CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY IN PRAGUE

MASTER'S THESIS

USAC 2021 with spatial coherence modelling robust to degenerate data

Maksym Ivashechkin

ivashmak@fel.cvut.cz

August 13, 2021

Thesis Advisor: prof. Ing. Jiří Matas Ph.D

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Ivashechkin Maksym**	Personal ID number:	**465693**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Open Informatics**		
Specialisation:	**Computer Vision and Image Processing**		

II. Master’s thesis details

Master’s thesis title in English:

USAC 2021 with Spatial Coherence Modelling Robust to Degenerate Data

Master’s thesis title in Czech:

USAC 2021 s modelem prostorové koherence robustní vůči degenerovaným datům

Guidelines:

RANSAC is a standard method for estimating geometric relations between images from noisy correspondences. In [3], a RANSAC method called USAC 2021, with a range of technical improvements of USAC, a version that implements local optimization, early termination and non-random sampling, has been proposed. Recently, other significant developments have been proposed - GC RANSAC [1] that considers the dependence of spatially adjacent correspondences and MAGSAC [2] that alleviates the problem of unknown inlier noise by marginalization. RANSAC-type algorithms have focused on returning the model with the highest support. The thesis will focus on the ability of the algorithm to guarantee non-randomness of the reported structure, or a failure to find it.

In the thesis, the student will:

1. Integrate [3], [2] and [1] in a single algorithm which should outperform the state-of-the-art. This task might also involve a modification of the termination criterion to reflect the loss functions of the GC and LO optimizers and dealing with the trade-off between speed (the number of GC runs) and accuracy.
2. Consider the problem of degenerate and close-to-degenerate configurations that is not fully covered in [1], [2] and [3]. The goal is to develop a method that either reports a failure to find a reliable model, or returns the correct solution, ie. solution that corresponds to a significant structure. Special care has to be taken in the LO and GC optimization steps to avoid local optima that correspond to degenerate configurations.

Evaluate the method on

a. standard datasets e.g. [4] [5],

b. a dataset focussing on difficult but realistic scenes, e.g. close to planar, highlighting the issues encountered in close-to-degenerate scenes.

Bibliography / sources:

[1] Daniel Barath and Jiri Matas. Graph-Cut RANSAC. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[2] Daniel Barath, Jana Nosková and Jiri Matas. MAGSAC: marginalizing sample consensus. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019.

[3] Maksym Ivashechkin, Daniel Barath and Jiri Matas: USAC21: Efficient and Accurate Estimator for H and F, 2020.

[4] Kusvod2 dataset. Karel Lebeda, Master’s Thesis: Robust Sampling Consensus, 2013.

[5] Image Matching Challenge 2020, PhotoTourism dataset.

Name and workplace of master’s thesis supervisor:

prof. Ing. Jiří Matas, Ph.D., Visual Recognition Group, FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **08.01.2021** Deadline for master's thesis submission: **13.08.2021**

Assignment valid until: **30.09.2022**

prof. Ing. Jiří Matas, Ph.D. Supervisor’s signature	prof. Ing. Tomáš Svoboda, Ph.D. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.
Date of assignment receipt

Student’s signature

Author statement for postgraduate thesis

I declare that the presented work developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, date 13.08.2021 ....

Maksym Ivashechkin

Acknowledgement

I would like to thank my supervisor Jiří Matas for his professional guidance and assistance towards robust estimation throughout the whole master's course. A substantial gratitude for support in a conference paper writing deserves my co-author Daniel Barath.

In additions, I am deeply grateful to my parents, relatives, and friends who supported me at every bit and without whom it was impossible to finish my project. Extra appreciation earns my cousin Denys Rozumnyi for his suggestions and help.

Abstract

Random Sample Consensus is one of the most popular and widely used method for robust estimation in computer vision. This work presents VSAC (USAC of 2021) that is a RANSAC-like robust estimator with a number of novelties and improvements. It introduces the concept of independent inliers to apply a statistical approach for distinguishing random models. This helps to avoid acceptance of a degenerate fundamental matrix if a number of points out of the dominant plane is negligible. Moreover, the method detects an image pair with no common field of view with close to zero false positive rate, and with zero false negatives on benchmark datasets. The VSAC exploits Gaussian elimination for a fast model estimation from a minimal number of points. A further speed-up is gained by modifying criteria for the local optimization to minimize excessive number of runs. An accurate final model parameter is found by efficiently implemented iterative least-squares. In VSAC, a substantial speed-up is achieved by adaptive sequential hypothesis verification.
Experiments on real-world publicly available datasets show that VSAC is significantly faster than all its predecessors and yet as precise as MAGSAC++, the currently most accurate state-of-the-art estimator of two-view geometry. In the repeated runs on EVD, HPatches, Photo-Tourism, StrechaMVS, and Kusvod2 datasets, it never failed.

Keywords: RANSAC, robust estimation, local optimization, homography, epipolar geometry.

Abstrakt

Random Sample Consensus je jedním z nejpopulárnějších a nejvíce používaných metod na robustní odhad v počítačovém vidění. Tato práce představuje VSAC (USAC 2021), který je zlepšený robustní odhadce podobný k RANSACu. Tato metoda přináší koncept nezávislých inlierů pro použití statistického přístupu na rozlišení náhodných modelů. Tento přístup se pomáhá vyhnout degenerované fundamentální matice, pokud počet bodů mimo rovinu je malý. Navíc, metoda detekuje dvojici obrázků, které nemají společné zorné pole, se skoro nulovou falešně pozitivnou mírou, a nulovým počtem falešných negativit na srovnávacích datových sadech. VSAC používá Gaussovou eliminační metodu na rychlý odhad modelů z minimálního vzorku. Další zrychlení je získáno pomocí změny podmínek na pouštění lokální optimalizace, aby se zmenšil počet její nadměrných běhů. Přesné parametry modelu jsou odhadované účinně implementovanou iterativní metodou na nejmenší čtverce. Ve VSACu podstatné zrychlení je dosahnuto s použitím adaptivního sekvenčního testu hypotéz.
Experimenty na veřejně dostupných datových sadech ukazují, že VSAC je podstatně rychlejší než všechny jeho předchůdce a stejně přesný jako MAGSAC++, nyní nejpreciznější odhadce geometrických struktur ze dvou pohledů. Na opakovaných bězích na EVD, HPatches, Photo-Tourism, StrechaMVS a Kusvod2 datových sadech, VSAC vždycky uspěl.

Klíčová slova: RANSAC, robustní odhad, lokalní optimalizace, homografie, epipolarní geometrie.

Contents

Abbreviations

GC-RANSAC	**G**raph-**C**ut RANSAC.
DEGENSAC	**DEGEN**nerate **SA**mple **C**onsensus.
RANSAC	**RAN**dom **SA**mple **C**onsensus.
PROSAC	**PRO**gressive **SA**mple **C**onsensus.
MLESAC	**M**aximum **L**ikelihood **E**stimation **SA**mple **C**onsensus.
MAGSAC	**MA**r**G**inalizing **SA**mple **C**onsensus.
SPRT	**S**equential **P**robability **R**atio **T**est.
USAC	**U**niversal RAN**SAC**.
SIFT	**S**cale **I**nvariate **F**eature **T**ransform.
IRLS	**I**terative **R**eweighted **L**east **S**quares.
PDF	**P**robability **D**ensity **F**unction.
CDF	**C**umulative **D**ensity **F**unction.
DLT	**D**irect **L**inear **T**ransformation.
EVD	**E**xtreme **V**iew **D**ataset.
SVD	**S**ingular **V**alue **D**ecomposition.
LSQ	**L**east **SQ**uares.
LO	**L**ocal **O**ptimization.
GT	**G**round **T**ruth.
GE	**G**aussian **E**limination.
Pn**P**	**P**erspective-n-**P**oint.
H, **F**, **E**	**H**omography, **F**undamental, **E**ssential.

1 Introduction

Robust estimation is undoubtedly an essential area for solving many tasks, including computer vision problems. The real world data are usually perturbed by inaccurate measurements (*i.e.,* outliers) regardless of whether it is a human mistake or not. A desired robust estimator must avoid outliers to provide an accurate result, *e.g.,* a median of data containing at most 50% of outliers is a reasonable estimate. In computer vision and image processing, a dimensionality and complexity of parametric estimation require more efficient methods.

The **RAN**dom **SA**mple **C**onsensus (RANSAC) algorithm presented by Fischler and Bolles [20] is one of the most popular robust method in computer science. It deals with data highly contaminated by outliers (*e.g.,* more than 50%), whereas other robust estimators like median fail. Therefore, it has become a widely used in computer vision where having imprecise and noisy data are widespread issues. The RANSAC is applied in wide / short baseline stereo matching [56, 58], image mosaicing [23], motion estimation [49], 3D reconstruction, detection of geometric primitives (*e.g.,* line, circles, homography), and structure, motion segmentation [56], etc.

A universal procedure of RANSAC allows to run it for a large range of estimation problems. Fischler *et al.* describe the algorithm as follows. At first, a minimal subset of points (*i.e.,* the sample) required for a model parameter estimation is drawn uniformly at random. Models estimated from the sample go through an evaluation process where each solution has a corresponding score – number of data points consistent with it (also inliers), *i.e.,* a model's support. If a model is *so-far-the-best*, that is, having the highest score so far, then it is getting saved. The RANSAC repeats each step in the loop until the probability of finding a better model falls below a user-defined tolerance level. Eventually, after the maximum number of iterations is reached, the final estimate is polished via least-squares fitting using all its support points.

As can be seen, the RANSAC can be divided into several main parts: sampling, model estimation, evaluation, verifying whether it is *so-far-the-best*, termination, and least-squares optimization. This separation is useful for further reading, as each part will be discussed in detail. Additionally, every component plays an important role for making RANSAC an efficient estimator.

There are certain features of RANSAC that make it more robust than other methods. First, a model parameter is estimated from a minimal sample, where points are drawn at random. In the presence of outliers, the least-squares method applied for large number of data points inevitably fails, as even one outlier destructs the estimation, while models from a minimal number of points are likely to be good. It means that for RANSAC, it is important to use as fewer points as possible for estimation to increase chances of a good sample. Second, the model evaluation is based on a robust 0 / 1 loss function which checks whether a point error distance to a current hypothesis falls below the inlier-outlier threshold. Depending on the ratio of correct points, most of the estimated parameters will have zero support (besides those points in sample), whereas good samples that generate better models create a competitive environment for the best model to be selected. Consequently, RANSAC seeks a model that fits the most data points. Finally, the maximum number of hypotheses (*i.e.,* samples, iterations) to be tested is well statistically justified by the upper bound equation in [20] which ensures that a solution will be found within the user-defined confidence.

In summary, RANSAC is a highly efficient robust estimator that works well even after 40 years since it was published. However, the amount of data and tasks to be processed have increased as computer vision has become widespread. This necessity for a better RANSAC, which can work faster and more accurately, motivates to present newer methods that will be discussed in the next section.

1.1 Related work

The RANSAC has undergone through many changes that can create a whole family of RANSAC-like methods. Modifications concern different parts of algorithm, the specific properties of an estimation problem, prior information, *etc*. The most significant and important updates will be addressed, starting not in chronological order but from the method's first step, *i.e.,* sampling.

A new sampling procedure based on *a priori* given point confidence was proposed in PROSAC by Chum *et al.* It exploits ranking of points to perform progressive sampling starting from the most promising ones, and it leads to finding a good model parameter much earlier. In NAPSAC [46] of Myatt *et al.* points in sample are drawn considering their proximity. In other words, the first point is selected uniformly at random as in RANSAC, while the rest of the subset points must be close to the first one. Such procedure has an advantage for searching a local structure, especially in high dimensional data, because close points are likely belong to the same substructure. Barath *et al.* in Progressive NAPSAC [3] combines both approaches to create a sampler that starts from local sampling and gradually converges to global sampling using ranking of points.

After sampling, several publications suggest verifying a minimal point subset to satisfy certain properties of the estimation problem. For instance, Marquez-Neila *et al.* in [47] proposes to check samples in homography estimation for the orientation constraint that 4 points on a plane must hold. Similarly, Chum *et al.* in [15] proposes a verification of epipolar geometry using its minimal sample for the orientation (chirality) constraint. These tests can significantly speed up RANSAC by reducing a number of model estimations and evaluations.

Many changes concern the model evaluation part. It was observed that evaluation is the most time-consuming comparing to sampling and estimation, because it depends on a number of data points. Therefore, different tests for pre-emptive model verification were proposed to reject bad solutions faster. Chum *et al.* in [12] proposed $T_{d,d}$ test which at first checks whether d out d randomly chosen points are consistent with the model, if it is true then the corresponding score is computed, otherwise the solution is rejected. However, this test leads to many false negatives (rejection of good models), hence later, a more accurate pre-emptive verification was introduced by Matas and Chum in RANSAC with sequential probability ratio test (SPRT) [43]. It starts evaluation on randomly shuffled points, during this process a model having low support is rejected using Wald's decision-making theorems, and the evaluation terminates. Another probabilistic approach for pre-emptive verification was presented, for instance, in RANSAC with bail-out [11] test of David Capel.

There are several proposals regarding the calculation of a model score, where in RANSAC it is a number of points that have an error distance w.r.t. model less than a user-defined inlier-outlier threshold. In MLESAC [57], Torr *et al.* assume that points' errors follow a mixture of Gaussian distribution for inliers and uniform for outliers. Consequently, using the maximum likelihood procedure, the score is computed with a log-likelihood function. The threshold can be a bottleneck for the whole RANSAC algorithm, since its wrong value leads to worse results. To avoid problems with a user-defined noise level, a-contrario RANSAC [19] by Epsuny *et al.* and MINPRAN [54] of Charles Stewart select internally the threshold value in a way inliers are least likely to occur at random. Barath *et al.* in MAGSAC [6] propose to marginalize the model's quality function of error distances over a range of noise levels.

The model only with the best score so far is accepted in RANSAC, however, it may result in saving degenerate solutions. Especially, it appears in fundamental matrix estimation, where points on a plane produce a degenerate epipolar geometry with a high support. Undoubtedly, this is a quite serious problem of the standard RANSAC as it yields to the failure of estimation. This issue was addressed in PLUNDER [59], and later in DEGENSAC [16] by Chum *et al.* who propose to check a fundamental matrix on degeneracy, and in the worst case recover it if possible. A universal approach for RANSAC in dealing with degenerate configurations was presented in QDEGSAC [21] by Frahm *et al.*.

Chum *et al.* in Locally Optimized (LO-)RANSAC [14] observed that models estimated from minimal samples in the presence could be inaccurate. Therefore, in [14] is proposed to refine a *so-far-the-best* model using a non-minimal subset of points consistent with it. This step leads the model parameter to be more accurate with a significantly increased number of inliers. The similar procedure was later presented in Graph-Cut (GC-)RANSAC [4] of Barath *et al.*, which additionally incorporates a spatial coherence of points, formulating the task as an energy minimization problem. In MAGSAC++[5], Barath *et al.* proposes an iterative re-weighted least squares optimization where weights are computed from the inlier probability of points, marginalized over a range of noise levels, similarly as in [6].

1.2 Motivation

The idea of a universal RANSAC integrating different *state-of-the-art* methods, and being able to efficiently solve one of the most popular computer vision problems as homography or epipolar geometry estimation appeared in USAC [51] by Raguram *et al.*. The estimator in [51] contains orientation tests, PROSAC, SPRT, DEGENSAC, and LO-RANSAC. Although, since then, new methods like MAGSAC++ and GC-RANSAC leading to better results were introduced. Moreover, a newer framework USACv20 [32] is providing more accurate solutions and working yet faster. Consequently, the necessity of RANSAC-like framework solving widespread estimation problems is actual.

Another problem of RANSAC is the ability to state whether the found solution is good, or it is just a random model with a non-zero number of inliers. In the real-world application, the maximum number of iterations for RANSAC is set by user, thus by finishing earlier than the upper bound equation predicts, it does not guarantee a correct estimate. Moreover, having a pair of images sharing no common field of view, all models found by RANSAC are bad, and yet they have points consistent with them.

To recognize a failure and random model, there is a need to distinguish the true inliers and ones that are by chance consistent with a model structure. Additionally, a degenerate configuration of points increases a support of bad hypotheses. For instance, assume a cluster of points in close proximity, and a line passing through them. The line's support using an inlier-outlier threshold includes all the cluster's points. However, two points that generate this line are always inliers, and the others lying closely to them are automatically included in the support of the line. The RANSAC can fail if the number of cluster's points exceeds the support of a true line.

1.3 Thesis contributions

This thesis presents a RANSAC-type estimator, VSAC, that exploits a number of novelties. It is significantly faster than all its predecessors, and yet as precise as MAGSAC++, currently one of the most accurate method both in the experiments and according to a recent survey [41]. The accuracy reaches, or is very near, the geometric error of the ground truth, estimated by cross-validation.

For homography ($\mathbf{H}$) and epipolar geometry ($\mathbf{F}$) estimation, VSAC runs on average several milliseconds (on a CPU) on all datasets, in orders of magnitude faster than MAGSAC++. In the repeated runs on datasets EVD [44], HPatches [2], PhotoTourism [53], StrechaMVS [55], and Kusvod2 [36], it never failed.

Moreover, VSAC is able to reject non-matching image pairs, with almost zero false positive rate on hundreds of random image pairs and a zero false negative rate on pairs from the above-mentioned datasets. The ability is underpinned by a novel concept of *independent random inliers* in the contrario context. We show that if dependent random inliers, *e.g.,* spatially co-located points, are not counted, the support of random models follows very closely a Poisson distribution with a single parameter λ that is easy to estimate reliably[1] for the given pair. The easily calculated CDF of Poisson raised to the power of the number of evaluated models provides the probability that a certain model quality was reached by chance. VSAC thus provides two confidence measures together with its result. The first is the classical one – the probability that RANSAC returned the model with the highest support. The second is the confidence that the returned solution was not obtained by chance.

The concept of independent random inliers plays a critical role in VSAC's improved accuracy and robustness. Experiments show that most failures of USAC-like methods for $\mathbf{F}$ estimation occur in the presence of a dominant plane, despite the DEGENSAC algorithm. In such cases, the number of out-of-plane inliers is often small, and due to structures within the outliers sets, incorrect models with high support exist. Removing the contribution of the dependent structures addresses the problem. Further improvements of dominant plane handling include a heuristic guess of the calibration matrix which allows dealing with fully planar scenes and detects pure rotation. If the guess is wrong, the support reveals it and nothing but a microsecond is lost.

The speed of VSAC is achieved with several technical improvements. Most significantly, we attack the problem of expensive local optimization. In the LO-RANSAC paper [14], the authors prove that the local optimization is run at most $\log(K)$ times, where K is the number of iterations. Nevertheless, despite $\log(K) \ll K$, the complex local optimization may end up being the efficiency bottleneck. We show that a fast local optimization combined with *a single complex final optimization* leads to a faster, yet equally precise algorithm. Moreover, by detecting the intersection over union of *so-far-the-best* and the current set of inliers and by not optimizing similar models, an algorithm is obtained that runs the local optimization on average about once and almost always fewer than two times.

Further speed up is gained by adaptive SPRT. The expected number of random inliers is estimated to tune SPRT [43] to the outlier density of the processed pair. We also measure, on the fly, the actual time of model estimation and model verification on the given hardware at the given moment, which is needed for calculating the quasi-optimal thresholds of the SPRT.

We also show that minimal model estimation via Gaussian elimination (GE) is highly efficient and surprisingly effective compared to SVD. In the context of RANSAC, where most models are computed from outlier-polluted data and therefore thrown away, the speed of GE more than compensates for its numerical issues, since imprecision or instability is addressed by local optimization or further sampling.

To broaden its application potential, VSAC provides novel outputs. Employing the highly efficient Lindstrom method for triangulation [38], it obtains the point pair exactly fitting the returned $\mathbf{F}$ that minimizes the geometric error. Similarly, correspondences are corrected for $\mathbf{H}$ by introducing a half homography, which is used to obtain error-free points. VSAC can be thus

[1] For geometric problems, the Poisson distribution is a tight approximation of the binomial. Moreover, only the mean λ of independent random inlier counts needs to be estimated, instead of T (number of trails) and δ (success probability) for the binomial.

employed for noise filtering.

Finally, the framework presents a number of minor novel features. The final model optimization is done by efficient iterative least-squares method. New solvers for a non-minimal essential matrix and minimal linear perspective-6-point algorithm are introduced. VSAC can filter incorrect correspondences by chance consistent with the found epipolar geometry. Additionally, the whole framework can run in parallel.

2 Non-randomness

One issue of RANSAC-like robust estimators is the inability to recognize failures. The estimator always returns a model maximizing some quality function, *e.g.,* the inlier count. Points consistent with a model parameter that stem from outlier structures – sets of neighboring data points that do not originate from the sought model manifold, significantly affect the quality function when considered as inliers. In such cases, the returned model might have a reasonably large number of inliers while being inconsistent with the underlying scene geometry. See figure 2.1 for examples.

This chapter aims to address this problem. At first, the concept of *independent* inlier will be presented. Further, this work will show its influence on the detection of failure by building a statistical model of one estimated parameter. Finally, the proposed approach will be integrated into pre-emptive verification making it more accurate than it was described in [43].

2.1 Independent inliers

In this section, we propose a new approach for failure model detection. Towards that end, we differentiate between *independent* and *dependent* inliers. This split is conceptual, helping exposition – in the a contrario calculation of the probability, we pick one point in a group of structured points and count it as an inlier arising by chance, it is an independent random inlier. The other inliers in the structure are ignored, since their inlier status is not a random event, but rather a consequence of their spatial dependence and the fact that the independent inlier is consistent. A *non-random* model must have a sufficient number of *independent* inliers.

We define data point $\mathbf{p}$ a *dependent* inlier if its point-to-model residual is smaller than the inlier-outlier threshold and one of the following conditions hold.

1. Point $\mathbf{p}$ is in the minimal sample used for estimating the model parameters. In such cases, the point will have zero residual by definition.
2. Point $\mathbf{p}$ is close to an independent inlier $\mathbf{q}$, $||\mathbf{p} - \mathbf{q}|| \to 0$. In such cases, points $\mathbf{p}$ and $\mathbf{q}$ form a spatial structure that affects the model quality significantly. Thus, only point $\mathbf{q}$ is considered independent random. Other points from the structure, *e.g.,* $\mathbf{p}$, are dependent inliers.

These conditions are valid for general data points and model to be estimated. In case of estimating epipolar geometry from point correspondences, we define the following additional conditions as well.

3. A correspondence $(\mathbf{p}, \mathbf{p}')$ where $\mathbf{p}$ or $\mathbf{p}'$ is close to the epipole in the corresponding image is considered dependent since $(\mathbf{p}, \mathbf{p}')$ always satisfies epipolar constraint $\mathbf{p}'^\top \mathbf{F}\mathbf{p} = 0$. This stems from the fact that $\mathbf{F}\mathbf{p} \backsimeq \mathbf{F}\mathbf{e} = \mathbf{0}$ if $||\mathbf{p} - \mathbf{e}|| \to 0$, where $\mathbf{e}$ is the epipole in the first image. The same holds in the second one.
4. Correspondence $(\mathbf{p}, \mathbf{p}')$ is a dependent inlier if it does not pass the chirality check [15]. In this case the point is not correct, thus it is just by a chance consistent with epipolar geometry.
5. Let $(\mathbf{l}, \mathbf{l}')$ be the corresponding epipolar lines of an independent inlier correspondence. All correspondences that are closer to lines $(\mathbf{l}, \mathbf{l}')$ than the inlier-outlier threshold are considered dependent (see appendix 11.1);

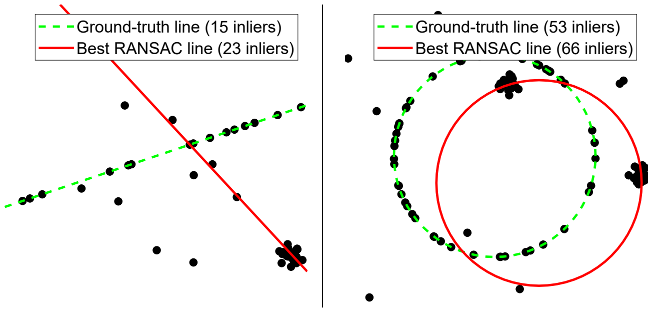

Figure 2.1 Examples of random models found by RANSAC with a large number of dependent inliers.

Data points that have a point-to-model residual smaller than the inlier-outlier threshold and do not satisfy any of the previous conditions are considered *independent* inliers.

2.2 Failure detection

To decide if a model estimated by RANSAC is random and inconsistent with the underlying scene geometry and, thus, should be considered failure, we use the univariate theory [52]. Suppose that we are given N models estimated inside RANSAC during its run and the corresponding numbers of random independent inliers $I_1, I_2, \dots, I_N \in \mathbb{N}$. The number of I points is consistent with a random model follows the binomial distribution. The sequence of inliers is an *i.i.d.* random variable with cumulative binomial distribution $C_{\mathcal{B}}(T, \delta)$, where T is the number of points and δ is the probability that a point is an independent inlier to a bad (random) model. The distribution of $I_{\max} = \max\{I_1, I_2, \dots, I_N\}$ over N models is $C_{\mathcal{B}}(T, \delta)^N$. In order to recognize a good (non-random model) with confidence $p \to 1$, the following condition must hold.

$$C_{\mathcal{B}}(I_{\max}; T, \delta)^N \geq 1 - p. \tag{2.1}$$

In our experiments, we found that probability δ is fairly low, *i.e.,* bad models usually have just a few *independent* inliers. In this case, the binomial distribution can be approximated by Poisson distribution which is faster to compute. The only parameter of the Poisson distribution is $\lambda = T\,\delta$ that represents the mean number of independent inliers of bad solutions. Finding independent inliers for every model generated in RANSAC is inefficient and redundant, because a good estimate could be obtained even from a small number of hypotheses. Hence, we estimate parameter λ from the first $n \ll N$ generated models. Parameter λ is the mean number of independent inliers consistent with a bad model. In RANSAC, all models are considered bad that have fewer inliers than the *so-far-the-best* one. However, for estimation of λ this is not the case since models with fewer inliers could be good non-random structures, thus they can destruct the λ estimate. Table 2.1 shows that even not all-inlier samples can generate solutions with high number of *independent* inliers. Therefore, the following robust approach for finding a Poisson parameter is needed:

1. Discard from n *independent* inlier counts a support of *so-far-the-best* model and inlier numbers of models that have significant overlap of inlier sets (Jaccard similarity [34]) with it.
2. Select $\tilde{\lambda}$ as a median number of the remaining values.
3. Compute 95% percentile ($I_{95}^{\%}$) of Poisson distribution with parameter $\tilde{\lambda}$.
4. Calculate $\hat{\lambda}$ as the average of *independent* inlier numbers lower than $I_{95}^{\%}$.

This procedure removes outliers (good structures with high support) using $I_{95}^{\%}$ that can spoil the estimation of $\hat{\lambda}$. Additionally, it confronts models that have similar inlier sets as the *so-*

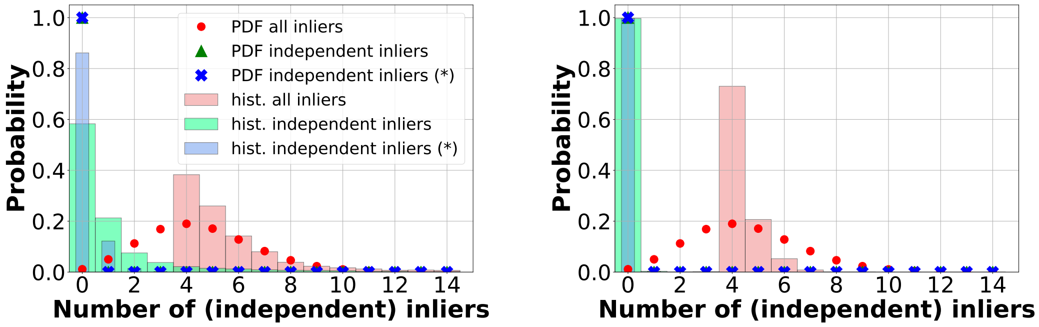

Figure 2.2 Probability histograms of numbers of all (red bar) and independent (green and blue bars) inliers of all models estimated in RANSAC with the fixed 10^5 number of iterations; and Poisson distribution with its parameters calculated from all (red circle) and from the independent inliers (green triangle and blue cross) on a scene from EVD dataset (left) and an image pair without common field of view (right). The blue histogram unlike to the green one shows independent inliers of models generated by artificially modified bad samples. Both histograms are related to homography estimation.

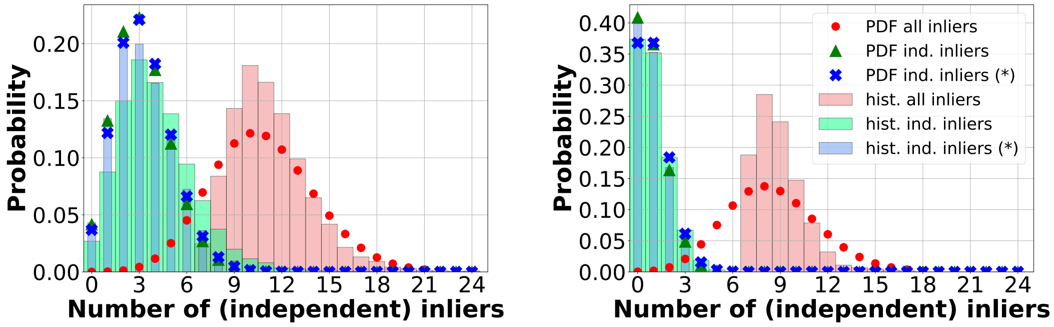

Figure 2.3 Histograms of (independent) inliers for fundamental matrix estimation. The histogram on the left is for an image pair ("Sacre Coeur") from PhotoTourism dataset, and the right plot is for an image pair with no common structure.

far-the-best, as they represent the same structure. It is done not only for an accurate detection of random solutions, but mainly for avoiding false negative results (detecting a good model as a random). If number of good structures among the first n is more than half, then the test for non-randomness is unnecessary. However, a precise Poisson parameter is also required for another reason, which will be described later.

Finally, to decide if the model returned by RANSAC is non-random and, thus, should be accepted, the cumulative distribution of $(\max\{I_1, \dots, I_N\}; \hat{\lambda})^N$ is calculated, and the model is considered a good one if (2.1) holds.

The proposed test has a false negative case in the extreme scenario, when multiple structures with no inlier overlap exist, and they have a similar number of *independent* inliers. To avoid this issue and improve estimation of $\hat{\lambda}$, this work suggests to deliberately generate a bad sample points, *e.g.,* by swapping coordinates of correspondences on the second image – correct points become bad, and incorrect matches remain bad. Therefore, the found number of points consistent with bad models will be close to the theoretical one. In table 2.1 are reported average numbers of *independent* inliers of corrupted samples and samples with no good point in it, they are very close.

The *independent* inliers play a crucial role in RANSAC failure detection. As is shown in figure 2.1, dependent inliers (*e.g.,* close points lying in clusters) increase supports of RANSAC

	Number of points consistent with GT model in sample, i								
	0*	0	1	2	3	4	5	6	7
F	7.39±2	8.67±3	11.64±5	16.77±8	26.10±13	40.14±24	63.56±51	>90	>100
H	0.50±1	0.18±1	0.75±1	3.96±4	23.01±24	>300	-	-	-

Table 2.1 In rows are mean numbers of independent inliers averaged over 50 images pairs with their standard deviation. The means of independent inliers are computed over 5000 samples containing only i points consistent with GT model; for each i from 0 to 7, where 0^* denotes artificially modified bad samples. The rows show results for $\mathbf{F}$ (7-point) – fundamental matrix and $\mathbf{H}$ (4-point) – homography problems.

models, which makes it more likely to output degenerate solutions. By removing those dependent points, the proposed test can calculate the true support of the final model and distinguish it from a random one. Most importantly is that the dependent inliers of bad models do not follow the theoretical Poisson distribution. For instance, in figure 2.4 is shown a detected invalid homography matrix, which has high support of degenerate correspondences in an image pair with no common field of view.

Figures 2.3-2.2 show histograms of independent and dependent inliers for an image pair with a common field of view and a pair without common structure, for fundamental and homography matrix estimation. Additionally, figures demonstrate Poisson distribution (PDF) calculated via the proposed method using independent and dependent inliers. In all scenarios, Poisson PDF estimated from all (including dependent) inliers does not fit histogram of inliers. For the image pair without common field of view, histograms of independent inliers have a perfect fit. For the image pair with the common structure, the estimated Poisson PDF does not fit histogram of independent inliers well due to samples perturbed by good points (see table 2.1). However, it fits a histogram of independent inliers of bad samples that implies that the estimation of Poisson parameter was correct.

2.3 Adaptive SPRT

The Sequential Probability Ratio Test (SPRT) proposed by Matas *et al.*[43] aims at speeding up the robust estimation procedure by addressing the problem that, in RANSAC, a large number of models are verified, *e.g.,* their support is calculated, even if they are unlikely to be better than the previous *so-far-the-best*. The time spent on these models is wasted. The SPRT is based on Wald's theory of sequential decision making. It interrupts the model verification when the probability of that particular model being a *good* one falls below threshold, which is optimally found inside the method.

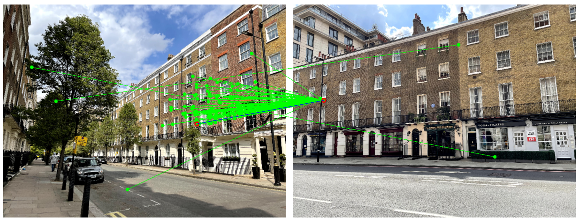

Figure 2.4 Homography estimation on an image pair sharing no common scene matched by SIFT detector. Green circles (95) are points consistent with the final model, the only red one – independent inlier. The model is predicted to be random.

SPRT has four user-defined parameters, *i.e.,* the initial probability of a correspondence being consistent with a *good* (ϵ_0) and a *bad* model (δ_0); avg. number of estimated models ($\bar{m}_{\mathcal{S}}$); time to estimate the model parameters (t_M). The actual parameters that lead to the fastest procedure are challenging to find manually and require a user to acquire knowledge about the problem at hand. Even the architecture of the computer impacts the minimal solver and point verification times that should be considered when setting t_M. To avoid the manual setting, we propose the Adaptive SPRT (A-SPRT) algorithm that finds the optimal SPRT parameters in a data- and architecture-dependent manner.

The model estimation time depends not only on the computer architecture, but also on the actual solver and error metric being used. Parameter t_M is calculated for free by measuring the model estimation and point verification run-times in the first k RANSAC iterations. **The average number** of models $\bar{m}_{\mathcal{S}}$ is found as the average number of *valid* models per sample in the first k iterations. **Inlier probabilities** ϵ_0 and δ_0 are estimated from the average number $\hat{\lambda}$ of inliers consistent with a bad model that is estimated from in the first n RANSAC models as $\delta_0 = \hat{\lambda} \, / \, T$, see section 2.2. Parameter δ_0 is the probability of a point being an inlier and $\hat{\lambda}$ is the mean of the corresponding binomial distribution $\mathcal{B}(T, \delta_0)$. From δ_0, we approximate the maximum number of inliers I_δ of a *bad* model as a high quantile (*e.g.,* 0.99) of the normal distribution with the same mean and standard deviation as $\mathcal{B}$. The approximation is as follows:

$$I_\delta = \hat{\lambda} + 3.719 \sqrt{\hat{\lambda} \, (1 - \delta_0)} \tag{2.2}$$

The initial probability ϵ_0 of a correspondence being inlier can be calculated using inlier number of *so-far-the-best* model. However, δ_0 gives an upper bound number of inliers of bad structures. Therefore, a *good* model must have at least number of inliers of *so-far-the-best* and the random one:

$$\epsilon_0 = \max(I_\delta, \hat{I}^*) \, / \, T, \tag{2.3}$$

where $\hat{I}^*$ is the inlier number of the *so-far-the-best* model.
If probabilities δ_0 and ϵ_0 are similar, the original SPRT is prone reject *good* models leading to increased run-time or, in extreme situations, total failure. To solve this issue, we propose to apply A-SPRT only if:

$$\frac{1}{1 - \alpha} \, t_v^w \, E^w(T) < t_v \, T, \tag{2.4}$$

where t_v^w and t_v are the times for verifying a single correspondence, respectively, with and without SPRT, and α is the probability of a false rejection [43], and $E^w(T)$ is the average number of points verified.

The new A-SPRT incorporates the Poisson parameter to estimate the probability of a point being consistent with a bad model. Correct estimation of $\hat{\lambda}$ decreases the number of good model rejections made by pre-emptive verification. Moreover, the right $\hat{\delta}_0$ probability could speed-up rejection of bad hypotheses. This is another reason for excluding models with high inlier count in $\hat{\lambda}$ estimation.

3 Degeneracies in epipolar geometry

Handling degenerate solutions and degenerate configuration of points is an important task. The standard RANSAC proposed by Fischler *et al.* is not safe against invalid models that may appear as ones with high support. It is explained by a score-driven acceptance of a *so-far-the-best* model, making it the only criterion to decide whether the model is good. As a result, for some data measurements, RANSAC without degeneracy checks may inevitably fail to find the correct model.

For instance, in line estimation, two identical points cannot generate a unique line; for a planar homography estimation, three collinear points in a 4-point sample cause a degenerate rank-deficient solution, which projects points to lines. In previous examples, the degenerate cases could be detected by preprocessing sample points to check whether they are in general position. However, for a fundamental matrix ($\mathbf{F}$) estimation, degenerate configurations are more sophisticated, the most common is when points of a minimal sample lie on a dominant plane. Since the fundamental matrix describes a 3D structure, a planar point set does not generate a valid solution. Additionally, the difficulty of the task is that no prior segmentation of points is usually given, while the occurrence of planes is widespread, especially in the urban environment.

The degenerate configuration of fundamental matrix was studied, *e.g.,* in PLUNDER [59] or DEGENSAC [16]. The latter work by Chum *et al.* shows that 5 correspondences lying on a dominant plane in a 7-point minimal sample could originate a degenerate fundamental matrix. In addition, DEGENSAC presents the whole approach to detect and recover a degenerate fundamental matrix, by trying to efficiently find a homography using correspondences of the minimal $\mathbf{F}$ sample.

Some degenerate configurations can be detected, avoided, and invalid solutions could be even recovered, however, there are rare cases, when input data points inevitably lead to degenerate structures. For instance, if a camera undergoes pure rotation, then all correspondences in a two-view scene are consistent with homography, hence no epipolar geometry can be estimated. Another critical scenario for $\mathbf{F}$ estimation is when a camera captures a fully planar scene.

Further, the reader will be introduced with a new DEGENSAC$^+$ method, which efficiently detects and recovers degenerate fundamental matrices. If calibration is given, the proposed estimator guarantees a non-degenerate solution or report a pure-rotation scene. Moreover, it incorporates the non-randomness criteria described in previous chapter 2 to report if the recovery of the degenerate fundamental matrix fails.

3.1 Non-randomness under degeneracy

The DEGENSAC of Chum *et al.* is an efficient estimator which is used in many RANSAC implementation, including *state-of-the-art* USAC framework [51]. The method, at first, tries to estimate homography using 3 correspondences from a 7-point sample. Second, runs plane-and-parallax RANSAC to find 2 points out of a plane that together with estimated homography generate a new non-degenerate fundamental matrix. However, a problem appears for the plane-and-parallax RANSAC when a number of good points out of the plane is low, thus the method could fail. The DEGENSAC is not able to recognize the failure since the found model has a

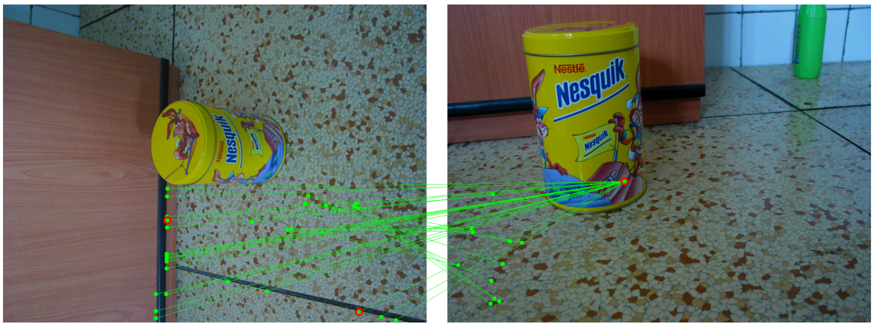

Figure 3.1 An image pair (”Box”) from KUSVOD [36] dataset shows two incorrect matches (red circles) found by plane-and-parallax RANSAC. All non-planar inliers of the wrong hypothesis (with the highest found support) are green bold circles. The dominant plane is a floor of the image.

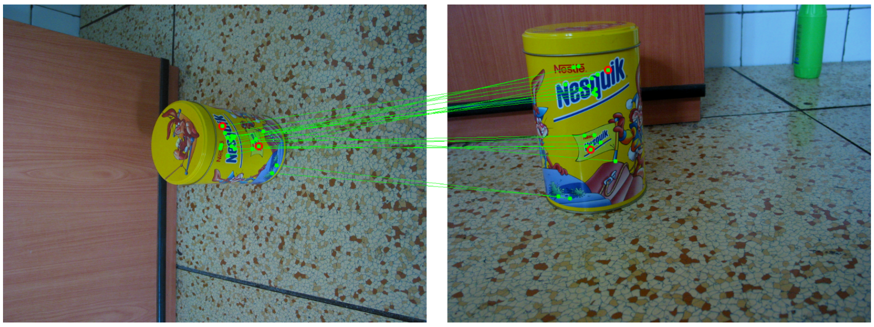

Figure 3.2 Plane-and-parallax RANSAC based on calculating *independent* non-planar support found two right correspondences out of the dominant plane. The fundamental matrix has the highest number of *independent* inliers out of the plane.

non-zero support of randomly consistent points. Consequently, the final model remains degenerate, which leads to the total failure of $\mathbf{F}$ estimation.

The non-randomness test proposed in 2 should be emplaced into plane-and-parallax RANSAC to detect a failure output. Moreover, when a number of points out of a single plane is small, then it becomes extremely important to distinguish among dependent and independent inliers. The rules for independent support are the same as is described in 2.

A figure 3.1 demonstrates the failure of two-point plane-and-parallax RANSAC without reducing dependent inliers. The majority of points on the left image incorrectly match one point on the right image, hence, all are dependent. The found fundamental matrix has the highest number of all inliers and yet being still degenerate. The same RANSAC including computation of independent inliers managed to find correct $\mathbf{F}$, its non-planar independent support is visualized in figure 3.2.

The problem can be also defined as follows. A set of point correspondences $\mathcal{T}$ can be divided into two sets $\mathcal{T}_{\mathbf{H}}$ and $\mathcal{T}_{\mathbf{H}}^c$. Where $\mathcal{T}_{\mathbf{H}}$ is set of points consistent with homography $\mathbf{H}$; and ^{c} stands for the complement set, that is, $\mathcal{T}_{\mathbf{H}}^c$ is a set correspondences not consistent with $\mathbf{H}$. A set $\mathcal{T}_{\mathbf{F}}$ of points consistent with fundamental matrix must satisfy:

1. $\mathcal{T}_{\mathbf{F}} \supseteq \mathcal{T}_{\mathbf{H}}$ – points consistent with homography are also consistent with fundamental matrix.
2. $\mathcal{T}_{\mathbf{F}} \cap \mathcal{T}_{\mathbf{H}}^c \neq \varnothing$ – fundamental matrix has non-planar support.

While the standard RANSAC intends to find a fundamental matrix $\hat{\mathbf{F}}$ which maximizes $|\mathcal{T}_{\hat{\mathbf{F}}}|$, the plane-and-parallax RANSAC should maximize $|\mathcal{T}_{\hat{\mathbf{F}}} \cap \mathcal{T}_{\hat{\mathbf{H}}}^c|$ of estimated homography $\hat{\mathbf{H}}$. However, wrong matches $\mathcal{T}_O$ tangle the objective, as they are non-planar and could be consistent with epipolar geometry, hence $(\mathcal{T}_{\hat{\mathbf{F}}} \cap \mathcal{T}_{\hat{\mathbf{H}}}^c) \cap \mathcal{T}_O \neq \varnothing$. Therefore, finding a non-degenerate

$\hat{\mathbf{F}}$, *i.e.,* a pair of non-planar matches, is a subtle problem. Outliers that are randomly consistent with an epipolar geometry could outnumber good non-planar support, leading to the incorrect solution as happened in figure 3.1.

The non-randomness criteria could be applied to verify a homography matrix. This is important since $\mathbf{H}$ is mainly used to recover a degenerate fundamental matrix. Partly, DEGENSAC checks $\mathbf{H}$ to have at least 5 inliers in 7-point sample, which determines the degeneracy of $\mathbf{F}$. However, there is no guarantee that these 5 inliers are not dependent. This issue will be addressed further in this chapter, suggesting that a homography should have a significant independent inlier support.

3.2 False negatives

In the texbook DEGENSAC, if 5 correspondences of 7-point fundamental matrix sample are consistent with homography then plane-and-parallax is applied, nonetheless, it does not necessarily mean that $\mathbf{F}$ is degenerate. One of the reasons for this is that two other matches in sample can be incorrect. However, this work has experimentally found that there are cases when a homography related samples generate good epipolar geometry.

By a simplification, in this chapter, a subset of image correspondences consistent with $\mathbf{H}$ is called planar, although its points almost never lie precisely on a plane. There are several reasons for this, *e.g.,* many planes in real world are not flat and contain some relief (*e.g.,* windows on a facade). For distant planes, the relief may not be visible, however, inaccuracies of matches (especially corner points) and their noise caused by point-detector spawn correspondences that do not lie exactly on plane. Moreover, thresholding a reprojection distance of a correspondence is not the most accurate way to check its consistency with a plane. Therefore, it suggests that a test to avoid wrong assumption on a model degeneracy is needed.

This section proposes to verify a fundamental matrix itself on a sufficient non-planar independent inliers. It aims to eliminate a number of false negatives cases (incorrect classification of good $\mathbf{F}$ as a degenerate), hence reduce a number of redundant plane-and-parallax runs. Initially, the sufficient number of independent inliers can be set as a lower bound of a non-degenerate solution. Later, it is estimated as an average during two-point RANSAC, the same way as is described in chapter 2.

In figure 3.3 is shown a non-degenerate fundamental matrix originated by a sample with all 7 points related by homography, however the correspondences are located on a raised facade. Independent support off the plane of $\mathbf{F}$ is sufficient, hence, the fundamental matrix does not need to be recovered.

3.3 Calibrated DEGENSAC

This section presents a new DEGENSAC$^+$ method based on the original DEGENSAC. It addresses the issue of finding non-degenerate fundamental matrix that has a sufficient number of independent inliers out of plane. Mainly, the procedure decreases a number of returning degenerate models (*i.e.,* failures) and number of plane-and-parallax executions. The proposed algorithm also uses intrinsic matrices that are sometime available, *e.g.,* in a vehicle's ego-motion. Even if calibration is not known, the method is able to approximate intrinsic matrices, which can be also beneficial for users. The calibrated DEGENSAC$^+$ has the following advantages over the original method:

(a) It can recover epipolar geometry if the scene is planar.
(b) It can distinguish pure rotation.

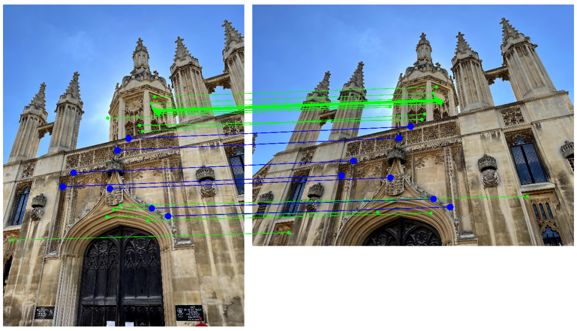

Figure 3.3 Seven matches (blue circles) of a minimal $\mathbf{F}$ sample are consistent with homography that was found in DEGENSAC$^+$. Green points are independent non-planar support of $\mathbf{F}$, which is not degenerate.

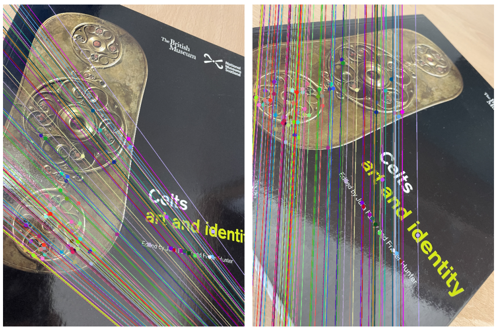

Figure 3.4 An image pair of a book shows epipolar lines of non-degenerate fundamental matrix and its inliers. $\mathbf{F}$ was found via homography decomposition and intrinsic matrix approximation in DEGENSAC$^+$.

(c) It runs significantly faster and in constant time.

(d) A failure of the method, *i.e.,* finding a bad model, is predicted by the non-randomness test.

Algorithm 1: DEGENSAC$^+$

```
Input: $\hat{\mathbf{F}}^*$ – so-far-the-best fundamental
       matrix; $\mathcal{S}$ – minimal sample
       initializing $\hat{\mathbf{F}}^*$, $I^c_{\hat{\mathbf{H}}}$ – initial guess for
       independent non-planar support.
1  $\hat{\mathbf{H}} \leftarrow$ estimateHomography $(\hat{\mathbf{F}}^*, \mathcal{S})$
2  if $\hat{\mathbf{H}} = \varnothing$ then
3      return: $\hat{\mathbf{F}}^*$ // no homography
4  $\mathcal{T}^c_{\hat{\mathbf{H}}} \leftarrow$ nonPlanarPoints$(\hat{\mathbf{H}})$
5  if support $(\hat{\mathbf{F}}^*, \mathcal{T}^c_{\hat{\mathbf{H}}}) \geq I^c_{\hat{\mathbf{H}}}$ then
6      return: $\hat{\mathbf{F}}^*$ // not degenerate
7  if $\mathbf{K}$ is given then
8      assert $(\mathbf{K}^{-1}\hat{\mathbf{H}}\mathbf{K}$ is not conj. to rotation)
9      $\hat{\mathbf{F}} \leftarrow$ findF $(\hat{\mathbf{H}}, \mathbf{K})$
10     return: $\hat{\mathbf{F}}$ //cannot be degen.
11 else
12     $\hat{\mathbf{F}}' \leftarrow$ findF $(\hat{\mathbf{H}}, \hat{\mathbf{K}})$ //use approx $\hat{\mathbf{K}}$
13 if support$(\hat{\mathbf{F}}', \mathcal{T}^c_{\hat{\mathbf{H}}}) \geq I^c_{\hat{\mathbf{H}}}$ then
14     return: $\hat{\mathbf{F}}'$
15 // re-estimate support $I^c_{\hat{\mathbf{H}}}$
16 $\hat{\mathbf{F}}'', I^c_{\hat{\mathbf{H}}} \leftarrow$ planeAndParallax $(\mathcal{T}^c_{\hat{\mathbf{H}}}, \hat{\mathbf{H}})$
17 if support $(\hat{\mathbf{F}}'', \mathcal{T}^c_{\hat{\mathbf{H}}}) \geq I^c_{\hat{\mathbf{H}}}$ then
18     return: $\hat{\mathbf{F}}''$
19 return: $\varnothing$ // reject $\hat{\mathbf{F}}^*$
```

The DEGENSAC$^+$ is outlined in the algorithm 1. The method's input is *so-far-the-best* fundamental matrix $\hat{\mathbf{F}}^*$, the corresponding 7-point sample $\mathcal{S}$, and an initial guess for the sufficient number of independent non-planar inliers of fundamental matrix ($I^c_{\hat{\mathbf{H}}}$). The latter value will be estimated further in the method, however, in the beginning it is an expected number that assures non-degeneracy, *e.g.,* 20 independent non-planar inliers, while degenerate $\mathbf{F}$ has zero excluding randomly consistent ones. The algorithm starts with an estimation of a homography matrix using 3 correspondences from $\mathcal{S}$ and $\hat{\mathbf{F}}^*$ (line 1). If homography is found, it separates points not consistent with $\mathbf{H}$ into a set $\mathcal{T}^c_{\hat{\mathbf{H}}}$ of tentative points off a plane (see line 4). This set is used primarily to calculate independent non-planar *support* of a fundamental matrix, and if necessary for the two-point sampling in plane-and-parallax RANSAC. In the line 5 the input $\hat{\mathbf{F}}^*$ is verified to be non-degenerate by checking whether it has a sufficient number of non-planar inliers. Next comes a novel part, where the fundamental matrix can be recovered with $\hat{\mathbf{H}}$ and calibration.

Given a homography and (approximated) intrinsic matrices $\mathbf{K}_1, \mathbf{K}_2 \in \mathbb{R}^{3\times3}$ the method is able to recover a fundamental matrix via $\mathbf{H}$ decomposition described in [42] by Malis *et al.* Exploiting calibration brings several advantages over the plane-and-parallax RANSAC. First, the decomposition takes a constant time, whereas plane-and-parallax repeatedly draws two correspondences out of plane. Second, more importantly, the new algorithm can handle pure rotation and scene containing only dominant plane. No translation case implies that all matches are consistent with homography, therefore, in both cases the original DEGENSAC fails, because no point off a plane exists.

According to [42], a homography $\mathbf{H}$ using calibration can be decomposed into rotation $\mathbf{R} \in$ SO(3) and translation $\mathbf{t} \in \mathbb{R}^3$ with four possible pairs $(\mathbf{R}_a, \pm\mathbf{t}_a), (\mathbf{R}_b, \pm\mathbf{t}_b)$. The fundamental matrix is then found by composing $\hat{\mathbf{R}} \in \{\mathbf{R}_a, \mathbf{R}_b\}$, $\hat{\mathbf{t}} \in \{\pm\mathbf{t}_a, \pm\mathbf{t}_b\}$, $\mathbf{K}_1$, and $\mathbf{K}_2$. Since the fundamental matrix is given up-to-scale then two solutions (*e.g.,* for $-\mathbf{t}_a$ and $-\mathbf{t}_b$) can be eliminated, and the best one is selected, *e.g.,* with the maximum inlier count.

$$\mathbf{F} = \mathbf{K}_2^{-\top}[\hat{\mathbf{t}}]_\times \hat{\mathbf{R}} \mathbf{K}_1^{-1} \tag{3.1}$$

In many cases, the calibration is not known a priori but can be approximated. Setting the principal points to coincide with the image centers is a widely used approach. An estimation of the focal length is, however, a more challenging problem. Testing a range of candidate focal lengths to find a reasonable approximation leads to a good $\mathbf{F}$ estimate while still being faster than the original plane-and-parallax RANSAC. The approximation procedure does not aim to find a precise calibration since the only goal is to produce a non-degenerate fundamental matrix, thus the same focal length for x and y axes is assumed. Moreover, even if a non-degenerate $\mathbf{F}$ is consistent with a wrong pair of intrinsic matrices, the final LSQ optimization generates a better $\mathbf{F}$ where this consistency is lost.

In figure 3.4 is an image pair with a dominant plane. For this problem, the plane-and-parallax RANSAC inevitably fails to find a pair of non-planar correspondences. While, the approximation approach returns intrinsic matrix with the best found focal length equals to 3000 (tested with the step 100) and the principal point – (2016, 1512) pixels. The ground truth calibration matrix has focal length for $x-$axis equal to 3075.3, for $y-$axis it is 3074.4, and the principal point is (1995.3, 1531.8) px. An estimated relative pose after homography decomposition is also close to the ground truth one.

Nevertheless, DEGENSAC$^+$ does not completely rely on approximated instrinsic matrices, as they could be imprecise. First, they are not used for pure-rotation detection. Second, a generated fundamental matrix from approximated calibration is verified for independent non-planar support (see line 13 in Alg. 1). If the model does not have sufficient support, then plane-and-parallax RANSAC runs, and additionally, the sufficient number of independent inliers out of the plane is re-estimated. In case true intrinsic matrices with correct homography are used, they cannot produce a degenerate epipolar geometry.

All computations in DEGENSAC$^+$ highly depend on a quality of an estimated homography. It is important that $\mathbf{H}$ is not a random structure, but the one having a significant support. Therefore, the estimation of homography includes the LSQ optimization. Additionally, two inlier-outlier thresholds to check point consistency with $\mathbf{H}$ (plane) are introduced. The first threshold is used to measure the quality of $\mathbf{H}$ itself, its value is, *e.g.,* 2.5 pixels. The MLESAC evaluation kernel can be applied since the goal is to find as good homography as possible. The second threshold reflects the confidence of a correspondence belonging to plane, and its value should be higher, *e.g.,* 10 px. The reason is that if the homography is still not accurate enough, then many planar points can have reprojection distances higher than the first threshold. However, for the plane-and-parallax RANSAC, it is better to filter out all possible planar points to avoid their selection in two-point sampling. Note, the idea of the maximum threshold is also mentioned in MAGSAC [6]. Although, its purpose is for a model evaluation to marginalize error distances over the noise level.

3.3.1 Pure rotation detection

This section shows how to detect a relative pose that does not have camera translation. Let $(\mathbf{x}, \mathbf{x}')$ be a point correspondence, $\mathbf{K}_1, \mathbf{K}_2$ are the intrinsic matrices, $\mathbf{R}_1, \mathbf{R}_2$ are the camera rotations, X is the unknown 3D object point, and a scene undergoes pure rotation. In this case, the following projection equations hold:

$$\mathbf{x} \sim \mathbf{K}_1 \, \mathbf{R}_1 \, X, \quad \mathbf{x}' \sim \mathbf{K}_2 \, \mathbf{R}_2 \, X \tag{3.2}$$

$$\mathbf{x}' \sim \mathbf{K}_2 \, \mathbf{R}_2 \, \mathbf{R}_1^\top \, \mathbf{K}_1^{-1} \, \mathbf{x} \tag{3.3}$$

where operator $\sim$ means equality up-to-scale. The equation above shows relation of correspondence $\mathbf{x}$ to $\mathbf{x}'$ after eliminating the object point X. Homography $\mathbf{H} = \mathbf{K}_2 \mathbf{R}_2 \, \mathbf{R}_1^\top \mathbf{K}_1^{-1}$ transforms image points as:

$$\mathbf{x}' \sim \mathbf{H} \, \mathbf{x} \tag{3.4}$$

The DEGENSAC$^+$ aims to find image homography using *so-far-the-best* $\mathbf{F}$ and triplet of matches from a 7-point sample. In the normalized by $\mathbf{K}_1$ and $\mathbf{K}_2$ point coordinates, a homography $\tilde{\mathbf{H}}$ is conjugated to rotation:

$$\tilde{\mathbf{H}} = \mathbf{K}_2^{-1} \mathbf{H} \mathbf{K}_1 = \mathbf{R}_2 \mathbf{R}_1^\top = \mathbf{R} \tag{3.5}$$

To obtain $\tilde{\mathbf{H}}$, the homography $\mathbf{H}$ is multiplied by the $\mathbf{K}_1$ and $\mathbf{K}_2^{-1}$, and the scale of $\tilde{\mathbf{H}}$ is removed (*e.g.,* via SVD). The conjugated to rotation homography must be orthonormal matrix, *i.e.,* $\tilde{\mathbf{H}}^\top \tilde{\mathbf{H}} = \mathbf{I}$. Therefore, it is enough to check whether matrix product is close to identity matrix, that is $||\tilde{\mathbf{H}}^\top \tilde{\mathbf{H}} - \mathbf{I}||_{\mathcal{F}} < \epsilon$, where $||.||_{\mathcal{F}}$ stands for Frobenius norm and ϵ is a tolerance threshold. When a homography conjugated to a rotation is detected, RANSAC can terminate and there is no epipolar geometry to be outputted.

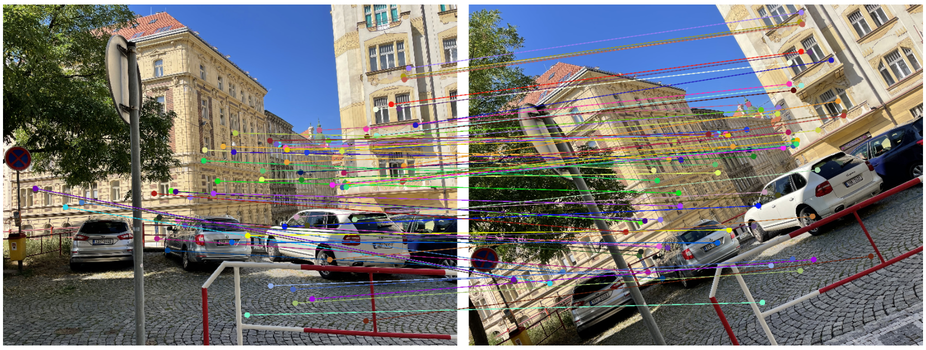

Figure 3.5 An image pair of a scene without camera translation demonstrates matches (colorful circles) related by homography conjugated to rotation. This case was detected by the proposed in this section test.

4 Solver

This chapter discusses a model parameter estimation from a minimal and non-minimal number of points. In RANSAC, the estimation is one of the most time-consuming parts. Undoubtedly, it is yet one of the most important parts since finding a model affects the overall accuracy of RANSAC. This chapter describes a desired *solver* that aims to speed up RANSAC, simultaneously without the final accuracy trade-off. The *solver* itself is divided into two parts: *minimal* – deals with the minimum number of points required for a specific task, and *non-minimal* – can estimate a model from higher number of points.

Additionally, a new approach of incorporating the *minimal* and *non-minimal* solvers works as follows. The *minimal* solver returns valid solutions as fast as possible, whereas accurate models are obtained by the *non-minimal* solver in local optimization, or in the final optimization, which is done only once. The RANSAC in its repetitive sample, estimate and evaluate manner should not contribute a lot of time to the estimation from a minimal sample as most hypotheses are bad. However, inaccuracies of *minimal* solver has to be compensated by the thorough final optimization.

4.1 Minimal solver

In RANSAC, an estimation from a minimal number of points in many tasks such as homography, epipolar geometry, linear perspective-n-point, *etc.* requires finding a null space of a matrix representing a system of linear equations. One of the most used methods is the singular value decomposition (SVD) that returns a set of orthonormal basis vectors for the null space of the input matrix. The SVD is a numerically stable procedure, which is often applied to solve many optimization problems such as least-squares minimization. However, in RANSAC running SVD in every iteration is quite inefficient since a lot of hypotheses can be arbitrary wrong. This section suggests to use Gaussian elimination (GE) of a matrix to find a set of null vectors. One can argue that GE is not accurate enough and in the presence of noise, it is going to affect the quality of the estimated models. That is true, and this is the reason why a non-minimal solver has to be introduced. Eventually, RANSAC estimates the final model from all inliers, hence even if the model found via GE is less accurate than from SVD, the final LSQ polishing makes either of them equally accurate. Experiments in section 9.2 support this statement.

4.1.1 Gaussian elimination

GE as a replacement of SVD can be used in estimation of homography, epipolar geometry, and other tasks that require to find a null space of a matrix. The advantages of GE over SVD are as follows:

1. It does not require any external libraries.
2. It can be used as detector of degenerate solution.
3. It is in order of magnitude faster.

First, the implementation of numerical methods such as SVD, eigen decomposition, or QR-factorization requires advanced knowledge of linear algebra, especially for understanding the milestones of these algorithms and making them stable. In C++, the most common libraries are OpenCV, Eigen, or LAPACK [10, 24, 1], while a clear advantage of GE is that its implementation is straightforward and short.

Second, since the code of GE can be easily written, it could be modified in a way to detect degenerate inputs during the computation of null vectors. For instance, the matrix representing linear equations for homography must have rank 8 if four correspondences that give two equations are in general position; however, if three points are collinear then the matrix is rank-deficient. During elimination of rows (columns), if a pivot happens to be zero, it becomes alarming that a matrix may not have full-rank, and the whole elimination process can be terminated. Using SVD one can also check the number of non-zero singular values which gives the rank of decomposed matrix, although since the code of SVD is not easily accessible, the earlier termination is not possible. It was also mentioned that GE is not numerically stable, but this issue arises from problematic points set. It implies that returning an imprecise solution will not influence RANSAC, because the sample is still bad. Moreover, it is another way to distinguish a bad hypothesis (*i.e.,* check if a null vector does not contain `NaN` or `Inf` values) and reject it.

Third, additional experiments show that by substituting GE instead of SVD, the speed-up of 4-point algorithm for $\mathbf{H}$ and 7-point method for $\mathbb{F}$ is around 10 times. Furthermore, GE occupies much less memory, a simple modification of the GE function can return only the desired number of null vectors, whereas SVD normally outputs three matrices.

An efficient estimation of homography using Gaussian elimination of a matrix representing a linear system was presented by Bazargani *et al.* in [8]. Authors observed that a coefficient matrix has special properties (*e.g.,* rows with many zeros) that are used to find a null vector very fast. The coefficient matrix of size 8×9 has one-dimensional null space which results in a homography that has 8 DoF after fixing the last element to 1.

For epipolar geometry one of the steps of estimation is to find null vectors and then apply them to further computations. Hartley *et al.* in the implementation of an essential matrix solver [27] extracted 4 null vectors accordingly to Nister's 5-point algorithm [48] using GE. Unlike homography estimation, the coefficient matrix is of size 5×9 and the next trick was used: the last four values for each null vector are fixed to 0 and 1, and other values then can be found using back-substitution. The four null vectors are as follows:

$$\begin{aligned}
\mathbf{e}_1 &= \begin{bmatrix} e_{11} & e_{12} & e_{13} & e_{14} & e_{15} & 1 & 0 & 0 & 0 \end{bmatrix}\\
\mathbf{e}_2 &= \begin{bmatrix} e_{21} & e_{22} & e_{23} & e_{24} & e_{25} & 0 & 1 & 0 & 0 \end{bmatrix}\\
\mathbf{e}_3 &= \begin{bmatrix} e_{31} & e_{32} & e_{33} & e_{34} & e_{35} & 0 & 0 & 1 & 0 \end{bmatrix}\\
\mathbf{e}_4 &= \begin{bmatrix} e_{41} & e_{42} & e_{43} & e_{44} & e_{45} & 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

A similar idea this thesis presents for fundamental matrix estimation from 7 points. The estimation consists of two main steps. First, constraint $\mathbf{x}'^\top \mathbf{F}\mathbf{x} = 0$ that each correspondence imply is used to build a linear system $\mathbf{A}\mathbf{f} = \mathbf{0}$, where $\mathbf{A}$ is the coefficient matrix of the system, and $\mathbf{f}$ contains the elements of $\mathbf{F}$ in vector form [28]. Coefficient matrix $\mathbf{A}$ is of size 7×9. Gaussian elimination is then used to make $\mathbf{A}$ an upper triangular matrix as follows:

$$\begin{pmatrix}
a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19}\\
0 & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29}\\
0 & 0 & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39}\\
0 & 0 & 0 & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49}\\
0 & 0 & 0 & 0 & a_{55} & a_{56} & a_{57} & a_{58} & a_{59}\\
0 & 0 & 0 & 0 & 0 & a_{66} & a_{67} & a_{68} & a_{69}\\
0 & 0 & 0 & 0 & 0 & 0 & a_{77} & a_{78} & a_{79}
\end{pmatrix}.$$

Since the fundamental matrix has 8 degrees-of-freedom, the two null vectors can have the last element fixed to one as $\mathbf{f}_9^{(1)} = \mathbf{f}_9^{(2)} = 1$. Let the first null vector has a fixed eighth element to zero $\mathbf{f}_8^{(1)} = 0$, thus, seventh element becomes $\mathbf{f}_7^{(1)} = -a_{79}/a_{77}$. Similarly, for the second null vector the seventh element can be fixed to zero $\mathbf{f}_7^{(2)} = 0$ and, thus, the eighth one is $\mathbf{f}_8^{(2)} = -a_{79}/a_{78}$. All other values of null vectors can be found by substituting the previously found elements:

$$\mathbf{f}_i^{(\{1,2\})} = \frac{-1}{a_{ii}} \sum_{j=i+1}^{9} \mathbf{f}_j^{(\{1,2\})} a_{ij} \qquad \forall i \in \{1, \dots, 6\} \tag{4.1}$$

Further, the two found null vectors are used to find coefficients of 3-degree polynomial accordingly to the 7-point fundamental matrix algorithm. Therefore, the speed-up concerns the part of finding two-dimensional null space, which is usually done by SVD.

4.1.2 Linear Perspective-n-Point

The benefits of GE could be applied in the estimation of perspective projection $\mathbf{P} \in \mathbb{R}^{3\times4}$ matrix. When the calibration is not known, the matrix can be found from 6 points using the direct linear transformation method. The projection matrix is given up-to-scale and hence has 11 degrees of freedom. Therefore, it requires 11 equations or 5 and ”half” points since each point gives two equations. One can apply SVD for least-squares minimization of all 12 equations or use Jackknife method [18]. As was already mentioned, RANSAC does not need necessarily to have a precise minimal model. This section proposes a method which exploits the structure of coefficient matrix representing a linear system of equations (similarly, as is done in [8]) to solve it via GE. Let $\mathbf{x} = [u\ v\ 1]^\top$ be an image point, and $X = [x\ y\ z\ 1]^\top$ is an object point. The projection matrix $\mathbf{P}$ has vectors $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ in rows. The up-to-scale relation of the image to the object point by the projection matrix is $\mathbf{x} \sim \mathbf{P}X$. Then a matrix of linear equations is as follows:

$$\begin{bmatrix} X^\top & \mathbf{0}_{1\times4} & -u\,X^\top \\ \mathbf{0}_{1\times4} & X^\top & -v\,X^\top \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = \mathbf{0} \tag{4.2}$$

The coefficient matrix above includes many zeros, and by replacing rows can be rewritten in a way to align all zeros in columns. The whole matrix then can be splittied into two matrices $\mathbf{A} \in \mathbb{R}^{5\times12}$ (the first five rows) and $\mathbf{B} \in \mathbb{R}^{6\times8}$ (the last six rows avoiding zero columns) as follows:

$$\begin{bmatrix} X_1^\top & \mathbf{0}_{1\times4} & -u_1\,X_1^\top \\ \vdots & \vdots & \vdots \\ X_5^\top & \mathbf{0}_{1\times4} & -u_5\,X_5^\top \\ \mathbf{0}_{1\times4} & X_1^\top & -v_1\,X_1^\top \\ \vdots & \vdots & \vdots \\ \mathbf{0}_{1\times4} & X_6^\top & -v_6\,X_6^\top \end{bmatrix} = \begin{bmatrix} & \mathbf{A} \\ \mathbf{0}_{6\times4} & \mathbf{B} \end{bmatrix} \tag{4.3}$$

At first, the matrix $\mathbf{A}$ is eliminated. The last row of the eliminated matrix $\tilde{\mathbf{A}}$ has 4 non-zero values. This row of $\tilde{\mathbf{A}}$ in the column range 4-12 is stacked to matrix $\mathbf{B}$, hence the new matrix $\mathbf{B}'$ has size 7×8. The matrix $\mathbf{B}'$ has one non-trivial null vector of size 8 in which the last element is fixed to 1. This null vector corresponds to the last 8 elements of the projection matrix $\mathbf{P}$. The other 4 values of $\mathbf{P}$ are then found by back-substitution of the null vector of $\mathbf{B}'$ to the eliminated matrix $\tilde{\mathbf{A}}$.

The elimination of a matrix has cubic complexity in the number of rows / columns. Therefore, elimination of a matrix of size 11×12 requires approximately 2-3 times more operations than elimination of two matrices of sizes 5×12 and 7×8. Moreover, it reduces memory allocation for matrices by around 10%. The speed-up of the proposed GE with respect to SVD version is around 6 times in the experiments. However, the disadvantage of this approach to SVD is that the latter one finds a solution in terms of least-squares for 6 points, while the projection matrix of GE perfectly fits only 5 points, and the sixth one may have a non-zero reprojection error in a presence of noise.

4.2 Non-minimal solver for epipolar geometry

In the minimal estimation, the idea is to generate models using cheap Gaussian elimination. However, for the non-minimal solver, which has to return accurate solutions, more precise methods should be used. For instance, homography and fundamental from non-minimal number of correspondences could be found by 4-point, 8-point algorithms respectively, in which the covariance matrix, constructed from coefficient matrix of data points, is used. While GE can still be used to find a solution, more accurate methods such as SVD should be employed.

The non-minimal optimization of a projection matrix can be similarly done using covariance matrix and direct linear transformation, however the presented framework suggests using it only in local optimization step, whereas for the final polishing to apply a much better direct least squares (DLS) method [30].

In this section will be discussed the non-minimal estimation of epipolar geometry, *i.e.,* fundamental and essential matrix. Experimentally, this work has found that the required constraints on the matrices (*e.g.,* must be 2-rank) destruct the optimization process. A couple of ideas on how to avoid this issue will be presented, whereas suggesting that the Gold standard method [26] should not be applied as its computations can overcome the whole RANSAC runtime.

4.2.1 Essential matrix

An efficient estimation of essential matrix from a non-minimal number of points is not an easy task, because linear optimization as in the case for homography or fundamental matrix are not suitable. Essential matrix must have rank two and the same singular values. If the same method for fundamental matrix is used for essential matrix, then to obtain a valid solution, it is necessary not only to enforce rank two, but also to equalize singular values. One can also try to estimate $\mathbf{F}$ and then convert it to an essential matrix using calibration. Although, if the fundamental matrix is consistent with a wrong pair of intrinsic matrices, then after conversion, the essential matrix will have different singular values. In experiment 9.8.1 are results of both methods. They sometimes generate better $\mathbf{E}$ than *so-far-the-best* but mostly do not improve the results or give even worse models.

Another approach to estimate $\mathbf{E}$ without enforcing any constraint is to use a minimal solver, where four null vectors are extracted from the covariance matrix of all inliers. It implies that a non-minimal solution suits all given correspondences, however, this approach does not help either.

This section proposes two options for estimating an essential matrix from non-minimal number of points. Both methods are based on the idea of partial (local) refinement of already known essential matrix (*e.g.,* from a minimal sample). Estimating both translation and rotation ($\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$) is not suitable for linear optimization due to additional constrains of rotation matrix. Moreover, it could be computationally expensive with non-linear methods. By assuming that either rotation or translation component is good, the goal is to improve the other one. Eventually, the methods can be combined, *i.e.,* firstly update rotation and then translation.

The first method is to fix the rotation matrix and refine translation. The objective function is the minimization of algebraic error. Let $\mathbf{y} = \mathbf{K}_1^{-1}\mathbf{x}$ and $\mathbf{y}' = \mathbf{K}_2^{-1}\mathbf{x}' = [y'_1, y'_2, 1]^\top$ be normalized image correspondences. Then epipolar constraint for $\mathbf{E}$ is

$$\mathbf{y}'^\top [\mathbf{t}]_\times \mathbf{R}\mathbf{y} = \mathbf{0} \tag{4.4}$$

Since, $\mathbf{R}$ is fixed then after substitution of $\mathbf{z} = \mathbf{R}\mathbf{y} = [z_1, z_2, z_3]^\top$ the translation $\mathbf{t}$ can be estimated with simple linear optimization (*e.g.,* PCA). The coefficient matrix is of size $n \times 3$, where $n \geq 3$ is the number of correspondences, and the covariance matrix is of size 3×3. In the matrix form, the problem is as follows:

$$\begin{bmatrix} z_2 - z_3 y'_2 & z_3 y'_1 - z_1 & z_1 y'_2 - z_2 y'_1 \\ \vdots & \vdots & \vdots \end{bmatrix} \mathbf{t} = \mathbf{0}$$

An initial $\mathbf{R}$ is obtained after decomposition of essential matrix. Since two rotations are possible, then two corresponding translation vectors can be estimated or just one for randomly chosen rotation matrix. The final essential matrix is composed back via the new translation and its corresponding rotation.

The second method is to fix translation obtained after decomposition (a scale of $\mathbf{t}$ is negligible) and refine rotation $\mathbf{R} \in \mathrm{SO}(3)$. The objective is again the minimization of algebraic error, although a new substitution is $\mathbf{q}^\top = \mathbf{y}'^\top [\mathbf{t}]_\times$. A similar approach as in the directed least squares method for PnP [30] is used. In spite of in PnP estimation, the cost function is different, however, the problem definition seeks to find rotation and translation. Hesch *et al.* in [30] uses several ideas to simplify the estimation. Authors employed Cayley-Gibbs-Rodriguez (CGR) parametrization [7] of rotation matrix to express it using three-dimensional vector. This step enables to have unconstrained optimization problem. Moreover, in [30] rotation appears linearly in the cost function as well as in the epipolar constraint $\mathbf{q}^\top \mathbf{R}\mathbf{y}$. Substituting the coefficient matrix of PnP problem by the coefficient matrix of the epipolar constraint, the method originates a valid rotation matrix. The maximum number of possible solutions is 27, however, in experiment 9.8.1 the average number is 3.6.

4.2.2 Singular values

The experiment in 9.8.1 reports that both methods presented in the previous subsection provide more accurate essential matrices than the ones estimated via linear 8-point algorithm. However, the pose error and geometric accuracy are even better if the singular values are not corrected in the 8-point method, *i.e.,* the third value is remained non-zero, and the first two remain non-equal. It suggests that the proposed methods can be helpful if the user needs precise essential matrices.

In practice, although, essential matrix estimation is widely used to obtain a relative pose. In this case, the precision of singular values of $\mathbf{E}$ does not matter. The reason is that the decomposition of essential matrix into relative pose applies SVD of $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$, where only matrices $\mathbf{U}$ and $\mathbf{V}$ are used. Ideally, is to find a function which maps a noisy matrix of singular values $\mathbf{D}$ to matrix $\mathbf{D}'$ such that it has equal first two singular values and the third one is zero, additionally, a RANSAC score of $\mathbf{E}'$ from $\mathbf{D}'$ is about the same as score of $\mathbf{E}$. However, this work does not intend to investigate the function as it seems to be sophisticated, whether it even exists.

Lifting rank constraint can be also used in finding a fundamental matrix from a non-minimal sample, where the 8-point algorithm must enforce 2-rank constraint by zeroing the third singular value. A precise $\mathbf{F}$ usually has fewer inliers than its noisy 3-rank version. This issue has to be addressed since models with the high inlier number lead to the earlier termination of RANSAC. Nevertheless, the goal remains to output an accurate 2-rank $\mathbf{F}$. This section suggests to lift rank constraint in the local optimization to save time by avoiding SVD and finding models with high support number. Therefore, 2-rank $\mathbf{F}$ is found in the final optimization, where 8-point algorithm zeroes the third singular value.

In summary, the constraints of singular values can be lifted for fundamental matrix in the local optimization, and for essential matrix if it is evaluated on the ground truth pose. The results showing the gained speed-up of the proposal approach is in the experiment 9.8.2.

5 So-far-the-best model

In RANSAC, processing a *so-far-the-best* model is crucial for a validity of the found hypothesis, and further performance of RANSAC. This step is responsible not only for the model acceptance, but also for a degeneracy check, the local optimization (LO) of the model, update of inlier ratio for the pre-emptive verification, and termination of the whole algorithm.

This chapter is going to discuss a role of LO method under the concept of the *minimal* and *non-minimal* solvers. Additionally, it addresses LO issues related to its time complexity and the likelihood of returning a degenerate model. Further, by incorporating a model non-randomness will be proposed a new adaptive inlier threshold calculation, and quasi-random point selection in the minimal sampling. Finally, a way of handling occurrence of multiple models structures is presented.

5.1 Local Optimization

In LO-RANSAC [14] of Chum *et al.* is observed that all-inlier samples in the presence of noise generate inaccurate models. The authors in [14] propose to apply a non-minimal model estimation using inliers of *so-far-the-best* hypothesis. Since the least squares minimization is done on the points consistent with model, it produces a better solution that fits more correspondences than the one from a minimal sample. GC-RANSAC [4] of Barath *et al.* further extends LO method by exploiting spatial coherence of points to apply graph-cut minimization problem. Later, in MAGSAC++ [5] was presented iterative re-weighted least-squares local optimization that appears to be more accurate than its predecessors.

The importance of LO method is not only improving the model quality, but in addition, it leads RANSAC to terminate earlier by finding a model with a significant number of inliers. However, in practice, LO could take a significant amount of time, overcoming the runtime of the whole RANSAC. While a model obtained from LO still undergoes the LSQ polishing.

This section proposes to employ the idea described in chapter 4 that stands for fast estimation of minimal models inside RANSAC, whereas the accurate solution is found in the final optimization only once. Therefore, a role of the desired LO is to return a model with the highest support so far to trigger RANSAC termination, and simultaneously to avoid time-consuming computations.

In LO-RANSAC [14] is found that RANSAC updates *so-far-the-best* model $\log(K)$ times (where K is the number of iterations). Experimentally, within 100 first iterations RANSAC saves the model around 3-4 times ($\log_e 100 \approx 4.6$), while for PROSAC where the first samples are likely to originate good models, this number is even higher. It implies that LO can spend a lot of time by optimizing hypotheses that will be rejected in RANSAC later within a small margin of iterations. Moreover, the non-randomness verification should be applied as well to avoid improving bad models. Additionally, models with a similar number of inliers as *so-far-the-best* one are likely to represent the same structure, thus do not need to undergo LO step.

In the proposed formulation, the primary objective is to find a light-weight LO procedure that runs swiftly and is applied only when it likely leads to termination. To do so, the following conditions that control when the LO is applied are introduced:

1. The required number of iterations needed to find adaptive SPRT setting and a support size of non-random model is reached. Additionally, it helps to avoid unnecessary first LO runs.

2. The *so-far-the-best* model has the required number of independent inliers defined by (2.2).
3. The local optimization is applied only if the inlier sets of the new best and previous best models has a lower than 0.95 Jaccard index, *i.e.,* the intersection over union. This condition is motivated by the tendency that if a model is just slightly different from the previous *so-far-the-best*, the LO step likely does not refine it significantly, but the final optimization does the main improvement.

```
Algorithm 2: Local Optimization, LO$^+$
   Input: $\mathcal{I}^*$ – inliers of so-far-the-best model,
   $\tau^*$ – its score, $s$ – non-minimal sample size,
   $K_{\max}$ – maximum iterations of LO.
1  $\theta^*_{\text{LO}} := \varnothing$;  $I_{\max} := |\mathcal{I}^*|$
2  for $t := 0$;  $t < K_{\max}$;  $t$++ do
3      $\theta_{\text{LO}} :=$ estimate (subset($\mathcal{I}^*$, $\min\{|\mathcal{I}^*|, s\}$))
4      $\tau_{\text{LO}}, \mathcal{I}_{\text{LO}} :=$ evaluate($\theta_{LO}$)
5      if $\tau^* \prec \tau_{\text{LO}}$ then
6          $\mathcal{I}^*, \tau^*, \theta^*_{\text{LO}} := \mathcal{I}_{\text{LO}}, \tau_{\text{LO}}, \theta_{\text{LO}}$
7      if $I_{\max} < |\mathcal{I}_{\text{LO}}|$ then
8          $I_{\max} := |\mathcal{I}_{\text{LO}}|$
9          if $K_{\text{RANSAC}} >$ MaxIters($I_{\max}$) then
10             break
11 return: $\theta^*_{\text{LO}}$, $\tau^*$, MaxIters($I_{\max}$)
```

The proposed LO$^+$ method based on the LO-RANSAC is shown in alg. 2. It is important to note that in this procedure, larger-than-minimal samples are selected that is typically avoided in RANSAC due to increasing the problem complexity and, thus, the number of iterations required to provide probabilistic guaranties of finding the sought model parameters. In alg. 2, the sample is selected from a set of points that likely are inliers. Therefore, the increased sample size does not affect the accuracy and processing time negatively.

At line 3 of alg. 3 the function *subset* returns an inlier subset of size either s or the number of inliers. The maximum inlier number of the estimated models is saved at line 8. Additionally, the LO$^+$ can terminate if new upper bound of iterations is lower than number K_{RANSAC} of iterations done by RANSAC, in line 10.

The sample size s and the maximum number of LO iterations $K_{\max}$ are found empirically to suit two-view geometric problems, minimizing the total run-time while maintaining the accuracy. For **F** / **E** estimation, the optimal sample size is 35 and number of iterations is 15, while for **H**, the sample size is 40 and number of iterations is 10. For **P**n**P** the sample size is 30, and the number of iterations is 4.

5.1.1 Termination

As was already mentioned, the local optimization aims to accelerate the termination of RANSAC by finding the model having the highest number of inliers so far. Simultaneously, when the model with sufficient inlier number is found, then LO itself should terminate. It is unnecessary to continue LO otherwise, because the final optimization takes care of polishing *so-far-the-best* model.

Moreover, one LO iteration is much slower than RANSAC iterations, *i.e.,* non-minimal model estimation is around 10 times slower as it uses SVD. By default, LO terminates if the maximum number of inner LO iterations reached. However, under VSAC framework LO can stop immediately when the termination criteria (*e.g.,* the standard upper bound, PROSAC) are satisfied. Despite the termination for minimal and non-minimal sampling is mixed, the locally optimized model that triggers criteria can be seemed as a noise-free version of its minimal model. Additionally, since the LO model is originated from inliers, it likely represents the same structure.

5.1.2 Degeneracy

The model that undergoes LO is already non-degenerate as it went through degeneracy check (*e.g.,* DEGENSAC^{+}). However, there is no guarantee that non-minimal estimation generates always valid models. While for homography it is fairly unlikely to draw a whole non-minimal sample consisting of points on a line, for epipolar geometry estimation the case when all correspondences are planar is possible. As it was suggested, the subset size for $\mathbf{F}$ should be around 35 points, hence the probability of fully planar sample is $\left(\frac{|\mathcal{I}_{\mathbf{H}}|}{|\mathcal{I}|}\right)^{35}$ if points are drawn uniformly. For instance, for 96% ratio of inliers consistent with homography, with the confidence 99% is drawn at least one non-minimal sample of 35 planar points within 15 iterations. Therefore, the odds become non-negligible when a field of view is a dominant plane, *e.g.,* a building facade.

There are several ways to avoid degenerate solutions for $\mathbf{F}$. First, using a homography matrix, a set of points likely to be planar can be separated from inliers. Every time a non-minimal sample is drawn, it must contain a few points out of plane. This test is extremely fast, because a Boolean mask of planar correspondences is already computed in DEGENSAC^{+}, hence the verification of a small amount of sample points is negligible. Although, the disadvantage of this approach stems from inaccurate classification of planar points based only on the reprojection distance and threshold. Consequently, good samples could be rejected if some *true* non-planar matches are close to homography manifold. Moreover, outliers by chance consistent with epipolar geometry in combination with planar points can lead to a degenerate solution. If estimation is done only on non-planar inliers then estimated $\mathbf{F}$ may not be accurate enough. The second approach is to do the same non-degeneracy verification as was discussed in DEGENSAC^{+}. A new fundamental matrix must have a sufficient number of independent inliers out of plane.

Both presented methods require homography (planar matches) to work with. However, even if the scene contains planar structures, $\mathbf{H}$ must not be necessarily found in DEGENSAC^{+}, *i.e.,* 7-point sample of *so-far-the-best* $\mathbf{F}$ consists of correspondences off a plane. It implies, that the LO should not be used in this case. Moreover, the whole problem with degeneracy becomes even more complicated if several planes are present in the scene. In this case, for instance, the second proposal must verify that $\mathbf{F}$ has independent inliers with respect to each plane, or if inliers of at least two planes are points consistent with $\mathbf{F}$ as well.

In section 6.2 will be proposed an approach that allows to run LO without degeneracy tests, nonetheless, the final RANSAC model subjects to verification. This is done by efficient estimation of homography via $\mathbf{F}$ inliers, and its further recovery if degenerate.

5.2 Adaptive inner threshold

Finding an optimal inlier-outlier threshold for RANSAC is not an easy task since many factors as image dimensions or noise can influence a quality of an estimated model, thus how it fits data points. This problem was addressed primarily in MAGSAC [6] which does marginalization of point errors over the noise level, or in a-contrario RANSAC [19] that selects the best model by a probabilistic approach avoiding threshold. However, in many cases, a set of inliers is required to work with, for instance, in local optimization or obtaining the upper bound of iterations.

By setting the inlier threshold value too high, it can lead to earlier termination, although usually a less accurate final model. Whereas a low threshold, first, does not always provide a good solution, and second, heavily influence performance of RANSAC (*i.e.,* primarily speed). The low threshold's value implies a low number of points consistent with a model that makes inlier based methods less efficient. For instance, the local optimization draws subsets of inliers to estimate model from a non-minimal number of points, hence the inlier set should be suffi-

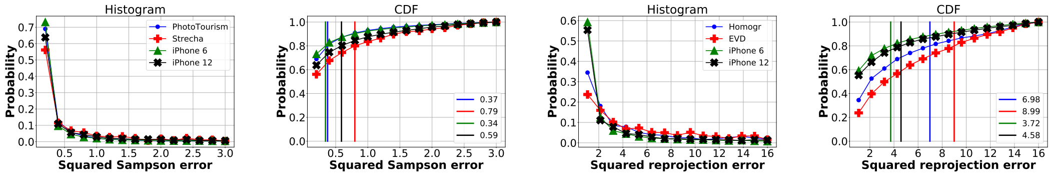

Figure 5.1 The first and third plots are probability density histograms of consistent points to the GT fundamental matrix (resp. homography). The x-axis shows square error distance of point to model, *i.e.,* Sampson error for $\mathbf{F}$ and reprojection distance for $\mathbf{H}$. In the legends of these figures are names of dataset from where image pairs selected. Histograms are averaged over image pairs within a dataset. The second and fourth plots are corresponding cumulative probability histograms. Their legends show the square threshold (error distance) of 80th percentile of histogram distributions.

ciently large for better sampling. In DEGENSAC$^+$, a threshold for separation of planar points is needed.

This section proposes to find so-called ”inner” threshold adaptively to solve the issue with a user-given value. The ”inner” threshold does not necessarily replace the input one, but mainly used for inner methods such as LO or DEGENSAC$^+$. The approach requires a model which is *so-far-the-best* and non-random. Next step is to compute a cumulative histogram of points' squared errors with respect to the model parameter in the error range from 0 to the maximum threshold (*e.g.,* 10 pixels). Then the desired square inner threshold should be, *e.g.,* 80th percentile of the cumulative histogram. It implies that the found threshold covers 80% of inliers in the given error range.

This method stems from the fact that inlier error distances follow a distribution (*e.g.,* Gaussian or chi-square), while the error distribution of outliers in most cases is uniform. Therefore, assuming a non-random model, it leads to expect that the density histogram (and CDF histogram) of point errors consistent with this model is likely to be similar to the true one.

Figure 5.1 shows error histograms of points with respect to the ground truth model. Datasets of different average image sizes are considered. It demonstrates that each dataset requires a slightly different threshold to cover at least 80% of points within the maximum error range.

5.3 Importance of multiple structures

It is generic for RANSAC to reject all model structures that have a worse score than the *so-far-the-best* one. However, some discarded models could be more suitable for users. It stems from the fact that in practice there may exist multiple structures in the scene, but only one of them is a desired one. Therefore, unless some prior information for RANSAC is given, it is difficult to detect the correct model. Moreover, considering RANSAC ”higher score wins” manner, an incorrect structure with the highest number of inliers among all will be outputted, which inevitably leads to a failure from the user perspective.

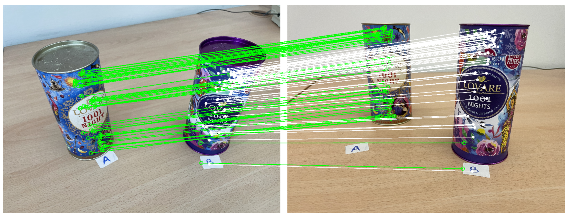

Figure 5.2 An image pair with two dominant objects. Green and white correspondences connected with lines are inliers of two fundamental matrices. The object ”A” (cylinder in the left) was deliberately moved to create two possible structures for epipolar geometry estimation, while the correct $\mathbf{F}$ is the one having white matches.

For instance, let the task be to find a homography of a planar building facade, although there is no guarantee that it is the only planar object in the scene. The similar problem can arise in epipolar geometry, where the found structure can represent an incorrect relative pose. In figure 5.2 is shown a scene with static and dynamic objects. The correct $\mathbf{F}/\mathbf{E}$ must represent a camera relative pose with respect to static objects, while dynamic objects with a number of good correspondences can destruct the estimation. In practice, it is often the case when two images are taken at different times, hence some objects in the scene may have changed their position.

The rules for applying LO in section 5.1 mentions Jaccard index of inlier sets to measure the similarity of models. Models with high intersection over the union (IoU) of inlier sets must not undergo LO. However, previous *so-far-the-best* models with low IoU (*e.g.,* 1-5%) and equally high inlier number should be additionally saved, as they may represent correct but smaller structures. The presented framework is not meant to be a multi-model estimator, although there is already a computation of Jaccard index before running the local optimization. Therefore, it makes nothing but a small modification to save and output all *structures of interest*, so the user can decide which one fits better.

5.4 Quasi-random sampling

Another improvement as finding a better model earlier can be achieved in RANSAC with uniform sampling. In the proposed framework the sampling is implemented using Fisher-Yates pseudo-random shuffle [45]. A sampler draws randomly at uniform without repetition m (sample size) numbers from array filled from 1 to N (number of points).

In the *so-far-the-best* step, when a model with significant number of *independent* inliers is found, this section suggests to order numbers in the array in a way that in the beginning there are I inliers' indices and the rest $(N - I)$ are indices of outliers. The indices of inliers and outliers are randomly shuffled. Then the quasi-random sampling draws m numbers from array in chunks. Consequently, next $\lfloor \frac{N}{m} \rfloor$ iterations in RANSAC are performed by quasi-random sampling, afterwards Fisher-Yates shuffle is back. To justify this approach, consider two cases.

First, assume the *so-far-the-best* model is *bad* thus its inliers are not consistent with the best possible model. In this case, the order of numbers in the array does not matter, and since the numbers are shuffled they are selected again uniformly at random. The only difference to the pseudo-random sampling is that all numbers in the array are drawn exactly one time. This distinction is not necessarily bad, on the contrary, a pseudo-random sampler might not draw some good points (numbers) with a non-zero probability.

Second, let the *so-far-the-best* model is good, hence its support are correct points. Therefore, since quasi-random sampler starts to draw points from the ordered array, then the first $\lfloor \frac{I}{m} \rfloor$ samples should be promising as they consist of inliers. This leads to finding a better model faster with earlier RANSAC termination. It could be seemed as LO with minimal sample size integrated in RANSAC sampling, which does not avoid outliers and has almost no time cost. The experiments of proposed quasi-random method is in experiment 9.7.

6 Final Optimization

The final optimization plays an important role in the presented VSAC estimator. As was mentioned earlier, a new framework stands by the idea that RANSAC should finish as fast as possible, providing a valid model. While an accurate solution is found in the final optimization (FO), which is run only once. For example, the minimal solver via Gaussian elimination and the new construction of local optimization support this idea. In the textbook RANSAC, it was suggested to run the least squares minimization (LSQ) on a set of all inliers in the end. Here, will be presented a few approaches for FO as one least squares run is not enough to compensate the deterioration of RANSAC minimal model accuracy. In addition, a couple of suggestions related to epipolar geometry will be proposed.

6.1 Outlier removal

For the final least squares optimization, all inliers – consistent points with the *so-far-the-best* model are used. However, in epipolar geometry estimation, there could be a small percentage of correspondences that are consistent with a good model structure, however, they are totally incorrect. For instance, matches that do not pass chirality constraint or near epipoles. Sampson error distances of correct correspondences follow the normal distribution as is assumed in, *e.g.,* MLESAC [57], which agrees with Gaussian noise of an image detector. Whereas, errors of mismatched points are not Gaussian. The number of points by chance consistent with $\mathbf{F}$ is very low, however, even a few of them can slightly destruct the least-squares solver. The experiment in section 9.8.3 shows that removing these outliers marginally improves accuracy of the final LSQ method. A few ways to filter out bad points are described further in details in section 7.1.

6.2 Homography detection

In section 5.1.2 was discussed that local optimization of fundamental matrix is prone to return degenerate solutions in the presence of the dominant plane. Several options were proposed to avoid this issue. However, it makes LO method more aggressive to reject good hypotheses. Moreover, these tests require a homography to work with. Therefore, if $\mathbf{H}$ is not yet found, or it is not present in the scene, then it implies that LO should not be run at all. The situation gets more complicated when a scene contains multiple planes, because $\mathbf{F}$ must have a non-planar support regarding every plane.

This section presents an approach that guarantees with 99.99% of confidence for RANSAC to output a non-degenerate fundamental matrix yet making no changes in the LO method. The idea is that locally optimized models are likely to be good structures with many points consistent with them. Even if a *so-far-the-best* model of LO is degenerate, the corresponding support is a lower bound of its non-degenerate version. The reason is that the latter model not only has around the same number of correspondences on a plane, but additionally non-planar support that makes it a non-degenerate. The outline of the procedure that aims to efficiently distinguish a degenerate $\mathbf{F}$ and recover if needed is as follows:

1. If *so-far-the-best* $\hat{\mathbf{F}}^*$ is from LO:
2. Try to estimate a homography using $\hat{\mathbf{F}}^*$ and 3-point $\mathbf{H}$ RANSAC of $\hat{\mathbf{F}}^*$ inliers.

a) If $\mathbf{H}$ has not found then $\hat{\mathbf{F}}^*$is not degenerate.
b) Otherwise, compute a non-planar independent support of $\hat{\mathbf{F}}^*$.

3. If the support is fewer than a sufficient one (already estimated in DEGENSAC$^+$) than try either to recover fundamental matrix via $\mathbf{H}$ and calibration, or use plane-and-parallax RANSAC.

a) Accept the recovered $\hat{\mathbf{F}}$ if its score is better than of the last non-degenerate *so-far-the-best* model.
b) Otherwise, select the latter one.

The step (2) assumes that $\hat{\mathbf{F}}^*$ is degenerate, hence its support is planar matches and some randomly consistent outliers. Therefore, an efficient 3-point RANSAC draws samples from inliers of *so-far-the-best* fundamental matrix. If the assumption is correct, *i.e.,* support of $\hat{\mathbf{F}}^*$ is dominated by planar inliers, then RANSAC must terminate within a few iterations. For instance, using confidence level 99.99%, and predicting at least 95% of $\mathbf{H}$ inliers the upper bound for 3-point RANSAC is only 5 iterations. If the assumption is wrong, thus $\hat{\mathbf{F}}^*$ is non-degenerate, then 3-point RANSAC will likely fail to find a good homography within 5 samples.

Step (3) of recovery is actually not necessary, because a non-degenerate model is known. Although, since $\hat{\mathbf{F}}^*$ is assumed to be a good structure (despite being degenerate) it is worth to make an attempt to recover it. In any case, the fundamental matrix undergoes the final optimization after this test.

6.3 Iterative least-squares

All-inlier least-squares optimization could significantly improve the quality of the final model. In the experiment in section 9.6 was observed that several iterative all-inlier LSQ runs can generate even more accurate solution. Each model after LSQ iteration becomes better and, thus, has larger support that is used for the next LSQ estimation until it converges. The iterative LSQ approach compensates the accuracy trade-off that is done for a minimal estimation and local optimization.

The main issue to be addressed in this section is on how to make the iterative LSQ fast. For homography and epipolar geometry estimation, the non-minimal estimation described in [28] is as follows. (1) Compute transformations 3×3 matrices $\mathbf{T}$ and $\mathbf{T}'$; normalize sample points such that their centroid is at the origin, and the average distance of points from the origin is $\sqrt{2}$; (2) compute a covariance matrix from a coefficient matrix of linear equations (either 4-point method for $\mathbf{H}$ or 8-point for $\mathbf{F}$); (3) find the solution in terms of least-squares by applying SVD or QR factorization on the covariance matrix; (4) transform the model back via $\mathbf{T}$ and $\mathbf{T}'$. Note, this method is algebraic error minimization.

The LSQ description above is expensive, even for a few runs. The most computationally exhaustive part is normalization and calculating the covariance matrix of inliers, as it linearly depends on a number of used points; while time for decomposition of covariance matrix is almost constant. Therefore, a new method LSQ with covariance matrix resolves both problems. First, the normalization is done only once for all points; the presence of incorrect correspondences does not destruct the transformation matrices $\mathbf{T}$ and $\mathbf{T}'$. Second, the method assumes that most of the inliers after each LSQ iteration remain the same, while some new ones appear. Therefore, the covariance matrix is also built only once, and in the new LSQ iteration, it is updated by adding new points or subtracting ones that become outliers. The number of LSQ iterations is set by user, however, the method terminates when Jaccard similarity of inlier sets of the previous and new model is higher than *e.g.,* 95%, because there is no reason to apply LSQ on the same inlier sets. The complexity of this procedure is one normalization of all points, one construction of covariance matrix, and k (*i.e.,* number of LSQ runs) covariance matrix decomposition. The only disadvantage is a slight overhead for normalization if most of the input correspondences are outliers.

A more accurate way of the final model polishing is to apply the iterative reweighted LSQ of MAGSAC++ weights. The idea of a single covariance matrix should not be used in this case, because weights of inliers are updated in each iteration. In the experiments 9.6 this is so far the most accurate LSQ approach.

7 RANSAC output

The RANSAC algorithm usually outputs only one model with the highest found support and corresponding tentative inliers. In some applications, *e.g.,* in 3D reconstruction or bundle adjustment, mainly inliers are used. However, the quality ranking of inliers, which could have been helpful in practice, is not returned. For instance, inliers could be sorted by their residuals to *so-far-the-best* model. This chapter presents a few ways to discard incorrect inliers consistent with the final epipolar geometry. Additionally, it suggests to ”correct” good points that could be beneficial for reducing geometric error on the ground points.

7.1 Detection of outliers

Inliers of epipolar geometry are points lying on epipolar lines, in RANSAC a point is inlier if, *e.g.,* its Sampson distance [25] falls below a user-defined threshold. However, if points are inliers, they must not necessarily be true correspondences. Tentative correspondences obtained from a key-point detector and undergone through a matching procedure, may by chance lie on epipolar lines of the true fundamental matrix, even if they are incorrect. If an image pair contains a lot of repetitive patterns (*e.g.,* windows on a building facade), then the matching process of *e.g.,* SIFT descriptors can match two points of different locations in the real scene.

The ratio of incorrect inliers by chance consistent with the true fundamental matrix can be found by considering image dimension and inlier-outlier threshold, *i.e.,* fraction of strip around epipolar line to the area of possible outlier appearance [43]. Inliers of epipolar geometry are points lying on epipolar lines, in RANSAC a point is inlier if, *e.g.,* its Sampson distance falls below a user-defined threshold.

7.1.1 Uncalibrated case

When intrinsic matrices are not given, this work proposes to detect incorrect inliers using epipolar orientation (chirality) constraint [15] of Chum *et al.* The test for a correspondence pair $(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^3$ is as follows:

$$\mathbf{e}' \times \mathbf{x}' \overset{+}{\sim} \mathbf{F}\mathbf{x} \tag{7.1}$$

The valid epipolar geometry always satisfies this constraint for true correspondences. Incorrect points by chance consistent with the true epipolar geometry can be easily distinguished if they validate the constraint. Note, this check is also used in RANSAC for verification of $\mathbf{F}$ and $\mathbf{E}$ using its minimal sample. Therefore, all models and their samples are correct, however, other inliers may not pass the orientation test.

Another robust and cheap method to detect outliers is to use symmetric geometric distance (SGD) instead of Sampson error, *i.e.,* calculate the sum of distances of image points $\mathbf{x}$ and $\mathbf{x}'$ to its corresponding epipolar lines $\mathbf{F}^\top\mathbf{x}'$ and $\mathbf{F}\mathbf{x}$. The Sampson distance is an accurate approximation of geometric error, *i.e.,* it is Euclidean distance in 4D space from the given correspondence to the one which perfectly lies on the epipolar lines. Sampson approximation is commonly used for evaluation of epipolar geometry, however, additional experiments show that it is not robust to incorrect points that lie near epipole. This problem can be discussed in two situations.

In the first case, if an image point is close to epipole it generates no epipolar line on the second image. Consequently, epipolar constraint holds even if this correspondence is not correct. Sampson approximation returns zero 0 error distance, while SGD error does not. For example, let a point $\mathbf{x}$ is located near epipole $\mathbf{e}$ and a corresponding point $\mathbf{x}'$ is not, then the generated epipolar line is $\mathbf{F}\mathbf{x} \approx \mathbf{F}\mathbf{e} = \mathbf{0}$. The situation which happens with both error functions is as follows:

$$\frac{\cancelto{0}{|\mathbf{x}'^\top \mathbf{F}\mathbf{x}|}}{\sqrt{\cancelto{0}{[\mathbf{F}\mathbf{x}]_1^2} + \cancelto{0}{[\mathbf{F}\mathbf{x}]_2^2} + [\mathbf{F}^\top \mathbf{x}']_1^2 + [\mathbf{F}^\top \mathbf{x}']_2^2}} = 0 \tag{7.2}$$

$$\frac{\cancelto{0}{|\mathbf{x}'^\top \mathbf{F}\mathbf{x}|}}{\sqrt{\cancelto{0}{[\mathbf{F}\mathbf{x}]_1^2} + \cancelto{0}{[\mathbf{F}\mathbf{x}]_2^2}}} + \frac{\cancelto{0}{|\mathbf{x}'^\top \mathbf{F}\mathbf{x}|}}{\sqrt{[\mathbf{F}^\top \mathbf{x}']_1^2 + [\mathbf{F}^\top \mathbf{x}']_2^2}} = \texttt{NaN} \tag{7.3}$$

The subscript $[\mathbf{v}]_i$ denotes the i-th element of a vector v. In practice, although, since points usually are not precisely near epipole and due to numerical issues, the SGD error is not NaN but far from 0, which helps to recognize such correspondences.

The second case is slightly more complicated and relates to the spatial location of correspondences. Let $\mathbf{x} \leftrightarrow \mathbf{x}'$ be a correspondence with Sampson error close to 0, while SGD is more than tens of pixels, and neither of the points is near epipole. It means that there exists a correspondence $\mathbf{x}^* \leftrightarrow \mathbf{x}'^*$ which perfectly lies on epipolar lines and moreover is close to $\mathbf{x} \leftrightarrow \mathbf{x}'$. In the experiments, the following scenario occurred, $\mathbf{x}^*$ is just slightly distant from $\mathbf{x}$ (similarly $\mathbf{x}'^*$ is close to $\mathbf{x}'$), however, an angle between epipolar lines $\mathbf{F}\mathbf{x}^*$ and $\mathbf{F}\mathbf{x}$ is around 15-20 degrees. As a result, $\mathbf{x}'$ is far away from $\mathbf{F}\mathbf{x}$ which explains the high SGD error.

Note that this section does not recommend to use SGD instead of Sampson error in RANSAC, whereas suggesting using it only for outlier removal. Incorporating Sampson distnce in RANSAC provides better results in the experiments. Additionally, the inlier-outlier threshold better agrees with Sampson approximation of a correspondence lying precisely on epipolar lines. In summary, the proposed approaches for detection of incorrect points consistent with *so-far-the-best* model can remove up to 0.5% of bad inliers, that are usually a couple of matches. However, experiments in section 9.8.3 show that removing even a few bad inliers can improve estimation results.

7.1.2 Calibrated case

An epipolar geometry represented by fundamental (essential) matrix together with intrinsic matrices are commonly used to filter incorrect correspondences by chirality constraint. Although, it does not guarantee that all wrong points are removed, because there are still small amounts of 3D points having positive depths, but theirs image projection correspondences are totally wrong despite satisfying an epipolar constraint. The true depth of 3D point remains unknown since the decomposition of essential matrix provides only the up-to-scale translation, however, for a method presented here, it is enough to detect bad points.

Experimentally, this work has noticed that depths of some reconstructed points could be significantly lower or higher than others. After a manual verification of their image correspondences on a number of image pairs, it was found that most of these matches are incorrect. Therefore, this section proposes to sort depths (*e.g.,* in increasing order) and cut 1% of points in both tails. It is very likely that invalid points will be cut out. However, depths of 3D points are scene-dependent, thus the suggested procedure could remove points corresponding to close or far objects. Nevertheless, a single outlier can harm much more than a correct point, consequently, even if some inliers are falsely removed it will not be as bad as if some bad points are used in the next post-RANSAC computations.

The new DEGENSAC$^+$ includes adaptive method for estimation of suitable intrinsic matrices. Experiments show that even if calibration matrices slightly differ from the given ones, they still can be used to remove incorrect correspondences with chirality constraint. The triangulated 3D points of approximated calibration are different, but wrong points have negative depths.

7.2 Correction of points

Methods for point corrections could be used for several reasons. First, it improves precision of the ground points obtained either by manual annotation or from keypoint detector followed by matching selection. In either case, the presence of noise in correspondences is almost inevitable. Even if annotation went well and points supposed to be noise-free, the discrete nature of photography (*i.e.,* the scene is projected to a grid of pixels) prevents having perfect points with no noise. Therefore, the correction of the ground truth (annotated) points can refine noisy points.

Additionally, the correction can make inlier points error-free with respect to the final model. It is convenient for users who do not want to run bundle adjustment. The VSAC framework can output corrected points for homography and epipolar geometry estimation problems.

7.2.1 Homography

A way to ”correct” points on both images is by introducing ”half” homography which transforms points from two images in the middle, average them and transforms them back. The ”half” homography $\mathbf{A}$ is a square root of homography matrix such that $\mathbf{H} = \mathbf{A}\mathbf{A}$. The existence of a matrix square root requires a positive real part of eigen values as is described in [17]. The proof that planar $\mathbf{H}$ satisfies this condition is not a part of this paper, however, in practice homographies fulfil it. In the figure 7.1 is visualization of a ”half” homography transformation. The middle point is defined as follows:

$$\mathbf{m} = \frac{\phi(\mathbf{A}\mathbf{x}) + \phi(\mathbf{A}^{-1}\mathbf{x}')}{2} \tag{7.4}$$

Where $\phi : \mathbb{R}^3 \to \mathbb{R}^3$ is a mapping which normalizes homogeneous point by z-coordinate. Then ”corrected” points are the following:

$$\tilde{\mathbf{x}} = \phi(\mathbf{A}^{-1}\mathbf{m}) \quad \text{and} \quad \tilde{\mathbf{x}}' = \phi(\mathbf{A}\mathbf{m}) \tag{7.5}$$

The proof showing that a ”corrected” correspondence has zero reprojection distance to homography matrix is in appendix 11.2.

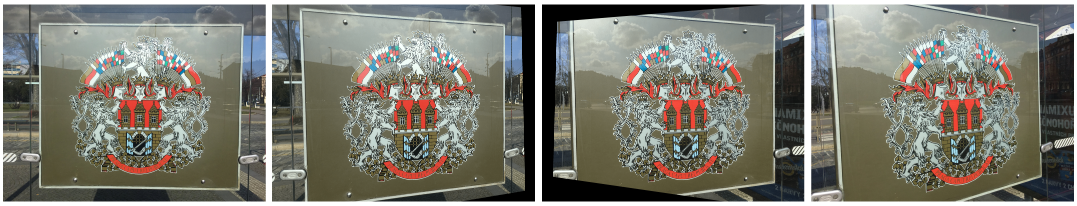

Figure 7.1 The left most image is a reference, the right most is a destination. Two images in the middle (supposed to be the same) represent ”half” homography transformation from the reference to destination and *vice-versa*. The second image was generated by projecting points of the destination image by the inverse of square root of homography, and the third image was generated by projecting points of the reference image by the square root of homography.

7.2.2 Epipolar geometry

In the case of epipolar geometry, a ”corrected” correspondence $(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')$ must lie *perfectly* on epipolar lines (*i.e.,* $\tilde{\mathbf{x}}'^{\top}\mathbf{F}\tilde{\mathbf{x}} = 0$). The exact procedure was presented by Hartley and Sturm in [29] which requires building a six-degrees polynomial; hence the method can be quite slow for a high number of points. A significantly faster iterative procedure was proposed by Lindstrom in [39] that usually converges just in a few iterations. Moreover, if the intrinsic matrices are given, then the procedure in [39] enables to efficiently obtain triangulated 3D points.

8 VSAC framework

This thesis presents a RANSAC-like framework – VSAC, which includes different settings, *state-of-the-art* methods, and proposed improvements described in previous sections. The framework solves computer vision problems such as fundamental, essential, homography, and perspective projection matrix estimation. The idea of VSAC is not only to be fast and accurate on the estimation tasks, but also to work under different configurations that suit specific problems, prior information, and to provide freedom of input setting for users.

8.1 Implementation

The framework is implemented in *C++* from scratch. The code is written in object-oriented programming style where objects and classes represent the main parts of RANSAC algorithm. That is, sampling, sample verification, minimal model estimation, model verification and evaluation, check for degeneracy, local optimization, termination, and final optimization. For instance, abstract classes are `Sampler`, `Verifier`, `Terminator` *etc.*

Algorithm 3: VSAC

```
Input: $\mathcal{P}$ – points, $\mathbb{P}$ – set of all
       parameters.
create objects using $\mathcal{P}$ and $\mathbb{P}$
for $i \leftarrow 0; i < K; i{+}{+}$ do
    $\mathcal{S} \leftarrow$ sampler.draw()
    if $\neg$verifier.valid($\mathcal{S}$) then
        continue
    $\hat{\Theta} \leftarrow$ solver.estimate($\mathcal{S}$)
    for $\hat{\theta} \in \hat{\Theta}$ do
        if $\neg$verifier.valid($\theta$) then
            continue
        $s \leftarrow$ evaluator.score($\hat{\theta}$)
        if $s \succ s^*$ then
            $\hat{\theta}' \leftarrow$ verifier.recover($\hat{\theta}$)
            if $\hat{\theta}' = \varnothing$ then
                continue
            $s'', \hat{\theta}'' \leftarrow$ LO.refine($\hat{\theta}'$)
            if $s'' \succ s^*$ then
                $s^*, \hat{\theta}^* \leftarrow s'', \hat{\theta}''$
            $K \leftarrow$ terminator.iters($\hat{\theta}^*$)
$\hat{\theta}^{**} \leftarrow$ polisher.refine($\hat{\theta}^*$)
return: $\hat{\theta}^{**}$, evaluator.inliers($\hat{\theta}^{**}$)
```

The reason for this is to make the framework more universal. The abstract class defines an interface and certain functions of its child class. For example, from `Sampler` are derived implementations of uniform or PROSAC sampling with the common function like `generate_sample()`. A user can switch among different provided options or easily integrate their own new method. Similarly, the framework is universal for all estimation problems that require RANSAC, hence by changing the minimal and non-minimal solver, VSAC is finding a solution to other tasks.

The framework is using mostly OpenCV library [10]. Some parts of the code also use Lapack [1] and Eigen [24] which are not essential but provide faster computation. An earlier stage of VSAC was integrated into OpenCV as part of Google Summer of Code 2020. It could be run with flags that begin `USAC_` as there are different settings, *e.g.,* fast, accurate, or parallel versions.

VSAC is outlined in algorithm 3. Inputs are points (sorted for progressive sampling) and set of all parameters used in each object (component). The parameters are extensive as they should allow a user to change tiny details of inner methods (*e.g.,* number of iterations in local optimization), hence it provides advanced settings. However, most of the values are set by default, and only the main parameters such as name of solving problem, confidence, the maximum number of iterations, and inlier threshold are mandatory. Afterwards, the objects are initialized accordingly to a given set of parameters. The framework can be used either in the experienced mode or by one function call.

The universal framework in alg. 3 extends the original RANSAC procedure in several ways. The first thing to note is a line 4 where `Verifier` checks correctness of sample, *e.g.,* there are no three points on a common line for homography estimation. It again plays a role in further verification of model (line 8), for instance if epipolar geometry satisfies the orientation constraint [15]. At line 10, for simplification `Evalutator` includes both a model score computation and pre-emptive verification like A-SPRT (the score is empty if solution is rejected). `Verifier` at line 12 checks *so-far-the-best* model for degeneracy (*e.g.,* fundamental matrix via DEGENSAC$^+$) ; for convenience it can return (1) the same model from minimal sample if the model is not degenerate, (2) (best found) non-degenerate model or (3) empty solution when it cannot recover. The test for degeneracy / recovery is more complex and computationally expensive than the verification of sample and model parameters, hence it is done separately in the model acceptance part. After a valid model is obtained, it undergoes the local optimization; all proposals regarding LO are not outlined in alg. 15 but they are preserved in the framework. `Terminator` returns an upper bound number of iterations or 0 for immediate termination at line 18. Finally, the *so-far-the-best* model is polished, and it is outputted together with (corrected) inliers.

8.2 Parallelization

Parallelization is an important part of computation as new technologies have become more advanced. It helps to gain a significant speed-up for free just using computer properties without any accuracy trade-off. While for some tasks the parallelization could be a bottleneck, although, for RANSAC it is more straightforward, because iterations are processed almost independently. This section presents a procedure that allows not only to parallelize the original RANSAC but the whole framework, *i.e.,* every combination of different components.

The main key of parallelization is to divide all computations among threads separately and avoid their synchronization. For instance, the textbook RANSAC proceeds as follows. At each iteration the algorithm draws a minimal sample, estimates model(s), calculates its support, and checks whether the model is a *so-far-the-best*. Every step can be processed by a thread *independently* until the part, where a new model's score is compared to the *so-far-the-best* one. Only one thread must process this step, because read / write access to shared variables by several threads causes wrong synchronization. This can be solved by introducing a mutex which encloses an area where a *so-far-the-best* model is updated. However, it is not an ideal solution, because mutex will be used in every iteration by every thread which results that some threads have to idly wait until they get a chance to enter the area.

A better and more elegant way is to parallelize not the loop over iterations, but the whole RANSAC(s) itself. In other words, the number of running RANSAC(s) is the same as the number of available threads. The communication among threads is given by a single boolean variable *terminate* which states when all individual runs must finish. The termination happens when the maximum number of iterations is reached or the termination criteria are satisfied. In either case, another integer variable (T_h in Alg. 4) has to be introduced, which counts how many iterations (hypotheses) have been processed by every RANSAC(s) in total. Both variables (*terminate* and T_h) should be atomic, so they guarantee thread exclusion for every read / write operation. Moreover, atomic operations are performed way much faster than using mutexes.

To integrate these variables with RANSAC, a couple of minor changes must be made. In the beginning, both *terminate* and T_h are initialized to *false* and 0 (respectively); additionally, since the RANSAC framework is based on objects (sampler, solver, *etc.*) then each thread receives its own copy of the object to avoid thread collision. At first, before drawing a sample RANSAC checks whether the variable *terminate* is true; if so, then it stops, otherwise it increments T_h. If the value of T_h is higher than the maximum number of iterations (given by user) then *terminate* is set to *true*. Additionally, the *terminate* can be updated to *true* when termination criteria are fulfilled in *so-far-the-best* step, *e.g.,* no better model exists (with predefined confidence) by the standard termination criterion. Eventually, each RANSAC thread has its own final model, the *so-far-the-best* one is selected with the best score among all threads.

Algorithm 4: Parallel RANSAC

```
Input: state – state of random generator,
       $K$ – max. iterations
atomic_bool terminate ← false
atomic_int $T_h \leftarrow 0$ // num hyp.
atomic_int $S \leftarrow m$ // subset size
atomic_int $S^* \leftarrow |\mathcal{P}|$ // stop size
// parallel for-loop
for $t \leftarrow 0;\ t <$ MAX_THREADS; $t$++ do
    t_state ← state + t
    for $i \leftarrow 0; i < K; i$++ do
        if terminate then
            break
        terminate ← $T_h$++> $K$
        if IS_PROSAC then
            if $S = f(T_h) \wedge S < S^*$ then
                $S \leftarrow S + 1$
            // continue ...
        else
            $\mathcal{S}$ ←generate(t_state)
        // continue ...
        if $\hat{\theta}^*$ is the-best-so-far then
            terminate ← criteria($\hat{\theta}^*$)
            models[t] ← $\hat{\theta}^*$
            if IS_PROSAC then
                // update $S^*$
return: BestModel(models)
```

Given PROSAC sampling, it slightly complicates the parallelization, because samples must be drawn sequentially from the most promising ones. A naive solution is to pregenerate all samples in the beginning and split them equally within threads, however, it is slow for the large number of iterations. The best option is to introduce two other integer atomic variables that define PROSAC behaviour. The first is subset size (S) which determines a range of top S points to be drawn; the second is termination length S^* which is the upper bound of S. Both parameters are described in details in [13]. Briefly, S is updated by a growth function (f in Alg. 4) of number of tested hypotheses (T_h), which can be precomputed for all threads in advance. While S^* is updated when a new *so-far-the-best* model is found. Atomic operations enable to safely change these variables and preserve sequential PROSAC performance. As well as in parallel uniform version, for parallel PROSAC random generators of different seeds are used. Therefore, generated samples are different.

The first thing to mention is that the presented parallel version makes the same number of iterations as the sequential RANSAC and no miscalculations related to parallelization are possible. It controls by two shared atomic variables, safely processed under concurrency, and the own local copy of objects for each thread. Second, the theoretical speed-up is proportional to the number of cores, as most of the computations are done independently. Although, there is a certain overhead caused by thread initialization. The parallel approach also works for different combinations of RANSAC-like algorithms, *e.g.,* including local optimization or SPRT. because atomic variables are not restricted by any of RANSAC components. All RANSAC(s) will immediately terminate when one of the thread set *terminate* to *true* regardless of which termination criteria are used.

However, the main disadvantage of the parallel version is no guarantee to output the same result as a sequential RANSAC. The reason for this stands by the fact that run of threads is random itself. For instance, for uniform sampling, each RANSAC thread has its own random number generator of a different seed, hence the order of drawn samples corresponds to runs of threads that are completely random. In theory, it is still possible to force the parallel version to work as sequential by pregenerating all random numbers in advance. Similarly, it also implies that giving the same seed for a random generator and the same input, the parallel method does not return the same result every time. While even the sequential RANSAC is not deterministic in theory either, although some implementations can make it possible by providing an initial state (integer number). It is convenient to compare how small changes reflect the change in output, and this is done in VSAC framework.

The parallel approach is outlined in algorithm 4. The scheme only shows the modifications that should be made to get the presented parallel version, while other RANSAC steps are the same, thus skipped. Under VSAC setting including A-SPRT, DEGENSAC$^+$ *etc.* the parallel version is being not only faster but surprisingly more accurate in experiment 9.9. The explanation for this could be that running several complex RANSAC(s) (*e.g.,* that require adaptive SPRT) may reduce the risk of bad initialization. Moreover, a number of local optimizations for parallel method is higher, suggesting that the probability of finding a better model is higher as well. Technical improvements that could help speed up the parallel framework even more, can be achieved by additional synchronization of a current *so-far-the-best* score. It is done by introducing an additional atomic variable corresponding to the best found score that is updated only in *so-far-the-best* step. This aims to keep all threads synchronized. Especially, it can boost rejection of bad models in A-SPRT for all threads, because more information about *so-far-the-best* model is available.

While there is a speed-up close to number of cores, although parallel version may not be faster for problems with high inlier ratio, and sometime for PROSAC as it usually finishes quickly, depending on the ranking quality. The reason is that the initialization of threads could overtake the whole time of even sequential version. Therefore, using parallel RANSAC is not suggested for ”easy” tasks.

9 Experiments

This chapter discusses and shows evaluations of the proposed methods (*e.g.,* Adaptive SPRT), and finally compares the framework itself against other existing RANSAC implementations. To minimize any confusion with methods' names, the VSAC framework is a set of *state-of-the-art* RANSAC methods like PROSAC, DEGENSAC, GC-RANSAC *etc*. However, the VSAC settings are new LO$^+$, A-SPRT, DEGENSAC$^+$, orientation tests, and PROSAC if point ranking is given. Some experiments include simple RANSAC setting (with DEGENSAC$^+$ and sample / model validation tests to confront invalid models) to avoid effects of other parts on the clearance of results. The common VSAC (RANSAC) input parameters for most of the experiments are 99% confidence level (no better solution exists), the inlier-outlier threshold: around 1 pixel for epipolar geometry and around 2 pixels for $\mathbf{H}$ and **PnP**. The threshold was set additionally considering average image dimension in datasets.

The experiments are mainly concern homography ($\mathbf{H}$) and fundamental matrix ($\mathbf{F}$) estimation, however some of them also include essential matrix ($\mathbf{E}$) and perspective projection (denoted as **PnP**). Estimation problems are run on a various publicly available real-world datasets: (1) KUSVOD [36] includes correct labelled correspondences and the ground truth (GT) $\mathbf{F}$; (2-3) PHOTOTOURISM [53] and STRECHAMVS [55] contain GT relative pose with intrinsic matrices (used for $\mathbf{F}$ and $\mathbf{E}$); (4-5) EVD [44] and HOMOGR [36] have labelled correct matches and GT homography matrix; (6) HPATCHES [2] includes GT homography matrix; (7-9) T-LESS [31], LM-O [9], and YCB-V [60] for **PnP** contain GT poses with calibration (in experiments only data with a single pose are used).

The evaluation of algorithms is in terms of running time and either geometric accuracy for $\mathbf{H}, \mathbf{F}$ or pose error for $\mathbf{E}$ and **PnP**. In the case of geometric accuracy, image matches are divided into two disjoint sets, where the first one is used for the RANSAC input, and the second set of a few dozens of selected points consistent with the GT model is used for a model validation. For $\mathbf{F}$ the accuracy is measured by mean geometric error (*i.e.,* distance of input GT points to their optimal correction on 4D manifold consistent with an estimated $\hat{\mathbf{F}}$), and for homography it is root-mean-square deviation of reprojection distances of GT correspondences to an estimated $\hat{\mathbf{H}}$. The pose error is measured as average of axis angle of composed rotation matrix $\mathbf{R} = \mathbf{R}^{*\top}\hat{\mathbf{R}}$ (where $\mathbf{R}^*$ – GT rotation, $\hat{\mathbf{R}}$ – estimated rotation) and arc cosine of the dot product of GT and estimated translation vectors. For $\hat{\mathbf{E}}$ the pose error is minimum of four decomposed poses, and the pose for **PnP** problem is found after decomposing a projection matrix.

Results report error in pixels (px) for $\mathbf{H}, \mathbf{F}$ or in degrees for $\mathbf{E}$, **PnP**; run-time in milliseconds (ms) averaged over image pairs from a dataset. Additionally, to avoid extreme (outlier) results, the median of errors (and run-time) over all runs and percentages of better results per image pair are shown. To avoid randomization of experiments, the same initial seed was given to the random number generator of RANSAC, therefore, all methods start operating equally. In case neither method has shown any difference, this run is excluded.

9.1 Independent inliers

This section describes the proposed method in chapter 2 for detection of random solutions. Additionally, it shows the influence of dependent inliers on the method's accuracy. A Poisson parameter for the non-randomness test is estimated from the first 21 models of RANSAC.

Problem		$\bar{\lambda} \pm \sigma$	# of RANSAC iterations			
			$\sim 10^2$	$\sim 10^3$	$\sim 10^4$	$\sim 10^5$
H	w/o	4.26 ± 0.19	95.3	92.0	84.9	79.6
	w	0.00 ± 0.00	100.0	100.0	100.0	100.0
	w*	0.00 ± 0.00	100.0	100.0	100.0	100.0
F	w/o	9.41 ± 1.20	84.6	80.3	78.6	75.3
	w	1.42 ± 0.75	99.0	99.0	99.3	99.3
	w*	1.35 ± 0.80	98.7	99.3	99.7	99.3

Table 9.1 Percentage of detected failures by the proposed criterion (2.1) with (w and w*) and without (w/o) removing dependent inliers on homography **H** and fundamental matrix **F** estimation problems when trying to match image pairs without a common field-of-view, *i.e.,* they do not match. For(w*) case, all models are generated by artificially corrupted point samples. The column $\bar{\lambda} \pm \sigma$ shows Poisson parameter averaged over image pairs with its standard deviation.

Table 9.1 reports percentages of successful failure detection of RANSAC running on image pairs with no field of view (300 pairs are matched via SIFT [40] detector with 0.8 ratio score; the same pairs are used for both **F** and **H**). Since there is no common structure, RANSAC fails every run. The methods with removing dependent inliers give better results as its role is crucial for the whole method. Dependent inliers (*e.g.,* close points) increase the support of bad models making it much higher than Poisson parameter (average number of points consistent with a bad model), hence the models appear to be good ones. Another observation is that since all sample contain bad points, the Poisson parameter of corrupted samples is almost equal to the one estimated from random samples. Both methods with removing dependent inliers show similar accuracy.

	$\lambda \pm \sigma$	$\lambda' \pm \sigma'$	$\lambda^* \pm \sigma^*$
H	65.73 ± 162.8	51.88 ± 137.7	0.02 ± 0.1
F	20.52 ± 12.7	12.66 ± 11.9	4.25 ± 1.2

Table 9.2 Comparison of estimated Poisson parameters averaged over matchable image pairs (HPATCHES dataset for **H** and PHOTOTOURISM dataset for **F**) with their standard deviation; λ – without removing dependent inliers, λ' – with removing, λ^* – with removing independent inliers of models generated from corrupted samples.

H	# of RANSAC iterations			
	$\sim 10^2$	$\sim 10^3$	$\sim 10^4$	$\sim 10^5$
$\# f$	16	16	16	16
$\# d$	10	4	3	0

Table 9.3 $\# f$ shows number of RANSAC failures for homography estimation for a range of the maximum iterations (in the columns). The second row $\# d$ shows number of failures detected by non-randomness test.

In table 9.2 are reported estimated Poisson parameters of image pairs from matchable datasets (100 pairs are used). Only Poisson parameter of independent inlier counts of models generated from corrupted samples is not spoiled, and it is close to the one reported in table 9.1. Surprisingly, the method with removing dependent inliers of destructed Poisson parameter does not have any false negatives (detecting a good model as the random one). However, to avoid any possible false negatives, the method with generating corrupted samples should be used. Additionally, table 9.3 shows number of failures detected in the matchable scenes only for **H** estimation, where the failure model has geometric error higher than 15 px. The method cannot detect all failures, because *so-far-the-best* model may describe a non-random structure which is not the correct one.

9.2 Gaussian Elimination

This section compares two RANSAC, one uses singular value decomposition (SVD) for all $\mathbf{H}, \mathbf{F}, \mathbf{E}$ and **PnP** to obtain a set of null-vectors, the other uses Gaussian elimination (GE). Table 9.4 experimentally suggests that replacing SVD by GE speeds up RANSAC several times with almost zero accuracy loss. The number of runs is 100 for each estimation problem.

In **E** estimation, it can be seen that RANSAC with GE makes slightly more iterations, while still having the same accuracy. It can be explained that in rare occasions, the output essential matrices of a minimal sample have a slightly fewer inlier number, hence RANSAC needs more hypotheses to test, however the accuracy is the same. Note, in this experiment, the geometric error for **E** is used.

For **P6P** the situation is less certain. First, in this experiment a synthetic dataset is used, because both P6P RANSAC were providing bad results on benchmark datasets [60, 31, 9]. Second, RANSAC with SVD solver seems to be more accurate as it makes much fewer iterations, although due to the efficacy of GE solver, the RANSAC with SVD is still being slower. The reason for being more accurate comes from finding a solution in terms of least-squares of 12 linear equations (6 points), while GE is applied to 11 equations (5 and half points). The Gaussian noise with 0 mean and standard deviation 0.4 was added to the image points, implying that for a higher noise level, the GE solver may work even less accurately.

		$K_{\max} = 10^4$			$K_{\max} = 5 \cdot 10^4$			$K_{\max} = 10^5$		
		t_{avg}	ε_{avg}	K_{avg}	t_{avg}	ε_{avg}	K_{avg}	t_{avg}	ε_{avg}	K_{avg}
H	SVD	5.6	0.70	372.9	5.6	0.70	372.9	5.9	0.70	372.9
	GE	2.8	0.70	372.9	2.9	0.70	372.9	3.0	0.70	372.9
F	SVD	155.7	0.47	6570.2	440.7	0.34	22697.6	664.7	0.34	36369.8
	GE	43.9	0.45	6621.2	116.0	0.38	22540.4	169.7	0.33	35657.6
E	SVD	173.2	0.53	2474.9	324.7	0.49	5168.7	381.0	0.49	6182.7
	GE	143.9	0.53	2475.2	261.5	0.49	5169.0	311.6	0.49	6183.0
P6P	SVD	222.0	26.92	10000.0	915.2	0.00	48136.8	1214.1	0.00	67357.4
	GE	131.1	6.30	10000.0	653.0	0.57	50000.0	1037.2	0.00	89946.8

Table 9.4 Comparison of average error (ε_{avg} in px. for $\mathbf{H}, \mathbf{F}, \mathbf{E}$, and in degrees for **P6P**), average time (t_{avg} in ms), and average number of iterations (K_{avg}) over a number of runs for RANSAC with SVD and GE (Gaussian elimination) minimal solver. The experiments have performed over a range of the maximum iteration number for RANSAC denoted as $K_{\max}$. The lowest values are highlighted in red. RANSAC with GE solver is the fastest in the whole table.

9.3 Adaptive SPRT

method	**H**					**F**				
	t_{avg}	t_{med}	ε_{med}	R	$\#f$	t_{avg}	t_{med}	ε_{med}	R	$\#f$
NO-SPRT	10.2	8.2	1.44	0.0	25	76.7	76.6	0.49	0.0	0
SPRT	2.1	2.0	1.45	574.9	40	40.4	39.5	0.59	3159.2	195
A-SPRT	2.9	2.5	1.44	564.7	25	44.4	45.8	0.49	3114.6	0

Table 9.5 Comparison of RANSAC without SPRT (NO-SPRT), with original SPRT settings (SPRT), and new adaptive A-SPRT, in terms of average and median run-time ($t_{\text{avg}}, t_{\text{med}}$ in ms) and median error in px. R denotes average number of rejected models by Wald's test; f – number of RANSAC failures.

The RANSAC with adaptive SPRT (A-SPRT) is compared against the ones with no Wald's test and with original SPRT setting from [43]. The results are reported in table 9.5 and in CDFs figure 9.1. According to the table, RANSAC with A-SPRT has the same accuracy as the one without, however, being only marginally slower than SPRT version. While the latter one has much more fails, where the solution is considered a failure if its error is higher than 15 pixels. A-SPRT rejects slightly fewer models than SPRT, because it uses the first 21 models for estimation of the probability ($\hat{\delta}$ from chapter 2) of point being consistent with a bad model. It suggests that the estimation of $\hat{\delta}$ is important as it reduces false negative rate (rejecting good models). In this experiment, only image pairs with a ratio of consistent points with GT model lower than 40% are selected; in total 49 for $\mathbf{H}$ and 150 for $\mathbf{F}$, additionally each image pair was repeatedly run 5 times.

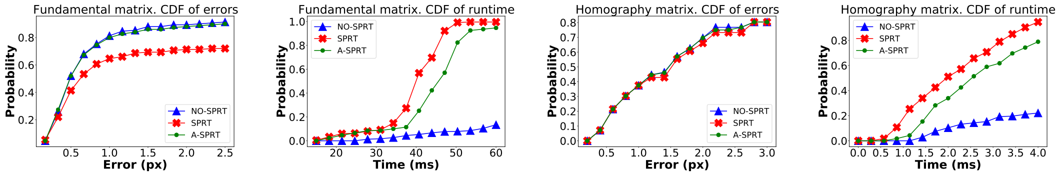

Figure 9.1 The CDF (cumulative distribution function) plots of processing times (ms) and geometric accuracy (px) of standard RANSAC with (1) no SPRT, (2) SPRT, and (3) A-SPRT for homography (figures 1-2) and fundamental matrix (figures 3-4) estimation.

9.4 Calibrated DEGENSAC$^+$

In table 9.6 three RANSAC with different methods for detection and recovery of the fundamental matrix are compared. The first one uses the approach proposed in DEGENSAC [16], the second is the novel DEGENSAC$^+$ with intrinsic matrix approximation, and the last one uses input calibration. The maximum number of plane-and-parallax RANSAC iterations is set to 200, and for RANSAC it is 5000. The DEGENSAC needs way more time for detection / recovery of *so-far-the-best* fundamental matrix, hence the total time of RANSAC is slower as well. The DEGENSAC$^+$ with input intrinsics is the fastest since it in constant time can recover $\mathbf{F}$. Moreover, the new DEGENSAC$^+$ is more accurate, suggesting that it returns models with sufficient non-planar support, while the standard plane-and-parallax RANSAC may fail to output non-degenerate models.

	ε_{avg}	ε_{med}	t_{avg}	t_{med}	$t^{\text{D}}_{\text{avg}}$
D	0.57	0.41	30.2	29.9	3.24
D$^+$	0.48	0.34	21.3	23.9	1.07
D$^+_{\mathbf{K}}$	0.39	0.29	19.5	21.5	0.95

	ε_{avg}	ε_{med}	t_{avg}	t_{med}	$t^{\text{D}}_{\text{avg}}$
D	1.28	0.32	33.0	30.7	3.33
D$^+$	1.19	0.34	22.4	24.0	0.86
D$^+_{\mathbf{K}}$	1.24	0.30	20.5	22.0	0.79

Table 9.6 Comparison of DEGENSAC (D) and DEGENSAC$^+$ (as D$^+$ and D$^+_{\mathbf{K}}$ which uses calibration) on two sets of image pairs (100 each) from PHOTOTOURISM dataset ("Sacre Coeur" on the left and "St. Peter's Square" on the right). In columns are $\varepsilon_{\text{avg}}, \varepsilon_{\text{med}}$ – average and median geometric errors; $t_{\text{avg}}, t_{\text{med}}$ – average and median computational time; and $t^{\text{D}}_{\text{avg}}$ – average time spent for detection and recovery of degenerate $\mathbf{F}$. The lowest and second lowest values are highlighted in red and blue, respectively.

9.4.1 F versus E estimation

One can argue that by knowing calibration, then RANSAC for $\mathbf{E}$ estimation could be used without applying RANSAC for $\mathbf{F}$ with the new DEGENSAC$^+$. In table 9.7 are comparison of two RANSAC estimating $\mathbf{F}$ and $\mathbf{E}$ on the same image pairs (100 in total). The number of iterations for $\mathbf{F}$-RANSAC is 20000, therefore it assumes an inlier ratio at least 30%. The corresponding number of iterations for $\mathbf{E}$-RANSAC is 1825 for the same confidence 99%. Results in 9.7 show that estimating the fundamental matrix is faster despite making almost in order of magnitude more iterations. The maximum number of solutions of a minimal solver for $\mathbf{E}$ can be up to 10 (even number), while for $\mathbf{F}$ it is either 1 or 3, hence it explains that the difference in the number of tested models is not dramatic. Surprisingly, the output is more accurate for $\mathbf{F}$ (it may be not true in general) with respect to both geometric and pose error.

	t_{med}	t_{avg}	$\varepsilon_{\text{avg}}^{\text{pose}}$	$\varepsilon_{\text{avg}}^{\text{geom}}$	K_{avg}	m_{avg}
F	88.8	109.2	4.83	0.47	12744.2	2706.7
E	101.7	118.0	6.39	0.92	1442.5	1574.1

Table 9.7 Comparison of fundamental and essential matrix estimation on the same image pairs of PhotoTourism dataset. The maximum number of iterations for $\mathbf{F}$ is 20000, for $\mathbf{E}$ is 1825. In columns are average and median run-time ($t_{\text{avg}}, t_{\text{med}}$), average pose and geometric errors ($\varepsilon_{\text{avg}}^{\text{pose}}, \varepsilon_{\text{avg}}^{\text{geom}}$); average number of iterations K_{avg}; and average number of models tested m_{avg}

9.5 Local optimization

Comparison of the VSAC setting (PROSAC + A-SPRT) with three different local optimization methods is shown in table 9.8. The first one uses Graph-Cut optimization from [4], the second and third ones have the same new structure proposed in chapter 5, only the second one additionally applies weights from MAGSAC++ [5] for non-minimal least-squares estimation. All methods follow the rules described in chapter 5 for applying local optimization, in addition, they use early termination if a model with the highest support so far is found. If VSAC finishes without applying local optimization, then it runs it in the end. The number of inner iterations of LO in this experiment is 20, the number of maximum VSAC iterations is 10^4, the total number of tested pairs is 100.

The results show that VSAC enters LO part only once, while *so-far-the-best* have been updating around 3-4 times. The average and median errors are quite low, suggesting that one LO run (or 20 least-squares optimizations) is enough to return a good model. All method have similar accuracy, while GC is the most computationally expensive, although making more optimizations than others. The proposed simple local optimization is the fastest, while being only marginally less accurate than others in $\mathbf{H}$ estimation.

		ε_{avg}	ε_{med}	t_{avg}	t_{med}	$t_{\text{avg}}^{\text{LO}}$	K_{avg}	#LO	#$\mathcal{O}$	#sftb
H	GC	0.86	0.70	7.4	6.7	4.7	17.9	1.0	55.2	4.3
	MG	0.83	0.69	4.0	4.0	2.4	18.1	1.0	20.0	4.3
	LO$^+$	0.93	0.69	2.7	2.7	1.2	20.0	1.0	20.2	4.4
F	GC	0.17	0.16	12.0	11.5	6.1	32.9	1.0	60.8	2.6
	MG	0.18	0.16	7.3	7.2	2.5	26.9	1.0	20.4	2.6
	LO$^+$	0.17	0.15	6.4	6.3	1.7	28.6	1.0	20.4	2.7

Table 9.8 Comparison of local optimization methods, GC – Graph-Cut, MG – (MAGSAC++) iterative re-weighted optimization, LO$^+$ – the proposed local optimization. In columns, $\varepsilon_{\text{avg}}, \varepsilon_{\text{med}}$ – average and median errors (in px.) over image pairs; $t_{\text{avg}}, t_{\text{med}}$ – average and median run-time (in ms); $t_{\text{avg}}^{\text{LO}}$ – average computational time of a method; K_{avg} – average number of RANSAC iterations; #LO – average number of entering to local optimization part; #$\mathcal{O}$ – average number of optimizations (non-minimal estimations) inside a method; #sftb – average number of *so-far-the-best* models in RANSAC.

9.6 Final Optimization

		0	1	2	3	4	5
H	ε	1.120	0.718	0.648	0.627	0.618	0.612
	ε'	1.120	0.718	0.648	0.627	0.617	0.612
	t/t'	-	0.751	1.338	1.851	2.298	2.689
F	ε	2.484	2.004	1.921	1.898	1.889	1.885
	ε'	2.484	1.997	1.909	1.886	1.878	1.875
	t/t'	-	0.894	1.140	1.263	1.335	1.384

Table 9.9 Final optimization using covariance matrix on homographies (**H**) and fundamental matrices (**F**). Columns (0–5) show the results in the subsequent iterations of iterated LSQ. The average error (px) of the standard LSQ approach (ε) and the proposed one (ε'), run-time ratio (t/t' – speed-up) of the standard fitting to the proposal are reported. Total number of tested final models and average number of inliers for homography is 13999 and 843; for fundamental matrix are 938 and 361, respectively.

This section aims to discuss the role of final model polishing on the accuracy of the output RANSAC model. First, in the experiments shown in table 9.9 was observed that all-inlier least squares applied several times in a row can significantly improve the accuracy of final model. Additionally, table 9.9 reports speed-up gained by the new polisher with respect to the ordinary least-squares optimization.

Second, table 9.10 shows comparison of different final polishing methods in terms of improving a *so-far-the-best* RANSAC model from a minimal sample. The polishers are as follows: (1) the standard LSQ, (2) the covariance LSQ, (3) LSQ with MAGSAC++ weights, (4) the proposed local optimization method (without termination) applied instead of final polisher. Polishers 1-3 use at most 5 iterations and all-inlier samples, while the LO uses 20 iterations and sample size is 40 points for **F** and 21 points for **F**. For each estimation problem, the covariance polisher is the fastest (except median time for **F** that means that normalizing all points in advance had a small overhead). The most accurate polishing method is weighted LSQ (except the average time for **F**). Iterative polisher appears to be more accurate on average in **F** estimation, however, it brings a potential risk of returning a degenerate solution as was discussed in chapter 5. Therefore, covariance polisher is an ideal choice since it is only slightly less accurate than weighted polisher, but works several times faster.

		$\varepsilon^{\text{R}}_{\text{avg}}$	ε_{avg}	ε_{med}	$w^{\varepsilon}_{\%}$	t_{avg}	t_{med}	$w^{t}_{\%}$
H	LSQ	1.67	0.70	0.38	8.0	697.3	590.5	27.0
	COV		0.70	0.35	10.0	365.5	345.5	73.0
	MGS		0.51	0.32	65.0	1546.0	1519.5	0.0
	ITR		0.71	0.39	17.0	1418.0	1396.0	0.0
F	LSQ	2.17	1.31	0.59	16.1	164.1	98.5	35.7
	COV		1.23	0.56	23.2	144.0	126.0	64.3
	MGS		1.40	0.50	16.1	542.0	481.0	0.0
	ITR		0.97	0.58	44.6	1328.4	1344.5	0.0

Table 9.10 Comparison of final polishers, LSQ – the standard all-inlier least squares, COV – fast LSQ using covariance matrix, MGS – weighted LSQ, ITR – local optimization. In columns, $\varepsilon^{\text{R}}_{\text{avg}}$ – average error of *so-far-the-best* RANSAC models of minimal sample; $\varepsilon_{\text{avg}}, \varepsilon_{\text{med}}$ – average and median errors on GT points after applying final optimization; $t_{\text{avg}}, t_{\text{med}}$ – average and median time of polishing method. $w^{\varepsilon}_{\%}, w^{t}_{\%}$ show percentages of polisher being the most accurate (resp., the fastest) over image pairs.

9.7 Quasi pseudo-random sampling

		t_{avg}	t_{med}	$w^t_{\%}$	ε_{avg}	ε_{med}	$w^{\varepsilon}_{\%}$	K_{avg}	m_{avg}	b	$\mathcal{S}_I$
H	U	4.2	2.0	75.0	0.90	0.66	61.0	834.8	149.5	4.5	2.2
	Q	4.2	3.0	25.0	1.00	0.68	39.0	613.4	131.6	6.0	2.9
F	U	96.6	118.9	17.0	0.34	0.21	59.0	13686.2	3078.4	4.7	2.1
	Q	70.9	84.7	83.0	0.38	0.23	41.0	14354.1	2512.9	4.6	1.5

Table 9.11 Comparison of uniform (U) and quasi-random (Q) sampling over 100 image pairs. In columns, $t_{\text{avg}}, t_{\text{med}}$ – average and median run-time (in ms); $\varepsilon_{\text{avg}}, \varepsilon_{\text{med}}$ – average and median errors (in px.); $w^t_{\%}, w^{\varepsilon}_{\%}$ – percentages of the lowest runtime (error) pairwise; K_{avg} – average number of iterations; m_{avg} – average number of tested models; b – average number of *so-far-the-best* models; $\mathcal{S}_I$ – average number points consistent with GT model in minimal sample. The best values are highlighted in red.

Quasi random sampling was proposed in chapter 5 as local optimization step of minimal sample size. The table 9.11 shows comparison of RANSAC with uniform sampling, and RANSAC with applied quasi-random sampling when *so-far-the-best* model is updated. Unfortunately, the reported results are quite contradictory and do not allow to make a final decision whether this optimization is worthful.

For **H** estimation, the RANSAC with uniform sampling is in 75% of cases faster (100 pairs tested in total, average ratio of points consistent with GT model is 52% for **H** and 30% for **F**). However, RANSAC with quasi-sampling makes fewer iterations and tested models, it has more *so-far-the-best* models, and number of consistent matches with GT model in 4-point sample is higher.

The results for **F** estimation shows almost the opposite. RANSAC with quasi-sampling optimization is in 83% cases faster, however number of iterations is higher, and number of points consistent with GT in sample is lower. The number of tested models has remained fewer as well.

9.8 Epipolar geometry solver

This section discusses the results regarding the proposed non-minimal solvers for essential matrix in section 4.2.1. Additionally, it addresses the issue with enforcing singular values to satisfy rank constraint (**F**) and, also, to force the first two of them to be equal (**E**). The problem stems from correcting the singular values, which makes the estimate worse.

9.8.1 Non-minimal optimization

In table 9.12 is comparison of presented in section 4.2.1 non-minimal solvers: (1) estimating non-minimal **F** via 8-point algorithm and its conversion to **E** via calibration, (2) estimating directly **E** from normalized points by intrinsics, (3-4) partial optimization of **E** via translation (resp. rotation) component. The table shows that the first two methods are best for both geometric and pose accuracy if the singular values of epipolar geometry are not corrected. However, in case a precise **E** is required then the best option considering time and geometric error is the one with partial optimization of the translation vector. An optimized essential matrix via rotation part is more often accepted by RANSAC as it has the lowest geometric error.

E		avg. geom.			avg. pose			t_{avg}	n_{avg}
		ε_{R}	ε_{O}	ε_{F}	ε_{R}	ε_{O}	ε_{F}		
PhotoTourism	A1	0.292	1.319	0.257	2.24	2.22	2.20	97.0	1.0
	A1′		0.209	0.209		2.22	2.22	83.3	1.0
	A2		1.371	0.221		2.59	2.22	74.4	1.0
	A2′		0.212	0.207		2.59	2.27	70.4	1.0
	A3		0.343	0.221		2.43	2.26	71.0	2.0
	A4		0.384	0.219		3.13	2.26	995.8	3.6

Table 9.12 Comparison of non-minimal solvers (A1-A4) for essential matrix (**E**), where A1 – 8-point algorithm for **F** followed by applying calibration to obtain **E**, A2 – 8-point method on normalized (”calibrted”) points, A3, A4 – partial optimization of translation (resp. rotation) part of **E**; in A1′ and A2′ methods correction of singular values is relaxed (rows are highlighted in pink, and they are not included in comparison). In columns are average geometric and pose errors: ε_{R} – error of RANSAC model of minimal sample, ε_{O} – error of optimized model, ε_{F} – error of final accepted model; t_{avg} (in ms) shows average run-time of solver; n_{avg} – average number of solutions of solver.

9.8.2 Correction of singular values

PhotoTour		ε_{avg}	ε_{med}	$w^{\varepsilon}_{\%}$	t_{avg}	t_{med}	$w^{t}_{\%}$	K_{avg}
E	RSC	2.21	0.38	60.0	154.4	160.7	3.0	1772.5
	RSC′	2.40	0.45	40.0	115.6	86.2	97.0	1595.7
F	RSC	0.16	0.13	54.5	25.4	25.5	16.0	1895.8
	RSC′	0.17	0.14	45.5	22.1	22.8	84.0	1905.9

Table 9.13 Comparison of two RANSAC (RSC, RSC′) with local optimization for **F** and **E** matrix estimation on PHOTOTOURISM dataset. In RSC after non-minimal optimization an epipolar geometry is subject to correction of singular values in order to satisfy its properties, while in RSC′ this constraint is relaxed. In columns are shown average and median errors (ε_{avg}, ε_{med}; in px for **F**, in degrees for **E**), average and median time ($t_{\text{avg}}, t_{\text{med}}$ in ms), percentages of pairwise better results w.r.t. time ($w^{t}_{\%}$) and error ($w^{\varepsilon}_{\%}$); the average number of RANSAC iterations – K_{avg}.

Table 9.13 reports a comparison of two RANSAC with local optimization, where the first has the correction of singular values, while for the second one this constraint is relaxed. The experiment is done for both **F** and **E** on 100 pairs of PHOTOTOURISM dataset.

As was mentioned in section 4.2.1, if **E** is evaluated on pose error then there is no need for singular value correction. The result for **E** in table 9.13 shows that RANSAC with the proposed approach is in 97% runs faster, and its median is almost twice less. In spite of the pose errors are just slightly higher, the gained speed-up is worthwhile.

For **F** in the experiment from table 9.13, zeroing of the last singular value has remained but only in the final optimization, hence RANSAC always return 2-rank matrix (if FO fails, the *so-far-the-best* of minimal sample is accepted). The idea is again to speed-up the total run-time by avoiding SVD of **F** every time after non-minimal estimation in the LO part, and to generate models that better fit data points. The table reports that this approach makes RANSAC in 84% of cases faster with negligible loss of accuracy.

9.8.3 Outlier removal

In table 9.14 are compared least-squares optimization of *so-far-the-best* fundamental matrix with and without discarding incorrect matches consistent with **F**. The results show that with removing outliers, the geometric error of the final optimization decreases by around 0.01-0.02 pixels, which is reasonable considering the number of mismatched correspondences consistent with the found epipolar geometry is low. However, the time for estimation has raised about 30% for filtering points.

	$\varepsilon^{\text{R}}_{\text{avg}}$	ε_{avg}	ε_{med}	$w^{\varepsilon}_{\%}$	t_{avg}	t_{med}	$w^{t}_{\%}$
LSQ	0.459	0.292	0.183	44.5	311.7	304.5	86.0
LSQ*		0.274	0.173	55.5	413.3	412.0	14.0

Table 9.14 Comparison of LSQ methods, where LSQ* additionally removes incorrect matches consistent with *so-far-the-best* $\mathbf{F}$. In columns, $\varepsilon^{\text{R}}_{\text{avg}}$ – average error of *so-far-the-best* RANSAC models of minimal sample; $\varepsilon_{\text{avg}}, \varepsilon_{\text{med}}$ – average and median errors (px) on GT points after applying final optimization; $t_{\text{avg}}, t_{\text{med}}$ – average and median time (ms). $w^{\varepsilon}_{\%}, w^{t}_{\%}$ – percentages of more accurate / faster output over 100 image pairs of PhotoTourism dataset.

9.9 Parallel RANSAC

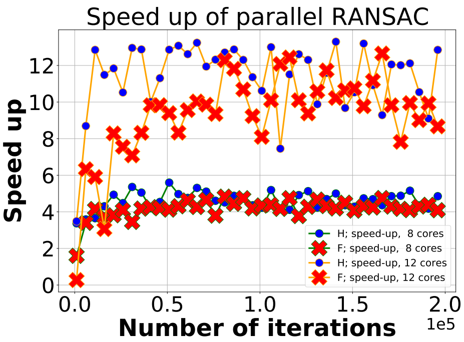

Figure 9.2 The plot shows a speed-up of parallel RANSAC to the standard one. Both were running with the fixed number of iterations (from 1000 to 200000 with step 5000, see x-axis) on two Linux machines with 12 cores (yellow lines) and 4 cores (green lines). Red crosses denote fundamental matrix and blue dots – homography estimation.

The figure 9.2 shows a theoretical speed-up proportional to a number of cores of machine for both $\mathbf{F}$ and $\mathbf{H}$ estimation. The RANSAC in experiment has the fixed number of iterations without earlier termination. It is done on purpose for two reasons: 1) see speed-up as a function of iteration number; 2) avoid randomization of results, since the parallel version does not output the same output as the sequential one. From the plot 9.2 can be observed that for a low number of iterations, the speed-up is less than predicted. The reason for this is the overhead caused by thread creation, which overcomes the time needed for the standard RANSAC. For the higher number of iterations, the speed-up remains constant. Graphs for 12-core machine[1] look very noisy, it could have been caused by running the experiment on a server that is also available for other users. Therefore, during computation, some threads may have been distracted by other tasks. While, the results of 4-core machine are generated on a local computer[2].

[1] Debian, 2 sockets, 6 cores per socket, 2 threads per core, Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz
[2] Ubuntu, 1 socket, 4 cores per socket, 2 threads per core, Intel(R) Core(TM) i7-3770K CPU @ 3.50GHz

		ε_{avg}	ε_{med}	t_{avg}	t_{med}	K_{avg}	E(t/t')
F	RSC	0.25	0.21	88.4	54.8	27874.0	2.70
	RSC′	0.24	0.18	26.5	15.9	23039.0	
	PSC	0.79	0.20	68.5	23.6	10059.0	2.53
	PSC′	0.20	0.18	21.1	11.9	8506.5	
H	RSC	1.29	0.84	1.9	1.0	397.0	1.06
	RSC′	1.09	0.84	1.5	1.2	594.5	
	PSC	1.33	0.84	2.3	1.3	329.0	1.03
	PSC′	1.23	0.85	1.9	1.6	674.0	

Table 9.15 Comparison of RANSAC (RSC) and PROSAC (PSC) against their parallel versions RSC′ and PSC′ respectively over a range of image pairs. In columns, $\varepsilon_{\text{avg}}, \varepsilon_{\text{med}}$ – average and median errors (int px.); $t_{\text{avg}}, t_{\text{med}}$ – average and median time (in ms); K_{avg} – average number of iterations; E(t/t') – average speed-up of parallel method pairwise.

The table 9.15 reports the speed-up of parallel VSAC with uniform (RANSAC) and PROSAC sampling in real-case scenarios. All runs were done on the 4-core machine. Image pairs with a ratio of consistent points with GT model higher than 60% for **F** and 50% for **H** were excluded as the cases when parallel implementation is slower due to its overhead, in total 100 pairs for **F** and 62 pairs for **H** were tested. The results suggest that parallel implementation is significantly faster for low inlier-ratio problems, where RANSAC has to draw more samples. For instance, in **F** estimation, methods make tens of thousands of iterations, therefore the speed-up is more visible than for **H** estimation, in which just hundreds of tested hypotheses. The parallel versions seem marginally more accurate.

9.10 VSAC versus competitors

The proposed VSAC setting incorporating the proposed A-SPRT, calibrated DEGENSAC$^+$, fast LO$^+$, PROSAC, and final covariance-based polishing, and its more accurate version (VSAC$_{\text{MGS}}$) with iterative re-weighted least squares polishing of weights from [5] are compared with the following *state-of-the-art* robust estimators for homography and fundamental matrix estimation:

1. USACv20 – framework from [32] with SPRT, GC-RANSAC, DEGENSAC, and Progressive NAPSAC.
2. Default OpenCV RANSAC implementation.
3. USAC framework from [51] with SPRT, LO-RANSAC, DEGENSAC, and PROSAC.
4. GC-RANSAC from [4] with SPRT, PROSAC, and DEGENSAC.
5. MAGSAC++ from [5] with PROSAC and DEGENSAC.
6. ORSA – RANSAC with a contrario approach [19].

The code of the methods listed above are taken either from GitHub or authors' webpages; all of them are implemented in C++. The methods were run on the same computer (Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz), and they were compiled into a single C++ executable, with all optimization switches on.

9.10.1 Homography

The table 9.16 reports comparison results of the proposed VSAC setting for homography matrix estimation The maximum number of iterations is 3000. The proposed VSAC has the lowest average, median, and the maximum time, while having a similar accuracy as Graph-Cut or MAGSAC++ implementations. Additionally, the slightly slower VSAC$_{\text{MGS}}$ version with weighted LSQ polisher has the top accuracy on some datasets. The CDFs graphs in figure 9.3 show that VSAC$_{\text{MGS}}$ has the highest probability of reaching accurate solutions than others. Similarly, the run-time CDFs report that VSAC finishes much earlier, *e.g.,* more than 40% of cases are solved within just a few milliseconds.

H	Method	Time (milliseconds)				Error (pixels)			
		t_{med}	t_{avg}	$t_{\max}$	$w^t_{\%}$	ε_{med}	ε_{avg}	$\varepsilon_{\max}$	$w^{\varepsilon}_{\%}$
EVD (10 pairs)	**VSAC**	① 0.6	① 1.0	① 4.1	① 100	① 2.64	3.16	② 6.34	① 28
	VSAC$_{\text{MGS}}$	② 0.9	② 1.3	② 4.3	0	② 2.67	① 3.02	7.18	13
	USACv20	2.6	5.3	24.6	0	2.99	3.34	8.17	16
	USAC	12.6	15.9	58.1	0	8.12	176.19	474.08	2
	OpenCV	19.7	22.4	62.7	0	3.51	4.06	7.11	1
	GC	10.2	17.6	64.7	0	2.77	② 3.14	7.51	② 21
	MGSC++	31.6	29.4	91.4	0	3.44	3.34	① 4.81	16
	ORSA	66.8	86.7	267.2	0	148.50	174.25	438.44	3
	Cross-validation error on the ground truth points:					1.75	1.80	2.29	
HPatches (100 pairs)	**VSAC**	① 2.2	① 2.3	① 4.7	① 57	0.70	0.93	7.34	8
	VSAC$_{\text{MGS}}$	4.7	② 5.0	② 9.1	0	② 0.69	① 0.78	① 2.12	11
	USACv20	7.4	7.9	21.8	0	0.70	0.86	2.93	7
	USAC	51.8	60.4	177.5	0	0.75	1.23	8.15	① 17
	OpenCV	② 2.9	11.2	76.2	② 43	0.72	0.94	2.70	① 17
	GC	37.7	41.2	102.2	0	① 0.68	① 0.78	② 2.18	② 16
	MGSC++	28.7	60.3	907.8	0	0.73	② 0.82	2.29	10
	ORSA	676.5	1149.8	6197.5	0	0.82	223.20	177.80	14
	Cross-validation error on the ground truth points:					0.65	0.65	1.15	
Homogr (10 pairs)	**VSAC**	① 0.6	① 0.9	① 4.0	① 79	1.14	1.28	2.74	② 18
	VSAC$_{\text{MGS}}$	② 1.0	② 1.6	7.3	0	① 0.95	① 1.18	① 1.83	7
	USACv20	1.4	1.9	② 6.2	0	1.18	1.26	2.33	② 18
	USAC	8.1	17.3	75.8	0	1.38	6.64	430.65	11
	OpenCV	2.4	4.0	12.5	② 21	1.39	1.63	4.54	11
	GC	9.4	11.9	35.9	0	1.08	② 1.20	② 2.12	0
	MGSC++	5.8	10.3	58.3	0	② 0.98	1.40	6.24	① 20
	ORSA	77.5	387.7	1761.2	0	1.26	2.11	20.04	16
	Cross-validation error on the ground truth points:					0.71	0.79	1.23	

Table 9.16 Comparison of speed and accuracy of the RANSAC methods. Average t_{avg}, median t_{med}, and the maximum $t_{\max}$ running time (ms) over all runs on HPatches, EVD, and Homogr datasets (for homography matrix estimation), including 10 repetitions for two latter ones. Similarly, the table reports average ε_{avg}, median ε_{med} and the maximum $\varepsilon_{\max}$ error (px) of ground truth points. To account the randomization, the percentage of top results is given in the $w^t_{\%}$, $w^{\varepsilon}_{\%}$ columns. The errors of the top methods are close to the accuracy of ground truth points (cross-validation error), estimated using leave-one-out.

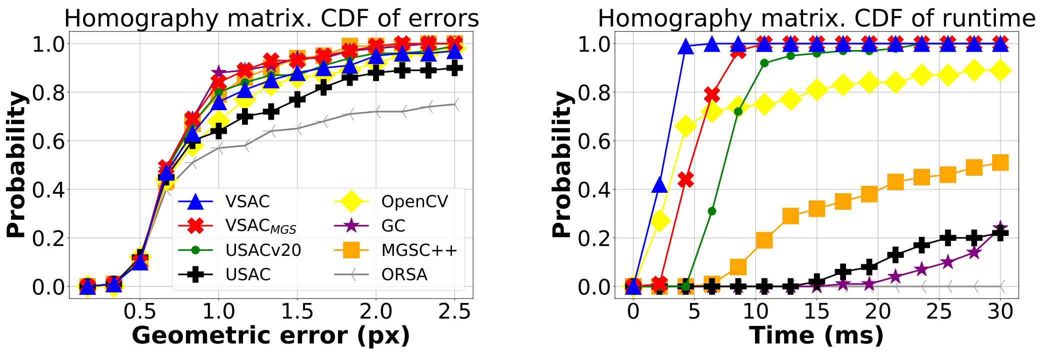

Figure 9.3 CDF plots of geometric accuracy (left figure) and run-time (right figure) of the proposed framework and its competitors for homography matrix estimation on HPATCHES dataset.

9.10.2 Fundamental matrix

F	Method	Time	(milliseconds)			Error	(pixels)		
		t_{med}	t_{avg}	t_{max}	$w^t_{\%}$	ε_{med}	ε_{avg}	ε_{max}	$w^{\varepsilon}_{\%}$
Kusvod (15 pairs)	**VSAC**	① 3.4	① 5.9	① 22.4	① 40	0.53	② 0.71	① 2.07	3
	VSAC$_{\text{MGS}}$	4.4	7.0	23.1	0	① 0.43	① 0.67	① 2.07	① 23
	USACv20	② 4.2	12.8	51.6	② 27	0.55	2.86	28.56	13
	USAC	② 4.2	② 6.0	② 22.9	20	0.54	4.56	51.58	13
	OpenCV	18.8	58.6	170.3	7	0.93	4.52	38.11	0
	GC	26.9	50.8	173.2	0	0.57	0.96	② 4.33	② 20
	MGSC++	53.1	110.3	309.9	0	② 0.46	4.39	56.39	② 20
	ORSA	38.6	80.7	634.5	7	0.50	10.77	111.25	7
	Cross-validation error on the ground truth points:					0.91	1.12	2.34	
PhotoTourism (100 pairs)	**VSAC**	① 6.2	① 6.2	① 11.5	① 91	② 0.16	① 0.16	0.78	① 22
	VSAC$_{\text{MGS}}$	② 7.7	② 7.8	② 14.7	0	① 0.15	② 0.17	0.80	10
	USACv20	46.6	56.1	123.4	1	0.17	0.21	1.08	9
	USAC	12.0	12.8	42.6	② 8	0.17	0.33	12.07	9
	OpenCV	229.8	203.0	349.9	0	0.35	0.63	7.47	0
	GC	243.3	209.5	350.2	0	0.19	0.20	① 0.47	12
	MGSC++	293.8	316.2	1270.4	0	② 0.16	0.18	1.24	② 16
	ORSA	108.1	129.7	448.2	0	① 0.15	② 0.17	② 0.50	① 22
	Cross-validation error on the ground truth points:					0.06	0.07	0.11	
StrechaMVS (100 pairs)	**VSAC**	① 4.2	① 5.2	② 30.9	① 40	① 0.21	① 0.31	② 1.99	13
	VSAC$_{\text{MGS}}$	5.7	② 6.5	31.6	0	① 0.21	① 0.31	① 1.84	11
	USACv20	② 5.4	18.7	113.1	6	② 0.22	0.42	3.29	② 16
	USAC	6.4	7.9	① 22.8	② 27	② 0.22	0.66	21.28	12
	OpenCV	20.8	60.6	197.2	② 27	0.50	0.83	6.21	7
	GC	30.6	57.4	216.3	0	② 0.22	② 0.33	2.68	① 18
	MGSC++	86.9	257.2	3442.7	0	① 0.21	0.36	5.99	15
	ORSA	74.5	168.4	923.7	0	① 0.21	23.15	599.78	8
	Cross-validation error on the ground truth points:					0.20	0.38	7.09	

Table 9.17 Geometric accuracy (px) and run-time (px) comparison of RANSAC methods for fundamental matrix estimation on KUSVOD, PHOTOTOURISM, and STRECHAMVS datasets.

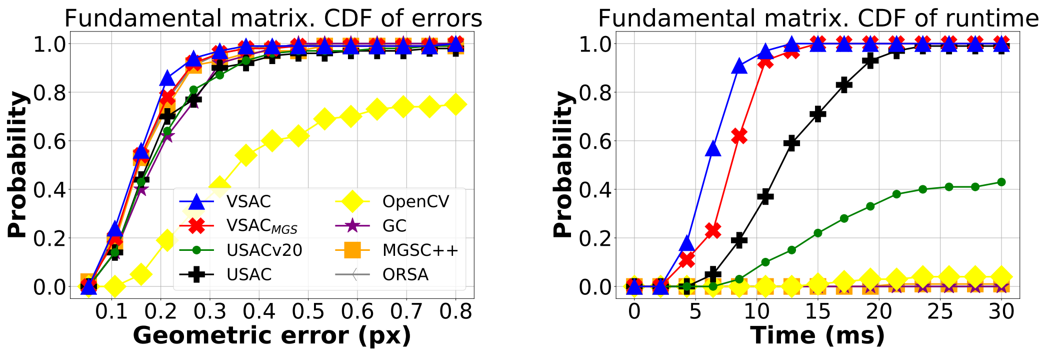

Figure 9.4 CDF plots of geometric accuracy (left figure) and run-time (right figure) of the proposed framework and its competitors for fundamental matrix estimation on PhotoTourism dataset.

In table 9.17 are compared robust estimators with the maximum number of iterations equal to 5000. Similarly to $\mathbf{H}$ estimation, VSAC shows top run-time performance, while VSAC$_{\text{MGS}}$ is one of the most accurate. The CDFs plots in figure 9.4 agree with the table's results.

9.10.3 Essential matrix

For a five-point essential matrix estimation (the maximum number of iterations is 1000), USAC and ORSA estimators are excluded from the comparison as they do not contain $\mathbf{E}$ solver. The results are reported in table 9.18. VSAC and VSAC$_{\text{MGS}}$ seem to be less accurate, which can be also observed from CDFs of errors in figure 9.5. However, the reported pose error is in degrees, hence the difference is not dramatic. Moreover, the proposed estimator is tens of times faster as can be seen from CDF of run-time in figure 9.5, *e.g.,* it finds a solution within 10 ms with probability 60%. This can compensate the slight loss of accuracy.

E	Method	Time (milliseconds)				Error (degrees)			
		t_{med}	t_{avg}	$t_{\max}$	$w^t_{\%}$	ε_{med}	ε_{avg}	$\varepsilon_{\max}$	$w^{\varepsilon}_{\%}$
PhotoTourism	**VSAC**	① 7.7	① 12.7	① 95.1	① 97	② 0.45	1.96	① 24.20	13
	VSAC$_{\text{MGS}}$	② 9.2	② 14.0	② 96.0	0	0.47	2.01	② 26.35	12
	USACv20	78.0	71.7	114.3	② 3	0.47	2.26	42.70	15
	OpenCV	368.5	300.9	437.5	0	1.18	4.84	43.93	6
	GC	118.9	115.8	185.7	0	0.53	② 1.83	30.23	② 23
	MGSC++	125.6	126.4	199.2	0	① 0.37	① 1.40	47.17	① 31
StrechaMVS	**VSAC**	① 4.9	① 13.4	84.6	① 87	0.40	0.94	7.15	9
	VSAC$_{\text{MGS}}$	② 6.3	② 14.6	85.1	0	0.37	0.89	② 6.10	11
	USACv20	18.8	29.9	① 83.7	1	① 0.24	0.99	34.84	① 34
	OpenCV	28.3	100.6	366.8	② 6	0.39	② 0.58	① 3.17	14
	GC	24.0	30.3	② 84.5	② 6	0.50	1.57	16.92	14
	MGSC++	16.5	42.4	284.4	0	② 0.34	① 0.51	6.36	② 18

Table 9.18 Pose error (in degrees) and run-time (ms) comparison of RANSAC methods for essential matrix estimation on PhotoTourism, and StrechaMVS datasets (100 tested pairs each).

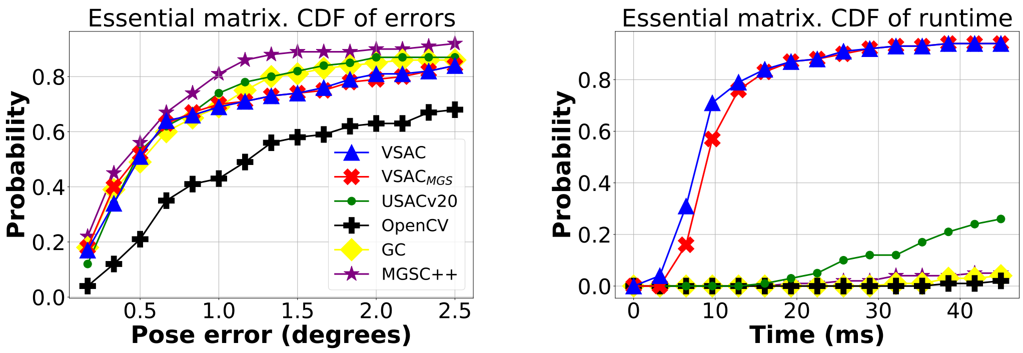

Figure 9.5 CDF plots of pose errors (left figure) and run-time (right figure) of the proposed framework and its competitors for essential matrix estimation on PhotoTourism dataset.

9.10.4 Perspective-n-Point

Table 9.19 reports comparison of **PnP** estimators. In this experiment, VSAC$_{\text{MGS}}$ is excluded since it is no longer a two-image problem. The VSAC uses the classical P3P solver described in [50] and directed least-squares (DLS) method from [30] for a non-minimal optimization. The competitors are the following OpenCV implementations:

1. EPnP – Efficient Perspective-n-Point Camera Pose Estimation [37].
2. P3P – Complete Solution Classification for the Perspective-Three-Point Problem [22].
3. AP3P – An Efficient Algebraic Solution to the Perspective-Three-Point Problem [35].

The results from table 9.19 and CDFs in figure 9.6 show that the presented estimator is less accurate (on average and median) for around 3 degrees in T-LESS, around 1 degree in LM-O, and around 1-2 degrees in YCB-V than the most accurate EPnP estimator. Although, VSAC is again the fastest implementation, and it is up to several times faster than the most accurate one. The maximum iteration number in this experiment is 5000.

PnP	Method	Time (milliseconds)				Error (degrees)			
		t_{med}	t_{avg}	t_{max}	$w^t_{\%}$	ε_{med}	ε_{avg}	ε_{max}	$w^\varepsilon_{\%}$
T-LESS	**VSAC**	① 12.1	① 15.5	① 44.5	① 85	4.03	17.97	90.20	13
	EPnP	242.9	392.2	1602.9	0	① 1.55	① 13.26	90.04	① 41
	P3P	② 62.2	② 124.2	686.0	7	1.86	② 14.56	① 89.90	② 24
	AP3P	63.0	124.5	② 668.7	② 8	② 1.82	15.28	② 90.00	23
LM-O	**VSAC**	① 10.0	① 12.3	① 74.9	① 65	4.99	6.46	29.69	② 22
	EPnP	71.3	142.4	631.8	0	① 4.14	② 5.47	① 25.05	① 42
	P3P	19.2	② 37.6	② 310.1	② 22	② 4.19	① 5.44	26.17	18
	AP3P	② 18.9	40.7	310.8	13	4.33	5.69	② 26.10	18
YCB-V	**VSAC**	① 10.9	① 11.5	① 26.5	① 70	1.98	3.58	27.40	② 24
	EPnP	45.1	100.5	875.1	2	1.67	① 1.74	① 4.86	① 41
	P3P	② 17.8	44.1	634.5	② 21	① 1.53	1.95	② 22.77	18
	AP3P	18.6	② 42.6	② 583.0	7	② 1.57	② 1.94	② 22.77	18

Table 9.19 Pose error (in degrees) and run-time (ms) comparison of RANSAC methods for perspective projection matrix estimation on T-LESS, LM-O, and YCB-V datasets (100 runs each).

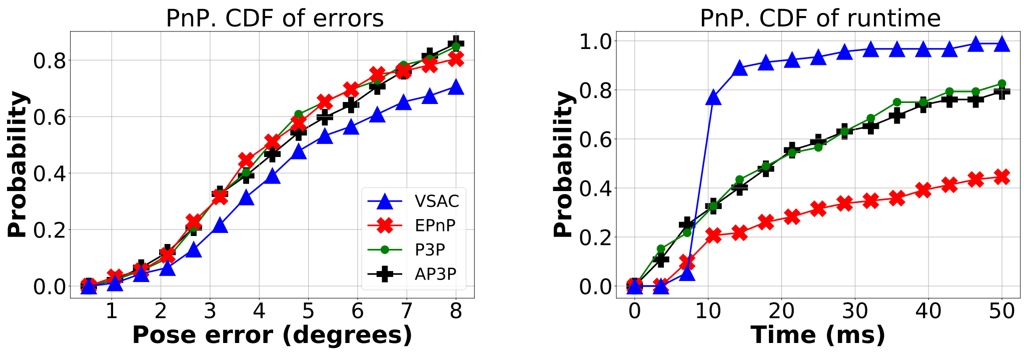

Figure 9.6 CDF plots of geometric accuracy (left figure) and run-time (right figure) of the proposed framework and its competitors for perspective projection estimation on LM-O dataset.

9.11 Correction of points

H	Method	GT	ε_{med}	ε_{avg}	ε_{max}
HPatches	VSAC$_{\text{MGS}}$	Input	0.69	0.78	2.12
		Corr.	0.43	0.57	2.08
	X-val	Input	0.57	0.58	0.80
		Corr.	0.00	0.00	0.00
EVD	VSAC$_{\text{MGS}}$	Input	2.67	3.02	7.20
		Corr.	2.14	2.69	7.60
	X-val	Input	1.75	1.80	2.29
		Corr.	0.00	0.00	0.00

F	Method	GT	ε_{med}	ε_{avg}	ε_{max}
Kusvod	VSAC$_{\text{MGS}}$	Input	0.43	0.68	2.07
		Corr.	0.42	0.65	2.07
	X-val	Input	0.91	1.12	2.34
		Corr.	0.00	0.00	0.00
MVS	VSAC$_{\text{MGS}}$	Input	0.21	0.31	1.84
		Corr.	0.19	0.28	1.77
	X-val	Input	0.20	0.38	7.09
		Corr.	0.00	0.00	0.00

Figure 9.7 The median (ε_{med}), average (ε_{avg}) and maximum (ε_{max}) errors in pixels on the used datasets when using the input ground truth correspondences and the corrected (Corr.) ones projected to the model manifold as reference inliers. **X-val** rows shows cross-validation error estimated using leave-one-out. The lowest errors are highlighted in red.

Table 9.7 demonstrates how correcting the ground truth point correspondences (proposed in section 7.2) affects the results of the accurate version VSAC$_{\text{MGS}}$. The corrected GT points have zero geometric error with respect to GT model, therefore as expected the cross-validation errors have become zero.

In all cases, using the ground truth corrected by being projected to the model manifold, it reduces the median and average errors of VSAC$_{\text{MGS}}$ method, allowing a more accurate comparison. For ε_{max}, the error is dominated by inaccuracies of the estimated model and the relatively small change between provided and corrected GT points randomly changes the error in either direction, either + or -, by a small amount. For **H** estimation, the errors $\varepsilon_{\text{avg}}, \varepsilon_{\text{med}}$ dropped by about 0.2-0.3 pixels, which is a reasonable value for the positional noise of GT points. For PhotoTourism and StrechaMVS, the GT points were selected from image correspondences perfectly fitting a model estimated from hundreds of points; their correction is minimal.

10 Conclusions

This thesis presents framework VSAC for robust estimation of homography, epipolar geometry, and perspective projection matrix. In experiments, VSAC is the fastest on average and median run-time, yet it is providing the similar accuracy as *state-of-the-art* estimators on benchmark datasets EVD, HPatches, PhotoTourism, StrechaMVS, and Kusvod2 for two-view geometry estimation. This is achieved by introducing a number of improvements.

The VSAC is able to recognize a random final model structure, *i.e.,* a failure. By using the concept of *independent* and *dependent* inliers, it is shown that support of random models follows a Poisson distribution with a single parameter that is easy to estimate. The method distinguishes image pairs without common field of view with close to zero false positive rate, while for image pairs from datasets the procedure has a zero false negative rate (rejection of a good model).

The efficacy is reached by employing fast minimal solver and light local optimization inside RANSAC loop. Exploiting Gaussian elimination in finding a matrix null-space instead of SVD brings significant speed-up for a minimal model estimation without noticeable deterioration of precision. By applying additional rules for LO$^+$ execution, it yields to its total number of runs close to 1 during the whole VSAC, in spite of a higher number of *so-far-the-best* models found. Similarly, earlier termination of LO$^+$ method saves the total run-time. The accurate model parameter is estimated in the final optimization, in which the proposed technical enhancements enable to accelerate iterative LSQ method.

Degenerate configurations of epipolar geometry are addressed in the new DEGENSAC$^+$ algorithm. The detection of random models helps to avoid failure of the method, *i.e.,* outputting degenerate model. Moreover, by incorporating intrinsic matrices, DEGENSAC$^+$ can recover the fundamental matrix in a fully planar scene in a constant time, or to detect pure rotation. If calibration is not known, the method approximates it to find a non-degenerate fundamental matrix, while still being faster than two-point plane-and-parallax RANSAC.

Adaptive input parameter estimation for A-SPRT tunes the method and reduces a number of false negatives, *i.e.,* rejection of a good model. Consequently, the method becomes as accurate as if no pre-emptive verification is used. Whereas, the gained speed-up of the whole VSAC run-time is significant.

The framework can provide a new output by correcting inliers to have zero error with respect to the final model estimate. This removes a noise from the point set, and could be helpful for further applications of inliers. In addition, some incorrect matches by chance consistent with *so-far-the-best* epipolar geometry are discarded.

The proposed robust estimator can be run in parallel, where a speed-up is proportional to a number of cores. The VSAC includes a quasi-random sampling which is likely to draw all-inlier samples, while keeping a uniform pseudo-random point selection. Another minor improvement is gained by the relaxation of singular value constraints for epipolar geometry when it is applicable. In addition, a few suggestions for essential matrix and linear P6P solvers are introduced.

Contributions of this thesis are presented in VSAC paper [33] accepted at ICCV 2021. Additionally, an earlier version of the framework is integrated into OpenCV library.

Bibliography

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. 18, 35

[2] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 4, 39

[3] Daniel Barath, Maksym Ivashechkin, and Jiri Matas. Progressive NAPSAC: sampling from gradually growing neighborhoods. *arXiv preprint arXiv:1906.02295*, 2019. 2

[4] Daniel Barath and Jiří 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. 3, 23, 43, 48

[5] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. Magsac++, a fast, reliable and accurate robust estimator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. https://github.com/danini/magsac. 3, 23, 43, 48

[6] Daniel Barath, Jana Noskova, and Jiří Matas. MAGSAC: marginalizing sample consensus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. https://github.com/danini/magsac. 2, 3, 16, 25

[7] Olivier Bauchau and Lorenzo Trainelli. The vectorial parameterization of rotation. *Nonlinear Dynamics*, 32:71–92, 04 2003. 21

[8] Hamid Bazargani, Olexa Bilaniuk, and Robert Laganière. A fast and robust homography scheme for real-time planar target detection. *J. Real-Time Image Process.*, 15(4):739–758, December 2018. 18, 19

[9] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 536–551, Cham, 2014. Springer International Publishing. 39, 41

[10] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 18, 35

[11] David Capel. An effective bail-out test for ransac consensus scoring. 01 2005. 2

[12] O. Chum and J. Matas. Randomized RANSAC with $T_{d,d}$ test. In *British Machine Vision Conference*, 2002. 2

[13] O. Chum and J. Matas. Matching with PROSAC-progressive sample consensus. In *Computer Vision and Pattern Recognition*. IEEE, 2005. 37

[14] O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In *Joint Pattern Recognition Symposium*. Springer, 2003. 3, 4, 23

[15] 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. 2, 6, 31, 36

[16] Ondrej Chum, Tomas Werner, and Jiri 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, 11, 42

[17] Edvin Deadman, Nicholas J. Higham, and Rui Ralha. Blocked schur algorithms for computing the matrix square root. In Pekka Manninen and Per Öster, editors, *Applied Parallel and Scientific Computing*, pages 171–182, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 33

[18] Bradley Efron. *The Jackknife, the bootstrap and other resampling plans*. CBMS-NSF Reg. Conf. Ser. Appl. Math. SIAM, Philadelphia, PA, 1982. Lectures given at Bowling Green State Univ., June 1980. 19

[19] Ferran Espuny, Pascal Monasse, and Lionel Moisan. A new a contrario approach for the robust determination of the fundamental matrix. In *Image and Video Technology – PSIVT 2013 Workshops*, pages 181–192, 2014. `https://github.com/pmoulon/IPOL_AC_RANSAC`. 2, 25, 48

[20] 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

[21] J.-M. Frahm and M. Pollefeys. Ransac for (quasi-)degenerate data (qdegsac). In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 453–460, 2006. 3

[22] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003. 52

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

[24] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010. 18, 35

[25] M. Harker and P. O'Leary. First order geometric distance (the myth of sampsonus). In *BMVC*, 2006. 31

[26] R. I. Hartley. In defense of the eight-point algorithm. *Transactions on Pattern Analysis and Machine Intelligence*, 1997. 20

[27] Richard Hartley and Hongdong Li. An efficient hidden variable approach to minimal-case camera motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 34, 01 2012. 18

[28] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. 18, 29

[29] Richard I. Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997. 34

[30] J. A. Hesch and S. I. Roumeliotis. A direct least-squares (dls) method for pnp. In *2011 International Conference on Computer Vision*, pages 383–390, 2011. 20, 21, 52

[31] Tomáš Hodaň, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017. 39, 41

[32] M. Ivashechkin, D. Barath, and J. Matas. USACv20: robust essential, fundamental and homography matrix estimation. In *Computer Vision Winter Workshop*, 2020. 3, 48

[33] Maksym Ivashechkin, Daniel Barath, and Jiri Matas. Vsac: Efficient and accurate estimator for h and f, 2021. 54

[34] Paul Jaccard. The distribution of the flora of the alpine zone. volume 11, pages 37–50, 1912. 7

[35] Tong Ke and Stergios I. Roumeliotis. An efficient algebraic solution to the perspective-three-point problem. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4618–4626, 2017. 52

[36] 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/. 4, 12, 39

[37] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81:155–166, 2008. 52

[38] Peter Lindstrom. Triangulation made easy. pages 1554–1561, 06 2010. 4

[39] Peter Lindstrom. Triangulation made easy. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1554–1561, 06 2010. 34

[40] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 40

[41] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, 129(1):23–79, 2021. 3

[42] Ezio Malis and Manuel Vargas. Deeper understanding of the homography decomposition for vision-based control. 01 2007. 15

[43] Jiri Matas and Ondrej 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, 4, 6, 9, 10, 31, 42

[44] D. Mishkin, J. Matas, and M. Perdoch. MODS: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 2015. 4, 39

[45] P. G. Moore. Statistical tables for biological, agricultural and medical research. by sir ronald fisher and f. yates. *Journal of the Institute of Actuaries*, 90(3):370–370, 1964. 27

[46] 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

[47] Pablo Márquez-Neila, Javier López-Alberca, José Buenaposada, and Luis Baumela. Speeding-up homography estimation in mobile devices. *Journal of Real-Time Image Processing*, 11, 01 2013. 2

[48] D. Nistér. An efficient solution to the five-point relative pose problem. *Transactions on Pattern Analysis and Machine Intelligence*, pages 756–770, 2004. 18

[49] David Nistér. Preemptive ransac for live structure and motion estimation. In *ICCV*, pages 199–206. IEEE Computer Society, 2003. 1

[50] Tomas Pajdla. Elements of geometry for computer vision. *FEE CTU,[online].[cit. 2013-05-19], March*, 2013. 52

[51] 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. `http://wwwx.cs.unc.edu/~rraguram/usac/USAC-1.0.zip`. 3, 11, 48

[52] Debra (Dallie) Sandilands. *Univariate Analysis*, pages 6815–6817. Springer Netherlands, 2014. 7

[53] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, page 835–846, New York, NY, USA, 2006. Association for Computing Machinery. 4, 39

[54] Charles V. Stewart. Minpran: A new robust estimator for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):925–938, 1995. 2

[55] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 4, 39

[56] 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

[57] 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, 28

[58] 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

[59] P.H.S. Torr, A Zisserman, and S.J. Maybank. Robust detection of degenerate configurations for the fundamental matrix. In *IEEE International Conference on Computer Vision*, pages 1037 – 1042, 07 1995. 3, 11

[60] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 06 2018. 39, 41

11 Appendix

11.1 Independent inliers

Let a correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ satisfies epipolar constraint $\mathbf{x}'^\top \mathbf{F}\mathbf{x} = 0$. If any correspondence $\mathbf{y} \leftrightarrow \mathbf{y}'$ lies on epipolar lines $\mathbf{F}^\top \mathbf{x}'$ and $\mathbf{F}\mathbf{x}$ (respectively) then it satisfies epipolar constraint too.

Proof: Assume $\mathbf{x}'^\top \mathbf{F}\mathbf{x} = \mathbf{y}'^\top \mathbf{F}\mathbf{x} = \mathbf{x}'^\top \mathbf{F}\mathbf{y} = 0$ and the goal is to show that $\mathbf{y}'^\top \mathbf{F}\mathbf{y} = 0$. Epipolar line $\mathbf{F}\mathbf{x}$ must cross epipole $\mathbf{e}'$ and point $\mathbf{x}'$. From $\mathbf{x}'^\top \mathbf{F}\mathbf{y} = 0$ it can be also seen that $\mathbf{F}\mathbf{y}$ is an epipolar line which goes through epipole $\mathbf{e}'$ and $\mathbf{x}'$. Therefore, lines $\mathbf{F}\mathbf{y} = \lambda \mathbf{F}\mathbf{x}$, $\lambda \neq 0$ are up-to-scale. Finally, $\mathbf{y}'^\top \mathbf{F}\mathbf{y} = \mathbf{y}'^\top (\lambda \mathbf{F}\mathbf{x}) = 0$ from the assumption.

11.2 Closest quadruple on the homography manifold

This section shows that a corrected correspondence $(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')$ has zero error with respect to homography $\mathbf{H} = \mathbf{A}\mathbf{A}$. The middle point $\mathbf{m}$ presented in section 7.2.1 defines the corrected points as follows:

$$\tilde{\mathbf{x}} \sim \mathbf{A}^{-1}\mathbf{m} \quad \text{and} \quad \tilde{\mathbf{x}}' \sim \mathbf{A}\mathbf{m} \tag{11.1}$$

Where $\sim$ denotes up-to-scale operator. By eliminating $\mathbf{m}$ from the two equations, the following relation holds:

$$\mathbf{A}\tilde{\mathbf{x}} \sim \mathbf{A}^{-1}\tilde{\mathbf{x}}' \quad \Longrightarrow \quad \mathbf{A}\mathbf{A}\tilde{\mathbf{x}} \sim \tilde{\mathbf{x}}' \tag{11.2}$$

Therefore, point $\tilde{\mathbf{x}}'$ is equal up-to-scale to point $\mathbf{H}\tilde{\mathbf{x}}$, and point $\tilde{\mathbf{x}}$ is equal up-to-scale to $\mathbf{H}^{-1}\tilde{\mathbf{x}}'$. By removing scale using ϕ mapping which normalizes homogeneous point by z-coordinate, the symmetric reprojection distance of corrected correspondence to $\mathbf{H}$ is indeed zero:

$$||\tilde{\mathbf{x}}' - \phi(\mathbf{H}\tilde{\mathbf{x}})|| + ||\tilde{\mathbf{x}} - \phi(\mathbf{H}^{-1}\tilde{\mathbf{x}}')|| = 0 \tag{11.3}$$

12 CD content

```
/
├─ thesis ............................................... LaTeX source code
└─ vsac ................................................. VSAC framework
   ├─ src ............................................... Source code files
   ├─ experiments ....................................... Files for experiments
   ├─ test .............................................. Files for testing
   ├─ samples ........................................... Demo files
   └─ include ........................................... Third party code and libraries
```