HAL
open science

Classification Performance of RanSaC Algorithms with Automatic Threshold Estimation

Clément Riu, Vincent Nozick, Pascal Monasse, Joachim Dehais

To cite this version:

Clément Riu, Vincent Nozick, Pascal Monasse, Joachim Dehais. Classification Performance of RanSaC Algorithms with Automatic Threshold Estimation. 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022), Feb 2022, En ligne, France. pp.723-733, 10.5220/0010873300003124 . hal-03602228

HAL Id: hal-03602228

https://hal.science/hal-03602228v1

Submitted on 8 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Classification Performance of RanSaC Algorithms with Automatic Threshold Estimation

Clément Riu[1], Vincent Nozick[1], Pascal Monasse[1] and Joachim Dehais[2]
[1]*Université Paris-Est, LIGM (UMR CNRS 8049), UGE, ENPC, F-77455 Marne-la-Vallée, France*
[2]*Arcondis AG 4153 Basel, Switzerland*
{*clement.riu,pascal.monasse*}*@enpc.fr, vincent.nozick@univ-eiffel.fr, dehais.joachim@gmail.com*

Keywords: Multi-View Stereo, Structue-From-Motion, RanSaC, Semi-synthetic Dataset, Benchmark

Abstract: The RANdom SAmpling Consensus method (RanSaC) is a staple of computer vision systems and offers a simple way of fitting parameterized models to data corrupted by outliers. It builds many models from small sets of randomly selected data points, and then scores them to keep the best. The original scoring function is the number of inliers, points that fit the model up to some tolerance. The threshold that separates inliers from outliers is data- and model-dependent, and estimating the quality of a RanSaC method is difficult as ground truth data is scarce or not quite reliable. To remedy that, we propose a data generation method to create at will ground truths both realistic and perfectly reliable. We then compare the RanSaC methods that simultaneously fit a model and estimate an appropriate threshold. We measure their classification performance on those semi-synthetic feature correspondence pairs for homography, fundamental, and essential matrices. Whereas the reviewed methods perform on par with the RanSaC baseline for standard cases, they do better in difficult cases, maintaining over 80 % precision and recall. The performance increase comes at the cost of running time and analytical complexity, and unexpected failures for some algorithms.

1 INTRODUCTION

Over fourty years have passed since the publication of the RANdom SAmple Consensus algorithm (Fischler and Bolles, 1981). The algorithm approximates a solution to the NP-hard consensus maximization problem (Chin et al., 2018): fitting a parameterized model to data corrupted by noise and outliers. RanSaC generates tentative models using random, minimal subsets of data and validates them with the distance of data points to each model. Data points whose model distance are below a given threshold are inliers, and the retained model maximizes the inlier count, or consensus. Defining this threshold however requires knowing the noise level in the data. Some methods lift this constraint by estimating jointly a threshold and a model. For them, the inlier count is replaced by new measures involving both threshold and inlier count for optimization. We call these methods *adaptive* or *automatic*.

We review these methods and examine their performance on a benchmark of three computer vision problems central to Multi-View Stereo (MVS) and Structure-From-Motion (SfM) pipelines (Schonberger and Frahm, 2016; Moulon et al., 2016): homography, fundamental and essential matrix fitting. Data for these problems consists of feature point pair correspondences, leading the algorithm to classify them as inlier or outlier.

Firstly, automatic threshold estimation does benefit MVS pipelines (Moulon et al., 2012), although its performance should be measured on ground-truth data independently. Secondly, incremental MVS pipelines also benefit from good inlier classification, as inliers are triangulated to from 3D points, thereafter used as anchors for following views to estimate their pose through Perspective from n Points (PnP). Thirdly, since RanSaC methods are the basis of SfM and MVS software, robust analysis will improve the state of the art. Fourthly and finally, deep-learning 3D pipelines use the output of MVS pipelines, most often (Schonberger and Frahm, 2016), for training database generation (Li and Snavely, 2018).

We first contribute a method to create semi-synthetic data with ground truth labels to assess classification power of RanSaC algorithms. Previous artificial datasets are unrealistic; *e.g.* (Cohen and Zach, 2015) performs the evaluation on plane estimation in random 3D point clouds, and (Barath et al., 2019) generates random homographies and cor-

Table 1: Notations

	Definition	Description
k	$\in \mathbb{N}_{>0}$	dimension of data points
$\mathcal{S}$	$= \mathbb{R}^k$	space of data points
$\mathcal{P}$	$\subset \mathcal{S}$	set of input data points
d	$\in \mathbb{N}_{>0}$	degrees of freedom of a model
Θ	$\subset \mathbb{R}^d$	space of model parameters
θ	$\in \Theta$	parameter vector of a model
s	$\in \mathbb{N}_{>0}$	data sample size
Sa	$: 2^{\mathcal{S}} \to \mathcal{S}^s$	sampling function
F	$: \mathcal{S}^s \to \Theta$	fitting function
p	$: [0,1] \to [0,1]$	proba. of sampling inliers only
D	$: \mathcal{S} \times \Theta \to \mathbb{R}$	point-model residual function
σ	$\in \mathbb{R}$	inlier threshold
I	$: \Theta \times \mathbb{R} \to 2^{\mathcal{P}}$	inlier selector function
I	$= I(\theta, \sigma) \subset \mathcal{P}$	set of model inliers
Q	$: 2^{\mathcal{P}} \times \Theta \to \mathbb{R}$	model quality function

respondences. Real datasets on the opposite end cannot offer a control over the noise and outliers: they often rely on an algorithm to calibrate an active sensor to the data which introduces error like the KITTI dataset (Geiger et al., 2012) and the 7 scenes dataset (Shotton et al., 2013) or do not correct inlier noise as in (Cordts et al., 2016). Our method offers the possibility to create, from any image dataset, a realistic semi-synthetic dataset.

Note that there have already been comparative studies of RanSaC, such as (Choi et al., 2009). These studies however focus on threshold-controlled RanSaC derivatives on a small real-world dataset without ground truth. We contribute an analysis of the classification and retrieval power of adaptive RanSaC algorithms. We thus pit threshold-free methods against each other, rather than against baseline RanSaC as was done before.

Section 2 reviews the automatic RanSaC methods, Section 3 the benchmark data and methods. Sections 4 and 5 cover the results and their analysis, and Section 6 concludes.

2 AUTOMATIC RANSAC METHODS

2.1 Notation

We reuse and simplify the notation of (Barath et al., 2019), standardized in Table 1. Input points exist in $\mathcal{S} = \mathbb{R}^k, k > 0$, with the input point set noted as $\mathcal{P} \subset \mathcal{S}$. For image coordinates, the point dimension $k = 2$; for pair correspondences, $k = 4$.

A model of d degrees of freedom exists in $\Theta \subset \mathbb{R}^d$, and can be obtained via the fitting function F, applied to a set of data points sampled by the sampling function Sa. $d = 8$ for homographies, 7 for fundamental matrices, and 5 for essential matrices.

The probability of Sa drawing an uncontaminated sample (including only correct data) is $p(\varepsilon)$, with $\varepsilon \in [0,1]$ the true (hence generally unknown) inlier ratio. Sa typically samples s points uniformly in $\mathcal{P}$, hence $p(\varepsilon) = \varepsilon^s$, an increasing function of ε.

Given a model θ, we can calculate the residual error between that model and a data point P using $D(P, \theta)$. I selects the points that are inliers to a model based on the residuals. Finally, Q defines the quality of the fit θ on $\mathcal{P}$.

While D is model-dependent, the standard selector I keeps points with residuals below a threshold σ: $I = I(\theta, \sigma) = \{P \in \mathcal{P} : D(P, \theta) < \sigma\}$. $Q(I, \theta) = |I|$ is then the number of inliers. Since inlier count increases with σ, standard Q definitions are unreliable quality measures for RanSaC variants. Algorithm 1 is the pseudo-code for a generic RanSaC. A type II error corresponds to missing the true model due to insufficient search.

Input: $\mathcal{P}$, Sa, F, p, Q, σ
Input: confidence against type II error β, it_{max} (or min. inlier rate ε)
Output: Best model parameters and set of inliers
$I_{max} = \emptyset$, $q_{max} = 0$
$it = 0$ (and $it_{max} = \frac{ln(1-\beta)}{ln(1-p(\varepsilon))}$ if ε is input)
while $it \leq it_{max}$ **do**
  $sample = Sa(\mathcal{P})$, $\theta = F(sample)$
  $I = I(\theta, \sigma)$, $q = Q(I, \theta)$
  if $q > q_{max}$ **then**
    $q_{max} = q, \theta_{max} = \theta, I_{max} = I$
    $\varepsilon = |I|/|\mathcal{P}|, it_{max} = \frac{ln(1-\beta)}{ln(1-p(\varepsilon))}$
  $it = it + 1$
return θ_{max}, I_{max}

Algorithm 1: RanSaC algorithm

2.2 Existing Algorithms

The methods gathered in Table 2 generate new inlier and model quality definitions to behave adaptively. They rely on three types of consistency models: inlier, outlier, and cross-model. An inlier- (resp. outlier-)consistent method enforces a parameterized distribution on the inlier (resp. outlier) residuals, and searches for a model and parameter set that fit the data. A cross-model-consistent method looks for re-

Table 2: Overview of robust fitting methods with adaptive inlier criteria. Bracketed terms in the "Consistency assumption" column specify where the assumption is made.

Ref.	Consistency assumption	Optimized function
(Fischler and Bolles, 1981)	Bounded residuals (inliers)	Inlier set size
(Miller and Stewart, 1996)	Gaussian residual distribution (inliers)	Scale estimate
(Wang and Suter, 2004)	Gaussian (inliers) Minimal residual density (transition)	Inlier/threshold
(Fan and Pylvänäinen, 2008)	Gaussian (inliers) Residuals correlate (cross-model)	Scale estimate
(Choi and Medioni, 2009)	Low parameter variance (cross-model)	Variance
(Moisan et al., 2012)	Problem specific (outlier)	Number of false alarms
(Cohen and Zach, 2015)	Uniform (outliers)	Likelihood
(Moisan et al., 2016)	Problem specific (outliers)	Number of false alarms
(Dehais et al., 2017)	Data-driven (outliers)	False discovery rate
(Barath et al., 2019)	Uniform (inliers) - Uniform (outliers)	Deviation
(Rabin et al., 2010)	Problem specific (outliers)	Greedy number of false alarms
(Isack and Boykov, 2012)		Graph cuts energy
(Toldo and Fusiello, 2013)	Residuals correlate (cross-model)	Inlier cluster merging cost
(Magri and Fusiello, 2014)	Residuals correlate (cross-model)	Residual cluster merging cost
(Magri and Fusiello, 2015)	Residuals correlate (cross-model)	Factorization error of residual matrix

peated patterns in residuals of different models using cross correlations or clustering. The original RanSaC (Fischler and Bolles, 1981) algorithm uses inlier consistency: all inliers have residuals below a certain threshold.

The first automatic threshold method was MUSE (Miller and Stewart, 1996). MUSE assumes that inlier residuals follow a Gaussian distribution, and estimates the noise level for all inlier sets of all tested models. The smallest noise for each model becomes its quality measure, and the model with the smallest noise level is selected overall.

(Wang and Suter, 2004) relies on inliers following a unimodal distribution (specifically Gaussian in experiments). For single models, this method puts a threshold on residual distribution density. The quality measure is the ratio of inliers to inlier threshold value.

The method in (Fan and Pylvänäinen, 2008) merges inlier and cross-model consistency. This method uses the weighted median absolute deviation of all residuals as both scale estimate and quality function. Each time a data point has a small residual for a sampled model, its weight and probability to be sampled increase. This way, the weight biases the quality function towards the median absolute deviation of inliers to stable models.

StarSaC (Choi and Medioni, 2009) uses cross-model consistency as well, namely that variance of model parameters is minimal for the optimal threshold. The method calls RanSaC with multiple thresholds, multiple times per threshold. The quality measure is the variance of resulting model parameters for a given threshold.

A Contrario RanSaC (Moisan et al., 2012; Moisan et al., 2016) (AC-RanSaC, originally named ORSA (Moisan and Stival, 2004)) was the first outlier-consistent method. AC-RanSaC approximates for each problem the probability distribution of residuals for random input: the background distribution. It then minimizes a combinatorial prediction of false inliers by comparing the cumulative distributions of data and background residuals.

An extension of AC-RanSaC (Dehais et al., 2017) adapts the method in two ways: by generating outlier data from inlier data, and simplifying the significance measure. This method scrambles input matches to generate background distributions, and seeks the model with the most inliers subject to a maximal rate of inliers in the background data.

The likelihood-ratio method (Cohen and Zach, 2015) relies on a uniform distribution of outlier residuals. The quality function approximates the likelihood of each model against this uniform distribution, using substantial statistical derivations.

MAGSAC (Barath et al., 2019) is the most recent publication on this work's subject. It combines inlier and outlier consistency assumptions, where both follow uniform distributions on different intervals. MAGSAC then estimates the model's deviation from the background noise, without explicitly classifying inputs into inliers and outliers.

Other methods exist to detect multiple models in data; though they are out of scope, we list some here. (Rabin et al., 2010) uses the principle of A Contrario RanSac (Moisan et al., 2012) in a greedy optimization to extract multiple models. (Isack and Boykov, 2012)

generates multiple models, before attaching data to those models to minimize total residuals and number of models. J-Linkage (Toldo and Fusiello, 2013), its extension T-Linkage (Magri and Fusiello, 2014), and Low-Rank Preference Analysis (Magri and Fusiello, 2015) rely on cross-model consistency, while using different methods to cluster inlier sets and models jointly.

3 BENCHMARK

3.1 Data Generator

The distributions of inliers and ouliers in the image space have a major impact on the effectiveness of model fitting. Benchmarks on data extracted from photographs are invaluable, but lack the power to generalize observations because of their small size and the lack of control over noise and outliers. Synthetic data on the other hand are flexible and customizable but unrealistic as the chosen model generation and inlier sampling methods impact the results.

We propose a simple procedure to generate realistic, semi-synthetic data with ground truth classification. It gives access to real models and inliers, with outliers whose distance to the model is controlled.

The data generator takes an image pair with point matches and evaluates its underlying model using AC-RanSaC (Moisan et al., 2012; Moisan et al., 2016) at arbitrary precision and up to 10 000 iterations. The returned model is then used as "ground truth", and the estimated inliers as the basis for data-driven yet perfect inliers.

To generate perfect inliers, the matches are modified with respect to the ground truth model. For homographies, inlier points in the first image are mapped into the second using the ground truth homography (figure 1). For fundamental and essential matrices, the point on the second image of each inlier match is projected on the true epipolar line of the first point (figure 2).

The artificial inlier matches and outlier matches can then be derived from those "ground truth" inlier matches. Inlier matches are created by adding noise uniformly drawn in $[-\sigma_{noise}, \sigma_{noise}]^2$ to the second image point of each match to generate a controled pixel error. Uniform noise gives here better control and testability for the inlier/outlier threshold than Gaussian noise, which is unbounded.

Once inlier matches are generated, outliers are inserted by generating random matches in the two images, uniformly sampled in error space ensuring a distance to the model greater than the maximum inlier error σ_{noise}. Sampling in error space is explained for homography estimation in figure 3 and for fundamental and essential matrix estimation in figure 4. This choice of distribution in error space returns a realistic outlier distribution and truly challenges the algorithms. A simple uniform distribution of outliers in match space on the contrary has too little structure to disturb the RanSaC algorithm. For each dataset and inlier ratio, we generate enough outliers to obtain the desired ratio, and downsample uniformly to have 4000 data points or fewer.

3.2 Tested Algorithms

We benchmark seven algorithms with varied approaches to inlier classification: the baseline RanSaC (Fischler and Bolles, 1981), MUSE (Miller and Stewart, 1996), StaRSaC (Choi and Medioni, 2009), A-Contrario RanSaC (AC-RanSaC) (Moisan et al., 2016), Likelihood Ratio Test (LRT) (Cohen and Zach, 2015), Marginalizing Sample Consensus (MAGSAC) (Barath et al., 2019) and Fast-AC-RanSaC—an adaptation of AC-RanSaC. (Isack and Boykov, 2012; Toldo and Fusiello, 2013; Magri and Fusiello, 2014), though relatively recent, have been excluded as they tackle the multi-model settup which we do not consider in this study.

RanSaC acts as baseline, with fixed inlier/outlier thresholds σ(px) to show non-adaptive results. To avoid degenerate cases in the worst case we strengthen its stopping criterion to ensure it draws 5 outlier-free samples instead of just 1 with confidence β. This requires a modification in the formula for it_{max} in Algorithm 1.

MUSE proposes an improvement over Least Median of Squares (Leroy and Rousseeuw, 1987) with a new objective function based on a scale estimate to increase robustness wrt outliers. The algorithm works similarly to RanSaC, with multiple iterations of a minimal sampler estimating a model and then ranking based on their scale estimate. We adapted the algorithm by adding the classic termination criterion of RanSaC to speed up the execution.

StaRSaC removes the need for a user set threshold of RanSaC by launching multiple RanSaCs for different thresholds. It then estimates at each threshold the variance over the parameters and the threshold yielding the lowest variance is selected. Our implementation uses the same thresholds as (Cohen and Zach, 2015) to speed up the process by only considering thresholds that make sense in pixel-scale problems.

AC-RanSaC seeks the model with the lowest risk of type I error. This is measured as the Number of False Alarms (NFA), which estimates the expected

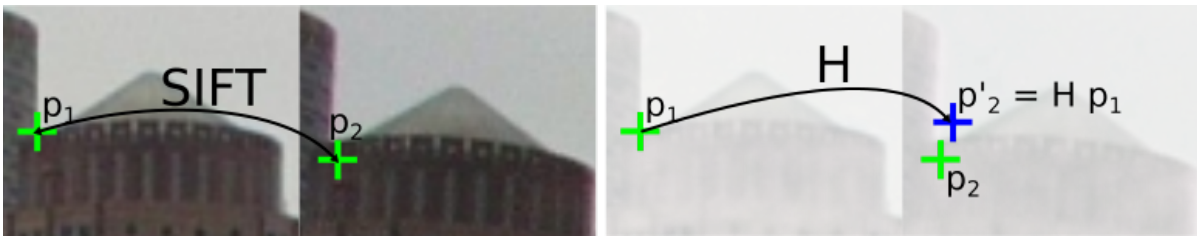

Figure 1: From an imperfect match (p_1, p_2) considered inlier by AC-RanSaC, the "perfect match" (p_1, p_2') is constructed such that $p_2' = Hp_1$ using a realistic homography H given by AC-RanSaC.

Figure 2: From an imperfect match (p_1, p_2) considered inlier by AC-RanSaC, the "perfect match" (p_1, p_2') is constructed using p_2' the orthogonal projection of p_2 on the epipolar line $\mathcal{L}_1 = Fp_1$ where F is a realistic fundamental matrix given by AC-RanSac. This does not guarantee that p_2' represents the same physical point as p_1, but that some 3D point at possibly different depth projects exactly at p_1 and p_2'.

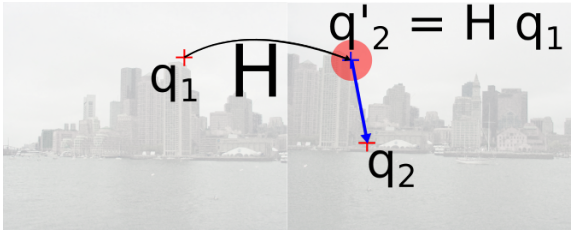

Figure 3: A random point q_1 is drawn from the left image. Using the ground truth model H, its perfect match $q_2' = Hq_1$ is computed. Then a direction and a distance to q_2' are drawn uniformly in order to create q_2 so that it remains in the image and out of the inlier zone (marked in red) defined by the inlier noise level.

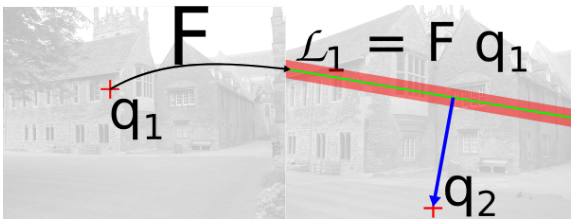

Figure 4: A random point q_1 is drawn from the left image. Using the ground truth model F, the epipolar line $\mathcal{L}_1 = Fq_1$ is computed. Then position on this line and a distance to $\mathcal{L}_1$ are drawn uniformly in order to create q_2 so that it remains in the image and out of the inlier zone (marked in red) defined by the inlier noise level.

number of false positive models generated by an inlier set:

$$NFA(\theta,\sigma) \sim \binom{|\mathcal{P}|}{k(\sigma)} \binom{k(\sigma)}{s} p_\sigma^{k(\sigma)-s}, \qquad (1)$$

with $k(\sigma) = |I(\theta,\sigma)|$ the number of inliers at threshold σ and p_σ the relative area of the image zone definining inliers at σ. This function is computed for all $\sigma \leq \sigma_{\max}$ such that $k(\sigma) \in \{s+1,\ldots,n\}$. The method is virtually parameter-free, with $\sigma_{max} = +\infty$ and the NFA upper bound $NFA_{max} = 1$.

The Likelihood Ratio Test (LRT) proposes a fast method to control type I and type II errors. Its quality function is the likelihood that the dataset is non-random for a given model:

$$L(\varepsilon,\sigma) = 2|\mathcal{P}| \left(\varepsilon \log \frac{\varepsilon}{p_\sigma} + (1-\varepsilon) \log \frac{1-\varepsilon}{1-p_\sigma} \right), \quad (2)$$

with inlier ratio $\varepsilon = k(\sigma)/|\mathcal{P}|$ and σ spanning a predefined list $\{\sigma_{min},\ldots,\sigma_{max}\}$. The adjustment of it_{max} is the same as RanSaC, with an early bailout system to sift through models faster, controlled by a new parameter γ for the type II error (rejection of a good model).

MAGSAC introduces a new quality function, based on the likelihood of a model given uniform inliers and outliers, a speedup method using computing blocks, and a post processing method, σ-consensus, that refits the final model on a weighted set of pseudo-inliers. It uses four parameters: the threshold for pseudo-inliers $\sigma_{\max}$, a reference threshold σ_{ref}, a lower bound on model likelihood, and the number of data blocks.

Fast-AC-RanSaC is a speed up of AC-RanSaC by exploiting ideas from LRT. The idea to create this method comes from our observations, see Section 5. It uses the same objective function as AC-RanSaC but instead of testing all thresholds it considers the same

quantified thresholds as LRT. Using this selection of thresholds removes the need for a sorting of the residuals and thus speeds up the method.

3.3 Performance Measures

To compare the classification power of the various algorithms, we use precision—the ratio of correctly detected inliers among all detected inliers—and recall—the ratio of correctly detected inliers among true inliers. To represent results more easily, we use the harmonic mean of precision and recall, the F1-Score. We quantify the impact of noise and outliers on these metrics using the semi-synthetic data of Section 3.1. For efficiency evaluation, we measure execution time.

As MAGSAC does not separate data points into inliers and outliers but weighs them, we evaluate its performance through three metrics: highest recall at the precision of AC-RanSaC, which will be called `Magsac-P`, highest precision at the recall of AC-RanSaC, `Magsac-R`, and weighted equivalents of precision and recall, `Magsac-W`.

3.4 Parameters

We experimented the data generator on the SIFT (Lowe, 2004) point matches from the USAC dataset as well as SIFT points computed on Multi-H, kusvod2 and the homogr datasets. A SIFT ratio of 0.6 was used. USAC (Raguram et al., 2012) proposes SIFT matches for 10 homography image pairs, 11 fundamental matrix image pairs, and 6 essential matrix image pairs with calibration matrices. Multi-H (Barath et al., 2016) proposes 24 fundamental matrix image pairs, kusvod2 16 fundamental matrix image pairs, and homogr 16 homography image pairs.[1]

The inlier noise σ_{noise} varied between 0 and 3 pixels by steps of 0.1. The outlier ratio $1-\varepsilon$ varied between 0 and 0.9 by steps of 0.1. For each image pair and setting, we generate $N_{gen} = 5$ different datasets, and run each algorithm $N_{run} = 5$ times on each for a total of 25 experiments. We average the precision and recall over each dataset and run, excluding cases where an algorithm failed to find a good model.

As minimal solvers F, we use the standard 4-point algorithm for homography and 7-point algorithm for fundamental matrix (Hartley and Zisserman, 2004), and the 5-point algorithm for essential matrix (Nistér, 2004).

We take the standard RanSaC algorithm as baseline and benchmark MUSE, StarSAC, LRT, AC-RanSaC, Fast-AC-RanSaC, and MAGSAC algorithms. RanSaC and AC-RanSaC implementations came from (Moulon, 2012), MAGSAC from (Barath, 2019), MUSE from VXL[2] whereas LRT, Fast-AC-RanSaC and StarSAC were implemented from scratch.

The parameters defined in section 3.2 are set as follows (from the relevant publicaion when available). β, the standard success confidence, is set to 0.99. σ_{max}, the inlier search cutoff threshold for AC-RanSaC, Fast-AC-RanSaC, StarSaC and LRT, is set to 16 pixels to improve speed. NFA_{max}, the NFA threshold for AC-RanSaC and Fast-AC-RanSaC, is set to 1 false alarm. The other parameters of LRT, were set to $\alpha = 0.01$, $\gamma = 0.05$.

The other parameters of MAGSAC were set to $p = 10$ data partitions, $\sigma_{max} = 10$ pixels, and $\sigma_{ref} = 1$.

4 RESULTS

Before presenting results for the different methods, StarSAC has been excluded from this benchmark. Indeed, after initial tests, it runs very slowly, even with modification to the initial algorithm to reduce runtime. A usual run will take 3 to 5 minutes with precision and recall only slightly above baseline.

Figure 5 gives an overview of each algorithm's execution time. LRT, MUSE, Fast-AC-RanSaC and RanSaC are usually the fastest, with LRT faster than the others on easy cases. Both LRT and RanSaC however are quite sensitive to the task complexity: when facing high inlier noise or outlier ratio, their runtime increases significantly, sometimes up to four or five seconds per run. MUSE and Fast-AC-RanSaC keep relatively low runtimes in all settings, always below $1s$. AC-RanSaC's runtime is not impacted by inlier noise, only by the number of matches in the dataset. Its runtime remains almost always under one second per run, though it is often the slowest algorithm on easy settings. However, it can be faster than Fast-AC-RanSac on easy settings with not many points where the sorting time of residuals is fast compared to the computation of the residuals themselves. MAGSAC is the most efficient algorithm on complex tasks with runtimes up to ten times lower than AC-RanSaC. However, because of instabilities in MAGSAC's execution, a 1-second cap was used: for inlier noise level above 1 pixel or outlier ratio below 0.4, MAGSAC can find a good model but still fail to terminate.

Except for RanSaC, no algorithm presented a compromise between precision and recall (one increasing while the other decreases). This justifies the

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

[2] https://github.com/vxl/vxl

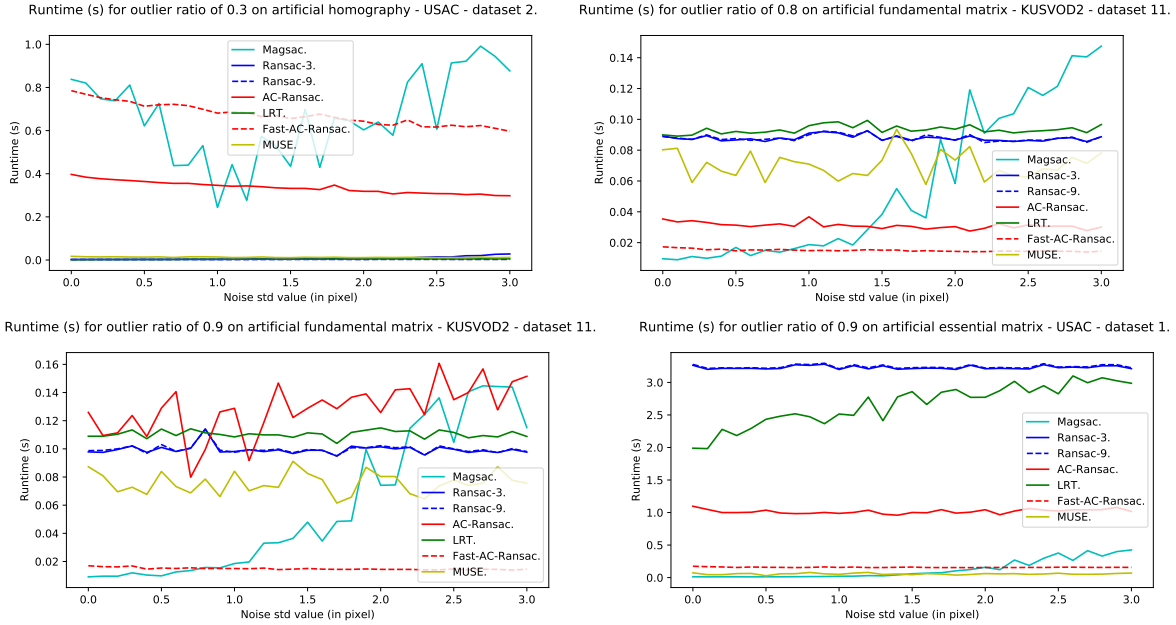

Figure 5: Runtimes over different inlier noise levels and outlier ratios. The fitting problem, dataset name, and image pair number are in each graph title. `Ransac-`σ corresponds to RanSaC with threshold σ.

use of F1-Score over either precision or recall.

Trends in behavior were impacted by the noise level and outlier ratio for all algorithms, but were neither by task, be it homography, fundamental matrix or essential matrix estimation, nor by dataset. While performance differed between image pairs or different tasks, performance always decreased consistently relative to the data generator parameters. Figures 6 and 7 illustrate the typical behaviors with low and high outlier ratios on a variety of estimation problems and datasets. An intermediate setting, for example an outlier ratio around 0.5, is not included as the behavior was observed to be intermediate between the two extremes.

The two fixed-threshold RanSaC have opposite behaviors. The "tight" RanSaC with a threshold of 3 pixels has high precision on easy situations, but poor recall, and both drop as noise and outliers increase. The "loose" RanSaC with a threshold of 9 pixels has stable recall, but precision drops as noise and outlier ratio increase. Both eventually perform poorly in complex cases as too many outliers are wrongly classified.

MUSE's performance is not impacted much by the semi-artificial generation parameters but does not show consistently good behavior. On the one hand, it can have poor performance on easy settings, with below baseline precision and recall. On the other hand, it can show very good precision, above 95% on very complex settings. However, its recall is always low as the thresholds selected are usually smaller than other methods.

AC-RanSaC shows a very good performance on low outlier ratios. It maintains good precision as the outlier ratio increases, but its recall drops, impacting the F1-score.

LRT returns slightly lower precision and recall than AC-RanSaC, for low outlier ratios or cases where all algorithms perform well. The gap increases significantly for the highest inlier noise levels or outlier ratios data become more challenging, the performance dropping to baseline.

Fast-AC-RanSaC has medium performance. It has usually good recall but average precision as it selects a threshold twice as big as the real inlier noise level. It can however, in some cases perform relatively well or even to the same level as normal AC-RanSac but it is not a consistent behavior.

MAGSAC displays two interesting behaviors. For outlier ratios below 0.4, MAGSAC alternates between great fits and nearly random models, instead of consistently returning a satisfactory model. This explains the F1-scores around 0.75 on figure 6, the result of averaging models with precision and recall below 0.5 and models with precision and recall above 0.95. For outlier ratios above 0.4, MAGSAC shows equal or greater performances than AC-RanSaC and far less sensitivity to inlier noise. MAGSAC is also the only method to maintain precision and recall scores above 0.9 (resp. 0.8) in challenging cases. The 1-second cap

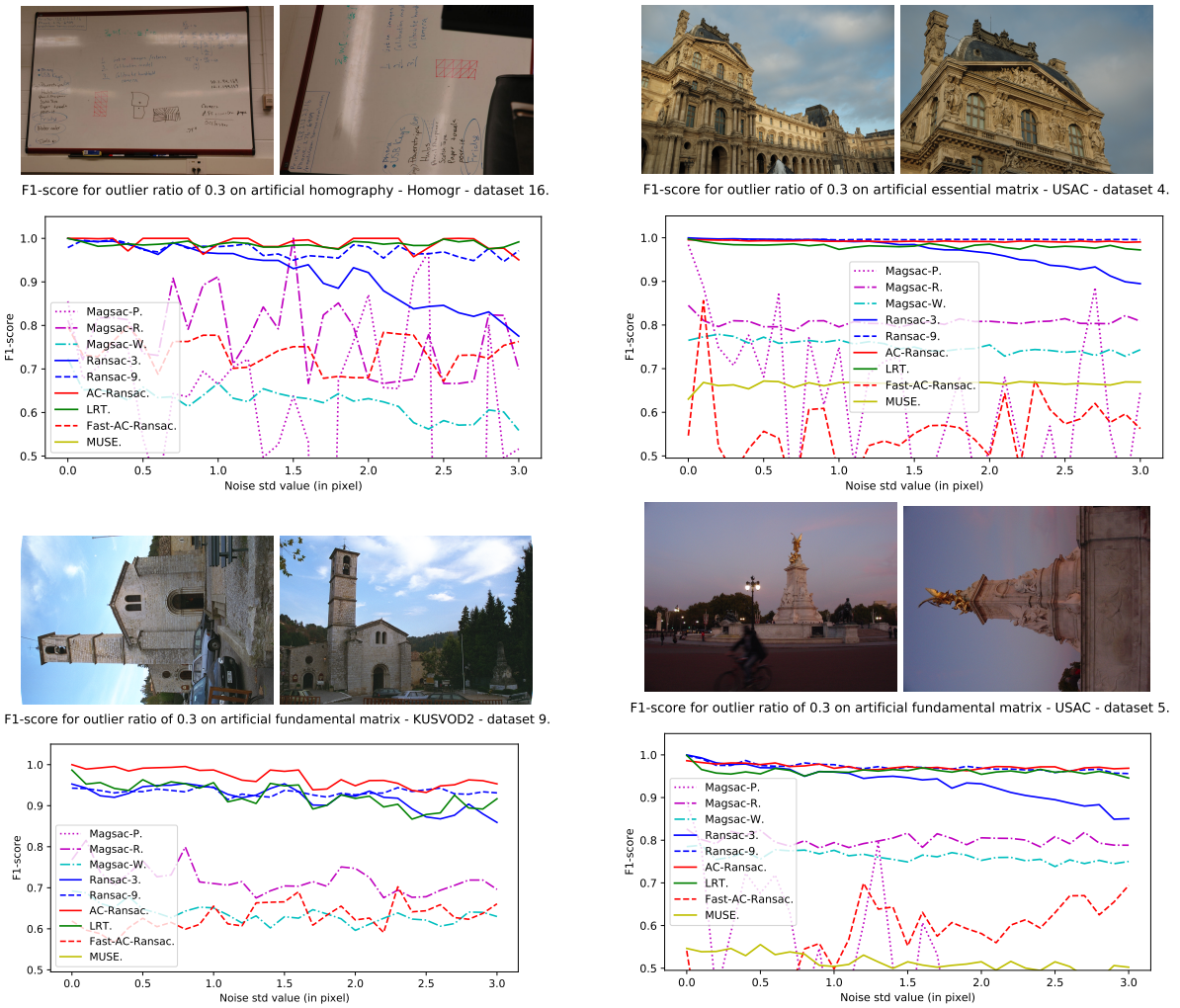

Figure 6: Typical F1-score evolution over inlier noise for a low outlier ratio (0.3). Estimation problem, dataset name and image pair number can be found in each graph's title. `Magsac-P`, `Magsac-R` and `Magsac-W` correspond to the metrics presented in Section 3.3.

on MAGSAC runtime does not impact performance in simple cases, where bad models are estimated irrespectively of runtime. On challenging cases, this cap is sometimes reached but performance remains overall better than other algorithms.

5 DISCUSSION

Differences in runtime can be explained by the number of iterations in each algorithm as much as their design. Indeed, each algorithm has to compute models—which is fast—and compute residuals for all points—slow. LRT's high and low speeds likely stem from its early bailout strategy, skipping residuals entirely. This is a benefit in easy settings, where correct models are plentiful, but hurts complexity in the opposite case, where some rare correct models are not fully evaluated. AC-RanSaC's high runtime for large datasets comes from its sorting step, which takes then as much time as computing residuals. However, this may be amended by bucket sorting and discretization of σ as with LRT to remove sorting entirely. This is what led to the development of Fast-AC-RanSaC which shows promising results but could be improved as its performance is usually worst than that of LRT that was developped directly in this settings.

StarSAC offers poor performance compared to other methods except RanSaC. Except in some rare cases, MUSE's performance is below that of newer methods. As explained below, in all settings one of the newest methods will offer better performance.

Though it is usually the slowest method, AC-

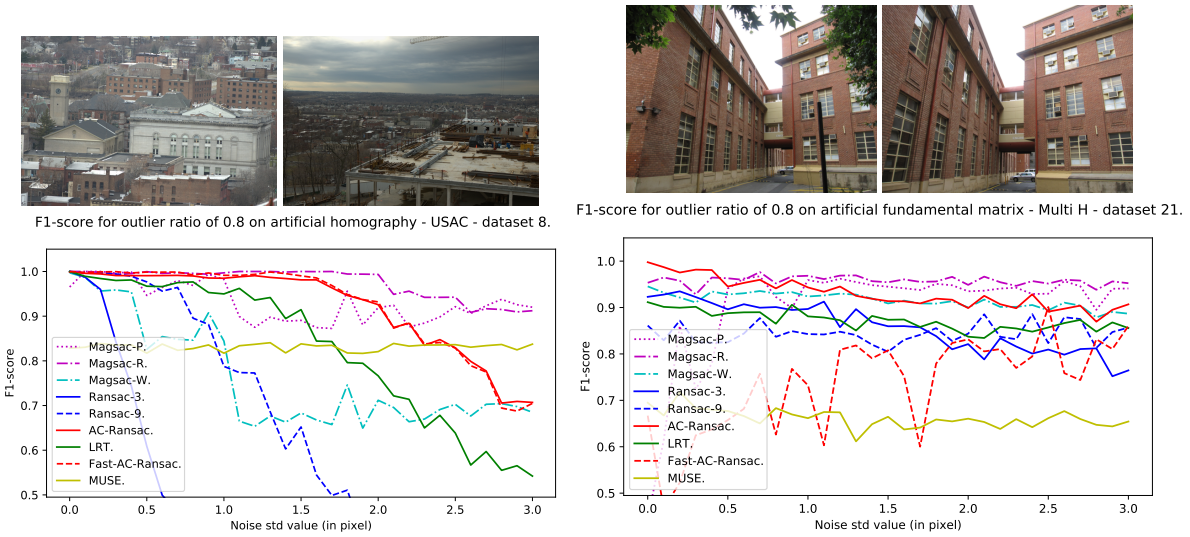

Figure 7: Typical F1-score evolution over inlier noise for a high outlier ratio (0.8). Estimation problem, dataset name and image pair number can be found in each graph's title. `Magsac-P`, `Magsac-R` and `Magsac-W` correspond to the metrics presented in Section 3.3.

RanSaC shows consistently good precision and recall. The algorithm does adapt well to change, and performs far above baseline in difficult cases. It is a robust but slower algorithm and except in our most complex settings, it produces a good result so it is a good pick when high classification performance is required.

LRT performed slightly worse than AC-RanSaC, with selected inlier threshold usually larger, yet much faster. LRT thus offers a valuable trade-off between quality and speed. It is however fairly sensitive to the complexity of the task: a decrease in performance to baseline level in complex settings and a steep increase in runtime.

Fast-AC-RanSaC was developped to improve over AC-RanSaC by reducing runtime but it fails to outperform LRT in most cases.

MAGSAC has the potential to perform similarly or better than AC-RanSaC, at comparable or better speeds with impressive performance on the most complex tasks. It has a very strong resilience to outlier and inlier noise and should be used when other algorithms fail to produce a satisfying result. The lack of robustness of its current implementation makes it more risky to use first hand.

Finally, our semi-artificial datasets help reveal behavior independent of the dataset chosen as initial input, and thus offers new insight in automatic RanSaC algorithm analysis.

6 CONCLUSION

RanSaC is a robust and efficient algorithm for a wide range of situations, whose major weaknesses appear in case of high outlier ratios and unknown inlier noise. Thanks to a new method to generate semi-synthetic ground truth data we have made a quantitative evaluation of three major algorithms for noise- and outlier-robust fitting based on RanSaC. For most standard usages, all algorithms perform well, with only execution speed varying. When outliers outnumber inliers and noise is large, adaptive methods justify their existence by outperforming standard RanSaC by a large margin, at the cost of speed. Overall, the algorithms offer a choice for those who seek to trade robustness, accuracy, and execution speed. For example, LRT is a good algorithm to use on simple cases, offering better performance than traditional RanSaC in comparable runtime. MAGSAC has a very strong resilience to outlier and inlier noise and should be used when other algorithms fail to produce a satisfying result. The lack of robustness of its current implementation makes it more risky to use first hand. AC-RanSaC would then be a more robust but slower algorithm. Except in our most complex settings, it produces a good result so it is a good pick when high accuracy is required. The variety of tests show the strength of the data generation procedure in exhibiting the sensitivity of the methods to each problem, to noise, and to outliers.

Thanks to such observations, we plan to improve these methods using the optimizations proposed above as we did with Fast-AC-RanSaC that proved a

potential improvement if a more stable implementation can be developped. It is also possible now to perform an ablation study of each proposed ingredient like the σ-consensus post-processing of MAGSAC to determine precisly their impact on the robustness and retrieval power of each method. Finally, we plan to extend the study on the remaining model-fitting problem involved in MVS pipelines: Perspective from *n* points (P*n*P). This problem recovers the camera pose of a view inserted in the pipeline from 2D-3D correspondences, and model quality measures should be adapted to that specific case.

REFERENCES

Barath, D. (2019). MAGSAC implementation. https://github.com/danini/magsac. [Online; accessed 22-January-2021].

Barath, D., Matas, J., and Hajder, L. (2016). Multi-H: Efficient recovery of tangent planes in stereo images. In Richard C. Wilson, E. R. H. and Smith, W. A. P., editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 13.1–13.13. BMVA Press.

Barath, D., Matas, J., and Noskova, J. (2019). MAGSAC: marginalizing sample consensus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10197–10205.

Chin, T.-J., Cai, Z., and Neumann, F. (2018). Robust fitting in computer vision: Easy or hard? In *Proceedings of the IEEE European Conference of Computer Vision (ECCV)*, pages 701–716.

Choi, J. and Medioni, G. (2009). StaRSaC: Stable random sample consensus for parameter estimation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 675–682.

Choi, S., Kim, T., and Yu, W. (2009). Performance evaluation of RANSAC family. In *Proceedings of the British Machine Vision Conference (BMVC)*.

Cohen, A. and Zach, C. (2015). The likelihood-ratio test and efficient robust estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2282–2290.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dehais, J., Anthimopoulos, M., Shevchik, S., and Mougiakakou, S. (2017). Two-view 3D reconstruction for food volume estimation. *IEEE Transactions on Multimedia*, 19(5):1090–1099.

Fan, L. and Pylvänäinen, T. (2008). Robust scale estimation from ensemble inlier sets for random sample consensus methods. In *Proceedings of the IEEE European Conference of Computer Vision (ECCV)*, pages 182–195. Springer Berlin Heidelberg.

Fischler, M. and Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hartley, R. and Zisserman, A. (2004). *Multiple view geometry in computer vision*. Cambridge University Press, 2nd edition. ISBN 978-0521540513.

Isack, H. and Boykov, Y. (2012). Energy-based geometric multi-model fitting. *International Journal of Computer Vision (IJCV)*, 97(2):123–147.

Leroy, A. M. and Rousseeuw, P. J. (1987). Robust regression and outlier detection. *Wiley series in probability and mathematical statistics*.

Li, Z. and Snavely, N. (2018). MegaDepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2041–2050.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110.

Magri, L. and Fusiello, A. (2014). T-linkage: A continuous relaxation of J-linkage for multi-model fitting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3954–3961.

Magri, L. and Fusiello, A. (2015). Robust multiple model fitting with preference analysis and low-rank approximation. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 20.1–20.12.

Miller, J. V. and Stewart, C. V. (1996). MUSE: Robust surface fitting using unbiased scale estimates. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 300–306.

Moisan, L., Moulon, P., and Monasse, P. (2012). Automatic homographic registration of a pair of images, with a contrario elimination of outliers. *Image Processing On Line (IPOL)*, 2:56–73.

Moisan, L., Moulon, P., and Monasse, P. (2016). Fundamental matrix of a stereo pair, with a contrario elimination of outliers. *Image Processing On Line (IPOL)*, 6:89–113.

Moisan, L. and Stival, B. (2004). A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *International Journal of Computer Vision (IJCV)*, 57(3):201–218.

Moulon, P. (2012). AC-RanSaC implementation. https://github.com/pmoulon/IPOL_AC_RANSAC. [Online; accessed 22-January-2021].

Moulon, P., Monasse, P., and Marlet, R. (2012). Adaptive structure from motion with a contrario model estimation. In *Proceedings ot the Asian Conference of Computer Vision (ACCV)*, pages 257–270. Springer.

Moulon, P., Monasse, P., Perrot, R., and Marlet, R. (2016). OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(6):756–770.

Rabin, J., Delon, J., Gousseau, Y., and Moisan, L. (2010). MAC-RANSAC: a robust algorithm for the recognition of multiple objects. *Proceedings of the Fifth International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 51–58.

Raguram, R., Chum, O., Pollefeys, M., Matas, J., and Frahm, J.-M. (2012). USAC: a universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(8):2022–2038.

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

Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., and Fitzgibbon, A. (2013). Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937.

Toldo, R. and Fusiello, A. (2013). Image-consistent patches from unstructured points with J-linkage. *Image and Vision Computing*, 31:756–770.

Wang, H. and Suter, D. (2004). Robust adaptive-scale parametric model estimation for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(11):1459–1474.