling

The possible options for sampling are listed below. The one chosen in USACv20 is written in bold.

Uniform [12]	The default option.
NAPSAC [24]	Selecting the first points and, then, local sampling from its neighborhood.
PROSAC [6]	Sampling from the most promising samples first and progressively blending to the uniform sampler of RANSAC.
P-NAPSAC [2]	Combination of PROSAC and NAPSAC sampling from gradually growing neighborhoods.

We chose P-NAPSAC since it leads to finding a good-enough sample earlier than PROSAC when the sought model is localized. In case of having a global model, e.g. the background motion in two images, it is found not noticeably later than by PROSAC due to progressively blending into global sampling.

2.3. Quality

The options for the model quality calculation are listed below. The one chosen is written in bold.

RANSAC [12]	The number of inliers.
MSAC [38]	The sum of truncated errors.
MLESAC [38]	Likelihood of the model.
LMedS [29]	The least median of errors.
MAGSAC [4]	Sum of errors marginalized over the noise-scale.

We chose MSAC quality calculation since it is always more accurate than that of RANSAC; it does not require expensive calculations like MLESAC or MAGSAC; and does not need to know the outlier ratio a priori as LMedS does.

2.4. Pre-emptive verification

The options for the pre-emptive verification are listed below. The one chosen is written in bold.

$T_{d,d}$ [7]	If d out d points are inliers then model is good.
SPRT [7]	Verify model by sequential decision making based on Wald's theory.

The $T_{d,d}$ test can make many false-negatives (rejecting good models) when the inlier ratio is low. Therefore we chose SPRT verification.

2.5. Termination criterion

The options for the termination criterion are listed below. The one chosen is written in bold.

Standard [12]	Terminates if the probability of finding a model with more inliers than the previous best falls below a threshold with some confidence.
PROSAC [6]	Terminates when the maximality and non-randomness criteria are satisfied.
SPRT [7]	Termination based on a sequence of subsequent model validations.
P-NAPSAC [2]	The standard RANSAC criterion relaxed by requiring the new model to select significantly more inliers than the previous best.
MAGSAC [4]	Marginalization of the standard RANSAC criterion over the noise-scale σ.

The termination of SPRT and P-NAPSAC depends on different properties of the robust procedure. P-NAPSAC stops when the relaxed RANSAC criterion is triggered, meaning that the probability of finding a significantly better model than the previous best falls below a threshold. The SPRT criterion is triggered by the number of subsequent model verification sequences made. These two techniques can straightforwardly be combined. Thus, we stop when at least one of them is triggered.

2.6. Degeneracy

USACv20 framework includes different tests on degeneracy. DEGENSAC [10] is about detecting when the majority of the drawn sample originates from the same 3D plane. For fundamental and essential matrix estimation oriented epipolar constraint [9] is evaluated as well. For homography estimation the verification of samples by its orientation is included.

2.7. Other features

For PROSAC or Progressive NAPSAC, exploiting an a priori known quality of the input data points makes the finding of a good-enough model significantly earlier than by other samplers. However, such prior information usually is unknown, degrading PROSAC to being the entirely uniform sampler of RANSAC. In the proposed USACv20 framework, when such quality function is not available, we use the density of the points as the quality function. This reflects the fact real-world data often forms spatially coherent structures and, thus, good correspondences tend to be close.

The spatial coherence of points plays important role in the estimation. For instance, it is exploited in the graph-cut-based local optimization or in P-NAPSAC sampler. Consequently, the neighborhood graph must be computed. The efficient way to do this is using a multi-layer grid described in [2]. In USACv20 such neighborhood estimation is implemented and used in the experiments.

3. Experimental results

We compared the proposed USACv20 to three robust estimators, i.e., USAC [28] [1], GC-RANSAC [3] and the RANSAC implementation of OpenCV. The applied USACv20 consists of SPRT verification, DEGENSAC [10], P-NAPSAC sampler and the local

---

[1] http://wwwx.cs.unc.edu/~rraguram/usac/USAC-1.0.zip

optimization of GC-RANSAC. USAC estimator [28] includes SPRT verification, DEGENSAC, PROSAC sampler and the local optimization of the original LO-RANSAC. All estimators were tested using the same number of maximum iterations (10,000 for **H** and 1,000 for **F**, **E** estimation) and confidence equals to 99%.

Fundamental matrix estimation was evaluated on the benchmark of [5]. The [5] benchmark includes: (1) the TUM dataset [35] consisting of videos of indoor scenes. Each video is of resolution 640×480. (2) The KITTI dataset [13] consists of consecutive frames of a camera mounted to a moving vehicle. The images are of resolution 1226×370. Both in KITTI and TUM, the image pairs are short-baseline. (3) The Tanks and Temples (T&T) dataset [17] provides images of real-world objects for image-based reconstruction and, thus, contains mostly wide-baseline pairs. The images are of size from 1080×1920 up to 1080×2048. (4) The Community Photo Collection (CPC) dataset [40] contains images of various sizes of landmarks collected from Flickr. In the benchmark, $1\,000$ image pairs are selected randomly from each dataset. SIFT [19] correspondences are detected, filtered by the standard SNN ratio test [19] and, finally, used for estimating the epipolar geometry.

The compared methods are USAC [28], GC-RANSAC [3], the RANSAC [12] implementation in OpenCV and the proposed USACv20. For all methods, the confidence was set to 0.99. For each method and problem, we chose the threshold maximizing the accuracy. The used error metric is Sampson distance. All methods were in C++.

The first four blocks of Table 1 report the median errors (ϵ_{med}, in pixels), the failure rates (f; in percentage) and processing times (t; in milliseconds) on the datasets used for fundamental matrix estimation. We report the median values to avoid being affected by the failures – which are also shown. A test is considered failure if the error of the estimated model is bigger than the 1% of the image diagonal. The best values are shown in red, the second best ones are in blue. It can be seen that *USACv20 leads to the lowest errors* on all datasets. Its failure ratio and processing time are always the lowest or the second lowest.

In Figures 4,5,7,6, the cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; in milliseconds) of the estimated fundamental matrices are

shown. Being accurate or fast is interpreted by a curve close to the top. It can be seen that USACv20 is always amongst the top performing methods in terms of geometric accuracy. The only methods which are faster than USACv20 on any dataset, are significantly less accurate on that particular dataset. For instance, on Tanks and Temples (Fig. 7), USACv20 is the second fastest method (right plot) right after USAC which is the least accurate one (left).

For homography estimation, we downloaded homogr (12 pairs) and EVD (15 pairs) datasets [18]. They consist of image pairs of different sizes from 329×278 up to 1712×1712 with point correspondences and inliers selected manually. The homogr dataset contains mostly short baseline stereo images, whilst the pairs of EVD undergo an extreme view change, i.e., wide baseline or extreme zoom. In both datasets, the correspondences are assigned manually to one of the two classes, i.e., outlier or inlier of the most dominant homography present in the scene. All algorithms applied the normalized four-point algorithm [15] for homography estimation and were repeated 100 times on each image pair. To measure the quality of the estimated homographies, we used the RMSE re-projection error calculated from the provided ground truth inliers.

The fifth and sixth blocks of Table 1 report the median errors (ϵ_{med}, in pixels), the failure rates (f; in percentage) and processing times (t; in milliseconds) on the datasets used for homography estimation. We report the median values to avoid being affected by the failures – which are also shown. A test is considered failure if the error of the estimated model is bigger than the 1% of the image diagonal. The best values are shown in red, the second best ones are in blue. It can be seen that USACv20 is the most accurate method on the Homogr dataset and the second most accurate one on ExtremeView. Its failure ratio and processing time are always the lowest or the second lowest.

In Figures 2,3, the cumulative distribution functions (CDF) of the re-projection errors (left plot; horizontal axis) and processing times (right; in milliseconds) of the estimated homographies are shown. Being accurate or fast is interpreted by a curve close to the top. It can be seen that USACv20 is always amongst the most accurate methods. Its processing time is the second best on Homogr dataset by a margin of 2-3 ms. On ExtremeView, USACv20 is significantly faster than all the competitor robust esti-

mators.

For essential matrix estimation, we downloaded the `Strecha` (1359 pairs) dataset and the `Piccadilly` scene from the `1DSfM` dataset[2] [40]. For the images of `Strecha`, both the intrinsic camera parameters and the ground truth poses are provided. First, we detected SIFT correspondences [19], filtered them by the standard SNN ratio test [19] The intrinsic parameters were used for normalizing the point coordinates. The ground truth pose was used for validation purposes selecting the ground truth inlier correspondences from the detected ones. These selected inliers were then used for measuring the error of the estimated essential matrices. The `1DSfM` dataset consists of 13 scenes of landmarks with photos of varying sizes collected from the internet. It provides 2-view matches with epipolar geometries and a reference reconstruction from incremental SfM (computed with Bundler [32, 33]) for measuring error. We iterated through the provided 2-view matches, detected SIFT correspondences [19], filtered them by the standard SNN ratio test [19], and calculated the ground truth relative pose from the reference reconstruction made by the Bundler algorithm. Note that all image pairs were excluded from the evaluation where fewer than 20 correspondences were found. For the evaluation, we chose the largest scene, i.e. Piccadilly, consisting of $7,351$ images.

The last two blocks of Table 1 report the median errors (ϵ_{med}, in pixels), the failure rates (f; in percentage) and processing times (t; in milliseconds) on the datasets used for essential matrix estimation. The best values are shown in red, the second best ones are in blue. It can be seen that USACv20 is the most accurate method on both datasets while being the second fastest one.

In Figures 8,9, the cumulative distribution functions (CDF) of the SGD errors (left plot; horizontal axis) and processing times (right; in milliseconds) of the estimated homographies are shown. Being accurate or fast is interpreted by a curve close to the top. It can be seen that USACv20 is always amongst the most accurate methods while being marginally slower than USAC. However, since USAC does not have essential matrix solver so only fundamental matrices were estimated and then converted to essential matrix using ground truth intrinsic matrices. In general, 5-points algorithm [25] is much slower than
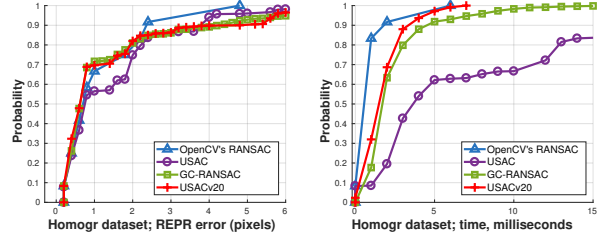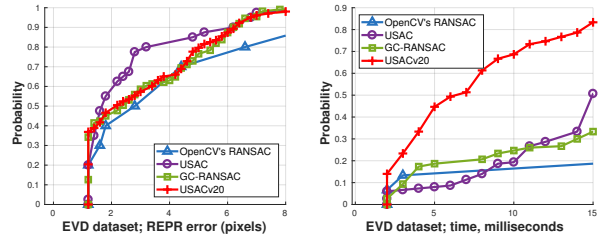
Figure 2. The cumulative distribution functions (CDF) of the Re-projection errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated homographies on the `Homogr` dataset.

Figure 3. The cumulative distribution functions (CDF) of the Re-projection errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated homographies on the `ExtremeView` dataset.

7-points algorithm which was used for F-estimation and number of output models for E ranges from 0 to 10 while number of estimated F matrices is at most 3; consequently all of these makes USAC framework faster.

In summary, the proposed USACv20 is, on all but one dataset (i.e., `ExtremeView`), more accurate than the original USAC algorithm while, usually, being faster. Even though USAC is more accurate on `ExtremeView`, it fails twice as often as USACv20.

The values reported in Table 1 are summarized in Table 2. It can be seen that the proposed algorithm is, on average, more accurate and faster than the compared state-of-the-art robust estimators. Its failure rate is the second best right behind GC-RANSAC.

4. Conclusion

In this paper, we reviewed some of the most recent RANSAC variants, combined them together and proposed a state-of-the-art variant, i.e. USACv20, of the Universal Sample Consensus [28] (USAC) algorithm. USACv20 is tested on 8 datasets, estimating homographies, fundamental and essential matrices. On average, it leads to the most geometrically accurate models and it is fastest compared to USAC, OpenCV's RANSAC and Graph-

	Fundamental matrix												Homography						Essential matrix					
	KITTI [13]			TUM [35]			T&T [17]			CPC [40]			Homogr [18]			EVD [18]			Strecha [34]			Piccadily [40]		
	ϵ_{med}	t	$f(\%)$	ϵ_{med}	t	f	ϵ_{med}	t	f	ϵ_{med}	t	f	ϵ_{med}	t	f	ϵ_{med}	t	f	ϵ_{med}	t	f	ϵ_{med}	t	f
USACv20	**0.2**	**1.9**	**0.2**	**0.3**	**2.1**	**8.4**	**0.6**	**5.6**	**12.9**	**0.5**	**5.3**	**43.0**	**0.7**	**2.2**	**0.0**	**2.3**	**8.5**	**31.3**	**0.4**	**8.1**	4.6	**0.9**	**7.3**	**2.2**
GC-RANSAC	**0.3**	2.3	**0.1**	**0.4**	3.1	8.6	**0.6**	8.8	**13.0**	**0.5**	7.2	**42.8**	**0.8**	2.8	**0.0**	2.5	**24.5**	**26.0**	**0.4**	**7.4**	**3.8**	**0.9**	14.5	3.1
USAC	0.4	3.3	0.3	0.6	**2.2**	9.2	1.4	**4.4**	15.0	0.8	**3.1**	44.0	0.9	10.0	**0.0**	**1.8**	25.0	73.3	0.8	9.1	**3.8**	1.3	**2.6**	3.1
OpenCV	0.4	**1.6**	**0.2**	0.5	4.4	**8.3**	0.8	18.5	**13.0**	0.7	14.9	45.2	0.9	**1.3**	**0.0**	3.5	136.0	33.3	0.5	69.2	**3.0**	1.0	121.0	**0.8**

Table 1. Median errors (ϵ_{med}), failure rates (f; as percentages) and avg. run-times (t, in milliseconds) are reported for each method on all tested problems and datasets. The error of the fundamental matrices is the Sampson distance from the ground truth. For homographies, the RMSE re-projection error from ground truth inliers is used. For essential matrix, the error is symmetric geometric distance (SGD) of normalized points. A test is considered a failure if the error is bigger than 1% of the image diagonal. For each method, the inlier-outlier threshold was set to maximize the accuracy (for fundamental matrix is 1 pixel, for homographies 2 pixels and for essential matrix, 1 pixel normalized by the intrinsic matrices) and the confidence to 0.99. The best values in each column are shown by red and the second best ones by blue.

	USACv20	GC-RANSAC	USAC	OpenCV
ϵ	**0.7**	**0.8**	1.0	1.0
t	**5.1**	8.8	**7.5**	45.9
f	**12.8**	**11.9**	18.6	13.0

Table 2. The avg. of the errors (ϵ; in pixels), processing times (t; in milliseconds) and failure rates (f; in percentages) in Table 1 are reported. The best values in each column are shown by red and the second best ones by blue.
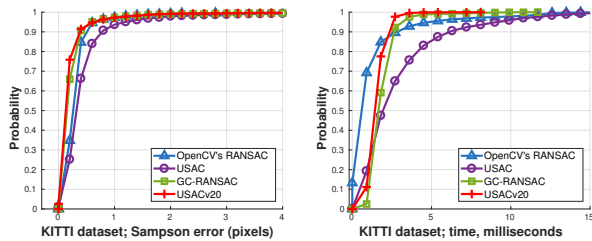

Figure 4. The cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated fundamental matrices on the KITTI dataset.
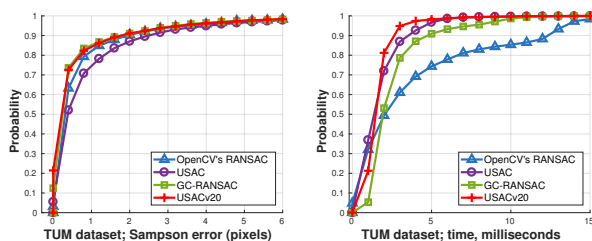

Figure 5. The cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated fundamental matrices on the TUM dataset.

Figure 6. The cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated fundamental matrices on the CPC dataset.

Figure 7. The cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated fundamental matrices on the Tanks and temples dataset.

Figure 8. The cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated essential matrices on the Strecha dataset.

Cut RANSAC. Compared to the original USAC, all reported properties improved significantly. Also, an important objective was to implement a modular and optimized framework in C++ to make future RANSAC modules easy to be combined with. The pipeline will be made available after publication.
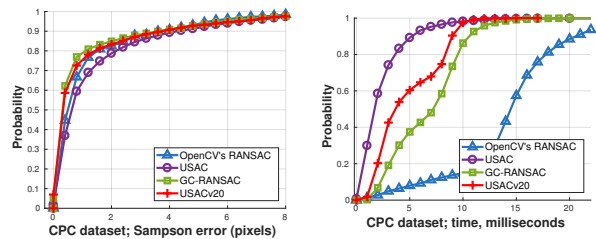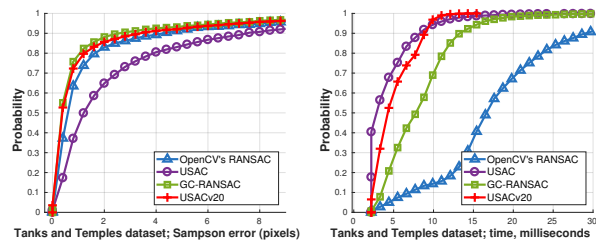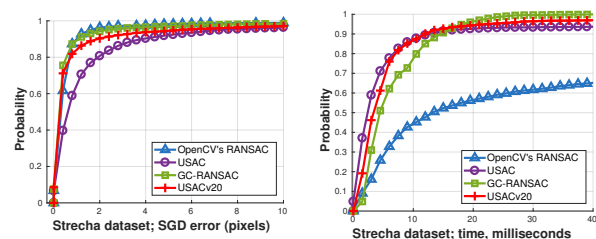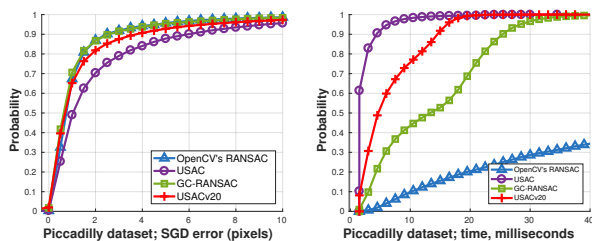
Figure 9. The cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated essential matrices on the Piccadilly scene of the 1DSfM dataset.

5. Acknowledgement

References

[1] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski. Bundle adjustment in the large. In *European conference on computer vision*, pages 29–42. Springer, 2010. 1

[2] D. Barath, M. Ivashechkin, and J. Matas. Progressive NAPSAC: sampling from gradually growing neighborhoods. *arXiv preprint arXiv:1906.02295*, 2019. 2, 3, 4

[3] D. Barath and J. Matas. Graph-Cut RANSAC. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6733–6741, 2018. https://github.com/danini/graph-cut-ransac. 2, 3, 4, 5

[4] D. Barath, J. Noskova, and J. Matas. MAGSAC: marginalizing sample consensus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. https://github.com/danini/magsac. 3, 4

[5] J.-W. Bian, Y.-H. Wu, J. Zhao, Y. Liu, L. Zhang, M.-M. Cheng, and I. Reid. An evaluation of feature matchers forfundamental matrix estimation. *arXiv preprint arXiv:1908.09474*, 2019. https://jwbian.net/fm-bench. 5

[6] O. Chum and J. Matas. Matching with PROSAC-progressive sample consensus. In *Computer Vision and Pattern Recognition*. IEEE, 2005. 2, 3, 4

[7] O. Chum and J. Matas. Optimal randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1472–1482, 2008. 2, 4

[8] O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In *Joint Pattern Recognition Symposium*. Springer, 2003. 2, 3

[9] O. Chum, T. Werner, and J. Matas. Epipolar geometry estimation via RANSAC benefits from the ori-

[10] O. Chum, T. Werner, and J. Matas. Two-view geometry estimation unaffected by a dominant plane. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 772–779. IEEE, 2005. 3, 4

[11] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 1

[12] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 1, 3, 4, 5

[13] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 5, 7

[14] D. Ghosh and N. Kaabouch. A survey on image mosaicking techniques. *Journal of Visual Communication and Image Representation*, 2016. 1

[15] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. 5

[16] H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *International Journal of Computer Vision*, 2012. 1

[17] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and Temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):78, 2017. 1, 5, 7

[18] K. Lebeda, J. Matas, and O. Chum. Fixing the locally optimized RANSAC. In *British Machine Vision Conference*. Citeseer, 2012. http://cmp.felk.cvut.cz/wbs/. 2, 3, 5, 7

[19] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer vision*. IEEE, 1999. 5, 6

[20] J. Matas and O. Chum. Randomized RANSAC with sequential probability ratio test. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1727–1732. IEEE, 2005. 2

[21] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 2004. 1

[22] D. Mishkin, J. Matas, and M. Perdoch. MODS: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 2015. 1

[23] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam

system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 1

[24] D. R. Myatt, P. H. S. Torr, S. J. Nasuto, J. M. Bishop, and R. Craddock. NAPSAC: high noise, high dimensional robust estimation. In *In BMVC02*, pages 458–467, 2002. 2, 3

[25] D. Nistér. An efficient solution to the five-point relative pose problem. *Transactions on Pattern Analysis and Machine Intelligence*, pages 756–770, 2004. 6

[26] T. T. Pham, T.-J. Chin, K. Schindler, and D. Suter. Interacting geometric priors for robust multimodel fitting. *Transactions on Image Processing*, 2014. 1

[27] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *International Conference on Computer Vision*. IEEE, 1998. 1

[28] 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. 1, 4, 5, 6

[29] P. J. Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984. 4

[30] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 1

[31] C. Sminchisescu, D. Metaxas, and S. Dickinson. Incremental model-based estimation using geometric constraints. *Pattern Analysis and Machine Intelligence*, 2005. 1

[32] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *ACM SIGGRAPH 2006 Papers*, page 835–846, New York, NY, USA, 2006. Association for Computing Machinery. 6

[33] S. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International journal of computer vision*, 80(2):189–210, 2008. 6

[34] 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. 7

[35] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012. 5, 7

[36] P. H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–61, 2002. 2

[37] P. H. S. Torr and D. W. Murray. Outlier detection and motion segmentation. In *Optical Tools for Manufacturing and Advanced Automation*. International Society for Optics and Photonics, 1993. 1

[38] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 2000. 2, 4

[39] P. H. S. Torr, A. Zisserman, and S. J. Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *Computer Vision and Image Understanding*, 1998. 1

[40] K. Wilson and N. Snavely. Robust Global Translations with 1DSfM. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 61–75. Springer, 2014. 1, 5, 6, 7

[41] M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multi-RANSAC algorithm and its application to detect planar homographies. In *International Conference on Image Processing*. IEEE, 2005. 1