Pacific Graphics 2016
E. Grinspun, B. Bickel, and Y. Dobashi
(Guest Editors)

Volume 35 (2016), Number 7

Geometrically Based Linear Iterative Clustering for Quantitative Feature Correspondence

Qingan Yan[1], Long Yang[1], Chao Liang[1], Huajun Liu[1], Ruimin Hu[1] and Chunxia Xiao[2,1]

[1]School of Computer, Wuhan University, Wuhan 430072, China
[2]State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, China

Abstract

A major challenge in feature matching is the lack of objective criteria to determine corresponding points. Recent methods find match candidates first by exploring the proximity in descriptor space, and then rely on a ratio-test strategy to determine final correspondences. However, these measurements are heuristic and subjectively excludes massive true positive correspondences that should be matched. In this paper, we propose a novel feature matching algorithm for image collections, which is capable of providing quantitative depiction to the plausibility of feature matches. We achieve this by exploring the epipolar consistency between feature points and their potential correspondences, and reformulate feature matching as an optimization problem in which the overall geometric inconsistency across the entire image set ought to be minimized. We derive the solution of the optimization problem in a simple linear iterative manner, where a k-means-type approach is designed to automatically generate consistent feature clusters. Experiments show that our method produces precise correspondences on a variety of image sets and retrieves many matches that are subjectively rejected by recent methods. We also demonstrate the usefulness of the framework in structure from motion task for denser point cloud reconstruction.

Categories and Subject Descriptors (according to ACM CCS): I.3.m [Computer Graphics]: Computational photography—; I.4.0 [Computer Graphics]: General—; I.4.7 [Computer Graphics]: Feature Measurement—;

1. Introduction

Establishing feature correspondences from image collections is one of the most fundamental problems in computer graphics and vision communities. Such associated connections indicate the visibility of 3D points in 2D images and form the basis for many applications, including photo ordering [AECO15], object modeling [IBP15] and image navigation [CDSHD13]. On the other hand, identifying feature correspondences is also a challenging problem, as there is no criteria so far that could objectively measure a correspondence is spurious or plausible. In this paper, we try to quantize the correctness of feature correspondences within image collections and use this metric to augment our matching performance.

A typical strategy, for correspondence measuring, is to compare the associated descriptor between features in the Euclidean distance and choose the closest feature in the target image as a visual correspondence. However, this measurement is inaccurate. It improperly associates each feature with a correspondence, and leads to a surge of false positive matches accordingly. Hence, in order to obtain more acceptable results, recent matching methods [SBBF12, YXX14, CLW*14], which differ in nearest neighbor (NN) search, commonly employ a 2-NN *ratio-test* strategy [Low04] to measure matches. This strategy heuristically defines that the matching probability can be approximated by the distance ratio of a query feature between its nearest neighbor and its second nearest neighbor. The underlying assumption is that a good correspondence usually has significant descriptor discrimination against the others.

There are two main deficiencies to these approaches. First, due to viewpoint changes, the closest feature in descriptor space may not be the true match. Another point in proximity with less similarity may be the right one. Second, the ratio-test criteria is easily interfered by repeated patterns in the image content, where its assumption would be violated. It punitively rejects many true positive matches in such repeated regions, as shown in Fig. 1, due to the indistinguishable distribution in descriptor space. In 3D reconstruction applications, such as structure from motion (SfM), this usually results in very sparse point clouds. In this respect the results demonstrated by recent feature matching systems are still less impressive.

In order to achieve desired feature matches, the key technical challenge is that of designing meaningful criteria which can quantitatively measure the plausibility of each feature correspondence. Yet, few approaches exist that attempt to tackle this problem. Comparing feature descriptors is not sufficiently meaningful to indicate a good match. Therefore, in practical applications, a pairwise ge-

Figure 1: *An example of punitive rejection. The two input images (left column) show the structural detail of Arc de Triomphe in Paris. Due to the repetition in carving patterns, the visual similarity of features in these regions are indistinctive. This leads to high distance ratios in the ratio-test and many true matches are thus punitively excluded (middle column: 193 matches using [MA10] followed by RANSAC [FB81]) as compared to our method (right column: 396 matches). Matching points are marked with disks of the same color.*

ometric verification step is usually imposed. Such geometric relationship constrains the possible motion of corresponding points transferred from one image to another and is an effective way to filter outliers, as they are frequently inconsistent according to viewpoint transformation. A seminal example is the epipolar geometry (e.g. *fundamental matrix* [HZ03]) using RANSAC [FB81]. However, recent methods mainly consider this metric as a supplementary post-process to feature matching which would further reduce the number of output correspondences.

In this paper, we consider the problem of designing quantitative criteria for feature matching. As the main contribution, we propose a novel geometrically-based approach for reliable feature correspondence in image collections, which overcomes the two deficiencies suffered by recent methods. We argue that the epipolar geometry suffices to quantize feature matching when analyzed in a more holistic fashion. That is, instead of matching locally for each image pair, we consider feature points of the whole image collection together. We first demonstrate how the epipolar constraint is able to quantitatively depict the correctness of feature correspondences. Based on the measure, we then cast feature matching as an optimization problem and propose a novel objective function to evaluate the overall quality of feature correspondences. Finally, we solve the optimization problem in a linear iteration manner, where a geometrically guided k-means approach is employed to automatically cluster visually closing and geometrically consistent features.

In this article we focus on matching static feature points. We evaluate the presented approach on different types of datasets and show superiority in quality and quantity to recent feature matching methods. Additionally, we also integrate our algorithm into a typical SfM pipeline for match graph construction. Using this information, we succeed to recover denser 3D details that are otherwise difficult to achieve.

2. Related Work

Feature matching is important for many applications. There have been a great deal of literatures on improving the efficiency of feature correspondence. However, rare work are working towards the direction of investigating more meaningful criteria, instead of the ratio-test, to improve matching performance. In this section, we first revisit prior efforts on fast nearest neighbor search and match graph construction in the context of large-scale image collections. Then we discuss the relevant techniques that are commonly used to densify SfM point clouds.

Nearest neighbor search In order to find feature matches, the nearest neighbor in descriptor space is usually required. For a query feature, the brute-force approach exhaustively compares distances with all features in the target image to find the nearest neighbor. With the purpose of improving efficiency, tree-based approximate nearest neighbor (ANN) methods, such as ANN Libary [MA10] and FLANN [ML14], are usually adopted in practice. These methods organize target features into a *kd*-tree and dramatically decrease the comparisons for each query. As an alternative, hashing-based ANN algorithms [SBBF12, CLW*14] recently have also been used in feature matching. These methods convert feature descriptors into bit codes and conduct a bitwise operation for fast similarity comparison. Shah et al. [SSN15] introduce an interesting geometry-aware algorithm which constrains the NN search, for each feature, in a very limited scope (i.e., along the epipolar line) in order to refine their matching accuracy and efficiency. All these work are mainly designed for fast nearest neighbor search. In order to measure matches, the heuristic 2-NN test strategy [Low04] is usually imposed afterward. However, as previously discussed, the nearest neighbor may not be the right match candidate in some cases, and the ratio-test would also punitively exclude many positive matches that should be matched. In contrast, Zhang and Koseckа [ZK06] propose a generalized RANSAC framework for correspondence establishing. While it takes multiple candidate matches into account and can avoid the early commitment to the nearest one in descriptor space, more true positive matches risk rejection in the robust estimation stage.

Large-scale image matching For datasets with thousands of photos, directly matching all possible image pairs via ANN models is intractable. In fact, most of images in a collection observe completely different scenes, thus a large portion of comparisons can be saved if exclude matching these irrelevant image pairs. There are various work built upon this observation. Agarwal *et al*. [ASS*09] integrate the Bag-of-Visual-Word model [NS06] into a distributed SfM system for overlapping images mining. Frahm *et al*. [FFGG*10] leverage GIST descriptors [OT01] to cluster sim-

ilar images and achieve large-scale match graph construction on a single PC using GPUs. Lou *et al*. [LSG12] design an algorithm exploiting connectivity in image collections and incorporate both relevance feedback and entropy minimization to improve retrieval quality over time. Kim *et al*. [KTT*12] reformulate the matching process as a linkage prediction problem. They first approximate the match graph with a very sparse graph, then iteratively increase its potential edges by spectral analysis.

Most recently, Changchang Wu [Wu13] introduces a preemptive matching algorithm by testing a few features with top-scale values to decide whether this image pair should be fully matched. Havlena and Schindler [HS14] show that, given a huge visual vocabulary, the problem of feature matching can be approached by image indexing. Such unique index words enable them to directly establish feature correspondences across all images instead of testing every individual pair. Yan *et al*. [YXX14] argue that while overlapping images contain certain correspondences, most features between them still remain unmatched. Thus instead of retrieving similar images, they present a feature-oriented matching algorithm which intelligently finds a small portion of related candidates for each feature to match. Schonberger *et al*. [SBF15] develop a new approach for quick overlap prediction. They infer the viewpoint relationship of pairwise images by means of encoding location and orientation properties of local features. There is also a work [TL09] which predicts the matchability of individual features and rejects non-matchable ones.

These methods are mainly designed for the purpose of efficiency improvement. Therefore, they also suffer from the preemptive commitment to the nearest neighbor and the impact of punitive rejection. In contrast, we investigate a different perspective, i.e., improving the matching accuracy. Instead of processing in pairwise manner, the proposed approach relies on the "wisdom of crowd", which analyzes the geometric consistency of features over an image collection. With this informative metric, not only can we filter outliers and establish precise feature correspondences, but also find more matches between images that are subjectively discarded by ratio-based methods.

Dense reconstruction Structure from motion (SfM) recently is one of the most popular techniques for 3D acquisition. However, a general deficiency to this technique is the sparseness of its output point clouds. This can be attributed to two reasons: first, local features are much fewer as compared to image pixels, and second, many true positive correspondences are incorrectly eliminated in feature matching. Thus in order to generate denser results, multi-view stereo (MVS) methods [FP10, GSC*07], which serve as a post-processing step to SfM, are usually utilized. For further enhancement, Shan *et al*. [SCF*14] incorporate image silhouette information into an MVS method and estimate a corresponding depth map using MRF to augment point sets around object boundaries. Chaurasia *et al*. [CDSHD13] use superpixels to interpolate the depth value of non-reconstructed pixels for more complete 3D geometry outputs. The focus of these approaches is to densify point clouds after SfM rather than increase feature matches before SfM. In this paper, we do not attempt to solve this specific problem posed by SfM. Instead, we would like to show that our matching algorithm can facilitate SfM to produce much denser geometries as compared to other matching methods.

3. Approach

We first introduce some notations for easy illustration. Formally, we are given a set of images $\mathcal{I} = \{I_1, ..., I_Z\}$ and the associated feature points $\mathcal{P}_i = \{P_1^i, ..., P_{M_i}^i\}$ to each image I_i. For each feature point P_m^i $(1 \le m \le M_i)$, it can be expressed in the form $P_m^i = (\mathbf{x}_m^i, \mathbf{d}_m^i)$, where $\mathbf{x}_m^i \in \mathbb{R}^2$ indicates the spatial location of feature P_m^i in its image plane, and $\mathbf{d}_m^i \in \mathbb{R}^D$ denotes its appearance descriptor, D is the dimensionality of the descriptor vector. Here, we make use of the SIFT [Low04] as our experimental element, where D is 128. Notice that the number of features in each image may be different. We use M_i to record the number in the i-th image.

Our goal in this work is to establish precise and complete feature-wise correspondences. The key challenge for this problem is how to design the matching criteria which is capable of measuring the quality of feature correspondences quantitatively. We approach this by integrating the geometric constraint into correspondence search and analyzing it in a more holistic fashion (rather than conducting on each image pair as a post-process). Typically, a putative feature match should correspond to the same 3D point, thus the matching quality of feature P_m^i and P_n^j between two images I_i, I_j can be validated and augmented with the geometric relationship of multiple additional images which also observe this physical point. That is, even when P_m^i and P_n^j do not have sufficient feature similarity directly, there may be sufficient indirect evidence from their geometric transformations between other images supporting their match.

We first describe our matching criteria in detail in Sec. 3.1. Then we formulate this metric into an optimization problem and provide a linear iterative clustering method to minimize this objective respectively in Sec. 3.2 and Sec. 3.3.

3.1. Epipolar Consistency

Besides of delivering visual information, pictures also reveal the geometric relationship of viewpoint changes. Epipolar geometry is a commonly used technique in image matching systems for outlier filtering. It constrains the possible transformation of corresponding feature points in an image pair. To formulate, given two images I_i and I_j, the epipolar constraint defines that if P_m^i in I_i and P_n^j in I_j are corresponding points, then they should satisfy the following equation:

$$\mathbf{x}_m^{i\mathrm{T}} \cdot \mathcal{F}_{ij} \cdot \mathbf{x}_n^j = 0, \tag{1}$$

where $\mathcal{F}_{ij}$ is the 3×3 *fundamental matrix* [HZ03] between I_i and I_j, T denotes the transpose operation. In other words, this equation also implies that if P_n^j has a correspondence P_m^i in image I_i, then this associated point must lie on the epipolar line $\mathbf{l}_n = \mathcal{F}_{ij}\mathbf{x}_n^j$ corresponding to the point P_n^j; otherwise, it is an outlier. Accordingly, for P_m^i, this constraint also holds.

It is important to note that such geometric cue actually offers us a necessary condition for points to correspond. However, with only two images, the epipolar constraint is insufficient to exactly locate

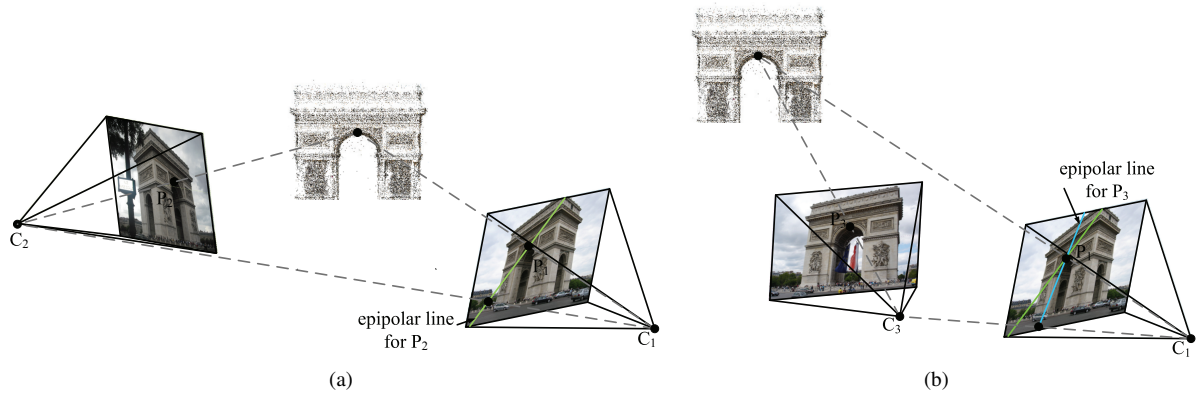


Figure 2: *Illustration of the epipolar consistency. C_1, C_2 and C_3 denote three camera centers. P_1, P_2 and P_3 represent different feature points in these cameras. (a) The plane passing through C_1, C_2 and P_2 forms an epipolar line in camera C_1. (b) With another image, a second line can be computed in camera C_1 from plane $C_1C_3P_3$. If P_1, P_2 and P_3 are corresponding points and epipolar consistent, the intersection point must be P_1.*

the matching point for a given feature P. The other points that lie on the line $\mathbf{l} = \mathcal{F}\mathbf{x}$ also satisfy above relation.

Therefore, instead of analyzing locally within a single pair, we explore the epipolar constraint over the whole dataset. Suppose there is another image I_s. If this image also has a match P_t^s with P_n^j, then P_t^s and P_m^i should be a correspondence as well (i.e., loop-closure constraint). According to Eq. 1, we can get another line $\mathbf{l}_t = \mathcal{F}_{is}\mathbf{x}_t^s$ in image I_i, and it intersects with $\mathbf{l}_n$. Due to the fact that two coplanar lines only determine a point, in theory, if P_n^j, P_t^s are correct matches to P_m^i, the intersection point of their epipolar lines must be P_m^i. That is,

$$\mathbf{x}_m^{i\mathrm{T}} \cdot \mathbf{l}_n + \mathbf{x}_m^{i\mathrm{T}} \cdot \mathbf{l}_t = 0. \tag{2}$$

Fig. 2 shows a visual illustration for this observation. It likewise holds true in the other two images. A similar property was also illustrated in [HZ03] on trifocal tensors.

With the growing of available images in the dataset, there would be more and more epipolar lines converging on a single point, if they all associate to a common correspondence. Using this metric, we hence are able to locate the unique matching point and quantitatively validate the plausibility to its other corresponding points. Here we name the consistent epipolar relationship between a feature and its corresponding points over a dataset as *epipolar consistency*. Normally, due to the diversity of viewpoints in image collections, the overlap of all epipolar lines is rarely happened, unless all images in the dataset are captured from the same perspective. This ensures the practicability of the proposed criteria.

3.2. Matching Objective

Intuitively, feature matching can then be formulated into a clustering problem in which feature points are intelligently grouped into a series of clusters. Each cluster represents a collection of corresponding points in different photos that satisfy the epipolar consistency criteria. Yet, due to inaccuracies in calculation, it is difficult to strictly restrict the constraint to be $\mathbf{x}^{\mathrm{T}} \cdot \mathbf{l} = 0$. Instead, we expect the global epipolar inconsistency computed over all clusters as small as possible. Given the above stated goals, we thus reach the following objective function for our matching problem, which needs to be minimized:

$$E_G = \sum_{k=1}^{K} \sum_{P_m^i \in \mathcal{C}_k} \sum_{P_n^j \in \mathcal{C}_k, i \neq j} \mathbf{x}_m^{i\mathrm{T}} \cdot \mathbf{l}_n, \tag{3}$$

where $\mathbf{l}_n = \mathcal{F}_{ij}\mathbf{x}_n^j$ ($\mathcal{F}_{ij} \neq \mathbf{0}$), K is the number of desired clusters, $\mathcal{C}_k$ denotes the feature set in the k-th cluster. The selection of K is important to our algorithm. We will discuss it in Sec. 3.3.

The objective function is intuitive. The term $f(P_m^i) = \sum_{P_n^j \in \mathcal{C}_k, i \neq j} \mathbf{x}_m^{i\mathrm{T}} \cdot \mathbf{l}_n$ tries to compute the epipolar inconsistency of feature P_m^i within cluster $\mathcal{C}_k$, and E_G expresses the global inconsistency residual over all clusters. Ideally, for correct feature correspondences, this residual should be zero ($E_G = 0$). In real data, it is often a small positive value because of the imperfect image registration. In comparison, feature clusters with spurious image matches will result in a large positive E_G. Thus, intuitively, the global minimum of Eq. 3 should correspond to a correct feature matching.

3.3. Optimization

We now get an objective function that captures our intuition about what makes a correspondence plausible or spurious. The rest of our concern is thus on how to minimize it. It is easy to note that Eq. 3 looks like a classical k-means model, which can be optimized by a predesigned value for the cluster number K and an iterative clustering algorithm (alternatively *assigning* and *updating*). Our optimization scheme derives from the normal k-means algorithm but with some modifications: the whole process is automatic (including the selection of parameter K) and requires two phases. The first phase involves the computation of adaptively cluster partitioning and feature clustering in descriptor space. In the next phase, initially clustered features are adjusted, according to their epipolar incon-

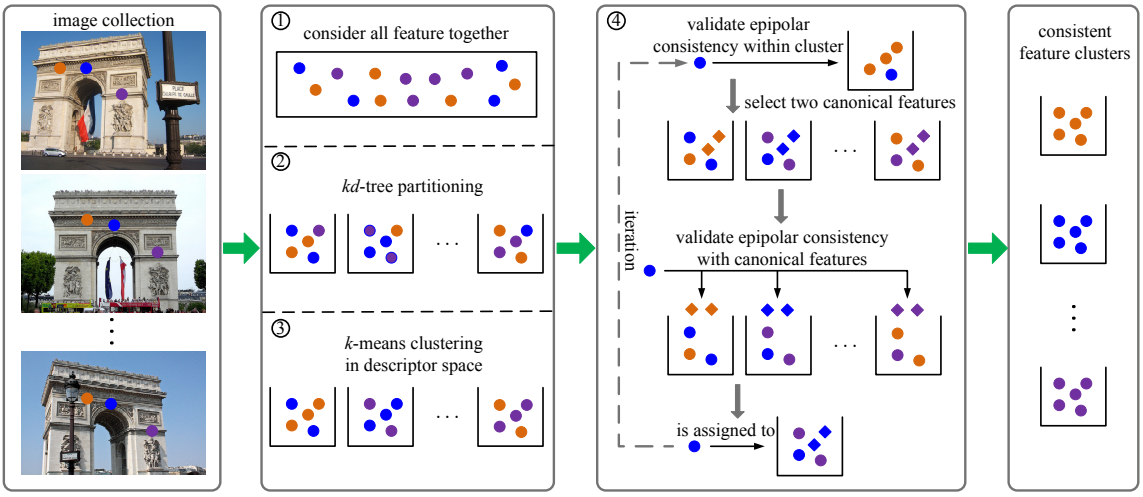


Figure 3: *The overview of our optimization procedure. Features are represented by disks in different colors, where the same color means correspondences. At first, all features are randomly distributed. With the kd-tree partitioning and k-means clustering, relevant features are gradually grouped together. However, these processes are insufficient to obtain putative correspondences. Hence next, geometric constraints are considered in the iteration, where each feature is arranged into the group that causes the minimal epipolar inconsistency.*

sistency within clusters, in order to minimize the global inconsistency. Fig. 3 shows a brief overview of the optimization pipeline.

Descriptor clustering phase A key problem for minimizing the objective function is how to know the K of cluster centers ahead. The number of clusters directly influences the matching accuracy. Yet, assigning the parameter manually is empirical and inaccurate. Excessive partitions usually lead to undesired separation of feature correspondences. On the contrary, inadequate cluster centers would cause many features incorrectly assembled. To address the issue, we employ a strategy by means of *kd*-tree partitioning [AGDL09, XL10] to adaptively form the initial centers. Our partitioning builds on the following intuition: the features originating from the same image must belong to different clusters, as they actually reflect the projection of two distinct 3D points.

To automatically return a list of cluster centers, we first place the entire features into a *kd*-tree. For each feature point, in addition to the spatial coordinate $\mathbf{x}$ and feature descriptor $\mathbf{d}$, we also preserve two additional variables in this step: h_{im} and h_{fe} which respectively represent the image it belongs to and its feature index in this image. In each cell, we first validate whether there are features originating from the same image, i.e., with the same h_{im}. If all features in this cell possessing different image indices, then we regard it as a terminal cell (a leaf node) and assign a center point being the mean vector of feature descriptors covered in this leaf domain. Otherwise, we split the cell into two child cells. The partitioning starts from the root cell of the *kd*-tree which contains all features in the dataset and recursively proceeds until all cells are leaf nodes.

By this mean, the *kd*-tree has adaptively produced K clusters, one cluster per leaf. The partitioning is conducted in descriptor space. It subdivides finely in the space in which distinct features from the same image distribute intensively, while subdividing coarsely in the region where the distribution is sparse. Each leaf node contains the grouped features and an associated cluster center with the same dimensionality.

Once the partitioning is achieved, we then perform a classical k-means clustering in descriptor space to improve the grouped result. However, there is a useful adaptation. In the conventional k-means algorithm, distances are computed from each cluster center to every element in the dataset. In contrast, we only compute distances from each cluster center to features within a limited distance scope, like [ASS*12]. This is based on the observation that features far from the center have less chance to be assembled into the group, so comparing distances between them is unnecessary. In order to confine the search space, we associate each cluster to its N_C nearest neighboring clusters by using a Gaussian search [AGDL09]. Afterward, each cluster center only needs to compare with features of its N_C neighboring clusters. This reduces the complexity of clustering process to be linear in the number of candidates N_C rather than the number of clusters K, and results in a significant speed advantage over conventional k-means method. In practice we found that when $N_C > 10$, the performance did not improve too much.

The modified k-means process in descriptor space is important to our optimization pipeline. First, it forms an initialization for the following geometry adjustment phase, which reduces the interference from distant outliers and enables our iteration converge more efficiently. Second, it provides a feasible way to compute the pairwise epipolar geometry, i.e., fundamental matrix, without known final matches.

Since our optimization relies on epipolar constraint, yet estimating epipolar geometry also requires feature matches. Hence it seems this is a chicken-and-egg problem. We overcome this problem by selecting a few high-confidence matches from the grouped clusters. If the distance L_1 from one feature to its cluster center is much less than the distance L_2 to the second closest center (here we

quantize the disparity in the form of $\frac{L1}{L2} < 0.6$), then we consider the feature is well represented by this cluster, and all well represented features in this cluster mutually are high-confidence matches. That is, in this case, the descriptor similarity is relatively reliable to indicate a match because of the distinctive clustering boundary (similar to the matching assumption of conventional method [Low04]). As an alternative, it is also feasible to select high-confidence matches using a small portion of top-scale features in each image, as illustrated in [Wu13, SSN15]. However, this method may lose some important correspondences between weakly overlapping images. Afterward, with a few high-confidence matches between an image pair (I_i, I_j) (at least 16 matches), we use the 8-point algorithm [HZ03] and RANSAC [FB81] to estimate the fundamental matrix. If the image pair does not have sufficient initial matches, we consider they are irrelevant. Then we will not compute geometric constrains between feature points in such pairs in the following phase.

Geometry adjustment phase In this stage, we have to reanalyze the allocation of each feature and arrange it into the cluster which causes the minimal increasing of epipolar inconsistency. Once again, we leverage the k-means algorithm but with two important distinctions:

- A distance measure evaluates the epipolar inconsistency instead of descriptor similarity. This requires computing the residual as defined by $u = \mathbf{x}^{\mathrm{T}} \cdot \mathbf{l}$.
- The cluster center is no longer a mean vector. Instead, two canonical features in this cluster are adopted.

Hence, in this phase, feature descriptors are no longer required, instead, the coordinates of features in image plane are leveraged. That means much CPU memory can be released.

Our iterative procedure begins with a *pre-processing step* where the K initial cluster centers produced in the former phase are replaced. For each feature P in a cluster $\mathcal{C}$, we first calculate its epipolar inconsistency $e = f(P)$ with other features that also belong to the cluster. Then we select the top two features with lowest residual errors e serving as agents of current cluster. Our implicit assumption is that not all features in the cluster are irrelevant; many feature points are correctly grouped after initial clustering in the first phase, as we illustrated in Fig. 3. Thus the feature with less residual error e is more reliable to reflect the geometric property of this cluster. In this phase, the mean vectors of cluster centers are not required. Each cluster is collaboratively represented by the two canonical features. This is done to tightly constrain the epipolar consistency using two epipolar lines. If a cluster has less than three features, then these features are directly used as agents. For later usage, we also calculate the total epipolar inconsistency of each cluster, i.e., $E = \sum_{P \in \mathcal{C}} f(P)$. Moreover, there is one thing deserving the notice. Some feature pairs in a cluster may share none fundamental matrices ($\mathcal{F} = \mathbf{0}$). In such case, we manually set a large constant for the inconsistency residual u ($u = 5$) between these pairs.

Next, in the *assignment step*, we have to associate each feature P with the "nearest" cluster. Here the nearest is measured by the epipolar consistency. In order to minimize Eq. 3, we expect the feature to be grouped into a cluster which causes the lowest increasing of epipolar inconsistency. So for each feature P, we compute its epipolar inconsistency $e = f(P)$ with the two feature agents of each cluster (not the entire features in the cluster). If P achieves the minimal residual error e_k against cluster $\mathcal{C}_k$ and $P \notin \mathcal{C}_k$, then we associate this feature to this cluster; otherwise, proceed for another feature. Like the process in the last phase, we confine the comparison scope of each feature, which is determined by the N_C neighborhood of its cluster. This is the key to speed up our iterative procedure because limiting the size of comparison region significantly reduces the number of calculations.

Once each feature has been associated to a proper cluster, an *update step* is required to readjust the agents of each cluster. We again calculate the epipolar inconsistency e of each feature in the cluster, and choose the top two as agents. Accordingly, the total epipolar inconsistency E' of this cluster is also updated. Then the ℓ_1 norm is utilized to compute a difference $\varepsilon = \sum_{\mathcal{C}_i} |E'_i - E_i|$, for each cluster $\mathcal{C}_i$, between current cluster state and previous cluster state. The assignment and update steps repeat iteratively until the difference converges: ε is below a given threshold. However, in experiments, we have found that $N_I = 8$ iterations suffice for most datasets due to the initialization process in the first phase, and we report all results in this paper using this setting.

Finally, we need a *post-processing step* to filter unreliable clusters. Because the cluster with less than three features is insufficient to decide correspondences, hence we neglect the matching results in such clusters. We also strictly constrain that the mean epipolar inconsistency of each cluster (corresponding to each residual u) should be below a given threshold $N_D = 5$; otherwise, this cluster is discarded. Additionally, we have to deal with the case that multiple features within a cluster may originate from the same image. We prune such redundant features according to their ascending order of residual errors e and only remain the minimal one. The entire algorithm of this phase is summarized in Algorithm 1.

Algorithm 1 Geometrically-based feature correspondence adjustment

```
1:  /* Pre-process */
2:  for each cluster C_i do
3:      for each feature in C_i do
4:          compute epipolar inconsistency e = f(P) with other features also in the cluster
5:      end for
6:      select two features with the lowest e
7:      compute the total inconsistency E of the cluster
8:  end for
9:  repeat
10:     /* Assignment */
11:     for each cluster C_i do
12:         for each feature in C_i do
13:             compute e = f(P) with the canonical features of other clusters within a search scope
14:             move the feature into the cluster with lowest e
15:         end for
16:     end for
17:     /* Update */
18:     the same to the process in /* Pre-process */ (line 2 to 8)
19: until ε ≤ threshold
```

Table 1: *Performance statistics of three matching methods on three image pairs sampled from different collections.*

Image Name	#Features		[MA10]		[CLW*14]		Our Method
	image1	image2	#matched	#accepted	#matched	#accepted	#matched
Office Table	3974	3095	90	56	120	84	392
Red Wall	4355	2633	51	44	88	76	389
Notre Dame	5760	22645	162	126	183	152	415

4. Experiments

In this section, we evaluate the performance of our matching algorithm on a variety of datasets. These images range from small-scale laboratory scenes to large-scale landmark architectures. Fig. 4 shows a series of image pairs samples from these datasets and Table 1 and Table 3 list a detail summary. In addition, as our algorithm is constructed for rigid feature-wise correspondence, so the collections mainly correspond to static objects with seldom non-rigid deformations.

In order to evaluate the effectiveness of our proposed method, it is compared to two state-of-the-art matching approaches [MA10] (*kd*-tree-based) and [CLW*14] (hashing-based). All the baseline algorithms adopt SIFT [Low04] as the experimental feature. In Table 1, we report the performance of our algorithm and comparisons with these methods. The statistics are based on several image pairs, which are randomly sampled from the datasets. We list the number of detected features in both images and the matches before (#matched) and after geometric verification (#accepted). Because our algorithm has integrated geometric constraints into the matching pipeline, so it has only one term (#matched). For classic *kd*-tree-based matching, we adopt the ANN library [MA10] for nearest neighbor search. As for hashing-based matching, we test the code provided by [CLW*14] and leave its parameters unchanged. Additionally, as we previously analyzed, these methods both require the ratio-test strategy [Low04] for matching measuring. In our experiments, we set the ratio threshold to be 0.6, which is a general value for recent matching systems. All comparisons are carried out on a single personal PC running Ubuntu 14.04 operating system with Xeon E3 3.3GHz CPU and 16GB memory space. From the table, it can be seen that our method, as compared to the other algorithms, is capable of retrieving much more matches from all these scenes. This is because our method is going to intelligently adjust inconsistent features rather than remove them directly. While for higher distance ratio (e.g., 0.8), [MA10] [CLW*14] could acquire more true positive matches, more false positive would also remain.

Fig. 4 shows a visual comparison of these pictures. The images we exhibit here are in different styles. Dataset "Office Table" shows an indoor scene with large viewpoint rotations. Dataset "Red Wall" is an outdoor scene with shift changes. "Notre Dame" is an unstructured dataset harvested from Internet by [SSS06]. Additionally, we do not choose the image pairs which are full of visual linkages, as it is hard to see other details. The first column of this figure expresses the feature connections (green lines) established by [MA10] and a geometrically verified version is followed. The third column is our outputs. Because [CLW*14] is constructed mainly for acceleration and produces similar results to [MA10], so we only visualize the matching results of [MA10] for easy illumination. As shown in the figure, our method achieves significantly denser feature connections as compared to [MA10], while having seldom outliers.

Next, we evaluate the accuracy of our method. However, due to the absence of ground truth, it is difficult to indicate which correspondences are truly plausible. Noisy still remains, even with the assistance of human interaction. Fortunately, with a set of calculated matches, it is feasible to compute the Homography matrix $\mathcal{H}_{ij}$ relating two given images I_i and I_j. Then for each SIFT keypoint in I_i, its expected position in I_j can be roughly determined via this Homography map $\mathcal{H}_{ij}\mathbf{x}_m^i$. According to the heuristic observation, we thus subjectively define that if (P_m^i, P_n^j) is a ground truth match, it should satisfy the criteria: $||\mathcal{H}_{ij}\mathbf{x}_m^i - \mathbf{x}_n^j|| < d_{shift}$. That is, the coordinate distance between P_n^j and the expected position of P_m^i in I_j should be less than a certain threshold. In our experiment, we set the distance threshold to 5 pixels. We present a table (Table 2) reporting the precision and recall achieved by the tested algorithms. The records are estimated on the three image pairs of Fig. 4 and imposed the geometric verification for [MA10] and [CLW*14]. It is easy to note that while precision rates are relatively satisfactory, recall rats are overall low for the two methods due to the preemptive rejection. In contrast, our algorithm maintains higher records in both precision and recall in the test.

Table 2: *Precision and recall statistics of three matching methods.*

Method	Precision	Recall
kd-tree-based method [MA10]	94.0	14.7
hashing-based method [CLW*14]	89.8	17.6
our geometrically-based method	98.5	53.4

To further evaluate our results, we also compare the accuracy of each approach by calculating the epipolar inconsistency as proposed in Eq. 3. As we have illustrated, lower residual error of E_G correspond to more consistent feature matching. Fig. 5 shows the mean residual curves of three matching methods with respect to different dataset sizes. In the condition of several images, all methods retain a relatively low residual value. However, with the growing of available images, their slopes become significantly different. Our method analyzes correspondence consistency over the whole dataset, so it has a lower inconsistency residual as compared to the other methods which are conducted in pairwise manner. This also highlights two things: 1) the performance of feature matching can be improved within photo collections; and 2) the measure E_G can be used as an alternative tool to evaluate the matching accuracy. Due to the noisy in calculations, our algorithm is inevitable to bring some mismatches. However, the proposed metric tries to ensure the inconsistency of our method as small as possible .

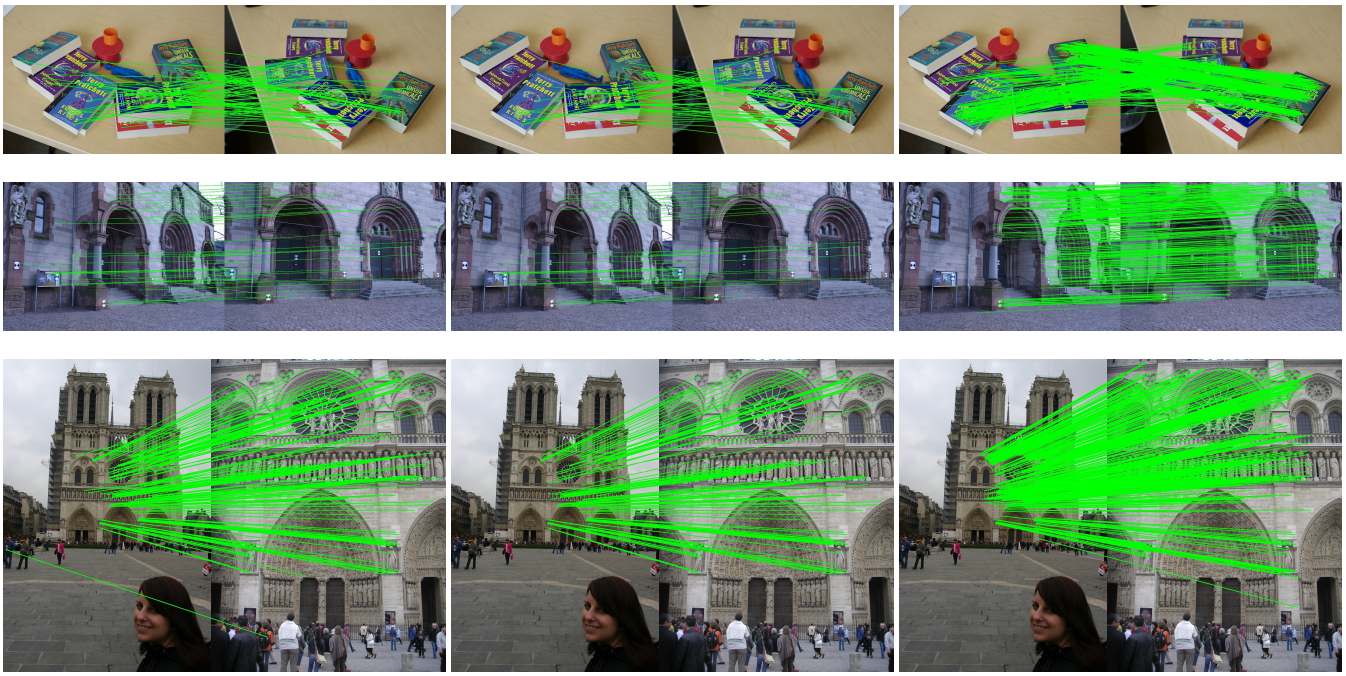


Figure 4: *The matching result comparisons on three datasets. The first column shows the feature connections (green lines) established by [MA10]. The second column are the results of [MA10] after geometric verification. The third column shows our outputs.*

We next relate our results to image-based modeling. Recent work in this direction aim to reconstruct denser point clouds. Image matching is a central element to this field, as each feature potentially corresponds to a spatial point and more matches usually lead to denser 3D models. Hence, the number of points generated by image-based modeling algorithms, such as structure from motion (SfM) [SSS06], to some extent, reflects the performance of feature correspondence as well. Table 3 shows the statistics of the three matching methods used in conventional SfM pipeline for match graph construction. It lists the number of input images and the number of 3D point clouds finally outputted by SfM. It is visible that, with regard to different photo collections, our algorithm is always capable of producing significantly denser results. For more comparisons, we will visualize the 3D reconstruction results in Sec. 5.

In terms of speed, Table 3 also shows the running time of the tested algorithms over image collections. Our algorithm, in contrast to [MA10] [CLW*14], is not constructed for acceleration in mind. Yet, as shown in the table, it performs reasonably well even in a single CPU core. It intelligently finds a subset of candidates for each feature to match and do not require constructing the tree structure repetitively. The comparisons are examined without any distributed computation or GPU acceleration. For fairness and simplicity, we have excluded the computational time of SIFT extraction and pairwise $\mathcal{F}$-matrix estimation for all methods. In the case of large-scale photo collections, image retrieval technique [NS06] is recommended to predict overlapping images.

There are only three easy setting parameters (N_C, N_I and N_D) used in the proposed frameworkd. N_C decides the quantity of related neighbors for each cluster to compare, while N_I determining the number of iterations required in the geometrically-based adjustment phase. In our experiment, we find that $N_C = 10$ and $N_I = 8$ are sufficient for the practical usage. N_D controls the threshold of cluster inconsistency. We set it to 5 for precision consideration. For some implicit parameters, such as the parameters in RANSAC and Gaussian *kd*-tree, we use them in default setting as appeared in previous work or their codes.

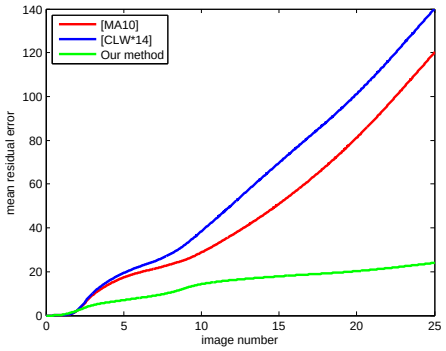


Figure 5: *The mean residual errors of the three test methods according to different dataset size.*

5. Application

Due to the simplicity of image acquisition and the growing ubiquity of handheld cameras, image-based modeling techniques have become a popular possibility for 3D reconstruction. Among them,

Table 3: *3D reconstruction performance of three matching methods.*

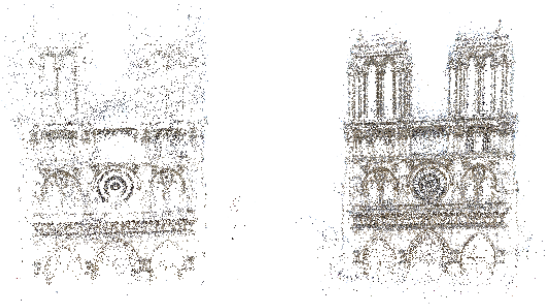
Datasets	#Images	#Point Clouds			Time		
		[MA10]	[CLW*14]	Ours	[MA10]	[CLW*14]	Ours
Arc de Triomphe	217	42 K	39 K	57 K	84 m	21 m	47 m
Museum	27	33 K	34 K	48 K	7 m	2 m	5 m
Notre Dame	18	13 K	12 K	34 K	4 m	1 m	3 m

the structure from motion (SfM) remains the most famous one. It relies heavily on accurate feature matching to recover the 3D scene geometry by triangulating the registered images.

In this section we show 3D reconstruction results from photo collections [CDSHD13, SSS06] based on our iterative matching method. We begin with SIFT keypoint detection on each given image. Then different feature matching methods are conducted. The output of our algorithm can be directly used in the following incremental SfM process, as each feature cluster corresponds to a *track* which reflects the same 3D point mapping into multiple 2D images. We use Bundler [SSS06] to compute point clouds.

(a) Two input image collections

(b) Museum: 27 images



(c) Notre Dame: 18 images

Figure 6: *3D reconstruction results on two small-scale datasets: "Museum" and "Notre Dame". (a) shows two representative images of these collections. The left column in (b)(c) are reconstruction results using [MA10] for match graph construction. The right column shows the results based on our framework.*

Table 3 lists the number of recovered 3D points using the three approaches and corresponding time. The 3D reconstruction results are illustrated in Fig. 6. In order to recover desirable 3D structures, SfM algorithms, based on traditional matching schemes, usually require large-scale image collections. However, some regions are still challenging to recover even with adequate input pictures, for example, the vegetation in "Museum". The authors [CDSHD13] suffer from the absence of point clouds in plants, as shown in the left image of Fig. 6(b). This is due to the analogy of feature points in these regions and most of them are rejected in matching by ratio-test. In contrast, our method can reconstruct very detailed 3D models in the vegetation and symmetric patterns even from very small-scale image collections, as shown in the right side of Fig. 6.

Figure 7: *A failure case of our algorithm. Due to overwhelming repetitions in the image content, most correspondences are spurious, although their epipolar inconsistency is low. The left shows the matching result using [MA10]; the right side is ours.*

Limitations Although we have tested the performance of our algorithm on a range of diverse datasets and 3D reconstruction application, it also suffers from the following limitations. First, our matching scheme depends on epipolar relations to guide feature clustering. Thus, our method mainly constructs feature correspondences for static scenes. For images with non-rigid deformation or when the epipolar geometry cannot be reliably recovered, it will not perform geometrically adjustment to these image pairs. In such case, our matching method would degenerate to a traditional matching scheme, where the descriptor clustering phase will play an important role in our framework. Second, in order to achieve more reliable matching results, a set of related images are desired. Given only two images, our criteria is insufficient to provide quantitative judgment. In such case, the ratio-test strategy is recommended. However, with the simplicity of image acquisition, such requirement would be generally satisfied. Third, repetitions may contribute to spurious correspondences. As shown in Fig 7, while our method can retrieve more geometrically consistent matches from these duplicate structures, it is insufficient to distinguish these ambiguities, as they also satisfy our criteria on epipolar consistency.

6. Conclusion and Future Work

In this paper, we have presented a novel algorithm for reliable feature correspondence based on the analysis of epipolar consistency. We reason that, with a set of images, the epipolar geometry can provide quantitative correspondence depict. With this metric, we creatively turn feature matching into an optimization problem and minimize it in a linear iterative manner. Experiments show that the proposed algorithm achieves more precise and complete visual correspondences as compared to the other tested approaches. We further demonstrate its usefulness in 3D reconstruction. Based on the constructed match graph, we get significantly denser point clouds and detailed 3D models.

Although in this work we mainly focus on the problem of correspondence quantification for rigid images, we also consider that the geometric relationship (in some other forms instead of epipolar geometry) upon image collections might be advisable for non-rigid scenes, like [LYP*14], or the problem of pixel-based correspondence [HSGL11] as well. In the future, we plan to investigate and extend our model to address these challenges.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and insightful suggestions. This work was partly supported by the National Basic Research Program of China (No. 2012CB725303), the NSFC (No.61472288, No. 61672390), NCET (NCET-13-0441), the Fundamental Research Funds for the Central Universities (2042015kf0181), and the State Key Lab of Software Engineering (SKLSE-2015-A-05).

References

[AECO15] AVERBUCH-ELOR H., COHEN-OR D.: Ringit: Ring-ordering casual photos of a temporal event. *ACM TOG 34*, 3 (2015), 33. 1

[AGDL09] ADAMS A., GELFAND N., DOLSON J., LEVOY M.: Gaussian kd-trees for fast high-dimensional filtering. *ACM TOG (SIGGRAPH) 28*, 3 (2009), 21–32. 5

[ASS*09] AGARWAL S., SNAVELY N., SIMON I., SEITZ S. M., SZELISKI R.: Building rome in a day. In *ICCV* (2009), pp. 72–79. 2

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SUSSTRUNK S.: Slic superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI 34*, 11 (2012), 2274–2282. 5

[CDSHD13] CHAURASIA G., DUCHENE S., SORKINE-HORNUNG O., DRETTAKIS G.: Depth synthesis and local warps for plausible image-based navigation. *ACM TOG 32*, 3 (2013), 30. 1, 3, 9

[CLW*14] CHENG J., LENG C., WU J., CUI H., LU H., ET AL.: Fast and accurate image matching with cascade hashing for 3d reconstruction. In *CVPR* (2014), pp. 1–8. 1, 2, 7, 8, 9

[FB81] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 6 (1981), 381–395. 2, 6

[FFGG*10] FRAHM J.-M., FITE-GEORGEL P., GALLUP D., JOHNSON T., RAGURAM R., WU C., JEN Y.-H., DUNN E., CLIPP B., LAZEBNIK S., ET AL.: Building rome on a cloudless day. In *ECCV*. 2010, pp. 368–381. 2

[FP10] FURUKAWA Y., PONCE J.: Accurate, dense, and robust multiview stereopsis. *IEEE TPAMI 32*, 8 (2010), 1362–1376. 3

[GSC*07] GOESELE M., SNAVELY N., CURLESS B., HOPPE H., SEITZ S. M.: Multi-view stereo for community photo collections. In *ICCV* (2007), pp. 1–8. 3

[HS14] HAVLENA M., SCHINDLER K.: Vocmatch: Efficient multiview correspondence for structure from motion. In *ECCV*. 2014, pp. 46–60. 3

[HSGL11] HACOHEN Y., SHECHTMAN E., GOLDMAN D. B., LISCHINSKI D.: Non-rigid dense correspondence with applications for image enhancement. *ACM TOG (SIGGRAPH) 30*, 4 (2011), 70. 10

[HZ03] HARTLEY R., ZISSERMAN A.: *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2, 3, 4, 6

[IBP15] ICHIM A. E., BOUAZIZ S., PAULY M.: Dynamic 3d avatar creation from hand-held video input. *ACM TOG (SIGGRAPH) 34*, 4 (2015), 45. 1

[KTT*12] KIM K. I., TOMPKIN J., THEOBALD M., KAUTZ J., THEOBALT C.: Match graph construction for large image databases. In *ECCV*. 2012, pp. 272–285. 3

[Low04] LOWE D. G.: Distinctive image features from scale-invariant keypoints. *IJCV 60*, 2 (2004), 91–110. 1, 2, 3, 6, 7

[LSG12] LOU Y., SNAVELY N., GEHRKE J.: Matchminer: Efficient spanning structure mining in large image collections. In *ECCV*. 2012, pp. 45–58. 3

[LYP*14] LIPMAN Y., YAGEV S., PORANNE R., JACOBS D. W., BASRI R.: Feature matching with bounded distortion. *ACM TOG 33*, 3 (2014), 26. 10

[MA10] MOUNT D. M., ARYA S.: Ann: A library for approximate nearest neighbor searching. *URL http://www. cs. umd. edu/~ mount/ANN/. Website 10* (2010). 2, 7, 8, 9

[ML14] MUJA M., LOWE D. G.: Scalable nearest neighbor algorithms for high dimensional data. *IEEE TPAMI 36*, 11 (2014), 2227–2240. 2

[NS06] NISTER D., STEWENIUS H.: Scalable recognition with a vocabulary tree. In *CVPR* (2006), vol. 2, pp. 2161–2168. 2, 8

[OT01] OLIVA A., TORRALBA A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV 42*, 3 (2001), 145–175. 2

[SBBF12] STRECHA C., BRONSTEIN A. M., BRONSTEIN M. M., FUA P.: Ldahash: Improved matching with smaller descriptors. *IEEE TPAMI 34*, 1 (2012), 66–78. 1, 2

[SBF15] SCHONBERGER J. L., BERG A. C., FRAHM J.-M.: Paige: pairwise image geometry encoding for improved efficiency in structure-from-motion. In *CVPR* (2015), pp. 1009–1018. 3

[SCF*14] SHAN Q., CURLESS B., FURUKAWA Y., HERNANDEZ C., SEITZ S. M.: Occluding contours for multi-view stereo. In *CVPR* (2014), pp. 4002–4009. 3

[SSN15] SHAH R., SRIVASTAVA V., NARAYANAN P.: Geometry-aware feature matching for structure from motion applications. In *WACV* (2015), pp. 278–285. 2, 6

[SSS06] SNAVELY N., SEITZ S. M., SZELISKI R.: Photo tourism: exploring photo collections in 3d. *ACM TOG (SIGGRAPH) 25*, 3 (2006), 835–846. 7, 8, 9

[TL09] TURCOT P., LOWE D.: Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshop on Large Amounts of Visual Data* (2009), vol. 4. 3

[Wu13] WU C.: Towards linear-time incremental structure from motion. In *3DV* (2013), pp. 127–134. 3, 6

[XL10] XIAO C., LIU M.: Efficient mean-shift clustering using gaussian kd-tree. *Computer Graphics Forum 29*, 7 (2010), 2065–2073. 5

[YXX14] YAN Q., XU Z., XIAO C.: Fast feature-oriented visual connection for large image collections. *Computer Graphics Forum 33*, 7 (2014), 339–348. 1, 3

[ZK06] ZHANG W., KOSECKA J.: Generalized ransac framework for relaxed correspondence problems. In *3DPVT* (2006), pp. 854–860. 2