ISPRS Journal of Photogrammetry and Remote Sensing 166 (2020) 140–152

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Structure from motion for complex image sets

Mario Michelini[*], Helmut Mayer

Institute for Applied Computer Science, Bundeswehr University Munich, 85577 Neubiberg, Germany

ARTICLE INFO

Keywords:
Structure from motion
Camera orientation
Wide/weak baseline
Graph
Hierarchical
3D reconstruction

ABSTRACT

This paper presents an approach for Structure from Motion (SfM) for unorganized complex image sets. To achieve high accuracy and robustness, image triplets are employed and an (approximate) internal camera calibration is assumed to be known. The complexity of an image set is determined by the camera configurations which may include wide as well as weak baselines.

Wide baselines occur for instance when terrestrial images and images from small Unmanned Aerial Systems (UAS) are combined. The resulting large (geometric/radiometric) distortions between images make image matching difficult possibly leading to an incomplete result. Weak baselines mean an insufficient distance between cameras compared to the distance of the observed scene and give rise to critical camera configurations. Inappropriate handling of such configurations may lead to various problems in triangulation-based SfM up to total failure.

The focus of our approach lies on a complete linking of images even in case of wide or weak baselines. We do not rely on any additional information such as camera configurations, Global Positioning System (GPS) or an Inertial Navigation System (INS). As basis for generating suitable triplets to link the images, an iterative graph-based method is employed formulating image linking as the search for a terminal Steiner minimum tree in the line graph. SIFT (Lowe, 2004) descriptors are embedded into Hamming space for fast image similarity ranking. This is employed to limit the number of pairs to be geometrically verified by a computationally and more complex wide baseline matching method (Mayer et al., 2012). Critical camera configurations which are not suitable for geometric verification are detected by means of classification (Michelini and Mayer, 2019). Additionally, we propose a graph-based approach for the optimization of the hierarchical merging of triplets to efficiently generate larger image subsets.

By this means, a complete, 3D reconstruction of the scene is obtained. Experiments demonstrate that the approach is able to produce reliable orientation for large image sets comprising wide as well as weak baseline configurations.

1. Introduction

Recent developments for Structure from Motion (SfM) or sparse 3D reconstruction techniques from unorganized image sets focus on large (Internet) photo collections (Heinly et al., 2015; Crandall et al., 2011; Frahm et al., 2010; Havlena et al., 2010; Agarwal et al., 2009; Snavely et al., 2008). They can contain millions of images, yet, often comprising a very high redundancy and moderate baselines. In contrast to these large photo collections, we focus on image sets with up to a few thousand images, but containing complex camera configurations comprising wide as well as weak baselines between images.

In our case, wide baselines often arise when terrestrial images and imagery taken from small Unmanned Aerial Systems (UAS) are combined. Failure to handle wide baselines can lead to incomplete SfM resulting in multiple disconnected reconstructions. On the other hand, weak baselines occur if the translation between image acquisitions is insufficient in relation to the distance to the observed scene. Such camera configurations are termed *critical* because they lead to a poor intersection geometry, which becomes undefined in case of zero baseline (i.e., pure rotation). Thus, an inappropriate handling of weak baselines can result in inaccurate or even failing orientation estimation and sparse 3D reconstruction.

A crucial step during SfM is the merging of image pairs or triplets with consistent geometry into larger image subsets with a common reference frame. The result is usually optimized by means of bundle adjustment (Triggs et al., 2000), which is a computationally intensive non-linear optimization method. Hence, hierarchical merging techniques (Toldo et al., 2015; Mayer, 2014; Gherardi et al., 2010; Farenzena

[*] Corresponding author.
E-mail addresses: mario.michelini@unibw.de (M. Michelini), helmut.mayer@unibw.de (H. Mayer).

https://doi.org/10.1016/j.isprsjprs.2020.05.020
Received 13 February 2020; Received in revised form 25 May 2020; Accepted 25 May 2020

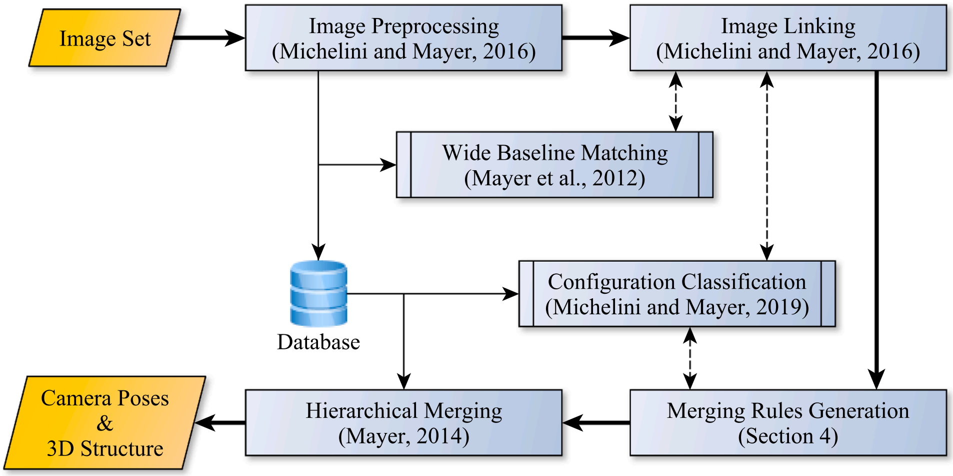


Image Set
Image Preprocessing (Michelini and Mayer, 2016)
Image Linking (Michelini and Mayer, 2016)
Wide Baseline Matching (Mayer et al., 2012)
Database
Configuration Classification (Michelini and Mayer, 2019)
Camera Poses & 3D Structure
Hierarchical Merging (Mayer, 2014)
Merging Rules Generation (Section 4)

Fig. 1. Structure from Motion Pipeline.

et al., 2009; Fitzgibbon and Zisserman, 1998) are applied to improve the efficiency. This way, disjoint image subsets can be merged independently and, thus, in parallel, improving the runtime on systems with multiple parallel processing units.

The aim of our approach is an efficient, complete and reliable linking of the entire image set even for complex configurations comprising wide as well as weak baselines. No additional information like camera configuration, Global Positioning System (GPS) or Inertial Navigation System (INS) is used. An overview of the processing stages is given in Fig. 1.

Input is an (unorganized) image set with an (approximate) internal camera calibration. Based on this, we start with image preprocessing where multi-resolution image pyramids are generated and SIFT features (Lowe, 2004) are extracted using GPU-acceleration (Wu, 2012). In the next stage, a graph-based method (Michelini and Mayer, 2016) establishes the links between the images (Section 3) employing a classification-based approach (Michelini and Mayer, 2019) for the detection of critical camera configurations. The wide baseline method (Mayer et al., 2012) is used for geometric verifications providing robustness even in case of large radiometric or geometric image distortions. Based on image linkage, a novel graph-based optimization strategy improves the efficiency of the subsequent hierarchical merging of image subsets (Section 4). In the last stage, relative camera orientations as well as the sparse 3D structure are determined using hierarchical merging (Mayer, 2014). Data exchange between processing stages is accomplished by means of a database.

Results which demonstrate the potential of our SfM approach on real-world datasets as well as in comparison to other state-of-the-art frameworks are presented in Section 5. Finally, in Section 6 conclusions are given.

2. Related work

Fitzgibbon and Zisserman (1998) as well as Koch et al. (1998) presented pioneering works dealing with SfM in image sequences. Later, Schaffalitzky and Zisserman (2002) have shown, that automatic SfM is achievable for general camera configurations without additional information (e.g., about the sequence). Snavely et al. (2006) introduced the Framework *Photo Tourism*, which can deal with larger image sets and could produce high quality results. However, it has a high runtime due to the employed exhaustive image matching and sequential image merging. An improved version of *Photo Tourism* is known as *Bundler*[1] and is often used for benchmarks to evaluate the achievable quality of other approaches.

Subsequent approaches focused, among other things, on further improvement of the efficiency. Snavely et al. (2008) formulated the problem depending on the scene complexity instead of the number of images. Li et al. (2008) employed clustering to reduce the complexity, Agarwal et al. (2009) used a computer cluster to accelerate the processing, whereas Frahm et al. (2010) have utilized the massive parallelization of graphic cards. Finally, Heinly et al. (2015) have shown, that with an efficient implementation on a highly parallel system, even millions of images can be processed in a few days.

The most commonly used method to reduce the combinatorial complexity is pruning of the image set (Li et al., 2008; Frahm et al., 2010; Philbin and Zisserman, 2008; Havlena et al., 2013). 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 (Nistér and Stewenius, 2006) to reduce the complexity. Acceleration of matching itself using GPU was employed in Wu (2012) and Frahm et al. (2010). Holistic features (Oliva and Torralba, 2001) indexed by compact hashing codes (Raginsky and Lazebnik, 2009; Torralba et al., 2008) were utilized in Frahm et al. (2010) 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.

Relations between images are often described using weighted graphs, where nodes correspond to images and an edge between nodes exists if the corresponding images are related based on the desired geometric properties. The edge weights describe the quality of the relations. For example, this can be the number of correspondences between the images (Schaffalitzky and Zisserman, 2002; Li et al., 2008; Jiang et al., 2013; Toldo et al., 2015) or the uncertainty of the estimated camera orientations (Snavely et al., 2008). In addition, complex hypergraphs[2] were used in Ni and Dellaert (2012).

Usually, undirected weighted graphs are employed for modeling (Schaffalitzky and Zisserman, 2002; Steele and Egbert, 2006; Zach et al., 2008; Philbin and Zisserman, 2008; Zach et al., 2010; Moulon et al., 2013; Toldo et al., 2015). There exist also approaches which use directed graphs (Snavely et al., 2008; Irschara et al., 2011; Wefelscheid, 2013; Wilson and Snavely, 2014) allowing for a more accurate modeling, but requiring a larger effort for the determination of the asymmetric relations.

[1] www.cs.cornell.edu/snavely/bundler.

[2] A *hypergraph* is a generalization of a graph in which an edge, called *hyperedge*, can be incident with any number of nodes.

Methods based on undirected graphs often employ minimal/maximal spanning trees to determine the processing order for images (Schaffalitzky and Zisserman, 2002; Steele and Egbert, 2006; Zach et al., 2008; Klopschitz et al., 2010; Zach et al., 2010; Toldo et al., 2015). In addition to spanning trees, also other graph-theoretic concepts like normalized graph cuts (Li et al., 2008), minimal dominating sets (Havlena et al., 2010), graph spectra (Heath et al., 2010; Kim et al., 2012) or betweenness centrality (Wefelscheid, 2013) were employed.

In the following, we describe our approach for linking images based on their estimated similarities and propose an optimization method for hierarchical merging of image subsets.

3. Image linking

Image linking describes the relations between images and, thus, implies knowledge about their overlap (i.e., the projection of the same parts of a scene). The latter is usually derived from feature correspondences, i.e., by means of *image matching* (Hartmann et al., 2016). Yet, because of the high combinatorial complexity, exhaustive image matching is not practical even for small image sets.

In addition, knowledge about geometric relations between images is in our case purely based on image features. Thus, 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 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 not suitable in our case, because they make the geometric verification faster but less robust against complex configurations which we intend to handle.

A vocabulary tree (Nistér and Stewenius, 2006) is often used to reduce the combinatorial complexity in large image sets. It is a special data structure which scales well, but requires an explicit training and parameter fine tuning (number of clusters and tree depth) to achieve satisfying accuracy and efficiency (Irschara et al., 2011). Instead, we, thus, employ a fast image similarity estimation method described in Section 3.1. That way, potentially overlapping images are filtered based on their estimated similarities and it is sufficient to perform complex geometric verifications by means of wide baseline matching (Mayer et al., 2012) only for a small subset.

To improve robustness and accuracy, image triplets instead of pairs are employed (Moulon et al., 2013; Klopschitz et al., 2010). However, the usage of triplets increases the complexity and, thus, we estimate the geometry for pairs first and derive triplets afterwards based on the information from the pairs. A theoretically well founded modeling for the latter is proposed in Sections 3.2 and 3.3 allowing for efficient sparse image linking. Finally, the density of image links is increased by means of loop closing described in Section 3.4 to further improve the stability of SfM.

3.1. Image similarity estimation

Image linking is based on pairwise relations between images. However, an accurate estimation of these relations is extremely time-consuming for larger image sets due to the quadratic complexity. Even acceleration techniques like (Wu, 2012) which utilize parallel processing on modern graphic cards, scale insufficiently.

We estimate similarities between images by matching their SIFT features (Lowe, 2004) employing the Jaccard index (Jaccard, 1912) as a relative similarity score to rank the images. The Jaccard index $\mathrm{J}(i, j)$ is defined as

$$\mathrm{J}\left(i, j\right) = \frac{|M_i \cap M_j|}{|M_i \cup M_j|}, \tag{1}$$

where M_i and M_j are the feature sets of the images i and j and $M_i \cap M_j$ the resulting correspondence set. It is later used in Section 3.3 to decide what images are to be linked.

Because we are only interested in relative similarities, a simplified matching can be used as long as the approximation errors are small or distributed evenly. To this end, a matching based on feature descriptors embedded from real space $\mathbb{R}^{128}$ into Hamming space $\mathbb{H}^{128} = \{0, 1\}^{128}$ is employed. This allows an efficient comparison using the Hamming distance

$$\mathrm{d_H}\left(\boldsymbol{u}, \boldsymbol{v}\right) = \sum_{i=1}^{128} \left[\left(u_i \circ v_i \right) := \begin{cases} u_i \neq v_i, & 1 \\ u_i = v_i, & 0 \end{cases} \right], \tag{2}$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ are SIFT descriptors with their elements u_i and v_i. The XOR bit operation $\circ$ is followed by bit counting to determine the number of different bit positions. Modern processors have integrated accelerated functions (Streaming SIMD Extensions – SSE) for both operations allowing for a very efficient comparison.

Descriptor embedding itself is based on the concept of orthants, which are the generalization of quadrants in two-dimensional to multidimensional space. A 128-dimensional descriptor space $\mathbb{R}^{128}$ can be partitioned by 128 independent (affine) hyperplanes in $\mathbb{R}^{128}$. Each hyperplane goes through the intersection point $\mathbf{p} \in \mathbb{R}^{128}$ and separates $\mathbb{R}^{128}$ into two half-spaces, termed the positive and the negative half-space. For the embedding one needs to define the 128 hyperplanes. The values of the normalized descriptor lie in the range 0 to 1.

The origin $\mathbf{p}$ is not a good choice for the intersection point $\mathbf{p} = \mathbf{0}$ of the hyperplanes and, thus, 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 $\mathbf{p}$.

Intersections of 128 mutually orthogonal half-spaces then determine 2^{128} orthants. Every orthant is determined by a sequence of 128 plus or minus signs, where the ith sign indicates whether the orthant is in the positive or negative half-space of the ith hyperplane. Thus, an orthant in $\mathbb{R}^{128}$ corresponding to the SIFT descriptor can be represented by a compact bit vector of length 128.

Matching of the embedded descriptors then means determining the number of corresponding half-spaces, instead of computing the Euclidean distance or cosine similarity in case of the original descriptors. On one hand, this provides only a rough approximation of the true correspondences. On the other hand, the comparison of embedded descriptors reduces the matching runtime drastically allowing for an exhaustive matching and, thus, an accuracy improvement. In addition, even more accurate image similarities only provide a limited benefit due to other relevant factors like image overlap, feature distribution or intersection geometry. Hence, the proposed image similarity estimation provides an efficient and meaningful way to reduce the number of expensive geometric verifications.

3.2. Modeling of relations between images

We describe relations between images by an undirected weighted *image graph* (IG), where nodes correspond to images and edges connect pairs of images that overlap. Edge weights and potential overlap are determined by the image similarities estimated using the approach described in the previous section.

Because our image linking is based on triplets, higher order relationships are required which the image graph with its pairwise relationships is lacking. In addition, we use pairs for geometry propagation throughout linking (see Section 4), meaning that linkable triplets must have two images in common. An image graph cannot model this constraint either. Thus, we employ a modeling based on the line graph of the image graph which can describe linking using triplets and allows for geometry propagation via pairs.

The *line graph* $L(G)$ of an undirected graph G has as set of nodes the

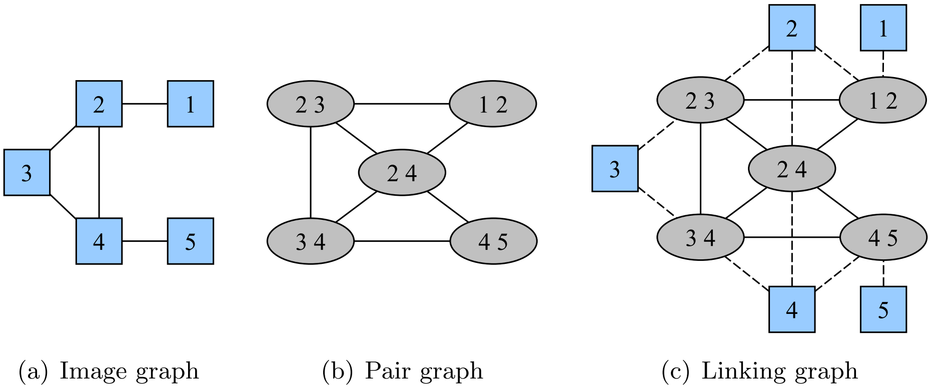


Fig. 2. Image graph with corresponding pair and linking graph. The pair graph corresponds to the line graph of the image graph describing relations between image pairs. Adding nodes of the image graph to the pair graph results in the linking graph, where rectangles represent the image and ellipses the pair nodes.

edges of G. Two nodes in $L(G)$ are adjacent iff they have exactly one node of G in common. 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

$$A_{L(G)} = R_G^T R_G - 2I \tag{3}$$

with I the identity matrix. Each node i in G with degree d_i generates d_i nodes as well as $\binom{d_i}{2}$ connections in the line graph $L(G)$. It follows, that the cardinality of the edge set of $L(G)$ is highly related to the density of G.

The line graph of the image graph *IG* is termed *pair graph* (PG), because it contains nodes corresponding to pairs of overlapping images. Two nodes in PG are adjacent if their corresponding pairs have an image with an overlapping region in common, yielding a triplet. By this means, a traversal through PG implicitly corresponds to linking triplets using pairs for geometry propagation.

By extending PG to explicitly represent the images, the *linking graph* (LG) is constructed. It comprises two node types corresponding to the nodes of IG (*image nodes*) and PG (*pair nodes*), as shown in Fig. 2. An image and a pair node are adjacent if the pair node contains the image corresponding to the image node. Because the image nodes are only required to model the image linking, there exist no edges between them.

An unweighted LG models potential links in the form of triplets between the images. The quality of those links can be described by weighting its edges. The quality of a particular link depends on the quality of the corresponding triplet, whose quality is in turn determined by the quality of the pairs it consists of. Thus, the weighting of the LG aims at selecting suitable links for reliable image linking based solely on the information about the pairs.

The roundness $\mathrm{R}(\boldsymbol{X})$ of a reconstructed 3D point $\boldsymbol{X}$ has been proposed in Beder and Steffen (2006) as the basis to measure the stability of a pair P. For each reconstructed 3D point the eigenvalues λ_1^x, λ_2^x and λ_3^x of the corresponding covariance matrix are computed. The roundness is then defined by

$$\mathrm{R}(\boldsymbol{X}) = \sqrt{\frac{\lambda_3^x}{\lambda_1^x}} \tag{4}$$

with $\lambda_1^x \geqslant \lambda_2^x \geqslant \lambda_3^x$. It lies between 0 and 1 and depends only on the relative geometry between cameras and image features. For coincident camera centers and correct feature correspondences, it is equal to zero.

Beder and Steffen (2006) have defined the stability of P by the mean roundness over the set of all reconstructed 3D points X_P of the pair. Unfortunately, a pair with high stability can comprise only a few correspondences, which may be insufficient for triplet construction. This can result from a wide baseline between the images, implying a good intersection geometry, but only very few correspondences (e.g., due to image distortions or occlusions). On the other hand, the number of correspondences alone is also not a suitable quality score, because high numbers can arise for critical configurations (cf. Section 1). However, the combination of both provides a quality score

$$\mathrm{Q}(X_P) = \sum_{\boldsymbol{X} \in X_P} \mathrm{R}(\boldsymbol{X}), \tag{5}$$

which incorporates the number of correspondences as well as their quality. By this means, a large image overlap can be enforced together with a stable camera geometry.

Besides the number of correspondences, their distribution across the images is important as well. A direct incorporation into the quality score is difficult due to the employed geometrical verification which uses different image resolutions for pairs and triplets. Therefore, we incorporate the feature distribution indirectly by considering only features which are contained in the threefold overlap area of the triplet's images.

Let F_b be the feature set of an image b. The convex hull $\mathrm{H}(F_b)$ around the features of F_b describes the area in b covered by them. However, this is not a robust measure of coverage because any extreme feature point, i.e., close to the corners of the image, yields a large area. Thus, we filter these extreme points out by iteratively determining the convex hull and removing points of the hull if there are less than three other points in their neighborhood. The latter is defined by the median distance between points. The result is a (reduced) convex hull $\mathrm{H}^*(F_b)$ providing a robust coverage measure.

The feature correspondences of a pair P form the feature set F_P consisting of features in both images and $\mathrm{H}^*(F_P \cap F_b)$ the overlapping region in image $b \in P$ with respect to P. The correspondences of pairs from a pair set Ω define the overlapping area

$$\mathrm{O}(b|\Omega) = \bigcap_{P \in \Omega \wedge P \ni b} \mathrm{H}^*(F_P \cap F_b) \tag{6}$$

in image $b \in P$ with $P \in \Omega$. Let e be an edge between pair nodes u and v in the LG corresponding to pairs P_u and P_v. e itself corresponds to the triplet $T = (P_u \cup P_v \cup P_e)$, the common image of P_u and P_v, or the pair $P_e = (P_u \cup P_v)\setminus(P_u \cap P_v)$. The weight function $\omega(e)$ for an edge e between the pair nodes is defined as

$$\omega(e) = \min[\mathrm{Q}(\chi(F_{P_u}^3)), \mathrm{Q}(\chi(F_{P_v}^3))], \tag{7}$$

where

$$F_P^3 = \{\boldsymbol{x} \;\middle|\; \boldsymbol{x} \in F_P \;\wedge\; \boldsymbol{x} \in \bigcup_{b \in P} \mathrm{O}(b|\{P_u, P_v, P_e\})\} \tag{8}$$

are feature correspondences of a pair P corresponding to the threefold correspondences of T and

$$\chi: \mathbb{R}^2 \rightarrow \mathbb{R}^3 \tag{9}$$

is the mapping of 2D feature points to the corresponding (reconstructed) 3D points of the pair. This way, only the quality of pairs P_u

and P_v is considered due to their primary influence on the quality of T. In order to incorporate the overlap between all images of a triplet, the relevant 3D points of P_u and P_v are restricted to the threefold overlap area of the triplet's images.

Edges incident with the image nodes contain no additional information, but can be used for the selection of appropriate pair nodes. Thus, we weight them using the quality Q_P from Eq. (5) corresponding to the pair node connected to the image node.

We employ the weighted LG to describe the linking between the images including the link quality. However, the LG can contain many redundant links, which can also be of varying stability concerning orientation estimation. The latter means, that triplets used for linking can be more or less stable, e.g., due to short baselines between their images. Thus, using all triplets would increase the runtime without a significant benefit and may even have a negative influence on the stability of the final result.

For this reason, we introduce the concept of a *block* as a linking subgraph containing stable links employed for the hierarchical merging (Section 4). Here, *stability* means robustness against poor intersection geometry affecting the quality of orientation estimation. The *size* of a block specifies the number of images linked, where a block is termed *complete* if it links all images of the image set. The block *density* is the number of triplets used for linking.

3.3. Determination of image links

The IG is constructed based on the estimated image similarities (Section 3.1). Thus, an edge between nodes exists only if the corresponding images have a sufficiently high similarity score given by the Jaccard index.

The LG can be directly constructed from the IG. However, this would result in a very dense LG unnecessarily increasing the complexity. Thus, only a subset of the most promising pairs corresponding to the edges of the maximum spanning tree (MST) in the IG is employed. To ensure their geometric consistency, selected pairs are verified using the wide baseline matching method (Mayer et al., 2012) and then classified using a random forest (Michelini and Mayer, 2019) in order to detect critical configurations (cf. Section 1). Pairs with inconsistent geometry or classified as unstable are removed from IG and the construction of a new MST is initiated. Finally, the LG is constructed by deriving the line graph from the MST.

Having a weighted LG, we can determine a block of minimum density to link the images. This is 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 $V \backslash R$ are termed *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 LG the image nodes correspond to the terminals and the pair nodes to the Steiner nodes.

The terminal Steiner minimum tree of the LG determines the block of minimum density. As the terminal Steiner problem has been shown to be NP-complete (Lin and Xue, 2002), an approximation (Chen, 2011) is employed. The geometric consistency of triplets selected by a block is again verified using the wide baseline matching method (Mayer et al., 2012). Inconsistent triplets are removed from the LG and the construction of a new terminal Steiner minimum tree is initiated.

Missing triplets may lead to the construction of multiple incomplete blocks. This is because a block described by an LG depends on the presence of triplets sharing two images. This requirement can be relaxed after the block construction to two arbitrary images which do not form an instable pair. This way, two incomplete blocks comprising any two common images forming a suitable pair can be merged into a single block. By applying this procedure recursively, a complete block is constructed in the optimal case.

3.4. Loop closing

An *image sequence* consists of sequentially arranged images with pairwise links in between. Images at the ends of the sequence are termed *end images*. If these overlap, the sequence forms an *image loop*. *Loop closing* is a method for detecting and linking end images in order to close image loops.

Inaccuracies in 3D points usually arise due to inexact feature localization as well as approximate intrinsic camera parameters. Their accumulation during the merging of image subsets (Section 4) may lead to a significant deviation of the estimated orientations compared to ground-truth (Steedly et al., 2003). The longer the image sequence, the larger the magnitude of the deviation. Thus, an appropriate loop closing must be applied to ensure a reliable SfM.

Unfortunately, the pair graph of a block (Section 3.2) is a tree, which is not able to implicitly model closed image loops. Hence, we employ the graph structure of a block together with (roughly) estimated camera orientations to efficiently search for end images and to close the loops.

Closing loops in short image sequences leads to no significant improvements (Repko and Pollefeys, 2005), but increases the time and effort for the determination of necessary image links. For this reason, we introduce a threshold l_{min} specifying the minimum length of a closable image sequence. The length of an image sequence l_s with end images b_1 and b_2 is related to the length of the path l_p between the image nodes corresponding to b_1 and b_2. The relation is given by $l_s = l_p - 2$ taking the two edges between image and pair nodes into account.

The detection of a potential image loop is formulated as the search for overlapping end images of an image sequence. In order to be able to detect loops even in complex image sets, each image is considered as potential end image. Starting from an end image b_1

$$B_{b_1} = \{b \mid \mathrm{l}(b_1, b) \geqslant \rho l_{min} \ \wedge\ \sphericalangle(b_1, b) \leqslant \alpha \ \wedge\ b \in \Psi_{b_1}\} \tag{10}$$

with

$$\Psi_{b_1} = \{b \mid \mathrm{d}(b_1, b) \leqslant d_{b_1} \ \vee\ \mathrm{s}(b_1, b) \geqslant s_{b_1}\} \tag{11}$$

gives the set of images suitable as the second end image b_2, which together with b_1 determines a sequence and a potential image loop. $\mathrm{l}(b_1, b)$ is the length of the sequence between b_1 and b, l_{min} the minimum length and $\rho \geqslant 1$ a scale factor described below. $\sphericalangle$ specifies the view direction angle difference of the cameras, d the Euclidean distance and s the estimated image similarity (cf. Section 3.1) between b_1 and b. Finally, an image $b \in B^*_{b_1} \subseteq B_{b_1}$ is considered as the most suitable end image b_2 if it is spatially close to b_1 and forms the longest sequence with it:

$$B^*_{b_1} = \underset{b \in B_{b_1}}{\arg\max}\ \mathrm{l}(b_1, b) \quad \text{and} \quad b_2 = \underset{b \in B^*_{b_1}}{\arg\min}\ \mathrm{d}(b_1, b) \tag{12}$$

Eq. (10) comprises a spatial and a similarity search. The former increases the probability of overlap employing the (roughly) known camera orientations. They are estimated efficiently using hierarchical merging of image subsets (Section 4) without the time-consuming bundle adjustment. Based on it, $\mathrm{d}(b_1, b)$ in Eq. (11) gives the Euclidean distance between the camera positions of images b_1 and b and $\sphericalangle$ their view angle difference. The restriction of the latter using the threshold α especially serves to exclude images with opposite viewing directions. The threshold d_{b_1} ensures spatial proximity. Due to its dependency on the environment of b_1, a uniform threshold cannot be applied. The reasons are varying scales between image subsets as well as camera displacements for which a uniform threshold would select either too many or not enough images.

Therefore, an appropriate environment of b_1 is derived from the paths from the image node corresponding to b_1 to other image nodes in the block. The path length l can be employed as indication for the

Euclidean distance, though a longer path does not necessarily imply a larger Euclidean distance. Restricting the environment of b_1 using the distance threshold l_d, we determine d_{b_1} from the Euclidean distances to images lying on the paths:

$$d_{b_1} = \max_{\forall b: l(b_1,b)\leqslant l_d} d\left(b_1, b\right) \tag{13}$$

To obtain a meaningful value for d_{b_1}, only paths to images inside of the vicinity are to be considered. This is accomplished by the threshold l_d, which, in contrast to d_{b_1}, does not depend on the camera configuration. For $l_d = 1$, d_{b_1} is given by distances to images contained in the same triplet as b_1. In general, the probability of including images from the vicinity decreases with an increasing value of l_d.

Long image sequences may cause large geometric deviations and, thus, erroneously large distances between its end images. In this case, the spatial search is not able to determine end images without the adjustment of the threshold d_{b_1}. However, a higher value of d_{b_1} would unnecessarily increase the complexity of the search. Therefore, we employ image similarities to additionally consider all images with sufficiently high similarity s_{b_1} to b_1 as potential second end images. The threshold s_{b_1} is given by the minimal similarity between images, whose corresponding image nodes are adjacent to the image node corresponding to b_1.

The minimum loop length remains fixed during the spatial search, i.e., $\rho = 1$ in Eq. (10). However, the similarity search requires an adjustment of the scale factor to $\rho > 1$ to restrict the number of image candidates. Because large geometric deviations occur only in long image sequences and the similarity search is designed exclusively for such cases, the increase of the minimum loop length provides a meaningful method for complexity reduction. This way, only similar images, which are far away in the block, are considered as potential end images.

The block may comprise multiple image loops, where some of them could be subsets of larger loops (see Fig. 3). Closing all loops in a large image set with many sub-loops is very time-consuming. However, closing only the largest loops may be insufficient to rectify the geometric deviations. Therefore, we strive to close only sub-loops which are significant for the correction of geometric deviations. Based on the threshold l_{min} the largest independent loops are identified and closed. The number of remaining sub-loops to close is reduced afterwards using the modified weight function

$$\omega^*(l_e) = \exp(-l_e^2 f) \tag{14}$$

for all edges of the block. The function reduces the weight depending on the number of loops l_e an edge e is involved in. By this means, sub-loops of already closed loops become less relevant. The *damping factor* $0 < f \leqslant 1$ is employed to restrict the number of sub-loops, where $f = 0$ means no restriction. In general, the lower the value of f, the more sub-loops are included.

4. Hierarchical merging of image subsets

Starting from known links in the form of triplets, images are merged to larger image subsets transforming the camera orientations into a common reference frame. We employ the hierarchical merging of Mayer (2014), which allows image subsets to grow independently from each other. This offers the opportunity to utilize parallel architectures by performing merging in parallel. However, while the reduction of 3D points with the aim of reducing the merging runtime was the focus in Mayer (2014), here we propose an extension in the form of an optimization strategy to improve the efficiency by better exploiting the parallel architectures.

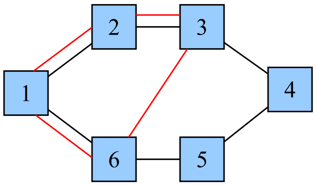


Fig. 3. Image loop comprising the images 1–6 (black lines) and a subloop consisting of images 1, 2, 3 and 6 (red lines). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.1. Hierarchical merging

An image subset (IS) consisting of $n = |\mathrm{IS}|$ images is written as n-IS, where $|\mathrm{IS}|$ specifies the cardinality. Starting with triplets as 3-IS, the first merging step generates 4-IS. In general, image subsets IS_i and IS_j with k_{ij} common images are merged into an $(|\mathrm{IS}_i| + |\mathrm{IS}_j| - k_{ij})$-IS. The transformation into a common reference frame is achieved by applying a (rigid) Euclidean transformation in relation to one IS using two common images to determine the scale factor. Thus, a *necessary condition* for merging two IS is the existence of at least two common images with sufficient baseline, i.e., $k_{ij} \geqslant 2$. A sufficient baseline is ensured using the classification-based approach for the detection of critical camera configurations (Michelini and Mayer, 2019).

After transformation into a common reference frame, 3D points are transferred and the result is optimized by means of robust bundle adjustment (Mayer et al., 2012). The accuracy of the 3D points depends on the number of observations, i.e., the number of images containing the projection of the points (Triggs et al., 2000). Therefore, larger image subsets tend to contain more accurate 3D points than smaller IS. Robust bundle adjustment weights 3D points according to their accuracy, removing insufficiently accurate points. The latter may lead to a loss of essential 3D points in case of merging of IS with a large size difference and, thus, very different accuracies of the 3D points. To avoid such cases, an *additional condition* is introduced demanding up to a certain size, that the merged image subsets should have similar sizes.

4.2. Merging rules

Merging is described by *merging rules*, which contain information about image subsets as well as the image pair used for geometry propagation. The dependencies between rules are described by a directed acyclic graph, termed *rule tree* (see Fig. 4). Rules on the same level in the rule tree are guaranteed to be independent and, thus, a dependency exists only between rules on different levels.

Hierarchical merging itself is modeled by a weighted undirected graph, termed *image subset graph* (ISG). Its nodes correspond to image subsets and an edge between two nodes exists only if image subsets corresponding to incident nodes fulfill the necessary condition from Section 4.1. The weight function $\delta(e)$ for an edge e connecting the nodes u and v is defined as

$$\delta(e) = \underbrace{||\mathrm{IS}_u| - |\mathrm{IS}_v||}_{\delta_d(e)} \underbrace{\mathrm{e}^{\max(c_u,c_v)}}_{\delta_c(e)} = \delta_d(e)\,\delta_c(e), \tag{15}$$

where IS_n is an image subset corresponding to node n. The function $\delta_d(e)$ incorporates the size difference between the image subsets to fulfill the additional constraint given in Section 4.1. A continuous growth of image subsets is ensured by the function $\delta_c(e)$, which exponentially increases the priority (i.e., edge weight) based on the merging status of incident image subsets. The priority depends on the maximum number of levels c_k one of the image subsets has not been involved in a merging operation.

4.3. Generation of merging rules

The generation of merging rules is formulated as a matching problem in the ISG, where a *matching* corresponds to a subset M of pair-

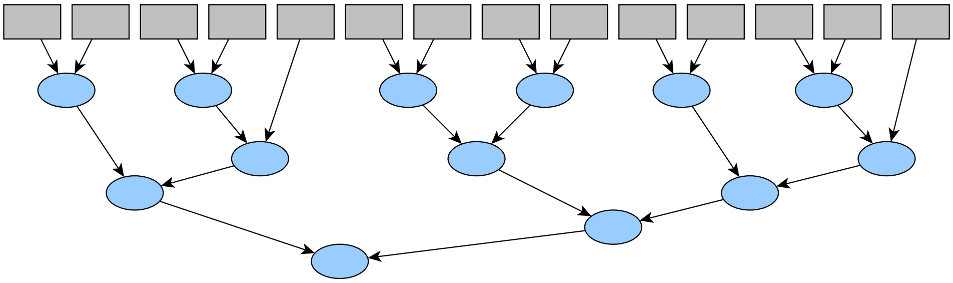
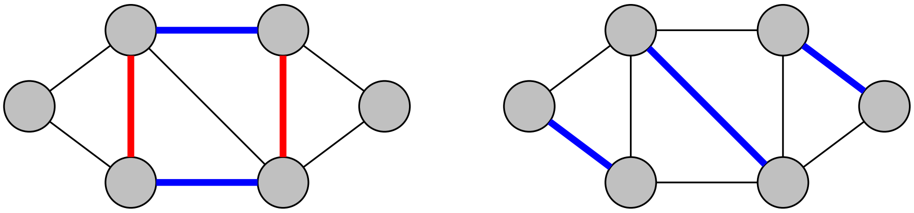


Fig. 4. Rule tree with five levels. Rectangular nodes represents the triplets and ellipses the merging rules.



(a) Two Maximal Matchings

(b) Maximum/Perfect Matching

Fig. 5. Graph Matchings represented by red and blue edges. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
System specification.

Processor (CPU)	2 × Intel® Xeon® E5-2643 v3 (6 cores, 3.40 GHz)
Graphic Card (GPU)	NVIDIA GeForce GTX Titan Z (5 760 CUDA cores, 705 MHz)
Memory	256 GB
Operating System	Windows 10 × 64

Table 2
Image set properties.

Image Set	Size	Cameras	Terrest. Images	UAV Images	Aerial Images
Village Overflight	232	1		×	
UQ St Lucia	351	1	×		
Church	1455	3	×	×	
Monastery	1769	3	×	×	
Settlement	3664	1		×	
Airfield	6210	2		×	
Village	6405	10	×	×	×

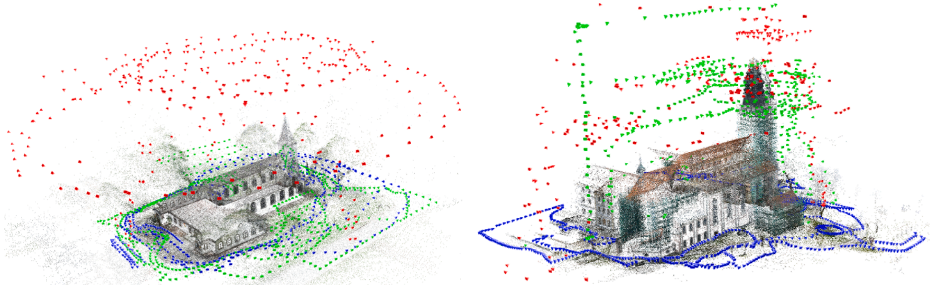
wise non-adjacent edges (see Fig. 5). It is termed *maximal* if no more edges can be added to *M* without violating the matching property. A *maximum* matching has in addition the largest possible cardinality of *M* and, thus, is also maximal. A maximum matching is *perfect* if each node of the ISG is incident with one of the edges in *M*. Finally, a *maximum weighted matching* is defined as a matching, which maximizes the sum of the edges in *M*.

Merging rules correspond to edges of the ISG. Thus, independent rules on a single level of the rule tree can be determined by a matching in the ISG. By contracting edges contained in the matching, a minor of the underlying ISG is formed. By applying matching followed by edge contraction recursively, merging rules are generated and the rule tree is constructed.

Hierarchical merging in Mayer (2014) employs agglomerative hierarchical clustering to generate the merging rules. It starts with triplets as separate clusters of 3-IS and forms larger clusters by maximizing the inter-cluster distances. Merging rules generated in this way correspond to rules generated by search for a maximal matching in the ISG.

However, for the finest granularity perfect matching is required. Unfortunately, perfect matching requires special graph properties which are not always given. Instead, we search for a maximum weighted matching in the ISG. This implies maximal matching and, thus, leads to merging rules which are at least as good as the rules generated by Mayer (2014). In the optimal case, maximum matching is also perfect, providing finer granularity and, hence, better potential for load balancing.

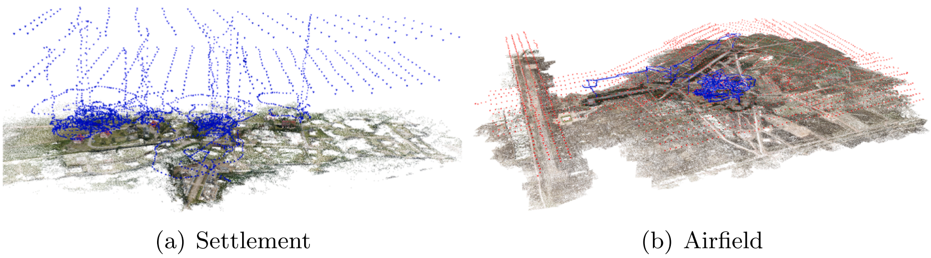
From the point of view of parallel processing, merging rules can be defined as *tasks* and the rule tree as *task tree* (Korch and Rauber, 2004).



(a) Church

(b) Monastery

Fig. 6. Image set *Church* (1455 images, 3 cameras) and *Monastery* (1769 images, 3 cameras) consisting of terrestrial and UAV images.



(a) Settlement (b) Airfield

Fig. 7. Image set *Settlement* (3664 images, one camera) and *Airfield* (6210 images, 2 cameras) solely comprising UAV images.

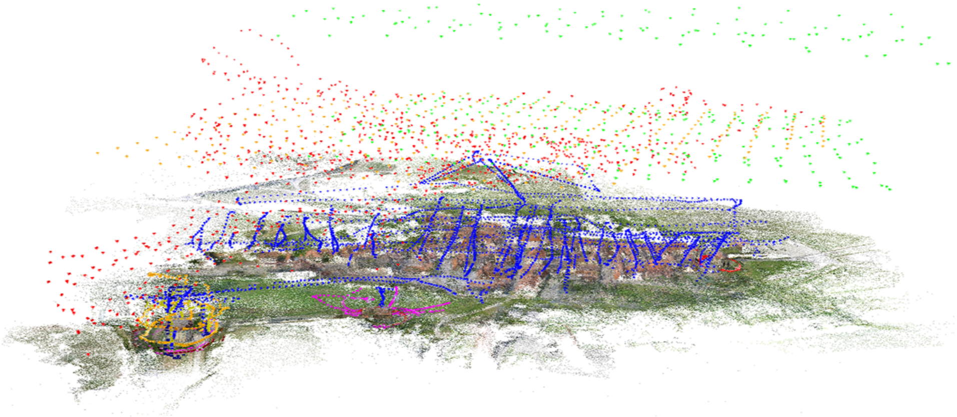


Fig. 8. Image set *Village* consisting of 6405 terrestrial, UAV and aerial images from 10 different cameras.

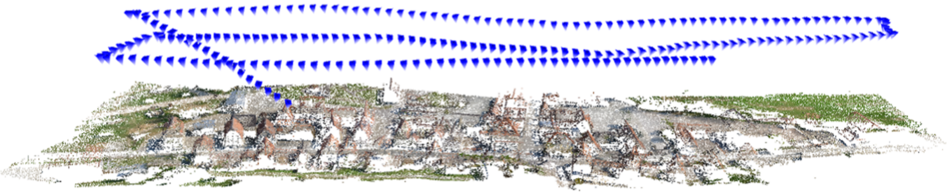


Fig. 9. Image set *Village Overflight* as subset of the image set *Village* comprising 232 UAV images.

Table 3
Influence of loop closing on orientation estimation considering the number of the reconstructed 3D points and the mean reprojection error σ_0 in pixel. f is the damping factor described in Section 3.4.

Image set	Closed loops	f	Points	σ_0
UQ St Lucia	no	–	203 466	0.228
	1	10	204 485	0.236
	55	0	199 373	0.253
Village Overflight	no	–	166 993	0.283
	4	10	170 472	0.342
	110	0	199 182	0.422

Optimal processing then corresponds to a task scheduling problem, where the general version is NP-hard (Bruckert, 2007). Various assumptions and simplifications about tasks and the underlying architecture are made in practical applications (Davis and Burns, 2011; Korch and Rauber, 2004). Assuming an architecture with a uniform memory access, we can neglect the communication overhead. The important task characteristics in our case remain the partial dependencies and the varying merging runtime depending on the IS size. Thus, we employ a dynamic scheduling strategy, which introduces a negligible overhead but allows for a significantly better utilization of the parallel processing units.

5. Results

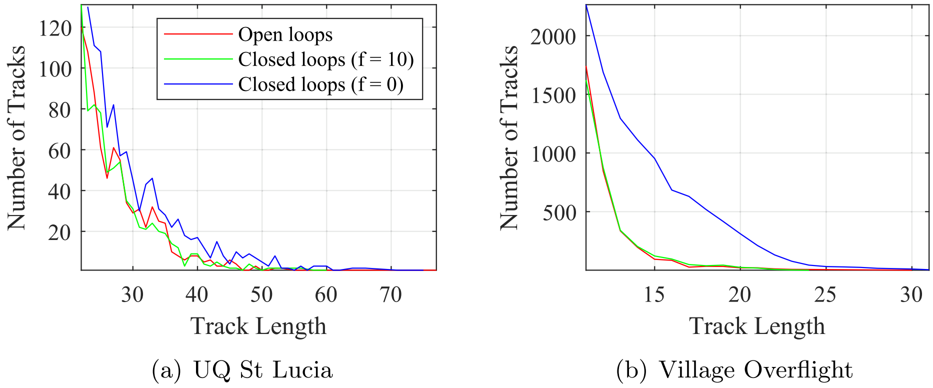
The capability of the proposed approach is demonstrated on image sets whose properties are specified in Table 2 and a system whose specification is listed in Table 1. Camera orientations of the image sets, estimated using the proposed approach, are shown in Figs. 6–9, where colors represent different camera types. The pyramids correspond to camera orientations with the apex of the pyramid giving the camera position and the rotation of the pyramid the camera direction.

Images are subdivided into terrestrial, UAV and aerial. *Terrestrial images* were taken with a handheld camera from the ground. Cameras mounted on a UAV (unmanned aerial vehicle) provide *UAV images* and a high-resolution aerial camera mounted on an airplane *aerial images*. The latter usually comprise a predefined configuration including overlap. Thus, they do not suffer from critical camera configurations. However, terrestrial and UAV images can contain arbitrary camera configurations, which may lead to strong relative image distortions as well as critical configurations.

The image set *UQ St Lucia* comprises a subset of images from the dataset of Warren et al. (2010), where images were taken by a stereo camera system mounted on a vehicle. The employed subset contains images with a distance of approximately one meter forming an image loop. We have used the provided internal camera parameters for our tests.

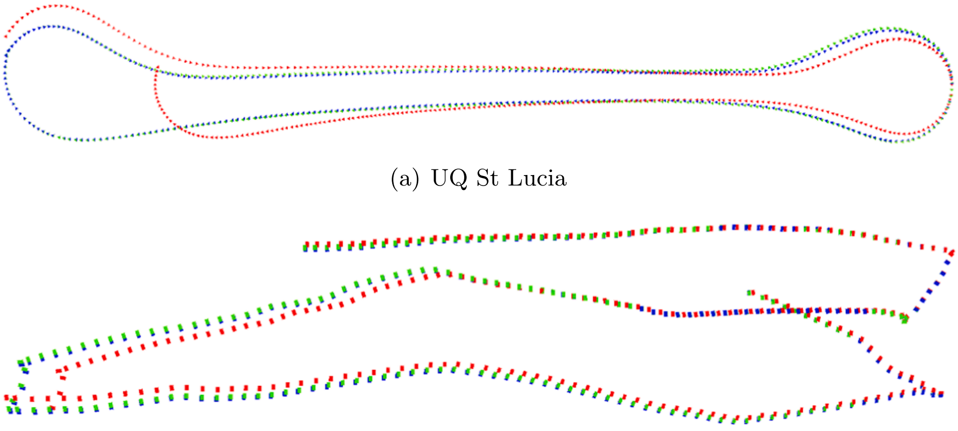
5.1. Loop closing

The potential of the loop closing method (cf. Section 3.4) is



(a) UQ St Lucia (b) Village Overflight

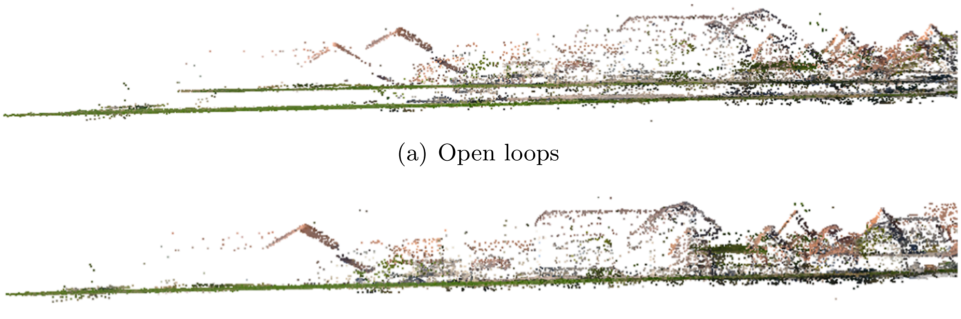
Fig. 10. Influence of loop closing on track lengths. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) UQ St Lucia

(b) Village Overflight

Fig. 11. Estimated camera orientations represented in red for open loops, in green for closed loops with damping factor $f = 10$ and in blue for closed loops $f = 0$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Open loops

(b) Closed loops ($f = 10$)

Fig. 12. Influence of loop closing on the reconstructed point clouds for the image set *Village Overflight*.

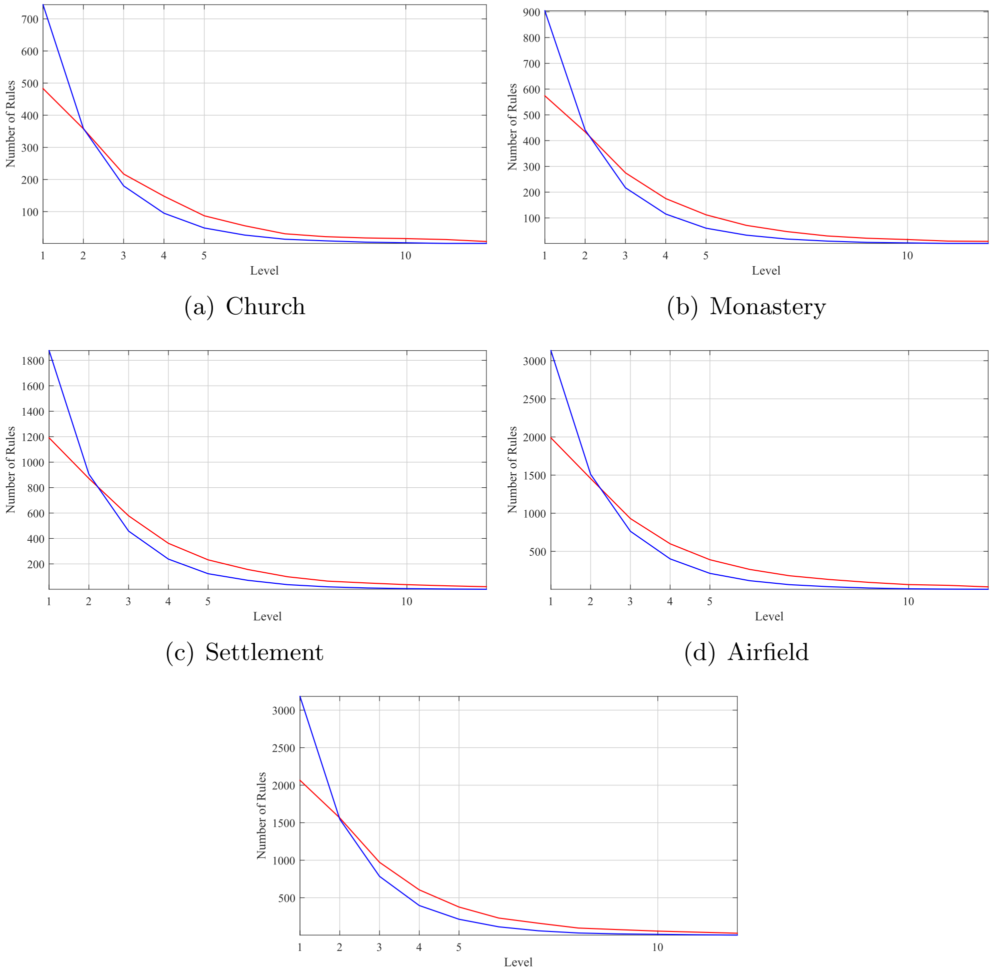
demonstrated on the image sets *UQ St Lucia* and *Village Overflight* which contain simple image loops and, thus, are well suited for visualization.

Orientation estimation of the presented SfM approach is based on hierarchical merging (Mayer, 2014) which for efficiency reasons randomly removes 3D points before bundle adjustment. However, we turned off this point reduction allowing all 3D points to be used in bundle adjustment. This is necessary for a meaningful analysis of the influence of loop closing.

Changes in the number of 3D points as well as the reprojection error in dependence on the number of closed loops are listed in Table 3. The influence of loop closing on mean reprojection error is complex. One important reason for the higher values is the usage of non-linear bundle adjustment, which only reduces the error locally. Furthermore, higher reprojection errors are realistic due to the larger mean track lengths (see below).

The reduced number of 3D points after loop closing in case of the image set *UQ St Lucia* results from the fact, that the same 3D point has been reconstructed multiple times because of the missing image links. However, loop closing establishes those missing links leading to a single 3D point. Another reason is the more accurate internal consistency check due to the multiple links. In this way, wrong 3D points can be detected more reliably and filtered out.

The influence of the loop closing on the track length is shown in Fig. 10. A *track* is a continuous link between feature points



(a) Church

(b) Monastery

(c) Settlement

(d) Airfield

(e) Village

Fig. 13. Merging rules per level of the rule tree using the cluster-based (Mayer, 2014) (red) and our novel matching-based (blue) rule ge.neration method. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Framework properties (FE – Feature extraction, BA – Bundle adjustment, M – Merging of image subsets, H – hierarchical, I – incremental).

Framework	FE	Image Matching	BA	M
APE	GPU (Wu, 2012)	Embedding CPU	CPU	H
COLMAP	GPU (Wu, 2012)	Vocabulary Tree GPU (Wu, 2012)	GPU (Wu et al., 2011)	I
VisualSFM	GPU (Wu, 2012)	GPU (Wu, 2012)	GPU (Wu et al., 2011)	I
OpenMVG	CPU	Cascade Hashing (Cheng et al., 2014) CPU	CPU	I
SAMANTHA	GPU	GPU	CPU	H

corresponding to a 3D point in the scene across multiple images. The number of images is termed *track length*. As expected, the number of longer tracks tends to increase with the number of closed image loops.

Fig. 11 shows the deviations between the estimated camera orientations with and without loop closing. The latter leads to a significant deviation at the ends of the image sequence in the image set *UQ St Lucia* and, thus, wrong camera orientations due to the accumulated errors along the elongated image sequence. In case of the image set *Village Overflight*, only slight deviations in the camera orientations are noticeable. However, multiple reconstructions of the same 3D points lead to double 3D structures with significant deviations, which are visible in form of double roofs and terrain in Fig. 12.

In summary, loop closing is essential for an accurate and reliable orientation estimation, especially in case of elongated image sequences. The tests have demonstrated, that already a few additional links can improve the quality of the estimated camera orientations as well as the sparse 3D reconstruction significantly. A higher value of the damping factor f reduces the computation time and is sufficient to produce geometrically consistent reconstructions. Depending on the specific goal, lower values can be employed to establish longer tracks.

5.2. Hierarchical merging of image subsets

In order to experimentally validate the theoretical concepts presented in Section 4.3 we have compared the number of merging rules in rule trees generated by the cluster-based method (Mayer, 2014) and our novel matching-based rule generation method.

The results for several larger image sets are given in Fig. 13. It shows that the matching-based method is able to generate significantly more independent merging rules in lower levels of the rule trees. This leads to lower trees allowing a more fine-grained parallelism. In addition, it is more likely to find independent rules even across different

Table 5
Results of automatic orientation estimation for the image sets described in Section 5. The runtime t is specified in hours, B is the number of images in the largest consistent block and N the size of the image set. The last two columns specify the number of the reconstructed 3D points and the mean reprojection error σ_0 in pixels for the largest block.

Image Set	Framework	t	$\frac{B}{N}$	Points	σ_0
Village Overflight (232 images)	APE	0.08	1.000	46 150	0.29
	COLMAP	1.06	1.000	229 431	0.69
	VisualSFM	0.29	0.600	22 570	1.04
	OpenMVG	1.14	1.000	442 261	0.39
	SAMANTHA	0.30	1.000	211 857	1.22
UQ St Lucia (351 images)	APE	0.22	1.000	61 289	0.23
	COLMAP	1.11	0.530	32 206	0.51
	VisualSFM	1.64	0.430	14 295	205.62
	OpenMVG	0.12	0.460	7615	0.58
	SAMANTHA	0.20	1.000	36 639	0.31
Church (1455 images)	APE	0.61	0.999	290 748	0.55
	COLMAP	4.31	0.999	560 887	0.95
	VisualSFM	10.35	0.198	37 476	0.74
	OpenMVG	21.47	0.994	168 4501	0.52
	SAMANTHA	5.47	0.970	156 329	3.31
Monastery (1769 images)	APE	1.66	0.992	330 749	0.63
	COLMAP	5.26	0.997	756 306	0.98
	VisualSFM	13.81	0.201	20 585	1.36
	OpenMVG	23.59	0.503	1 415 697	0.58
	SAMANTHA	12.35	0.994	789 725	8.24
Settlement(3664 images)	APE	6.88	0.997	701 765	0.59
	COLMAP	14.94	0.999	3 038 212	1.17
	VisualSFM	77.93	0.165	49 845	1.50
	OpenMVG	–	–	–	–
	SAMANTHA	–	–	–	–
Airfield(6210 images)	APE	13.46	0.985	1 191 525	0.50
	COLMAP	46.48	0.999	6 099 781	1.17
	VisualSFM	–	–	–	–
Village(6405 images)	APE	7.69	0.976	1 318 265	0.38
	COLMAP	26.99	0.992	3 741 066	0.85
	VisualSFM	–	–	–	–

Table 6
Detailed timing results for automatic orientation estimation for the image sets described in Section 5. The runtime for feature extraction t_m, image linking t_v, merging of the image subsets t_r as well as the total runtime $t = t_m + t_v + t_r$ are given in hours.

Image Set	Framework	t_m	t_v	t_r	t
Village Overflight (232 images)	APE	0.01	0.05	0.02	0.08
	COLMAP	0.02	0.63	0.41	1.06
	VisualSFM	0.03	0.24	0.02	0.29
	OpenMVG	0.21	0.19	0.74	1.14
	SAMANTHA	0.09	0.04	0.17	0.30
UQ St Lucia (351 images)	APE	0.01	0.18	0.03	0.22
	COLMAP	0.01	0.38	0.72	1.11
	VisualSFM	0.02	1.29	0.33	1.64
	OpenMVG	0.03	0.07	0.02	0.12
	SAMANTHA	0.01	0.02	0.17	0.20
Church (1455 images)	APE	0.12	0.38	0.11	0.61
	COLMAP	0.16	2.56	1.59	4.31
	VisualSFM	0.21	10.06	0.09	10.35
	OpenMVG	2.41	9.21	9.85	21.47
	SAMANTHA	0.83	0.32	4.33	5.47
Monastery (1769 images)	APE	0.48	1.04	0.14	1.66
	COLMAP	0.24	2.27	2.75	5.26
	VisualSFM	0.29	13.38	0.14	13.81
	OpenMVG	3.68	13.03	6.88	23.59
	SAMANTHA	1.10	1.28	9.97	12.35
Settlement (3664 images)	APE	3.42	2.14	1.32	6.88
	COLMAP	0.69	6.30	7.96	14.94
	VisualSFM	0.80	76.82	0.32	77.93
	OpenMVG	12.47	>144	–	–
	SAMANTHA	3.70	5.19	>144	–
Airfield (6210 images)	APE	4.18	6.91	2.37	13.46
	COLMAP	0.90	10.53	35.05	46.48
	VisualSFM	1.12	209.82	×	–
Village (6405 images)	APE	1.43	4.34	1.92	7.69
	COLMAP	0.50	13.38	13.12	26.99
	VisualSFM	0.62	222.25	×	–

levels which can further improve parallelization.

5.3. Comparison with other structure from motion frameworks

The capability of the proposed approach, termed *Automatic Pose Estimator (APE)*, is compared to the following state-of-the-art frameworks with properties summarized in Table 4:

VisualSFM [3] (Wu, 2013) is an incremental approach which utilizes parallelization on a graphic card to accelerate orientation estimation. However, it scales poorly for large image sets due to the employed exhaustive image matching, but is included in the comparison to depict the limitations of solely an efficient implementation.

COLMAP [4] (Schönberger and Frahm, 2016) is an incremental approach which is based on similar techniques as VisualSFM, but employs a *vocabulary tree* (Schönberger et al., 2017) for accelerating image matching. We have used the provided vocabulary trees according to the image set size for our tests. Because of the efficient implementation as well as up-to-date techniques it employs, we have considered it as the reference approach in the comparison.

OpenMVG [5] (Moulon et al., 2013) is an incremental approach which employs *cascade hashing* (Cheng et al., 2014) to accelerate image matching. We have included it in the comparison as an approach with an alternative image matching technique to COLMAP.

SAMANTHA (Toldo et al., 2015) is hierarchical approach and a part of the commercial photogrammetric software 3DF Zephyr[6]. We have used it in the comparison as an additional hierarchical approach besides ours.

Except SAMANTHA, all frameworks are based on SIFT features (Lowe, 2004). Instead, SAMANTHA employs features which are extracted according to Lindeberg (1998) and, thus, are similar to SIFT features.

No additional information in the form of GPS/INS data or a predefined camera configuration has been employed. Internal camera parameters have been estimated using Exif tags of the images (except for the image set *UQ St Lucia*). Because of the large number of potential parameters as well as the magnitude of parameter combinations and also the diversity of the frameworks, we have used the standard settings in each framework assuming that these are the parameter settings most suitable for general cases.

[3] ccwu.me/vsfm, Version 0.5.26

[4] github.com/colmap/colmap, Version 3.2

[5] github.com/openMVG/openMVG, Version 1.2.0

[6] 3dflow.net, 3DF Zephyr Aerial, Version 3.503

Tables 5 and 6 summarize the results for the different frameworks. The best results have been achieved with APE and COLMAP, where APE outperforms COLMAP in terms of the runtime. On the other hand, except for the image set *UQ St Lucia*, COLMAP has been able to produce slightly larger blocks. However, these properties depend on each other, i.e., the runtime increases for more intense search for missing links. Overall, APE produces results similar to the state-of-the-art approach COLMAP demonstrating that there are different means leading to a similar end.

Interestingly, all frameworks, except APE and SAMANTHA, had problems with the image set *UQ St Lucia*. They have not been able to link the complete image set, despite the specification of internal camera parameters. VisualSFM even failed completely which is indicated by the large mean reprojection error.

Overall, VisualSFM has shown the worst performance. It failed with an undefined error during processing of the image sets *UQ St Lucia*, *Airfield* as well as *Village*. In addition, it has only been able to build relatively small blocks. It also produces an insufficient number of points for the image sets *Church* and *Settlement*.

OpenMVG scales insufficiently for large image sets. One of the reasons lies probably in the large number of the extracted features, which leads to a higher number of reconstructed 3D points in case of image sets *Church* and *Monastery*, but increases the runtime significantly. In addition, the lack of parallelization on a graphic card leads to a relatively high runtime even for feature extraction which has a linear complexity making this approach inferior to APE and COLMAP.

The cluster-based, hierarchical approach employed in SAMANTHA starts to become at about $^3/_4$ of the processing progress inexplicably extremely slow. Consequently, the processing of the larger image sets *Settlement*, *Airfield* as well as *Village* could not been completed even after one week.

6. Conclusion

In this paper, an automatic SfM approach for (unordered) image sets comprising complex configurations has been presented. Apart from (approximate) internal camera calibration, no other information like GPS or INS data is required.

We proposed a graph-based method allowing for an efficient and unsupervised search for image links even in case of strong image distortions as well as critical camera configurations. In addition, an optimization technique is presented which improves the load balancing properties of the hierarchical image subset merging, thus, allowing a better utilization of the parallel processing hardware.

The robustness of our SfM framework concerning various camera configurations, but also the capability to efficiently handle large image sets is demonstrated on various complex image sets. Finally, its potential is highlighted by comparison with several state-of-the-art SfM frameworks.

By being able to produce results with similar speed and quality as the state-of-the-art approach COLMAP, we demonstrate that different means can lead to the same end. This gives additional options for future developments. Particularly, our approach for image linking reduces the number of pairs and triplets which have to be verified, thus, opening design options for the use of methods for pairs and triplets which can deal with wide baselines and, therefore, have a high computational complexity.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R., 2009. Building Rome in a Day. In: International Conference on Computer Vision, pp. 72–79.

Beder, C., Steffen, R., 2006. Determining an initial image pair for fixing the scale of a 3D reconstruction from an image sequence. In: Pattern Recogn. pp. 657–666.

Bruckert, P., 2007. Scheduling Algorithms. Springer-Verlag, Berlin Heidelberg.

Cai, H., Mikolajczyk, K., Matas, J., 2011. Learning linear discriminant projections for dimensionality reduction of image descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 33 (2), 338–352.

Chen, Y.H., 2011. An improved approximation algorithm for the terminal Steiner tree problem. Comput. Sci. Appl. – ICCSA 6784, 141–151.

Cheng, J., Leng, C., Wu, J., Cui, H., Lu, H., 2014. Fast and accurate image matching with cascade hashing for 3D reconstruction. In: Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Crandall, D., Owens, A., Snavely, N., Huttenlocher, D., 2011. Discrete-continuous optimization for large-scale structure from motion. In: Conference on Computer Vision and Pattern Recognition, pp. 3001–3008.

Davis, R.I., Burns, A., 2011. A survey of hard real-time scheduling for multiprocessor systems. ACM Comput. Surv. 43 (4), 1–44.

Farenzena, M., Fusiello, A., Gherardi, R., 2009. Structure-and-motion pipeline on a hierarchical cluster tree. In: International Conference on Computer Vision Workshop, pp. 1489–1496.

Fitzgibbon, A.W., Zisserman, A., 1998. Automatic camera recovery for closed or open image sequences. In: European Conference on Computer Vision, pp. 311–326.

Frahm, J.-M., Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M., 2010. Building Rome on a cloudless day. In: European Conference on Computer Vision, pp. 368–381.

Gherardi, R., Farenzena, M., Fusiello, A., 2010. Improving the efficiency of hierarchical structure-and-motion. In: Conference on Computer Vision and Pattern Recognition, pp. 1594–1600.

Hartmann, W., Havlena, M., Schindler, K., 2016. Recent developments in large-scale tiepoint matching. ISPRS J. Photogramm. Remote Sens. 115, 47–62.

Havlena, M., Hartmann, W., Schindler, K., 2013. Optimal reduction of large image databases for location recognition. In: International Conference on Computer Vision Workshop, pp. 676–683.

Havlena, M., Schindler, K., 2014. VocMatch: efficient multiview correspondence for structure from motion. In: European Conference on Computer Vision, pp. 46–60.

Havlena, M., Torii, A., Pajdla, T., 2010. Efficient structure from motion by graph optimization. In: European Conference on Computer Vision, pp. 100–113.

Heath, K., Gelfand, N., Ovsjanikov, M., Aanjaneya, M., Guibas, L.J., 2010. Image webs: computing and exploiting connectivity in image collections. In: Conference on Computer Vision and Pattern Recognition, pp. 3432–3439.

Heinly, J., Schönberger, J.L., Dunn, E., Frahm, J.-M., 2015. Reconstructing the World in six days. In: Conference on Computer Vision and Pattern Recognition, pp. 3287–3295.

Irschara, A., Hoppe, C., Bishof, H., Kluckner, S., 2011. Efficient structure from motion with weak position and orientation priors. In: Conference on Computer Vision and Pattern Recognition Workshops, pp. 21–28.

Jaccard, P., 1912. The distribution of the flora in the alpine zone. New Phytol. 11 (2), 37–50.

Jegou, H., Douze, M., Schmid, C., 2008. Hamming embedding and weak geometric consistency for large scale image search. In: European Conference on Computer Vision, pp. 304–317.

Jiang, N., Cui, Z., Tan, P., 2013. A global linear method for camera pose registration. In: International Conference on Computer Vision, pp. 481–488.

Ke, Y., Sukthankar, R., 2004. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In: Conference on Computer Vision and Pattern Recognition. vol. 2. pp. 506–513.

Kim, K.I., Tompkin, J., Theobald, M., Kautz, J., Theobalt, C., 2012. Match graph construction for large image databases. In: European Conference on Computer Vision, pp. 272–285.

Klopschitz, M., Irschara, A., Reitmayr, G., Schmalstieg, D., 2010. Robust Incremental Structure from Motion. In: International Symposium on 3D Data Processing, Visualization and Transmission.

Koch, R., Pollefeys, M., Gool, L.V., 1998. Automatic 3D Model Acquisition from Uncalibrated Image Sequences. In: Computer Graphics International Conference. pp. 597–604.

Korch, M., Rauber, T., 2004. A comparision of task pools for dynamic load balancing of irregular algorithms. Concurr. Comput.: Pract. Exp. 16 (1), 1–47.

Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.-M., 2008. Modeling and recognition of landmark image collections using iconic scene graphs. In: European Conference on Computer Vision, pp. 427–440.

Lin, G., Xue, G., 2002. On the terminal Steiner problem. Inform. Process. Lett. 84 (2), 103–107.

Lindeberg, T., 1998. Feature detection with automatic scale selection. Int. J. Comput. Vision 30 (2), 79–116.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60 (2), 91–110.

Mayer, H., 2014. Efficient hierarchical triplet merging for camera pose estimation. In: Pattern Recogn. pp. 399–409.

Mayer, H., Bartelsen, J., Hirschmüller, H., 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. Springer-Verlag, pp. 285–304.

Michelini, M., Mayer, H., 2016. Efficient wide baseline structure from motion. ISPRS Ann.

Photogramm. Remote Sens. Spatial Inf. Sci. III-3, 99–106.

Michelini, M., Mayer, H., 2019. Detection of critical camera configurations for structure from motion using random forest. In: Asian Conference on Pattern Recognition.

Moulon, P., Monasse, P., Marlet, R., 2013. Adaptive structure from motion with a Centrario model estimation. In: Asian Conference on Computer Vision.

Ni, K., Dellaert, F., 2012. HyperSfM. In: International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission. pp. 144–151.

Nistér, D., Stewenius, H., 2006. Scalable recognition with a vocabulary tree. In: Conference on Computer Vision and Pattern Recognition, pp. 2161–2168.

Oliva, A., Torralba, A., 2001. Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vision 42 (3), 145–175.

Philbin, J., Zisserman, A., 2008. Object mining using a matching graph on very large image collections. In: Indian Conference on Computer Vision, Graphics & Image Processing, pp. 738–745.

Raginsky, M., Lazebnik, S., 2009. Locality-sensitive binary codes from shift-invariant kernels. Adv. Neural Inf. Process. Syst. 22, 1509–1517.

Raguram, R., Tighe, J., Frahm, J.-M., 2012. Improved geometric verification for large scale landmark image collections. In: British Machine Vision Conference. pp. 77. 1–77.11.

Repko, J., Pollefeys, M., 2005. 3D models from extended uncalibrated video sequences: addressing key-frame selection and projective drift. In: International Conference on 3-D Digital Imaging and Modeling, pp. 150–157.

Schaffalitzky, F., Zisserman, A., 2002. Multi-view matching for unordered image sets, or How Do I organize my holiday snaps? In: European Conference on Computer Vision, pp. 414–431.

Schönberger, J.L., Berg, A.C., Frahm, J.-M., 2015. PAIGE: PAirwise image geometry encoding for improved efficiency in structure-from-Motion. In: Conference on Computer Vision and Pattern Recognition, pp. 1009–1018.

Schönberger, J.L., Frahm, J.-M., 2016. Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition. pp. 4104–4113.

Schönberger, J.L., Price, T., Sattler, T., Frahm, J.-M., Pollefeys, M., 2017. A vote-and-verify strategy for fast spatial verification in image retrieval. In: Asian Conference on Computer Vision, pp. 321–337.

Sivic, J., Zisserman, A., 2003. Video Google: a text retrieval approach to object matching in videos. In: International Conference on Computer Vision. vol. 2. pp. 1470–1477.

Snavely, N., Seitz, S.M., Szeliski, R., 2006. Photo tourism: exploring image collections in 3D. In: Special Interest Group on Graphics and Interactive Techniques. vol. 25. pp. 835–846.

Snavely, N., Seitz, S.M., Szeliski, R., 2008. Skeletal graphs for efficient structure from motion. In: Conference on Computer Vision and Pattern Recognition. pp. 1–8.

Steedly, D., Essa, I., Dellaert, F., 2003. Spectral partitioning for structure from motion. In: International Conference on Computer Vision. vol. 2. pp. 996–1003.

Steele, K.L., Egbert, P.K., 2006. Minimum spanning tree pose estimation. In: International Symposium on 3D Data Processing, Visualization and Transmission, pp. 440–447.

Strecha, C., Bronstein, A., Bronstein, M., Fua, P., 2012. LDAHash: Improved matching with smaller descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 34 (1), 66–78.

Toldo, R., Gherardi, R., Ferenzena, M., Fusiello, A., 2015. Hierarchical structure-and-motion recovery from uncalibrated images. Comput. Vis. Image Underst. 140 (C), 127–143.

Torralba, A., Fergus, R., Weiss, Y., 2008. Small codes and large image databases for recognition. In: Conference on Computer Vision and Pattern Recognition. pp. 1–8.

Triggs, B., McLauchlan, P.F., Hartley, R., Fitzgibbon, A.W., 2000. Bundle adjustment – a modern synthesis. In: Vision Algorithms: Theory and Practice. pp. 298–372.

Warren, M., McKinnon, D., He, H., Upcroft, B., 2010. Unaided stereo vision based pose estimation. In: Australasian Conference on Robotics and Automation.

Wefelscheid, C., 2013. Monocular Camera Path Estimation Cross-linking Images in a Graph Structure. PhD Thesis. Technische Universität Berlin.

Wilson, K., Snavely, N., 2014. Robust global translations with 1DSfM. In: European Conference on Computer Vision, pp. 61–75.

Wu, C., 2012. SiftGPU: A GPU Implementation of Scale Invariant Feature Transform (SIFT).

Wu, C., 2013. Towards linear-time incremental structure from motion. In: International Conference on 3D Vision, pp. 127–134.

Wu, C., Agarwal, S., Curless, B., Seitz, S.M., 2011. Multicore Bundle Adjustment. In: Conference on Computer Vision and Pattern Recognition. pp. 3057–3064.

Zach, C., Irschara, A., Bischof, H., 2008. What can missing correspondences tell us about 3D structure and motion. In: Conference on Computer Vision and Pattern Recognition, pp. 1–8.

Zach, C., Klopschitz, M., Pollefeys, M., 2010. Disambiguating visual relations using loop constraints. In: Conference on Computer Vision and Pattern Recognition, pp. 1426–1433.