Graph-Based Consistent Matching for Structure-from-Motion

Tianwei Shen, Siyu Zhu, Tian Fang(✉), Runze Zhang, and Long Quan

Department of Computer Science and Engineering,
Hong Kong University of Science and Technology, Hong Kong, China
{tshenaa,szhu,tianft,rzhangaj,quan}@cse.ust.hk

Abstract. Pairwise image matching of unordered image collections greatly affects the efficiency and accuracy of Structure-from-Motion (SfM). Insufficient match pairs may result in disconnected structures or incomplete components, while costly redundant pairs containing erroneous ones may lead to folded and superimposed structures. This paper presents a graph-based image matching method that tackles the issues of completeness, efficiency and consistency in a unified framework. Our approach starts by chaining all but singleton images using a visual-similarity-based minimum spanning tree. Then the minimum spanning tree is incrementally expanded to form locally consistent strong triplets. Finally, a global community-based graph algorithm is introduced to strengthen the global consistency by reinforcing potentially large connected components. We demonstrate the superior performance of our method in terms of accuracy and efficiency on both benchmark and Internet datasets. Our method also performs remarkably well on the challenging datasets of highly ambiguous and duplicated scenes.

Keywords: Structure-from-Motion · Image matching · Loop consistency

1 Introduction

Image matching is a computationally expensive step in 3D reconstruction, especially for large-scale unordered image datasets. Due to the large number of high-dimensional feature descriptors in an image, the naive quadratic matching scheme imposes a heavy computational burden on large-scale high-resolution 3D reconstruction [35]. Tremendous progress has been achieved either on reducing the cost of feature matching [19] or image indexing techniques [15,22] to pre-compute a subset of match candidates. Modern large-scale Structure-from-Motion (SfM) systems [1,12] usually use vocabulary tree [22] to choose the visually similar match pairs, which decreases the complexity of pairwise image matching from $O(n^2)$ to $O(kn)$ with respect to the number of images.

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46487-9_9) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part III, LNCS 9907, pp. 139–155, 2016.
DOI: 10.1007/978-3-319-46487-9_9

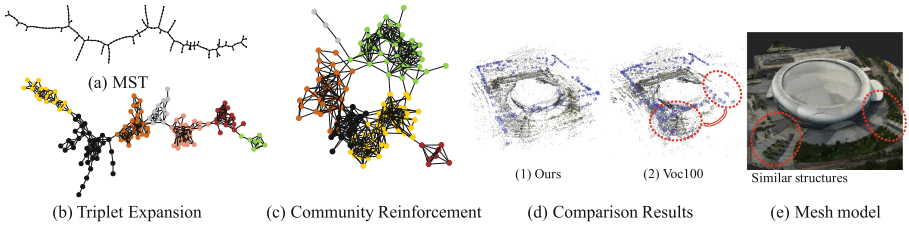

Fig. 1. Pipeline of the matching framework. (a) The minimum spanning tree (MST); (b) The triplet expansion process; (c) The final match graph after component merging, with different colors representing different communities; (d) Comparison of the proposed method with image retrieval techniques (see Sect. 5 for details); (e) The reference mesh model. (Color figure online)

However, two problems remain to be solved. One major drawback of the image indexing techniques is that the number of retrieved items k for a query image is hard to determine. Many previous works have adopted an empirical similarity threshold or a fixed retrieved number, which ignores the global connectivity of the image collection. Sometimes post-processing steps, such as query expansion [1,4], are also needed to prevent the missing of true positive matches. As a result, the actual number of matches is still large to ensure the completeness and accuracy of 3D reconstruction.

On the other hand, large-scale image datasets often contain ambiguous scenes due to symmetric and repetitive textured patterns. These repetitive yet distinct patterns are not only visually similar, but can also pass the two-view geometric verification and form erroneous epipolar geometry. The false match pairs can collapse the SfM results and lead to folded or superimposed structures. Due to the existence of ambiguous patterns, adding more yet potentially incorrect pairwise matches may severely hurts the performance of SfM. Therefore, a sufficient and consistent subset of matches is superior to a redundant matching set that may contain false matches, which somewhat contradicts with the principle of mining as much connectivity as possible. Moreover, since it is difficult to filter out the wrong epipolar geometry and relative poses during camera pose estimation, a consistent match graph is crucial to the success of 3D reconstruction.

In this paper, we propose a matching algorithm that efficiently generates a sparse match graph spanning the whole image dataset, while simultaneously filters out inconsistent matches which pass the two-view geometric verification. Our method jointly discovers the connectivity pattern of the scene and achieves a good trade-off between computational efficiency and sufficient image connectivity in a consistent manner. The consistency of the matching set is guaranteed by enforcing *loop consistency* [33] both locally and globally along the successive steps.

As our main contribution, we propose the first unified framework, to the best of our knowledge, that jointly conducts efficient pairwise image matching and solves the SfM ambiguity problem. This novel matching algorithm significantly accelerates the matching process without sacrificing the accuracy of SfM models. Moreover, it is capable of handling extremely ambiguous scenes with loop consistency checking.

2 Related Work

2.1 Image Retrieval Techniques for 3D Reconstruction

To avoid the costly exhaustive match, image retrieval has been extensively employed as a pre-processing step for large-scale SfM. Vocabulary tree [22] is the most widely used technique to rank the database images given a query image. Several later methods [3,13,23] incorporate geometric cues to improve the retrieval performance for 3D reconstruction. Query expansion [4] is a popular technique to increase the recall of retrieval results. Lou et al. [16] employ relevance feedback and entropy minimization to explore the connectivity between images as quickly as possible. As an example of exhaustive matching techniques, preemptive matching proposed by Wu [31] argues that features at a larger pyramid scale tend to be more stable. Therefore, by matching a small subset of local descriptors in an image, we can decide whether to continue the full putative feature matching. Recently, Schönberger et al. [25] compare these matching techniques and propose a learning-based method to predict whether a pair of images have overlapping regions. Zhou et al. [34] propose a multi-image matching algorithm based on loop consistency [33] and low-rank modeling. Most of these works have focused on improving the performance of image retrieval in terms of precision and recall. Little attention has been paid on the actual effect of increased recall on the final results of SfM. As we have demonstrated in Sect. 5, in large-scale urban scenes, more matches do not necessarily guarantee a better reconstruction.

2.2 Optimization of the Viewing Graph

Another line of works is to reduce the geometric computation by optimizing the match graph (also known as the viewing graph). Snavely et al. [26] propose a efficient SfM pipeline by computing a skeletal image set that approximates the accuracy of the full set in terms of covariance. Havlena et al. [9] also relies on image indexing techniques and selects a minimal connected dominating set of the image collection. The images in the reduced set form atomic 3D model and are incrementally merged into a connected model, similar to [10]. Recently, Sweeney et al. [28] propose a viewing graph optimization method that achieves excellent accuracy while remaining efficient for the SfM process. It is similar to our work in that they enforce loop consistency in the viewing graph, while we check loop consistency in the matching process.

All of the above methods accelerate the reconstruction process but either start from a time-consuming full match graph, or use vocabulary tree to initialize the match graph whose candidate number is difficult to choose. Instead, we concentrate on the optimization of the match graph and come up with an efficient match graph construction method without altering either incremental or global SfM pipelines.

2.3 Ambiguous Structures

Identification and removal of erroneous epipolar geometry is a recent research focus for SfM. Zach et al. [32] use the supplementary information in the third view which does not exist in the two-view relation to infer the correctness of two-view geometry. Their subsequent work [33] exploits loop consistency to infer incorrect geometric transformations, which forms the basis of our work. However, this formulation has strong assumptions on the statistical independence of erroneous matches, thus it will fail on highly ambiguous scenes where similar but distinct patterns become norms instead of outliers. Roberts et al. [24] sample a minimal configuration (a spanning tree of the match graph) to infer data associations based on the missing correspondence cue [32] and the timestamp cue. This is based on the assumption that the time and sequence information are correlated, which is generally not satisfied in unordered datasets. Jiang et al. [14] also sample a spanning tree from a relatively complete pairwise match graph, and iteratively replace problematic edges in a greedy way. Wilson et al. [29] analyse disambiguation on a visibility graph encoding relations of cameras and points. This method identifies bad tracks based on the observation that bad tracks in urban scenes connect two or more clusters of useful tracks. Heinly et al. [11] use a post-processing step which first splits the camera graph and then leverages conflicting observations to identify duplicated structures. The set of camera subgraphs that are free from conflict are then merged into a correct reconstruction.

Instead, the proposed method solves the ambiguity problem jointly with the efficient construction of a robust and consistent match graph, without modifying the SfM pipeline. We argue that the origin of ambiguity comes from a faulty matching process, thus the early detection of erroneous edges in the match graph would be beneficial to the later geometric computation. This generic matching framework is orthogonal to and can be combined with the other disambiguation methods.

3 Problem Formulation

In this section, we introduce a couple of basic building blocks of the graph-based matching algorithm and a set of criteria that needs to be satisfied. The inputs of the method are a set of images $\mathcal{I} = \{I_i\}$ and their corresponding feature points. The matching method is based on the analysis of the underlying graph encoding pairwise matches and epipolar geometry. We denote the undirected match graph as $G = (\mathcal{V}, \mathcal{E})$ where each vertex $v_i \in \mathcal{V}$ corresponds to an image $I_i \in \mathcal{I}$. Two vertices v_i and v_j are connected by an edge $e_{ij} \in \mathcal{E}$ if their corresponding images have more than S_I inliers after epipolar geometric verification. Each edge e_{ij} is associated with the epipolar geometry and relative motion between an image pair computed using five-point algorithm [21]. The initial edge set $\mathcal{E}$ is empty and we aim to incrementally build the match graph. We ensure that the running time of all the graph algorithms used in the method is an order of magnitude lower than the image matching operation.

Let T_{ij} be an abstract geometric relation associated with the edge e_{ij}, e.g. T_{ij} can be the relative rotation R_{ij} computed from feature correspondences. We further require that this geometric relation can be chained, denoted by $\circ$, and satisfies $T_{ij} \circ T_{ji} = \mathbb{I}(\forall i, j)$ where $\mathbb{I}$ denotes the identity map. Then in a consistent yet noisy setting, the discrepancy should be small between the identity map and the chained transformation on a closed loop. We first consider the minimum configuration of closed loops and give the following definition for the weakly consistent match graph.

Definition 1 *(Weak Consistency). A match graph $G = (\mathcal{V}, \mathcal{E})$ is **weakly $(\epsilon, \mathcal{E})$-consistent**, if the pairwise geometric relations T_{ij}, T_{jk}, T_{ki} of any 3-length loop (i, j, k) with respect to the edge set $\mathcal{E}$ satisfy the following loop consistency constraint*

$$\begin{gathered} d(T_{ij} \circ T_{jk} \circ T_{ki}, \mathbb{I}) \leq \epsilon \\ \forall (i, j, k), e_{ij} \in \mathcal{E}, e_{jk} \in \mathcal{E}, e_{ki} \in \mathcal{E} \end{gathered} \tag{1}$$

where the distance function $d(\widetilde{T}, \mathbb{I})$ measures the discrepancy between the chained motion $\widetilde{T}$ and the identity map $\mathbb{I}$. The above definition does not capture all essences of a consistent match graph because some erroneous matches may only manifest themselves in longer loops. Therefore, we refine this notion by defining *strong consistency*:

Definition 2 *(Strong Consistency). A match graph $G = (\mathcal{V}, \mathcal{E})$ is **strongly $(\epsilon, \mathcal{E})$-consistent** if for any loop $(n_0, n_1, \ldots, n_{m-1})$ of length m with respect to the edge set $\mathcal{E}$, the following condition holds*

$$\begin{gathered} d((\prod_{i=0}^{m-2} T_{n_i n_{i+1}}) \circ T_{n_{m-1} n_0}, \mathbb{I}) \leq \epsilon \\ \forall (n_0, n_1, \ldots, n_{m-1}), e_{n_0 n_1} \in \mathcal{E}, e_{n_1 n_2} \in \mathcal{E}, \ldots, e_{n_{m-1} n_0} \in \mathcal{E}, \end{gathered} \tag{2}$$

where $\prod$ denotes the chaining of a set of geometric transformations with $\circ$ operator.

To find a consistent match graph, we need to balance the following three performance criteria:

(1) *Completeness.* The match graph should span as many as images to guarantee the completeness of 3D models. This criterion corresponds to minimizing the number of connected components in G.
(2) *Efficiency.* The time complexity of the match graph construction should depend on the underlying connectivity pattern of the image collection.
(3) *Consistency.* The edges should be both robust meaning that each of them contains a large number of inlier feature matches, and consistent measured by ϵ (the smaller the better) and $|\mathcal{E}|$ (the larger the better) in Definitions 1 and 2. This criterion may contradict with *efficiency*, hence we need to find a good trade-off between them.

Algorithm 1. Online Minimum Spanning Tree

Input: The match graph $G = (\mathcal{V}, \mathcal{E})$ with empty edge set $\mathcal{E} = \emptyset$, the singleton rejection threshold S_R, the match inlier threshold $S_I^\star$, an array recording the failure time ft[] $\leftarrow 0$

Output: A minimum spanning tree or forest of G

```
for v ∈ V do
    MAKE-SET(v)
end for
for e_ij ordered by w(e_ij), increasing do
    if UNION-FIND(i) ≠ UNION-FIND(j) & ft[i] < S_R & ft[j] < S_R then
        Verify whether (i, j) is a true match using a strict inlier threshold S_I^⋆
        if (i, j) matches then
            UNION(i, j)
        else
            ft[i]++; ft[j]++;
        end if
    end if
end for
```

4 Graph-Based Consistent Matching

The proposed method can be decomposed into three steps illustrated in Fig. 1: (a) match graph initialization, (b) graph expansion by strong triplets and (c) community-based graph reinforcement. The purpose of *match graph initialization* is to minimize the number of connected components and discard singleton images in the match graph (*completeness*). The *expansion* and *reinforcement* steps are successively applied to efficiently explore the scene structure (*efficiency*), while weak and strong consistency are iteratively verified along the process (*consistency*). The three steps are detailed in the following sections.

4.1 Match Graph Initialization

Criterion (1) can be separately accomplished by quickly chaining the views in an image collection. To achieve this goal, we try to find a *minimum spanning tree* of the match graph. This seems impossible since we do not have the connectivity information before computing feature correspondences and epipolar geometry. However, similarity scores and rank information given by the vocabulary tree parameterizes a *priori* match graph. We can modify Kruskal's algorithm to get an online version of minimum spanning tree algorithm for the ongoing match graph.

If the image collection contains singleton views or separated scenes, the initialization process may be unreasonably long since it needs to explore every possible edge to join the singleton image. To increase the stability of the tree structure and cope with singleton images, we consider the mutually-connected edge weight. We query the i-th image with respect to the other images in the dataset and get the rank list $Rank_i$. The rank of image j in $Rank_i$ is denoted as

$Rank_i(j)$. The edge weight $w(e_{ij})$ of node i and node j is defined as the quadratic mean of $Rank_i(j)$ and $Rank_j(i)$, namely $w(e_{ij}) = \sqrt{\frac{Rank_i^2(j)+Rank_j^2(i)}{2}}$. Since quadratic mean is greater or equal to other mean metrics, such as arithmetic mean ($\frac{x_1+x_2}{2}$) or harmonic mean ($\sqrt{x_1 x_2}$), it can be viewed as a worst-case metric to penalize more severely on the edge weight if either of $Rank_i(j)$ or $Rank_j(i)$ is large.

The algorithm first orders the edge set by weights in increasing order and then probes (feature correspondences and geometric verification) the most probable pair that can join two disjoint sets using the union-find data structure. If it succeeds, the two disjoint sets are merged; otherwise it proceeds to the next best probable edge that connects two components. If an image has been involved in S_R failed tests, it is regarded as a singleton image and discarded from the dataset.

The tree match graph seems to be fragile since it contains no loop for consistency checking. To get the most robust initial match pairs, a stricter inlier threshold $S_I^\star (= 40)$ is applied in the match verification. It is assumed that in self-similar environments true positive matches have larger similarity responses compared to false positive ones, even for highly ambiguous scenes (as shown in Fig. 4(a)). Therefore, the tree edges are consistent in nature due to the greedy property of the online minimum spanning tree algorithm. Since our aim is to get a consistent matching set that generates accurate and complete SfM models, we assume that the matching algorithm in the following sections operates on a connected component of the image collection. This *online minimum spanning tree* algorithm is described in Algorithm 1.

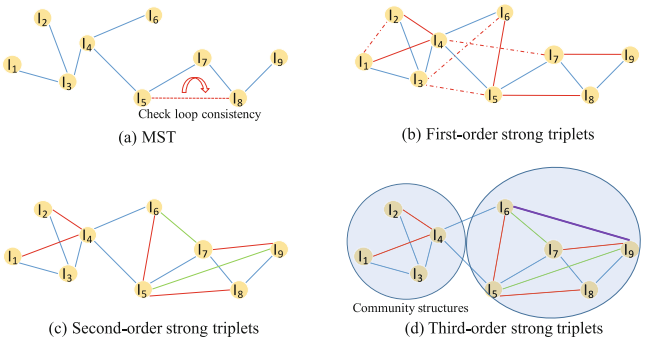

Fig. 2. Illustration of the tree expansion step up to third-order strong triplets. (a) The minimum spanning tree; (b) First-order strong triplets are selected by traversing the minimum spanning tree and checking loop consistency of two adjacent edges. The dashed red lines represents either unmatched image pairs or inconsistent weak triplets, while the solid red lines are verified edges; (c) Second-order strong triplets (marked by solid green lines) built upon the first-order ones; (d) The partial match graph after the expansion step. The community structures are further utilized for checking strong consistency. (Color figure Online)

4.2 Graph Expansion by Strong Triplets

We now consider the trade-off between *efficiency* and *consistency*. Intuitively, strong consistency (Definition 2) is much harder to satisfy because the enumeration of all loops in a graph costs exponential time. Weak consistency (Definition 1) would be relatively easy to achieve, although the time complexity of verifying all 3-length loops is $O(n^3)$ in the worst cases, which is unacceptable for large-scale datasets. Therefore, we aim to exactly satisfy weak consistency with respect to the match graph and approximately guarantee strong consistency on this graph.

We refer to 3-length loops as *strong triplets*, which differs from *weak triplets* that do not form closed loops. The number of strong triplets depends on the structure of the scene dataset, which is agnostic to the matching algorithm. To get a consistent match graph, ideally we want as many strong triplets as possible. Since the adjacent match pairs are the most likely to compose strong triplets, we propose a greedy tree expansion method to grow the match graph from weak triplets.

After connecting different views with a spanning tree, we get a weakly-connected match graph. Two adjacent edges with a common vertex induce a weak triplet and once the two end points get connected it becomes a strong triplet. The detection of all weak triplets can be done efficiently by traversing two steps starting from each node. The first-order strong triplets are formed by traversing the minimum spanning tree, while the second-order strong triplets are built upon the first-order ones and the tree, so on and so forth. The gain of exploring local connectivity diminishes as the triplet expansion process iterates. Therefore, the match graph is expanded up to the third-order strong triplets mainly for the efficiency concern. Figure 2 illustrates the match graph expansion process. Specifically, the pairwise rotation R_{ij} is used as the surrogate for the abstract relative geometric relation T_{ij} in Definitions 1 and 2. The distance function $d(\widetilde{R}, \mathbb{I})$ between the chained rotation $\widetilde{R}$ and $\mathbb{I}$ is defined as the rotation angle of $\widetilde{R}$ as $d(\widetilde{R}, \mathbb{I}) = \theta(\widetilde{R}) = \arccos(\frac{\mathrm{Tr}(\widetilde{R})-1}{2})$.

Different from the methods in [18,33] that uses explicit Bayesian inference to remove inconsistent matches after getting a complete match graph, we generate a consistent match graph in a bottom-up way, thus preventing the interference of good and bad matches. However, since this step only addresses local loops consistency with length 3, the error may accumulate along the longer sequence and cause motion drifts in the SfM model. We further address this issue in the next section.

4.3 Community-Based Graph Reinforcement

The expanded match graph robustly estimates the local structures of scenes and generates a consistent matching set enforced by strong triplets. Although this simplified match graph suffices to generate a consistent reconstruction, it has two major drawbacks. First, in this match graph, only strong triplets get verified and the consistency of longer loops (*strong consistency*) is neglected. Second, this

Algorithm 2. Component Merging Algorithm

Input: The intermediate match graph after triplet expansion $G_t = (\mathcal{V}, \mathcal{E}_t)$, community-wise match number S_C, loop discrepancy threshold θ

Output: The final match graph $G_f = (\mathcal{V}, \mathcal{E}_f)$

Compute structures on G_t and split $\mathcal{V}$ into m communities $\{\mathcal{V}_1, \ldots, \mathcal{V}_m\}$
Create the candidate matching set Φ
Rank Φ in decreasing order by edge weights
Reserve the first $\frac{S_c m(m-1)}{2}$ elements of Φ to get Φ'
for each image pair $(i, j) \in \Phi'$ **do**
    Match (i, j) and compute the relative rotation R_{ij}
    Find the shortest path of length l between (i, j) using *Breath-First-Search* algorithm
    Compute the chained rotation R_c and the discrepancy angle $\theta_c = \arccos(\frac{\mathrm{Tr}(R_c)-1}{2})$
    if $\theta_c < \theta/\sqrt{l}$ **then**
        $\mathcal{E}_t = \mathcal{E}_t \cup (i, j)$
    end if
end for
Iterate 1-12 if the stopping criterion does not satisfy
$\mathcal{E}_f = \mathcal{E}_t$

match graph, without closed-loop structures at a global scale, does not reflect the genuine pose graph of the dataset. Because this matching algorithm starts with a tree structure, the match graph after the triplet completion stage would roughly preserve this skeletal structure.

To tackle the above weaknesses, we propose a component merging algorithm inspired by techniques in community detection. *Community detection* [7] is widely used in the analysis of complex networks. It aims to divide a graph into groups with denser connections inside and sparser connections outside. This allows us to attain a coarse-grained description of the match graph and detect higher-level connectivity. The intra-connectivity within groups is strong enough since it contains consistent strong triplets, while the inter-connectivity in longer loops is left to be detected and verified.

Let A_{ij} be an element of the adjacent matrix of a general graph where $A_{ij} = 1$ if i and j are connected and $A_{ij} = 0$ otherwise. The degree d_i of a node i is the number of other nodes that connects to it, denoted as $d_i = \sum_j A_{ij}$. If the graph is randomized without a significant community structure, the probability of an edge existing between node i and node j is $\frac{d_i d_j}{2m}$, where $m = \frac{1}{2}\sum_{ij} A_{ij}$ is the total number of edges in G. Suppose that the match graph is structured such that the node i belongs to a community V_p and the node j belongs to a community V_q, then the modularity [20] Q measures the difference of the fraction of intra-community connections between a graph and the random graph:

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{d_i d_j}{2m})\delta(V_p, V_q) \tag{3}$$

Table 1. Reconstruction accuracy of three small datasets [27] with ground-truth. The absolute camera location errors c_{err} and camera rotation errors R_{err} are measured in meters and degrees respectively. The *#matches* for our method is showed as *the number of consistent matches / the number of total attempted matches*.

	fountain-P11			*Herz-Jesu-P25*			*Castle-P30*		
	#matches	c_{err}	R_{err}	*#matches*	c_{err}	R_{err}	*#matches*	c_{err}	R_{err}
Ours	52/55	0.019	0.414	155/231	0.030	0.399	112/283	0.220	0.476
Full match	55	0.016	0.407	300	0.028	0.389	435	0.167	0.513

Table 2. Running time and re-projection error for different methods. The meaning of the first row: *Dataset*, the name of the scene; *#views*, the number of cameras; *#Rviews*, the number of successfully registered cameras, with F meaning that the reconstruction fails; *#UM/#TM*, the number of useful matches which pass geometric verification and loop consistency verification / the number of total attempted matches; *M+GV*, the running time of matching and geometric verification; *GO*, the running time of graph operations including loop consistency verification, community detection, etc.; *Total time*, the total running time of the proposed matching algorithm; *Speedup*, the speedup factor of our matching algorithm w.r.t *Voc100*; *ReprojError*, mean re-projection error (in pixels) of resulted SfM models; The running time of vocabulary tree is not documented since both methods depend on it. The matching time of *Voc25* is not recorded as well since it has roughly the same number of matches as that of the graph-based matching method.

Dataset	#views	#Rviews			#matches				Running Time					ReprojError		
		Ours	Voc25	Voc100	Ours		Voc25	Voc100	Ours			Voc100	Speedup	Ours	Voc25	Voc100
					#UM	#TM			M+GV	GO	Total time					
SportsArena	157	151	F	F	529	2621	2654	9554	8.8 min	0.1 min	8.9 min	29.4 min	3.3x	0.736	F	F
TempleOfHeaven	341	341	F	F	2795	2901	4483	18466	14.2 min	0.2 min	14.4 min	68.0 min	4.7x	0.544	F	F
NotreDame	699	675	685	687	11765	15729	12142	45280	0.98 hrs	0.7 min	0.99 hrs	2.76 hrs	2.8x	**0.654**	0.705	0.672
TreviFountain	1906	1906	1906	1906	27182	30187	33821	124925	1.87 hrs	2.1min	1.90 hrs	7.10 hrs	3.9x	**0.568**	0.693	0.602
Colosseum	2006	1980	1999	2006	26723	32653	35130	143439	2.72 hrs	2.3 min	2.76 hrs	11.23 hrs	4.1x	**0.466**	0.615	0.489

where $\delta(i, j) = 1$ if $i = j$ and 0 otherwise. If every node is itself a community, the modularity is zero. In practice, a value larger than 0.3 indicates that the graph has a significant community structure. After the triplet expansion step, the community structure on the match graph manifests the sparse connections between communities that may yields incomplete SfM models due to insufficient tracks and wide baseline. In this case, even though the match graph is connected as a whole, SfM may still fail into separated models. We aim to find the community structure of the match graph and reinforce intra-community connectivity visual similarity cues.

To avoid defeating the purpose of speeding up the pairwise matching, we choose a fast greedy approach to estimate the community structure [5]. This hierarchical algorithm starts with each node being a sole community and iteratively joins separate communities whose amalgamation results in the largest increase in Q. Specifically, we use the weighted graph with the edge weight being

the number of fundamental matrix inliers F_{inlier} between an image pair, which helps identify weak connections. Therefore, A_{ij} of the match graph in Eq. 3 is defined as

$$A_{ij} = \begin{cases} F_{inlier} & i \, and \, j \, are \, connected \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The match graph is merged into a single community after $n-1$ such joins. The modularity Q has a single peak over the generation of the dendrogram [5] which indicates the most significant community structure. We take the vertex partition when the modularity reaches the peak. Hence the number of communities depends upon the connectivity pattern of the match graph.

After getting the community structure of the intermediate match graph, image pairs across groups constitutes a candidate list $\Phi = \{(i,j)|i \in \mathcal{V}_p, j \in \mathcal{V}_q, p = [1,\ldots,m], q = [1,\ldots,m], p \neq q\}$ for matching and geometric verification. For a match graph with n nodes, if the community detection algorithm generates m groups of roughly equal size, the scale of the candidate matching set is $O((\frac{n}{m})^2)$, which is still quadratic in the number of images. To reduce the cost of matching, we rank the candidate list by quadratic mean of $Rank_i(j)$ and $Rank_j(i)$, and only match the most probable S_C community-wise pairs. Thus the candidate list is pruned to a smaller matching set Φ' of size $\frac{S_C m(m-1)}{2}$, which only depends on the number of communities. This component merging process is iterative and stops if the number of communities does not change.

The issue with *strong consistency* is taken care of during the graph reinforcement stage. Intuitively it can be only achieved approximately since verifying all loops is computationally intractable. As is observed in several previous studies [6,18], random errors accumulated in the longer loops affect the effectiveness of loop consistency checking. As a result, the verification process further simplifies to checking the strong consistency with respect to the shortest loop that contains the new edge. For an image pair (I_i, I_j), we use breath-first-search algorithm to find the shortest path in the match graph. Together with the direct link between I_i and I_j, they form the shortest loop. The loop consistency of this cycle is verified with a discrepancy threshold weighted by the cycle length [6]. The full component merging algorithm is given in Algorithm 2.

5 Experiments

Implementation. We used SiftGPU [30] to extract and match SIFT [17] features. To compute similarity scores and rank information, we implemented a multi-threaded version of the vocabulary tree algorithm [22] to ensure the cost of image retrieval is significantly lower than that of the matching process. The vocabulary tree has a depth of 6 and a branching factor of 8 with tf-idf weighting and min-distance metric. We used 7-point algorithms embedded in RANSAC [8] to compute the fundamental matrix of image pairs for geometric verification. We obtained the SfM models using a standard incremental SfM pipeline using [2] as the underlying bundle adjustment solver. All experiments were running on a multi-core PC with Intel(R) Core(TM) i7-4770K processors and 32 GB main

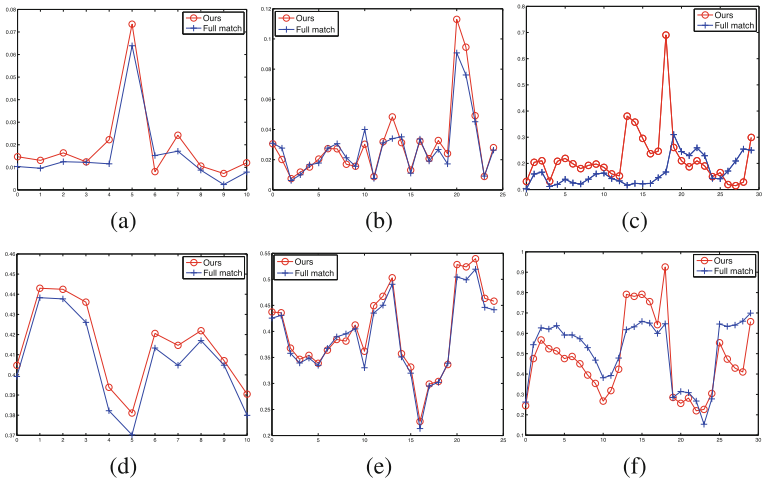

Fig. 3. Per-camera absolute location errors of (a) *fountain-P11* (b) *Herz-Jesu-P25* (c) *Castle-P30* and the corresponding per-camera orientation errors (d)(e)(f). Our method achieves the same level of accuracy as that of the full match, despite the fact that the number of matches is much smaller than the full match.

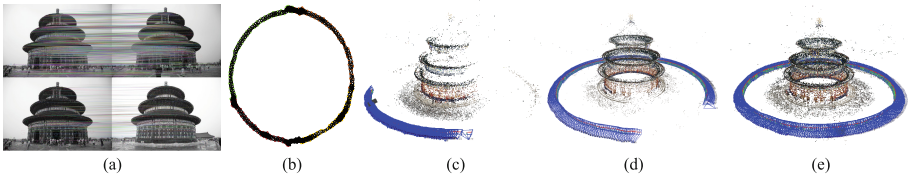

Fig. 4. Disambiguation performance of different methods on the highly ambiguous *TempleOfHeaven* dataset. (a) The true-positive feature correspondences (top: the front-front match) and the false-positive feature correspondences (bottom: the front-back match) of *TempleOfHeaven*. After geometric verification using fundamental matrix, the erroneous false-positive match has much fewer inliers (246) than that of the true-positive match (2349); (b) The match graph of our method; (c) The SfM models using *Voc100* as input; (d) The SfM models using [33] applied on *Voc100* as input; (e) The correct SfM model using the consistent matching method.

memory. We used the same set of parameters for all experiments with pairwise inlier number $S_I = 20$, singleton rejection threshold $S_R = 20$, community-wise match number $S_C = 30$ and loop discrepancy threshold $\theta_c = 2°$.

Datasets. We tested the algorithm on three types of datasets, namely the benchmark datasets, the Internet datasets, and the ambiguous datasets. First, the datasets *fountain-P11*, *Herz-Jesu-P25*, and *Castle-P30* were obtained from the well-known benchmark datasets [27] with the ground-truth camera calibrations. Further, we tested the scalability and efficiency of the method on relatively large Internet datasets from [1], namely *NotreDame*, *TreviFountain* and *Colosseum*.

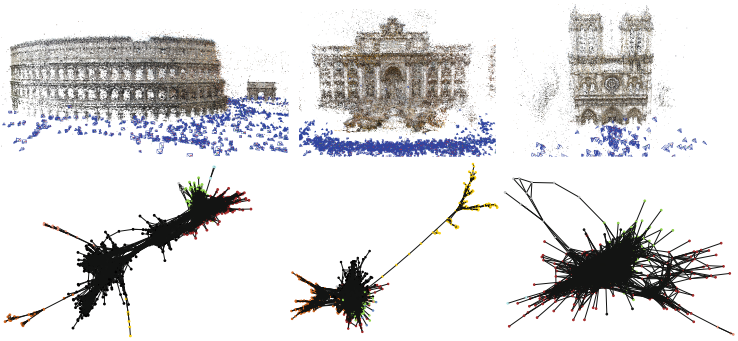

Fig. 5. 3D Reconstruction models and their corresponding match graphs for Rome16 K [1] datasets. From left to right: *Colosseum*, *TreviFountain*, *NotreDame*. The corresponding match graphs show the community partition using different colors. (Color figure online)

Finally, we show our method has superior performance on ambiguous datasets which are *Cup*, *Books*, *Desk*, *ForbiddenCity*, *Indoor*. We also introduce two more highly ambiguous datasets, namely *SportsArena* (Fig. 1) and *TempleOfHeaven* (Fig. 4).

Accuracy Evaluation. We conducted experiments on multi-view benchmark datasets [27] to evaluate accuracy. We fix the SfM pipeline and use the full matching and the graph-based matching results as inputs respectively. The reconstruction accuracy is measured by the error of absolute camera location c_{err} and the absolute error of camera orientation R_{err}, in meters and degrees respectively. We do not conduct comparison with image retrieval techniques because the datasets are too small and candidate lists for vocabulary tree are hard to choose. The experiment results (see Table 1 and Fig. 3) show that the graph-based method is adaptive to the complexity of the datasets and achieves the same level of accuracy as that of the full match.

Scalability and Efficiency. Next, we tested the scalability and time efficiency of the method on Internet datasets [1]. For comparison the matching set containing top-100 candidates per image, denoted as *Voc100*, is retrieved and matched. We ensure that this fixed number is enough for reconstruction and all failure cases of *Voc100* are not caused by insufficient matches. A smaller matching set *Voc25*, composed of top-25 candidates per image, is also retrieved and compared. The aim is to test the effectiveness of the proposed method. *Voc25* roughly contains the same number of matches as the consistent matching set but does not necessarily guarantee complete SfM results. We measure the re-projection error of SfM results with different matching inputs. The comparison result (see Table 2) shows that the graph-based matching algorithm significantly accelerates the matching process, with speedup factors ranging from 3.3 for *SportArena*, to 4.7 for *TempleOfHeaven* compared with *Voc100*. The overhead introduced by various graph operations is negligible compared to the cost of pairwise image

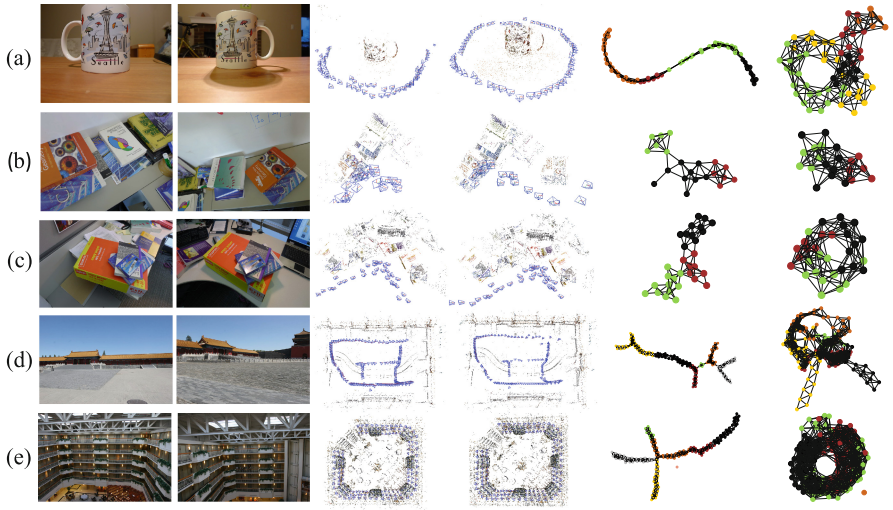

Fig. 6. Experiment results on ambiguous datasets from previous works [14,24]. From left to right: the first two columns - two views of an ambiguous scene; 3rd column - the SfM model using full match; 4th column - the SfM model using consistent matching; 5th column - the match graph after triplet expansion; 6th column - the match graph after component merging (different colors represents different communities). From top to bottom: (a) *Cup* (b) *Books* (c) *Desk* (d) *ForbiddenCity* (e) *Indoor*. (Color figure online)

matching, as is shown in the GO column of Table 2. The re-projection error of the SfM models with the consistent matching set is systematically smaller. We also found that the community-based graph reinforcement step is crucial in the success of large-scale reconstructions since the local consistency achieved by triplet expansion fails to detect the community-level connectivity. Please refer to the supplementary materials for more details.

Redundancy and Disambiguation. We then used ambiguous datasets to demonstrate how erroneous matches can ruin the final SfM result. *TempleOfHeaven* dataset is composed of 341 rotationally symmetric images, which is a failure case of Jiang et al. [14]. In this extremely symmetric dataset, even the front views and the back views would match and form a reasonable epipolar geometry (see Fig. 4(a)).

The 3D reconstruction with *Voc100* and *Voc25* of the *TempleOfHeaven* dataset both yielded folded structures. The same went for the *SportsArena* dataset, in which *Voc100* generated a 3D reconstruction with superimposed structures (see Fig. 1) and the reconstruction of *Voc25* contained erroneous registrations of camera poses. All these cases failed because the methods mentioned above ignore the structure of the scene and lack geometric consistency checking. In contrast, the proposed matching method progressively explores the connectivity from the local to the whole and check consistency along the path. Figure 4(b)

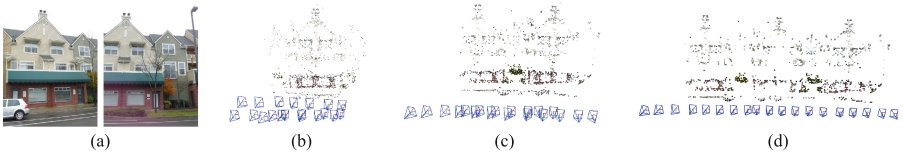

Fig. 7. A failure case. (a) Two views from a self-similar scene; (b) The SfM model using full match; (c) The SfM model using the consistent matching algorithm; (d) The correct SfM model solely using the skeletal tree match graph as input.

visualizes the final match graph which greatly resembles the actual scene. We also tried the iterative version [18] of removing erroneous match pairs using Bayesian inference proposed by Zach et al. [33] on *TempleOfHeaven*. Although it removed 3575 out of 18466 match pairs, the obtained matching set still failed to render a correct reconstruction. We also applied this method on Internet datasets to filter the matches. But it was generally infeasible for large-scale datasets since a single iteration would take more than 8 h on *Voc100* of the *TreviFountain* dataset, due to the fact that the inference on Bayesian networks is generally NP-hard.

We further tested our matching algorithm on several ambiguous datasets from previous works [14,24] and the results are showed in Fig. 6. Solely by optimizing the input match graph, our consistent matching yields efficient and correct camera pose registrations compared with the exhaustive matching method.

Limitations. The current loop consistency checking is solely based on pairwise rotation, making it difficult to detect the inconsistency in datasets with pure translation motion, such as *Street* dataset (see Fig. 7). Thus it is possible to extend our algorithm to check the chained pose consistency, namely using displacements on a fixed scale as the surrogate relative transformation to verify loop consistency.

6 Conclusions

In this paper, we present a unified image matching framework using greedy graph expansion and community detection to discover both local and inter-community consistent match pairs. Our method significantly reduces the number of image pairs for matching without degrading the quality of subsequent SfM pipeline, and improves the robustness of SfM in scenes with ambiguous structures.

Our approach provides a sufficient and consistent image matching set as the input of SfM. This matching framework does not assume knowing any global motion information, nor incorporate translation or other scale-dependent constraints into the loop consistency checking. Hence, our future work is to combine the components in SfM, e.g. track selection and global pose registration, to further optimize 3D reconstruction.

Acknowledgement. The authors would like to thank all the anonymous reviewers for their constructive feedbacks. This work is supported by Hong Kong RGC 16208614, T22-603/15N, Hong Kong ITC PSKL12EG02, and China 973 program, 2012CB316300.

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Commun. ACM **54**(10), 105–112 (2011)
2. Agarwal, S., Mierle, K., et al.: Ceres solver. http://ceres-solver.org
3. Chum, O., Matas, J.: Large-scale discovery of spatially related images. PAMI **32**(2), 371–377 (2010)
4. Chum, O., Mikulik, A., Perdoch, M., Matas, J.: Total recall ii: query expansion revisited. In: CVPR, pp. 889–896 (2011)
5. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6), 066111 (2004)
6. Enqvist, O., Kahl, F., Olsson, C.: Non-sequential structure from motion. In: ICCV Workshops, pp. 264–271 (2011)
7. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3), 75–174 (2010)
8. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, New York (2003)
9. Havlena, M., Torii, A., Pajdla, T.: Efficient structure from motion by graph optimization. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6312, pp. 100–113. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15552-9_8
10. Havlena, M., Torii, A., Knopp, J., Pajdla, T.: Randomized structure from motion based on atomic 3d models from camera triplets. In: CVPR, pp. 2874–2881 (2009)
11. Heinly, J., Dunn, E., Frahm, J.-M.: Correcting for duplicate scene structure in sparse 3D reconstruction. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 780–795. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10593-2_51
12. Heinly, J., Schonberger, J.L., Dunn, E., Frahm, J.M.: Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In: CVPR, pp. 3287–3295 (2015)
13. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88682-2_24
14. Jiang, N., Tan, P., Cheong, L.F.: Seeing double without confusion: structure-from-motion in highly ambiguous scenes. In: CVPR, pp. 1458–1465 (2012)
15. Kulis, B., Grauman, K.: Kernelized locality-sensitive hashing for scalable image search. In: ICCV, pp. 2130–2137 (2009)
16. Lou, Y., Snavely, N., Gehrke, J.: MatchMiner: efficient spanning structure mining in large image collections. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7573, pp. 45–58. Springer, Heidelberg (2012)
17. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2), 91–110 (2004)
18. Moulon, P., Monasse, P., Marlet, R.: Global fusion of relative motions for robust, accurate and scalable structure from motion. In: ICCV, pp. 3248–3255 (2013)

19. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. VISAPP **1**, 331–340 (2009)
20. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
21. Nistér, D.: An efficient solution to the five-point relative pose problem. PAMI **26**(6), 756–770 (2004)
22. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR, pp. 2161–2168 (2006)
23. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR, pp. 1–8 (2007)
24. Roberts, R., Sinha, S.N., Szeliski, R., Steedly, D.: Structure from motion for scenes with large duplicate structures. In: CVPR, pp. 3137–3144 (2011)
25. Schönberger, J.L., Berg, A.C., Frahm, J.M.: Paige: pairwise image geometry encoding for improved efficiency in structure-from-motion. In: CVPR, pp. 1009–1018 (2015)
26. Snavely, N., Seitz, S.M., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: CVPR (2008)
27. Strecha, C., von Hansen, W., Gool, L.V., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: CVPR, pp. 1–8 (2008)
28. Sweeney, C., Sattler, T., Hollerer, T., Turk, M., Pollefeys, M.: Optimizing the viewing graph for structure-from-motion. In: ICCV, pp. 801–809 (2015)
29. Wilson, K., Snavely, N.: Network principles for sfm: disambiguating repeated structures with local context. In: ICCV, pp. 513–520 (2013)
30. Wu, C.: Siftgpu: A gpu implementation of scale invariant feature transform (sift) (2007)
31. Wu, C.: Towards linear-time incremental structure from motion. In: 3DV, pp. 127–134 (2013)
32. Zach, C., Irschara, A., Bischof, H.: What can missing correspondences tell us about 3d structure and motion? In: CVPR, pp. 1–8 (2008)
33. Zach, C., Klopschitz, M., Pollefeys, M.: Disambiguating visual relations using loop constraints. In: CVPR, pp. 1426–1433 (2010)
34. Zhou, X., Zhu, M., Daniilidis, K.: Multi-image matching via fast alternating minimization. In: CVPR, pp. 4032–4040 (2015)
35. Zhu, S., Fang, T., Xiao, J., Quan, L.: Local readjustment for high-resolution 3d reconstruction. In: CVPR, pp. 3938–3945 (2014)