Optimizing the Viewing Graph for Structure-from-Motion

Chris Sweeney\textsuperscript{1} \hspace{1em} Torsten Sattler\textsuperscript{2} \hspace{1em} Tobias Höllerer\textsuperscript{1} \hspace{1em} Matthew Turk\textsuperscript{1} \hspace{1em} Marc Pollefeys\textsuperscript{2}

\textsuperscript{1}University of California Santa Barbara \\ \{cmsweeney, holl, mturk\}@cs.ucsb.edu
\textsuperscript{2}Department of Computer Science \\ ETH Zürich, Switzerland \\ \{sattlert, marc.pollefeys\}@inf.ethz.ch

Abstract

The viewing graph represents a set of views that are related by pairwise relative geometries. In the context of Structure-from-Motion (SfM), the viewing graph is the input to the incremental or global estimation pipeline. Much effort has been put towards developing robust algorithms to overcome potentially inaccurate relative geometries in the viewing graph during SfM. In this paper, we take a fundamentally different approach to SfM and instead focus on improving the quality of the viewing graph before applying SfM. Our main contribution is a novel optimization that improves the quality of the relative geometries in the viewing graph by enforcing loop consistency constraints with the epipolar point transfer. We show that this optimization greatly improves the accuracy of relative poses in the viewing graph and removes the need for filtering steps or robust algorithms typically used in global SfM methods. In addition, the optimized viewing graph can be used to efficiently calibrate cameras at scale. We combine our viewing graph optimization and focal length calibration into a global SfM pipeline that is more efficient than existing approaches. To our knowledge, ours is the first global SfM pipeline capable of handling uncalibrated image sets.

1. Introduction

The viewing graph is a fundamental tool in the context of Structure-from-Motion (SfM) [20, 26, 29]. This graph encapsulates the cameras that are to be estimated as vertices and the relative geometries between cameras as edges. SfM algorithms take the relative geometries from the viewing graph as an input and output a reconstruction consisting of camera poses and 3D points. The traditional method for computing a SfM reconstruction is incremental SfM [28, 32] which progressively grows a reconstruction by adding one new view at a time. Incremental SfM requires repeatedly performing nonlinear optimization (i.e., bundle adjustment) as the reconstruction grows in size. As a result, incremental SfM is able to overcome noise in the viewing graph because errors and inaccuracies from the viewing graph are consistently corrected through bundle adjustment.

Much recent work has focused on so-called “global SfM” techniques that consider all relative poses (i.e., edges in the viewing graph) to simultaneously estimate all camera poses in a single step [3, 11, 12]. These methods operate on calibrated image sets by first estimating the global orientation of all cameras simultaneously [6, 13, 14, 21], then solving for the camera positions simultaneously [3, 16, 22, 31]. Finally, structure is estimated and a global bundle adjustment is applied. Since bundle adjustment is generally the most expensive part of SfM, global SfM methods are generally more efficient and scalable than incremental methods as they only require a single bundle adjustment.

Since global SfM relies on averaging relative rotations and translations, the quality of the input relative poses di-
rectly affects the final reconstruction quality. Various filtering techniques exist [15] [31] to remove outlier edges from the viewing graph; however, it is clear to see that the effectiveness of these methods will decrease when the accuracy of relative geometries in the viewing graph decreases, since it will be more difficult to distinguish noise from outliers. Inaccurate relative geometries are common in the context of SfM from internet photo collections [28] and may arise from a variety of reasons including poor calibration, repeated structures, image noise, and poor or sparse feature matches. Indeed, much effort has been put towards designing robust SfM algorithms that are capable of overcoming potentially inaccurate relative geometries.

In this paper, we approach SfM from a fundamentally different perspective: rather than treating potentially inaccurate two-view geometry as static input to SfM, we instead attempt to recover a consistent viewing graph from a noisy one such that the performance of any SfM method will be improved. In practice, it is unlikely that we are able to recover a perfectly consistent viewing graph; however, we show that enforcing loop consistency in the viewing graph makes estimating structure and motion easier by improving the convergence of current SfM algorithms. As our main contribution, we propose a novel method to optimize the viewing graph and enforce global consistency through loop constraints. We use the epipolar point transfer across triplets in the viewing graph as a geometric error for loop consistency and directly optimize fundamental matrices connecting views. An important contribution of our viewing graph optimization is that it is able to operate on calibrated or uncalibrated datasets, and we present a scalable calibration method for determining focal lengths of uncalibrated cameras (see Section 5).

Our optimization is able to greatly improve the accuracy of relative poses in the viewing graph (see Section 6), and the resulting optimized viewing graph does not require any filtering steps during SfM to remove “bad” relative geometries. This is in contrast to alternative methods [16] [22] [31] which require complex filtering steps throughout camera pose estimation. As a result, we are able to design a simple global SfM pipeline (compared to alternative approaches such as [16] [22] [31]) that is extremely efficient. To our knowledge, this is the first global SfM method that is able to handle uncalibrated image sets. We demonstrate on several large scale datasets that our optimization and simplified SfM pipeline is able to greatly improve the efficiency of large scale SfM while maintaining comparable accuracy.

1.1. Related Work

We will briefly present some of the related works here, and will present other related works throughout the remainder of the paper.

Much previous work has analyzed the viewing graph. Levi and Werman [20] presented a theoretical analysis of viewing graphs, and provide linear methods for inferring missing edges from a consistent viewing graph given up to 6 views. Rudi et al. [24] present a followup to this work by analyzing the solvability of viewing graphs in the context of creating reconstructions. Both of these works, however, only analyze characteristics of consistent viewing graphs.

In contrast, Pillai and Govindu [25] assume they are given a non-consistent viewing graph and present a method that attempts to modify it to form a consistent viewing graph. They iteratively re-estimate feature points locations of observed feature points based on the epipolar point transfer, then use these updated feature points to re-estimate fundamental matrices connecting views. This process is repeated until convergence; however, convergence is not guaranteed and even on the small datasets presented (fewer than 15 images) the method does not converge after 200 iterations.

2. The Viewing Graph

A scene consisting of n views may be represented by a viewing graph $G = \{V, E\}$ whose vertices $V$ correspond to views in the scene and whose edges $E$ correspond to feature matches and relative geometries between two views, namely the fundamental matrix connecting two views. Specifically, $F_{ij}$ is the fundamental matrix that transfers points in image $j$ to lines in image $i$. The viewing graph contains information about the relative geometry between views but does nothing to enforce geometric constraints beyond 2-view geometry. For example, there may be triplets (loops of size 3) whose relative geometry is not geometrically feasible when considering all three edges $\{i, j, k\}$. Ideally, the edges in these loops would be consistent with each other.

Condition 1. A triplet of fundamental matrices is consistent when they satisfy [15]:

$$e_{ij}^T F_{ij} e_{jk} = e_{ij}^T F_{ik} e_{kj} = e_{ji}^T F_{jk} e_{ki} = 0,$$

(1)

where $e_{ij}$ is the epipole of $F_{ij}$ corresponding to the image of camera center $j$ in view $i$ and $e_{ij} \neq e_{ik}$ i.e., the non-collinearity condition is satisfied.

Definition 1. A consistent viewing graph is a viewing graph where all triplets satisfy Condition 7.

The geometric interpretation of Definition 1 is that the projection of view $k$’s camera center in image $i$ is consistent with the projection of view $k$’s camera center in image $j$ transferred to image $i$ by the fundamental matrix $F_{ij}$.

Let us now consider the existence of a consistent viewing graph:

Theorem 1. Given a reconstruction $R = \{P, X\}$ consisting of projection matrices $P$ and 3D points $X$, a non-empty set of consistent viewing graphs exists.
Proof. A consistent viewing graph may be constructed directly from the reconstruction \( \mathcal{R} \) by setting each edge \( e_{ij} \in \mathcal{E} \) to the fundamental matrix composed from the two corresponding projection matrices [15]. By construction, Condition [1] is satisfied.

Thus, for every reconstruction \( \mathcal{R} \) there exists a consistent viewing graph \( \mathcal{G}_C \) that will generate \( \mathcal{R} \). Further, it is known that computing a reconstruction from a consistent viewing graph may be done trivially by chaining projection matrices computed directly from the fundamental matrices in the viewing graph [26, 27]. Computing a reconstruction from a non-consistent viewing graph, however, is much more difficult and is the crux of most SfM methods.

3. Creating a Consistent Viewing Graph

Rather than facing the difficult task of computing a reconstruction from a non-consistent viewing graph \( \mathcal{G} \), we propose to instead recover a consistent viewing graph \( \mathcal{G}_C \) from \( \mathcal{G} \) so that computing a reconstruction is simplified [15, 26]. Thus, the goal of this paper is to optimize a noisy, non-consistent viewing graph \( \mathcal{G} = \{V, \mathcal{E}\} \) to recover a consistent viewing graph \( \mathcal{G}_C \) that will improve SfM. This requires adjusting the edges \( F_{ij} \in \mathcal{E} \) to enforce Condition [1].

We propose an optimization scheme that uses a geometric error to enforce loop constraints that attempt to satisfy Condition [1]. If we are able to recover a consistent viewing graph then computing a reconstruction is trivial; however, even in the case that we cannot recover a fully consistent viewing graph the accuracy of the relative geometries improves enough that computing structure and motion is greatly simplified (c.f. Section 3.4).

In the remainder of this section we propose an optimization that operates on the viewing graph, enforcing loop consistency with the epipolar point transfer. Our optimization recovers an approximately consistent viewing graph \( \mathcal{G}_{OPT} \) that improves the performance of SfM by improving convergence in the estimation process.

3.1. Enforcing Loop Consistency

We now propose a cost function for adjusting \( \mathcal{E} \) to enforce triplet consistency in \( \mathcal{G} \). While Condition [1] is a sufficient condition for consistency [26], it is an algebraic metric and is significantly under-constrained. Instead, we propose to use the epipolar point transfer to enforce loop consistency. The epipolar point transfer is defined as the intersection of two transfer lines of two views into a third view (c.f. Figure 2).

\[
\hat{x}_{ij}^k = F_{ij} x_j \times F_{ik} x_k ,
\]

where \( x_i \) is the feature point in image \( i \) and \( \hat{x}_{ij}^k \) is the estimated pixel location of \( x_i \) based on the epipolar transfers from views \( j \) and \( k \). In the ideal case we will have \( x_i = \hat{x}_{ij}^k \); however, this is almost never the case in real data because of image noise and outliers in the feature matching process. Instead, we define a cost function based on the epipolar point transfer:

\[
C(x)_{ij}^k = ||x_i - \hat{x}_{ij}^k||_2 .
\]

This cost is a geometric error in terms of pixel distance and has previously been shown to be effective [10, 25]; however, care must be taken to avoid numerical instabilities (see Section 3.4).

3.2. Updating Fundamental Matrices

We seek to adjust fundamental matrix edges \( F_{ij} \in \mathcal{E} \) in \( \mathcal{G} \) based on Eq. (3). Fundamental matrices are a special class of rank-2 matrices [1]. Thus, updating a fundamental matrix during the nonlinear optimization must be done carefully to ensure that the resulting \( 3 \times 3 \) matrix remains a valid fundamental matrix. We use the nonlinear fundamental matrix representation of Bartoli and Sturm [4] to update the fundamental matrices and briefly summarize their method here.

Note that a fundamental matrix \( F \) may be decomposed into matrices \( U \), \( S \), and \( V \) by singular value decomposition \( F = U S V^\top \), where \( U \) and \( V \) are orthonormal matrices and \( S \) is a \( 3 \times 3 \) diagonal matrix of the form \( \text{diag}(1, s, 0) \). To update \( F \), we apply a \( SO(3) \) rotation to the \( O(3) \) matrices \( U \) and \( V \), and a simple scalar addition to \( s \).

\[
U \leftarrow R_u U \quad (4)
\]
\[
V \leftarrow R_v V \quad (5)
\]
\[
s \leftarrow s + \delta_s \quad (6)
\]

Since \( R_u \) and \( R_v \) are \( SO(3) \) rotations, they may be represented with the minimal 3 parameters (by Euler angle or angle axis representation), thus requiring 7 parameters total (3 for \( R_u \), 3 for \( R_v \), and 1 for \( \delta_s \)) to update \( F \). Since \( F \) has 7 degrees of freedom, this is a minimal parameterization and has been shown to maintain valid fundamental matrices [4].
3.3. Nonlinear Optimization

We create a large nonlinear optimization using the cost function of Eq. (5) and the presented method for updating fundamental matrices. We only optimize edges that are present in triplets $T$ in the viewing graph:

$$F^* = \arg\min_F \sum_{t \in T} \sum_{x \in t} C(x)^{ik} + C(x)^{jk} + C(x)^{ij}, \quad (7)$$

where $x$ is a feature track present in the triplet $t = \{i, j, k\}$ and $F$ is the set of fundamental matrices $F \in \mathcal{E}$. That is, for all triplets, we minimize the epipolar point transfer cost of all feature tracks within the triplet. Although the epipolar point transfer cost function does not require a triplet of fundamental matrices, we found that using triplets greatly improved the rate of convergence. Further, since each camera interacts with other cameras that might not be linked together in a triplet, larger loops are implicitly created.

Finally, it should be noted that the feature points $x$ are treated as constant in Eq. (7) and alternatively could be treated as free parameters that are optimized with the fundamental matrices. We found that additionally optimizing feature points with fundamental matrices resulted in a dramatic decrease in efficiency and did not provide significantly better results.

3.4. Numeric Instabilities

The epipolar point transfer has known degeneracies and numeric instabilities [15]. In particular, any configuration in which the transfer point lies on the trifocal plane of the images $i$, $j$, and $k$ will be degenerate and points near this degeneracy are increasingly ill-conditioned. To avoid ill-conditioned points, we do not consider points where the two transfer lines are nearly parallel or when the transfer lines lay near the epipole. The latter scenario can be checked by examining the norm of the transfer line. Since the epipole is in the null space of $F_{ij}$, the norm of the transfer line will be very small when it is near the epipole.

It should be noted that if the three camera centers are collinear then there is a one-parameter family of planes containing the three cameras and thus the trifocal plane is ambiguous. We explicitly avoid this scenario by removing collinear triplets where the epipoles are equal. In practice, we did not find this to be a limitation since nearly all cameras in real datasets are constrained by at least one non-collinear camera triplet.

4. Estimating Structure and Motion

Given a consistent viewing graph, estimating structure and motion is extremely simple. To see why this is the case, let us consider a consistent and calibrated viewing graph $\mathcal{G}_C$. Since the graph is consistent, this means that the relative rotations in each triplet in $\mathcal{G}_C$ are also consistent (i.e., concatenating the relative rotations in a triplet will form a loop: $R_{ij}R_{jk}R_{ki} = I$). The global orientations of each camera may be easily obtained from a random spanning tree [6] or from a linear orientation method [21]. A consistent viewing graph also means that the relative translation directions in $\mathcal{G}_C$ are perfect i.e., $\alpha_{ij}t_{ij} = R_i(c_j - c_i)$. Thus, estimating the camera positions (assuming orientation is known) is equivalent to recovering the baselines $\alpha_{ij}$ between cameras. This pipeline is simpler than alternative global SfM approaches that require many filtering steps and more complex motion estimation algorithms [16, 22, 31] (c.f., Algorithm 1).

While our viewing graph optimization is not guaranteed to create a consistent viewing graph, the optimization enforces enough of a consistency constraint that the SfM process can be simplified. In fact, we are able to remove all filtering steps from our SfM pipeline, and are able to further simplify the orientation and position estimation algorithms.

4.1. Viewing Graph Optimization

The viewing graph optimization described in Section 3 has $O(|\mathcal{E}|)$ free parameters, and thus the run time of the nonlinear optimization scales directly with the number of edges. Viewing graphs may contain highly redundant information, and so we would like to reduce the number of edges in the viewing graph so as to reduce the size of the nonlinear optimization. This is similar to the skeletal set selection of Snavely et al. [29], whose goal is to find a minimal set of views in the viewing graph that represent the entire scene. Our goal, in contrast, is to find a minimal set of edges that provide sufficient coverage over all views in the viewing graph.

Algorithm 1 Standard Global SfM Pipeline

1: procedure GLOBAL SFM($\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, Focal lengths)
2: Filter $\mathcal{G}$ from loop constraints [8, 22, 33]
3: Robust orientation estimation [6]
4: Filter relative poses [16, 22, 31]
5: Robust Position Estimation [8, 16, 22, 31]
6: Triangulate 3D points
7: Bundle Adjustment
8: end procedure

Figure 3. In order to reduce size of the viewing graph optimization, we construct a subgraph from the maximum spanning tree (MST). Edges in the MST (left) are shown with thick lines. Edges from the original viewing graph (dashed lines) are then added to the MST if they form a triplet to form $\mathcal{G}'$ (right).
Algorithm 2 Our Global SfM Pipeline

1: procedure OUR GLOBAL SfM($G = \{V, E\}$)
2: Choose subgraph $G'$ (Section 4.1)
3: Optimize the $G'$ for consistency (Section 3)
4: [optional] Calibrate cameras (Section 5)
5: Estimate camera orientation from Eq. (8)
6: Estimate camera positions from Eq. (9)
7: Triangulate 3D points
8: Bundle Adjustment
9: end procedure

Given an input viewing graph $G = \{V, E\}$, we aim to create a subgraph $G'$ that sufficiently covers the viewing graph with a minimum number of edges. Similar to [2], we first select the maximum spanning tree $G' = G'MST$ where edge weights are the number of inliers from fundamental matrix estimation between two views then find all edges $E_T \in E$ that, if added to $G'$ would create a triplet in the graph (i.e., a loop of size 3) as show in Figure 3. Among the edges in $E_T$ we select a set of “good” edges $E_G$ that have a triplet projection error less than $\tau$ (see Appendix A) and add these to the graph. The triplet projection error is an approximate measurement to determine how close a triplet of fundamental matrices is to being consistent (Condition [1]). We repeat this procedure (i.e., $G' = G' \cup E_G$) until every view in the viewing graph participates in at least one triplet, or there are no more “good” edges that can be added.

After we have obtained a representative viewing graph $G'$, we must choose which feature tracks to use for the optimization. Similar to Crandall et al. [7], we use a set cover approach to select a subset of all feature tracks to accelerate optimization. In each image, we create an $N \times N$ grid and choose the minimum number of feature tracks such that all grid cells in all images contain at least one track in the optimization. We have found that choosing spatially distributed feature points helps the viewing graph optimization to converge to a better minimum.

Finally, we use all selected edges and feature tracks to optimize the viewing graph by minimizing Eq. (7) using the Ceres Solver library [2]. We use a Huber loss function to remain robust to outliers from feature matching.

4.2. Estimating Motion

The resulting optimized viewing graph provides accurate fundamental matrices that nearly form a consistent viewing graph (c.f. Figure 5). As a result, there is no need for further outlier filtering during the structure and motion estimation. Further, there is no longer a need for robust methods such as [6] or [31]. This simplifies the SfM pipeline from a mathematical standpoint and for implementation purposes. The result is a more efficient pipeline with comparable accuracy to current methods.

Assuming the cameras are calibrated (or calibration is obtained with the method of Section 4.5), computing the orientations is simple. We solve for orientations by enforcing the relative rotation constraint $R_{ij} = R_i R_j^\top$. Similar to the method of [21], we minimize the cost function

$$\sum_{i,j} ||R_i R_{ij} - R_j||_2$$

(8)

to solve for camera orientations. Martinec and Pajdla [21] use a linear least squares technique to solve for matrices that minimize Eq. 8; however, this requires the solutions of the linear system to be projected into SO(3) matrices in order to obtain valid rotations. In contrast, we use the angle-axis parameterization (which ensures that all rotations $R_i$ remain on the rotation manifold throughout the optimization) and minimize Eq. (8) with a nonlinear solver. The orientations are initialized by chaining relative rotations from a random spanning tree as is done in the initialization for [6]. This simplified orientations solver is more efficient than the method of [6] while producing orientations that typically differ less than $1^\circ$ for the datasets in Table 2.

To compute camera positions, we use the same nonlinear position constraint as Wilson and Snavely [31], though our pipeline does not require filtering steps before solving for camera positions. Given a relative translation $t_{ij}$ and a known camera orientation $R_i$, we use the following constraint to estimate camera centers $c_i$ and $c_j$:

$$t_{ij} = R_i (c_j - c_i) / ||c_j - c_i||.$$  (9)

This nonlinear constraint is known to be more stable than other cross-product constraints [3][11]. We use the Ceres Solver library [2] to solve the nonlinear Eq. (8) and Eq. (9) for recovering camera orientations and positions. After estimating camera poses, we triangulate 3D points and run a single bundle adjustment. Our SfM pipeline is summarized in Algorithm 2.

5. Focal Length Calibration

A current limitation of global SfM methods is that they require relative poses in the form of relative rotations and translations as input. For calibrated image sets, the relative poses may be obtained by decomposing the essential matrix [15]. For uncalibrated cameras, only the fundamental matrix is available between two views. Focal lengths may be obtained from the fundamental matrix in closed form [13] and the resulting essential matrix may be decomposed into relative rotations and translations. The relative rotations and translations obtained through fundamental matrix decomposition, however, are far less accurate compared to when calibration is known (c.f. Figure 4) so obtaining accurate
calibration has a direct effect on the quality of SfM algorithms.

Individually decomposing fundamental matrices from all relative geometries containing a particular camera, however, is not guaranteed to yield a single consistent focal length value. That is, each decomposition of a fundamental matrix containing a particular camera may yield a different focal length value for that camera. Further, the quality of the focal lengths computed from a fundamental matrix is solely dependent on the quality of the fundamental matrix estimation. Focal lengths are not a lie group and so a simple averaging of focal lengths does not give statistically meaningful results [5] and a more meaningful metric is needed to effectively “average” focal lengths. In this section we propose a new calibration method for simultaneously determining the focal lengths of all cameras in a viewing graph using only fundamental matrices as input.

5.1. Focal Length from a Fundamental Matrix

First, let us review a technique for determining focal lengths from a single fundamental matrix. An essential matrix $E$ has the form $t \times R$ for a given relative translation $t$ and rotation $R$ if and only if $E$ is rank 2 with its two non-zero singular values equal $\lambda_1 = \lambda_2$ [15]. This property may be encapsulated by the scalar invariants of $E$ [17]:

$$C = ||EE^T||^2 - \frac{1}{2}||E||^4.$$  \hfill (10)

For a valid essential matrix $E$, the cost function $C$ will be 0. Kanatani and Matsunaga [18] show that Eq. (10) may be used to recover the two focal lengths from a fundamental matrix by noting that:

$$E = K^T FK.$$  \hfill (11)

When the focal lengths are unknown, $C$ is a non-negative cost function whose minimum is at 0. By inserting Eq. (11) into Eq. (10), we may solve for the focal length values that minimize $C$. This may be solved in closed form by noting that the first order partial derivatives $\partial C/\partial f'$ and $\partial C/\partial f$ must also be 0 [18].
Figure 6. We show the accuracy of calibration methods on the Pisa dataset \cite{16} and show the focal length error $|f - f_{\text{gt}}|/f_{\text{gt}}$ compared to ground truth focal lengths obtained from a reconstruction from VisualSFM \cite{32}. Our method is at least as accurate as using EXIF, and is significantly more accurate than using the median focal lengths obtained from fundamental matrix decomposition.

6.2. Focal Length Calibration

To determine the accuracy of our calibration method, we use images from the Pisa and Trevi datasets \cite{16} that contain EXIF focal lengths and compare our calibration to reference focal lengths that were obtained from a reference reconstruction generated with VisualSFM \cite{32} after bundle adjustment of the internal and external camera parameters. We compare our method to using EXIF data for calibration as well as the median focal length. The median focal length is obtained by decomposing all fundamental matrices connected to a view and taking the median of the focal lengths obtained from the decompositions.

We plot the accuracy of the focal lengths obtained with each method in Figure 6. For simplicity, we only plot the results from the Pisa dataset; however, the results from the Trevi dataset were similar. For both datasets our calibration method converged in less than 10 seconds. Our method is at least as accurate as using focal length values from EXIF data. The accuracy stems from the use of many two-view constraints to estimate the focal length. EXIF values can be accurate but have the potential to be inaccurate if the image has been resized or cropped. Using the median focal length is very inaccurate and is not sufficient for use in a 3D pipeline.

6.3. Structure-from-Motion

We ran our pipeline on the small-scale dataset of \cite{30} and the large-scale datasets of \cite{31} to measure the performance and feasibility of our method on real data. We compare our SFM pipeline to several alternative global SFM pipelines, and the results are summarized in Tables 1, 2, and 3.

Table 2 shows that our method is approximately up to 10 times more efficient than alternative methods, while maintaining comparable accuracy to the state-of-the-art. The increase in efficiency is a direct result of our simplified SFM pipeline (see Section 4) that is able to efficiently utilize the high quality relative poses obtained from the optimized viewing graph. The statistical pose averaging (c.f. Eq. \ref{eq:stats}) and Eq. (9)) converges to a high quality result very quickly because our optimized viewing graph is extremely accurate (c.f. Figure 5). Visualizations of the reconstructed datasets are included in the supplemental material.

7. Conclusion

In this paper, we have presented a new approach to large-scale SFM. Rather than focusing on creating potentially complex algorithms to overcome noise and outliers in the reconstruction process, we propose an optimization that

---

\footnotesize

$^1$The reconstructions obtained with VisualSFM \cite{32} are not meant to serve as ground truth but merely a reference for a good reconstruction.

$^2$The supplemental material can be found on the author’s website.
Table 1. We evaluate several SfM pipelines on the Strecha MVS datasets [30]. Our method shows excellent accuracy while remaining extremely efficient. Timing results of Cui et al. [8] were not available.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FountainP11</td>
<td>7.6</td>
<td>2.2</td>
<td>2.5</td>
<td>2.5</td>
<td>2.4</td>
<td>3.5</td>
<td>133</td>
<td>5</td>
<td>4.5</td>
</tr>
<tr>
<td>EntryP11</td>
<td>63.0</td>
<td>6.9</td>
<td>-</td>
<td>5.9</td>
<td>5.7</td>
<td>3</td>
<td>88</td>
<td>5</td>
<td>3.8</td>
</tr>
<tr>
<td>HerzJesuP8</td>
<td>19.3</td>
<td>3.9</td>
<td>-</td>
<td>3.5</td>
<td>3.5</td>
<td>2</td>
<td>34</td>
<td>2</td>
<td>1.9</td>
</tr>
<tr>
<td>HerzJesuP25</td>
<td>22.4</td>
<td>5.7</td>
<td>5.0</td>
<td>5.3</td>
<td>5.3</td>
<td>12</td>
<td>221</td>
<td>10</td>
<td>9.3</td>
</tr>
<tr>
<td>CastleP19</td>
<td>258</td>
<td>76.2</td>
<td>-</td>
<td>25.6</td>
<td>38.2</td>
<td>9</td>
<td>99</td>
<td>6</td>
<td>5.7</td>
</tr>
<tr>
<td>CastleP30</td>
<td>522</td>
<td>66.8</td>
<td>21.2</td>
<td>21.9</td>
<td>32.4</td>
<td>18</td>
<td>317</td>
<td>14</td>
<td>11.6</td>
</tr>
</tbody>
</table>

Table 2. We compare results of several global SfM pipelines on the large-scale 1DSfM dataset [31]. We show the number of cameras reconstructed $N_C$ and the median position error approximately in meters $\bar{x}$. For our method, $\bar{x}$ indicates position errors before bundle adjustment, and $\bar{x}_{BA}$ are the errors after bundle adjustment. Our method produces accurate camera poses before bundle adjustment and has comparable accuracy to alternative methods after bundle adjustment.

<table>
<thead>
<tr>
<th>Name</th>
<th>1DSfM [31]</th>
<th>LUD [24]</th>
<th>Cui et al. [8]</th>
<th>Our Pipeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piccadilly</td>
<td>2153</td>
<td>1956.0</td>
<td>-</td>
<td>1928</td>
</tr>
<tr>
<td>Union Square</td>
<td>789</td>
<td>710.3</td>
<td>-</td>
<td>701</td>
</tr>
<tr>
<td>Roman Forum</td>
<td>1084</td>
<td>989.2</td>
<td>-</td>
<td>966</td>
</tr>
<tr>
<td>Vienna Cathedral</td>
<td>836</td>
<td>770.4</td>
<td>750</td>
<td>771</td>
</tr>
<tr>
<td>Piazza del Popolo</td>
<td>328</td>
<td>308.2</td>
<td>305</td>
<td>302</td>
</tr>
<tr>
<td>NYC Library</td>
<td>332</td>
<td>295.4</td>
<td>320</td>
<td>294</td>
</tr>
<tr>
<td>Alamo</td>
<td>577</td>
<td>529.3</td>
<td>547</td>
<td>533</td>
</tr>
<tr>
<td>Metropolis</td>
<td>341</td>
<td>291.5</td>
<td>-</td>
<td>272</td>
</tr>
<tr>
<td>Yorkminster</td>
<td>437</td>
<td>401.0</td>
<td>404</td>
<td>409</td>
</tr>
<tr>
<td>Montreal N.D.</td>
<td>450</td>
<td>427.4</td>
<td>433</td>
<td>416</td>
</tr>
<tr>
<td>Tower of London</td>
<td>572</td>
<td>414.1</td>
<td>425</td>
<td>409</td>
</tr>
<tr>
<td>Ellis Island</td>
<td>227</td>
<td>214.3</td>
<td>-</td>
<td>203</td>
</tr>
<tr>
<td>Notre Dame</td>
<td>553</td>
<td>507.1</td>
<td>536</td>
<td>501</td>
</tr>
</tbody>
</table>

computes a consistent triplet of fundamental matrices. We use their technique to define a triplet projection error that measures the consistency of a triplet of fundamental matrices. We will briefly summarize the method here.

First, projection matrices for views $i$ and $j$ and $k$ are constructed from the fundamental matrices

$$ P_i = [I][0] $$

$$ P_j = [[e_{ij}^x] F_{ij} e_{ij}] $$

$$ P_k = [[e_{kj}^x] F_{ik} e_{ki}] $$

where $v$ is an unknown 4-vector. Recall from [15] that a fundamental matrix may be constructed from the projection matrices of the two views it connects:

$$ F_{jk}^\top = [e_{kj}^x] P_k P_j^\top. $$

$F_{jk}$ is linear in $v$ and all possible solutions for $F_{jk}$ span the subspace of possible fundamental matrices that will form a consistent triplet as defined in Condition (1) [27]. We solve for $v$ that yields $F_{jk}$ closest to $F_{ijk}$. We define the triplet projection error as the difference of $F_{jk}$ and $F_{ijk}$ by Frobenius norm:

$$ Err_{ijk} = ||F_{jk} - F_{ijk}||. $$

References


A. Triplet Projection Error

We define here the triplet projection error used in Section 4.1. Given three views, $i$, $j$, and $k$, and the corresponding fundamental matrices $F_{ij}$, $F_{ik}$, and $F_{jk}$, Sinha et al. [27] corrects the viewing graph and enforces global consistency via loop constraints before applying SfM. We demonstrated that this optimization improves the quality of relative geometries in the viewing graph and removes the need for complex filtering steps as part of the SfM pipeline. Our viewing graph optimization works on calibrated or uncalibrated image sets and we provide a new method for calibrating cameras from a set of fundamental matrices. We incorporate the viewing graph optimization and focal length calibration into a global SfM pipeline that is intuitive to understand and easy to implement, and showed that this pipeline achieves greater efficiency and comparable accuracy to the current state-of-the-art methods. For future work we plan to examine the guarantees we can make (if any) on the “consistency” of the viewing graph we obtain from the viewing graph optimization. Additionally, it would be interesting to see if our method may be applied for global SfM on projective reconstructions.

Acknowledgements: This work was supported in part by NSF Grant IIS-1219261, ONR Grant N00014-14-1-0133 and NSF Graduate Research Fellowship Grant DGE-1144085.


