Estimating the Number of Correct Matches Using Only Spatial Order

Lior Talker, *Student Member, IEEE,* Yael Moses, *Member, IEEE,* and Ilan Shimshoni, *Member, IEEE*

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. 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, and analyzed using the Kendall and Spearman Footrule distance metrics between permutations. This result is interesting in its own right. Moreover, we demonstrate three useful applications of our method: (i) a new halting condition for RANSAC based epipolar geometry estimation methods, (ii) discarding spatially unrelated image pairs in the Structure-from-Motion pipeline, and (iii) computing the probability that a given match is correct based on 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 90% while preserving about 85% of the image pairs with spatial overlap.

Index Terms—Feature Matching, RANSAC, Spatial Order, Correct Matches.

1 Introduction

Matching feature points between a pair of images is a fundamental problem in computer vision. The estimation of epipolar geometry between images [1], [2], 3D structure reconstruction (SfM) [3], [4], and scene recognition [5] 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) [1], [2], [6].

We propose a method[1] for analyzing the set of correct matches, without explicitly computing it, using the spatial order of the features in each image. Our method estimates the number of correct matches, the overlap region of the pair of images and whether they overlap at all, and the probability that a given match is correct. Our estimations can be used as a preprocessing step to improve the efficiency of existing methods such as RANSAC and SfM, as described below. The method can be applied to sets of matching features irrespective of their descriptors (e.g., [7], [8], [9]) or the matching method used to compute them.

The basic idea is as follows. We represent the image features as sequences defined by their spatial order along the x-axis (or the y-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 Figure 1). The matching is analyzed using measures of correlation between permutations, the Kendall and Spearman Footrule distance metrics [10], [11]. We use statistical assumptions on the distribution of correctly and incorrectly matched features; the spatial order of correctly matched features is usually preserved, whereas incorrectly matched features are expected to have random order. These assumptions are justified empirically in Section 6.3.1. Note that the problem is not trivialized by these assumptions; simply computing the largest set of features that preserve their spatial order does not provide the correct set of matches since some incorrect matches also preserve order (see discussion in Section 3). To obtain our estimations, it is therefore necessary to consider these two assumptions in addition to analyzing the interaction between correct and incorrect matches.

We next describe three applications of the estimated number of inliers computed by our method.

Halting condition for RANSAC:

In adaptive RANSAC [1], [2], [6], 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 Section 6).

Improving the efficiency of the SfM pipeline:

Computing the structure of the scene and the cameras' parameters from a set of images using SfM methods (e.g., [3], [4], [12], [13]) is a fundamental problem in computer

Manuscript received X; revised Y.

[1] A MATLAB code for our method is publicly available at http://liortalker.wixsite.com/liortalker/code.

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 pairwise image matching and RANSAC. Hence, to improve SfM efficiency, methods for detecting candidate pairs on which to apply a robust matching method were proposed (e.g., [13], [14], [15], [16], [17]). The number of correct matches computed by our method can be used to significantly shorten the SfM pipeline by running RANSAC only on image pairs with a sufficiently large number of correct matches. In Section 6 we show that in this task our method outperforms the Bag of Visual Words (BoW) method [14] and the Hough Pyramid Matching (HPM) [18].

Guided RANSAC:

The probability that a given match is correct can be used as a sampling prior in guided RANSAC methods, where matches that are more likely to be correct are sampled more often (e.g., [1], [2], [6], [19], [20], [21]). In existing methods the probability that a given match is correct is based only on feature descriptors. We show that the number of correct matches can be used to compute the probability that a given match is correct based only on the features' rank in the set of ordered features in each of the images. Thus, our method is an alternative to appearance based methods for estimating the probability for match correctness (see details in Section 5).

The rest of the paper is organized as follows. In the next section we review related work. In Section 3, we formalize the problem of estimating the number of correct matches. In Section 4, we analyze the problem and present a method for estimating the number of correct matches. In Section 5, we present a method for computing the probability that a match is correct. In Section 6 we present quantitative results for our method, compare it to other methods, and demonstrate its usefulness in three applications. Finally, in Section 7, we conclude the paper and propose future research directions.

2 Related Work

Feature point matching between a pair of images has been studied extensively in computer vision. Improving the accuracy of feature matching is still a widely studied research area. 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., [7], [8], [9]). Recently, fully learned feature detectors and descriptors have been proposed using Convolutional Neural Networks (CNNs) [22], [23], [24]. However, their improved accuracy mostly comes with a significant increase in descriptor size and hence with increasing matching runtime. Appearance based methods, such as [25], discard features by learning the success rate in matching their descriptors, 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., [26], [27]. In [28] a coarse 3D reconstruction is used to filter out incorrect matches.

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, in particular when there are only incorrect matches. This can be a major bottleneck in online applications involving large sets of images. To improve RANSAC's runtime (and accuracy), methods for guiding the sampling of matches according to their probability to be correct (instead of a uniform sampling) were proposed [1], [2], [6], [19], [20], [21], [29], [30]. In [29], a spatial coherency measure for the matches is considered as a likelihood. In [30], Extreme Value Theory is leveraged to estimate a confidence measure for each match to be correct. In [1], [2], the similarity of feature appearance defined by Lowe's ratio test between the closest and the second closest matches [7] is used as a match correctness likelihood. 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 Section 6.

The largest portion of the runtime in a SfM pipeline is spent on image pair matching; thus, filtering spatially unrelated image pairs is essential. The goal is often to obtain a Scene Graph (SG), where the nodes correspond to images, and an edge, (i, j), exists if images i and j have a spatial overlap (i.e., their fundamental matrix can be computed). This is usually done by assigning similarity measures between pairs of images and using this measure to filter out spatially unrelated images, either by a simple threshold or by more complex methods that consider the structure of the graph, e.g., [13], [15], [16], [17]. One of the most widely used methods for computing similarity between pairs of images is the BoW method [14]. The keypoints from all images are first indexed in a visual vocabulary. Each image is then represented as a (sparse) histogram of vocabulary word occurrences. The similarity between a pair of images is given by a weighted ℓ_2 distance between the histograms of the image pair, where the weights reflect the frequency of word occurrences. One major drawback of BoW is the lack of spatial information.

Another popular approach for filtering spatially unrelated image pairs in a SfM pipeline is using techniques inspired by image retrieval [18], [31], [32], [33], [34], [35], [36], which ranks images from a dataset based on their similarity to a query image. In [18] a spatially aware voting method inspired by Hough transform is applied to efficiently determine which sets of feature matches are related by a geometric transformation with high probability. In [33], [34], a compact descriptor of fixed length called VLAD, which encodes the image descriptors and is directly used for image pair matching, is introduced. In [32] a new matching score between images is defined from a combination of the matching score from the VLAD approach and the Hamming embedding approach, which binarizes the visual words in the BoW approach. In [35], [36], the local shape (i.e., orientation and scale) of feature descriptors is used to approximately determine which image pairs have scene overlap. In addition, these methods are able to obtain an approximate geometric transformation for the image pair. A comprehensive survey on image retrieval techniques is given in [31]. Our method is complementary to these approaches and may also be used as a similarity measure in order to filter out spatially unrelated image pairs, as

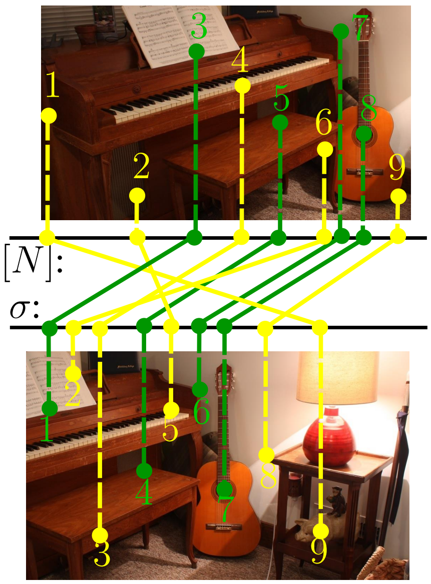


Figure 1: Correctly/incorrectly matched features are marked as green/yellow circles, respectively. The feature sequences are given by $[N] = \langle 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\rangle$ and $\sigma = \langle 9, 5, 1, 3, 4, 2, 6, 7, 8\rangle$. For example, $\sigma(1) = 9$ and $\sigma(5) = 4$. Note that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

demonstrated in Section 6.2.2.

A method for estimating the inlier rate was used in [37] to compute a homography transformation between two images, which is guaranteed to find an approximation of the global optimum. The rate is estimated 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 applicable only to homography transformations.

Using the spatial order of features has a long history in computer vision, mostly for stereo correspondence [38], [39]. The spatial order was used to constrain the location of matching features on corresponding epipolar lines. This constraint was regarded as a special case emerging from the continuity of surfaces and the assumption of opacity (objects are usually not translucent). The conditions for the violation of this constraint were first suggested in [40], and termed “the forbidden zone” in [41].

Recently, the spatial orders of features in a collection of images were used in [42] to guide a user to rotate his or her camera such that its FOV overlaps with that of another user’s camera. In this approach, the scene is represented by two feature sequences obtained by aggregating partial feature sequences, defined by the spatial order of features in each image. This representation is then used to derive the correct direction and magnitude of the rotation.

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 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 $N_B = |B|$. Given N, our goal is to estimate N_G. In addition, we estimate the overlapping region (see Figure 5) of the pair of images, and the probability of each pair to be a correct match. We do so by analyzing the relative spatial orders of features in the two images.

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 of the points. 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 Figure 1). That is, the matching is represented by the permutation σ; if i is the rank of a feature in the sequence of I_1, then $\sigma(i)$ is the rank of its matched feature in the sequence of I_2.

We analyze the spatial orders of matched features in the pair of images, which is represented by the permutation σ, using two distance metrics on permutations. The first is the Kendall distance [43], which corresponds to the sum of order inversions in σ, and the second is the Spearman Footrule (SF) distance [10], [11], which corresponds to the sum of rank shifts in σ. We use statistical assumptions on the distributions of the correct and incorrect matches to estimate the desired values.

4 Estimating N_G

We first present three statistical assumptions on the correct and incorrect matched pairs. Then we present the Kendall and Spearman distance metrics and how they are used. In Section 6 we compare the results obtained by using the two distance metrics.

4.1 Statistical Assumptions

The following assumptions on correct and incorrect matching are used, although they do not strictly hold in practice.

A1: *The spatial order of the correctly matched features in I_1 is preserved in I_2.*

Assumption **A1** is often used in stereo matching algorithms, e.g., [38], [44]. It may not hold, for example, when the cameras’ orientations differ by a relative roll (rotation around the z-axis), or when the scene points that correspond to a pair of features have significant depth differences (e.g., the pole in Figure 2). However, we show empirically that the spatial order is mostly preserved in image pairs of real outdoor scenes (Section 6.3.1). In Section 6.3.3 we propose an extension of our method to estimate the roll value when it is unavailable, as in [45], [46].

A2: *The spatial order of incorrectly matched features is random.*

Assumption **A2** holds when the spatial location of incorrectly matched features in different images is arbitrary.

We note here that it does not follow from **A1** and **A2** that the set of correct matches can be obtained directly by finding the longest increasing (nonconsecutive) subsequence, $L(N)$. First, $L(N)$ is expected to be longer than the set of correct matches since it contains a mixture of correct and incorrect matches. In particular, for any random permutation of size N, the expectation of $|L(N)|$ is $2\sqrt{N}$ [47]. Second,

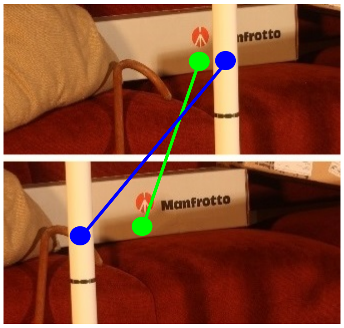
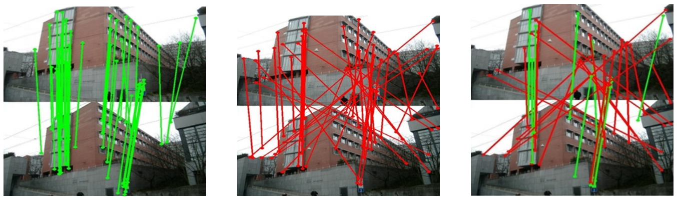


Figure 2: An example of an order inversion due to a "pole" that contradicts assumption **A1**. The blue feature in the upper image is to the left of the green feature, and vice versa in the bottom image.



(a) K_G (b) K_B (c) K_{GB}

Figure 3: Classification of the pairs of matches into (a) only correct matches, (b) only incorrect matches and (c) correct and incorrect matches. Note that, for clarity, some of the matched pairs are omitted from (c).

local order inversions between neighboring correct features result in correct matches being removed from $L(N)$ and hence decrease its length. In contrast, our analysis also takes into account the order of the incorrect matches and their interaction with the correct ones. Furthermore, the distribution of order inversions rather than exact inference of the correct matches is used. As a result, it better estimates N_G even for low values.

The third assumption refers to the distribution of the features that have correct matches (and similarly incorrect matches) in the entire sequence in each image.

A3: *The ranks of correctly matched features are distributed uniformly in* $[N]$ *and in* σ.

The interpretation of assumption **A3** is that the expected number of features in every interval of size s is sN_G/N in G and sN_B/N in B, respectively. Clearly, this assumption does not hold when there are non-overlapping regions in the fields of view (FOV) of the two cameras, that is, regions visible to one of the cameras and not the other. Features in a non-overlapped region cannot have a correct match; hence, all of them are in B. In Section 4.2.1 we present our analysis when **A3** holds, and in Section 4.2.2 we relax this assumption. In particular, we present a method to detect the overlapped regions in the two images, where **A3** holds.

4.2 Using the Kendall Distance

The Kendall distance [10], [11] is defined as the number of pairwise order inversions between the two sequences $[N]$ and σ. Two pairs of matched features, $m(i) = (p_i, q_{\sigma(i)})$, and $m(j) = (p_j, q_{\sigma(j)})$, have *order inversion* if the orders (i,j) and $(\sigma(i), \sigma(j))$ are inverted.

Formally, let us define a binary function for an inversion between $m(i)$ and $m(j)$ from right to left, $\eta^r_\sigma(i,j) = 1$ if $i < j$ & $\sigma(i) > \sigma(j)$. Similarly, from left to right, $\eta^\ell_\sigma(i,j) = 1$ if $i > j$ & $\sigma(i) < \sigma(j)$. An inversion between $m(i)$ and $m(j)$ is defined by $\eta_\sigma(i,j) = \eta^r_\sigma(i,j) + \eta^\ell_\sigma(i,j)$.

The Kendall distance is thus given by:

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

An equivalent definition of the Kendall distance, which will be used in this paper, is based on $H_\sigma(i)$, the number of inversions of $m(i)$ with other matches. Let $H_\sigma(i) = H^\ell_\sigma(i) + H^r_\sigma(i)$, where $H^\ell_\sigma(i) = \sum_{j<i} \eta^\ell_\sigma(i,j)$ and $H^r_\sigma(i) = \sum_{j>i} \eta^r_\sigma(i,j)$. The Kendall distance is given by

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

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 (see Figure 3). That is,

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

The expected values of K_G and K_B are given directly by assumptions **A1** and **A2**. We will show in Section 4.2.1 the expected value of K_{GB} under assumption **A3**. For $N_G = N$ (i.e., $|B| = 0$), it follows directly from **A1** that $K = K_G = 0$. On the other hand, 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 Equation 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 is 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, the normalized values denoted by $\hat{}$ are given by:

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

Using some algebraic manipulation after substituting the terms in Equation 3 with the terms in Equation 5 and replacing $N_B = N - N_G$, we obtain the following quadratic

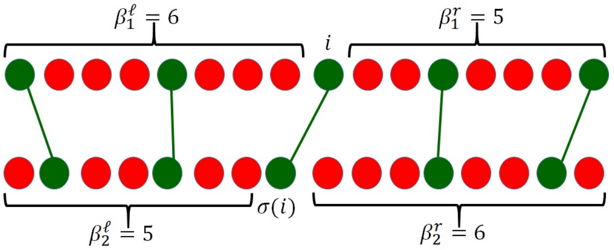


Figure 4: An example of a permutation with β_1^l, β_2^l, β_1^r and β_2^r indicated.

equation in N_G:

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

Note that N is given and $\hat{K}$ can be directly computed from the set of matches, $\mathcal{M}$, using Equation 2. From assumption **A1** it directly follows that $\hat{K}_G = 0$, and 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 the supplementary material, and an alternative proof in [48, p. 257]). We next show that $E(\hat{K}_{GB}) = 1/3$ for fully overlapped sequences.

4.2.1 Full Overlap

Here we consider the case that **A3** holds, i.e., the two images are of the same part of the scene.

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

Proof. Using equations 4&5, the desired value, $E(\hat{K}_{GB})$, can be written as

$$E(\hat{K}_{GB}) = E\left(\frac{\sum_{i\in G} H_\sigma(i)}{N_G N_B}\right) = \frac{1}{N_G N_B}\sum_{i\in G} E(H_\sigma(i)). \tag{7}$$

Since $H_\sigma(i) = H_\sigma^r(i) + H_\sigma^\ell(i)$, it is sufficient to determine both $E(H_\sigma^r(i))$ and $E(H_\sigma^\ell(i))$. Denote by β_1^l the number of bad indexes to the left of i and by β_2^l the number of bad indexes to the left of $\sigma(i)$ (see Figure 4). We first derive $E(H_\sigma^\ell(i)|\beta_1^l,\beta_2^l)$ and $E(H_\sigma^r(i)|\beta_1^l,\beta_2^l)$ for $i \in G$, using assumption **A2**. We then use the expected values of β_1^l and β_2^l to finally derive $E(H_\sigma^\ell(i))$ and $E(H_\sigma^r(i))$.

Consider the hypergeometric probability density function (PDF), $\mathcal{H}(n,k;M,K)$. It is the probability for k successes out of n draws *without replacement* from a population of size M that contains exactly K successes. The analogue for the distribution of $H_\sigma^\ell(i)$ is that a draw is a bad index j to the left of i, and a success is an inversion of $m(j)$ with $m(i)$. That is, the bad index to the left of i is matched to a bad index to the right of $\sigma(i)$. Hence, $M = N_B$, $K = N_B - \beta_2^l$, which is the number of bad indexes to the right of $\sigma(i)$, $k = H_\sigma^\ell(i)$, and $n = \beta_1^l$. It is well known that the expectation of $\mathcal{H}(n,k;M,K)$ is given by $E(\mathcal{H}(n,k;M,K)) = nK/M$; thus,

$$E(H_\sigma^\ell(i)|\beta_1^\ell,\beta_2^\ell) = \frac{\beta_1^\ell(N_B - \beta_2^\ell)}{N_B},$$

and similarly for $H_\sigma^r(i)$:

$$E(H_\sigma^r(i)|\beta_1^\ell,\beta_2^\ell) = \frac{\beta_2^\ell(N_B - \beta_1^\ell)}{N_B}.$$

Using $E(x) = \sum_y E(x|y)P(y)$ (i.e., the law of total expectation), we get:

$$E(H_\sigma^\ell(i)) = \sum_{\beta_1^\ell=0}^{i-1}\;\sum_{\beta_2^\ell=0}^{\sigma(i)-1} E(H_\sigma^\ell(i)|\beta_1^\ell,\beta_2^\ell)P(\beta_1^\ell)P(\beta_2^\ell).$$

Using algebraic manipulation and

$$\begin{aligned} \sum_{\beta_1^\ell=0}^{i-1} \beta_1^\ell P(\beta_1^\ell) &= E(\beta_1^\ell) \\ \sum_{\beta_2^\ell=0}^{\sigma(i)-1} \beta_2^\ell P(\beta_2^\ell) &= E(\beta_2^\ell) \\ \sum_{\beta_1^\ell=0}^{i-1} P(\beta_1^\ell) &= 1, \end{aligned}$$

we obtain

$$E(H_\sigma^\ell(i)) = \frac{1}{N_B}E(\beta_1^\ell)(N_B - E(\beta_2^\ell)).$$

The probabilities, $P(\beta_1^l)$ (and similarly $P(\beta_2^l)$), are considered as another hypergeometric PDF, $\mathcal{H}(n,k;M,K)$. The analogue here is that a draw is an index to the left of i, and a success is an incorrect index to the left of i. Hence, $M = N$, $K = N_B$, $n = i-1$ and $k = \beta_1^l$. The hypergeometric expectation is given by $E(\beta_1^l) = (i-1)N_B/N \approx iN_B/N$. Assumption **A3** implies that $\sigma(i) \approx i$, so approximately $E(\beta_1^l) = E(\beta_2^l) = iN_B/N$; thus,

$$E(H_\sigma^\ell(i)) = E(H_\sigma^r(i)) = N_B\tfrac{i}{N}\left(1 - \tfrac{i}{N}\right),$$

so together

$$\begin{aligned} E(H_\sigma(i)) &= E(H_\sigma^\ell(i)) + E(H_\sigma^r(i)) \\ &= 2N_B\tfrac{i}{N}\left(1 - \tfrac{i}{N}\right). \end{aligned}$$

Substituting $E(H_\sigma(i))$ in Equation 7 we get:

$$\begin{aligned} E(\hat{K}_{GB}) &= \tfrac{1}{N_G N_B}\sum_{i\in G} 2N_B\tfrac{i}{N}\left(1 - \tfrac{i}{N}\right) \\ &= \tfrac{2}{N_G}\left(\tfrac{1}{N}\sum_{i\in G} i - \tfrac{1}{N^2}\sum_{i\in G} i^2\right). \end{aligned}$$

Under assumption **A3**, we approximate the sequence of indices, $i \in G$, by an arithmetic sequence, $a_k = kd = i$ for $1 \le k \le N_G$, where $d = \left\lfloor \frac{N}{N_G} \right\rfloor$. The sum of an arithmetic sequence is given by $\sum_{k=1}^{n} a_k = n(a_1 + a_n)/2$; thus, for our sequence, $\sum_{k=1}^{N_G} a_k = N_G(\left\lfloor \frac{N}{N_G} \right\rfloor + N_G\left\lfloor \frac{N}{N_G} \right\rfloor)/2 \approx \frac{1}{2}N_G N$. In the supplementary material it is shown that $\sum_{i\in G} i^2 \approx \frac{1}{3}N_G N^2$; thus,

$$\begin{aligned} E(\hat{K}_{GB}) &= \tfrac{2}{N_G}\left(\tfrac{1}{N}\sum_{i\in G} i - \tfrac{1}{N^2}\sum_{i\in G} i^2\right) \\ &= \tfrac{2}{N_G}\left(\tfrac{1}{N}\tfrac{1}{2}N_G N - \tfrac{1}{N^2}\tfrac{1}{3}N_G N^2\right) = \tfrac{1}{3}. \end{aligned}$$

□

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

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

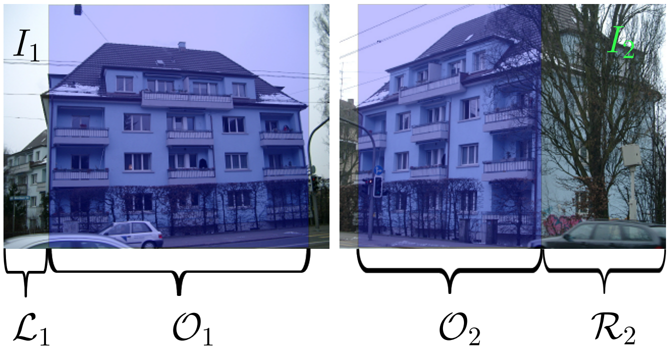


Figure 5: A partial overlap between a pair of images from the ZuBuD dataset.

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

4.2.2 Partial Overlap

When parts of the scene appear in only one of the images (e.g., Figure 5), assumption **A3** does not hold. In this case, the estimation of N_G given by Equation 8 is an underestimation of N_G (see Claim 2 below). We next describe a method to estimate N_G by computing the regions of overlap, using this observation.

We assume that each image can be divided into at most three regions, left, center, and right. The center region corresponds to the same scene region which appears in the other image. Formally, the sequence $[N]$ is partitioned into three intervals: $\mathcal{L}_1$, $\mathcal{O}_1$ and $\mathcal{R}_1$, where $\mathcal{L}_1$ and $\mathcal{R}_1$ consist of features that appear only in I_1 and hence no correct matches exist for them. The lowest and highest ranks of the features with correct matches in I_1, ℓ_1 and h_1 define the interval $\mathcal{O}_1 = [\ell_1, h_1]$. Similarly, we define $\mathcal{O}_2 = [\ell_2, h_2]$.

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 (Section 4.2.1), we discard such indexes and define the *fully overlapped subsequences*, $\hat{\mathcal{O}} = (\hat{\mathcal{O}}_1, \hat{\mathcal{O}}_2)$, as follows:

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

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

The values of N, N_G, and $\hat{K}$ on the sequences defined by a candidate pair $\omega = (\hat{\mathcal{O}}_1, \hat{\mathcal{O}}_2)$ are given by $N(\omega)$, $N_G(\omega)$, and $\hat{K}(\omega)$. These new values can be used in Equation 8 to compute $N_G(\omega)$ by solving the quadratic equation as before:

$$\begin{aligned} 0 = \ & \tfrac{1}{6}N_G^2(\omega) - (\tfrac{1}{2} - \tfrac{1}{3}N(\omega))N_G(\omega) \\ & -N(\omega)(N(\omega)-1)(\tfrac{1}{2} - \hat{K}(\omega)). \end{aligned} \tag{10}$$

The following claim allows us to determine ω^*, which is the region of overlap between the pair of images, and hence $N_G(\omega^*)$.

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

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

We use assumptions **A1-A3** to prove it (see the appendix attached to the supplementary material). That is, the desired N_G is the maximal value obtained for all possible ω's.

For efficiency, we avoid considering all possible ω's given by the 4-tuples. Instead, we consider only a sample of the subset of intervals defined by a single parameter q. The set of intervals considered is as follows:

$$S_q = \{(\ell, h) | \ell = tq + 1, h = t'q,\ 1, 0 < t < t' \leq \frac{N}{q}\}.$$

The value of $N_G(\omega)$ is computed for each $\omega \in S_q \times S_q$. We refer to this algorithm, which considers all the q-intervals in I_1 and I_2, as $\boldsymbol{K_2}$.

To further improve efficiency, we compute sequentially the max value obtained for the q-intervals of I_1 and the entire sequence of I_2. That is, we sample the value $N_G(\omega)$ on $\omega \in [N] \times S_q$. Then, we fix the detected optimal q-interval, $[\ell_1^*, h_1^*]$, in I_1 and search over all the q-intervals of I_2. That is, we sample the value $N_G(\omega)$ on $\omega = [\ell_1^*, h_1^*] \times S_q$ to arrive at our final estimate for ω^* and $N_G(\omega^*)$. We refer to this algorithm as the $\boldsymbol{K_1}$ algorithm.

4.2.3 Kendall Distance Computation & Complexity

The Kendall distance can be computed on a sequence of length N in $\mathcal{O}(N \log N)$ steps using the merge sort algorithm [49], 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, we need to compute w^* (Equation 11). It requires multiple Kendall distance computations for various intervals. We compute these distances efficiently by first computing the Kendall distance for q disjoint intervals of length N/q, using the merge sort algorithm. The Kendall distance of an interval of size dN/q is obtained by counting the inversions when merging two successive intervals of size d_1N/q and d_2N/q, where $d_1 + d_2 = d$. Using this method, the time complexity for the K_2 method is $\mathcal{O}(N \log N)$, where typically in our implementation the constant is ~ 21 when $q = 10$. Similarly for the K_1 method, the time complexity is $\mathcal{O}(N \log N)$, where typically in our implementation the constant is ~ 2. The full analysis of the complexity is given in the supplementary material.

4.3 The Spearman Footrule Distance

We next present an alternative distance measure between permutations for computing N_G. The Spearman Footrule (SF) distance [10], [11] is defined to be the sum of rank differences between matching features. That is, let $(p_i, q_{\sigma(i)}) \in \mathcal{M}$, and denote by $\lambda(i) = |\sigma(i) - i|$ the absolute difference in the ranks of p_i and $q_{\sigma(i)}$ in the sequences of I_1 and I_2, respectively. Then the SF distance is defined by

$$D = \sum_{i=1}^{N} \lambda(i). \tag{12}$$

Let the sets of absolute rank differences in G and B be $\Lambda_G = \{\lambda(i) \mid i \in G\}$ and $\Lambda_B = \{\lambda(i) \mid i \in B\}$, respectively.

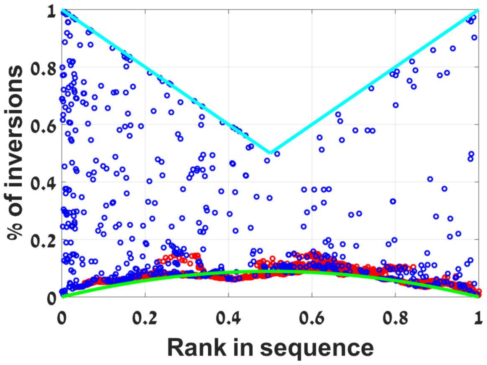


Figure 6: A graph of order inversions, H_σ. The x-axis corresponds to the rank of the feature in I_1, and the y-axis corresponds to the percentage of order inversions out of the maximal value, $N-1$. The blue and red dots correspond to the inliers and outliers, respectively (computed by USAC). The green and cyan lines correspond to the range of inversions for the incorrect matches.

The means of these sets are given by

$$\begin{aligned}\overline{\Lambda}_G &= \frac{1}{N_G}\sum_{i\in G}\lambda(i),\\ \overline{\Lambda}_B &= \frac{1}{N_B}\sum_{i\in B}\lambda(i).\end{aligned}$$

To show that the value D is a function of $E(\Lambda_G)$, $E(\Lambda_B)$, N and N_G, we split the sum in Equation 12 into two terms and obtain:

$$\begin{aligned}D &= \sum_{i\in G}\lambda(i)+\sum_{i\in B}\lambda(i)\\ &= \sum_{i\in G}\frac{\lambda(i)}{N_G}N_G+\sum_{i\in B}\frac{\lambda(i)}{N_B}N_B \qquad (13)\\ &= \overline{\Lambda}_G N_G+\overline{\Lambda}_B N_B.\end{aligned}$$

It follows that N_G can be directly computed using the estimated expectations $\overline{\Lambda}_B$ and $\overline{\Lambda}_G$.

Generally, $\overline{\Lambda}_G > 0$. However, when full overlap is considered, it follows directly from assumption **A3** that $\overline{\Lambda}_G \approx 0$, i.e., $\overline{\Lambda}_G$ is small. We use assumption **A2**, that the spatial order of the outliers is random, to show that $\overline{\Lambda}_B = N/3$ (see [50]). Using some algebraic manipulation after substituting the terms in Equation 13 and replacing $N_B = N - N_G$, we obtain the following equation in N_G:

$$N_G = N - \frac{3D}{N}. \qquad (14)$$

When partial overlap is considered, we claim that, similarly to the method for partial overlap in the Kendall distance (Section 4.2.2), the estimation of N_G given by Equation 14 reaches its maximal value for the fully overlapped sequences, $\hat{O}_1$ and $\hat{O}_2$. Therefore, we maximize N_G in Equation 14 (rather than Equation 10) using the same search method on intervals.

5 Matching Probabilities

We propose to compute a probability, $P_K(i)$, that a match $m(i) = (p_i, q_{\sigma(i)})$ is correct. We do so using the number of inversions of matching pairs with $m(i)$ as well as the estimation of $\hat{O}_1, \hat{O}_2$ and N_G. This probability can then be used as a likelihood function for sampling matches, in particular in guided RANSAC methods (Section 1). We present the full derivations of $P_K(i)$. However, due to efficiency considerations, in practice we use only its approximations.

When $i \notin \hat{O}_1$ or $\sigma(i) \notin \hat{O}_2$, we set $P_K(i) = 0$ (or a small value), since, by the definition of $\hat{O}_1$ and $\hat{O}_2$, $i \in B$. Hence, w.l.o.g. we consider here only fully overlapping sequences. A typical example of inversions in a fully overlapped permutation is presented in Figure 6, where the number of inversions, $H_\sigma(i)$, is plotted as a function of the index i. As expected from the analysis presented in Section 4.2.1, the distribution of $i \in G$ (red dots) is around the function $\frac{i}{N}(1-\frac{i}{N})$, where $N = |\hat{O}_1|$. For $i \in B$ (blue dots), $H_\sigma(i)$ is approximately uniformly distributed in a range that depends on i.

We next present the computation of $P_K(i)$ for $i \in G$, given N_G, $H^1_\sigma(i)$ and $H^2_\sigma(i)$ (defined in Section 4.2). Using Bayes' theorem it is given that

$$\begin{aligned}P_K(i) &= P\left(i\in G\middle|H^1_\sigma(i),H^2_\sigma(i)\right)\\ &= \frac{P\left(H^1_\sigma(i),H^2_\sigma(i)\middle|i\in G\right)P\left(i\in G\right)}{P\left(H^1_\sigma(i),H^2_\sigma(i)\right)}.\end{aligned} \qquad (15)$$

Let us use the following notations: $P_{H|G} = P\left(H^1_\sigma(i),H^2_\sigma(i)\middle|i\in G\right)$, $P_{H|B} = P\left(H^1_\sigma(i),H^2_\sigma(i)\middle|i\notin G\right)$, $P_G = P(i\in G)$ and $P_B = P(i\notin G) = 1-P_G$. Then, we also use the law of total probability for the denominator and obtain:

$$P_K(i) = \frac{P_{H|G}P_G}{P_{H|G}P_G+P_{H|B}P_B}. \qquad (16)$$

It is left to show how each of the right-hand terms in the above equation is estimated. Under the assumption of fully overlapped sequences, the probabilities P_G and P_B are given by the ratio of the correct and the incorrect matches to N, respectively; that is, $P_G = N_G/N$ and $P_B = N_B/N$. We next describe our estimation of $P_{H|G}$.

Estimating $\mathbf{P_{H|G}}$

Under assumption **A1**, correct matches are not inverted with other correct matches. Hence, only inversions with incorrect matches are considered for computing $P_{H|G}$. Let S_β be the set of all possible $\beta = (\beta^l_1, \beta^l_2)$ values:

$$S_\beta = \left\{\beta\middle|0\le\beta^l_1\le N_B,\ \ 0\le\beta^l_2\le N_B\right\}.$$

Recall that β^l_1 and β^l_2 are defined in Section 4.2.1, where β^l_1 is the number of indexes with incorrect matches to the left of i and β^l_2 is the number of indexes to the left of $\sigma(i)$.

Since β is unknown, we compute $P_{H|G}$ using the law of total probability over the set of possible values, $\beta \in S_\beta$:

$$P_{H|G} = \sum_{\beta\in S_\beta} P(H^1_\sigma(i),H^2_\sigma(i)|i\in G,\beta)P(\beta). \qquad (17)$$

For efficiency, $P_{H|G}$ is approximated by ignoring terms in the sum that are likely to be negligible; we use only k values of $\beta \in S_\beta$ where β^l_1 and β^l_2 are close to their expectations given by $(i-1)P_B$ and $(\sigma(i)-1)P_B$, respectively. In our implementation we take the 5 values in a small window around each expectation (2 lower and 2 higher), resulting in $k = 25$.

The probability $P(\beta)$ is given by $P(\beta) = P(\beta^l_1)P(\beta^l_2)$

since β_1^l and β_2^l are assumed to be independently distributed. That is, $P(\beta)$ is the probability that *both* i and $\sigma(i)$ have β_1^l and β_2^l incorrectly matched indices to the left of them. To compute $P(\beta_1^l)$ (and similarly $P(\beta_2^l)$), we consider the hypergeometric PDF, $\mathcal{H}(n, k; M, K)$ (see proof of Claim 1). Recall that the analogue here is that a draw is an index to the left of i, and a success is an incorrect index to the left of i. Hence, $M = N$, $K = N_B$, $n = i - 1$ and $k = \beta_1^l$. Putting it all together, we obtain:

$$P(\beta) = \mathcal{H}(i-1, \beta_1^l; N, N_B)\mathcal{H}(\sigma(i)-1, \beta_2^l; N, N_B).$$

We next estimate the probability $P(H_\sigma^1(i), H_\sigma^2(i)|i \in G, \beta)$ of Equation 17, where *both* $H_\sigma^1(i)$ and $H_\sigma^2(i)$ inversions occur given β. Under the independence assumption, it is given by:

$$\begin{aligned} &P(H_\sigma^1(i), H_\sigma^2(i)|i \in G, \beta) \\ &\qquad = P(H_\sigma^1(i)|i \in G, \beta)P(H_\sigma^2(i)|i \in G, \beta). \end{aligned}$$

The probability $P(H_\sigma^1(i)|i \in G, \beta)$ (and similarly $P(H_\sigma^2(i)|i \in G, \beta)$) is modeled as another hypergeometric PDF, $\mathcal{H}(n, k; N, K)$, discussed in the proof of Claim 1. Recall that the analogue here is that a draw is a bad index j to the left of i, and a success is an inversion m_j with m_i. Hence, $N = N_B$, $K = N_B - \beta_2^l$ which is the number of bad indexes to the right of $\sigma(i)$, $k = H_\sigma^1(i)$, $n = \beta_1^l$. Hence,

$$\begin{aligned} &P(H_\sigma^1(i)|i \in G, \beta) \\ &= \mathcal{H}(\beta_1^l, H_\sigma^1(i); N_B, N_B - \beta_2^l)\mathcal{H}(\beta_2^l, H_\sigma^2(i); N_B, N_B - \beta_1^l). \end{aligned}$$

For efficiency, we approximate the hypergeometric PDFs by Gaussian PDFs with the same mean, nK/M, and variance, $\frac{nK(N-K)(N-n)}{N^2(N-1)}$, of the hypergeometric PDFs.

Estimating $\mathbf{P_{H|B}}$:
Here we consider the number of inversions of a *bad* index, i. Similarly to $P_{H|G}$, $P_{H|B}$ is given by

$$P_{H|B} = \sum_{\beta \in S_\beta} P(H_\sigma^1(i), H_\sigma^2(i)|i \in B, \beta)P(\beta),$$

where $P(\beta)$ is defined above. We first note that an incorrect match is inverted with both correct and incorrect matches.

The number of inversions with correct matches can be directly computed given β, i and $\sigma(i)$. It is given by the difference in the number of good indices to the left of i and to the left of $\sigma(i)$; that is, $\gamma = |(i - 1 - \beta_1^l) - (\sigma(i) - 1 - \beta_2^l)|$. This follows from the assumption that the order of correct matches is preserved (assumption **A1**).

The inversions due to only incorrect matches are given by $H_\sigma^1(i) - \gamma$ and $H_\sigma^2(i) - \gamma$. To compute $P_{H|B}$, we use the same probabilistic derivations as for $P_{H|G}$, while using $H_\sigma^1(i) - \gamma$ and $H_\sigma^2(i) - \gamma$ instead of $H_\sigma^1(i)$ and $H_\sigma^2(i)$.

An approximation of $P_{H|B}$ is required since the above computations are time consuming. Let $H_{high}(i)$ and $H_{low}(i)$ be, respectively, the high and low boundaries for $H_\sigma(i)$ (see cyan and green curves in Figure 6). $H_{high}(i)$ is given for the case where $|\sigma(i) - i|$ is the largest, which is either $\sigma(i) = 1$ (i.e., $i > N - i$) or $\sigma(i) = N$ (i.e., $i < N - i$). That is, when $i > N - i$, then $H_{high}(i) = i - 1$, and when $i < N - i$, then $H_{high}(i) = N - i - 1$.

In theory, $H_{low}(i) = 0$. However, in practice it is unlikely that $H_\sigma(i)$ will be lower than the expected number of inversions as if it were a correct match (i.e., on the green curve in Figure 6). Thus, we set $H_{low}(i) = E(H_\sigma(i)) = 2N_B\frac{i}{N}\left(1 - \frac{i}{N}\right)$, where $i \in G$ (as in the proof of Claim 1). We then model $P_{H|B}$ as a uniform distribution in the range $[H_{low}(i), H_{high}(i)]$; that is,

$$P_{H|B} = \begin{cases} \frac{1}{H_{high}(i) - H_{low}(i)} & H_\sigma(i) \in [H_{low}(i), H_{high}(i)] \\ 0 & H_\sigma(i) \notin [H_{low}(i), H_{high}(i)]. \end{cases}$$

DS	Value	K_{GT}	K	K_2	K_1	S_{GT}	S	S_2	S_1
T1	$\mu(\mathbf{N_G})$	0.6	14.5	3.2	4	1.1	21.4	10.8	9.7
	$\mu(\mathcal{O})$	–	–	0.91	0.89	–	–	0.8	0.85
	Runtime	–	0.6	45.1	6.3	–	0.01	322	120
T2	$\mu(\mathbf{N_G})$	0.5	10.3	3.2	3.6	0.8	18.3	8.6	7.5
	$\mu(\mathcal{O})$	–	–	0.9	0.8	–	–	0.84	0.86
	Runtime	–	0.6	42.3	6.1	–	0.01	319	116

Table 1: The mean normalized absolute error (percentage) in the estimation of N_G in the synthetic datasets. The first and last 3 rows correspond to datasets "Test 1" and "Test 2", respectively. Irrelevant data is marked by '–'.

6 Experiments

Our algorithms are implemented in MATLAB and tested on an Intel i3-2130 CPU with 12 GB of RAM. We present the results of computing N_G and the overlapping region estimations on both synthetic and real data (Section 6.1), and then we demonstrate the effectiveness of our method for the three proposed applications (Section 6.2). Finally, we present in Section 6.3 experiments to show the validity of our assumptions and the robustness of our methods.

For all experiments we extracted SIFT features [7] using VLFeat [51] with the default parameters. The number of features per image varies from a few hundreds to a few thousands depending on the dataset. To match the features between a pair of images, we used the Lowe ratio test [7], i.e., the ratio of the distances to the closest NN and the second closest NN must be < 0.8.

6.1 Estimation of N_G and the Overlapped Regions

We tested 6 variants of our method for computing N_G using the Kendall and the Spearman Footrule distances. For each distance between permutations we considered the computation of N_G on the entire image without removing the image margins ($\boldsymbol{K}$ and $\boldsymbol{S}$ algorithms). In addition, we considered two variants in which the margins were removed, where the window of overlap was computed using sequential sampling ($\boldsymbol{K_1}$ and $\boldsymbol{S_1}$) and simultaneous sampling ($\boldsymbol{K_2}$ and $\boldsymbol{S_2}$) (see Section 4.2.2). The results, in these cases, were the estimates N_G as well as the overlapped regions $\mathcal{O}_1$ and $\mathcal{O}_2$. For evaluation purposes we also considered $\boldsymbol{K_{GT}}$ and $\boldsymbol{S_{GT}}$ (for Kendall and SF, respectively), where the ground truth (GT) of $\mathcal{O}_1$ and $\mathcal{O}_2$ was given, and only N_G was computed.

We defined $\mu(N_G)$ to be the percent of error out of N, i.e., $|N_G - N_G^{GT}|/N$. The accuracy of the overlap was measured using intersection over union (IoU) of the detected interval with respect to the ground truth: $\mu(\mathcal{O}) = |\mathcal{O} \cap \mathcal{O}_{GT}|/|\mathcal{O} \cup \mathcal{O}_{GT}|$. For each pair of images, we averaged the IoU for the two detected intervals, $\mathcal{O}_1$ and $\mathcal{O}_2$.

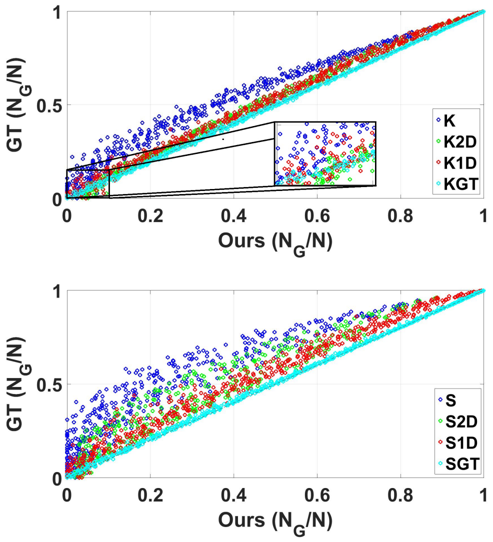


Figure 7: Scatter plots for synthetic experiment Test2; the top and bottom scatter plots are for the Kendall and Spearman distances, respectively.

6.1.1 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 1: We set $N_G = 300$; the size and location of $\mathcal{O}_1$ and $\mathcal{O}_2$ were randomly chosen ($|\mathcal{O}_1|, |\mathcal{O}_2| > N_G$). Table. 1 presents $\mu(N_G)$, $\mu(\mathcal{O})$ and the mean runtime (displayed in milliseconds).

As expected, the best results were obtained for $\boldsymbol{K_{GT}}$ where the overlapped windows, $\mathcal{O}_1$ and $\mathcal{O}_2$, are given. The worst results were obtained for $\boldsymbol{K}$ since **A3** does not hold when there exists only a partial overlap ($|\mathcal{O}_1|, |\mathcal{O}_2| < N$), and hence N_G is underestimated. Finally, $\boldsymbol{K_1}$ was less accurate than $\boldsymbol{K_2}$ since the former traverses only a subset of overlap windows. The accuracy of the computed intervals $\mathcal{O}_1$ and $\mathcal{O}_2$ was similar for both $\boldsymbol{K_1}$ and $\boldsymbol{K_2}$ ($\mu(\mathcal{O}) = 0.89$ and $\mu(\mathcal{O}) = 0.91$). The runtime of $\boldsymbol{K}$ was an order of magnitude faster than $\boldsymbol{K_1}$ and $\boldsymbol{K_2}$ (0.6ms vs 6.3ms & 45.1ms), but the error was unacceptably large (14.5%). The results obtained by $\boldsymbol{K_1}$ and $\boldsymbol{K_2}$ were very good and comparable (4% and 3.2%). However, $\boldsymbol{K_1}$ was an order of magnitude faster than $\boldsymbol{K_2}$.

The results of the four Spearman based methods followed a similar trend, except for $\boldsymbol{S_1}$, which was slightly more accurate than $\boldsymbol{S_2}$. Overall, the SF based methods were less accurate both in computing N_G and the overlapped regions, $\mathcal{O}_1$ and $\mathcal{O}_2$. In addition, SF was more time consuming in our implementation. Hence, Kendall is superior to Spearman.

Test 2: We set N_G between 0 to 1000, and randomly chose $\mathcal{O}_1$ and $\mathcal{O}_2$ as in Test 1. The results of are presented in Table 1, and are similar both in accuracy and runtime to those of Test 1. In addition, scatter plots of the eight algorithms (four for Kendall and four for SF) with respect to the GT are presented in Figure 7. A perfect score corresponds to the diagonal. As expected, the $\boldsymbol{K}$ and $\boldsymbol{S}$ algorithms underestimated N_G, and hence their results are above the diagonal. $\boldsymbol{K_2}$ and $\boldsymbol{K_1}$ are similar in accuracy and slightly underestimate N_G. The results for $\boldsymbol{K_{GT}}$, when the ground truth windows were given, were very close to the diagonal.

These results are probably due to the sampling used for estimating $\mathcal{O}_1$ and $\mathcal{O}_2$, since for $\boldsymbol{K_{GT}}$, the errors were negligible. For both tests, $\boldsymbol{K_1}$ was almost an order of magnitude faster than $\boldsymbol{K_2}$. The $\boldsymbol{S_2}$ and $\boldsymbol{S_1}$ algorithms were similar in their error, $\mu(N_G)$, however, worse than the $\boldsymbol{K_2}$ and $\boldsymbol{K_1}$ algorithms. Note that as expected, when the inlier rate, N_G/N, is very low (see the "zoom-in" in Figure 7), the results are less accurate (and more variant) than when the inlier rate is high.

6.1.2 Real Data

We used the Middlebury 2005, 2006, and 2014 datasets [52], [53], [54], for real data with GT (denoted by MFull05&06 and MFull14, respectively). We also evaluated our method on the USAC dataset [2] and BLOGS dataset [6] (combined), the ZuBuD dataset [55], and the Yorkminster [56], for which no GT is available (denoted by U&B, ZuBuD and York, respectively). We ran the BEEM [1] and USAC [2] algorithms (both with their default settings), where the number of inliers returned is compared to N_G. The lowest and highest inlier indexes of the computed matched features were used as the GT for $\mathcal{O}_1$ and $\mathcal{O}_2$. Figure 8 presents the complete comparison details for the datasets, which are described next.

6.1.3 Middlebury Full Overlap

The mean errors and runtimes (in milliseconds) are presented in Table 2. The errors, $\mu(N_G)$, were similar for $\boldsymbol{K}$, $\boldsymbol{K_2}$ and $\boldsymbol{K_1}$, while $\boldsymbol{K_2}$ was much slower. The errors for $\boldsymbol{S}$, $\boldsymbol{S_2}$, $\boldsymbol{S_1}$ were also similar, however, better than the Kendall methods in MFull14 and worse in MFull05&06. The SF methods, $\boldsymbol{S_2}$ and $\boldsymbol{S_1}$, were significantly slower than the Kendall methods, $\boldsymbol{K_2}$ and $\boldsymbol{K_1}$, while the $\boldsymbol{S}$ method was significantly faster than $\boldsymbol{K}$. $\boldsymbol{K_1}$ and BEEM were similar in $\mu(N_G)$, while USAC was more accurate. The runtime of $\boldsymbol{K_1}$ was between one and two orders of magnitude faster than USAC and BEEM runtimes.

6.1.4 Middlebury Partial Overlap

The overlap between image pairs in the Middlebury datasets is very large (about 90% of the image). Therefore, we vertically cut them to obtain smaller overlaps, where the location of the cut was chosen randomly. This enabled us to test the partial-overlap method (Section 4.2.2) against the GT (Table 2). These datasets are denoted by MPartial05&06 and MPartial14 for Middlebury 2005&2006 and 2014, respectively. The errors of both $\boldsymbol{K_2}$ and $\boldsymbol{K_1}$ were low and similar to the full overlap case, demonstrating the success of the partial overlap method. However, the runtime was much faster for $\boldsymbol{K_1}$. The errors for $\boldsymbol{S_2}$ and $\boldsymbol{S_1}$ were low and similar to the full overlap case for MPartial14, however, quite high for MPartial05&06. The error, $\mu(N_G)$, was large for methods $\boldsymbol{K}$ and $\boldsymbol{S}$ since they ignore the margins. As in the full overlap case, BEEM was similar in accuracy to $\boldsymbol{K_2}$ and $\boldsymbol{K_1}$, while USAC was more accurate. The error for the overlap, $\mu(\mathcal{O})$, was 0.89, 0.89, 0.81 and 0.8 for $\boldsymbol{K_2}$, $\boldsymbol{K_1}$, $\boldsymbol{S_2}$ and $\boldsymbol{S_1}$ methods, respectively (see examples in Figure 9).

	MFull14	MFull05&06	MPar14	MPar05&06	U&B	ZuBuD	Yorkminster
#im pairs	23	27	23	27	21	201	14196
GT	1584	2776	712	1032	138	69	31

	μ	t	μ	t	μ	t	μ	t	μ	t	μ	t	μ	t
BEEM	9.9	2065	13	4198	5	1825	5.3	2344	–	1444	–	692	–	1770
USAC	3	1782	3	1670	3.6	1152	4.1	1276	8.6	1440	7.3	1257	4.5	1310
$\boldsymbol{K}$	10	8.2	3.3	13.6	11.5	4.3	23.1	6	11	2.6	9.3	0.9	8.8	0.1
$\boldsymbol{K_2}$	9.9	280	3.3	412	6.8	162	6.3	223	8.8	125	6.1	94	7.3	40
$\boldsymbol{K_1}$	9.9	38	3.3	53	7.1	22.6	6.7	30	9	18	6.5	11	7.8	4.7
$\boldsymbol{S}$	8.7	0.5	5.8	0.51	17.4	0.51	32.9	0.52	11.2	1.3	6.5	0.5	11.8	0.05
$\boldsymbol{S_2}$	8.7	1999	5.7	4401	7.5	834	10.2	1306	8.4	343	6.1	151	9.1	149
$\boldsymbol{S_1}$	8.7	680	5.8	1523	8.3	262	11.7	427	8.7	104	6.1	34	9.2	42

Table 2: The mean normalized absolute error (percentage), μ (short for $\mu(N_G)$), of the Kendall and Spearman based methods in comparison to the ground truth (its mean, N_G^{GT}) for the left four datasets (columns) and to BEEM (its mean, $|G_{BEEM}|$) for the right two datasets (columns). The mean runtime, t, is also presented for each of the methods (in milliseconds).

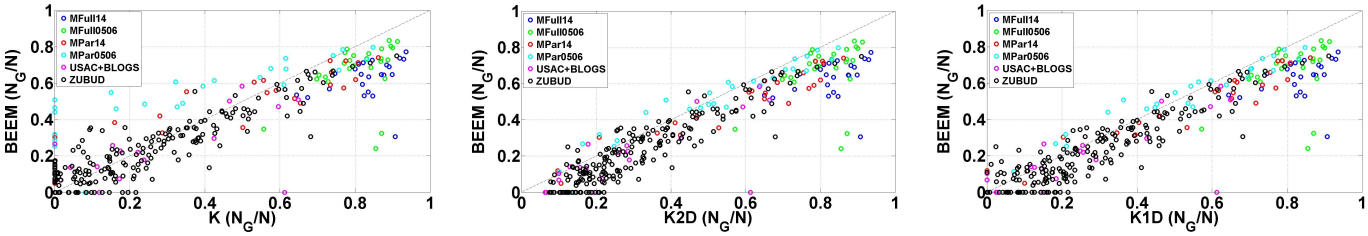


Figure 8: Scatter plots for the real datasets. From left to right: $\boldsymbol{K}$, $\boldsymbol{K_2}$ and $\boldsymbol{K_1}$ methods. The comparison is against BEEM. The axes correspond to N_G/N.

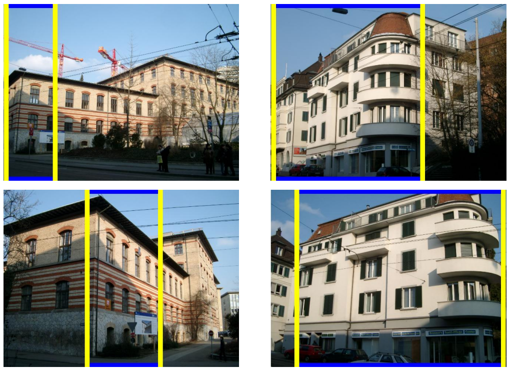


Figure 9: Examples for the estimation of the overlap between pairs of images. The columns correspond to pairs of images from the ZuBuD dataset.

6.1.5 USAC & BLOGS, ZuBuD and Yorkminster

Since no GT is available for these datasets, the estimation results were compared to BEEM (see Table 2). The errors for $\boldsymbol{K_2}$, $\boldsymbol{K_1}$, $\boldsymbol{S_2}$, $\boldsymbol{S_1}$ and USAC were similar ($\boldsymbol{K}$ and $\boldsymbol{S}$ were larger as expected). $\boldsymbol{K_2}$ and $\boldsymbol{K_1}$ were one and two orders of magnitude faster than BEEM and USAC, respectively, while $\boldsymbol{S_2}$ and $\boldsymbol{S_1}$ were about one order of magnitude faster than BEEM and USAC.

Yorkminster: this dataset was collected by [56] for scene reconstruction from an Internet image search engine. It contains some outlier images, i.e., images that do not capture the scene. Since this dataset is very large (3368 images), we use only every twentieth image, starting from the first, to get a total of 169 images. To evaluate the methods we use all $(169 \cdot 168)/2 = 14196$ pairs of images. Note that we present only a summary of the results for this dataset (in Table 2) due to its large size.

6.2 Applications

We next present three applications of N_G estimation, which were briefly discussed in the Introduction.

6.2.1 Halting Condition for RANSAC

Ideally, RANSAC should halt when the cardinality of the inliers set equals the number of correct matches. As this number is usually unknown, the classic halting condition [57, Chapter 11.6] in adaptive RANSAC methods is based on the probability that at least one consensus set was constructed from an uncontaminated minimal set of matches. This probability is computed as a function of the maximal number of inliers, computed up to a given iteration. The method halts when the confidence in the solution is high; the halting condition is based on the size of the consensus set. We propose an alternative halting condition based on the estimated N_G computed by our algorithm. It can be used directly to halt RANSAC, when a consensus set of at least N_G matches has been found.

We tested the effectiveness of the proposed halting condition on the abovementioned datasets. The USAC [2] algorithm was applied using the original halting condition as well as ours. The results of the inlier rate and runtime of the two versions are presented in Figure 10. The total runtime with our halting condition was 69% of the runtime of the original halting condition. This runtime improvement was obtained with nearly no loss of inliers (error of 0.78%). The decrease in runtime (while the accuracy is preserved) is mostly due to image pairs with a low inlier rate; the number of iterations required with the original halting condition is inversely proportional to the inlier rate. Thus, low inlier rates require longer runtimes.

6.2.2 Improving the Efficiency of the SfM Pipeline

A major time-consuming stage in the pipeline of SfM methods [3], [4], [12], [13] is robustly matching all image pairs. The naïve method is (i) to match all pairs of images using ℓ_2 distance and Lowe's ratio test [7], and (ii) apply RANSAC to obtain the fundamental matrices and filter out incorrect matches, which typically takes $\sim 25\%$ of the total runtime of the entire SfM pipeline (e.g., [4]). We followed the dominant paradigm to decrease the runtime of SfM by filtering spatially unrelated image pairs. This reduces the number of pairs for which (ii) is computed.

Existing methods are typically based on $\boldsymbol{BoW}$ [14], which reduces the runtime of both (i) and (ii). In $\boldsymbol{BoW}$, each image is represented by a histogram of the visual words from the vocabulary. The similarity is defined by the ℓ_2 distance between the images' corresponding histograms, weighted by the term frequency to inverse document frequency (tf-idf). Since $\boldsymbol{BoW}$ is agnostic to the spatial configuration of the matches, methods like Hough Pyramid Matching ($\boldsymbol{HPM}$) [18] were proposed to fill in this gap. HPM uses a pyramid to partition the space of similarity transformations into bins. First, putative matches are obtained using a standard visual vocabulary. The local shape, i.e., position, scale and orientation, of the putative matches is then used to assign them to the pyramid bins. Matches that fall into the same bin are more likely to be inliers, while isolated matches are marked as outliers.

Our method filters unrelated images based on the estimated N_G. That is, RANSAC is run only on image pairs with sufficiently large N_G (defined by a threshold). We ran both $\boldsymbol{K_1}$ and $\boldsymbol{K_2}$ on a pair of images after (i) was performed; we denote the first such image by $\boldsymbol{K_1^L}$ and the second by $\boldsymbol{K_2^L}$. The pairwise matching was computed using Lowe's ratio test [7]. We also tested alternative methods, $\boldsymbol{K_1^V}$ and $\boldsymbol{K_2^V}$, for computing N_G based on matches obtained on visual words instead of pairwise matching using (i). This significantly reduced the number of pairs to which (i) was applied and hence the total runtime. Because 1-to-1 matching was not guaranteed, this method resulted in matching ambiguities. To obtain the required 1-to-1 matching, we simply randomly discarded matches.

We compared these variants of our method with filtering unrelated image pairs using $\boldsymbol{BoW}$ and $\boldsymbol{HPM}$. We denote by $\boldsymbol{M} = \{\boldsymbol{K_2^V}, \boldsymbol{K_1^V}, \boldsymbol{K_2^L}, \boldsymbol{K_1^L}, \boldsymbol{BoW}, \boldsymbol{HPM}\}$ the set of considered methods. Each method was tested on the following three datasets: "LunchRoom" [58], "Barcelona" [59] and "Person-Hall" [60], which consist of 72, 191 and 330 images, respectively.

We evaluated the algorithms based on the recall and runtime. The ground truth was taken to be the set of image pairs that have more than 16 inliers according to USAC [2].

We computed the total runtime required to obtain the set of correct matches after RANSAC was applied to the relevant image pairs, assuming that the set of features in each of the images is given. The naïve method consists of computing matches and RANSAC on all image pairs. This was taken as a baseline. For $\boldsymbol{K_1^L}$ (and $\boldsymbol{K_2^L}$), the runtime consisted of computing matches using Lowe's ratio test, computing N_G, and running RANSAC only on image pairs with $N_G > \alpha_1$. For $\boldsymbol{K_1^V}$ (and $\boldsymbol{K_2^V}$), the runtime consisted of computing matches using the vocabulary, computing N_G, and then computing matches using Lowe's ratio test followed by RANSAC only on image pairs with $N_G > \alpha_2$. The runtime of $\boldsymbol{BoW}$ consisted of the same components as $\boldsymbol{K_1^V}$, except for the filtering step. In $\boldsymbol{BoW}$, the runtime of the histogram computations and the tf-idf was taken instead of the N_G computation. A threshold α_3 was used to define the set of filtered pairs. In $\boldsymbol{HPM}$ the runtime included all components of the method including the computation of matches using the vocabulary, as in $\boldsymbol{K_1^V}$,$\boldsymbol{K_2^V}$ and $\boldsymbol{BoW}$.

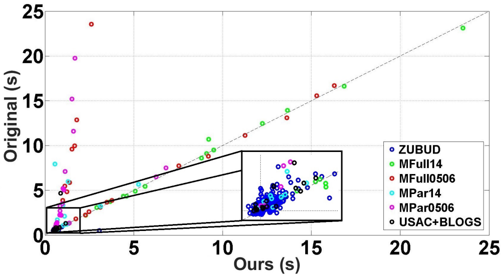


Figure 10: USAC for inlier rate estimation *without* the halting condition (y-axis) vs *with* the halting condition (x-axis).

Let the *runtime-ratio* be defined by the ratio between the runtime of each method and the runtime of the naïve method. We present the runtime-ratio as a function of the recall. The recall was taken to be the ratio between the number of correctly detected overlapped image pairs and the ground truth. The results are presented in Figure 11(a). (Each graph point is defined by the relevant threshold.)

We analyzed the tradeoff between the runtime-ratio and the recall when computing N_G using vocabulary matches ($\boldsymbol{K_1^V}$ and $\boldsymbol{K_2^V}$) or pairwise matches ($\boldsymbol{K_1^L}$ and $\boldsymbol{K_2^L}$). The results show that although the runtime of computing N_G using pairwise matches was longer than when using the vocabulary, its higher recall significantly reduced the number of pairwise RANSAC calls. As a result, $\boldsymbol{K_1^L}$ (and $\boldsymbol{K_2^L}$) performed better than $\boldsymbol{K_1^V}$ (and $\boldsymbol{K_2^V}$). Consider, for example, the recall of 0.8 in the dataset "Barcelona". The runtime ratios are given by 0.35 and 0.39, for $\boldsymbol{K_1^L}$ and $\boldsymbol{K_2^L}$, respectively, (and similarly, by 0.38 and 0.56 for $\boldsymbol{K_1^V}$ and $\boldsymbol{K_2^V}$, respectively.)

Kendall based methods (except for $\boldsymbol{K_1^V}$) generally achieved better results than the $\boldsymbol{BoW}$ method, where in the abovementioned example, the runtime ratio was 0.54. $\boldsymbol{K_2^L}$ generally achieves significantly better results than $\boldsymbol{HPM}$, where in the abovementioned example, the runtime ratio was for $\boldsymbol{HPM}$ was 0.4. $\boldsymbol{K_1^L}$ is slightly worse than $\boldsymbol{HPM}$, while $\boldsymbol{K_1^V}$ and $\boldsymbol{K_2^V}$ are significantly worse than $\boldsymbol{HPM}$.

To analyze the runtime spent on correctly detected pairs, we defined the *effective-runtime ratio* as the ratio between the runtime spent on correctly detected pairs out of the total runtime. The total runtime of each algorithm is defined above. For computing the runtime spent on correctly detected pairs, we discarded the runtime of RANSAC calls spent on incorrectly detected pairs from the total runtime. For the vocabulary based methods, we also discarded the time for computing the pairwise matching using Lowe's ratio test on the incorrect pairs. The effective-runtime ratios as a function of the recall are presented in Figure 11(b). The general behavior was similar to the runtime ratio.

To conclude, the accuracy of the Kendall based algorithms ($\boldsymbol{K_1^L}$ with $\boldsymbol{K_2^L}$) compensated for the time consuming pairwise matching. Hence, discarding unrelated image pairs using the Kendall based algorithms produced superior results than for the $\boldsymbol{BoW}$ algorithm.

6.2.3 Guided RANSAC

Computing matching probabilities, P_K, can be useful for several applications, including guided RANSAC. To evaluate our estimate for P_K, we compared the mean inlier precision of the $x \leq 200$ highest ranked matches, given by P_K (Figure 12), to the Lowe ranking [7], denoted by P_L. The mean precision was calculated on all image pairs from all datasets from Section 6.1.2. 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. The blue curve shows a combination of the two probabilities, given by $P_C = \frac{P_K P_L}{P_K P_L + (1-P_K)(1-P_L)}$. P_C is higher than both P_K and P_L if they are high, and it is lower than both P_K and P_L if they are low. Using low values of x, P_C outperformed its components. For higher values of x, our estimation was better. To conclude, P_K, which is mostly based geometry, is a comparable alternative to P_L, which is solely based on patches' appearance. We note here that we found the actual difference in runtime when using P_K, P_C or P_L in a guided RANSAC method to be negligible.

6.3 Validity of Assumptions

We tested the validity of our assumptions and the sensitivity of our method to these assumptions. Here we present the results and provide an algorithm for using our method to estimate the roll rotation (Section 4.1).

6.3.1 Measured Values of $\hat{K}_G$, $\hat{K}_B$ and $\hat{K}_{GB}$

Assumptions **A1-A3** (Section 4.1) were used to deduce the values $\hat{K}_G = 0$, $\hat{K}_B = 1/2$ and $\hat{K}_{GB} = 1/3$. We tested these values on real data (same datasets as in Section 6.1.2). The three values were measured using the inliers and outliers, taken from the GT in the Middlebury datasets, and from the BEEM estimation in the USAC&BLOGS and ZuBuD datasets.

Table 3 presents the results of this experiment. The mean values for $\hat{K}_G$, $\hat{K}_B$ and $\hat{K}_{GB}$ were 0.09, 0.43 and 0.36, respectively. These values show that our assumptions roughly hold on real datasets. One significant deviation is dataset MFull14, where $\hat{K}_B = 0.28$ and $\hat{K}_{GB} = 0.18$. This is probably due to a skewed distribution of inversions for the incorrect matches of repeated pattern. In some images there is a repeated pattern, e.g., basket, bicycle rim, which is located only in a small part of the image. This is in contrast, for example, to MFull05&06, where repeated patterns are spread over a large part of the image.

We also tested the same values when the sequences, $[N]$ and σ, were defined by the order in the y-axis. The measured values were $\hat{K}_G = 0.02$, $\hat{K}_B = 0.37$ and $\hat{K}_{GB} = 0.25$. This is a significant deviation from the expected values. It suggests that our choice to use the x-order rather than the y-order is expected to yield more accurate results.

	$\hat{\mathbf{K}}_{\mathbf{G}}$	$\hat{\mathbf{K}}_{\mathbf{B}}$	$\hat{\mathbf{K}}_{\mathbf{GB}}$
Assumed	0	0.5	0.33
All Datasets	0.09	0.43	0.36
MFull14	0.02	0.28	0.18
MFull05&06	0.02	0.39	0.3
MPar14	0.05	0.43	0.39
MPar05&06	0.07	0.5	0.38
U&B	0.1	0.46	0.39
ZuBuD	0.1	0.44	0.38
All Datasets Y-Axis	0.02	0.37	0.25

Table 3: Real values for $\hat{K}_G$, $\hat{K}_G$ and $\hat{K}_G$, obtained from the real datasets (see Section 6.1.2). From left to right: the assumed values, the computed values for all datasets, the computed values for the real datasets, and the computed values for all datasets on the y-axis.

6.3.2 Deviation from Assumption **A1**

The assumption that the order of correct matches is preserved (**A1**) only roughly holds in real data ($\hat{K}_G = 0.09$). To systematically test the effect of the deviation from this assumption on the computation of N_G, we repeated the synthetic experiments of Section 6.1.1, with $\hat{K}_G \geq 0$. For $\hat{K}_G = 0$ (the original experiment) the error was $\mu(N_G) = 4$. For $\hat{K}_G \in \{0.002, 0.05, 0.1, 0.15\}$, the errors were $4.1, 6.2, 8.5$ and 10.7, respectively. The results show that our method is robust to a relatively large deviation from assumption **A1**. For example, where $\hat{K}_G = 0.002$ and $\hat{K}_G = 0.02$, which correspond, respectively, to 90 and 900 inversions, the errors were $\mu(N_G) = 4.1$ and $\mu(N_G) = 5$. Note that if we were to switch one-third of the correct matches with their correct neighbor, we would obtain approximately 100 inversions.

6.3.3 The Roll Rotation (z-axis Rotation)

When there is a roll rotation between the cameras, assumption **A1** does not strictly hold. Particularly, the estimated N_G value is expected to be lower than the correct one.

We tested the estimated N_G as a function of the roll angle by generating rotated images using $\alpha \in [-90, 90]$ angles in 10 degree steps. As expected, the value of N_G decreases, but at a moderate rate. Hence, small changes (of < 10 degrees) of the relative roll do not dramatically affect the results.

Moreover, although not the focus of our method, we also tested whether maximal estimate of the N_G value for different rolls can be used to estimate the relative roll. The mean error in the estimated roll angle between the image pairs of the ZuBuD dataset (Section 6.1.5) was ~ 5.7 degrees. In most runs we tested, the maximal value for N_G is given for the correct roll angle. Examples from the ZuBuD dataset is presented in the Appendix (in the supplemental material).

Since the sensitivity of our method to the roll angle is low, its accuracy for roll estimation as is not very high, and other methods specially designed for roll estimation can be used as a preprocessing method [45], [46].

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 derived a probability function for a match to be correct. This was

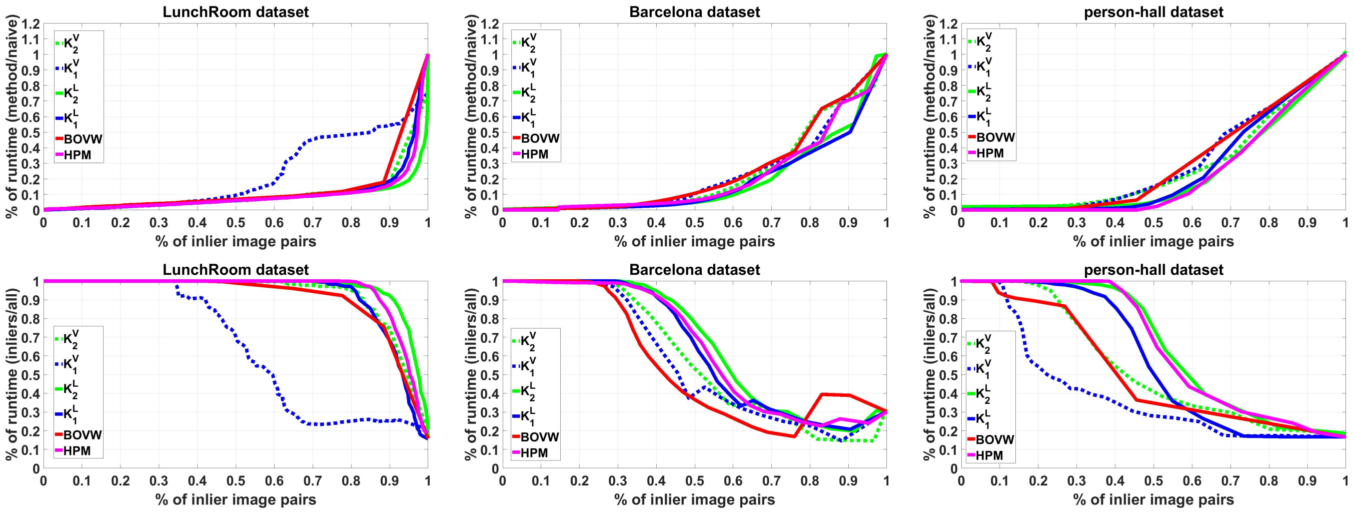


Figure 11: Using our estimate for N_G to discard pairs of images from a SfM pipeline in the "LunchRoom", "Barcelona" and "Person-Hall" datasets (left to right); The continuous and broken blue lines correspond to the $\boldsymbol{K_2^L}$ and $\boldsymbol{K_2^V}$ methods, respectively. Similarly, the green lines correspond to $\boldsymbol{K_1^L}$ and $\boldsymbol{K_1^V}$, respectively. The red and magenta curves correspond to the $\boldsymbol{BoW}$ and $\boldsymbol{HPM}$ methods, respectively. **Top row:** the x-axis corresponds to the percentage of remaining image pairs that have spatial overlap after thresholding. The y-axis corresponds to the percentage of runtime required to process the images in comparison to the näive method. **Bottom row:** the x-axis is the same as in the top row. The y-axis corresponds to the percentage of runtime spent on correctly identifying spatially overlapped image pairs in comparison to the total runtime required to process the images.

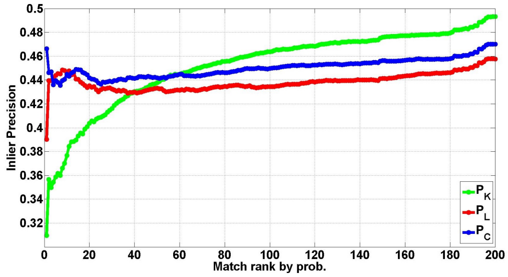


Figure 12: Sampling methods for guided RANSAC using the real datasets (Section 6.1.2). The x-axis corresponds to the rank of the matches when sorted by the probability. The y-axis corresponds to the precision of correct matches.

done using only the spatial order of a given set of matches and some reasonable statistical assumptions. We considered two alternative metrics between permutations, the Kendall and the Spearman distance metrics.

We tested the effectiveness of these estimations on real datasets. Our experiments show that the Kendall based method is superior to the Spearman based method. Our method successfully competes with methods that compute the set of inliers explicitly by recovering the epipolar geometry. However, our method is much faster. Three applications were presented to demonstrate the practicality of our results. This demonstrates the power of analyzing the spatial order of matched features. A limitation of our method is moving objects, which do not agree with assumptions 1-3.

Acknowledgments

This work was partially supported by the Israel Science Foundation, grant no. 930/12, by the Israeli Ministry of Science, grant no. 3-8744, and by the Israeli Innovation Authority in the Ministry of Economy and Industry.

References

[1] L. Goshen and I. Shimshoni, "Balanced exploration and exploitation model search for efficient epipolar geometry estimation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1230–1242, 2008.

[2] 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.*, vol. 35, no. 8, pp. 2022–2038, 2013.

[3] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010.

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

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

[6] 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.*, vol. 35, no. 3, pp. 755–762, 2013.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

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

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

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

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

[12] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion*." *Acta Numerica*, vol. 26, pp. 305–364, 2017.

[13] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *CVPR*, 2016.

[14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007.

[15] T. Shen, S. Zhu, T. Fang, R. Zhang, and L. Quan, "Graph-based consistent matching for structure-from-motion." in *ECCV*, 2016.

[16] M. Havlena, A. Torii, and T. Pajdla, "Efficient structure from motion by graph optimization," *ECCV*, 2010.

[17] N. Snavely, S. M. Seitz, and R. Szeliski, "Skeletal graphs for efficient structure from motion." in *CVPR*, 2008.

[18] Y. Avrithis and G. Tolias, "Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval," *International Journal of Computer Vision*, vol. 107, no. 1, pp. 1–19, 2014.

[19] O. Chum and J. Matas, "Matching with PROSAC-progressive sample consensus," in *CVPR*, 2005.

[20] O. Chum, J. Matas, and J. Kittler, "Locally optimized RANSAC," in *Pattern Recognition*. Springer, 2003, pp. 236–243.

[21] O. Chum and J. Matas, "Optimal randomized RANSAC," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 30, no. 8, pp. 1472–1482, 2008.

[22] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *CVPR*, 2015.

[23] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *ECCV*, 2016.

[24] H. Altwaijry, A. Veit, S. J. Belongie, and C. Tech, "Learning to detect and match keypoints with deep architectures." in *BMVC*, 2016.

[25] W. Hartmann, M. Havlena, and K. Schindler, "Predicting matchability," in *CVPR*, 2014.

[26] 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.

[27] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *CVPR*, 2015.

[28] R. Shah, V. Srivastava, and P. Narayanan, "Geometry-aware feature matching for structure from motion applications," in *WACV*, 2015.

[29] D. Nasuto and J. B. R. Craddock, "Napsac: High noise, high dimensional robust estimation-it's in the bag," in *BMVC*, 2002.

[30] V. Fragoso, P. Sen, S. Rodriguez, and M. Turk, "Evsac: accelerating hypotheses generation by modeling matching scores with extreme value theory," in *ICCV*, 2013.

[31] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[32] G. Tolias, Y. Avrithis, and H. Jégou, "Image search with selective match kernels: aggregation across single and multiple images," *International Journal of Computer Vision*, vol. 116, no. 3, pp. 247–261, 2016.

[33] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2010.

[34] R. Arandjelovic and A. Zisserman, "All about vlad," in *CVPR*, 2013.

[35] J. L. Schönberger, A. C. Berg, and J.-M. Frahm, "Paige: pairwise image geometry encoding for improved efficiency in structure-from-motion," in *CVPR*, 2015.

[36] ——, "Efficient two-view geometry classification," in *GCPR*, 2015.

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

[38] A. L. Yuille and T. Poggio, "A generalized ordering constraint for stereo correspondence," DTIC Document, Tech. Rep., 1984.

[39] R. D. Arnold and T. O. Binford, "Geometric constraints in stereo vision," in *24th Annual Technical Symposium*. International Society for Optics and Photonics, 1980, pp. 281–292.

[40] H. H. Baker, "Depth from edge and intensity based stereo." DTIC Document, Tech. Rep., 1982.

[41] A. Verri, "Methodi matematici per la visione stereografica," Ph.D. dissertation, Ph. D. thesis, University of Genoa, 1984.

[42] L. Talker, Y. Moses, and I. Shimshoni, "Have a look at what I see," *arXiv preprint arXiv:1505.04873*, 2015.

[43] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, pp. 81–93, 1938.

[44] A. F. Bobick and S. S. Intille, "Large occlusion stereo," *International Journal of Computer Vision*, vol. 33, no. 3, pp. 181–200, 1999.

[45] Y. Goldman, E. Rivlin, and I. Shimshoni, "Robust epipolar geometry estimation using noisy pose priors," *Image and Vision Computing*, vol. 67, pp. 16–28, 2017.

[46] S. Mills, "Accelerated relative camera pose from oriented features," in *3DV*, 2015.

[47] J. M. Steele, "Variations on the monotone subsequence theme of erdös and szekeres," in *Discrete probability and algorithms*. Springer, 1995, pp. 111–131.

[48] W. Feller, *An Introduction to Probability Theory and its Applications*. John Wiley & Sons, 2008, vol. 2.

[49] J. Kleinberg and É. Tardos, *Algorithm Design*. Pearson Education India, 2006.

[50] P. Diaconis and R. L. Graham, "Spearman's footrule as a measure of disarray," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 262–268, 1977.

[51] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[52] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *CVPR*, 2007.

[53] H. Hirschmüller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *CVPR*, 2007.

[54] 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*. Springer, 2014, pp. 31–42.

[55] 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*, vol. 260, 2003.

[56] K. Wilson and N. Snavely, "Robust global translations with 1dsfm," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

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

[58] G. Meneghetti, M. Danelljan, M. Felsberg, and K. Nordberg, "Image alignment for panorama stitching in sparsely structured environments," in *SCIA*, 2015.

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

[60] A. Cohen, J. L. Schönberger, P. Speciale, T. Sattler, J.-M. Frahm, and M. Pollefeys, "Indoor-outdoor 3D reconstruction alignment," in *ECCV*, 2016.

Lior Talker received his BA degree in computer science (2005) from the Technion institute in Haifa, and his MSc degree in computer science (2013) from the Interdisciplinary Center in Herzliya. He is currently pursuing a PhD degree in computer vision at Haifa university, under the supervision of Prof. Yael Moses and Prof. Ilan Shimshoni. His research interests include efficient and robust algorithms for processing a large number of images, and computer vision based collaboration between photographers.

Yael Moses received the BA degree in mathematics and computer science (1984) from the Hebrew University and an M.Sc and a Ph.D degrees (1984,1994) in applied mathematics and computer science from the Weizmann Institute of Science. She was a post-doctoral fellow at the Robotics group of Oxford University 1993-1994, and in the Weizmann Institute between the years 1994-1997. Her current position is a Professor in the Efi Arazi School of Computer Science at the Interdisciplinary Center, Herzliya, which she joined in 1999. She served as a deputy dean for several years. She spent the years 2004-2005 on sabbatical in Sydney at NICTA and UNSW, and in 2013 she spent a short sabbatical at Berkeley and Columbia University. Her research interests include multi-camera systems, visual surveillance, and analyzing CrowdCam images.

Ilan Shimshoni received the BSc degree in mathematics and computer science from the Hebrew University in Jerusalem, the MSc degree in computer science from the Weizmann Institute of Science, Rehovot, Israel, and the PhD degree in computer science from the University of Illinois at Urbana Champaign. Currently, he is a professor in the Department of Information Systems at Haifa University. His research interests are computer vision, robotics and computer graphics. He is a member of the IEEE.