EFFICIENT WIDE BASELINE STRUCTURE FROM MOTION

Mario Michelini and Helmut Mayer

Institute for Applied Computer Science
Bundeswehr University Munich
{mario.michelini,helmut.mayer}@unibw.de

Commission III, WG III/1

KEY WORDS: Structure from Motion, Wide Baseline, Pose Estimation, Hierarchical, Graph, Reconstruction, Embedding

ABSTRACT:

This paper presents a Structure from Motion approach for complex unorganized image sets. To achieve high accuracy and robustness, image triplets are employed and (an approximate) camera calibration is assumed to be known. The focus lies on a complete linking of images even in case of large image distortions, e.g., caused by wide baselines, as well as weak baselines. A method for embedding image descriptors into Hamming space is proposed for fast image similarity ranking. The later is employed to limit the number of pairs to be matched by a wide baseline method. An iterative graph-based approach is proposed formulating image linking as the search for a terminal Steiner minimum tree in a line graph. Finally, additional links are determined and employed to improve the accuracy of the pose estimation. By this means, loops in long image sequences are implicitly closed. The potential of the proposed approach is demonstrated by results for several complex image sets also in comparison with VisualSFM.

1 INTRODUCTION

Recent developments for Structure from Motion (SfM) techniques from unorganized image sets focus on large photo collections downloaded from the internet (Heinly et al., 2015; Snavely et al., 2008; Agarwal et al., 2009; Frahm et al., 2009; Havlena et al., 2010; Crandall et al., 2011). Such collections can contain thousands or even millions of images comprising a very high redundancy and often moderate baselines. In contrast to these large photo collections, we focus on smaller image sets up to a few thousand images, but containing complex configurations comprising wide as well as weak baselines between images.

Wide baselines often arise by combining terrestrial and aerial imagery. Failure to handle them can lead to an incomplete pose estimation. On the other hand, weak baselines result if the translation between image acquisitions is insufficient in relation to the distance to the observed scene. They lead to a poor intersection geometry which becomes undefined in case of zero baseline (i.e., pure rotation). Incorrect handling of weak baselines results in an inaccurate or failed estimation of camera poses.

In this paper, the goal is a complete linking of all images in sets of moderate size to obtain accurate estimates of camera poses even for complex configurations consisting of wide as well as weak baselines. In our case, wide baselines arise primarily when combining terrestrial images and images from small Unmanned Aerial Systems (UAS).

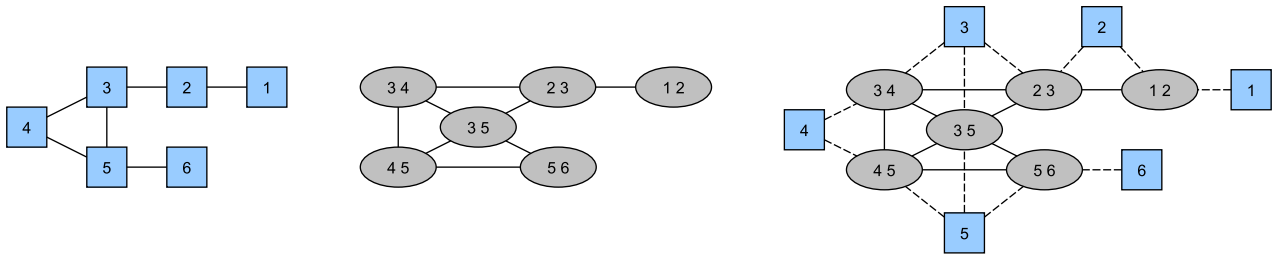
Usually, the first step in SfM is the establishment of feature correspondences, i.e., *image matching* (Hartmann et al., 2015). Yet, because of the high combinatorial complexity, exhaustive image matching is not practical even for small image sets. The most commonly used method to reduce the combinatorial complexity is pruning of the image set. In (Li et al., 2008; Frahm et al., 2009) clustering of the image set based on the global GIST descriptor (Oliva and Torralba, 2001) is carried out to find representative images which are then employed to incrementally compute the 3D structure. On the other hand, the number of local feature matches was used in (Simon et al., 2007; Quack et al., 2008; Philbin and Zisserman, 2008) for clustering. In (Havlena et al., 2013), the reduction of the image set is formulated as the search for a minimally connected dominating set of the graph of pairwise connections between the images.

Recent approaches for large photo collections (Havlena and Schindler, 2014; Agarwal et al., 2009; Klopschitz et al., 2010) use quantized local features (Sivic and Zisserman, 2003) indexed by a vocabulary tree (Nister and Stewenius, 2006) to reduce the complexity. Vocabulary trees scale well for large image sets, but require a training phase and the specification of parameters (number of clusters and tree depth) which have a strong influence on the accuracy. Acceleration of matching itself using GPU was employed in (Wu, 2011; Frahm et al., 2009). Holistic features (Oliva and Torralba, 2001) indexed by compact hashing codes (Raginsky and Lazebnik, 2009; Torralba et al., 2008) were used in (Frahm et al., 2009) to reduce the memory consumption and speed up the matching. Also dimension reduction (Cai et al., 2011; Ke and Sukthankar, 2004) or embedding (Cheng et al., 2014; Strecha et al., 2012; Jegou et al., 2008; Torralba et al., 2008) of feature descriptors were employed.

When computing geometric relations between images purely based on image features, the ability to find correspondences even between images with large geometric or radiometric distortions, e.g., caused by wide baselines or different acquisition times, is highly desirable. Unfortunately, the establishment of correspondences in these cases requires complex algorithms (Mayer et al., 2012) with a strongly negative influence on the scalability of SfM. Applying accelerations techniques similar to Schönberger et al. (2015) and Raguram et al. (2012) is less suitable, because they would make the geometric verification faster but less robust against complex configuration which we intend to handle.

To this end, we employ fast filtering for overlapping images forming pairs based on their similarities and perform complex geometric verification only for a small subset of pairs. For the estimation of the image similarities a technique for fast and unsupervised descriptor embedding is proposed. It produces binary descriptors allowing for fast estimation of relative image similarities.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
doi:10.5194/isprsannals-III-3-99-2016



(a) Image graph (b) Pair graph (c) Linking graph

Figure 1. Image graph with corresponding line and linking graph. The line graph of the image graph is the pair graph describing relations between pairs. Adding nodes of the image graph to the pair graph results in the linking graph where rectangles represent the image and ellipses the pair nodes.

Pairs comprising a consistent geometry are merged into a common reference frame followed by bundle adjustment (Triggs et al., 1999) to obtain relative camera poses (Agarwal et al., 2009; Wu, 2011; Frahm et al., 2009; Li et al., 2008; Snavely et al., 2006). To improve robustness and accuracy, image triplets instead of pairs are employed (Moulon et al., 2013; Klopschitz et al., 2010).

Yet, using triplets increases the combinatorial complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$, where n is the number of images. Thus, we estimate the geometry for pairs first and derive triplets afterwards based on the information from the image pairs. We present a theoretically well founded modeling for the later allowing for efficient image linking. It is based on the concept of line graphs formulating optimal linking as the search for a terminal Steiner minimum tree (Lin and Xue, 2002).

The paper is organized as follows: In Section 2 the descriptor embedding is presented. The theoretical background used for the image linking is presented in Section 3. Our SfM pipeline is described in Section 4. Results which demonstrate the potential of the proposed pipeline are presented in Section 5. Finally, in Section 6 conclusions are given.

2 DESCRIPTOR EMBEDDING

This section describes an approach for embedding SIFT descriptors (Lowe, 2004) from real space $\mathbb{R}^{128}$ into Hamming space $\mathbb{H}^{128} = \{0,1\}^{128}$. The embedding allows for a compact representation of the descriptors as bit vectors and, thus, a very fast comparison.

Existing work concerning embedding of feature descriptors focuses on distance preservation, i.e., two descriptors which are close in $\mathbb{R}$ should be also close in $\mathbb{H}$. For example, in (Strecha et al., 2012; Torralba et al., 2008) supervised machine learning techniques were applied to achieve this goal. In our case, we want to rank images based on their similarities. Therefore, we are only interested in relative similarities. Thus, a simplified embedding can be used as long as the approximation errors are small or distributed evenly.

A d-dimensional real space $\mathbb{R}^d$ can be partitioned by d independent (affine) hyperplanes of codimension one in $\mathbb{R}^d$. Each hyperplane goes through the intersection point $\mathbf{p} \in \mathbb{R}^d$ and separates $\mathbb{R}^d$ into two halfspaces, termed the positive and the negative halfspace. Intersections of d mutually orthogonal halfspaces determine 2^d orthants, i.e., the generalization of quadrants in $\mathbb{R}^2$ to $\mathbb{R}^d$. Every orthant is determined by a sequence of d plus or minus signs where the ith sign indicates whether the orthant is in the positive or negative halfspace of the ith hyperplane. Thus, an orthant in $\mathbb{R}^d$ can be represented by a bit vector of length d.

A 128-dimensional SIFT descriptor points to one of the 2^{128} orthants. Therefore, it can be represented by a bit vector of length 128 corresponding to the orthant it points to. For the embedding one needs to define the 128 hyperplanes. The values of the normalized descriptor lie in the range 0 to 1. Hence, the origin as the intersection point $\mathbf{p} = \mathbf{0}$ for the hyperplanes is not a good choice and, thus, $\mathbf{p}$ must be determined to ensure an appropriate embedding. For this, the median of all descriptor values is computed for each dimension i and used as the ith coordinate of the intersection point $\mathbf{p}$.

Matching of the embedded descriptors then means determination of the number of corresponding halfspaces, instead of computing the Euclidean distance in case of the original descriptors. The former can be computed very fast on recent CPU architectures using XOR followed by bit counting. This provides only a rough approximation of the true correspondences, but it turns out to be accurate enough for the image similarity ranking (cf. Section 5). On the other hand, the comparison of the embedded descriptors reduces the matching runtime drastically and therefore allows for exhaustive matching of image descriptors.

3 IMAGE LINKING

We term the merging of images to larger image subsets *image linking*. The most common way to model linking is based on formulating it as a graph problem. The relationships between images are modeled by an undirected graph, where nodes correspond to the images and edges connect pairs of images that overlap. We denote this representation *image graph*.

3.1 Linking Graph

The image graph provides a straightforward modeling if pairs are used as the basic elements for linking, because it concisely describes the pairwise relationships between images. However, when linking is based on triplets, the image graph lacks descriptiveness because it describes pairwise relationships, whereas higher order relationships are required. In addition, we use pairs to propagate the geometry throughout linking, meaning that linkable triplets must have two images in common. This constraint cannot be modeled by the image graph.

A more appropriate modeling could probably be achieved by using 3-uniform hypergraphs where an edge connects three nodes. But despite of the inherent complexity this also does not provide an intuitive way to model the geometry propagation via pairs. Thus, we propose a modeling based on the concept of line graphs. It can be used to handle linking over triplets and enforce geometry propagation via pairs, still allowing the usage of ordinary graph algorithms.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
doi:10.5194/isprsannals-III-3-99-2016

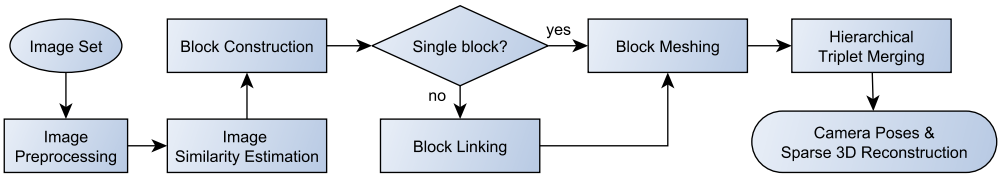


Figure 2. Wide Baseline Structure from Motion Pipeline

The *line graph* $L(G) = (V_L, E_L)$ of an undirected graph $G = (V_G, E_G)$ has as set of nodes the edges of G. Two nodes in $L(G)$ are adjacent iff they have exactly one node of G in common. V_g represents the node set and E_g the edge set of a graph g. Given the incidence matrix R_G of graph G, the adjacency matrix $A_{L(G)}$ of the corresponding line graph $L(G)$ is given by

$$A_{L(G)} = R_G^T R_G - 2I, \tag{1}$$

with I the identity matrix. The number of nodes in $L(G)$ equals the number of edges in G, i.e., $|V_L| = |E_G|$. The number of edges in the line graph is

$$|E_L| = \frac{1}{2} \sum_{i=1}^{|V_G|} d_i^2 - |E_G|, \tag{2}$$

where d_i is the degree of the i-th node in G. It follows that each node i in G with degree d_i generates d_i nodes in the line graph $L(G)$ that are all connected to each other, corresponding to $\binom{d_i}{2}$ connections. Thus, E_L is highly related to the density of graph G.

Hence, line graphs are suitable to describe higher ordered relationships (see Figure 1). The line graph $L(IG)$ of the image graph IG contains nodes corresponding to pairs of overlapping images. Therefore, we denote this representation the *pair graph* (PG). Two nodes in the PG are adjacent if the pairs have an image with an overlapping region in common, yielding a triplet. By this means, a traversal through the pair graph implicitly corresponds to the linking of triplets using pairs to propagate the geometry.

By extending the pair graph to explicitly represent the images using a second node type, the *linking graph* is constructed (see Figure 1), which we use to model the linking of images. It comprises two node types corresponding to the nodes of the image graph (*image nodes*) and the pair graph (*pair nodes*). An image and a pair node are adjacent if the pair node contains the image corresponding to the image node. Image nodes are only required to model the image linking. Therefore, there exist no edges between image nodes.

3.2 Block

The linking graph completely describes the linking between the images. However, it can contain links of varying stability concerning pose estimation. It means that, some triplets used for linking can be more or less stable, e.g., due to short baselines between one or more images. What is more, not all triplets (implicitly) described by the linking graph are required for pose estimation. Thus, using all triplets would increase the runtime without a significant benefit.

We, therefore, introduce the concept of a *block* as a linking subgraph used for the hierarchical merging of triplets to obtain the camera poses. Its *size* describes the number of linked images. A *complete* block is a block linking all images of the image set. The block *density* is the number of triplets used for linking and the *stability* represents its robustness against poor intersection geometry, affecting the quality of the pose estimation.

A pair/triplet is termed *valid* if sufficient feature correspondences between its images could be established, otherwise *invalid*. For valid pairs/triplets we further distinguish between *stable* with good intersection geometry, *instable* with an insufficient baseline and *critical* which could be instable. Only stable and critical triplets are used in a block, where critical triplets are included only if they are essential for the completeness of the block.

In (Beder and Steffen, 2006) a score for the stability of an image pair is proposed based on the error ellipsoids of the reconstructed 3D points. The quality of a reconstructed 3D point x is estimated by the roundness $R(x)$ of the error ellipsoid which is defined as

$$R(x) = \sqrt{\frac{\lambda_3}{\lambda_1}}, \tag{3}$$

where C_x is the covariance matrix for x and $\lambda_1^x \geq \lambda_2^x \geq \lambda_3^x$ are the eigenvalues of C_x. $R(x)$ lies between 0 and 1 and only depends on the relative geometry of the two cameras and the feature positions. If the two camera centers are identical and the feature positions were correct, the roundness would be equal to zero. The mean roundness for all reconstructed points determines the stability of a pair.

We employ the proposed quality measure to weight the edges between the pair nodes in the linking graph. For an accurate pose estimation a compromise between a sufficiently wide baseline and a large number of correspondences must be found. To achieve that, we weight each correspondence using R and compute the sum to obtain the quality score $s(P)$ for a pair P:

$$s(P) = \sum_{x \in P} R(x) \tag{4}$$

By this means, we do not only take the number of correspondences into account but also their quality. The stability of a triplet is determined by the stability of its weakest image pair. Therefore, we define the weight $w(e)$ of an edge e between pair nodes by

$$w(e) = \min \{s(P_1), s(P_2), s(P_3)\}, \tag{5}$$

where P_1 and P_2 are pairs corresponding to the adjacent pair nodes and $P_3 = (P_1 \cup P_2) \setminus (P_1 \cap P_2)$.

Having a weighted linking graph, we can determine a block of minimum density to link the images into a single coordinate frame. This can be formulated as search for a terminal Steiner minimum tree (Lin and Xue, 2002): Given an undirected, weighted Graph $G = (V, E)$ and a subset $R \subseteq V$ of nodes (*terminals*), a *Steiner tree* is an acyclic subgraph of G that spans all terminals. Other nodes $V \setminus R$ are referred to as *Steiner nodes*. The weight of a Steiner tree is the sum of the weights of all its edges. The *Steiner tree problem* is concerned with the determination of a Steiner tree with minimum weight in G. A Steiner tree is a *terminal Steiner tree* if all terminals are leaves of the Steiner tree. In the context of the linking graph the image nodes correspond to the terminals and the pair nodes to the Steiner nodes.

It should be noted that a formulation using spanning trees is not appropriate in case of the linking graph. A spanning tree deter-

mines a subset of edges of a given graph whereas the Steiner tree can be used to determine also a subset of nodes. The later is essential in our case to reduce the complexity.

4 WIDE BASELINE STRUCTURE FROM MOTION

In this section we present a calibrated SfM approach for unorganized image sets. It employs triplets to improve robustness and accuracy as well as pairs for geometry propagation during image linking. The objectives of the proposed approach are a complete image linking and a robust handling of complex configurations.

The SfM pipeline is summarized in Figure 2. The first stage is image preprocessing generating image pyramids and features used for matching (Section 4.1). Based on it, similarities between images are estimated and employed for the selection of pairs which are matched using a wide baseline method (Section 4.2). A block suitable for fast pose estimation is constructed in the next stage (Section 4.3). Multiple blocks are linked into a single block (Section 4.4) and refined by including additional links (Section 4.5). Finally, hierarchical triplet merging (Mayer, 2014) generates relative camera poses for all images of the block.

4.1 Image Preprocessing

In the image preprocessing stage image pyramids are computed and SIFT features (Lowe, 2004) are detected for all images using a GPU-based implementation (Wu, 2007). The SIFT descriptors are embedded into Hamming space (Section 2). To reduce the memory consumption, image pyramids and features are stored in a database.

4.2 Image Similarity Estimation

The most time consuming part of many SfM approaches is image matching, i.e., determination of feature correspondences between images followed by the estimation of their relative pose. Hence, it is essential to limit the number of required image matchings for an efficient SfM. This can be achieved by reduction of the combinatorial and/or the algorithmic complexity.

In our case, reduction of the algorithmic complexity would mean improvement of the runtime needed for wide baseline matching (WBM). Unfortunately, a WBM method such as (Mayer et al., 2012) requires more complex computations compared to small baseline matching to be able to deal with the large deformations caused by wide baselines. Thus, a reduction of algorithmic complexity would lead to a decrease of the robustness concerning the scenario it was designed for, namely wide baselines.

We, therefore, reduce the combinatorial complexity, avoiding to match all images to each other. To this end, a two-stage-matching scheme is employed: The first stage employs very fast matching based on the embedded descriptors allowing pair-wise matching to estimate the similarities between all images. It is important to note that we are interested in the relative similarity only and do not strive for global relevance. Thus, inaccuracies in matching of the embedded descriptors do not have a significant negative influence (cf. Section 5). The obtained similarities are used for ranking and, thus, the selection of image pairs (complexity reduction) which are to be matched using WBM in the second stage. By this means, only a very small fraction of pairs needs to be matched using time-consuming WBM.

The embedded descriptors are compared using Hamming distance followed by the distance ratio test (Lowe, 2004). Dissimilarities between images are estimated using the Jaccard distance

$$J_\delta(i,j) = 1 - J(i,j) = 1 - \frac{|F_i \cap F_j|}{|F_i \cup F_j|} \qquad (6)$$

as a normalized similarity metric (Levandowsky and Winter, 1971). $J(i,j)$ is the Jaccard index (Jaccard, 1912) for the feature sets F_i and F_j of the two images i and j, where $|F_i \cap F_j|$ refers to the number of correspondences. $J(i,j)$ can be viewed as the probability that both images have a randomly selected feature in common (Liben-Nowell and Kleinberg, 2003).

The dissimilarities between images are the basis of the weighted image graph with (6) the edge weight function. It is important to note that no threshold is used for image similarity. Therefore, even images with very high Jaccard distance, except those with distance 1, are not considered as dissimilar at this stage. This is essential for handling pairs with a wide baseline and leads to a dense image graph.

4.3 Block Construction

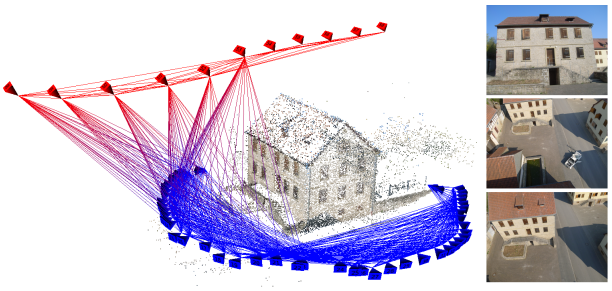
The block construction stage aims to construct a complete and stable block with minimum density. To this end, the linking graph is constructed followed by the determination of the terminal Steiner minimum tree.

The linking graph (LG) can be directly constructed from the image graph. However, from equation (2) one can deduce the large size of the resulting LG, unnecessarily increasing the complexity. Thus, only a subset of the most promising pairs corresponding to the edges of a minimum spanning tree (MST) is used. To ensure the geometric consistency, pairs are verified using WBM (Mayer et al., 2012) as well as low image resolutions for reasons of efficiency.

Valid pairs are classified into stable, instable and critical based on the stability score given by equation (4) and empirically determined thresholds. These were derived using thousands of manually classified stable and instable pairs from various image sets, and, thus, do not have to be adjusted. Instable and invalid pairs are discarded and edges corresponding to them are removed from the MST. Finally, $\mathrm{LG}^{\mathrm{MST}}$ is constructed by deriving the line graph from the MST and merged into the LG, which is empty initially.

A necessary conditions for a complete block is a connected LG and a connected pair graph (PG) induced by the image nodes of the LG. If these conditions are not fulfilled, a new MST is determined. This iterative procedure is repeated as long as the number of connected components of the LG decreases.

If the LG or the PG is disconnected at the end, invalid pairs are examined. They arise if their images do not overlap or comprise a wide baseline configuration. In the latter case, correspondences could not be established due to the use of low image resolutions during WBM. The employed resolutions leading to a couple of hundred points per image are usually sufficient for successful matching of pairs with moderate baselines. By this means, the runtime needed for matching of such pairs, which occur much more often than those with really wide baselines, is kept low. But in case of the latter, the resolution can be insufficient. Simply increasing the resolution for all pairs would lead to an unnecessarily high overall runtime. Instead, we rematch and reclassify only invalid pairs again using higher resolutions and add stable pairs to the LG. To reduce the number of rematched pairs, only those with more than five correspondences so far are used and all other considered as unlikely to overlap. If the construction of a complete block is still not feasible, one could also try to find missing connections for other pairs. But without additional information, this corresponds more or less to a random search and, thus, is deferred to the block linking phase (Section 4.4), where more information is available.



(a) (b)

Figure 3. Image set *House* containing 59 terrestrial and aerial images with wide baselines between them. The obtained camera poses are shown as pyramids on the left hand side. Lines connect images which have at least ten points in common. The images of one of the triplets used to connect terrestrial and aerial images are shown on the right hand side.

Pairs classified as critical could be instable leading to an inaccurate of failed pose estimation. Hence, a search for more stable pairs which can be used as the replacement for the critical is initiated. The basis for the search forms an image subgraph (ISG) corresponding to the LG. For each edge e in an ISG corresponding to a critical pair, cross-edges connecting nodes incident with e with mutual neighbors are determined. Let the edge $(3, 5)$ between nodes 3 and 5 in Figure 1(a) correspond to a critical pair. Then, the cross-edges would be the edges $(5, 2)$ and $(3, 6)$. Cross-edges corresponding to stable pairs are added to the ISG. Derivation of the line graph from the ISG leads to a LG which allows for a construction of a more stable block.

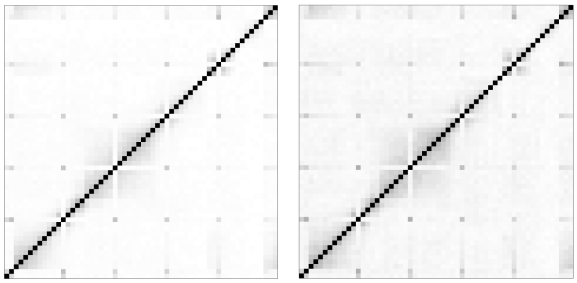
The terminal Steiner minimum tree (TSMT) of the LG determines the block. As terminal Steiner problem has been shown to be NP-complete (Lin and Xue, 2002), an approximation (Chen, 2011) is used. However, the resulting block mostly exhibited a minimum density in practice. The geometric consistency of its triplets is verified using WBM (Mayer et al., 2012) removing invalid triplets.

In the case of invalid triplets, the block becomes invalid and a new TSMT is constructed. This is repeated until a valid block is obtained or the PG becomes empty. If images exist which are not contained in any of the constructed blocks, the LG construction is repeated. This procedure of constructing the LG followed by the determination of the TSMT is iterated until a complete block is constructed or no increase of the block size occurs.

4.4 Block Linking

Missing triplets may cause the construction of multiple incomplete blocks. This is because block construction (Section 4.3) depends on the presence of triplets in blocks sharing two images to be able to link the blocks. This requirement can be relaxed now to two arbitrary images contained in the blocks, not forming an instable pair. Thus, the goal of this stage is to link incomplete blocks to larger ones leading to a complete block in the optimal case.

A compound LG (C-LG), constructed by merging the LGs of the blocks into a single C-LG, is used to guide the search for links from one block to the other. A connected C-LG is a necessary condition for linkable blocks which corresponds to at least one shared image. Yet, a minimum of two shared images is required. Alternatively, a sufficient condition is the connected compound PG which corresponds to linking over a triplet, respectively a pair shared by two triplets.



(a) (b)

Figure 4. Image similarity matrices for the image set *House* constructed using full pairwise matching of original (a) and embedded (b) SIFT descriptors. Each row/column correspond to an image, where darker cells represent higher similarities.

To find the links, pairs containing images from different blocks are constructed. Unlikely pairs are rejected using the model-free outlier rejection rule X84 (Hampel et al., 2011) over image dissimilarities given by the equation (6). The remaining images are sorted in ascending order according to their dissimilarities and used to find connections by verifying and adding them to the C-LG. If all conditions are fulfilled, the blocks are linked to a larger block. Repeating this procedure leads to larger and larger blocks and finally to a complete block.

4.5 Block Meshing

The block exhibits more or less a structure similar to a tree so far containing only triplets essential for image linking. This allows for fast, but potentially not very accurate pose estimation, especially for large scenes containing long image sequences. In addition, shorter baselines are preferred by trend during block construction to ensure successful establishment of correspondences. However, this increases the probability of less stable configurations. Thus, additional links are added to the block increasing its density to extend the baselines and to stabilize long image sequences. This also leads to an implicit loop closing in case of images sequences forming loops.

The idea behind block meshing is to find so called *cross-links* between distant triplets of a block, which are able to link an image contained in one triplet to the other. By this means, the length of a feature track is increased, thus, improving the stability of the block and increasing its density.

Given a pair P corresponding to a pair node of the pair graph (PG), the candidate image set K_p for each image $p \in P$ is determined. An image k is contained in K_p if it fulfills the following constraints restricting the set of potential cross-links to the most promising:

(C_1) k and p are images of distant pairs respectively triplets, i.e., $d_{PG}(P, P_i) \geq D_{PG}$ with $p \in P$ and $k \in P_i$.

(C_2) k and p overlap forming a stable pair, i.e., $D'_E(p) < d_E(p, k) \leq D^p_E$ and $\angle(p, k) < 120°$.

The constraint (C_1) ensures cross-links between distant triplets. The graph distance $d_{PG}(P, P_i)$ between pairs P and P_i can be determined by breadth-first search in the PG. The minimum required distance is given by a threshold D_{PG} which depends on the intended purpose. Larger distances are suitable for loop closing avoiding dense blocks. On the other hand, shorter distances allows for a better stabilization of the block. We use $D_{PG} = 5$, preferring the later.

Number of Images	500	1000	1500
OD 1xGPU / 4xGPU	1.3 / 0.4	4.9 / 1.7	10.7 / 3.7
ED 1xCPU / 4xCPU	1.3 / 0.2	4.9 / 0.8	11.4 / 1.8

Table 1. Comparison of the image similarity estimation runtime in minutes using the original and the embedded descriptors. The original real descriptors (OD) were matched using GPU (Wu, 2007) and the embedded binary descriptors (ED) using CPU with one and four GPUs/CPU cores.

Images with overlapping regions are enforced by constraint (C_2), where d_E corresponds to the Euclidean distance and $\angle$ to the angle between the view directions between the cameras corresponding to the images. These are given by the estimated camera poses resulting from the hierarchical triplet merging stage (without bundle adjustment). To ensure that only potentially stable configurations are added, a minimum Euclidean distance $D'_E(p)$ is included which is the shortest baseline of a pair containing image p in the block.

The determination of the search radius D^p_E as well as the neighbors inside them is accelerated by a kd-tree. The choice of D^p_E is critical because due to the drift in long sequences of images the distances between actually close images might become rather large. Thus, we determine the $2N_p + 5$ nearest neighbors and use the distance to the most distant neighbor for D^p_E. N_p gives the number of triplets which contain the image p. If p is contained in N_p triplets, then it has at most $2N_p$ direct neighbors which are in the same triplet as p. We take 5 more neighbors to increase the search radius to consider also not direct neighbors.

Each image $k \in K_p$ forms together with the pair P_i a potential cross-link. We sort the cross-links according to the image similarities to increase the probability of establishing a correspondence. Finally, sorted cross-links are iteratively added to the block if they fulfill the constraint (C_1) in the meshed block.

5 RESULTS

We demonstrate the potential of our approach concerning wide and weak baselines as well as a larger number of images for three image sets. No additional information except (an approximate) camera calibration is used. We give the accuracy in terms of the mean reprojection error in pixels. As the final merging is done according to (Mayer, 2014), for which it has been shown that the obtained accuracy complies statistically with the differences between individual runs of the complete system, the obtained accuracy estimates can be regarded as repeatable. All reported results have been obtained on a system with an Intel Xeon E7-8870 deca-core CPU and an NVIDIA GTX 690 dual-GPU. For results in Table 1 a second NVIDIA GTX 690 has been used.

For a qualitative visual evaluation of the descriptor embedding, the image similarity matrices for the image set *House* of Figure 3 are compared in Figure 4. The similarity matrices are created using full pairwise matching of the original and the embedded descriptors, respectively. In general, using the embedded descriptors the similarities are slightly overestimated in comparison to the original descriptors, but the relative similarities between images are mostly preserved allowing for a reliable ranking. This is probably due to the high dimensionality of the embedded SIFT descriptors.

In addition, matching of the embedded descriptors requires only a fraction of the time needed for the matching of the original descriptors, even if GPU acceleration (Wu, 2007) is used for the latter. Table 1 gives an overview of the image similarity estimation runtimes using the original and the embedded descriptors. While the runtime is approximately the same if one CPU core or one GPU is employed, using more CPU cores reduces the runtime considerably. This scalability on the CPU together with todays multi-core CPUs, low memory consumption and a simpler implementation make the binary descriptors such an attractive choice. However, matching of embedded descriptors provides only a constant speedup and, thus, cannot achieve the scalability of vocabulary trees. On the other hand side, it is parameter-free, requires no training phase and achieves a good performance.

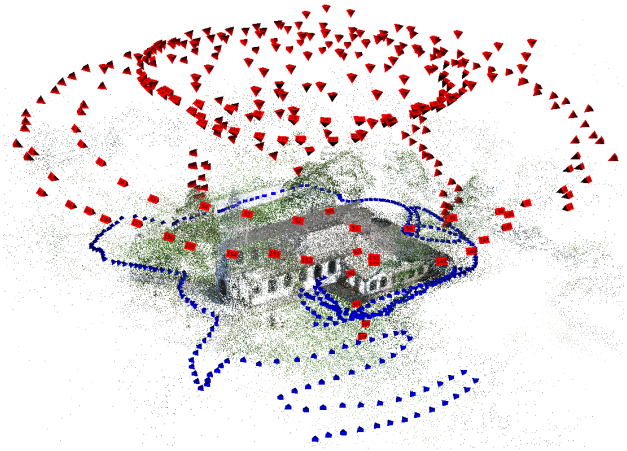


Figure 5. Image set *Church* consisting of 657 terrestrial and aerial images. The pyramids visualize the obtained camera poses.

The capability of the proposed approach to handle wide baselines is demonstrated for the image set *House* (Figure 3). The set consists of terrestrial and aerial images with wide baselines between them. Figure 3(b) presents a triplet which connects terrestrial with aerial images from a camera mounted on an unmanned aerial vehicle (UAV) and contains two pairs with a significantly wide baseline. Nevertheless, all images could be linked into a single block.

In Figure 5, the registration of the image set *Church* is presented. The image set contains a couple of pairs with weak baselines as well as a combination of UAV and terrestrial images. The challenge lies in filtering the weak pairs in order to exclude them from image linking. The block construction stage produces two blocks which were linked to a complete block during the block linking stage.

Estimated camera poses of the image set *Village* with 3531 images are given in Figure 7(a). Image acquisition occurred at very different points in time, thus, severe radiometric distortions exist between images. Block construction yields twenty blocks which are successfully linked in the subsequent stage into a single block. In comparison, VisualSFM could not register the images correctly as shown in Figure 7(b).

The runtimes of each stage of the SfM pipeline as well as VisualSFM are presented in Table 6. The dominating stage is the hierarchical triplet merging, especially the last merging step as well as the final bundle adjustment, which are not yet parallelized. VisualSFM (Version 0.5.26) was executed without graphical interface, with default parameters and option *sfm*. Compared to VisualSFM, our SfM pipeline has a lower runtime and a better accuracy for the larger image sets. This is particularly significant as our approach does not use GPU acceleration, except for feature extraction, and employs triplets for image linking instead of only pairs. Triplets improve the overall accuracy, but also increase the runtime.

Image Set	#Images	Image Preprocessing	Similarity Estimation	Block Constr.	Block Linking	Block Meshing	Hierarchical Triplet Merging	Total	σ_0
House	59	0.01	0	0.05	0.03	0	0.01	0.11	0.22
Church	657	0.04	0.01	0.16	0.04	0.01	0.33	0.59	0.39
Village	3531	0.25	0.20	0.58	1.51	0.89	9.7	13.13	0.30

Image Set	Feature Extraction	Image Matching	Sparse Reconstruction	Total	σ_0
House	0.01	0.02	0.01	0.04	0.51
Church	0.09	2.42	0.12	2.63	1.09
Village	0.52	84.88	1.24	86.64	1.35

Figure 6. Runtimes in hours for each stage of the proposed SfM approach (top) and VisualSFM (bottom) as well as the achieved accuracy (σ_0) in pixels for the presented image sets.

6 CONCLUSION

In this paper, an automatic SfM approach for unordered image sets with complex configurations is presented. We have demonstrated its robustness concerning wide as well as weak baselines and the capability to efficiently and completely handle complex image sets of moderate size. Apart from (approximate) camera calibration no other information is used.

Our first contribution consists of an unsupervised yet powerful descriptor embedding used for fast image similarity ranking. The latter is employed to significantly reduce the number of complex, wide baseline image matching operations. As core contribution, an iterative graph-based method is proposed which is based on line graphs allowing to formulate efficient image linking as the search for the terminal Steiner minimum tree. To our knowledge, this is the first application of a Steiner tree in the context of SfM. Robustness and accuracy are improved by applying subsequent meshing allowing for implicit loop closing. By this means, an accurate, efficient and complete pose estimation of complex image sets is achieved.

References

Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M. and Szeliski, R., 2009. Building Rome in a Day. In: ICCV.

Beder, C. and Steffen, R., 2006. Determining an Initial Image Pair for Fixing the Scale of a 3D Reconstruction from an Image Sequence. In: DAGM.

Cai, H., Mikolajczyk, K. and Matas, J., 2011. Learning Linear Discriminant Projections for Dimensionality Reduction of Image Descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(2), pp. 338–352.

Chen, Y., 2011. An Improved Approximation Algorithm for the Terminal Steiner Tree Problem. In: Computational Science and Its Applications - ICCSA 2011, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 141–151.

Cheng, J., Leng, C., Wu, J., Cui, H. and Lu, H., 2014. Fast and Accurate Image Matching with Cascade Hashing for 3D Reconstruction. In: CVPR.

Crandall, D., Owens, A., Snavely, N. and Huttenlocher, D., 2011. Discrete-Continuous Optimization for Large-scale Structure from Motion. In: CVPR.

Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S. and Pollefeys, M., 2009. Building Rome on a Cloudless Day. In: ECCV.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J. and Stahel, W. A., 2011. Robust Statistics: The Approach Based on Influence Functions. John Wiley & Sons, Inc.

Hartmann, W., Havlena, M. and Schindler, K., 2015. Recent Developments in Large-scale Tie-point Matching. ISPRS Journal of Photogrammetry and Remote Sensing pp. 47–62.

Havlena, M. and Schindler, K., 2014. VocMatch: Efficient Multiview Correspondence for Structure from Motion. In: ECCV.

Havlena, M., Hartmann, W. and Schindler, K., 2013. Optimal Reduction of Large Image Databases for Location Recognition. In: ICCV Workshop.

Havlena, M., Torii, A. and Pajdla, T., 2010. Efficient Structure from Motion by Graph Optimization. In: ECCV.

Heinly, J., Schonberger, J., Dunn, E. and Frahm, J.-M., 2015. Reconstructing the World* in Six Days. In: CVPR.

Jaccard, P., 1912. The Distribution of the Flora in the Alpine Zone. New Phytologist 11(2), pp. 37–50.

Jegou, H., Douze, M. and Schmid, C., 2008. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In: ECCV.

Ke, Y. and Sukthankar, R., 2004. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In: CVPR.

Klopschitz, M., Irschara, A., Reitmayr, G. and Schmalstieg, D., 2010. Robust Incremental Structure from Motion. In: 3DPVT.

Levandowsky, M. and Winter, D., 1971. Distance between Sets. Nature 234(5), pp. 34–35.

Li, X., Wu, C., Zach, C., Lazebnik, S. and Frahm, J.-M., 2008. Modeling and Recognition of Landmark Image Collections using Iconic Scene Graphs. In: ECCV.

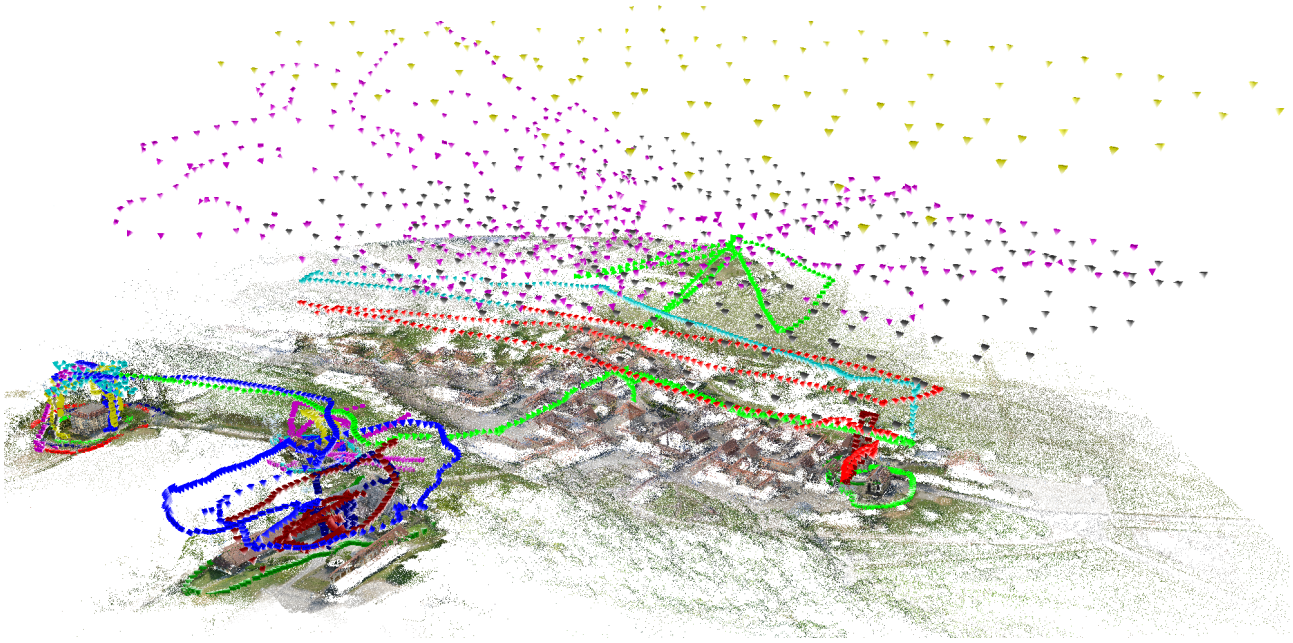
Liben-Nowell, D. and Kleinberg, J., 2003. The Link Prediction Problem for Social Networks. In: 12th International Conference on Information and Knowledge Management, CIKM, ACM, pp. 556–559.

Lin, G. and Xue, G., 2002. On the Terminal Steiner Tree Problem. Information Processing Letters 84(2), pp. 103–107.

Lowe, D. G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2), pp. 91–110.

Mayer, H., 2014. Efficient Hierarchical Triplet Merging for Camera Pose Estimation. In: GCPR.

Mayer, H., Bartelsen, J., Hirschmüller, H. and Kuhn, A., 2012. Dense 3D Reconstruction from Wide Baseline Image Sets. In: Outdoor and Large-Scale Real-World Scene Analysis, Lecture Notes in Computer Science, Vol. 7474, Springer-Verlag, pp. 285–304.



(a) Wide Baseline SfM

(b) VisualSFM

Figure 7. Image set *Village* containing 3531 images acquired from the ground and from various Unmanned Aerial Systems. Reconstructed camera poses are visualized by colored pyramids, where each color represents a different camera type. Registration obtained using the proposed SfM approach is shown in (a), whereas (b) shows the erroneous reconstruction produced by VisualSFM.

Moulon, P., Monasse, P. and Marlet, R., 2013. Global Fusion of Relative Motions for Robust, Accurate and Scalable Structure from Motion. In: ICCV.

Nister, D. and Stewenius, H., 2006. Scalable Recognition with a Vocabulary Tree. In: CVPR.

Oliva, A. and Torralba, A., 2001. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. International Journal of Computer Vision 42(3), pp. 145–175.

Philbin, J. and Zisserman, A., 2008. Object Mining using a Matching Graph on Very Large Image Collections. In: 6th Indian Conference on Computer Vision, Graphics and Image Processing.

Quack, T., Leibe, B. and Van Gool, L., 2008. World-scale Mining of Objects and Events from Community Photo Collections. In: Conference on Content-based Image and Video Retrieval.

Raginsky, M. and Lazebnik, S., 2009. Locality-sensitive Binary Codes from Shift-invariant Kernels. In: Advances in Neural Information Processing Systems 22, Curran Associates, Inc., pp. 1509–1517.

Raguram, R., Tighe, J. and Frahm, J.-M., 2012. Improved Geometric Verification for Large Scale Landmark Image Collections. In: BMVC.

Schönberger, J., Berg, A. and Frahm, J.-M., 2015. Efficient Two-View Geometry Classification. In: GSPR.

Simon, I., Snavely, N. and Seitz, S., 2007. Scene Summarization for Online Image Collections. In: ICCV.

Sivic, J. and Zisserman, A., 2003. Video Google: A Text Retrieval Approach to Object Matching in Videos. In: ICCV.

Snavely, N., Seitz, S. M. and Szeliski, R., 2006. Photo Tourism: Exploring Photo Collections in 3D. In: SIGGRAPH.

Snavely, N., Seitz, S. M. and Szeliski, R., 2008. Skeletal Graphs for Efficient Structure from Motion. In: CVPR.

Strecha, C., Bronstein, A., Bronstein, M. and Fua, P., 2012. LDAHash: Improved Matching with Smaller Descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(1), pp. 66–78.

Torralba, A., Fergus, R. and Weiss, Y., 2008. Small Codes and Large Image Databases for Recognition. In: CVPR.

Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A., 1999. Bundle Adjustment – A Modern Synthesis. In: International Workshop on Vision Algorithms: Theory and Practice, Springer-Verlag, pp. 298–372.

Wu, C., 2007. SiftGPU: A GPU Implementation of Scale Invariant Feature Transform (SIFT). `http://www.cs.unc.edu/~ccwu/siftgpu`.

Wu, C., 2011. VisualSFM: A Visual Structure from Motion System. `http://ccwu.me/vsfm/`.