3D reconstruction software comparison for short sequences

Adam Strupczewski, Blażej Czupryński
Warsaw University of Technology

ABSTRACT

Large scale multiview reconstruction is recently a very popular area of research. There are many open source tools that can be downloaded and run on a personal computer. However, there are few, if any, comparisons between all the available software in terms of accuracy on small datasets that a single user can create. The typical datasets for testing of the software are archeological sites or cities, comprising thousands of images. This paper presents a comparison of currently available open source multiview reconstruction software for small datasets. It also compares the open source solutions with a simple structure from motion pipeline developed by the authors from scratch with the use of OpenCV and Eigen libraries.

Index terms.

3D reconstruction, 3D modelling, multiview reconstruction.

1. INTRODUCTION

There are several open source applications performing 3D reconstruction which can be found on the internet. These include most importantly Bundler,\(^1\) V3DSfM\(^2\) and OpenMVG.\(^3\) In this paper, all three of these systems have been tested and evaluated in terms of the reprojection error, the number of vertices they produce, and the speed.

There are also systems that provide binaries, but with no access to the source code. These are Samantha\(^4\) and VisualSfM.\(^5\) The first only displays the created 3D model after it is finished, so it was run only for a high-level visual comparison. VisualSfM does provide all the necessary parameters as output, but uses a fixed method of calculating feature points. Therefore, it was also used only for a high-level visual comparison.

Last but not least, the authors have created their own implementation of a basic structure from motion pipeline based on an example found in the web.\(^6\) It uses the OpenCV and Eigen libraries for most of the algebra. Also, some algorithms were written from scratch. A description of this software can be found in section 3.1. The interesting part, however, is how this basic structure from motion pipeline compares to much more sophisticated systems. An evaluation of this is presented in section 5.

2. PROCESSING PIPELINES

Structure from motion has been first solved by Tomasi and Kanade using a factorization approach.\(^7\) The factorization approach inherently assumes orthographic projection and a single camera. It has later turned out that the geometrical approach using fundamental matrices, triangulation and bundle adjustment provides better accuracy. Therefore, during the last decade geometric approaches have been the dominant area of research. All the recent work in the field of multiview reconstruction is based on geometrical methods and can be classified into two categories : incremental and global approach.

2.1 The basic incremental pipeline

Since the beginning of the 21st century, a certain standard for multiview reconstruction steps was established. This has been presented at many conferences, for example CVPR tutorials on 3D vision by Hartley and Zisserman,\(^8\) or later by Pollefeys and Zisserman.\(^9\) There are three steps, as shown in figure 1.

The first step consists of calculating features in all the images and establishing matches. The second step involves choosing an initial pair of images for creating a stereo model. After this, consecutive images are selected to improve and grow the initial 3D model. For each image, the 2D points in the image are matched to 3D points in the model. This allows to estimate the camera pose of the new image. Later, triangulation with other images that were already added to the 3D model provides new 3D points to grow the 3D model. In order to
improve the accuracy of the recovered 3D structure and camera poses, a global optimization procedure known as bundle adjustment is used. The aim of this step is to reduce the reprojection error for all recovered points using non-linear optimization techniques.

2.2 The global pipeline
During the last few years an alternative approach has emerged, known as the global pipeline approach. The most popular tools using it are V3DSfM and OpenMV G. In the global pipeline images are not processed sequentially as in the incremental pipeline. Instead, the camera poses and structure are recovered in single, global steps, as shown in figure 2.

First, image features are calculated and matched for all image pairs. These in turn yield relative essential matrices for all image pairs using the five or eight point algorithm. The next step is calculating all camera poses - first relative ones, and later global ones using the epipolar graph. Finally, once the camera poses are known, the 3D structure can be calculated. Bundle adjustment is typically used at the very end as a refinement step.

3. INCREMENTAL APPROACHES
There are a number of possible incremental approaches to multiview reconstruction and they will be covered in somewhat more detail in this section.
3.1 Simple incremental library

The basic pipeline created by the authors should be mentioned first, as the other open source systems simply expand on the idea. The basic pipeline follows the steps shown in Figure 1, and is shown in more detail in figure 3.

![Figure 3. SFM library](image)

Feature detection and extraction can be performed in various ways. Numerous techniques have been published over the last decade including SIFT, SURF and FAST to name a few. The features can be matched by brute force or using an approximate method such as based on k-d trees. In practice, for larger datasets approximate methods preferred. The most simple implementation of feature extraction and matching can be found in the OpenCV library. After feature extraction and matching, the stereo initialization step is performed. First, a pair of images needs to be selected. This is done according to the description in. The selected pair needs to have many well matched features but at the same time a wide baseline - so the features should not be homography inliers. After the two initial views are selected, the fundamental matrix (F) for them is found using RANSAC and the eight-point algorithm. An implementation of both can be found in OpenCV. After calculation of F, the essential matrix is found assuming known camera parameters. Finally, the essential matrix is decomposed using SVD to obtain the relative camera poses, as described by Hartley and Zisserman in their book. Once the camera poses are known, triangulation is used to calculate an initial estimate of the 3D model. An implementation of Iterative Linear Least Squares Triangulation as described in was found to perform much better than the default OpenCV implementation.

The last step of adding consecutive views is performed iteratively. It starts with an optimisation step known as bundle adjustment which adjusts the camera poses and 3D model structure so as to minimize the reprojection error. Afterwards, the next best view is selected to grow the initial model. This is done based on the number of 2D to 3D matches. The image with the most feature points matched to the existing 3D model is chosen. First, the matches are used to calculate the camera pose. There are a number of algorithms to do this. The authors chose an implementation of the algorithm published by Lepetit et al available in OpenCV. After the new camera pose is known, triangulation can be repeated to calculate new 3D points. Once the new points are calculated and added to the model, the bundle adjustment step is run at a global level. Because of usage simplicity, the SSBA implementation was chosen.

3.2 Bundler

Bundler is a piece of software created by Noah Snavely in 2007. It is virtually the first open source multiview reconstruction software intended for large scale datasets. As such, it is impressively accurate and robust. Bundler is an efficient implementation of the incremental reconstruction pipeline. The reconstruction sequence is very similar to that shown in figure 3. There are however some improvements.
The first improvement is a smarter way of filtering reconstructed points. In the basic approach points are rejected at certain stages based on the reprojection error. Bundler also uses rejection based on how small is the angle between cameras used to triangulate the point. If at any moment this angle is smaller than two degrees, the point is removed. The second improvement is a more clever way of running bundle adjustment. The optimization is performed only after adding a set of new images which have sufficiently many matched feature points with the existing model. The optimization is performed in two steps. First, only the newly added cameras and reconstructed points are adjusted. Next, all cameras and the whole 3D model is adjusted. This is more efficient than adjusting everything each time.

3.3 OpenMVG - Incremental toolchain

OpenMVG is a library intended for computer vision researchers maintained by Pierre Moulon. It includes both an incremental and global pipeline. The incremental pipeline is quite similar to that used by Bundler, with one significant difference - the standard RANSAC procedure is changed to a contrario RANSAC, known as AC-RANSAC. The standard RANSAC procedure repeatedly selects random sample sets from the data, whose minimal size is sufficient to estimate the parameters of a model. At each trial, inliers are defined as the data that fits the model within an acceptable error threshold, which is fixed. After a given number of iterations, the model parameters that maximize the number of corresponding inliers are returned.

The main idea of the AC-RANSAC algorithm is to adaptively select error thresholds for each usage. In case of fundamental matrix estimation, homography calculation or camera pose calculation the thresholds used for the RANSAC algorithm are very important. Setting them to fixed values can only be optimal for one kind of data. The authors of claim that their system clearly outperforms Bundler on certain datasets because the typical threshold used in RANSAC is not very good for them.

3.4 VisualSfM

VisualSfM is the most recent development in terms of purely incremental structure from motion software, developed by Changchang Wu and available in binary form. As described in, the focus of the author was to find the limit of incremental structure from motion in terms of efficiency without changing the sequentiality of the approach. The published experiments demonstrate, that a \( O(n) \) time complexity can be achieved in practice for collections of up to 15k images. This is about two orders of magnitude faster than Bundler. The time gain stems from two major contributions. Firstly, preemptive feature matching is introduced. This is a twofold matching strategy which first matches only a few top-scale features between all image pairs, and only performs full matching of features between pairs which seem promising in the first stage. Secondly, the bundle adjustment step is largely modified. A multicore implementation based on conjugate gradients is used. The full bundle adjustment is performed in a geometric fashion depending on the number of processed images. In the meantime, local bundle adjustment for the last 20 added views is performed at fixed intervals.

Despite the focus on speedwise improvements, VisualSfM also introduces a new step that reportedly improves the reconstruction quality. This is called the re-triangulation step. Its purpose is to refine cases when the camera pose is poorly estimated despite many feature matches. This is a typical problem for SFM systems and the retriangulation strategy developed for VisualSfM seems to be a good way of dealing with it.

3.5 Samantha

Samantha is a piece of software provided by the 3dFlow company only in binary form. It is worth mentioning, however, as it represents the evolution of the incremental multiview reconstruction pipeline towards the global pipeline. Despite that the source code is not available, publications by the creators of Samantha give insight into the algorithms utilized by the application. Instead of performing a traditionally sequential reconstruction, Samantha performs a hierarchical one. First, features are matched for all image pairs in a two-stage approach similar to that of VisualSfM. Next, homographies and fundamental matrices are calculated for each well-matched pair. These are later refined using least squares. After this stage pairwise image similarity is calculated based on the area covered by matched keypoints in the images. The pairwise similarity is used to create a binary cluster tree - called a dendrogram. An example dendrogram is shown in figure 4.
The clusters in the dendrogram are clustered in an agglomerative bottom-up fashion. There are three types of operations possible: two-view reconstruction, one-view addition and cluster merging. The main novelty of the approach lies in the third step. The clusters are merged based on their similarity using MLESAC for model fitting. The similarity between two clusters is determined by the similarity of two closest views. An improvement of the cluster selection algorithm was later designed to obtain more balanced trees.

The hierarchical structure from motion pipeline has several advantages over the sequential approach. Because of the treelike structure fewer merges are performed, thus reducing the time complexity by an order of magnitude. More importantly however, using many `seed` image pairs removes the bias and error accumulation typical for the sequential approach. It no longer matters which two images are chosen to create the first 3D model. All images are equally privileged, which leads to more accurate final solutions. One of the easily observable effects on large datasets is the much better ability of Samantha to preserve straight lines compared to Bundler. If the lower-level improvements used in VisualSfM were combined with the hierarchical pipeline of Samantha, a better system than all seen so far could perhaps be created.

4. GLOBAL APPROACHES

Unlike incremental approaches, global approaches solve the SfM problem in two steps. The first step computes global camera rotations for every view, while the second step computes the camera translations and structure. Similarly to the approach described in section 3.5, global approaches are not biased by choosing an initial pair of views, while are also capable of distributing the drift error evenly among all measurements.

4.1 V3DSfM

V3DSfM is a piece of software now only available on Github. There used to be a website explaining the software but it is currently offline. Based on reverse engineering and the analysis of publications however, it is possible to deduce the pipeline followed in the software. It is obviously a global multiview reconstruction pipeline, probably the first open source implementation of this kind.

The first performed step is computing matches between images using SIFT points and calculating essential matrices. These give the relative pairwise rotations and translations. Next, Bayesian inference is used to remove outlier rotations. Later global rotations are computed by exploiting the cyclic structure of the data (loops). After this, camera translations are computed from trifocal tensors. Finally, once the camera positions are known, 3D structure can be found from triangulation. At the end, a bundle adjustment step is performed for all recovered data using SSBA.

4.2 OpenMVG - Global toolchain

As has already been mentioned, OpenMVG can be run in both a sequential and global set-up. The global set-up used in the software was published very recently and seems to be a somewhat improved version of the approach described in the previous section. The first improvement is using AC-RANSAC instead of standard RANSAC for model estimation when the essential matrices are calculated for all pairs of images. The cycle length weighting scheme from is added to improve global rotation estimation. Furthermore, a new trifocal tensor estimation
method based on the $l_{\infty}$ norm is introduced. It is shown to improve the efficiency and robustness of camera translation computation. Last but not least, the bundle adjustment step is performed in two stages. First only translations and structure are refined, with the rotations fixed. This is because the rotations in the global reconstruction pipeline are estimated with much better accuracy than other parameters. Only in the second stage all reconstructed parameters are optimized. The ceres-solver library from Google is used for bundle adjustment.

Experiments published by the authors show that OpenMVG in the global setup achieves an absolute accuracy nearly an order of magnitude better than Bundler for large datasets. It also runs an order of magnitude faster, slightly faster than VisualSfM despite using a slower bundle adjustment algorithm.

5. EXPERIMENTAL COMPARISON

5.1 Test environment and datasets

The results of the experimental comparison were obtained using available open source software and a library developed by the authors. Both incremental (Bundler, OpenMVG incremental, SfM library) and global (V3DSfM, OpenMVG global) approaches were evaluated. Tests were performed on three different datasets. Each dataset is a sequence of 20 images of the same scene, captured at various poses using one camera. The first dataset - books - presents an artificial scene containing books on a desk. The simple geometry of this scene allows to evaluate the quality of reconstructing flat surfaces and angles. The second test sequence, named PW out, contains the facade of the main building of Warsaw University of Technology. It is the closest to a typical 3D reconstruction dataset. The images were taken during a walk in front of the building. The last dataset - PW in - presents an indoor scene. Reconstruction of this scene is a challenging task due to plenty of repetitive structures and sophisticated geometry of the hall’s galleries.

![Exemplary input images and obtained 3D models](image)

Figure 5. Exemplary input images and obtained 3D models

Results of the structure from motion algorithms strictly depend on the first step of the pipeline - feature extraction. To make the comparison more reliable, the same feature points were provided as an input for each piece of software. All implementations were modified to read a unified file representation of the SIFT feature point sets. Further steps (including feature point matching) remained unchanged. SiftGPU was used as a feature point extractor. In order to perform the 3D reconstruction comparison, a simple test framework in Python was developed. It contains scripts to run each piece of software, providing parameter control, timing and result logging. Each test can also be executed on a subset of all the images of a sequence.
The open source reconstruction tools were compared in two areas: quality and speed. The basic test scenario assumed measuring the quality of the result depending on the number of input frames. Additionally, the influence of image resolution was analyzed.

5.2 Reconstruction quality comparison

The goal of optimization algorithms, which are crucial for every structure from motion algorithm, is the minimization of the reprojection error. Thus, theoretically, the quality of the reconstruction process can be measured with the reprojection error. Unfortunately, it is possible to get completely incorrect models and achieve small reprojection errors at the same time. This poses a problem in the interpretation of test results. To overcome this, each generated 3D model in the software comparison was inspected visually. To further evaluate the model quality, the number of reconstructed points is also taken into account. In many cases the number of points is strictly related to the reprojection error. It is easier to achieve small reprojection errors when most of the points are filtered out, and only a small number of points remain. Therefore, algorithms yielding more points with the same reprojection error are preferred.

The dependence of the reconstruction quality on the number of input images was also analyzed. Initially, a subset of four most representative views was chosen from the test sequence. The reconstruction process was executed iteratively and two consecutive images were added at each iteration. This experiment examined the efficiency of scene reconstruction using few views. The sequence PW out is the easiest to reconstruct due to simple geometry, uniform illumination and a lot of feature points, which can be robustly matched. For most algorithms, the reprojection error is almost constant in the function of sequence length and has a small value around half a pixel. Measurements are presented in figure 6. The fixed value of the reprojection error is caused by outlier filtering, which is an important step of every structure from motion pipeline. Points with the large reprojection error are discarded during the reconstruction process to maintain an intended total reprojection error. This operation directly influences the number of reconstructed points. Figure 7 illustrates the dependence of the number of images and the number of reconstructed points for the PW out dataset. All compared algorithms work well and recover a similar number of vertices.

![Figure 6. Reprojection error comparison](image)

The situation changes greatly when a more challenging scene is reconstructed. The relationship between the number of views and the number of reconstructed points measured on the books sequence is presented in figure 7. Incremental approaches work well even for very few views. Bundler performs best for all numbers of views. The
Figure 7. Number of reconstructed vertices dependence on number of views
incremental OpenMVG pipeline also performs well. In both cases adding images results in more reconstructed points, and the gain is near to linear. For all measurements, incremental methods properly calibrate and use all the views. An exception is the SFM library developed by the authors. For the *books* sequence it only starts to work properly beginning from 14 views. This is probably because of less accurate camera pose estimation and triangulation algorithms used compared to other software. For the *PW out* sequence it works best for 10 views, and later deteriorates. This is once again caused by less accurate algorithms and the crude point filtering mechanism based solely on reprojection error. On the *PW in* sequence the SFM library failed completely, which is why the results are not shown on the graphs.

Contrary to incremental methods, global methods fail for small numbers of views. V3D software computes the model starting from 6 views. However, in the range from 6 to 12 views it yields only a partial reconstruction, where only 3 to 5 views are used. For a larger number of views V3D software aligns all the views properly and performs comparably to incremental methods. Global OpenMVG works only for higher numbers of images, but even then it doesn’t align all the views (5/14, 13/18 and 16/20 used/total views). Poor results of global methods are caused by large camera pose differences in the sequences. Global approaches ensure accuracy of the relative pose between cameras by building a cyclic structure with all camera poses. In case of big pose differences between the views some of them are rejected. Moreover, when the reconstruction is performed on a very small number of views, the cycle cannot be completed and the global alignment fails. On the *PW out* dataset camera pose differences are smaller, so both incremental and global algorithms perform similarly (figure 7). For this sequence, all views are aligned and used regardless of how many images are available. The last test sequence is *PW in*. Similarly as for the *books* dataset, incremental methods outperform global ones, especially for a small number of views. Repetitive structures in the scene generate a lot of falsely matched feature points. In this case only Bundler managed to successfully align all the views. Surprisingly, the global algorithm from OpenMVG fails even for a large number of views. This is probably a result of falsely matched feature points. Similarly to global methods, incremental OpenMVG aligned only part of the cameras when the sequence was limited to less than 10 images.

To evaluate the relation between reconstruction quality and image resolution, the number of reconstructed points was measured for three resolutions and presented in figure 8. Using higher image resolutions affects mainly incremental methods. Bundler shows very high and nearly linear increase in the number of reconstructed points along with the increase of the input image sizes. On the other hand, global methods fail to produce better results for higher resolutions. For instance, the number of vertices recovered by V3DSfM for the *PW in* sequence is even smaller than that for the lowest resolution. This is most likely caused by small repetitive objects in the scene, which have great impact on the accuracy of feature point matching. In general, the small feature points visible at high resolution can be unstable and hard to match. Additionally, the closed loops which are so important for global methods may be harder to obtain with multiple repeating structures in each image.

![Figure 8. Resolution](image-url)
5.3 Speed comparison

The compared software performance was evaluated by measuring reconstruction time. Feature point detection and description, as a common step of all compared approaches, was excluded from the measurements. However, feature point matching was treated as an integral reconstruction step and included in timing measurements. Each tested pipeline incorporates an algorithm for filtering matched points, based on relative camera pose estimation. Moreover, global methods implement computationally expensive feature point matching algorithms, which take up most of the reconstruction time. All the measurements were performed on a standard laptop PC equipped with Core 2 Duo T9300 CPU. Results of the speed comparison for all sequences and three image resolutions are presented in table 1. The dependence of execution time on sequence length is presented in figure 9. In most cases, V3DSfM or Bundler is the fastest choice. However, bearing in mind the number of recovered points from figure 8, Bundler has a significantly higher point number to execution time ratio. Generally, the SFM library is about two to five times slower than other tested implementations depending on the dataset size. This is caused by suboptimal routines from the OpenCV library and lack of optimization for a large number of input images. Among global methods, V3DSfM is faster than global OpenMVG. The reason is GPU optimization in V3DSfM, which utilizes CUDA at several steps. Among incremental approaches, OpenMVG is slightly slower than Bundler. Once again the number of reconstructed points should be taken into account in this comparison, where Bundler recovers a few times more vertices than other algorithms, especially at the highest resolution.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Resolution</th>
<th>Bundler</th>
<th>Inc. OpenMVG</th>
<th>V3DSfM</th>
<th>Glob. OpenMVG</th>
<th>Our Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>PW out</td>
<td>640x480</td>
<td>34.6</td>
<td>33.7</td>
<td>47.3</td>
<td>49.9</td>
<td>63.1</td>
</tr>
<tr>
<td></td>
<td>1280x960</td>
<td>74.2</td>
<td>100.7</td>
<td>99.8</td>
<td>106.8</td>
<td>191</td>
</tr>
<tr>
<td></td>
<td>1920x1440</td>
<td>107.7</td>
<td>116.8</td>
<td>92.8</td>
<td>127.4</td>
<td>543</td>
</tr>
<tr>
<td>Books</td>
<td>640x480</td>
<td>10.9</td>
<td>10.1</td>
<td>8.8</td>
<td>19</td>
<td>73.4</td>
</tr>
<tr>
<td></td>
<td>1280x960</td>
<td>28.4</td>
<td>34.5</td>
<td>18.6</td>
<td>44.9</td>
<td>209.3</td>
</tr>
<tr>
<td></td>
<td>1920x1440</td>
<td>50.6</td>
<td>91.2</td>
<td>33.2</td>
<td>61.4</td>
<td>601</td>
</tr>
<tr>
<td>PW in</td>
<td>640x480</td>
<td>24.6</td>
<td>17.2</td>
<td>28.8</td>
<td>38.5</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1280x960</td>
<td>64.6</td>
<td>75.2</td>
<td>78.1</td>
<td>79.2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1920x1440</td>
<td>110</td>
<td>95</td>
<td>75.9</td>
<td>121.3</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 9. Reconstruction speed comparison
6. CONCLUSIONS

First of all, evaluating reconstruction quality using solely the reprojection error is not reliable. Moreover, comparing results of different algorithms using only the reprojection error has little sense because of large variations in number of reconstructed 3D points. The intended error can be obtained by outlier point filtering during the reconstruction process. In order to evaluate the reconstruction quality correctly, datasets with 3D ground through should be used.

The experiments conducted in this comparison have focused on the ability to recover 3D models from short sequences. Incremental methods perform better in this use case, simply because global methods need many images that will allow the creation of closed loops with relative estimations. Only then global values can be recovered correctly. Despite many new approaches, it seems that for short sequences Bundler can still be considered as the state-of-the-art reconstruction software. This is probably thanks to good and accurate algorithms used for point triangulation and camera pose extraction.

Also, for small datasets, speed improvements typical for global methods can hardly be noticed. The overhead for set-up and initialization is evidently larger than the speed gain if only several input images are available, and incremental methods clearly remain the better choice.

7. SUMMARY

In this paper, several major open source structure from motion implementations have been presented and compared. The basic concepts of incremental and global SFM approaches have been described. A simple SFM library created by the authors was also introduced. In the experiments section, available open source SFM toolchains and the self-developed library were compared. Reconstruction quality, speed, and ability to perform reconstruction for a small number of views have been evaluated. The general conclusion is that small scale usage and large scale usage are two different problems, and different approaches are best for each. For small datasets incremental methods still perform better than global methods, with Bundler producing the most accurate results from the compared group.

REFERENCES