IMAGE PROCESSING ON LINE

Published in Image Processing On Line on 2022–03–28.
Submitted on 2021–05–10, accepted on 2022–02–27.
ISSN 2105–1232 © 2022 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2022.357

Automatic RANSAC by Likelihood Maximization

Clément Riu, Vincent Nozick, Pascal Monasse

Université Paris-Est, LIGM (UMR CNRS 8049), UGE, ENPC, F-77455 Marne-la-Vallée, France
({clement.riu,pascal.monasse}@enpc.fr, vincent.nozick@univ-eiffel.fr)

Communicated by Mariano Rodríguez *Demo edited by* Pascal Monasse

Abstract

In computer vision, and particularly in 3D reconstruction from images, it is customary to be faced with regression problems contaminated by outlying data. The standard and efficient method to deal with them is the Random Sample Consensus (RANSAC) algorithm. The procedure is simple and versatile, drawing random minimal samples from the data to estimate parameterized candidate models and ranking them based on the amount of compatible data. Such evaluation involves some threshold that separates inliers from outliers. In presence of unknown level of noise, as is usual in practice, it is desirable to remove the dependency on this fixed threshold and to estimate it as an additional unknown. Among the numerous variants of RANSAC, few, that we call "automatic", propose this approach, which involves changing the maximization criterion of consensus, as it is naturally increasing with the varying threshold. An algorithm of Zach and Cohen (ICCV 2015) uses the likelihood statistics. We present the details and the implementation of their method along with quantitative and qualitative tests on standard stereovision tasks: estimation of homography, fundamental and essential matrix.

Source Code

The ANSI C++ 03 implementation of the code that we provide is the one which has been peer reviewed and accepted by IPOL. The source code, the code documentation, and the online demo are accessible at the IPOL web part of this article[1]. Compilation and usage instructions are included in the README.txt file of the archive.

Keywords: random sample consensus (RANSAC); stereopsis

---

[1] https://doi.org/10.5201/ipol.2022.357

Clément Riu, Vincent Nozick, Pascal Monasse, *Automatic RANSAC by Likelihood Maximization*, Image Processing On Line, 12 (2022), pp. 27–49. https://doi.org/10.5201/ipol.2022.357

1 Introduction

RANdom SAmpling Consensus (RANSAC) is a fundamental tool of computer vision systems that fits data to a model while being robust to outliers. Since its introduction by Fischler and Bolles [8], it has become an essential step in the computation of homographies, fundamental matrices, and many other fitting algorithms. In its early forms, RANSAC was designed to process according to a user defined data-to-model threshold in order to characterize a data point as an inlier or outlier. Then, some automatic methods have emerged, that evaluate the data at runtime to no longer require such predefined parameters. This paper focuses on one of the most efficient automatic RANSAC methods based on the Likelihood Ratio Test [7] and its implementation.

By "automatic", we mean that the inlier-outlier threshold does not need to be fixed, but is estimated along with the model in the RANSAC framework [8]. Indeed, extensive work has been made to improve RANSAC algorithms while keeping the need for a user specified inlier-outlier threshold. Most of this work has been summarized and aggregated in USAC [14], a compilation of "best of breed" RANSAC tricks (the "U" stands for Universal).

Efforts have been made to improve the sampling strategy of RANSAC and increase the probability of drawing a good sample like in the N-Adjacent Point Sample Consensus (NAPSAC) [17], the Progressive Sample Consensus (PROSAC) [5] or GroupSAC [10]. NAPSAC rests on the idea that inliers tend to be closer together and as dimension of the data space increases, the probability of drawing an inlier close to a given inlier decreases slower than the probability of drawing a uniformly distributed outlier. Thus, a minimal sample for NAPSAC is built by uniformly drawing a point, finding all points that lie within a given sphere around it and drawing uniformly points from that sphere to build the minimal sample. This method performs well in high dimensional space where the usual drawing method requires too many iterations to succeed, but is not universally accepted as it is prone to degeneracy. PROSAC uses a quality measure of the input data points to draw inlier-only samples in a more likely order than RANSAC. This measure of quality can be derived from the similarity function used to compute the matches. This strategy usually greatly reduces runtime but can also lead to degeneracy and relies on the quality of the similarity function. GroupSAC lies in between both strategies, assuming that inliers are regrouped in regions, that can be computed by optical flow for example, and ordering the samples according to this grouping to increase the probability of drawing an all-inlier sample. This method improves the sampling efficiency but relies on an efficient and reliable grouping strategy.

Some other improvements have been proposed to speed up the verification process and increase the number of iterations: RANSAC can perform with prechecks like the $T_{d,d}$ test in [4] or a bailout test like [3]. The $T_{d,d}$ test rejects a model before testing it against all data points if it is not consistent with d random points. In [4] the optimal value of d is computed to be 1. This means this test refuses a model that is not consistent with a single randomly drawn point. This greatly increases the number of iterations required as good models are rejected by this method and needs a fast model estimation compared to the testing time to actually improve the runtime. In [3] it is proposed a bailout test that can end the consensus computation if not enough inliers have been observed so far. The stopping criterion is evaluated at each datapoint and depends on the current best inlier ratio. It requires data points to be evaluated in a random order to work efficiently. Although efficient in practice, there is no safeguard concerning the rejection of good models by this test.

Lastly, it is possible to add degeneracy checks [6] to ensure the estimated models are not corrupted by situations that can yield arbitrary good models—like points on a plane for a fundamental matrix estimation—or that the model is realistic—like adding a chirality check to ensure all points are in front of the cameras. However, such methods usually rely on some knowledge of the problem at hand and still require the user to choose an inlier/outlier threshold, whose best value varies across datasets.

All these constraints motivated investigations on “automatic” RANSAC methods like *a contrario* RANSAC (AC-RANSAC) [11, 12] or Likelihood Ratio Test (LRTSAC) [7]. These methods try to estimate this threshold during execution. Some other methods, like Residual Consensus (RECON) [15] or Marginalised Sample Consensus (MAGSAC) [1], completely remove this threshold. As the fixed inlier/outlier threshold is removed, the consensus cannot be used as a metric to compare estimated models, the inlier ratio increasing with the estimated threshold or even being impossible to compute at all if no threshold is used. The following methods all introduce a new metric to evaluate models. AC-RANSAC relies on an statistical metric, the Number of False Alarms, to determine the quality of a model for all possible inlier error cutoffs. It represents an estimate of the expected number of false positives. When this value goes below a threshold, usually set to 1, the current set of inliers is used to iterate and estimate the best model with only inliers. AC-RANSAC presents good results on difficult datasets and has increased the quality of a multi-view stereo pipeline [13]. LRTSAC uses the likelihood that a model is not issued from random data points to measure the quality of its models. This algorithm and its pros and cons are detailed in the following sections of this paper. RECON follows the hypothesis that inlier based models have a consensus on low residual points. Thus, during execution, it compares the ordered set of residual of an estimated model to previous models and the ones with the best consensus are considered inlier based. When enough good models are found—[15] proposes 3 models—they are used to refine an estimated model following a PROSAC-like method. This algorithm is slower than traditional RANSAC and shows similar results on synthetic datasets when the inlier threshold is well estimated for RANSAC. MAGSAC measures the quality of a model by measuring its likelihood given a hypothesis of uniform distribution of the inlier/outlier threshold in some interval. It removes the need for a threshold by marginalizing it and weighing pseudo-inliers to estimate a model. It shows good results on difficult datasets but does not provide an inlier set which could be used for SFM pipelines.

2 Background

2.1 Notations

The aim of the LRTSAC algorithm is to find as many inliers as possible of an unknown parameterized model among n data points $\mathbf{X} = (X_i)_{i\in[n]}$. For a stereo geometry problem, data points X_i live in a bounded part P of a D-dimensional space with $D = 4$, each data point encoding a keypoint correspondence between left and right image, $X_i = (x_i^L, y_i^L, x_i^R, y_i^R)$. Point correspondences being extracted from images, we have

$$P = [0, w_L] \times [0, h_L] \times [0, w_R] \times [0, h_R], \tag{1}$$

with w and h denoting the pixel dimensions of left and right images.

The parameters of the model to be estimated are gathered in a vector θ in a space $\Theta \subset \mathbb{R}^d$. In our case $d = 9$ for homography or fundamental matrix. Typically, θ defines a manifold in $\mathbb{R}^D$ through one of the equations

$$\begin{pmatrix} x^L & y^L & 1 \end{pmatrix} F(\theta) \begin{pmatrix} x^R & y^R & 1 \end{pmatrix}^\top = 0 \quad \text{or} \quad \left(H(\theta) \begin{pmatrix} x^L & y^L & 1 \end{pmatrix}^\top \right) \times \begin{pmatrix} x^R & y^R & 1 \end{pmatrix}^\top = 0, \tag{2}$$

where $F(\theta)$ is a fundamental and $H(\theta)$ a homography 3×3 matrix, the operator $\times$ denoting the cross product of $\mathbb{R}^3$. The parameter θ gathers the nine coefficients of the matrix and Θ is the unit sphere of $\mathbb{R}^9$. Such a constraint prevents the trivial solution $F = 0$ or $H = 0$. In the case of fundamental matrix, which is of rank 2, Θ is actually the intersection of the unit sphere with the manifold of $\mathbb{R}^{3\times 3}$ defined by the cubic equation $\det F(\theta) = 0$.

A threshold $\sigma \in \mathbb{R}$ discriminates inliers and outliers: a point X_i is considered an inlier if its residual $e(X_i, \theta)$ does not exceed σ, where the real-valued function e measures the error of a point to the model parameterized by θ. The residual can be measured by the absolute value of the left hand side of (2), called the algebraic error, or a more geometrically meaningful error (usually, the distance of X_i to the manifold defined by (2), called the gold standard error, is too difficult to compute). Ideally, σ is linked to the noise level of correct data, but in practice it is unknown. For this algorithm it is chosen from a discrete set $\Sigma = \{\sigma_{\min}, \dots, \sigma_{\max}\}$. We also define the inlier region as the set of points that are inliers given model parameters θ and inlier noise level σ: $I(\theta, \sigma) = \{X \in P, e(X, \theta) \leq \sigma\}$.

The classical RANSAC algorithm requires a reasonable σ and tests models based on consensus, the count of inliers: the tested parameter vector θ with the maximum consensus is the output of the algorithm. It is often beneficial to not consider σ a priori, but to estimate it along with the parameters of the model. However, the inlier count for a fixed model θ is obviously an increasing function of σ, and maximal consensus cannot be used anymore to evaluate the quality of a pair (θ, σ), since it would always select the maximum $\sigma_{\max}$. AC-RANSAC replaces this criterion by the number of false alarms, NFA, measured with a statistical model, that must be minimized [11]. The alternative detailed in this paper is based on the likelihood, as presented in [7]. The criterion is the likelihood of the hypothetic pair (θ, σ) as supported by the available data $\mathbf{X}$, based on a background model for its distribution.

2.2 Regular RANSAC

The usual RANSAC procedure follows Algorithm 1. A random subset of s_Θ data points is drawn from $\mathbf{X}$ at each iteration. This subset usually consists of the minimal number of points to fit a model. We get $s_\Theta = 4$ for a plane homography, $s_\Theta = 7$ or 8 for a fundamental matrix, depending on whether the 7-point or 8-point algorithm is used [9]. From such a sample, the function M_Θ returns the set of model parameters that fit the sample. In the simplest case, this is a single θ, but it may be a higher number of parameter vectors, such as 3 for the 7-point algorithm for fundamental matrix. It is also possible that M_Θ returns no model parameters, such as when the sample produces a degenerate system of equations.

Algorithm 1: Regular RANSAC procedure. Blue lines are optional.

```
parameter : maximum number of iterations $T$, error tolerance $\sigma \geq 0$, Confidence w.r.t type II error $p_{II} \in (0, 1)$
input     : Model class $\Theta \subset \mathbb{R}^d$, input data $\mathbf{X} \subset \mathbb{R}^D$
output    : Model parameters $\theta^* \in \Theta$, inliers $X^* \subset \mathbf{X}$
$\theta^* \leftarrow$ Null, $X^* \leftarrow \emptyset$
for $i = 1 \dots T$ do
    $X_s \leftarrow \text{Rand}(\mathbf{X}, s_\Theta)$                // Draw random subset of $s_\Theta$ data points
    for $\theta \in M_\Theta(X_s)$ do                              // Estimate models from sample
        $I \leftarrow \{X_i \in \mathbf{X} : e(X_i, \theta) \leq \sigma\}$   // Inliers for model $\theta$
        if $|I| > |X^*|$ then
            $\theta^* \leftarrow \theta$, $X^* \leftarrow I$              // Keep maximum consensus model
            $\epsilon \leftarrow |I|/|\mathbf{X}|$                        // Ratio of inliers
            Compute $T(\epsilon)$                                  // Updated number of iterations (4)
            if $T > T(\epsilon)$ then $T \leftarrow T(\epsilon)$
return $(\theta^*, X^*)$
```

For each candidate model θ, its set of inliers is determined using the error to model function e, with a tolerance σ. If it results in a higher consensus than the current one, the latter is updated.

At the end of the procedure, a refined model is usually computed taking into account all inliers X^*. Typically, a least squares minimization is performed

$$\tilde{\theta} = \arg\min_{\theta} \sum_{X \in X^*} e(X, \theta)^2. \tag{3}$$

An optional feature of the procedure consists in an update of its maximum number of iterations T when a better model is found. Initially, T is proportional to the time budget allocated to the algorithm. This can be lowered during the procedure: the procedure stops when it is confident that at least one uncontaminated sample has been drawn. With ϵ the ratio of inliers, the probability of drawing a sample contaminated by outliers is $1 - \epsilon^s$. This event happening for T samples has a probability $(1-\epsilon^s)^T$. Given a confidence level p_{II} (such as 95% or 99%), at least one uncontaminated sample among T was drawn if T is at least

$$T(\epsilon) = \left\lceil \frac{\log(1 - p_{II})}{\log(1 - \epsilon^s)} \right\rceil. \tag{4}$$

The probability $1-p_{II}$ represents a type II error: missing a good model because the number of tested samples is not sufficient. The function $T(\epsilon)$ is decreasing, hence a higher inlier ratio means fewer iterations are needed. The value $p_{II} = 1$ (100% confidence) requires an infinite number of iterations in (4), in which case dynamic adaptation of T is disabled.

2.3 Automatic RANSAC

We call a RANSAC variant "automatic" if the inlier threshold σ is not fixed in advance but is estimated along with the model parameters. In essence, it changes lines 5 to 10 of Algorithm 1. In particular, line 5 is modified to compute an inlier set depending on σ, I_σ. The quality criterion is not anymore the consensus $|I_\sigma|$, since it is always increasing with σ, but is based on statistical grounds. The one presented here is based on the likelihood and proposes two optional features to make the algorithm faster:

1. A bailout test at line 5, which may avoid computing the whole inlier set I_σ, when the probability of beating the current best model is low.

2. An adapted formula to replace (4): since the inlier ratio ϵ depends on σ, the minimal number of iterations must be evaluated with the worst case, that is the lowest σ, which corresponds to the lowest ϵ.

3 Likelihood of Point Correspondences

3.1 Likelihood Computation

For this algorithm, the data is supposed to be drawn from a mix of the inlier distribution f^{in}, in ratio $\rho \in [0, 1]$, and from the outlier distribution f^{out}, in ratio $1 - \rho$. With parameters θ and σ, the inlier distribution is assumed uniform in $I(\theta, \sigma)$. The relative volume of $I(\theta, \sigma)$ w.r.t. the one of P will be called p_σ, omitting the θ-dependency. The value of p_σ is the probability that a background point is in the inlier region. The outlier distribution is assumed uniform on P. We can write the probability density function of X_i as

$$\begin{aligned} X_i &\sim \rho f^{\text{in}} + (1-\rho) f^{\text{out}} \\ &\sim (\rho f^{\text{in}} + (1-\rho) f^{\text{out}}) \mathbf{1}_I + (\rho f^{\text{in}} + (1-\rho) f^{\text{out}}) \mathbf{1}_{P \setminus I(\theta,\sigma)} \\ &\sim (\rho \tfrac{1}{|I|} + (1-\rho) \tfrac{1}{|P|}) \mathbf{1}_I + (\rho \cdot 0 + (1-\rho) \tfrac{1}{|P|}) \mathbf{1}_{P \setminus I(\theta,\sigma)} \\ &\sim \tfrac{1}{|P|} \left(1 - \rho + \tfrac{\rho}{p_\sigma}\right) \mathbf{1}_{I(\theta,\sigma)} + \tfrac{1-\rho}{|P|} \mathbf{1}_{P \setminus I(\theta,\sigma)} \end{aligned} \tag{5}$$

with $|P|$ the volume of P and $|I|$ the volume of I. The right hand side of the sum is due to the fact that outside I only the background distribution applies.

Given these hypotheses the likelihood of $\mathbf{X}$ with k observed inliers is

$$\mathcal{L}(\mathbf{X};\theta,\sigma,\rho)=\frac{1}{|P|^n}\left(1-\rho+\frac{\rho}{p_\sigma}\right)^k(1-\rho)^{n-k}. \tag{6}$$

Notice that $k=k(\theta,\sigma)$, since an inlier is defined by its position in the region $I(\theta,\sigma)$.

Given the observations, the goal is to maximize $\mathcal{L}$ with respect to θ, σ and ρ. The maximisation of the likelihood is done by exhaustive evaluation of $\sigma\in\Sigma$ and the usual RANSAC sampling for θ by drawing minimal sample sets of size s. By contrast, the maximum likelihood estimate with respect to ρ, at fixed σ and θ, can be computed in closed form

$$\hat{\rho}=\max\left\{0,\frac{\epsilon-p_\sigma}{1-p_\sigma}\right\}, \tag{7}$$

with $\epsilon=k/n$, the proportion of inliers (details in Appendix A). The lower bound 0 comes from the constraint that $\rho\in[0,1]$. Its interpretation is that if $\epsilon<p_\sigma$, we have not even observed the expected number of inliers predicted by the pure uniform distribution on P.

Using this, the usual RANSAC quality function of the model parameters θ, its number k of inliers, is replaced by $L(\epsilon,\sigma)=\log\mathcal{L}(\mathbf{X};\theta,\sigma,\hat{\rho})$, or equivalently, discarding constant terms in the optimization

$$L(\epsilon,\sigma)=\begin{cases}\epsilon\log\left(\frac{\epsilon}{p_\sigma}\right)+(1-\epsilon)\log\left(\frac{1-\epsilon}{1-p_\sigma}\right) & \text{if } \epsilon\geq p_\sigma;\\ 0 & \text{otherwise.}\end{cases} \tag{8}$$

The behavior of this function is studied in Appendix B.

3.2 Significance Test of the Likelihood

Given the maximal likelihood, we can check its statistical significance in the following manner. Let us consider two competing hypotheses:

1. $\mathcal{H}_0$: the observed data is fully explained by the uniform distribution over the whole P, i.e., $\rho=0$.
2. $\mathcal{H}_1$: the observed data present some concentration in a volume $I(\theta,\sigma)$, the inliers, hence $\rho>0$.

We can then define the two types of errors. The type I error (false positive) is accepting $\mathcal{H}_1$ while the model may be explained by $\mathcal{H}_0$. The type II error (false negative) is rejecting $\mathcal{H}_1$, while a significant model under $\mathcal{H}_1$ was present.

To compare the likelihood to the null hypothesis, we can use the likelihood ratio test

$$\Lambda(k,\theta,\sigma,\rho)=\frac{\sup_{\theta,\sigma,\rho}\mathcal{L}(\mathbf{X};\theta,\sigma,\rho)}{\mathcal{L}(\mathbf{X};0,0,0)}=\sup_{\theta,\sigma,\rho}\left(1-\rho+\frac{\rho}{p_\sigma}\right)^k(1-\rho)^{n-k}. \tag{9}$$

According to Wilks's theorem, under $\mathcal{H}_0$, $2\log\Lambda(k,\theta,\sigma,\rho)$ is asymptotically of law χ^2 with $d_\Theta+2$ degrees of freedom, where d_Θ is the intrinsic dimension of Θ, that is $d_\Theta=9-1=8$ for homography (sphere of $\mathbb{R}^9$) and $d_\Theta=9-2=7$ for fundamental matrix (sphere intersection with manifold of matrices with null determinant). The other two degrees of freedom (+2 in the equation) come from σ

and ρ. Given this information, the confidence p_I of avoiding a type I error can be reached by setting a minimal value Λ_0 computed as follows

$$\int_0^c \chi^2_{d_\Theta+2}(t)\,dt = p_I. \tag{10}$$

If $2\log\Lambda < c = 2\log\Lambda_0$, the method may choose to ignore the best model as its significance is not high enough, and the $\mathcal{H}_0$ hypothesis may not be rejected. An analogous significance test is present in the competing automatic AC-RANSAC, where the NFA must be lower than 1.

This significance test is considered important in the original article of Cohen and Zach [7], giving its name LRT to their modified RANSAC: p_I is an input to the algorithm, hence c is computed according to (10); as $2\log\Lambda = 2nL$, a lower bound

$$L_{\min} = c/2n \tag{11}$$

is required for L. This bound $L_{\min}$ is then used to estimate the number of required iterations T (detailed in Algorithm 5). We rather consider that the maximum value of T is fixed based on time allocated to the algorithm. By default, we choose $p_I = 0$, hence $c = 0$, meaning we accept any likelihood ratio.

4 Algorithm Description

The LRTSAC algorithm works in the same way as the RANSAC algorithm, with a different quality function and a variable σ. It also introduces an optional bailout strategy when evaluating a model θ in order to accelerate the algorithm. We first present the core algorithm, then the two optional speedup propositions: the early bailout strategy when counting the inliers and the adjustment of the number of sampling iterations.

4.1 Base Algorithm

The procedure of LRTSAC is described in Algorithm 2. Notice that essentially LRTSAC replaces the quality criterion of RANSAC, maximal consensus, by maximal likelihood. The optional early bailout strategy is detailed in Algorithm 4. As in RANSAC, the maximum number of iterations may be dynamically adjusted, see Algorithm 5.

4.2 Minimal Equivalent Inlier Ratios

Applying any of both options of LRTSAC, early bailout when counting inliers and adjustment of the number of iterations, requires computing a minimal equivalent inlier ratio for the current likelihood. This represents, for a given σ, the minimum inlier ratio to reach the likelihood L^*

$$L(\epsilon_{\min}(\sigma), \sigma) = L^*. \tag{12}$$

Since L is an increasing function of its first variable, computing $\epsilon_{\min}(\sigma)$ can be done by bisection of the equation above. It may happen that $L(1, \sigma) < L^*$. As L is only defined when $\epsilon \leq 1$, it means that the target likelihood cannot be reached for this value of σ. When the current best likelihood is L^*, we can then discard σ from the list Σ. Actually, we can even discard all $\sigma' > \sigma$ since

$$\sigma \leq \sigma' \Rightarrow L(1, \sigma) \geq L(1, \sigma'). \tag{13}$$

Algorithm 2: LRTSAC. Blue lines are optional.

```
hyperparameter: $\Sigma$, set of possible values for threshold $\sigma$
parameter      : $T$, allocated number of iterations; confidence probability $p_I$ of avoiding
                 type I error
Input: $\Theta$, the model type; $\mathbf{X}$, the input data
Output: model $\theta^*$, threshold $\sigma^*$
$L^* \leftarrow 0$, $\theta^* \leftarrow$ Null, $\sigma^* \leftarrow 0$
Compute $\epsilon_{\min}(\sigma)$, $\forall\sigma \in \Sigma$ from $L_{\min}$   // Algorithm 3, $L_{\min}$ computed from (10) and (11)
while number of iterations is below $T$ AND $\Sigma$ is not empty do
    $X_s \leftarrow \mathrm{Rand}(\mathbf{X}, s_\Theta)$                          // Get random minimal sample
    for $\theta \in M_\Theta(X_s)$ do
        Compute ratio of inliers $\hat{\epsilon}(\sigma)$, $\forall\sigma \in \Sigma$ with early bailout   // Algorithm 4
        $\hat{\sigma} \leftarrow \arg\max_\sigma L(\hat{\epsilon}(\sigma), \sigma)$, $\hat{L} \leftarrow L(\hat{\epsilon}(\hat{\sigma}), \hat{\sigma})$
        if $L^* < \hat{L}$ then
            $L^* \leftarrow \hat{L}$, $\theta^* \leftarrow \theta$, $\sigma^* \leftarrow \hat{\sigma}$
            Compute $\epsilon_{\min}(\sigma)$, $\forall\sigma \in \Sigma$ from $L^*$          // Algorithm 3
            Update $T$ from $\epsilon_{\min}(\sigma)$                                      // Algorithm 5
return $(\theta^*, \sigma^*)$                                  // Model parameters and threshold
```

This has the effect of possibly reducing $\sigma_{\max}$ when an improved likelihood is found. The procedure follows Algorithm 3, which may be called whenever L^* is improved in Algorithm 2. Notice that

$$L(1, \sigma) = -\log p_\sigma \tag{14}$$

can be tabulated.

Algorithm 3: Compute equivalent inlier ratios and reduce Σ

```
Input: $L^*$, target likelihood; $\Sigma$, set of possible values for threshold $\sigma$
Output: Minimal equivalent inlier ratios $\epsilon_{\min}(\sigma)$, $\forall\sigma \in \Sigma$; reduced set $\Sigma$
for $\sigma \in \Sigma$ do
    if $L(1, \sigma) < L^*$ then
        Discard $\sigma$ from $\Sigma$
    else
        Compute $\epsilon_{\min}(\sigma)$, from a bisection of (12)
```

4.3 Counting Inliers With Early Bailout

When counting inliers, LRTSAC proposes an early bailout strategy while evaluating a model. If according to this strategy the given model would not give enough inliers to improve the likelihood when the count is finished, it is discarded immediately.

Let $(Z_i)_{i\in[m]}$ be i.i.d. random variables that represent whether data point i is an inlier with $Z_i = 1$, or an outlier with $Z_i = 0$, at given noise level σ and model θ. Let m be the number of data points counted so far. The Hoeffding inequality evaluates the probability of deviation of the mean $\overline{Z}$ of the Z_i from their expectation

$$\mathbf{P}(\overline{Z} \le \mathbf{E}(\overline{Z}) - t) \le e^{-2t^2 m}. \tag{15}$$

In this formula, we can take $\overline{Z}$ as $\hat{\epsilon}_m(\sigma)$ the observed inlier ratio after m data points. This means that the probability of the event $\hat{\epsilon}_m(\sigma) \leq \mathbf{E}(\hat{\epsilon}_m(\sigma)) - t$ decreases to 0 exponentially fast. If we want $\mathbf{E}(\hat{\epsilon}_m(\sigma)) \geq \epsilon_{\min}(\sigma)$, the probability of $\hat{\epsilon}_m(\sigma) \leq \epsilon_{\min}(\sigma) - t$ decreases to 0 just as fast.

Thus, when we observe $\forall\sigma,\ \hat{\epsilon}_m(\sigma) \leq \epsilon_{\min}(\sigma) - t$ $(t > 0)$ and we discard the model, estimating that the chances of reaching $\epsilon_{\min}(\sigma)$ for any σ are low, we bailout too early with a probability lower than $\beta = e^{-2t^2 m}$. The probability β may be specified; we can then introduce $\tau_m = \sqrt{-\frac{\log\beta}{2m}}$ and we have a bound of the probability of premature bailout

$$\mathbf{P}(\forall\sigma, \hat{\epsilon}_m(\sigma) \leq \epsilon_{\min}(\sigma) - \tau_m | \mathbf{E}(\hat{\epsilon}_m(\sigma)) \geq \epsilon_{\min}(\sigma)) \leq \beta. \tag{16}$$

This test entails a computational cost so it is checked every B data point only, where B is chosen to balance the cost of the test and the cost of the model evaluation. Thus we perform $\lfloor n/B \rfloor$ bailout tests. The user may specify a confidence probability p'_{II} of avoiding a premature bailout. If we replace β in (16) by $(1 - p'_{II})/\lfloor n/B \rfloor$, the probability of a type II error due to premature bailout is bounded by $1 - p'_{II}$. This gives the new expression of τ_m

$$\tau_m = \sqrt{-\frac{\log(1 - p'_{II}) - \log\left\lfloor \frac{n}{B} \right\rfloor}{2m}}. \tag{17}$$

Algorithm 4 sums up the process. Notice that to disable the early bailout and not risk increasing the type II error, it is enough to take $B > n$. Equivalently, we consider that the user-defined parameter value $p'_{II} = 1$ means that no bailout is applied.

Algorithm 4: Compute inlier ratio $\hat{\epsilon}(\sigma)$, $\forall\sigma \in \Sigma$ with early bailout

```
hyperparameter: Set of possible thresholds $\Sigma$; bailout test periodicity $B$
parameter     : Probability of avoiding premature bailout $p'_{II}$
Input: data $\mathbf{X}$; model to evaluate $\theta$; $\{\epsilon_{\min}(\sigma) : \sigma \in \Sigma\}$, minimal required inlier ratios
Output: $\{\hat{\epsilon}(\sigma) : \sigma \in \Sigma\}$
1  $\hat{\epsilon}(\sigma) \leftarrow 0,\ \forall\sigma \in \Sigma$
2  for $X_i \in \mathbf{X}$ do
3      Compute $e(X_i, \theta)$
4      for $\sigma \in \Sigma$ do
5          if $e(X_i, \theta) \leq \sigma$ then
6              $\hat{\epsilon}(\sigma) \leftarrow \hat{\epsilon}(\sigma) + \frac{1}{n}$
7      if $i \equiv 0 \pmod{B}$ then                          // Early bailout
8          Compute $\tau_i$                                    // Apply (17)
9          for $\sigma \in \Sigma$ do
10             if $\hat{\epsilon}(\sigma) \times n/i \geq \epsilon_{\min}(\sigma) - \tau_i$ then
11                 Go to next point                            // Back to line 2
12         Bailout                               // Back to Algorithm 2, line 5
13 return $\{\hat{\epsilon}(\sigma) : \sigma \in \Sigma\}$
```

4.4 Number of Sampling Iterations

Ignoring first the bailout strategy, the number of iterations of LRTSAC at a given σ, $T(\sigma)$, follows the classic RANSAC iteration formula (4), but using $\epsilon_{\min}(\sigma)$, the minimal inlier ratio to improve the current likelihood L with the threshold σ, instead of $\hat{\epsilon}(\sigma)$, the current observed inlier ratio at σ.

Indeed, we have $\epsilon_{\min}(\sigma) \geq \hat{\epsilon}(\sigma)$ with equality if σ is the current σ^*. Therefore, applying (4) and the decreasing trend of T, we get $T(\epsilon_{\min}(\sigma)) \leq T(\hat{\epsilon}(\sigma))$ and we define

$$T(\sigma) = \left\lceil \frac{\log(1 - p_{II})}{\log(1 - \epsilon_{\min}(\sigma)^s)} \right\rceil, \tag{18}$$

with p_{II} the confidence after $T(\sigma)$ iterations of not committing type II error at noise level σ.

We can now introduce T, the number of iterations of LRTSAC, that ensures that the probability of drawing at least one good model is larger than p_{II}. To ensure that, T just needs to be larger than $T(\sigma_{\min})$ as $T(\sigma)$ is monotonically decreasing with respect to σ, since $\epsilon_{\min}(\sigma)$ is monotonically increasing with respect to σ.

When the bailout strategy is used, a model estimated from an uncontaminated sample still has a probability $1 - p'_{II}$ of being discarded. Therefore, the probability of drawing an uncontaminated model *and* escaping the bailout filter becomes $p'_{II}\,\epsilon_{\min}(\sigma)^s$, and this modifies the formula for $T(\sigma)$

$$T(\sigma) = \left\lceil \frac{\log(1 - p_{II})}{\log(1 - p'_{II}\,\epsilon_{\min}(\sigma)^s)} \right\rceil. \tag{19}$$

The update of T is simply applying (19) to $\sigma_{\min}$, see Algorithm 5.

Algorithm 5: Update number of iterations T while controlling type II error

```
parameters: Confidence probabilities against type II error: $p_{II}$, confidence that no valid
            model was missed because of lack of iterations; $p'_{II}$, probability of a valid model
            avoiding premature bailout
input     : $\Sigma$, set of possible values for threshold $\sigma$
output    : $T$, number of iterations
return $T(\sigma_{\min})$                                            // Apply (19)
```

4.5 Implementation and Hyperparameters Values

This section describes the implementation details and the choices made for the hyperparameters of LRTSAC.

4.5.1 Hyperparameters

The threshold set Σ is defined as follows

$$\Sigma = \begin{cases} \{\sigma = \sigma_{\min} * \sigma_{\text{step}}^k;\ \forall k \in \mathbb{N},\ \sigma \leq \sigma_{\max}\} & \text{if } \sigma_{\min} \leq \sigma_{\max}; \\ \{\sigma_{\max}\} & \text{otherwise.} \end{cases} \tag{20}$$

with $\sigma_{\min} = 0.25$ pixel, $\sigma_{\text{step}} = \sqrt{2}$ and $\sigma_{\max}$ is user defined (with a default value of 16). This implementation gives a good spread of inlier/outlier thresholds with a moderate amount of thresholds (at most 13).

The relative area of the inlier region p_σ depends on the model space. For a $1D$ error (point to line distance), like for fundamental or essential matrix estimation, $p_\sigma = 2\sigma\frac{Diam}{Area}$ where $Diam$ is the diameter and $Area$ the area of the image. For a $2D$ error (point to point distance), like for homography estimation, $p_\sigma = \frac{\pi\sigma^2}{Area}$. Such upper bounds are the same as in the method AC-RANSAC.

The parameter B of the early bailout strategy is set to balance bailout evaluation time and error computation time. We use the value proposed in [7], that is $B = 100$: for every new batch of 100 data points, the bailout test is performed when counting inliers.

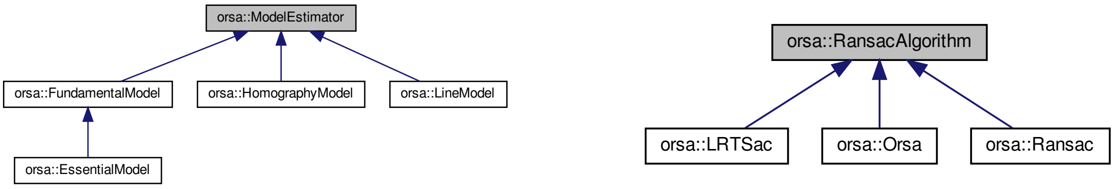


Figure 1: Object inheritance in `libOrsa`. Two abstract base classes `ModelEstimator` and `RansacAlgorithm` allow to mix and match at will.

When using a controlled type I error to verify the quality of the given model as presented in section 3.2, the parameter c introduced in Equation (10) is extracted from a table given user defined p_I. Because of this procedure, the current implementation only allows $p_I = 0.90, 0.91 \dots 0.99$ (and default value $p_I = 0$ disabling any control on type I error).

Default values for other parameters are: $p_{II} = 0.99$ (confidence relatively to the number of iterations) and $p'_{II} = 0.95$ (confidence relatively to the bailout test). The bisection used in Algorithm 3 stops when the width of the search interval is below $1/n$, the granularity of ϵ. This requires $\log_2 n$ iterations, 10 for a typical value of $n = 1000$ data points.

All RANSAC algorithms presented in the experiments Section 5 used the same data, same solvers to estimate models from minimal samples and same refinement steps, only the RANSAC algorithm varied.

The maximum iteration limit T can be tuned by an option `maxIter` when calling the code, but fixed at the default 50 000 in the demonstration system.

4.5.2 Code

The C++ code of LRTSAC is integrated into the `libOrsa` library of [11] and [12]. A modular inheritance structure is used to easily integrate different model classes and RANSAC type algorithms, see Figure 1. On the one hand, the abstract class `ModelEstimator` keeps the data and has virtual functions to fit points to subsets of the data and compute errors. On the other hand, the class `RansacAlgorithm` has a virtual method `run` implemented in the subclasses `LRTSac`, `Orsa` (a.k.a. AC-RANSAC) and regular `Ransac`. The function returns the estimated model parameters and selected σ (the same as input for RANSAC), along with some run statistics. Such a decorrelation between models and algorithms allows to mix at will and extend to different models or algorithms.

4.5.3 Demo

The demo proposes only homography and fundamental matrix estimation, since essential matrix requires the input of calibration matrices, which are not readily available for most image pairs. The usual RANSAC, AC-RANSAC and LRTSAC are proposed. The latter proposes three tunable confidence probabilities p_I, p_{II} and p'_{II}. The result is the mosaic for homography and points and parts of their epipolar line for fundamental matrix.

5 Experimental Evaluation of LRTSAC

5.1 The Test Data

The LRTSAC algorithm is tested by evaluating the estimated threshold σ^* compared to the true noise level σ_{noise} of semi-synthetic data and by qualitative observation of real images. All options proposed in Algorithms 3, 4 and 5 are evaluated in runtime efficiency on the USAC dataset [14] to ensure they do reduce runtime.

The early bailout strategy described in Algorithm 4 is also evaluated in terms of precision and recall on semi-synthetic data to determine its impact on the classification power of LRTSAC.

The USAC dataset consists of:

1. 10 image pairs for homography registration, which we name $H1$ to $H10$;
2. 11 image pairs for fundamental matrix, we name them $F1$ to $F11$;
3. 6 images pairs for essential matrix (hence the camera calibration parameters are given), named $E1$ to $E6$;
4. For each image pair, a set of pair correspondences, ranging from several hundred to a few thousand, without any ground truth about their status of inlier or outlier.

The semi-synthetic data used to measure precision and recall is based on the images and the input correspondences of the USAC [14] dataset. As no ground truth is associated, we build an artificial but realistic one [16]: from the correspondences, a model is estimated with AC-RANSAC [11, 12] and used to create realistic noisy ground truth inliers with a noise level σ_{noise}. Before the noise addition, perfect matches are produced by projection on the evaluated model. Finally, outliers are generated to control the inlier ratio. They follow the background model of a uniform distribution, that is, each point of a pair is drawn randomly in a uniform distribution in its respective image.

For all the experiments, the parameters and hyperparameters are set to their default value as presented in Section 4.5.

5.2 LRTSAC Performance

RANSAC algorithms are used to filter outliers and estimate a model on inliers only. As Figure 2 illustrates, using the estimated model to compare quality of the algorithm is not pertinent: Figure 2 presents the mosaic reconstruction of a homography on a difficult semi-artificial dataset. Using the protocol described in Section 5.1, from pair $H1$ of USAC [14], an artificial dataset is generated with an inlier noise of $\sigma_{\text{noise}} = 3$ pixels and an outlier ratio of 0.9. These parameters produce a very complex dataset to handle but a model can still be estimated and so, it is difficult to visually assess which algorithm did best to estimate an inlier set and a good model. This is why Figures 3 and 4 present the inliers matches on the images. As the inlier set is for some usage as useful as the estimated model itself, this allows a better analysis of the classification power of the algorithm in terms of inlier/outlier.

Figure 3 shows the results of LRTSAC and RANSAC with user-threshold $\sigma = 3$ on an artificial dataset created from $H3$ with $\sigma_{\text{noise}} = 2$ pixels and outlier ratio 0.7. On this dataset, RANSAC performs relatively well, recovering 23% of the inliers and producing a satisfactory reconstruction and LRTSAC recovers 54% of the inliers with a good reconstruction as well. Here both algorithms perform well, in similar runtimes (around 0.55s each).

Figure 4 shows the results of LRTSAC and RANSAC with user-threshold $\sigma = 3$ on an artificial dataset created from $E4$ with $\sigma_{\text{noise}} = 1$ pixel and outlier ratio 0.5. On this dataset, RANSAC

Figure 2: Mosaic reconstruction of dataset $H1$ with $\sigma_{\text{noise}} = 3$ and an outlier ratio of 0.9. This setting represents a difficult dataset for RANSAC. From left to right, the results of LRTSAC, RANSAC with $\sigma = 3$ and RANSAC with $\sigma = 9$ are presented.

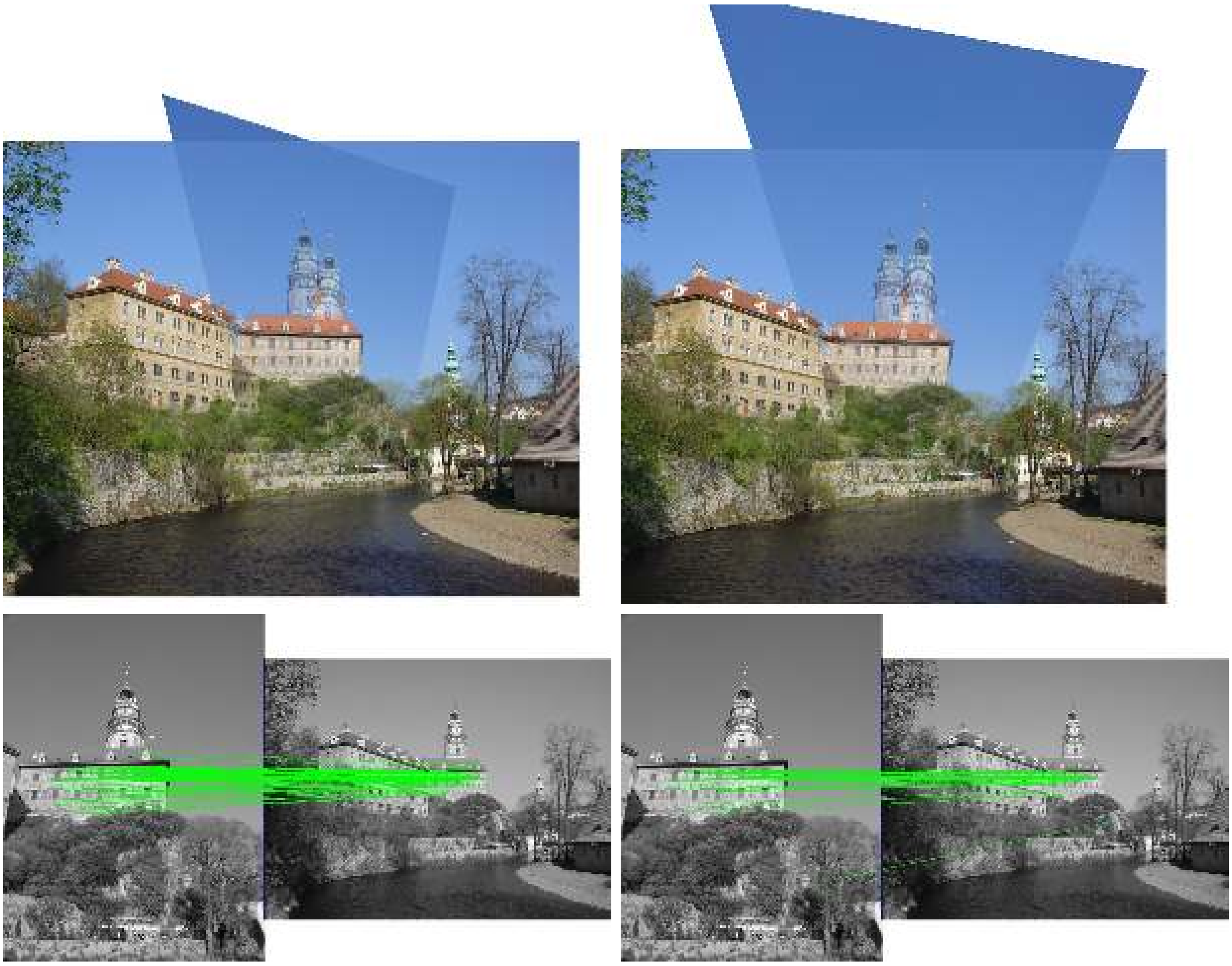


Figure 3: Results on dataset $H3$ with $\sigma_{\text{noise}} = 2$ and an outlier ratio of 0.7. Top line: Mosaic reconstruction, bottom line: inliers matches representation. The left column is LRTSAC, the right one is RANSAC with $\sigma = 3$. In this setup LRTSAC obtains a precision of 0.54 and a recall of 0.54 while RANSAC obtains a precision of 0.55 and a recall of 0.23. For this dataset, the plane of the homography is the facade of the middle building, this is why the out of plane tower is mismatched.

performs badly, recovering only 2% of the inliers while LRTSAC recovers 47% of the inliers. This example shows the adaptability of LRTSAC which, without any change in parameter settings, can still recover a high percentage of inliers when RANSAC fails to do so.

Table 1 shows the evolution of the estimated parameters of LRTSAC over some datasets from USAC dataset [14]: $H1$, $H2$, $F7$ and $E1$. Each line corresponds to a new set of best-so-far parameters and at what iteration it was found. As the algorithm that estimates models from a minimal set of

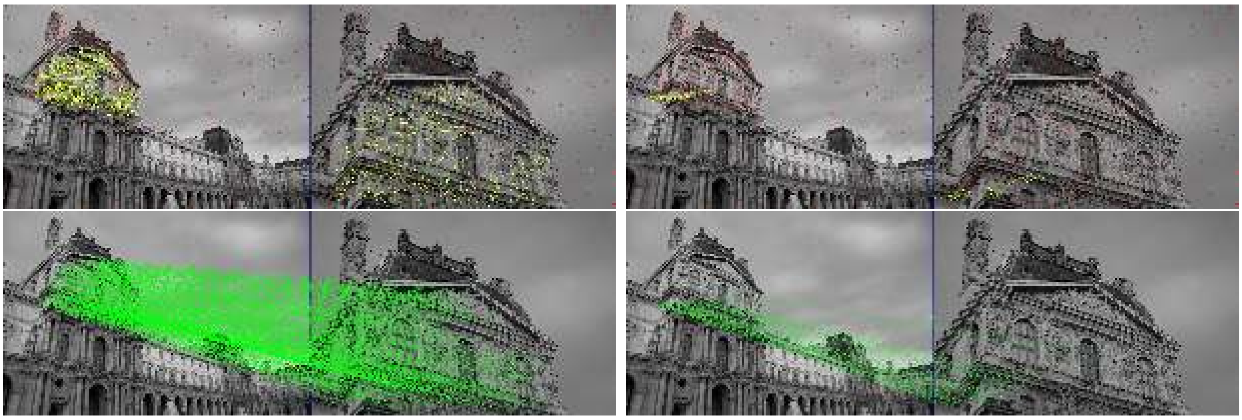


Figure 4: Results on dataset $E4$ with $\sigma_{\text{noise}} = 1$ and an outlier ratio of 0.5. Top: distance to epipolar lines of inliers (yellow lines with green dots) and outliers (red dots). Bottom: inlier matches representation. The left column is LRTSAC, the right one is RANSAC with $\sigma = 3$. In this setup LRTSAC obtains a precision of 0.50 and a recall of 0.47 while RANSAC obtains a precision of 0.34 and a recall of 0.02.

points can return multiple models, more than one best-so-far model can be found at the same iteration. Note also that the first valid model found does not necessarily happen at the first iteration as the early bailout strategy can refuse it before evaluating its likelihood. For all runs, the algorithm was set with default parameters and was allowed at most 100 000 iterations, so that the iteration cutoff happened because of the required number of sampling iterations T and not the allocated resource limit. First, we can note that the algorithm can have different orders of magnitudes for T and L^*—the likelihood of the best-so-far model—across different runs. However, the end values of L^* are always way higher than the values of the parameter L_{min} introduced in Section 3.2 which is around 0.001. That is why this implementation makes the type I error of the ratio test an optional parameter, as it has rarely an impact over the end result. We can also notice that the number of elements removed from Σ by Algorithm 3 is quite small: in this setting, the initial set Σ contains 13 different values for σ. This explains the results of Figures 7, 8 and 9 when comparing LRTSAC without any option and LRTSAC with only Algorithm 3. Still, as this procedure is necessary to compute the minimum inlier ratio to increase the likelihood, $\epsilon_{\min}(\sigma)$, we cannot remove it.

5.3 Estimated Threshold σ^* Compared to True Noise Level σ_{noise}

Figure 5 presents for different datasets and different settings the evolution of the ratio of the estimated inlier/outlier threshold σ^* over the true noise level σ_{noise}, as a function of the latter. Results are averaged for each noise level over 25 runs of LRTSAC.

For all tested configurations this ratio is high for small noise levels and stabilizes between 2 and 4 as soon as the noise level reaches 1 pixel. This means the estimated noise level is between 2 and 4 times higher than the true noise level. A numerical analysis of the behavior of the likelihood can show that increasing σ will always increase $\mathcal{L}(\hat{\epsilon}(\sigma), \sigma)$ if at least about 100 new inliers are gained. Usually, when a good model is found an increase of σ yields between 200 and 400 new inliers. Line 7 of Algorithm 2 thus yields higher thresholds $\hat{\sigma}$ than the true noise level.

It may be regarded as a disappointment that the estimated σ^* is always higher than the true one. This is still to be expected, since the model is evaluated from noisy inliers. However, as presented in Section 5.2 this does not impact the quality of the result.

Image pair $H1$					Image pair $F7$				
Iteration	T	L^*	σ^*	$\|\Sigma\|$	Iteration	T	L^*	σ^*	$\|\Sigma\|$
12	5032805	0.2998	11.3137	12	13	2108656	0.6433	11.3137	12
14	73941	0.9520	11.3137	12	57	262098	0.9318	4	12
19	346	4.1072	5.65685	12	14552	32520	1.3471	1.41421	12
40	236	4.5597	4	12					

Image pair $H2$					Image pair $E1$				
Iteration	T	L^*	σ^*	$\|\Sigma\|$	Iteration	T	L^*	σ^*	$\|\Sigma\|$
4	362040781	0.0702	11.3137	12	0	2982	1.4892	11.3137	12
90	321678580	0.0725	11.3137	12	1	343	2.4975	4	12
123	6592569	0.2113	11.3137	12	1	56	3.8547	5.65685	11
128	1301458	0.3294	11.3137	12	2	43	4.0902	2.82843	11
232	9311	1.2660	1.41421	12	6	31	4.4084	2	10
					16	31	4.4373	2	10

Table 1: Evolution of the number of iterations T, the best-so-far likelihood L^* and threshold σ^* and the size of the set of thresholds Σ during a run of LRTSAC over different image pairs from the USAC dataset. The iteration column indicates at which sampling iteration the best-so-far model was found. As the algorithm that estimates models from a sample can return multiple models per sample, two good models can be found at the same iteration.

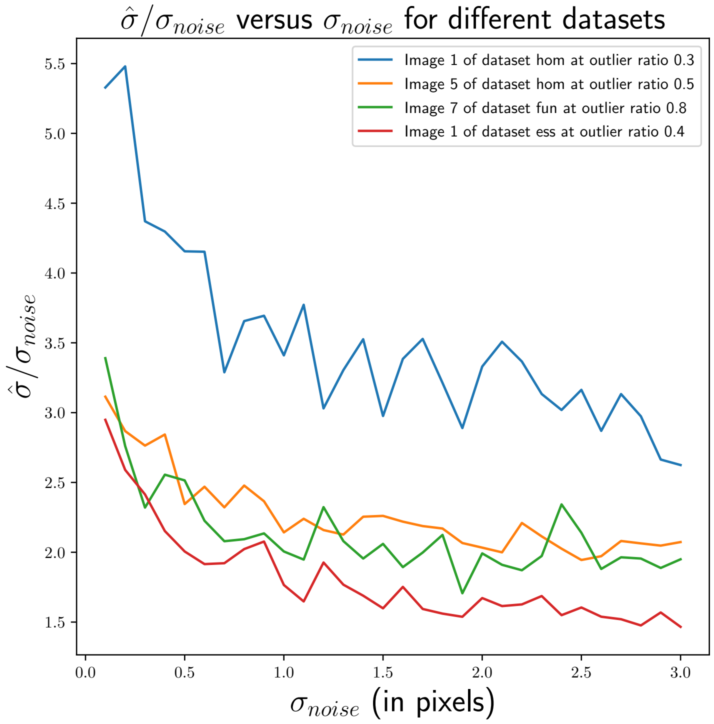


Figure 5: Ratio between estimated inlier/outlier threshold σ^* and true noise level σ_{noise} with respect to σ_{noise}. These results are measured on semi-artificial data where noise level and outlier ratio is controlled. Images numbers follow the USAC dataset [14] numbering and `hom` corresponds to homography estimation, `fun` to fundamental matrix estimation and `ess` to essential matrix estimation.

5.4 Runtime Impact of the Different Options

Figure 6 presents the runtime of LRTSAC with and without bailout and the number of verifications per model (VPM) which is the mean number of times the residual function $e(X, \theta)$ is called. Both metrics are presented as ratios: for the runtime it is the ratio of LRTSAC runtime with early bailout and without bailout, and the VPM is divided by the total number of points of the dataset—which corresponds to the ratio of the VPM with early bailout and without bailout. Results are averaged over 100 runs for each dataset.

The runtime ratio is not always below 1 so the early bailout does not always improve the runtime efficiency of LRTSAC. However, the increase in runtime stays contained below 50% and the early bailout often reduces runtime. The VPM ratio is usually correlated to the runtime ratio but some particular cases appear where VPM ratio is low whereas runtime ratio is high. They could mean that the early bailout strategy often refuses good models for those datasets and thus increases runtime. As the early bailout strategy can impact the number of iterations and the rest of the execution, even in standard cases, it is expected that the relationship between runtime and VPM is not linear. We can thus conclude that the early bailout strategy does not impact negatively the runtime in most cases and when it does have a negative impact, it is small whereas it can cut runtime by 80%.

Figure 7, 8 and 9 present the runtime of LRTSAC with different options enabled as a ratio of the runtime of LRTSAC without any options. In each case, LRTSAC was not allowed to go over 20000 iterations. This limit is set above the usual end value of the computed number of sampling iterations T but some runs might have been stopped by this value. As can be expected, the biggest runtime impact comes from Algorithm 5 which stops the computation when it has confidence that at least one uncontaminated sample was drawn—see blue *versus* orange bars or green *versus* red bars. As discussed previously, the early bailout strategy has an overall positive impact on runtime, rarely increasing it and being able to reduce it down to 20%—see blue *versus* green bars or orange *versus* red bars. Finally, the impact of the elimination of elements of the set of thresholds Σ is minimal but as this is more a side benefit from the computation of the minimum inlier ratio to increase the likelihood, $\epsilon_{\min}(\sigma)$ by Algorithm 3 it is to be expected.

Now that we have observed that the desired outcome of the early bailout strategy is happening, that is, a general reduction in runtime, we need to make sure it does not impact too much the quality of the results.

5.5 Quality Impact of the Early Bailout Strategy

Figure 10 presents the ratio of the precision of LRTSAC with early bailout and of LRTSAC without bailout. Figure 11 presents the ratio of the recall of LRTSAC with early bailout and of LRTSAC without bailout. Both graphs are presented as boxplots with first and third quartiles defining the box and the median for the middle line. The whiskers are set at 1.5 times the interquartile range. The results shown are averaged over different noise levels σ_{noise}—from 0 to 3 pixels by 0.1 pixel increments, different outlier ratios—from 0 to 90% by increments of 10%, and 25 different runs.

Both metrics present a median slightly lower than 1 with a spread on a very small area around this value, sometimes above 1. This means the early bailout strategy is not consistently worsening the quality of the result, and not by a lot when it does, thus it is a valid improvement of the algorithm.

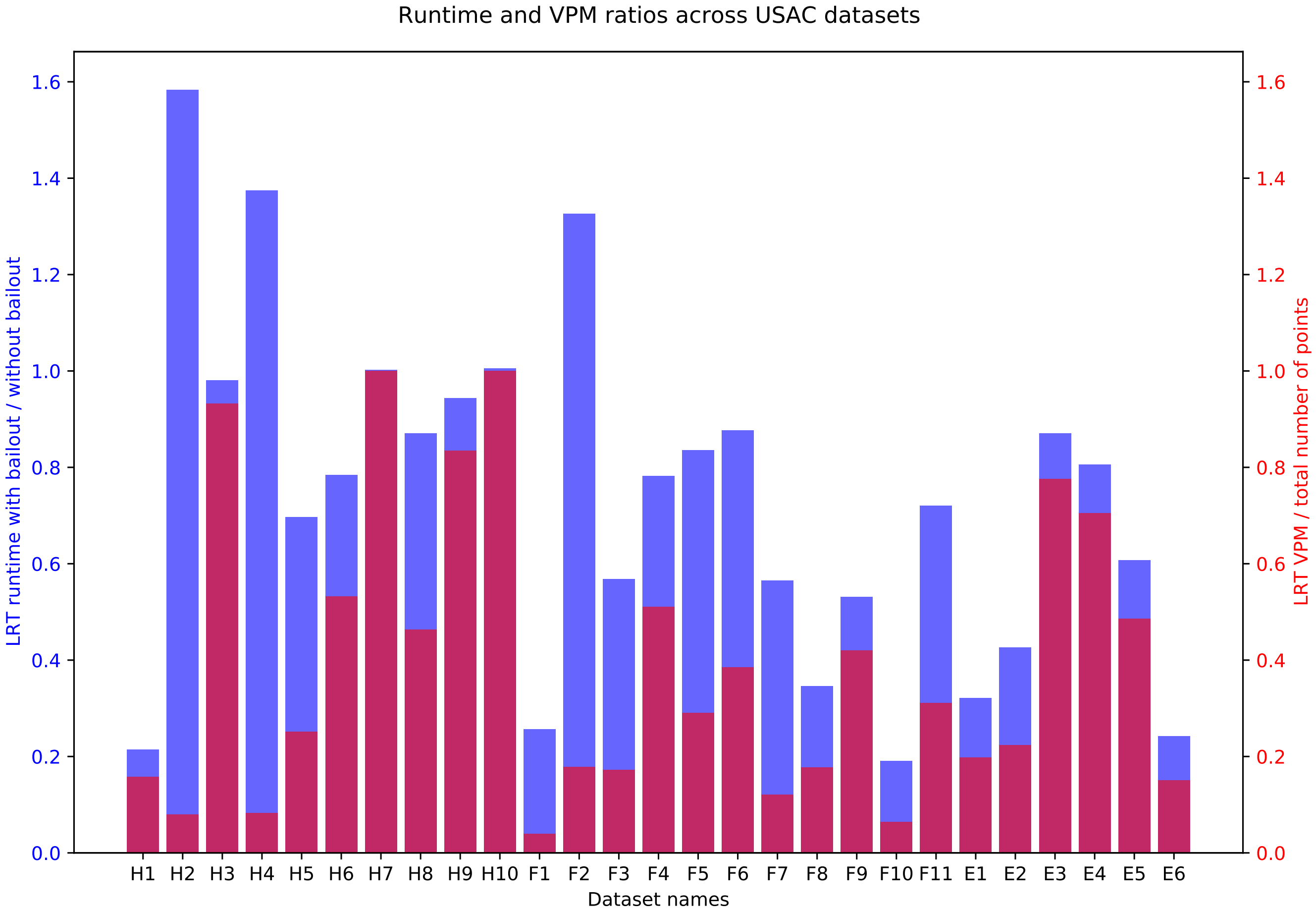


Figure 6: Ratios of LRTSAC Runtime and Verification Per Model (VPM) with the early bailout strategy and without the bailout. Results are presented across all available USAC [14] datasets where `H` stands for homography estimation, `F` for fundamental matrix estimation and `E` for essential matrix estimation with same numbering as USAC.

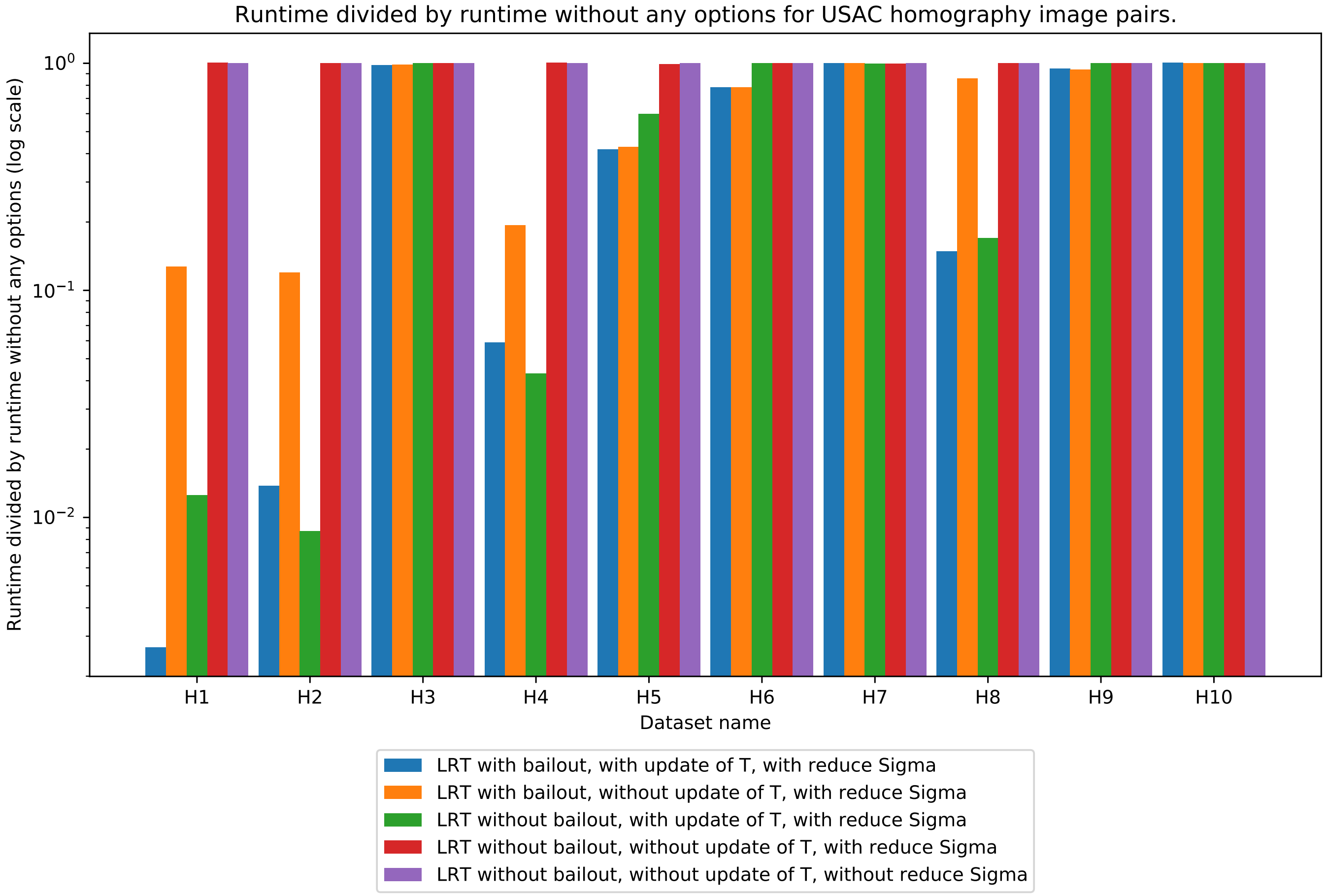


Figure 7: Ratios of LRTSAC Runtime with different options enabled over LRTSAC runtime without any options. Results are presented across all available USAC [14] datasets for homography estimation. The ratio is presented in logscale.

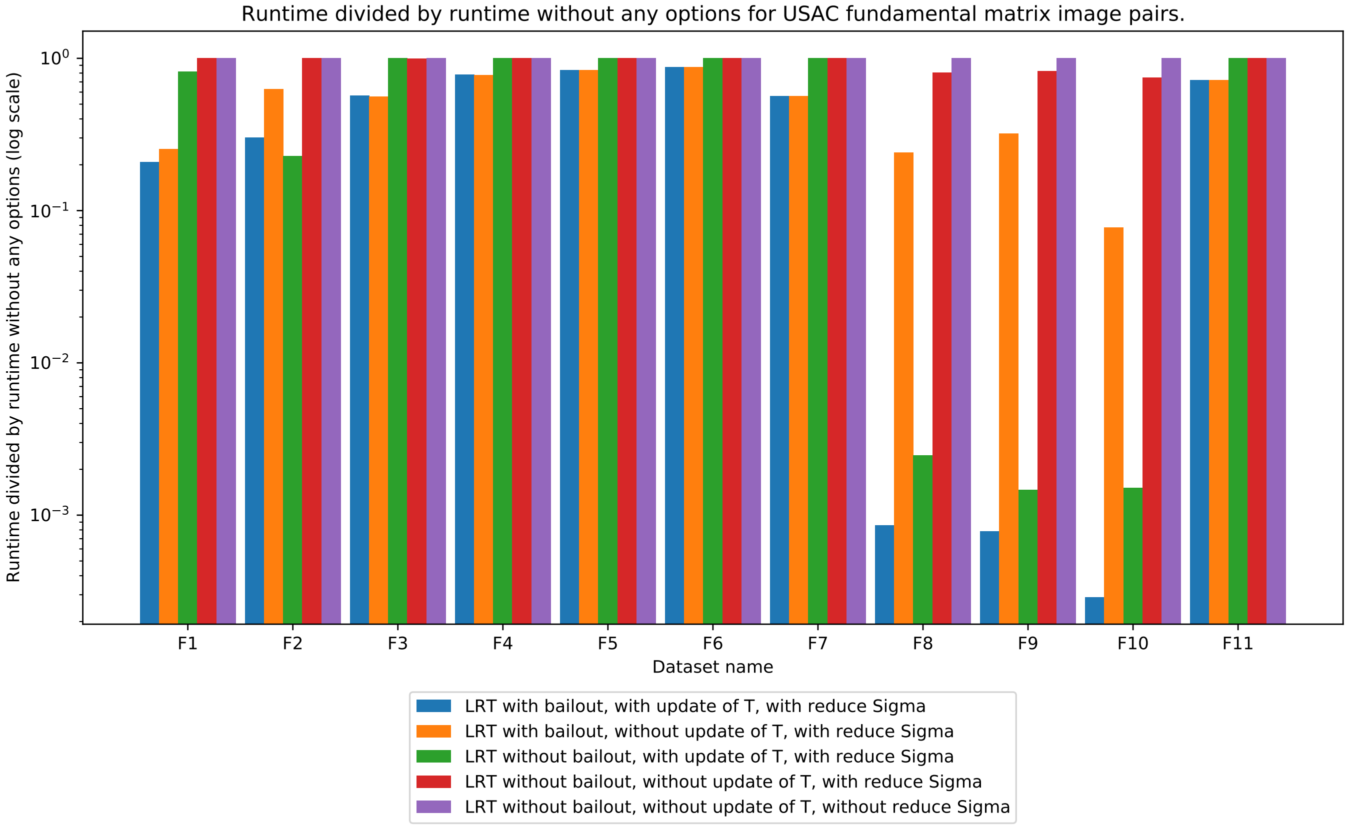


Figure 8: Ratios of LRTSAC Runtime with different options enabled over LRTSAC runtime without any options. Results are presented across all available USAC [14] datasets for fundamental matrix estimation. The ratio is presented in logscale.

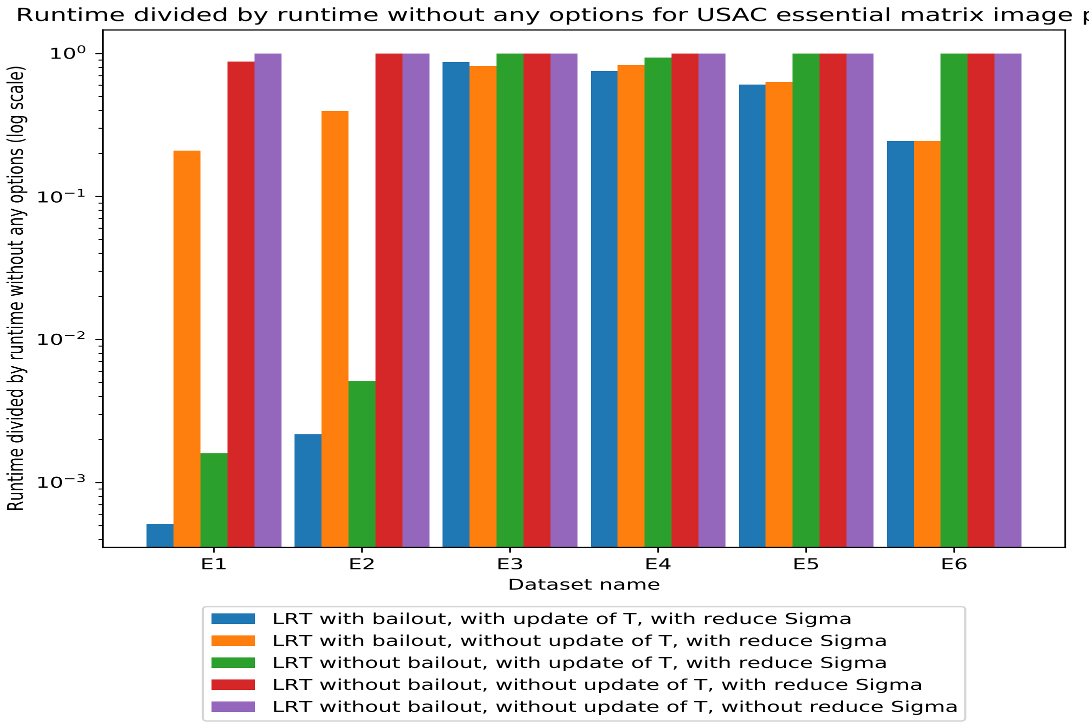


Figure 9: Ratios of LRTSAC Runtime with different options enabled over LRTSAC runtime without any options. Results are presented across all available USAC [14] datasets for essential matrix estimation. The ratio is presented in logscale.

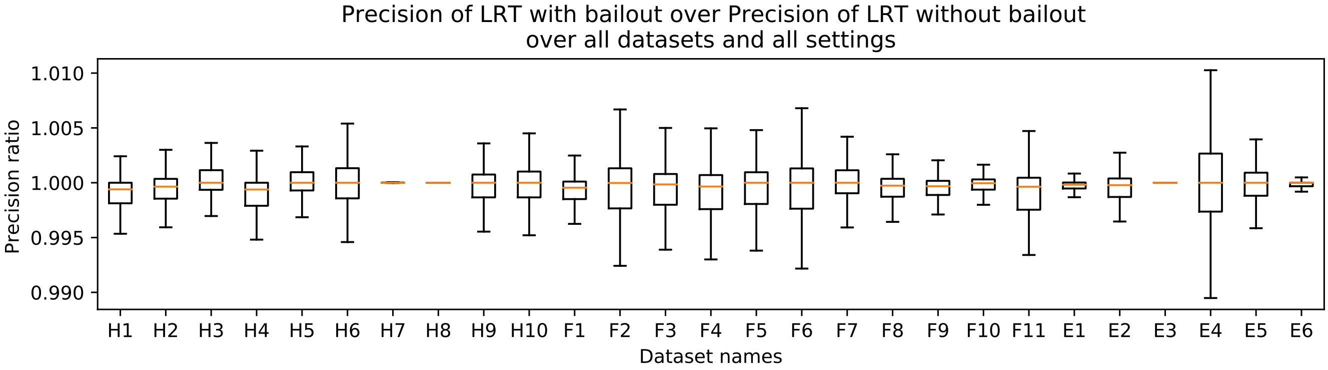


Figure 10: Boxplot of the ratio of the precision of LRTSAC with the early bailout strategy and without the bailout. Results are presented across all available USAC [14] datasets and over a high range of semi-artificial dataset settings. `H` stands for homography estimation, `F` for fundamental matrix estimation and `E` for essential matrix estimation with same numbering as USAC.

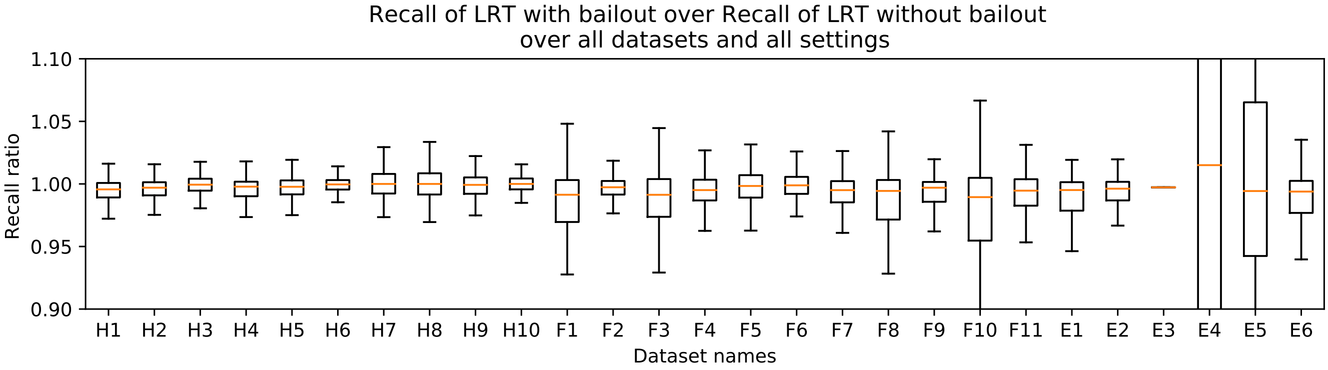


Figure 11: Boxplot of the ratio of the recall of LRTSAC with the early bailout strategy and without the bailout. Results are presented across all available USAC [14] datasets and over a high range of semi-artificial dataset settings. `H` stands for homography estimation, `F` for fundamental matrix estimation and `E` for essential matrix estimation with same numbering as USAC. Dataset $E4$ and $E4$ are cut as they spread much more than others and made them hard to read. For $E4$ the quartiles are $[0.89, 1.15]$ and the whiskers are $[0.51, 1.56]$. For $E5$ the quartiles are $[0.94, 1.07]$ and the whiskers are $[0.78, 1.25]$.

6 Conclusion

We presented a method using the likelihood as the criterion to select automatically the inlier threshold with the parameters of the model. Modulo this change, the algorithm is almost identical to the classical RANSAC. Two optional features accelerating the algorithm, early bailout in model verification and dynamic adjustment of the number of iterations, require adaptations to face the presence of multiple possible thresholds but are shown to be efficient. Compared to AC-RANSAC, the other automatic RANSAC algorithm, LRTSAC is faster because of these options and the limited number of tested thresholds. However, they may be adapted to the criterion of AC-RANSAC without too much difficulty. Each algorithm has its own criterion to decide if an estimated model is statistically significant. In the case of AC-RANSAC, the number of false alarms (expected number of type I errors) must be below 1, whereas in LRTSAC the likelihood ratio must reach a user-defined confidence probability. The normal behavior of AC-RANSAC is to reserve remaining iterations to models estimated by samples among the *inliers* of the valid model, which allows refining it. LRTSAC could adopt the same strategy and may present better results with such a modification.

It has been shown in a study [13] that replacing *bona fide* RANSAC by AC-RANSAC in a standard multi-view stereo pipeline yields improved accuracy, whatever the (fixed) RANSAC threshold used. It remains to be checked that such behavior is reproduced with LRTSAC. This requires appending the case of pose estimation from 2D-3D correspondences, the Perspective from n Points (PnP) problem, to the models estimated by LRTSAC. Finally, another RANSAC method without a fixed threshold, the marginalized sample consensus, MAGSAC [1] and its extension MAGSAC++ [2], could be considered as automatic as they do not require a fixed threshold. A comparison to this algorithm is available in [16].

A Optimal Mixture Parameter

We give details about the computation of the optimal mixture parameter $\hat{\rho}$. From (6), the case $k = 0$ can be easily treated as $\mathcal{L}$ is a decreasing function of ρ so the maximum is at $\hat{\rho} = 0$ and (7) stands, since $\epsilon = 0$ so that the right hand of the maximum is negative. If $k = n$, $\mathcal{L}$ is an increasing function of ρ and the maximum is at $\hat{\rho} = 1$, which coincides with the right hand of the maximum in (7). If $0 < k < n$, then

$$\frac{|P|^n}{n}\frac{\partial \mathcal{L}}{\partial \rho} = (1-\rho+\frac{\rho}{p_\sigma})^{k-1}(1-\rho)^{n-k-1}\left(p'_\sigma\epsilon(1-\rho)-(1-\epsilon)(1+p'_\sigma\rho)\right), \tag{21}$$

with $p' = 1/p_\sigma - 1 \geq 0$. The sign is given by the rightmost factor, which can be simplified as the affine function of ρ

$$-p'_\sigma\rho + \frac{\epsilon - p_\sigma}{p_\sigma}. \tag{22}$$

This term is positive when $\rho \leq \frac{\epsilon-p_\sigma}{1-p_\sigma}$ and negative after, hence (7). The maximum comes from the fact that $\rho \geq 0$.

B Behavior of the Log-Likelihood

We study the behavior of the bivariate function L of (8). Considering ϵ as a continuous variable, we get

$$\frac{\partial L}{\partial \epsilon} = \log\frac{\epsilon}{p_\sigma}\frac{1-p_\sigma}{1-\epsilon}, \tag{23}$$

and both fractions inside the logarithm are greater than 1 when $1 > \epsilon \geq p_\sigma$. Function L is thus an increasing function of ϵ: the likelihood increases with the number of inliers, as expected.

Since p_σ is increasing with σ, let us study the function

$$f(x) = -\epsilon \log(x) - (1-\epsilon)\log(1-x). \tag{24}$$

Its derivative is

$$f'(x) = -\frac{\epsilon}{x} + \frac{1-\epsilon}{1-x} = -\frac{\epsilon - x}{x(1-x)} \leq 0 \tag{25}$$

when $0 < x \leq \epsilon < 1$. Since

$$L(\epsilon, \sigma) = f(p_\sigma) - f(\epsilon), \tag{26}$$

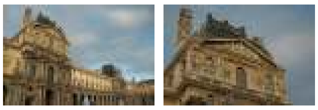
we conclude that L is a decreasing function of σ. We see that L increases with ϵ but decreases with σ. The maximum achieves a balance between a higher number of inliers (that increases with σ) and a lower threshold σ.

Image Credits

Images from the USAC dataset [14][2]:


(H1)


(H3)


(E4)

References

[1] D. Barath, J. Matas, and J. Noskova, *MAGSAC: marginalizing sample consensus*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 10197–10205. `https://doi.org/10.1109/CVPR.2019.01044`.

[2] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, *MAGSAC++, a fast, reliable and accurate robust estimator*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 1304–1312. `https://doi.org/10.1109/CVPR42600.2020.00138`.

[3] D.P. Capel, *An effective bail-out test for RANSAC consensus scoring.*, in British Machine Vision Conference (BMVC), 2005, pp. 78.1–78.10. `https://doi.org/10.5244/C.19.78`.

[4] O. Chum and J. Matas, *Randomized RANSAC with $\mathcal{T}_{d,d}$ test*, Image and Vision Computing, 22 (2004), pp. 837–842. `https://doi.org/10.1016/j.imavis.2004.02.009`.

[5] ——, *Matching with PROSAC - progressive sample consensus*, in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, 2005, pp. 220–226. `https://doi.org/10.1109/CVPR.2005.221`.

[6] O. Chum, T. Werner, and J. Matas, *Two-view geometry estimation unaffected by a dominant plane*, in Computer Vision and Pattern Recognition (CVPR), vol. 1, 2005, pp. 772–779. `https://doi.org/10.1109/CVPR.2005.354`.

---

[2] `https://github.com/gfacciol/usac-cmake` (fork of defunct original repository)

[7] A. Cohen and C. Zach, *The likelihood-ratio test and efficient robust estimation*, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), Dec 2015, pp. 2282–2290. `https://doi.org/10.1109/ICCV.2015.263`.

[8] M. Fischler and R. Bolles, *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM, 24 (1981), pp. 381–395. `https://doi.org/10.1145/358669.358692`.

[9] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2nd ed., 2004. ISBN 978-0521540513.

[10] N. Kai, J. Hailin, and F. Dellaert, *GroupSAC: Efficient consensus in the presence of groupings*, in 12th International Conference on Computer Vision (ICCV), 2009, pp. 2193–2200. `https://doi.org/10.1109/ICCV.2009.5459241`.

[11] L. Moisan, P. Moulon, and P. Monasse, *Automatic Homographic Registration of a Pair of Images, with A Contrario Elimination of Outliers*, Image Processing On Line, 2 (2012), pp. 56–73. `https://doi.org/10.5201/ipol.2012.mmm-oh`.

[12] ——, *Fundamental Matrix of a Stereo Pair, with A Contrario Elimination of Outliers*, Image Processing On Line, 6 (2016), pp. 89–113. `https://doi.org/10.5201/ipol.2016.147`.

[13] P. Moulon, P. Monasse, and R. Marlet, *Adaptive structure from motion with a contrario model estimation*, in Proceedings of the Asian Conference on Computer Vision (ACCV), Springer, 2012, pp. 257–270. `https://doi.org/10.1007/978-3-642-37447-0_20`.

[14] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, *USAC: a universal framework for random sample consensus*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 35 (2012), pp. 2022–2038. `https://doi.org/10.1109/TPAMI.2012.257`.

[15] R. Raguram and J.-M. Frahm, *RECON: Scale-adaptive robust estimation via residual consensus*, in International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 1299–1306. `https://doi.org/10.1109/ICCV.2011.6126382`.

[16] C. Riu, V. Nozick, P. Monasse, and J. Dehais, *Classification performance of RanSaC algorithms with automatic threshold estimation*, in Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), vol. 5, 2022, pp. 723–733. `https://doi.org/10.5220/0010873300003124`.

[17] P.H. Torr, S.J. Nasuto, and J.M. Bishop, *NAPSAC: High noise, high dimensional robust estimation—its in the bag*, in British Machine Vision Conference (BMVC), 2002, pp. 44.1–44.10. `https://doi.org/10.5244/C.16.44`.