Using Spatial Order to Boost the Elimination of Incorrect Feature Matches

Lior Talker
The University of Haifa
ltalke01@campus.haifa.ac.il

Yael Moses
The Interdisciplinary Center
Herzliya 46150, Israel
yael@idc.ac.il

Ilan Shimshoni
The University of Haifa
ishimshoni@mis.haifa.ac.il

Abstract

Correctly matching feature points in a pair of images is an important preprocessing step for many computer vision applications. In this paper we propose an efficient method for estimating the number of correct matches without explicitly computing them. In addition, our method estimates the region of overlap between the images. To this end, we propose to analyze the set of matches using the spatial order of the features, as projected to the x-axis of the image. The set of features in each image is thus represented by a sequence. This reduces the analysis of the matching problem to the analysis of the permutation between the sequences. Using the Kendall distance metric between permutations and natural assumptions on the distribution of the correct and incorrect matches, we show how to estimate the above-mentioned values. We demonstrate the usefulness of our method in two applications: (i) a new halting condition for RANSAC based epipolar geometry estimation methods that considerably reduce the running time, and (ii) discarding spatially unrelated image pairs in the Structure-from-Motion pipeline. Furthermore, our analysis can be used to compute the probability that a given match is correct based on the estimated number of correct matches and the rank of the features within the sequences. Our experiments on a large number of synthetic and real data demonstrate the effectiveness of our method. For example, the running time of the image matching stage in the Structure-from-Motion pipeline may be reduced by about 99% while preserving about 80% of the correctly matched feature points.

1. Introduction

Matching feature points between a pair of images is a fundamental problem in computer vision. The estimation of epipolar geometry between images [13, 22], 3D structure reconstruction (SfM) [29, 30], and scene recognition [17] are typical examples of useful tasks that are based on feature matching. While many methods for feature matching exist, the critical stage of filtering incorrect matches is costly when using algorithms such as Random Sample Consensus (RANSAC) [12, 13, 22, 3].

In this paper we propose to analyze the set of correct matches, without explicitly computing it, using the spatial order of the features in each image. We estimate the number of correct matches and the probability that a given match is correct. In addition, we estimate the overlap region of the pair of images and whether they overlap at all. Our method can be applied to sets of matching features irrespective of their descriptors (e.g., [21, 1, 4, 24]) or the matching method used to compute them. Our method can be used as a preprocessing step to improve the efficiency of existing methods such as RANSAC and SfM, as described below.

The basic idea is as follows. We represent the image features as sequences defined by their spatial order along the x-axis of the image. The matching between features in a pair of images induces a permutation that relates the spatial order in one image to that in the other image (see Fig. 1). The matching is analyzed using a measure of correlation between permutations, the Kendall distance metric [9, 10]. We use statistical assumptions on the distribution of correctly and incorrectly matched features; the spatial order of correctly matched features is usually preserved, while incorrectly matched features are expected to have random order. Note that the problem is not trivialized by these assumptions since simply computing the largest set of features that preserve their spatial order does not necessarily provide the correct set of matches (see discussion in Sec. 3). Hence, we also consider the interaction between correct and incorrect matches for obtaining our estimations.

Our estimation of the number of correct matches can be used to improve the running time of RANSAC methods [12, 13, 22, 3]. In adaptive RANSAC, a subset of matches in a pair of images is randomly sampled and used to compute the expected geometric transformation between the features (e.g., homography or epipolar geometry), which is then verified against all matches. The transformation with the largest set of *inliers*, features consistent with the computed geometric transformation, is chosen. Since the number of inliers is usually unknown, the number of required iterations is high. This increases the running time of RANSAC, which is its major drawback. Our estimate of the number of correct matches can be used to improve the running time by halting when a consensus set of this size is obtained (see Sec. 6).

Computing the structure of the scene and the cameras' parameters from a set of images using SfM methods (e.g., [29, 30]) is a fundamental problem in computer vision with many applications. Such methods, which typically require tens or hundreds of images, strongly rely on correct matches. A major time consuming step in SfM methods is the robust matching of features between image pairs, which is typically obtained by RANSAC. Our method can significantly shorten the SfM pipeline [29, 30] by running RANSAC only on image pairs with a sufficiently large number of correct matches (see Sec. 6).

We show that the number of correct matches can also be used to compute the probability that a given match is correct. One application of such probability is in improving the performance of guided RANSAC, where matches that are more likely to be correct are sampled more often (e.g., [13, 3, 22, 5, 7, 6]). In existing methods the probability that a given match is correct is based only on feature descriptors, whereas in our method it is based on the features' order in the image. Thus, our method by itself or in combination with appearance based methods give a more accurate probability for the correctness of a match than when using the method based only on feature descriptors (see details in Sec. 5).

The main contribution of the paper is a novel analysis of the correct and incorrect matches with respect to their spatial order in the images. Given a set of putative matches, our method efficiently estimates the number of correct ones, the region of overlap between a pair of images, and for each match the probability that it is correct. These estimations can be used to improve the efficiency of RANSAC and SfM methods, as demonstrated empirically in Sec. 6.

2. Related Work

Feature point matching between a pair of images has been studied extensively in computer vision. Over the years many interest point detectors and patch descriptors were proposed to detect and match the images of the same 3D points in the scene (e.g., [21, 1, 4, 24]). Appearance based methods, such as [15], discard descriptors by learning their success rate in matching between their corresponding features, in order to increase the proportion of correct to incorrect matches. Other methods use local geometric structures between a number of matches in order to decrease the probability of mismatching, e.g., [23].

A major drawback of the RANSAC method for filtering incorrect matches is its running time; for example, running RANSAC on a pair of images with 1000 matches may take a few seconds, which can be a major bottleneck in online applications involving large sets of images. To improve RANSAC's runtime, methods for guiding the sampling of matches according to their probability to be correct (instead of a uniform sampling) were proposed [13, 3, 22, 5, 7, 6]. For example, the similarity of feature appearance defined by Lowe's distance ratio between the closest and the second closest matches [21] is used as a match correctness likelihood by [13, 22]. This method ignores important geometric information. In our method, we use the spatial order of the matched features, which carries geometric information. A comparison between probabilities computed by our method and Lowe's distance ratio is provided in Sec. 6.

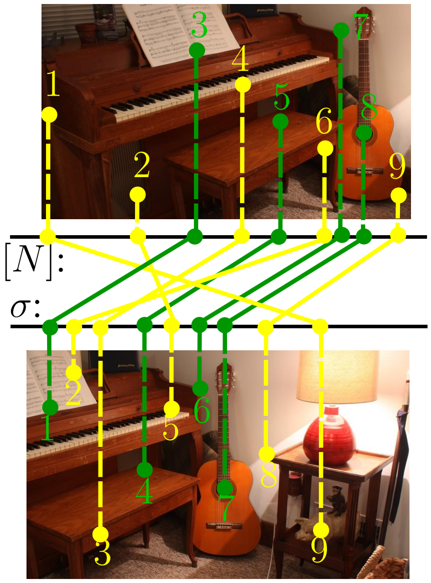


Figure 1. Correctly/incorrectly matched features are marked as green/yellow circles, respectively. The feature sequences are given by $[N] = \langle 1,2,3,4,5,6,7,8,9,10\rangle$ and $\sigma = \langle 9,5,1,3,4,2,6,7,8\rangle$. For example, $\sigma(1) = 9$ and $\sigma(5) = 4$. Note that the green lines (correct matches) do not intersect, while most of the yellow lines (incorrect matches) do.

An inlier rate estimation was also used by [20] to compute a homography transformation between two images, which is guaranteed to find an approximation to the global optimum. The rate is computed by counting the number of homographies that agree with each inlier rate. The drawback of the method is the search process, which is time consuming and is applicable only to homographies.

3. Problem Formulation

Let $\mathcal{M} = \{(p_i, q_j)\}$ be a set of putative matches between two feature sequences, $p_1, \ldots, p_N$ and $q_1, \ldots, q_N$, in a pair of images, I_1 and I_2, respectively. The index i of p_i represents the position (the rank) of the feature in the sequence of I_1, when sorted according to the x coordinate. Similarly, the index j of q_j corresponds to the rank in the feature sequence of I_2. For the rest of the paper, let us represent $\mathcal{M}$ by two sequences of indexes, $[N] = \langle 1, \ldots, N \rangle$ and σ, where $(p_i, q_{\sigma(i)}) \in \mathcal{M}$ (see Fig. 1).

The set of matches, $\mathcal{M}$, can be partitioned into two disjoint sets, the correct ("**G**ood") and the incorrect ("**B**ad") matches. Let G and B be the sets of indexes in I_1 for which the matching is correct and incorrect, respectively. In this case, the number of matches is given by $N = N_G + N_B$, where $N_G = |G|$ and $N_B = |B|$.

We analyze the spatial orders of matched features in the pair of images, which is represented by the permutation σ. We use statistical assumptions on the distribution of the correct and incorrect matches, which allows us to estimate the number of correct matches, N_G without explicitly computing the set of correct matches. In addition, we propose a method to compute the probability that a given match is correct.

To do so, we use the Kendall [18] distance that corresponds to the sum of order inversions in σ. The two main statistical assumptions used are:

A1: The spatial order of the correctly matched features in I_1 is preserved in I_2. This assumption holds when (i) the cameras have no relative roll (rotation around the z-axis), and (ii) there are no "poles", i.e., features with significant difference in depth on different surfaces.

A2: The spatial order of incorrect matches is random. This can be justified since the spatial location of incorrectly matched features in different images is arbitrary.

Assumption **A1** mostly holds since in natural scenes there are few inversions between correct matches. (This assumption is often used in stereo matching algorithms e.g., [31, 2].) Moreover, the roll value is often available in the camera so it can be compensated for.

We note here that, based on assumptions **A1** and **A2**, it may seem that the set of correct matches can be obtained directly by finding the longest increasing (nonconsecutive) subsequence, $L(N)$. This approach yields poor results for two reasons. First, due to local order inversions between adjacent correct matches, caused by small changes in depth and camera roll, $|L(N)|$ is usually not a good estimation for N_G. Second, it is proven in [28] that for a random permutation of size N the expectation of $|L(N)|$ is $2\sqrt{N}$; thus, when N_G is low, $L(N)$ may correspond to only incorrect matches or to a mixture of correct and incorrect matches. In contrast to this naïve approximation, our analysis also considers the distribution of the order of the incorrect matches and therefore better estimates N_G even for low values.

4. The Number of Correct Matches

Two sequences are considered *fully overlapped* if the fields of view (FOV) of the corresponding images consist of the same region of the scene. We first consider the case of fully overlapped sequences (Sec. 4.2) and then present a method to compute the regions of overlap between the pair of images (Sec. 4.3). The Kendall distance metric that is used to estimate N_G is reviewed next.

4.1. The Kendall Distance

The Kendall distance is the number of pairwise order inversions in the two sequences $[N]$ and σ. Two pairs of matched features, $(p_i, q_{\sigma(i)})$ and $(p_j, q_{\sigma(j)})$, have *order inversion* if the orders (i, j) and $(\sigma(i), \sigma(j))$ are inverted. Formally, let us define a binary function, η_σ:

$$\eta_\sigma(i,j) = \begin{cases} 1 & (i < j) \wedge (\sigma(i) > \sigma(j)) \\ & \qquad\qquad \textit{or} \\ & (i > j) \wedge (\sigma(i) < \sigma(j)) \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The Kendall distance is thus given by:

$$K([N], \sigma) = \sum_{1 \le i \le N} \sum_{i < j \le N} \eta_\sigma(i,j). \tag{2}$$

To compute N_G from the value $K = K([N], \sigma)$ we formulate K as the sum of three terms:

$$K = K_G + K_B + K_{GB}, \tag{3}$$

where K_G corresponds to the number of order inversions between correct matches, K_B between incorrect matches, and K_{GB} between pairs of correct and incorrect matches. That is,

$$\begin{aligned} K_G &= \sum_{i \in G} \sum_{\substack{j \in G \\ i<j}} \eta_\sigma(i,j), \\ K_B &= \sum_{i \in B} \sum_{\substack{j \in B \\ i<j}} \eta_\sigma(i,j), \\ K_{GB} &= \sum_{i \in G} \sum_{\substack{j \in B \\ i<j}} \eta_\sigma(i,j). \end{aligned} \tag{4}$$

For $N_G = N$ (i.e., $|B| = 0$) it follows directly from **A1** that $K = K_G = 0$. On the other hand it follows directly from **A2** that if $N_G < N$ (i.e., $|B| > 0$), then $K > K_B > 0$. Hence, a simple case to consider is when $K = 0$ which implies $N_G = N$.

To obtain an explicit equation in N_G for $K > 0$, the terms in Eq. 3 are normalized by the maximal possible number of pairwise order inversions; that is, the number of pairs in each term. The number of pairs are given by $N(N-1)/2$ in a sequence of length N, and by $N_G N_B$ between two disjoint sequences of lengths N_G and N_B. That is,

$$\begin{aligned} \hat{K}_B &= \tfrac{2K_B}{N_B(N_B-1)}, & \hat{K}_G &= \tfrac{2K_G}{N_G(N_G-1)}, \\ \hat{K} &= \tfrac{2K}{N(N-1)}, & \hat{K}_{GB} &= \tfrac{K_{GB}}{N_G N_B}. \end{aligned} \tag{5}$$

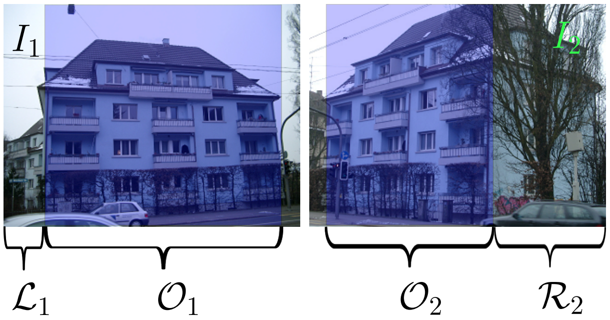


Figure 2. A partial overlap between a pair of images from the ZuBuD dataset.

Using some algebraic manipulation after substituting the terms in Eq. 3 by Eq. 5 and replacing $N_B = N - N_G$, we obtain the following quadratic equation in N_G:

$$\begin{aligned} 0 = \quad & N_G^2[\hat{K}_G + \hat{K}_B - 2\hat{K}_{GB}]+ \\ & N_G[2N\hat{K}_{GB} - (2N-1)\hat{K}_B - \hat{K}_G]+ \\ & N(N-1)(\hat{K}_B - \hat{K}). \end{aligned} \tag{6}$$

Note that N is given and $\hat{K}$ can be directly computed from the set of matches, $\mathcal{M}$, using *Eq.* 2. Using assumption **A1** it directly follows that $\hat{K}_G = 0$, from assumption **A2** it follows that $E(\hat{K}_B) = 1/2$, where $E(x)$ is the expected value of x (see our proof in supplementary material; and alternative proof [11, p. 257]). We next show how to estimate $E(\hat{K}_{GB})$ for fully overlapped and partially overlapped sequences.

4.2. Full Overlap

For fully overlapping sequences a third statistical assumption is used:

A3: The distribution of the features that correspond to correct matches within the features that correspond to incorrect matches is uniform in $[N]$ and σ; that is, the expected number of features in G and B in every interval of size s is sN_G/N and sN_B/N, respectively.

This assumption obviously does not hold for the non-overlapping margins which contain only incorrect matches (see Sec. 4.3 which considers this case). In the full-overlap case, the following claim holds:

Claim 1. *Under assumptions **A1-A3**,* $E(\hat{K}_{GB}) = 1/3$.

To prove this claim, let us first define $H_\sigma(i)$ to be the number of inversions in which a match, $(i, \sigma(i))$, participates. The desired value, $E(\hat{K}_{GB})$, is obtained by averaging $E(H_\sigma(i))$ over all $i \in G$ and normalizing it by $N_G N_B$. By definition, $H_\sigma(i)$ is the number of features from the left of i that match features to the right of $\sigma(i)$ and those from the right of i that match a features to the left of $\sigma(i)$. We can use assumption **A2** to compute the expected value of $H_\sigma(i)$ for $i \in G$ given the number of indexes with incorrect matches to the left and right of i (β_1^l and β_1^r) and to the left and right of $\sigma(i)$ (β_2^l and β_2^r). The probability of inversion between $i \in G$ and $j < i$ is given by β_2^r/N_B. A similar argument is used for $j \in B$ and $j > i$. Hence we obtain:

$$E(H_\sigma(i)|i \in G, \beta_1^l, \beta_1^r, \beta_2^l, \beta_2^r) = \beta_1^l \frac{\beta_2^r}{N_B} + \beta_2^l \frac{\beta_1^r}{N_B}. \tag{7}$$

Using assumption **A3** we can approximate $\sigma(i) \approx i$, and $E(\beta_1^l) = E(\beta_2^l) = iN_B/N$ and $E(\beta_1^r) = E(\beta_2^r) = (N - i)N_B/N$; thus

$$E(H_\sigma(i)) = 2\frac{i(N-i)N_B}{N^2} = 2N_B x(1-x),$$

where $x = i/N$. The mean of $x(1-x)$ for $0 \leq x \leq 1$ is $1/6$, and thus

$$E(\hat{K}_{GB}) = \frac{1}{N_G N_B} \sum_{i \in G} E(H_\sigma(i)) = \frac{2N_G N_B}{N_G N_B}\frac{1}{6} = \frac{1}{3}.$$

Substituting $E(\hat{K}_G)$, $E(\hat{K}_B)$ and $E(\hat{K}_{GB})$ with their obtained values in Eq. 6, we obtain the following quadratic equation in N_G:

$$0 = \tfrac{1}{6}N_G^2 - (\tfrac{1}{2} - \tfrac{1}{3}N)N_G - N(N-1)(\tfrac{1}{2} - \hat{K}). \tag{8}$$

There are two solutions to this equation; when $0 \leq \hat{K} \leq 1/2$, we take the only solution in the range $[0, N]$. When $\hat{K} > 1/2$, we set $N_G = 0$.

4.3. Partial Overlap

When the sequences are not fully overlapping (e.g., Fig. 2) the value of $E(K_{GB})$ depends on the non-overlapping margins of the sequences, defined below. In this case, the estimation of N_G given by Eq. 8 is an underestimation (see Claim 2). We next describe a method to estimate N_G by computing the regions of overlap, using this observation.

Formally, the sequence $[N]$ is partitioned into three intervals: $\mathcal{L}_1$, $\mathcal{O}_1$, and $\mathcal{R}_1$, defined by the lowest, i_L^1, and highest, i_H^1, indexes of the correct matches. Similarly, we define $\mathcal{L}_2$, $\mathcal{O}_2$, and $\mathcal{R}_2$ with the indexes i_L^2 and i_H^2. Note that an index of an incorrect match, $i \in \mathcal{O}_1 \cap B$, is not necessarily matched to an index in $\mathcal{O}_2$. To use our results for the fully overlapped sequences (Sec. 4.2), we discard such indexes and define the *fully overlapped subsequences*, $\hat{\mathcal{O}} = (\hat{\mathcal{O}}_1, \hat{\mathcal{O}}_2)$ as follows:

$$\hat{\mathcal{O}}_1 = \{i \mid (i \in \mathcal{O}_1) \land (\sigma(i) \in \mathcal{O}_2)\}, \tag{9}$$

and $\hat{\mathcal{O}}_2$ is the sequence of indexes of the matched features to $\hat{\mathcal{O}}_1$.

Note that $\hat{\mathcal{O}}_1$ and $\hat{\mathcal{O}}_2$ are defined by the 4-tuple of indexes, $\omega^* = (i_L^1, i_H^1, i_L^2, i_H^2)$. Moreover, N, N_G and $\hat{K}$, on the subsequences defined by ω^* are given by $N(\omega^*)$, $N_G(\omega^*)$ and $\hat{K}(\omega^*)$. We can now use these new values in Eq. 8 to obtain:

$$0 = \begin{array}{l} \frac{1}{6}N_G^2(\omega^*) - (\frac{1}{2} - \frac{1}{3}N(\omega^*))N_G(\omega^*) \\ -N(\omega^*)(N(\omega^*)-1)(\frac{1}{2} - \hat{K}(\omega^*)). \end{array} \tag{10}$$

In this case we claim the following:

Claim 2. *The expected maximal value of $N_G(\omega)$, for any ω, is obtained for $\omega = \omega^*$, i.e.,*

$$\max_{\omega} N_G(\omega) = N_G(\omega^*). \tag{11}$$

We use assumptions **A1**-**A3** to prove it (see the supplementary material). We use this claim to estimate ω^*, which determines the region of overlap between the pair of images, as described below. Given ω^* we can estimate the desired N_G using Eq. 10.

Estimating ω^*. To compute $N_G(\omega^*)$, we need to consider all 4-tuples $\omega \in [1, \ldots, N]^4$. For efficiency, we consider only a small subset of intervals. Formally, we consider all intervals in the sequence of the form $[i, j] = [t_i q + 1, t_j q]$ where $0 < t_i < t_j \leq N/q$ are integers. The algorithm that considers all these intervals in I_1 and I_2, we name *K2D*. To further improve efficiency, we propose the *K1D* algorithm which considers sequentially the intervals in each image. That is, we first use the intervals, $\omega = (i_L^1, j_H^1, 1, N)$, to sample the values of $N_G(\omega)$. The resulting interval, $[i_L^1, j_H^1]$, is then used to fix the interval in I_1 and estimate the interval in I_2; that is, we use the intervals $\omega = (i_L^1, j_H^1, i_L^2, j_H^2)$ to arrive at our final estimate for ω^* and $N_G(\omega^*)$.

Kendall distance computation. The Kendall distance can be computed in $\mathcal{O}(N \log N)$ time using the merge sort algorithm [19], applied on σ. The basic idea is that the number of inversions can be computed at the merge stage (when merging two sorted arrays into one). The number of inversions that should be added to the count is the number of elements that remain in the left array when the next minimal element is taken from the right array.

For partial overlap, maximizing the estimated value of $N_G(\omega)$ in Eq. 11 requires multiple Kendall distance computations for various intervals. We compute these distances efficiently by first computing the Kendall distance for each of the non-overlapping intervals of size q using the merge sort algorithm. The Kendall distance of an interval of size dq is obtained by counting the inversions when merging two successive intervals of size $d_1 q$ and $d_2 q$, where $d_1 + d_2 = d$.

5. Matching Probabilities

We propose to compute a probability, $P_K(i)$, that a match, $m = (p_i, q_{\sigma(i)})$, is correct, given the computed $H_\sigma(i)$ (defined in Sec. 4.2) and our estimation of $\theta = (\hat{\mathcal{O}}_1, \hat{\mathcal{O}}_2, N_G)$. This probability can then be used as a likelihood function for sampling matches, in particular in guided RANSAC methods.

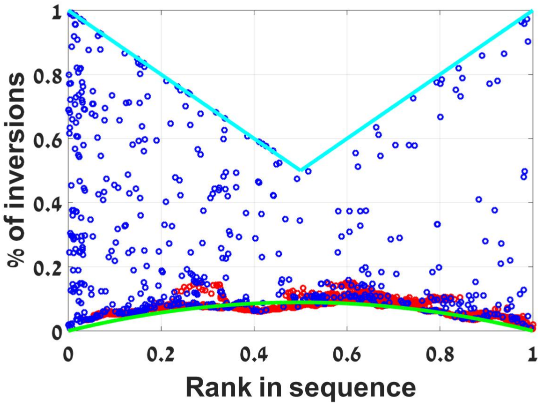


Figure 3. A graph of order inversions, H_σ, for an image from the Middlebury2014 dataset. The x-axis corresponds to the rank of the feature in I_1, and the y-axis corresponds to the percentage of order inversions out of the maximal value, $N-1$. Computed by USAC, the blue and red dots correspond to the inliers and outliers, respectively. The green and cyan lines correspond to the range of inversions for the incorrect matches. The expected number of inversions for the correct matches is given by the green line.

When $i \notin \hat{\mathcal{O}}_1$ or $\sigma(i) \notin \hat{\mathcal{O}}_2$ we set $P_K(i) = 0$ (or to a small value), since, by the definition of $\hat{\mathcal{O}}_1$ and $\hat{\mathcal{O}}_2$, $i \in B$. Hence we would like to compute $P_K(i)$ for the remaining indices. For a full overlap, a typical graph of the number of order inversions, $H_\sigma(i)$, as a function of the index i is presented in Fig. 3. For $i \in B$ (blue dots), $H_\sigma(i)$ is approximately uniformly distributed in a range which depends on i. As expected from the analysis presented in Sec. 4.2, the distribution of $i \in G$ (red dots) is around the function $x(1-x)$, where $x = i/N$ and $N = |\hat{\mathcal{O}}_1|$.

To compute $P_K(i)$ for $i \in G$, we use the Bayes' theorem and the law of total probability:

$$\begin{aligned} P_K(i) = P\big(i \in G \big| H_\sigma(i)\big) &= \frac{P\big(H_\sigma(i)\big|i\in G\big)P\big(i\in G\big)}{P\big(H_\sigma(i)\big)} \\ &= \frac{P_{H|G}P_G}{P_{H|G}P_G + P_{H|B}P_B}, \end{aligned} \tag{12}$$

where $P_{H|G} = P\big(H_\sigma(i)\big|i \in G\big)$, $P_{H|B} = P\big(H_\sigma(i)\big|i \notin G\big)$, $P_G = P\big(i \in G\big)$ and $P_B = P\big(i \notin G\big)$. We assume w.l.g. that there is a full overlap; hence, the probabilities $P_G = N_G/N$ and $P_B = 1 - P_G$ are given by the ratio of the correct and the incorrect matches to N, respectively.

We next describe our estimation of $P_{H|G}$. Consider $\beta = (\beta_1^l, \beta_1^r, \beta_2^l, \beta_2^r)$ (defined in Sec. 4.2). Note that β_1^r and β_2^r are given directly from β_1^l, β_2^l, N, and the estimated N_G. Since β is unknown, we compute $P_{H|G}$ using the law of

Dataset	Value	KGT	K	K2D	K1D
Syn1	$\mu(\mathbf{N_G})$	0.6	14.5	3.2	4
	$\mu(\mathcal{O})$	n.a.	n.a.	0.91	0.89
	Runtime	2	2.2	83.2	11.2
Syn2	$\mu(\mathbf{N_G})$	0.5	10.4	3.2	3.7
	$\mu(\mathcal{O})$	n.a.	n.a.	0.9	0.8
	Runtime	2.9	2	79.2	11.1

Table 1. The mean normalized absolute error (percentage) in the estimation of N_G in the synthetic datasets.

total probability over the set of possible values, $\beta \in S_\beta$:

$$P(H_\sigma(i)|i \in G) = \sum_{\beta \in S_\beta} P(H_\sigma(i)|i \in G, \beta)P(\beta).$$

For efficiency, only k values of $\beta \in S_\beta$ are considered by restricting the values of β_1^l and β_2^l to be around their expectations, $(i-1)P_B$ and $(\sigma(i)-1)P_B$, respectively (in our implementation we set $k = 5$).

The probability $P(\beta)$ is modeled as the product of two hypergeometric distributions, one for β_1^l and the other for β_2^l. Given a value for β, we model $P(H_\sigma(i)|i \in G, \beta)$ as a product of two hypergeometric distributions. In a similar way we estimate $P(H_\sigma(i)|i \in B, \beta)$. Substituting these terms into Eq. 12 yields $P_K(i)$. We approximate the hypergeometric PDFs by Gaussian PDFs with the same mean and standard deviation, and approximate $P_{H|B}$ as a uniform distribution whose boundaries we compute (see the supplementary material for the complete derivation).

6. Experiments

Our method was tested on both synthetic and real data. For each dataset we tested the following algorithms. The first, which we refer to as K, is the basic one where N_G is estimated using the entire sequence and ignoring the margins; that is, without first computing the overlap (see Sec. 4.2). The other two algorithms compute N_G, $\mathcal{O}_1$ and $\mathcal{O}_2$. One uses the joint sampling of intervals from I_1 and I_2, while the other uses a separate sampling of intervals in I_1 and I_2 (see Sec. 4.3). These algorithms are referred to as K2D and K1D, respectively. For evaluation purpose we also consider KGT, where the ground truth of $\mathcal{O}_1$ and $\mathcal{O}_2$ are given, and compute only N_G.

To evaluate the accuracy of N_G we measure the normalized absolute error, $\mu(N_G) = |N_G - N_G^{GT}|/N$, where N_G^{GT} is the ground truth. Note that $\mu(N_G)$ is presented as a percentage in all tables. To evaluate the accuracy of $\mathcal{O}_1$ and $\mathcal{O}_2$, we measure the average of their normalized overlap with the ground truth; that is, the average between the two $\mu(\mathcal{O}) = 100|\mathcal{O} \cap \mathcal{O}_{GT}|/|\mathcal{O} \cup \mathcal{O}_{GT}|$, for $\mathcal{O} \in \{\mathcal{O}_1, \mathcal{O}_2\}$, where $\mathcal{O}_{GT}$ is the ground truth.

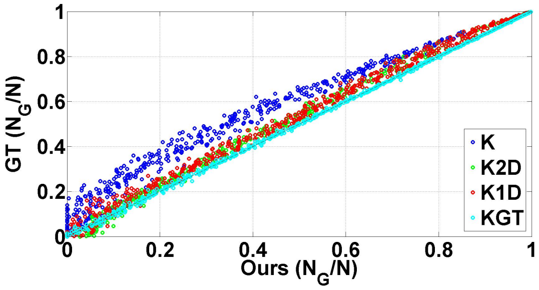


Figure 4. Scatter plot for synthetic experiment Syn2.

6.1. Synthetic Data

A matching between features in a pair of images is depicted by a permutation. We generated 500 different permutations of $N = 1000$ indexes, using our assumptions **A1-A3**, for various values of $\mathcal{O}_1$, $\mathcal{O}_2$ and N_G.

Test Syn1: We set $N_G = 300$, the size ($> N_G$) and location of $\mathcal{O}_1$ and $\mathcal{O}_2$ are randomly chosen. Tab. 1 presents $\mu(N_G)$, $\mu(\mathcal{O})$ and the mean runtime (displayed in milliseconds). The error $\mu(N_G)$ is the largest for K, while K2D and K1D are similar and closer to KGT.

Test Syn2: We set N_G between 0 to 1000, and randomly choose $\mathcal{O}_1$ and $\mathcal{O}_2$ as in "Test Syn1" (see results in Tab. 1). The scatter plot of the four algorithms with respect to the ground truth is presented in Fig. 4. The K algorithm, where $|\mathcal{O}_1| = |\mathcal{O}_2| = N$ is wrongly assumed, underestimates N_G, and $\mu(N_G)$ is relatively large. The estimation using K2D and K1D are similar and slightly underestimate of N_G. These results are probably due to the sampling used for estimating $\mathcal{O}_1$ and $\mathcal{O}_2$, since for KGT, the errors are negligible. For both tests, K1D is almost an order of magnitude faster than K2D.

6.2. Real Data

We use the Middlebury 2005&2006 and 2014 datasets [26, 16, 25], for real data with ground truth (denoted by MFull05&06 and MFull14, respectively). We also evaluate our method on the USAC dataset [22] and BLOGS dataset [3] (combined), and the ZuBuD dataset [27], for which no ground truth is available (denoted by U&B and ZuBuD, respectively). We run the BEEM [13] and USAC [22] algorithms (both with their default settings), where the number of inliers returned is compared to N_G. The lowest and highest inlier indexes returned by these methods can be regarded as their estimations of $\mathcal{O}_1$ and $\mathcal{O}_2$.

Middlebury Full Overlap. The mean errors and runtimes (in milliseconds) are presented in Tab. 2. The error, $\mu(N_G)$, is similar for K, K2D and K1D, while K2D is much slower. The K1D and BEEM are similar in $\mu(N_G)$, while USAC is more accurate. The runtime of K1D is between one and two orders of magnitude faster than USAC and BEEM's run-

Dataset	#im pairs	N_G^{GT}	BEEM		USAC		K		K2D		K1D	
			$\mu(N_G)$	Time	$\mu(N_G)$	Time	$\mu(N_G)$	Time	$\mu(N_G)$	Time	$\mu(N_G)$	Time
MFull14	23	1584	9.9	2065	3	1782	10	30	9.9	291	9.9	62
MFull05&06	27	2776	13	4198	3	1670	3.3	51	3.3	427	3.3	76
MPar14	23	712	5	1825	3.6	1152	11.5	13	6.8	195	7.1	31
MPar05&06	27	1032	5.3	2344	4.1	1276	23.1	11	6.3	249	6.7	39

Table 2. The mean normalized absolute error (percentage), $\mu(N_G)$, in comparison to the ground truth (its mean, N_G^{GT}) and the mean runtime (in milliseconds) for each of the methods in the Middlebury datasets.

Dataset	#im pairs	$\lvert G_{BEEM}\rvert$	BEEM		USAC		K		K2D		K1D	
			$\mu(N_G)$	Time	$\mu(N_G)$	Time	$\mu(N_G)$	Time	$\mu(N_G)$	Time	$\mu(N_G)$	Time
U&B	21	138	n.a.	1444	8.6	1440	11.7	11	8.8	142	9	18
ZuBuD	201	69	n.a.	692	7.3	1257	9.3	2	6.1	90	6.5	11

Table 3. The mean normalized absolute error (percentage), $\mu(N_G)$, in comparison to BEEM (its mean, $|G_{BEEM}|$) and the mean runtime (in milliseconds) for each of the methods in the USAC&BLOGS and ZuBuD datasets.

time. Fig. 5 presents the complete comparison details.

Middlebury Partial Overlap. The overlap between image pairs in the Middlebury datasets is very large (about 90% of the image). Therefore, we randomly vertically cut them to obtain smaller overlaps. This enables us to test the partial-overlap method (Sec. 4.3) against the ground truth (Tab. 2). These datasets are denoted by MPartial05&06 and MPartial14 for Middlebury 2005&2006 and 2014, respectively. Both the errors of K2D and K1D are low and similar to the full overlap case, demonstrating the success of the partial overlap method. However, the running time is much lower for K1D. The error, $\mu(N_G)$, is large for method K since it ignores the margins. As in the full overlap case, BEEM is similar in accuracy to K2D and K1D, while USAC is more accurate. The similarity of the $\mathcal{O}$ evaluation, $\mu(\mathcal{O})$, is 0.89 and 0.89 for both the K2D and K1D methods. Fig. 5 presents the complete comparison details.

USAC & BLOGS and ZuBuD. Since no ground truth is available for these datasets, the estimation results are compared to BEEM (see Tab. 3). The errors for K2D and K1D and USAC are similar (K is larger as expected). K2D and K1D are one and two orders of magnitude faster than BEEM and USAC. Fig. 5 presents the complete comparison details.

6.3. Applications

We next evaluate three applications of N_G estimation.

Halting condition. The classic halting condition [14, Chapter 11.6] for adaptive RANSAC methods is based on the probability that at iteration t at least one consensus set was constructed from an uncontaminated minimal set of matches. This probability is computed as a function of the number of correct matches. As this number is usually unknown, the probability is updated when a larger consensus set is found. The method halts when the confidence in the solution is high based on the size of the consensus set. Since our algorithm is able to estimate N_G, we are able to halt RANSAC instantly when a consensus set of at least N_G matches has been found

To test this we run USAC with and without the halting condition on all the datasets and compare the inlier rate and runtime (see Fig. 7). The total runtime with the halting condition is 69% of the total runtime without the halting condition. This is achieved with nearly no loss of inliers (normalized absolute error of 0.78%).

SfM image pairs. A major time-consuming stage in the pipeline of SfM methods [30] is running RANSAC on all image pairs to filter out incorrect matches. We propose to run RANSAC only on image pairs with sufficiently large N_G, avoiding spatially unrelated image pairs.

We use the dataset "Barcelona" [8], which consists of 191 images taken from all around a building. We consider the total number of inliers and the total runtime from running USAC on all image pairs as 100% of inliers and runtime. Our goal is to maintain the inliers, while reducing the runtime. We use a threshold, t, to discard image pairs for which our estimation of $N_G < t$, using the K2D and K1D algorithms. Fig. 6(a)&(b) shows the reduction in inliers and runtime as a function of t.

For K2D, the total runtime decreases to 17% (or 4.5%) while maintaining 91% (or 86%) of the inliers. In both of these examples, all images remain with at least one other image for which the fundamental matrix could be calculated. For K1D, the total runtime decreases to 7% (or 1.2%!) while maintaining 87% (or 80%) of the inliers. In the second example only one image out of the 191 images was lost. Note that the percentage of time spent discarding matches outliers in a typical SfM method (e.g., [30]) is ~ 25 % of the total runtime.

Matching probability. Computing matching probabilities, P_K, can be useful for several applications, including

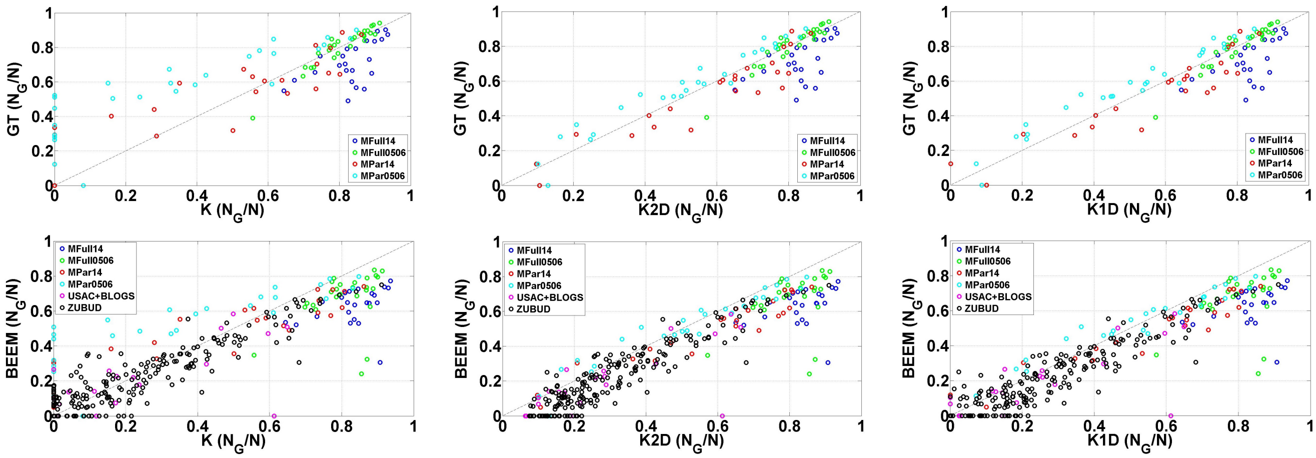
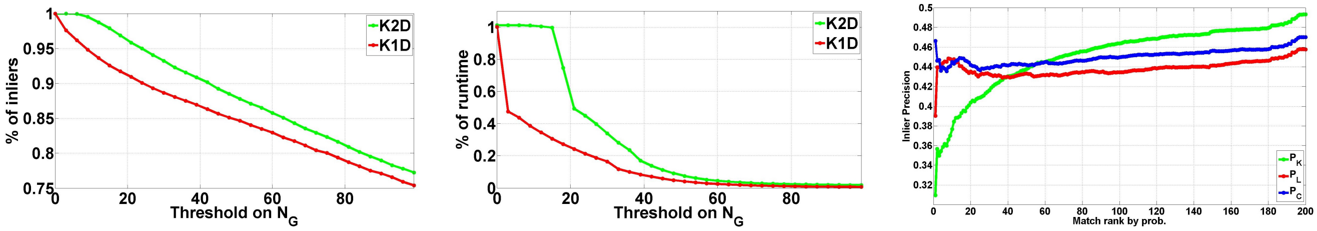


Figure 5. Scatter plots for the real datasets. From left to right: K, K2D and K1D methods. Top against the ground truth (only in Middlebury datasets), and bottom is a comparison against BEEM. The axes correspond to N_G/N.



(a) (b) (c)

Figure 6. (a)-(b) Using our estimate for N_G to discard pairs of images from a SfM pipeline in the "Barcelona" dataset; (a) percentage of the remaining correct matches as a function of the threshold; (b) percentage of the remaining runtime as a function of the threshold; (c) Comparison of sampling methods for Guided RANSAC using the pairs of images from all datasets (except for "Barcelona"). The x axis corresponds to the rank of the matches when sorted according to the probability. The y axis corresponds to the precision of correct matches; that is, a point on the graph, (x, y), denotes the ratio of the number of correct matches from the x matches to all x matches.

guided RANSAC. For evaluating our estimate for P_K, we present the mean inlier precision of the $x \leq 200$ highest ranked matches, given by P_K (Fig. 6(c)). It is compared with the Lowe ranking ([21]), denoted by P_L. The mean precision was calculated on all image pairs from all datasets. The curve for the mean P_K (green) is below the curve for the mean P_L (red) for the first 50 matches, and then vice versa. A combination of the two probabilities, given by $P_C = \frac{P_K P_L}{P_K P_L + (1-P_K)(1-P_L)}$. is the blue curve. Using low values of x, P_C outperforms its components. For higher values of x, our estimation is better.

7. Conclusions and Future Work

In this paper we introduced a novel method to estimate the number of correct matches, and a method to estimate the region of overlap between a pair of images. We also derive a probability function for a match to be correct. All these were done using only the spatial order of a given set of matches, and some reasonable statistical assumptions. We demonstrated the effectiveness of this estimation on real datasets for an analysis of feature matches using only the spatial order of features in each image. Our method successfully competes with methods that compute the set of inliers explicitly by recovering the epipolar geometry, but at a much lower cost. It is impressive to see what can be learned about matching using only spatial orders.

Acknowledgments: This work was partially supported by the Israel Science Foundation, grant no. 930/12, and by the Israeli Ministry of Science, grant no. 3-8744.

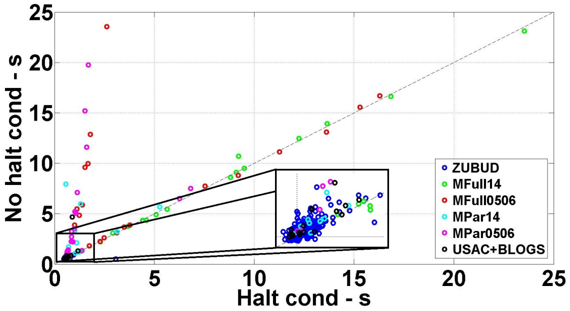


Figure 7. USAC for inlier rate estimation *without* the halting condition (y-axis) vs *with* the halting condition (x-axis).

References

[1] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: speeded up robust features. In *ECCV*. 2006.

[2] A. F. Bobick and S. S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):181–200, 1999.

[3] A. S. Brahmachari and S. Sarkar. Hop-diffusion monte carlo for epipolar geometry estimation between very wide-baseline images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 35(3):755–762, 2013.

[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *ECCV*. 2010.

[5] O. Chum and J. Matas. Matching with PROSAC-progressive sample consensus. In *CVPR*, 2005.

[6] O. Chum and J. Matas. Optimal randomized RANSAC. *IEEE Trans. Patt. Anal. Mach. Intell.*, 30(8):1472–1482, 2008.

[7] O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In *Pattern recognition*, pages 236–243. Springer, 2003.

[8] A. Cohen, C. Zach, S. N. Sinha, and M. Pollefeys. Discovering and exploiting 3d symmetries in structure from motion. In *CVPR*, 2012.

[9] M. Deza, L. N. Supérieure, and T. Huang. Metrics on permutations, a survey. In *Journal of Combinatorics, Information and System Sciences*. Citeseer, 1998.

[10] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW*, 2001.

[11] W. Feller. *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 2008.

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

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

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

[15] W. Hartmann, M. Havlena, and K. Schindler. Predicting matchability. In *CVPR*, 2014.

[16] H. Hirschmüller and D. Scharstein. Evaluation of cost functions for stereo matching. In *CVPR*, 2007.

[17] E. Johns, D. Edward, and G. Z. Yang. Pairwise probabilistic voting: Fast place recognition without RANSAC. In *ECCV*. 2014.

[18] M. G. Kendall. A new measure of rank correlation. *Biometrika*, pages 81–93, 1938.

[19] J. Kleinberg and É. Tardos. *Algorithm design*. Pearson Education India, 2006.

[20] R. Litman, S. Korman, A. Bronstein, and S. Avidan. Inverting RANSAC: Global model detection via inlier rate estimation. In *CVPR*, 2015.

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

[22] R. Raguram, O. J. Chum, M. Pollefeys, J. Matas, and J. M. Frahm. USAC: a universal framework for random sample consensus. *IEEE Trans. Patt. Anal. Mach. Intell.*, 35(8):2022–2038, 2013.

[23] S. Ramalingam, M. Antunes, D. Snow, L. G. Hee, and S. Pillai. Line-sweep: Cross-ratio for wide-baseline matching and 3D reconstruction. In *CVPR*, 2015.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *ICCV*, 2011.

[25] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition*, pages 31–42. Springer, 2014.

[26] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *CVPR*, 2007.

[27] H. Shao, T. Svoboda, and L. V. Gool. Zubud-zurich buildings database for image based recognition. *Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland, Tech. Rep*, 260, 2003.

[28] J. M. Steele. Variations on the monotone subsequence theme of erdös and szekeres. In *Discrete probability and algorithms*, pages 111–131. Springer, 1995.

[29] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[30] C. Wu. Towards linear-time incremental structure from motion. http://ccwu.me/vsfm/, 2013.

[31] A. L. Yuille and T. Poggio. A generalized ordering constraint for stereo correspondence. Technical report, DTIC Document, 1984.