SCRAMSAC: Improving RANSAC's Efficiency with a Spatial Consistency Filter

Torsten Sattler, Bastian Leibe, Leif Kobbelt
RWTH Aachen University
{tsattler@cs, leibe@umic, kobbelt@cs}.rwth-aachen.de

Abstract

Geometric verification with RANSAC has become a crucial step for many local feature based matching applications. Therefore, the details of its implementation are directly relevant for an application's run-time and the quality of the estimated results. In this paper, we propose a RANSAC extension that is several orders of magnitude faster than standard RANSAC and as fast as and more robust to degenerate configurations than PROSAC, the currently fastest RANSAC extension from the literature. In addition, our proposed method is simple to implement and does not require parameter tuning. Its main component is a spatial consistency check that results in a reduced correspondence set with a significantly increased inlier ratio, leading to faster convergence of the remaining estimation steps. In addition, we experimentally demonstrate that RANSAC can operate entirely on the reduced set not only for sampling, but also for its consensus step, leading to additional speed-ups. The resulting approach is widely applicable and can be readily combined with other extensions from the literature. We quantitatively evaluate our approach's robustness on a variety of challenging datasets and compare its performance to the state-of-the-art.

1. Introduction

Local feature based matching has become the dominant paradigm for Structure-from-Motion [20, 22], wide-baseline stereo [4], and large-scale image retrieval [21]. Those applications build upon a standard pipeline consisting of steps for extracting local features (*e.g.* SIFT [15]) from both images, matching them to find correspondences, and applying some form of geometric verification to find correspondence sets that are inliers to an affine, homography, or epipolar transformation. This geometric verification is critical for the pipeline's success, and RANSAC [10] has proven the method of choice for its implementation [1].

Consequently, numerous extensions have been proposed in the recent past in order to speed up the different RANSAC stages [2, 4, 5, 16], to deliver run-time guarantees for real-time performance [19, 23], and to improve the quality of the estimated solution [6, 9, 11, 24]. However, even with those extensions, the geometric verification is still a major bottleneck in many applications. In addition, some of the above extensions require considerable implementation effort and are hard to tune for optimal performance.

In this paper, we propose a fast and simple method for improving RANSAC's performance. Our main idea is to introduce a spatial consistency check that results in a reduced correspondence set with a higher inlier percentage, on which RANSAC converges faster to a correct solution. We experimentally show that the resulting estimation procedure, termed SCRAMSAC ("Spatially Consistent RAndoM SAmple Consensus"), is widely applicable to a large number of different datasets, while yielding results that are robust to degenerate configurations. In particular, we compare our approach to PROSAC and show that it achieves similar speedups while being considerably simpler to implement.

In detail, this paper makes the following contributions: (1) We propose a test for quality of correspondences in a RANSAC framework that does not only rely on appearance, but that takes into account the quality of neighboring matches in the image space. (2) We empirically show that this test leads to a reduced correspondence set with significantly increased inlier ratio that can be used to considerably speed up RANSAC, while still delivering similar-quality results. (3) We additionally show that this reduced set is not only suitable for sampling, but also sufficient for the verification step (in contrast to the reduced sets delivered by, *e.g.*, PROSAC), thus again improving run-time. (4) Our proposed algorithm leads to an estimation process that is more robust to degenerate configurations than PROSAC, so that it does not become as critical to add special degeneracy checks. (5) We quantitatively evaluate our approach on challenging test data and experimentally compare it to RANSAC and PROSAC with and without degeneracy tests.

The paper is structured as follows. The following section discusses related work. Sec. 3 then introduces the spatial consistency check that forms the basis of our algorithm and describes how it is integrated into a RANSAC framework. Sec. 4 discusses various extensions, and Sec. 5 presents experimental results. A final discussion concludes our work.

2. Related Work

RANSAC [10] has become the most popular tool to solve geometric estimation problems in datasets containing outliers. It operates in a hypothesize-and-verify framework. Given a set of tentative correspondences, RANSAC ran-

domly samples a minimal subset of size m from this set in order to hypothesize a geometric model. This model is then verified against the remaining correspondences, and the number of inliers, *i.e.* of correspondences consistent with the model, is determined as its score. This process is iterated until a certain termination criterion is met. An important property of RANSAC is that given an estimate for the true ratio ε of inliers to all correspondences, it is possible to determine the number of samples that must be drawn until an uncontaminated sample is found with failure probability η_0. This ensures a bounded runtime, as well as a guarantee on the quality of the estimated result. Since RANSAC's introduction, various improvements and extensions have been proposed [1]. In the following, we briefly discuss the ones that are most closely related to our approach.

When estimating epipolar geometries, problems arise with (quasi-) degenerate point configurations. Chum *et al.* show that if five or more points of the epipolar sample can be described by a single homography, the resulting epipolar geometry will not be correct, but the model will still have a similar number of inliers, causing RANSAC to terminate too early [9]. DEGENSAC [9] overcomes this problem by simultaneously keeping a homography and an epipolar hypothesis and applying model selection to choose the final result. QDEGSAC [11] iteratively applies standard RANSAC using fewer constraints to find the most appropriate model describing the data. Degenerate cases are thus detected, and the correct model can often be recovered by taking into account the outliers of the degenerate hypothesis.

While standard RANSAC guarantees with a certain probability that the estimated model is indeed the best, no a-priori guarantee is given on its runtime. Several extensions have therefore been proposed for real-time applications. Preemptive RANSAC [19] evaluates a fixed number of hypotheses in a breadth-first manner, reducing the number of hypotheses by half after each stage. ARRSAC [23] first evaluates a few hypotheses in depth in order to estimate the inlier ratio and from this derives an adaptive number of hypotheses to evaluate in a similar breadth-first search. Both approaches are guaranteed to finish in a certain time frame, but cannot give guarantees on the estimation quality.

A lot of work has been invested in speeding up RANSAC while preserving result quality. LO-RANSAC [6] improves hypothesis quality by performing a local optimization step on the current inlier set every time a new best hypothesis is found. Other approaches aim to reject bad hypotheses early on. [8] checks whether the estimated geometry fulfills the oriented epipolar constraint that only points in front of the camera are visible. The $T_{d,d}$ test [16] first evaluates the hypothesis on d randomly drawn points and only accepts it if all d are inliers to the model. The Bail-Out test [2] examines the inlier-ratio of a new model on all correspondences evaluated so far to estimate the probability that this model will improve on the current best model. The SPRT test [5] takes a similar approach to estimate the likelihood that the current model is incorrect. This approach is further extended in the same paper, leading to an optimal randomized RANSAC formulation. MLESAC [24] takes a different approach by improving the rating function for models. Instead of counting inliers to a model, it uses the maximum likelihood estimate as score to directly rate estimation quality.

Most directly related to our approach, several algorithms try to improve hypothesis quality (and thus convergence time) by non-uniform sampling. NAPSAC [18] assumes that features that are close together are more likely to lie on the same model. It therefore combines the image coordinates of the features in a correspondence pair into a single 4D vector in order to guide RANSAC sampling in a close spatial neighborhood. PROSAC [4] takes the feature matching score into account in order to preferably sample from high-ranking correspondences. It orders the correspondences by their similarity scores and operates on progressively increasing correspondence sets to generate hypotheses. In practice, this often results in significant computational savings, since good hypotheses are generated earlier. Both NAPSAC and PROSAC are however vulnerable to degenerate configurations. For NAPSAC, the reason is that adjacent feature points often lie in an almost-planar neighborhood. Similarly, correspondences with high similarity scores, as used in PROSAC, often lie on the same spatial structure, *e.g.* the same plane seen in an almost-frontal view. It has therefore been advised to combine them with checks against degeneracy [1, 9, 11, 23].

In this paper, we propose to create a reduced correspondence set by taking into account the matching quality of other features in a feature's scale-invariant neighborhood. Similar neighborhood definitions have been used for other purposes. [13] and [26] use a constant number of nearest neighbors in image space to establish correspondences by finding local affine transformations between the feature neighborhoods. [25] uses elliptical MSER regions to bundle SIFT features for an image retrieval task. The definition of neighborhood most similar to ours is used by [7] in the context of min-hashing for image retrieval. This paper uses the features in an affine invariant neighborhood of a hash key feature to generate further keys and show that this increases the recall rate of hashing. However, none of those approaches apply this idea for improving RANSAC.

3. Approach

Motivation. When verifying hypotheses, RANSAC categorizes the correspondences into inliers and outliers. Since the number of samples taken by RANSAC depends on the inlier ratio, it is desirable to reduce the fraction of outliers in the correspondence set. Our approach is motivated by the observation that there are generally two types of such outliers. Outliers of the first kind are caused by valid scene
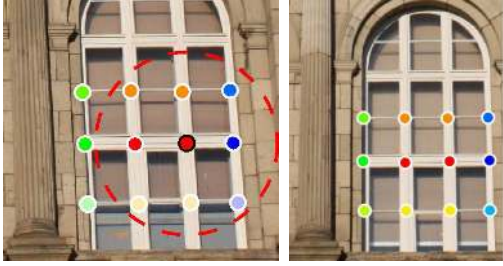
Figure 1: Local features often lead to incorrect correspondences due to similar structures (colored dots). By taking into account a larger spatial neighborhood, they can however be disambiguated (red circle). We use this idea to develop a faster and more robust RANSAC procedure.

matches that just fail the error bound because of limitations of the geometric model, the specific minimal subset it is estimated from, noise, or combinations of those factors. Outliers of the second kind are due to mismatching structures that are mistakingly put into correspondence because of limitations of the descriptor. In our approach, we focus on the latter and try to reduce their fraction by working on a reduced set of more confident correspondences.

Consider the example shown in Fig. 1. As depicted by the colored dots, similar image structures often cause ambiguous feature descriptors, which may lead to incorrect correspondences. The basic idea proposed in this paper is to augment the discriminative power of the raw feature descriptor by a *spatial consistency check* (SCC), which takes into account the matching quality in a larger spatial neighborhood. Specifically, we measure the fraction of neighboring features in a circular region around a feature in image 1 whose correspondences fall into a similar region in image 2. We then restrict all further processing to features where this fraction surpasses a threshold θ, resulting in a reduced set of higher-quality correspondences. By defining the size of the neighborhood relative to the feature scale, this procedure is made scale-invariant.

While this may sound like a relatively simple idea, we show that it has important consequences for the runtime and the robustness of the estimated results. In particular, we show that our approach achieves speedups comparable to and often surpassing those of PROSAC [4], while yielding results that are more robust to degenerate configurations. We begin by giving a formal definition of our spatial consistency check. We then discuss the geometric implications of our definition and deliver an intuitive interpretation for its effects. In addition, we discuss several potential limitations. Sec. 5 will then experimentally confirm that those do not have an adverse effect on our method's applicability.

Formal Definition. Given an image $\mathcal{I}_i$, we extract a set of scale-invariant features $\mathcal{F}(\mathcal{I}_i) = \{(x_j, y_j, \sigma_j, d_j)\}$ with center coordinates (x_j, y_j), scale σ_j, and descriptor d_j. For each feature f_j from this set, we define its *neighborhood set*

$N_{\mathcal{I}_i}(f_j)$ to comprise all features that fall inside a circular region around (x_j, y_j) with radius $r\sigma_j$ and whose scale lies within a similar range $(s_{min}\sigma_j, s_{max}\sigma_j)$:

$$N_{\mathcal{I}_i}(f_j) = \{f_k \in \mathcal{F}(\mathcal{I}_i) \setminus \{f_j\} \mid \|(x_k - x_j, y_k - y_j)\|_2 \leq r\sigma_j$$
$$\wedge s_{min} < \frac{\sigma_k}{\sigma_j} < s_{max}\}. \quad (1)$$

Given an image pair $(\mathcal{I}_1, \mathcal{I}_2)$, *point correspondences* are established by matching feature descriptors. We assume one-sided nearest-neighbor matching with a cut-off threshold on the similarity score, such that each feature in image 1 has at most one correspondence in image 2. Thus, we obtain a correspondence set C

$$C = \{(f^1, f^2) \mid f^1 \in \mathcal{F}(\mathcal{I}_1) \wedge f^2 \in \mathcal{F}(\mathcal{I}_2)\}. \quad (2)$$

Spatial Consistency Check. Given $(\mathcal{I}_1, \mathcal{I}_2)$ and C, we define the neighborhood set $N(c)$ of a correspondence $c = (f_j^1, f_k^2) \in C$ as

$$N(c) = \{(f^1, f^2) \in C \mid f^1 \in N_{\mathcal{I}_1}(f_j^1) \wedge f^2 \in N_{\mathcal{I}_2}(f_k^2)\} \quad (3)$$

and we accept a correspondence as *spatially consistent* iff

$$|N(f_j^1)| = |\{(f^1, f^2) \in C \mid f^1 \in N_{\mathcal{I}_1}(f_j^1)\}| > 0 \wedge \frac{|N(c)|}{|N(f_j^1)|} \geq \theta$$

with a threshold $\theta \in [0, 1]$. This results in a reduced set $C_{red} \subseteq C$ of spatially consistent correspondences.

Algorithm 1 RANSAC with SCC (SCRAMSAC)

---

1. Computation of the reduced set
$C_{\text{red}} \leftarrow SCC(C)$, $N = |C_{\text{red}}|$
2. RANSAC application
$k = 0$, $\varepsilon_{\text{red}} = m/N$, $I^* = 0$
while $\eta = (1 - \varepsilon_{\text{red}}^m)^k \geq \eta_0$ **do**
    Sample m random correspondences from C_{red}.
    Compute model Φ from samples.
    Compute number I of inliers for Φ on C_{red}.
    if $I > I^*$ **then**
        $I^* = I$, $\varepsilon_{\text{red}} = I^*/N$, store Φ.

RANSAC with SCC (SCRAMSAC). The spatial consistency check can be easily integrated into the RANSAC procedure (c.f. Alg. 1). Let ε_{red} denote the ratio of inliers to all correspondences in C_{red}. Then the probability that in k steps RANSAC has only chosen samples with at least one outlier follows as $\eta = (1 - \varepsilon_{\text{red}}^m)^k$. Note that both model generation and verification are carried out on the *reduced set* C_{red}, leading to considerable speedups in both stages. Once a result is obtained on C_{red}, we can additionally compute the hypothesis's support on the full set C. In our experiments, we however only perform this last step to report the corresponding inlier numbers.

Geometric Interpretation. The above definition lends itself to an intuitive geometric interpretation. The SCC enforces a minimum fraction of consistent matches in a circular neighborhood around both features in a correspondence
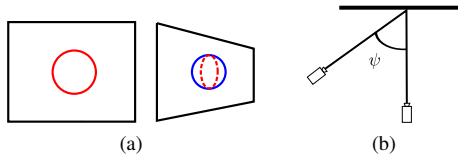
Figure 2: The SCC criterion penalizes scene structures that are viewed at vastly different angles, since it is based on circular neighborhoods whose area of overlap (a) diminishes with increasing difference in viewing angle ψ (b).

pair. This fraction is naturally bounded by the degree to which those neighborhoods depict the same region in the scene. Let us assume for simplicity that both image neighborhoods are locally planar and that $N_{\mathcal{I}_1}$ shows the relevant scene patch in a fronto-parallel view, as visualized in Fig. 2. If $\mathcal{I}_2$ views the same scene patch at an angle ψ, then the region defined by $N_{\mathcal{I}_1}$ corresponds to an ellipse in $\mathcal{I}_2$. The chance of finding a neighborhood correspondence thus depends on the overlap between the two regions in $\mathcal{I}_2$, which scales with $\cos\psi$.

This means that our SCC definition automatically penalizes scene structures that are viewed at vastly different angles in both images. This is a desired effect, since both the repeatability and the descriptive power of SIFT features decrease with the change in viewing angle [15, 17]. For large viewpoint changes, the risk of finding an incorrect match is therefore increased. The SCC still accepts matches under such viewpoint changes, but demands stronger support from neighboring features with increasing viewing angle. Concretely, if we enforce a threshold θ on the neighborhood correspondence fraction, then this limits the accepted feature matches to scene patches fulfilling $\psi \leq \arccos\theta$ (where equality only holds if *all features* in $N_{\mathcal{I}_1}$ have their matching features in $N_{\mathcal{I}_2}$). A threshold of $\theta = 0.8$ thus places a hard limit at $\psi \leq 36.9°$, whereas a threshold of $\theta = 0.55$ still accepts correspondences up to $\psi \leq 56.6°$, which roughly corresponds to the reliability range of SIFT [17].

Effect of SCC on RANSAC. The net effect is a reduction of the correspondence set to a reduced set of more confident matches. This leads to a significant speedup of the RANSAC procedure for two reasons. First, RANSAC only needs to operate on a substantially smaller set of correspondences C_{red} for verifying model hypotheses. This is in contrast to PROSAC, which verifies model hypotheses against all correspondences. In Sec. 5, we experimentally verify that C_{red} is still sufficiently representative, such that this step produces the desired results. Second, the additional constraints enforced by our SCC lead to an increased inlier ratio $\varepsilon_{\mathrm{red}}$ in C_{red}. This directly affects the number k of iterations RANSAC has to take until the failure probability η falls below some constant η_0:

$$k = \log(\eta_0)/\log(1 - \varepsilon_{\mathrm{red}}^m) . \qquad (4)$$

Thus, SCRAMSAC converges faster to a correct solution,

as will also be verified in Sec. 5.

Limitations. A potential limitation of our approach is that the SCC may reduce the absolute number of correspondences in C_{red} to a level that is insufficient to compute the desired geometric model. As discussed above, such a case is most likely to occur with large viewpoint changes. In Sec. 5, we therefore experimentally evaluate SCRAMSAC's robustness to viewpoint changes on several test sequences and show that the estimation only breaks down for very large viewpoint changes of more than $50°$.

In addition, the SCC does not guarantee that only false correspondences are rejected but might also reject inliers. Moreover, the ratio of rejected inliers to outliers cannot be bounded theoretically. So there is no formal guarantee that SCRAMSAC will indeed converge to the correct model. However, also for this case, Sec. 5 will empirically show that the ratio of inliers in C_{red} is (in most cases substantially) higher than in C, reducing the number of SCRAMSAC iterations while still leading to the correct solution.

Comparison with PROSAC. An important difference of our approach to PROSAC is that SCRAMSAC performs a single reduction step as a pre-filter to a regular RANSAC back-end. In contrast, PROSAC draws its speedup from operating on a progressively increasing set of correspondences. Such a progressive search is potentially dangerous, since the highest-ranking correspondences often lie on a degenerate configuration, *e.g.* the same 3D plane viewed from favorable angles. This is often the case when taking the feature similarity as ordering criterion, as proposed in the original PROSAC paper [4]. As a result, PROSAC may terminate too early and return an incorrect solution. Fig. 3 shows an example where this happens on an image pair from the Leuven Castle sequence [12]. As is also visible in the same figure, the points chosen by SCRAMSAC are distributed over a far larger part of the scene, leading to robust estimation results. In addition to being far simpler to implement, this is a major advantage of SCRAMSAC over PROSAC.

4. Extensions

In order to evaluate our approach, we will compare it to PROSAC [4]. As explained above, PROSAC is however susceptible to degenerate point configurations. To ensure a fair comparison, we therefore combine PROSAC with two extensions that have been designed to make the estimation more robust: local optimization (LO-RANSAC) [6] and a degeneracy test [9, 11]. For our own algorithm, we argue that local optimization is not necessary, since SCRAMSAC already achieves comparable results without it. In addition, we empirically show that the degeneracy test is less often needed in SCRAMSAC than in PROSAC because of its more robust feature selection.

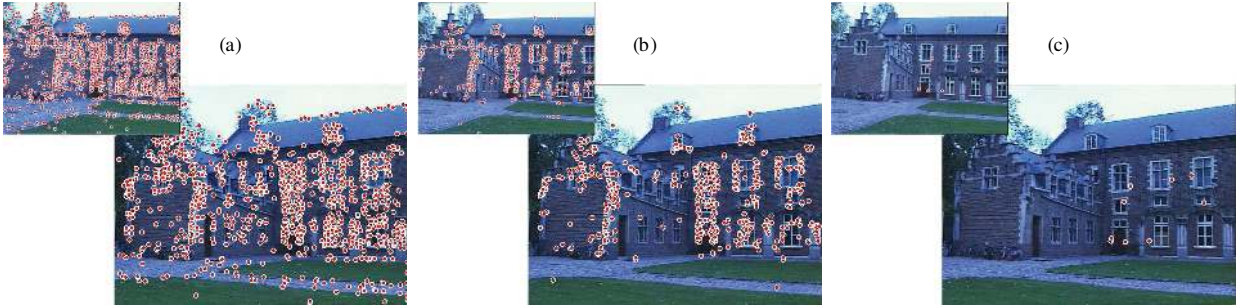**Early Model Rejection.** As SCRAMSAC only modifies

Figure 3: (a) Initial correspondences computed by SIFT matching (50% inliers). (b) Remaining correspondences after one application of the SCC (80% inliers). (c) Correspondences used by PROSAC for hypothesis generation. The SCC results in a reduced set of higher-quality correspondences, while still providing a sufficient coverage to ensure robust model estimation.

the initial correspondence set, it can be readily combined with early model rejection techniques such as the $T_{d,d}$ [16] or SPRT tests [5]. In our experiments in Sec. 5, we verify this claim by combining SCRAMSAC with $T_{d,d}$. Our results show that this combination indeed improves runtime. Further improvements are possible by plugging in additional RANSAC extensions.

Iterative Application. One may also extend the SCRAMSAC idea and perform several successive iterations of the SCC in order to further reduce the correspondence set. Fig. 4(e) shows the resulting correspondence set after six such iterations. As can be seen, the reduced set is still nicely distributed over the entire scene structure and leads to a correct estimation result. In general, we found that 2-3 iterations were typically safe to execute, while more iterations sometimes reduced the correspondence set too much. In practice, the benefits of performing this iteration will however be application dependent and will also depend on the efficiency of the SCC implementation compared to the remaining RANSAC steps. In all subsequent experiments, we only report results using one single SCC iteration.

5. Experimental Results

In the following, we experimentally evaluate our approach and compare it to both RANSAC and PROSAC.

Experimental Setup. For each image pair, initial correspondences are obtained by nearest-neighbor matching on SIFT features using Lowe's original implementation [15]. These correspondences are filtered using the second-NN criterion from [15] with a threshold of 0.8. The normalized 8-point algorithm is used for estimating fundamental matrices, while the normalized DLT algorithm is applied for computing homographies [12]. Inliers are determined as those points with a Sampson error of at most 1.0, while the termination probability of all RANSAC variants is set to 5%. When we refer to LO-PROSAC, we imply the usage of the LO-step as described in [6]. The parameter experiments were performed on a 2.4GHz Intel QuadCore with 4 GB of RAM, while all other tests were performed on a 3GHz Intel

CoreDuo with 4 GB of RAM. Unless specified otherwise, all experiments were repeated 1000 times to obtain statistically meaningful results. Plots show the resulting average values with the standard deviation denoted by error bars.

Setting the Parameters. We set the values of s_{min} and s_{max} to 0.5 and 2, respectively, thus defining feature neighborhoods to span at most 2 octaves in scale-space. Tests with different values for s_{min} and s_{max} led to no improvements. According to the argumentation in Sec. 3, we chose $\theta = 0.55$ as a starting value. Using this value, we performed tests for various values of r on a set of image pairs and measured the inlier ratio of both the full set and the reduced set (Fig. 4(a)), the number of iterations needed by SCRAMSAC (Fig. 4(b)), and the overall running time (Fig. 4(c)). As can be seen, performance is stable in the range $4.5 \leq r \leq 7.5$. We chose $r = 7$ as a compromise between robustness and efficiency. Finally, we validated our choice of θ by fixing the value of r while iterating over different values of θ. The results can be found in Fig. 4(d) and show that values of $\theta \geq 0.5$ are indeed sensible, with $\theta = 0.55$ being the most general choice. We therefore fixed $r = 7$ and $\theta = 0.55$ for all subsequent experiments.

Robustness to Viewpoint Changes. The number and reliability of feature matches and thus of neighboring correspondences strongly depend on the lighting and viewing conditions. To evaluate SCRAMSAC's robustness over a large range of viewpoint changes and varying lighting conditions, we apply it to several standard test sequences from the literature. For robustness under epipolar viewpoint changes, we use the Leuven Castle and Oxford Corridor sequences [12]. For homography estimation and robustness to lighting changes, we test on the Wall and Leuven Illumination sequences, respectively [17]. In all cases, we match the first image of each sequence against all other images and compare the results to those of RANSAC and LO-PROSAC.

The results of this experiment are shown in Fig. 5. It can be seen that SCRAMSAC remains robust up to large viewpoint changes and reaches a comparable result quality as standard RANSAC, while consuming only a fraction of the
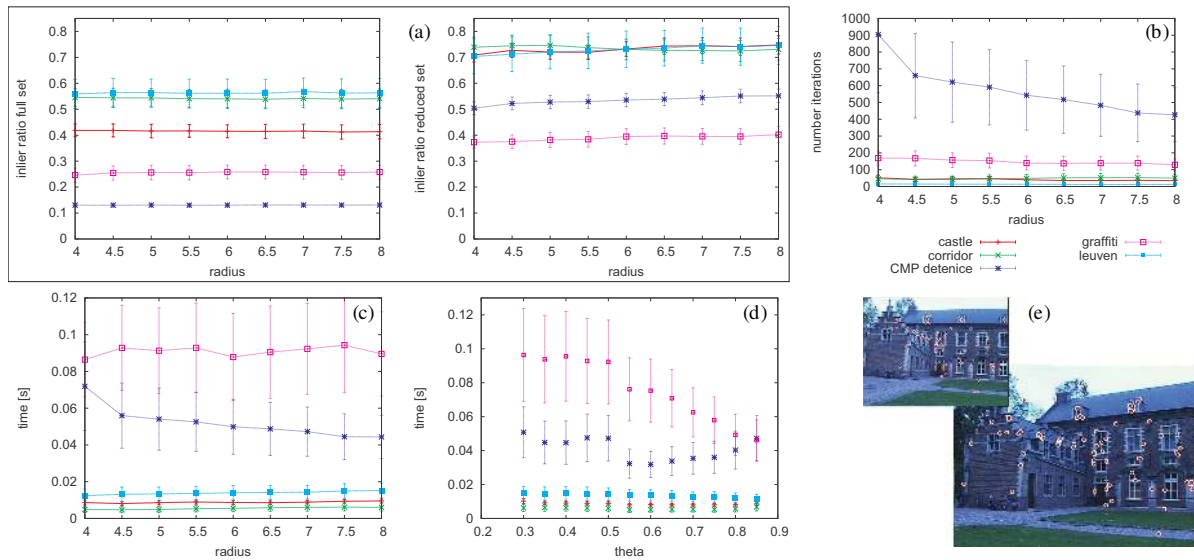
Figure 4: Experiments with different parameter choices: (a) inlier ratios, (b) number of iterations, and (c) runtimes for different radius settings; (d) runtimes for different values of θ; (e) remaining correspondences after 6 iterations of the SCC.

latter's runtime. Compared to LO-PROSAC, it can be seen that SCRAMSAC yields more stable results (as visible from the smaller standard deviation of its inlier ratios) at a comparable runtime. Only for very large viewpoint changes (the last images of each sequence) small differences become apparent due to the low absolute number of correspondences available here. As all four experiments show, the inlier ratio of the reduced set is consistently and significantly increased, verifying the general applicability of our idea.

Application to Challenging Image Pairs. To further demonstrate our approach's robustness, we compare it with RANSAC and PROSAC on challenging image pairs from [3], [4], and [14]. Besides large numbers of candidate correspondences, these test scenes also contain other challenges such as repetitive and small-scale structures (**A**, **H**, **J**), small overlap (**B**, **E**, **G**), and extreme scale (**F**) and lighting changes (**C**, **D**). As PROSAC repeatedly yielded degenerate results for some test cases, we combine it with both the LO and the QDEGSAC extension (denoted Q-LO-PROSAC in Tab. 1), as recommended in [4, 23]. For each algorithm, we report the inlier ratio ε, the number of iterations k, and the running time, averaged over 1000 runs. As Tab. 1 shows, SCRAMSAC improves over RANSAC's runtime by up to 2 orders of magnitude, while yielding similar-quality results. Also compared to PROSAC, it often achieves a considerable speedup. On some images PROSAC is still faster, but in those cases the inlier ratios reveal that PROSAC's solution is often suboptimal. Note again the large standard deviation of PROSAC's results, indicating that its runtime and estimation quality vary considerably between runs. In contrast, SCRAMSAC performs stably at runtimes that are in many cases still suitable for real-time applications. A

detailed comparison highlights the conceptual differences between both approaches: SCRAMSAC typically requires a larger number of iterations than PROSAC, but takes less time in each iteration. This is a direct consequence of its operation on a reduced correspondence set.

In general, we noticed that PROSAC's performance strongly depends on its parameter settings, in particular on the allowable Sampson error and the β parameter from [4]. We tried to ensure a fair comparison and report the best results we obtained for PROSAC over a variety of choices for β. Still, we cannot rule out that other settings might improve the above results. However, the necessity to perform this adaptation is an argument in favor of SCRAMSAC, which does not require parameter tuning.

Combination with Early Model Rejection. Finally, we demonstrate that the performance of SCRAMSAC can be improved through other RANSAC extensions. For this, we combine it with a $T_{d,d}$ test, setting $d = 1$ as suggested by [16]. The results of this experiment are shown in Tab. 1. As can be seen from the table, this combination results in noticeable performance improvements for cases with low inlier ratios (*e.g.* almost a factor of 6 speedup for **D**). In other cases, the performance remains similar due to SCRAMSAC's already high inlier ratio on the reduced set. We expect that more sophisticated methods such as SPRT [5] will further improve the results.

6. Conclusion

In this paper, we have presented an approach for improving RANSAC's efficiency in geometric matching applications. Our approach, termed SCRAMSAC, is based on a spatial consistency check that lets RANSAC operate
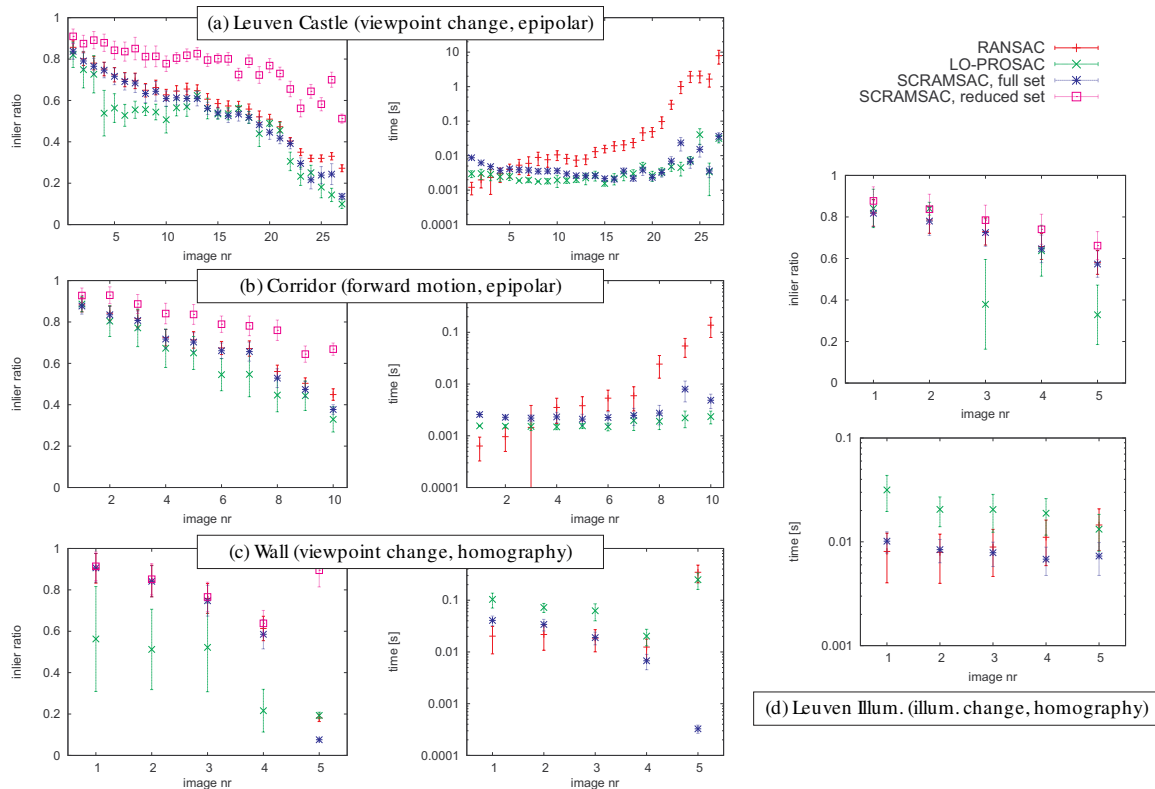
Figure 5: Results evaluating SCRAMSAC's robustness to (a)-(c) viewpoint and (d) lighting changes. The plots show inlier ratios of the original and reduced sets and runtimes (in log-scale), compared to RANSAC and LO-PROSAC. SCRAMSAC reaches the same result quality as RANSAC (1st column), while performing at the same speed as LO-PROSAC (2nd column).

on a reduced set of more confident correspondences with a higher inlier ratio. As shown in our experiments, the resulting approach improves RANSAC's runtime by a large factor, while yielding similar-quality results. Also in comparison to PROSAC, it achieves similar speed-ups but its results are more stable. Its wide applicability and simplicity of implementation make SCRAMSAC well-suited for combination with other RANSAC extensions. For example, it could be used for model generation in real-time variants such as Preemptive RANSAC [19] or ARRSAC [23]. In addition, we are planning to investigate extensions of the SCC to affine invariant features.

References

[1] *Proc. IEEE Int'l Workshop "25 Years of RANSAC" in conjunction with CVPR'06 (RANSAC25 06)*, 2006.

[2] D. Capel. An Effective Bail-Out Test for RANSAC Consensus Scoring. In *BMVC'05*, 2005.

[3] J. Cech, J. Matas, and M. Perdoch. Efficient sequential correspondence selection by cosegmentation. In *CVPR'08*, 2008.

[4] O. Chum and J. Matas. Matching with PROSAC - Progressive Sample Consensus. In *CVPR'05*, 2005.

[5] O. Chum and J. Matas. Optimal Randomized RANSAC. *PAMI*, 30(8):1472–1482, 2008.

[6] O. Chum, J. Matas, and S. Obdržálek. Enhancing RANSAC by Generalized Model Optimization. In *ACCV'04*, 2004.

[7] O. Chum, M. Perdoch, and J. Matas. Geometric min-Hashing: Finding a (Thick) Needle in a Haystack. In *CVPR'09*, 2009.

[8] O. Chum, T. Werner, and J. Matas. Epipolar Geometry Estimation via RANSAC Benefits from the Oriented Epipolar Constraint. In *ICPR'04*, 2004.

[9] O. Chum, T. Werner, and J. Matas. Two-View Geometry Estimation Unaffected by a Dominant Plane. In *CVPR*, 2005.

[10] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6), 1981.

[11] J.-M. Frahm and M. Pollefeys. RANSAC for (Quasi-) Degenerate data (QDEGSAC). In *CVPR'06*, 2006.

[12] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, 2nd edition, 2004.

[13] I. Jung and S. Lacroix. A Robust Interest Points Matching Algorithm. In *ICCV'01*, 2001.

[14] A. Kelman, M. Sofka, and C. V. Stewart. Keypoint Descriptors for Matching Across Multiple Image Modalities and Non-linear Intensity Variations. In *CVPR'07*, 2007.

homography		$\varepsilon\,/\,\varepsilon_{\text{red}}$	RANSAC	RANSAC + $T_{d,d}$	Q-LO-PROSAC	SCRAMSAC	SCRAMSAC + $T_{d,d}$
A	$|C| = 1712$, $|C_{\text{red}}| = 917$	$\varepsilon\,/\,\varepsilon_{\text{red}}$ k t [ms]	0.53 ± 0.05 47 ± 19 67.3 ± 27.7	0.51 ± 0.06 65 ± 34 8.4 ± 3.8	0.19 ± 0.13 2 ± 1 34.1 ± 11.0	$0.53\pm0.05\,/\,0.72\pm0.07$ 13 ± 5 19.4 ± 4.4 (7.7)	$0.52\pm0.06\,/\,0.71\pm0.08$ 15 ± 8 12.6 ± 1.7 (7.6)
B	$|C| = 879$, $|C_{\text{red}}| = 451$	$\varepsilon\,/\,\varepsilon_{\text{red}}$ k t [ms]	0.43 ± 0.04 102 ± 38 73.8 ± 27.3	0.41 ± 0.05 153 ± 85 10.1 ± 5.1	0.35 ± 0.11 2 ± 0 16.0 ± 3.5	$0.43\pm0.04\,/\,0.71\pm0.06$ 14 ± 5 8.7 ± 2.1 (2.7)	$0.42\pm0.04\,/\,0.70\pm0.07$ 16 ± 8 5.5 ± 0.9 (2.7)
C	$|C| = 364$, $|C_{\text{red}}| = 102$	$\varepsilon\,/\,\varepsilon_{\text{red}}$ k t [ms]	0.19 ± 0.01 2792 ± 746 919.5 ± 245.7	0.17 ± 0.02 4688 ± 1865 256.8 ± 101.8	0.09 ± 0.04 4 ± 2 8.6 ± 2.6	$0.18\pm0.02\,/\,0.37\pm0.03$ 172 ± 48 22.5 ± 6.0 (0.7)	$0.17\pm0.02\,/\,0.35\pm0.03$ 237 ± 88 14.1 ± 4.9 (0.7)
D	$|C| = 294$, $|C_{\text{red}}| = 61$	$\varepsilon\,/\,\varepsilon_{\text{red}}$ k t [ms]	0.16 ± 0.2 5820 ± 2274 1591.4 ± 622.1	0.14 ± 0.02 15001 ± 9670 817.1 ± 526.4	0.03 ± 0.02 8 ± 2 10.0 ± 3.2	$0.15\pm0.02\,/\,0.46\pm0.04$ 81 ± 31 41.0 ± 8.3 (0.5)	$0.14\pm0.02\,/\,0.44\pm0.05$ 114 ± 57 6.7 ± 3.0 (0.5)
E	$|C| = 1019$, $|C_{\text{red}}| = 547$	$\varepsilon\,/\,\varepsilon_{\text{red}}$ k t [ms]	0.39 ± 0.03 153 ± 57 126.8 ± 47.6	0.37 ± 0.04 245 ± 139 15.7 ± 8.3	0.3 ± 0.10 4 ± 1 24.6 ± 8.0	$0.38\pm0.03\,/\,0.61\pm0.04$ 26 ± 10 16.7 ± 4.6 (3.7)	$0.38\pm0.04\,/\,0.59\pm0.05$ 31 ± 15 7.7 ± 1.3 (3.7)
F	$|C| = 308$, $|C_{\text{red}}| = 136$	$\varepsilon\,/\,\varepsilon_{\text{red}}$ k t [ms]	0.47 ± 0.04 78 ± 36 22.5 ± 10.4	0.46 ± 0.05 113 ± 67 6.5 ± 3.7	0.41 ± 0.08 3 ± 1 7.7 ± 2.6	$0.47\pm0.04\,/\,0.84\pm0.08$ 7 ± 4 2.0 ± 0.6 (0.6)	$0.47\pm0.04\,/\,0.84\pm0.08$ 8 ± 4 1.5 ± 0.4 (0.6)
epipolar geom.							
G	$|C| = 1348$, $|C_{\text{red}}| = 632$	$\varepsilon\,/\,\varepsilon_{\text{red}}$ k t [ms]	0.57 ± 0.2 299 ± 80 23.5 ± 6.3	0.57 ± 0.2 348 ± 111 20.1 ± 6.4	0.55 ± 0.07 2 ± 1 12.3 ± 5.9	$0.54\pm0.03\,/\,0.82\pm0.05$ 20 ± 9 5.5 ± 0.7 (4.0)	$0.53\pm0.03\,/\,0.81\pm0.05$ 21 ± 11 5.5 ± 0.7 (4.1)
H	$|C| = 835$, $|C_{\text{red}}| = 364$	$\varepsilon\,/\,\varepsilon_{\text{red}}$ k t [ms]	0.46 ± 0.02 1904 ± 640 132.5 ± 44.6	0.44 ± 0.02 2597 ± 1165 145.1 ± 65.2	0.36 ± 0.09 5 ± 3 4.6 ± 1.2	$0.46\pm0.02\,/\,0.70\pm0.04$ 67 ± 28 6.1 ± 1.8 (1.8)	$0.44\pm0.03\,/\,0.69\pm0.04$ 77 ± 35 6.1 ± 2.0 (1.8)
I	$|C| = 306$, $|C_{\text{red}}| = 106$	$\varepsilon\,/\,\varepsilon_{\text{red}}$ k t [ms]	0.54 ± 0.04 693 ± 371 41.5 ± 22.2	0.53 ± 0.5 1041 ± 667 57.3 ± 36.7	0.36 ± 0.11 4 ± 2 3.7 ± 0.8	$0.52\pm0.05\,/\,0.78\pm0.06$ 38 ± 24 2.8 ± 1.3 (0.6)	$0.51\pm0.05\,/\,0.78\pm0.07$ 45 ± 29 3.1 ± 1.6 (0.6)
J	$|C| = 1420$, $|C_{\text{red}}| = 507$	$\varepsilon\,/\,\varepsilon_{\text{red}}$ k t [ms]	0.47 ± 0.03 1783 ± 920 139.9 ± 72.2	0.46 ± 0.04 2820 ± 1718 157.9 ± 96.1	0.35 ± 0.11 5 ± 4 5.3 ± 1.3	$0.46\pm0.04\,/\,0.76\pm0.06$ 44 ± 24 6.3 ± 2.0 (3.2)	$0.46\pm0.04\,/\,0.75\pm0.06$ 51 ± 32 6.2 ± 1.8 (3.2)

Table 1: Results on challenging image pairs comparing the inlier ratios ε, the number of iterations k, and the average runtime for RANSAC, PROSAC, and SCRAMSAC. For SCRAMSAC, we additionally report the inlier ratio ε_{red} on the reduced set and the part of the total runtime taken for the SCC (in parantheses). As can be seen, SCRAMSAC yields low runtimes and achieves the same result quality as RANSAC, whereas PROSAC sometimes returns lower-quality results (shown in red).

[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[16] J. Matas and O. Chum. Randomized RANSAC with T(d,d) test. *Image and Vision Computing*, 22(10):837–842, 2004.

[17] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1/2), 2005.

[18] D. Myatt, P. Torr, S. Nasuto, J. Bishop, and R. Craddock. NAPSAC: High Noise, High Dimensional Robust Estimation - it's in the Bag. In *BMVC'02*, 2002.

[19] D. Nistér. Preemptive RANSAC for Live Structure and Motion Estimation. In *ICCV'03*, 2003.

[20] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *CVPR'04*, 2004.

[21] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR'07*, 2007.

[22] M. Pollefeys *et al.*(19 authors). Detailed real-time urban 3d reconstruction from video. *IJCV*, 78:143–167, 2008.

[23] R. Raguram, J.-M. Frahm, and M. Pollefeys. A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. In *ECCV'08*, 2008.

[24] P. Torr and A. Zisserman. MLESAC: a new robust estimator with application to estimating image geometry. *CVIU*, 78(1):138–156, 2000.

[25] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling Features for Large Scale Partial-Duplicate Web Image Search. In *CVPR'09*, 2009.

[26] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. *AI Journal*, 78(1-2), 1995.