EVSAC: Accelerating Hypotheses Generation by Modeling Matching Scores with Extreme Value Theory

Victor Fragoso Pradeep Sen Sergio Rodriguez Matthew Turk
University of California, Santa Barbara
{vfragoso@cs, psen@ece, srodriguez@pstat, mturk@cs}.ucsb.edu

Abstract

Algorithms based on RANSAC that estimate models using feature correspondences between images can slow down tremendously when the percentage of correct correspondences (inliers) is small. In this paper, we present a probabilistic parametric model that allows us to assign confidence values for each matching correspondence and therefore accelerates the generation of hypothesis models for RANSAC under these conditions. Our framework leverages Extreme Value Theory to accurately model the statistics of matching scores produced by a nearest-neighbor feature matcher. Using a new algorithm based on this model, we are able to estimate accurate hypotheses with RANSAC at low inlier ratios significantly faster than previous state-of-the-art approaches, while still performing comparably when the number of inliers is large. We present results of homography and fundamental matrix estimation experiments for both SIFT and SURF matches that demonstrate that our method leads to accurate and fast model estimations.

1. Introduction

Many applications in computer vision use image correspondences to estimate important model parameters such as homographies, fundamental matrices, and others. However, these correspondences can often be "corrupted" by measurement noise or features that do not comply with the "true" model to be estimated, *i.e*., outliers. Random Sample Consensus (RANSAC) [7] has been the method of choice to estimate model parameters in the presence of outliers, and many improvements have been proposed to increase its speed and its accuracy, *e.g*., [6, 14, 15, 16, 20].

Many methods improve RANSAC by exploiting prior information such as matching scores [2, 5, 9, 19] or geometrical cues [4, 13, 16] in order to bias the generation of hypotheses (models) with matches that are more likely to be correct, hence avoiding outliers as much as possible. However, even these state-of-the-art approaches are slow when the percentage of inliers in the correspondences is low (*e.g*., $< 10\%$), as can happen in many real-world situations. This slowdown is caused by a substantial increase in the number of iterations required for convergence.

In this work, we focus on the problem of increasing the speed of RANSAC in these conditions. Our method extracts information from the matching scores that are available in many computer vision applications to compute a correctness confidence for the matches. Specifically, our contributions are:

1. A new probabilistic parametric model for matching scores generated by a nearest-neighbor matcher that is based on *extreme value theory* [3] and which accurately models the distribution of the lowest scores.
2. EVSAC, a novel algorithm that leverages our probabilistic framework to assign confidence values to each match in order to accelerate accurate hypothesis generation in RANSAC.

2. Previous work

In this section we review previous approaches that tackle the problem of generating good hypotheses in RANSAC. Since our method uses matching scores, we focus on approaches that do the same. Given the image correspondences and their matching scores (typically computed using a distance or similarity metric), these methods model the statistics of the scores to assess the "correctness" of a match. Note we use the term "match" to refer to a feature correspondence between the query and reference images.

Several approaches that speed up the generation of hypotheses compute a correctness confidence by attempting to model the distributions of matching scores produced by correct and incorrect matches. Tordoff and Murray [19] model these distributions in Guided-MLESAC (GMLESAC) from data pre-labeled as correct and incorrect by fitting appropriate curves in an offline stage. Goshen and Shimshoni [9] model these distributions in BEEM by using a non-parametric kernel density estimation and considering Lowe's ratio [11] as the random variable.

Brahmachari and Sarkar [2] compute a confidence for every match on the fly by using the closest matching scores.

Their BLOGS algorithm assigns a higher correctness confidence when the best matching score is far from the two second-best scores. Chum and Matas [5] take a different approach in PROSAC, where promising models are generated and tested earlier by sampling from a subset containing ranked correspondences by a quality measure. PROSAC starts with a subset of good matches and progressively expands it until convergence. Although PROSAC requires a good quality measure to rank correspondences for success, it is guaranteed to perform no worse than random sampling.

Fragoso and Turk [8] also compute a confidence on the fly by using the closest-matching scores. Specifically, they employ a heuristic that models the correct matches using information from a tail distribution and use this simple confidence metric to guide sampling. While we use their confidence values to preclassify correct/incorrect matches as input to our approach, our algorithm takes a different approach, focusing on modeling the entire nearest-neighbor matching process from all the data. By leveraging extreme value theory, we can accurately model the minima for all features and use it effectively for accelerating the hypothesis generation.

Finally, there is also work that uses extreme value theory (EVT) for modeling the tail of the underlying distribution to predict the correctness of a classifier, such as Meta-Recognition of Scheirer *et al*. [17]. In contrast, we propose a fundamentally different application of EVT that models the minimum scores produced by a nearest-neighbor feature matcher, which we represent as a stochastic process. We show that EVT can be used to estimate the distributions for both correct and incorrect matches using our proposed mixture model, which then can be used to compute confidence values to accelerate hypothesis generation.

3. Modeling the matching scores

3.1. The matching process

Given a pair of images, *i.e*., a query and a reference image, we first detect image features (interest points or keypoints) on both using standard techniques (e.g., [1, 11]). For each feature, we then compute its descriptor (SIFT [11] or SURF [1]) and use the photometric information captured in these descriptors to obtain matches or correspondences between the query image and the reference. Formally, for every query feature i we first compute the distance between the query descriptor $\mathbf{q}_i$ and each of the reference descriptors $\mathbf{r}_j$ to get a matching score $s_{i,j} = d(\mathbf{q}_i, \mathbf{r}_j)$. We then select the reference feature with the minimum score as the best match, satisfying the nearest-neighbor rule:

$$j^\star = \arg\min_j \{s_{i,j}\}_{j=1}^m . \tag{1}$$

When using a similarity metric instead of a distance, the algorithm seeks the maximum score; in this paper we will focus on the distance metric since several descriptors are compared using a Euclidean distance.

It is well known that the matching process given by Eq. 1 can return either correct or incorrect matches. An inlier (correct) match is one where the associated reference and query features both specify to the same physical location in the scene, while outlier (incorrect) matches are those that refer to different features in the scene yet produced a lower matching score. Incorrect matches could be due to several factors, such as repeating textures in the scene, features in the query image that are not visible in the reference, changes of lighting or shading, and others. Although descriptors are typically designed to be as invariant as possible to these effects, this problem still occurs quite often. In fact, the majority of nearest matches found for features in real-world images are typically outliers.

The processes that corrupt matches are complex and hard to model. Hence, we take a stochastic approach and model the probabilistic behavior of the matching scores. We represent the nearest-neighbor matching process as comprising two random processes: one that produces correct matches and another that produces incorrect matches for each query feature. These correct and incorrect matching scores are then merged together into the sequence $\{s_{i,j}\}_{j=1}^m$, where the matching score of the correct match (if it exists) may or may not be the smallest in the list. We now describe our probabilistic framework to model these distributions.

3.2. Our probabilistic model

Formally, the nearest-neighbor matcher can be modeled with two stochastic processes, one producing independent correct matching scores with a distribution F_c and another producing independent incorrect matching scores with a distribution $F_{\bar{c}}$ (see Fig. 1). Note we write probability densities (pdf's) with lower case letters and distribution functions (cdf's) with capital letters.

Because incorrect matches can have lower scores than correct matches, there is overlap between F_c and $F_{\bar{c}}$. Therefore, the minimum score from a nearest-neighbor matcher might be produced by either F_c or $F_{\bar{c}}$. If we could tell which of these two distributions produced the minimum, we would know if the minimum corresponds to a correct match or not.

First, we consider the distribution of incorrect matches. If we have m features in our reference image and assume that there is only one correct matching feature, then the matching process will produce $m - 1$ "incorrect" scores drawn from distribution $F_{\bar{c}}$ for each query feature. Since our nearest-neighbor matcher will only consider the minimum score, if an incorrect score is selected as the minimum then it will follow a distribution that models the *minimum* of $F_{\bar{c}}$. But what is this distribution? To answer this question, we draw insight from one of the classical theorems in extreme value theory (see, *e.g*., [3]):

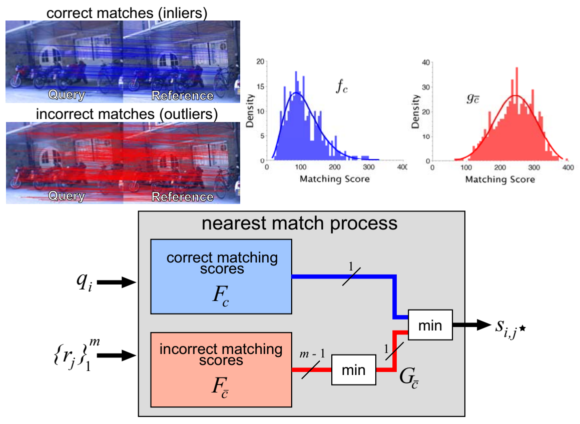


Figure 1. Overview of the matching process. **(top)** Given query and reference images where correct matches have been pre-identified, we show the pdf's of the matching scores for both correct (f_c) and incorrect matches ($g_{\bar{c}}$). **(bottom)** We pose the process of matching a query descriptor $\mathbf{q}_i$ to a set of reference descriptors $\{\mathbf{r}_j\}_{j=1}^m$ in a probabilistic framework. First, a random process generates *at most* one correct matching score using cdf F_c. Another random process generates at least $m-1$ incorrect matching scores using distribution $F_{\bar{c}}$. From this last set of incorrect matches the minimum is taken, modeled by distribution $G_{\bar{c}}$. Finally, the minimum of these two outputs is the best matching score $s_{i,j^\star}$. This process therefore models a nearest-neighbor matcher. Our work leverages extreme value theory to model this matching process without knowing the distributions *a priori*, and computes the confidence that score $s_{i,j^\star}$ is from a correct match.

Theorem: 1. *Let X_i be a sequence of i.i.d. random variables and let $M_n = \max\{X_1, \ldots, X_n\}$ denote the maximum. If there exist sequences of normalizing constants $a_n > 0, b_n \in \mathbb{R}$, and a non-degenerate probability distribution function, G, such that*

$$\mathbb{P}(a_n^{-1}(M_n - b_n) \leq x) \rightarrow G(x) \;\; as \;\; n \rightarrow \infty \qquad (2)$$

then $G(x)$ is of the same type as one of the three extremal-type distributions: Gumbel, Fréchet, and Weibull.

Intuitively, Theorem 1 states that the statistics of maxima M_n converges to distribution G (which can be Gumbel, Fréchet, or Weibull) asymptotically when the size of the sequence goes to infinity (*i.e.*, a large sequence). Although the theorem considers the maximum of a sequence, it can be trivially applied to the minimum as well, since a minimization problem can be recast as a maximization: $\max_i\{-X_i\} = \min_i\{X_i\}$. Therefore, we can apply this theorem to model the minima of $F_{\bar{c}}$ as distribution $G_{\bar{c}}$.

In order to derive analytically which of the three extremal-type distributions to use for $G_{\bar{c}}$, we technically need full knowledge of $F_{\bar{c}}$ which we do not have *a priori* in our application. However, we can use the Generalized Extreme Value distribution (GEV), which unifies the three extremal-type distributions, to address this issue, so $G_{\bar{c}}(s) = \text{GEV}(s; \mu, \sigma, \xi)$ (see [3] or supp. material).

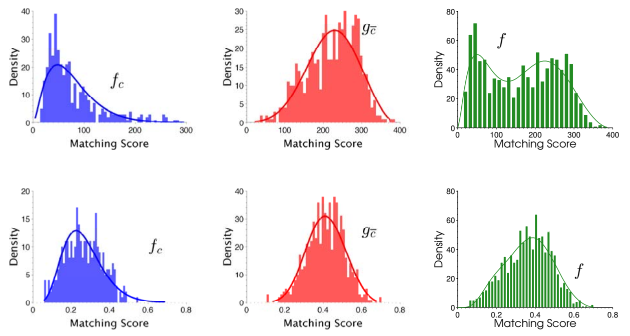


Figure 2. **(left)** Fitted Gamma distributions to matching scores from pre-identified correct matches. **(center)** Fitted GEV distributions to matching scores from pre-identified incorrect matches. **(right)** The histograms show the distribution of all the best matching scores (which include both correct and incorrect matches) and the continuous curve shows that our mixture model of the two densities is a good fit. In all cases, SIFT matches are shown on top and SURF matches on the bottom.

To model the correct matching distribution F_c, we assume that the statistics of the correct matching scores will be skewed towards the minimum since many of the state-of-the-art descriptors such as SIFT or SURF are designed to be as invariant as possible, resulting in low scores. Therefore, we can expect distributions with longer right-tails, and so we pose that the correct matching scores follow a Gamma distribution, *i.e.*, $F_c(s) = \text{Gamma}(s; \alpha, \beta)$. Fig. 2 shows fitted corresponding distributions to a set of pre-identified correct/incorrect matching scores for SIFT and SURF matches to demonstrate that the models we have selected for F_c and $G_{\bar{c}}$ are reasonable in practice.

So now we have distribution F_c that produces the inlier scores and $G_{\bar{c}}$ that produces the best outlier score for each feature. The nearest neighbor matcher then selects between the two depending on which one is smaller. We observe that the probability of selecting one distribution or the other is given by the inlier ratio ε, which states the percentage of the nearest matches that are actually inliers for all query features. Therefore, the statistics of nearest-neighbor score $s_{i,j^\star}$ can be modeled by the following mixture distribution:

$$F = \varepsilon F_c + (1-\varepsilon) G_{\bar{c}} \qquad (3)$$

where we use the inlier ratio ε as the mixing parameter between the two distributions. The plots on the right in Fig. 2 show that this mixture model fits the measured minimum values for each feature reasonably in real examples.

We can then use this mixture model to calculate weights or correctness confidences as a function of a matching score for every correspondence by computing the posterior probability from Eq. (3):

$$\begin{aligned} p(c|s) &= \frac{p(s|c)p(c)}{p(s|c)p(c) + p(s|\bar{c})p(\bar{c})} \\ &= \frac{\varepsilon f_c}{\varepsilon f_c + (1-\varepsilon) g_{\bar{c}}} \end{aligned} \tag{4}$$

where $p(c) = \varepsilon$, $p(s|c) = f_c$, and $p(s|\bar{c}) = g_{\bar{c}}$. In the section that follows, we describe an algorithm that automatically estimates the necessary parameters for our model in order to use it to accelerate model generation.

3.3. Building the probabilistic model from the data

We now introduce the EVSAC algorithm (summarized in Algorithm 1), which was inspired by BEEM's prior search method [9] and which estimates the parameters for our theoretical model from real image data. EVSAC requires the image feature correspondences $\{\mathbf{x} \leftrightarrow \mathbf{x}'\}_{i=1}^{n}$, and the k nearest neighbor matching scores $\{s_{i,1:k}\}_{i=1}^{n}$ sorted in an ascending order for every i-th correspondence. We denote the r-th element in the sorted sequence $s_{i,1:k}$ for the i-th match as $s_{i,(r)}$, then $s_{i,(1)} = s_{i,j^\star}$. The goal of EVSAC is to produce the set of weights $\{w\}_{i=1}^{n}$ for every correspondence, which will be used for generating hypotheses.

Our algorithm begins by computing the distributions for correct and incorrect matches for the data provided. In order to start the process, we need a correct-match predictor to preliminarily label each match as correct or incorrect (*e.g.*, Lowe's ratio [11] or MR-Rayleigh [8]). We then fit a two-parameter Gamma distribution to the data identified as correct to estimate F_c. Subsequently, we use all the second nearest matching scores, *i.e.*, $s_{i,(2)}, \forall i = 1, \ldots, n$, to find the three parameters of the GEV distribution for $G_{\bar{c}}$. We use the second nearest matching scores (instead of all the matches labeled incorrect by the predictor) since in practice this results in a better approximation of the true GEV, as if we had a perfect incorrect match detector (see Fig. 3).

Next, we estimate ε, which is the mixing parameter between these two computed distributions. To find this parameter, we build the empirical cdf of F, see Eq. (3), using all the lowest matching scores, *i.e.*, $s_{i,(1)}$. Subsequently, we then solve the following constrained least-squares problem:

$$\begin{aligned} \underset{\mathbf{y}}{\text{minimize}} \quad & \frac{1}{2}\|A\mathbf{y} - \mathbf{b}\|_2^2 \\ \text{subject to} \quad & \mathbf{1}^{\mathrm{T}}\mathbf{y} = 1 \\ & \mathbf{0} \preceq \mathbf{y} \preceq \mathbf{u} \end{aligned} \tag{5}$$

where the symbol $\preceq$ indicates entrywise comparison, and

$$A = \begin{bmatrix} F_c(s_1) & G_{\bar{c}}(s_1) \\ \vdots & \vdots \\ F_c(s_L) & G_{\bar{c}}(s_L) \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} F(s_1) \\ \vdots \\ F(s_L) \end{bmatrix},$$

$$\mathbf{y} = \begin{bmatrix} \varepsilon \\ \varepsilon' \end{bmatrix}, \text{ and } \mathbf{u} = \begin{bmatrix} \tau \\ 1 \end{bmatrix}.$$

Algorithm 1 EVSAC

```
Require: ${\mathbf{x} \leftrightarrow \mathbf{x}'}_{i=1}^{n}$ and $\{s_{i,1:k}\}_{i=1}^{n}$
Ensure: $\{w_i\}_{i=1}^{n}$ and $\{p_i\}_{i=1}^{n}$
1: $\mathbf{v} \leftarrow$ Predict$(\{s_{i,1:k}\}_{i=1}^{n})$
2: $(\alpha, \beta) \leftarrow$ FitGamma$(\{s_{i,(1)}$ such that $v_i = 1\})$
3: $(\mu, \sigma, \xi) \leftarrow$ FitGEV$(\{s_{i,(2)}\})$
4: Calculate the empirical cdf using $s_{i,j^\star}$
5: Find $\varepsilon$ by solving (5)
6: Calculate posterior-weights $p_i$ using Eq. (4)
7: Calculate weights $w_i$ using Eq. (6)
8: Use the weights $w_i$ for generating hypotheses
```

The first entry of vector $\mathbf{u}$, *i.e.*, τ, is the inlier ratio computed by the predictor in step 1. We set this upper bound to the estimate of ε as in practice the predictor introduces some false-positives (false-alarms) and so the true inlier ratio must be less than or equal to this number.

Intuitively, the solution to (5) is the mixture parameter that produces the lowest error between the observations (the minimum scores returned by the nearest-neighbor matcher for all query features) and the mixture model that combines our estimates for the correct and incorrect distributions. Once this has been found, we can use Bayes' theorem in Eq. (4) to calculate a correctness confidence for each correspondence (step 6). Although the confidences determined by the posterior lead to speed ups in the convergence of the model estimation, we noticed that the overlap between distributions causes some incorrect matches to be assigned a high confidence, costing extra iterations in RANSAC. To alleviate this problem, we calculate an "agreement" between the predictor in step 1 and the posterior. Assuming that the predictor returns a binary vector $\mathbf{v}$ where 1 denotes correct match and 0 otherwise, we calculate the final weights as

$$w_i = p_i v_i, \tag{6}$$

where p_i is the posterior for the i-th match. In the case where no agreement exists, *i.e.*, all weights are zero, or when the agreement within some number of iterations did not converge to a solution, then the confidences p_i computed with the posterior can be used. Finally, we use weights w_i to sample matches to generate hypotheses.

4. Experiments

We present in this section two different experiments to assess the performance of EVSAC. The first experiment evaluates the accuracy of calculating the parameters of our probabilistic framework, *i.e.*, the distribution parameters and the mixing parameter ε. The second experiment measures the performance of our approach against well-established non-uniform sampling algorithms for the estimation task of homographies and fundamental matrices.

Table 1. Estimation of ε comparison: $\hat{\varepsilon}$ is the estimation with τ set as an upper bound (see Eq. (5)), and $\tilde{\varepsilon}$ is without. The upper bounded estimate tends to provide more accurate estimations.

Image Pairs	ε	$\hat{\varepsilon}$	$\tilde{\varepsilon}$
Oxford-Bark (1-4 SURF)	0.0131	0.0141	0.1870
Oxford-Boat (1-6 SURF)	0.0257	0.0270	0.1429
Oxford-Bark (1-3 SIFT)	0.0479	0.0438	0.1291
Oxford-Trees (1-6 SIFT)	0.1028	0.1119	0.2467
Strecha-Brussel (2-3 SIFT)	0.1855	0.2067	0.2263
Strecha-Brussel (1-2 SURF)	0.2964	0.3115	0.3632

The estimation experiments consider cases ranging from a very low inlier-ratio to cases where the inlier ratio is larger, which are more commonly presented in previous work.

Datasets: We use Oxford datasets [12] and Strecha's multi-view stereo datasets [18] for our experiments. Each Oxford dataset contains a reference image and five query images, as well as five homographies that relate the reference image and the query images. The three Strecha's datasets provide the set of camera parameters, *i.e.*, intrinsic and extrinsic matrices, for every image.

To generate the ground truth of image feature correspondences, we first detected approximately a thousand keypoints per image by using OpenCV's Hessian keypoint detector and also computed their SIFT [11] and SURF [1] descriptors using OpenCV's implementation. For the Oxford datasets, we exploited the homographies provided and mapped the reference image keypoints onto every query image. Subsequently, we then selected for every query keypoint the closest mapped reference keypoint with a minimum Euclidean distance less than five pixels. When no reference keypoint was found with this process, then that query keypoint did not have a true match. We then manually verified the result of this process, and used it as our ground truth for the Oxford datasets.

For the multi-view dataset, we calculated the fundamental matrices between subsequent images (first and second image, second and third image, and so on) using the provided intrinsic and extrinsic matrices (see [10], pg. 246). We then matched the keypoints on the subsequent images using their descriptors, and filtered out those query keypoints that produced a distance greater than or equal to 3 pixels from the epiline. The resulting set of matches was verified manually to ensure that only correct matches were left.

4.1. Parameter estimation experiment

We now present an evaluation of the performance of our algorithm to find the parameters of our probabilistic framework: ε, and the distribution parameters using the predictor from [8] only as the predictor in step 1. We compared the estimated parameters against the parameters obtained assuming that we had a perfect correct match detector.

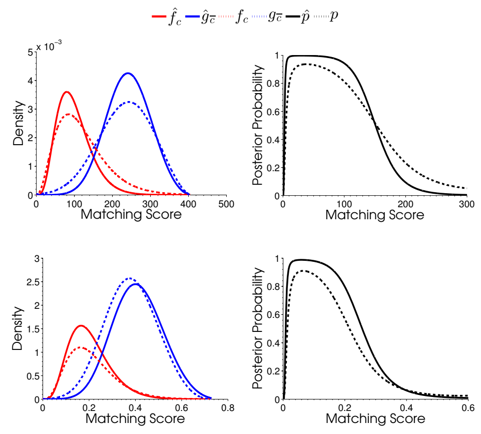


Figure 3. Comparison of the mixture of densities and posterior probability computed using EVSAC against the ground truth for a pair of images with SIFT matches (top row) and SURF matches (bottom row). Our density estimations $\hat{f}_c$ and $\hat{g}_{\bar{c}}$ are close to the densities f_c and $g_{\bar{c}}$ computed with an oracle. In the second column, we compare our estimated posterior probability $\hat{p}$ with the posterior p computed with the oracle.

We first examine the accuracy of the estimation of ε in Table 1. The estimate of ε using the upper bound in vector $\mathbf{u}$ used in (5), $\hat{\varepsilon}$, tends to be closer to the real value, while the estimate without the upper bound ($\tilde{\varepsilon}$) can overshoot sometimes.

Next, we examine the quality of our estimation of the different probability densities and the posterior we use to compute the weights w_i. In the first column of Fig. 3, we can observe that our algorithm (continuous curves) is able to approximate with a good accuracy the mixture of densities obtained with the ground truth data (dashed curves). In the second column, we present the posterior probabilities computed from the estimated model (continuous curves) and the posterior obtained from the ground truth (dashed curves). This means that our algorithm estimates an accurate posterior that essentially maximizes the information in the matching score when computing a confidence value.

4.2. Homography experiment

In this experiment we assess the performance of our non-uniform sampling algorithm for estimating homographies. We implemented the probabilistic model parameter estimation in Matlab, and produced the set of weights for every correspondence. We also computed the weights produced by BEEM, BLOGS, and GMLESAC in Matlab. These weights were then read by our C++/OpenCV implementation of the respective algorithms: RANSAC (guided by our weights), Guided-MLESAC [19], BEEM's prior estimation step [9], and BLOGS' global search mechanism [2]. All these sampling algorithms (along with PROSAC [5] and classical RANSAC) were then included in a classical hypothesis-test loop, where the support was always being maximized, and a solution was considered "good" if it satisfied the maximality constraint, *i.e.*, the constraint that a good hypothesis was generated within a certain number of iterations (see [5] for more details on this constraint). The homography was computed using the OpenCV findHomography( ) function without the RANSAC option. An inlier was considered if the reprojection error of the homography was less than 5 pixels. The algorithms were allowed to run until a maximum number of iterations (hypothesis-test loops) calculated adaptively is reached, and the algorithm converged when 90% of the inliers (correct-matches) were detected. The found hypothesis was refined afterwards using a non-linear method.

The results of this experiment are summarized in Table 2. The Oxford datasets used for the experiment presented very challenging scenarios, where the inlier-ratios ε ranged from 1-10% for SIFT and SURF matches. The experiments were run 300 times. We present the average number of inliers detected; the average RMS reprojection error in pixels w.r.t. to the error achieved by the ground truth data; the average number of models/hypotheses generated; the average time in milliseconds; the average Frobenius norm of the error between estimated homography and the computed homography with the ground truth (f-error); and the percentage of "good" runs where each algorithm converged. The results are sorted in ascending order by the inlier-ratio. We can observe that our algorithm (EVSAC) tends to perform overall faster when the inlier ratio is very low (see rows **A**, **B**, **C**, **D**, and **E**), and performs equivalent or faster than BEEM and BLOGS as soon as the inlier-ratio increased (see rows **F**, **G**, **H**, **I**). PROSAC and GMLESAC struggled to converge fast when the inlier-ratio was very low ($\varepsilon < 11\%$).

To measure the effect of the inlier-ratio on the convergence time, we used the entire Oxford-Trees dataset, where we observed that the inlier-ratio decreased as the blurring increased in a systematic manner. In Fig. 4 we present a plot of the convergence time as a function of the inlier-ratio. We only considered BEEM, BLOGS, and EVSAC because the other methods did not converge when the inlier ratio was low. We can observe that EVSAC tends to converge faster when the inlier-ratio is less than 0.1 and performs equivalently when the inlier-ratio starts to increase.

4.3. Fundamental matrix experiment

In this experiment, we assess the performance of EVSAC in estimating fundamental matrices. We have the same implementation as in the homography experiment using Matlab and C++/OpenCV implementation. The fundamental matrix was computed using the 7-point algorithm provided by the OpenCV findFundamentalMat( ) function without the RANSAC option. When the function returned more than one solution, we kept the matrix that had the biggest inlier support. A match was considered to be an inlier when the distance between a query keypoint and the epiline was less than a pixel.

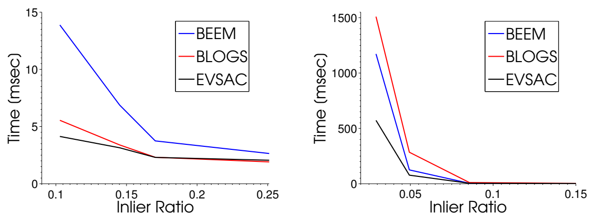


Figure 4. Convergence time as a function of the inlier ratio for SIFT matches (left) and SURF matches (right) on the Oxford-Trees dataset.

The results of this experiment are shown in Table 3. Strecha's multi-view dataset provided different relatively high inlier ratios; ranging from 29-43% for SIFT and SURF matches. The experiments were run 300 times, and we present the same quantities as in the homography experiment. We can observe that in all the experiments our algorithm (EVSAC), BEEM, BLOGS, and PROSAC were the fastest regardless of the descriptor used. GMLESAC was the second fastest algorithm and RANSAC was the slowest. All of the algorithms converged in all the trials, and provided an accurate estimation as the Frobenius norm indicates. This confirms that our algorithm can perform equivalently to other methods when the inlier-ratio is not so low.

5. Conclusions and future directions

We have introduced a probabilistic framework that uses extreme value theory to model the statistics of the best matching scores selected by a nearest-neighbor feature matcher. We then use the posterior probability of our mixture model to compute the correctness weight for every correspondence and thereby accelerate model generation. Our homography and fundamental matrix estimation experiments showed that our algorithm (EVSAC) performs robustly and is faster than existing state-of-the-art methods (BEEM, BLOGS, PROSAC, and GMLESAC) when the inlier-ratio is low ($< 11\%$). Moreover, the experiments also demonstrated that EVSAC is comparable to these other methods when the inlier-ratio increases ($> 20\%$). The results suggest that EVSAC is a very useful algorithm for applications that require fast and robust model estimation in complex environments where the number of inliers is low.

This work opens the possibility of using extreme value theory for developing models for related problems that involve a minimum (or maximum) which can be cast as stochastic processes. For example, we are interested in extending this work to similarity metrics and to develop statistical tools for analyzing and designing descriptors/metrics for these applications.

Table 2. Homography estimation results for SIFT and SURF matches. The results are sorted by inlier-ratio (ε) in ascending order. EVSAC performed well when the inlier-ratio is low, and performed equivalently when the inlier-ratio increased.

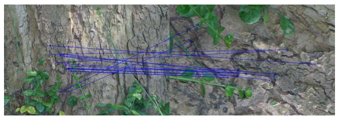
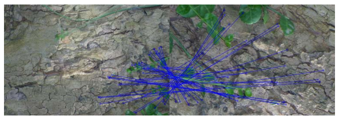
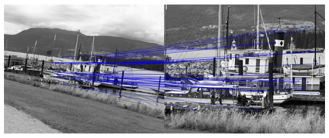
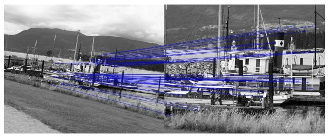
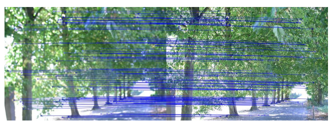
		RANSAC	BEEM	BLOGS	PROSAC	GMLESAC	EVSAC
A: $\varepsilon = 0.01$, $n = 992$, SURF	inliers	NA	14 ± 0	14 ± 0	14 ± 0	14 ± 0	14 ± 0
	error	NA	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
	models	NA	1443	2524	4	1521	11
	time	NA	563.1	1008.3	2	511	4.2
	f-error	NA	0	0	0	0	0
	good runs	0 %	100%	96%	0.33%	0.33%	100%
B: $\varepsilon = 0.02$, $n = 992$, SIFT	inliers	10 ± 2	12 ± 3	12 ± 2	10 ± 2	11 ± 3	12 ± 2
	error	0.36 ± 0.1	0.1 ± 0.02	0.1 ± 0.04	0.37 ± 0.03	0.24 ± 0.04	0.16 ± 0.03
	models	2436910	41	17	2752900	10044	10
	time	338618	13.3	5.3	375482	1446.4	3.3
	f-error	94.4	6.6	10.7	136.7	37.8	12.1
	good runs	37%	100%	100%	100%	100%	100%
C: $\varepsilon = 0.035$, $n = 992$, SURF	inliers	27 ± 4	24 ± 2	22 ± 4	27 ± 4	29 ± 2	24 ± 2
	error	0.1 ± 0.03	0.1 ± 0.01	0.1 ± 0.01	0.1 ± 0.02	0.03 ± 0.05	0.2 ± 0.1
	models	1313580	3741	4072	1458250	209963	1965
	time	286881	1172.3	1509.5	284838	28092.1	572.3
	f-error	2.5	2	9	3.2	2.9	4.2
	good runs	89%	100%	99%	100%	100%	100%
D: $\varepsilon = 0.04$, $n = 981$, SURF	inliers	38 ± 6	39 ± 2	39 ± 2	38 ± 6	38 ± 9	39 ± 2
	error	0.01 ± 0.1	0.04 ± 0.02	0.01 ± 0.001	0.02 ± 0.1	0.01 ± 0.2	0.02 ± 0.01
	models	697832	39	82	655073	4151	4
	time	155067	10.9	21.5	145207	838.4	1.3
	f-error	1.1	0	0.1	1.3	3.9	0.1
	good runs	91.33%	100%	100%	98%	99%	100%
E: $\varepsilon = 0.05$, $n = 807$, SURF	inliers	38 ± 6	40 ± 2	40 ± 2	38 ± 6	33 ± 9	40 ± 2
	error	1.5 ± 0.2	0.04 ± 0.03	0.002 ± 0.001	0.45 ± 0.53	0.02 ± 0.13	0.02 ± 0.01
	models	304532	149	355	321952	4713	92
	time	61170.4	42.9	98.5	56176.5	2111.9	26.3
	f-error	0.3	0.3	0.2	0.6	35.7	0.8
	good runs	100%	100%	100%	100%	100%	100%
F: $\varepsilon = 0.05$, $n = 807$, SIFT	inliers	37 ± 6	41 ± 3	41 ± 4	36 ± 6	41 ± 3	41 ± 3
	error	2.08 ± 0.3	0.08 ± 0.02	0.002 ± 0.01	0.17 ± 0.04	0.05 ± 0.02	0.04 ± 0.02
	models	221667	71	14	218811	341	22
	time	42002.5	15.7	3.5	40750.7	67.5	5.4
	f-error	8.4	1.7	0.7	8.4	1.1	0.1
	good runs	100%	100%	100%	100%	100%	100%
G: $\varepsilon = 0.10$, $n = 992$, SURF	inliers	58 ± 23	81 ± 8	81 ± 9	60 ± 23	81 ± 8	82 ± 7
	error	0.003 ± 0.06	0.02 ± 0.01	0.03 ± 0.009	0.02 ± 0.06	0.02 ± 0.006	0.03 ± 0.004
	models	4899	177	193	4507	918	73
	time	1738.9	49.6	53.6	1616.2	226.5	21.9
	f-error	16	0.8	0.7	13.3	0.6	0.4
	good runs	100%	100%	100%	100%	100%	100%
H: $\varepsilon = 0.103$, $n = 992$, SIFT	inliers	62 ± 16	82 ± 9	82 ± 9	69 ± 16	80 ± 8	82 ± 8
	error	0.1 ± 0.05	0.02 ± 0.016	0.05 ± 0.008	0.09 ± 0.04	0.03 ± 0.003	0.03 ± 0.003
	models	4727	72	26	3649	773	8
	time	1183.3	13.9	5.5	834	120.1	4.1
	f-error	8.9	0.7	1.1	4.5	1	0.8
	good runs	100%	100%	100%	100%	100%	100%
I: $\varepsilon = 0.103$, $n = 992$, SIFT	inliers	70 ± 15	81 ± 11	81 ± 10	77 ± 13	80 ± 12	82 ± 10
	error	0.09 ± 0.02	0.03 ± 0.007	0.003 ± 0.002	0.05 ± 0.001	0.02 ± 0.015	0.002 ± 0.02
	models	4675	13	13	1961	89	13
	time	1032.4	3.4	3.3	507.2	18.4	3.4
	f-error	5.5	0.4	0.8	3.3	1.4	1.3
	good runs	100%	100%	100%	100%	100%	100%

Table 3. Fundamental matrix estimation results for SIFT and SURF matches. The results are sorted by inlier-ratio (ε) in ascending order. EVSAC performed as fast as BEEM, BLOGS, and PROSAC.

		RANSAC	BEEM	BLOGS	PROSAC	GMLESAC	EVSAC
A: $\varepsilon = 0.29$, $n = 992$, SURF	inliers	221 ± 30	236 ± 32	241 ± 34	237 ± 34	229 ± 31	237 ± 34
	error	0.5 ± 1.5	0.4 ± 2.2	0.3 ± 0.7	0.4 ± 1.0	1.0 ± 7	0.3 ± 0.4
	models	160	3	3	3	16	2
	time	1349	69	64	67	497	64
	f-error	0.07	0.05	0.1	1.0	0.06	0.08
	good runs	100%	100%	100%	100%	100%	100%
B: $\varepsilon = 0.33$, $n = 992$, SURF	inliers	261 ± 44	278 ± 39	283 ± 38	288 ± 35	262 ± 43	279 ± 38
	error	0.8 ± 5	0.4 ± 1.8	0.3 ± 0.6	0.4 ± 1.3	0.4 ± 1.4	0.4 ± 2
	models	18	1	1	1	15	1
	time	214	60	61	59	95	51
	f-error	0.002	0.002	0.002	0.001	0.001	0.002
	good runs	100%	100%	100%	100%	100%	100%
C: $\varepsilon = 0.40$, $n = 992$, SIFT	inliers	306 ± 34	321 ± 42	335 ± 42	331 ± 41	305 ± 36	330 ± 40
	error	0.3 ± 0.5	0.3 ± 0.6	0.2 ± 0.8	0.2 ± 0.3	0.6 ± 5	0.2 ± 0.4
	models	59	3	3	3	44	3
	time	593	83	75	78	410	81
	f-error	1.0	0.3	0.1	0.1	0.1	0.1
	good runs	100%	100%	100%	100%	100%	100%
D: $\varepsilon = 0.43$, $n = 992$, SIFT	inliers	340 ± 55	368 ± 50	380 ± 38	387 ± 36	353 ± 51	373 ± 50
	error	0.1 ± 0.25	0.08 ± 0.13	0.07 ± 0.15	0.06 ± 0.07	0.13 ± 0.32	0.08 ± 0.11
	models	5	1	1	1	4	1
	time	117	78	76	80	108	78
	f-error	0.002	0.003	0.002	0.002	0.002	0.002
	good runs	100%	100%	100%	100%	100%	100%

Acknowledgments. VF and SR would like to thank UC MEXUS-CONACYT (Fellowships 212913 and 311443) for the funding. This research was supported in part by NSF under award number 1219261.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. 2, 5

[2] A. S. Brahmachari and S. Sarkar. BLOGS: Balanced local and global search for non-degenerate two view epipolar geometry. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2009. 1, 5

[3] E. Castillo, A. S. Hadi, N. Balakrishnan, and J. M. Sarabia. *Extreme value and related models with applications in engineering and science*. Wiley-Interscience, 2005. 1, 2, 3

[4] T.-J. Chin, J. Yu, and D. Suter. Accelerated hypothesis generation for multistructure data via preference analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(4):625–638, 2012. 1

[5] O. Chum and J. Matas. Matching with PROSAC – progressive sample consensus. In *Proc. IEEE Computer Vision and Pattern Recognition*, 2005. 1, 2, 5, 6

[6] O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In *Proc. Pattern Recognition (DAGM Symposium)*, 2003. 1

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 1

[8] V. Fragoso and M. Turk. SWIGS: A Swift Guided Sampling Method. In *Proc. IEEE Computer Vision and Pattern Recognition*, 2013. 2, 4, 5

[9] L. Goshen and I. Shimshoni. Balanced exploration and exploitation model search for efficient epipolar geometry estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1230–1242, July 2008. 1, 4, 5

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

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 60(2):91–110, Nov. 2004. 1, 2, 4, 5

[12] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A Comparison of Affine Region Detectors. *Intl. Journal of Computer Vision*, 65(1-2):43–72, Nov. 2005. 5

[13] K. Ni, H. Jin, and F. Dellaert. GroupSAC: Efficient consensus in the presence of groupings. In *Proc. IEEE Intl. Conf. in Computer Vision*, 2009. 1

[14] R. Raguram and J.-M. Frahm. RECON: Scale-adaptive robust estimation via residual consensus. In *Proc. IEEE Intl. Conf. in Computer Vision*, 2011. 1

[15] R. Raguram, J.-M. Frahm, and M. Pollefeys. A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. In *Proc. European Conf. on Computer Vision*, 2008. 1

[16] T. Sattler, B. Leibe, and L. Kobbelt. SCRAMSAC: Improving RANSAC's efficiency with a spacial consistency filter. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2009. 1

[17] W. J. Scheirer, A. Rocha, R. J. Micheals, and T. E. Boult. Meta-Recognition: The Theory and Practice of Recognition Score Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1689–1695, Aug. 2011. 2

[18] C. Strecha, R. Fransens, and L. Van Gool. Combined depth and outlier estimation in multi-view stereo. In *Proc. IEEE Computer Vision and Pattern Recognition*, 2006. 5

[19] B. Tordoff and D. W. Murray. Guided sampling and consensus for motion estimation. In *Proc. European Conf. on Computer Vision*, 2002. 1, 5

[20] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Comput. Vis. Image Underst.*, 78(1):138–156, Apr. 2000. 1