Abstract. In this paper we propose, design and implement a method of Bundle adjustment (BA) incorporating constraints that describe a camera undergoing a circular motion. Using these constraints we are able to capture the physical properties of various rotating camera systems into the BA process, for example panoramic camera systems used on Mars rovers or turntables. We incorporated our method into the recently released Google non-linear least squares solver Ceres. By using constrained BA, we aim at improving the accuracy of reconstructing both the 3D points and the camera positions. The improvement in accuracy was experimentally verified on synthetic as well as real datasets.

1. Introduction

There has been a lot of attention given to the Structure from motion reconstruction, resulting in many practical applications such as Photosynth and Bundler [13]. Current research aims to provide more precise reconstruction as well as the ability to handle larger datasets [1], [3].

Bundle Adjustment (BA) [14] is an important part of the structure from motion reconstruction as it optimizes the resulting estimates of 3D point coordinates and the position, orientation and calibration of cameras. Detailed analysis of BA optimization methods, parametrizations, error modeling and constraints has been given in [14]. An efficient and comprehensive algorithm that utilizes the sparsity of BA has been developed by Lourakis and Argyros [10] and the code is made freely available. This algorithm has been further used in [13], to build a full structure from motion pipeline. An extended version of [10] has been developed in [2] utilizing the sparsity even further in order to reduce computation time. Recently, the performance of BA on large datasets has been scrutinized [1]. The use of conjugate gradients and its effect on performance has been investigated in [2]. In [6], significant performance improvements using multiple techniques, such as embedded point iterations and preconditioned conjugate gradients were shown.

1.1. Constrained Bundle Adjustment

A general graph solver which can be used for BA is described in [12] and available online. Although [12] seems to be capable of handling camera constraints, authors have not investigated this possibility. BA with constraints on the structure has been used in [15] and [1]. Experiments with camera constrained BA designed specifically for stereo rig were carried out in [8]. The results showed that modeling stereo camera pair with proper constraints and incorporating it into BA can improve the precision of the reconstruction. In [4], authors incorporate constraints for a Pancam system consisting of two cameras mounted on a rotating shaft used on a Mars Exploration Rover mission. BA designed for a turntable has been presented in [15], where authors described the object as a set of points rotating around an axis with a fixed distance from the camera. They showed that this projection model brought an improvement in terms of precision of the reconstruction.

1.2. Contribution

In this paper we provide a general approach to describe a rotating camera and incorporate it into Bundle Adjustment optimization implemented in the open source solver Ceres from Google. In comparison to [4] and [15] our method is not limited to a specific Pancam system or a turntable even though it is capable of handling both of them. Using our approach we can optimize wide variety of scenarios, such as multiple panoramic cameras in one scene,
cameras undergoing circular motions with different radii, orientations and on different locations or turntables with more than just one camera position. This is done by adjusting the camera model, keeping only the circular motion constraints. The initialization does not require any previous knowledge about the scene as in the case of [4], instead we estimate the constraints from an initial unconstrained SfM result. In our experiments we confirm that using just the constraints from an initial unconstrained SfM reconstruction does not require any previous knowledge about the scene as in the case of [4], instead we estimate the initial radii, orientations and on different locations or turntables with more than just one camera position. Such systems can be a camera rotating around an axis, where our additional knowledge is represented by the fact that camera centers lie on a circular trajectory. Each camera center $c_i$ can therefore be expressed by a function involving center of rotation, the orientation of the rotation plane in space, the radius of the rotation and a displacement angle of the camera. Such equation can be written as

$$P_i = R_e [I - C_i]$$

When we express the rotation matrices using quaternions, we obtain the following parameter vector

$$a_j = [k, q, C_c^T, q_c, \rho]$$

where $k$ is the calibration matrix, $R$ is a rotation matrix from world to camera coordinates and $C$ is the camera center in the world coordinates. According to [5], the calibration matrix can be described by 5 parameters as

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & ar\alpha_x & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where

$$ar = \frac{\alpha_y}{\alpha_x}$$

is the aspect ratio.

3. Camera systems with rotational constraint

When dealing with multiple camera systems, which have some fixed physical properties, it would be meaningful to utilize this additional information in the BA process. We did that by using an adapted projection function describing the relationships between cameras. Such systems can be a camera rotating around an axis, where our additional knowledge is represented by the fact that camera centers lie on a circular trajectory. Each camera center $c_i$ can therefore be expressed by a function involving center of rotation, the orientation of the rotation plane in space, the radius of the rotation and a displacement angle of the camera. Such equation can be written as

$$C_i = R_e \begin{bmatrix} \rho \cos \phi_i \\ \rho \sin \phi_i \\ 0 \end{bmatrix} + \begin{bmatrix} C_{cx} \\ C_{cy} \\ C_{cz} \end{bmatrix}$$

where $R_e$ is a rotation matrix representing the orientation of the panoramic rotation plane in space, vector $C_c = [C_{cx}, C_{cy}, C_{cz}]^T$ represents the center of rotation, $\rho$ the radius and $\phi$ the camera displacement angle. Having $C_i$ we can construct $P_i$ following

$$P_i = R_i [I - C_i]$$

In section 2 we describe the basics of BA and introduce the notation we will use further in the paper. Section 3 presents the model used for rotating camera systems. Experiments and their results are shown in section 4 and the results discussed in section 5.
so the blocks denoted by index \( i \) belong to each individual camera and blocks denoted by \( j \) describe the circular constraints for a cluster of cameras. It is clear, that parameters describing the panoramic rotation \((C_c, q_c, \rho)\) will be identical for all cameras from the same panorama. It is possible within the Ceres platform to make any of these blocks shared by multiple cameras. The reason why we chose such blocks is that we can set any combination of radius, circle orientation and circle position shared also between multiple camera clusters. As an example, Pancam in \[4\] could be described by two camera clusters with shared rotation axis position and orientation but different radii.

![Parameter Diagram](image)

Figure 1. Visualization of parameters used to describe panoramic camera. Parameters \( C_c, R_c \) and \( \rho \) describe the circle of rotation, therefore they are shared among all cameras within this panorama. \( C_i \) and \( \phi_i \) determine the positions of individual camera centers and \( R_i \) their orientations.

4. Experiments

To validate our method, we performed a set of tests on synthetic and real data. We compared our extended bundle adjustment (XBA) algorithm to SBA, upon which our algorithm was built.

4.1. Parametrization

Shared parameters in XBA were set to utilize all physically meaningful properties of the camera system. In the case of panoramic cameras, the camera centers were placed on circles and all circles shared the same radius. Stereo setup used a shared baseline and orientation of the second camera with respect to the first one over the whole sequence. Both SBA and XBA adjusted 3 internal camera calibration parameters \((f_x, x_0, y_0)\) and kept the 2 remaining fixed \((a, r, s)\). We also estimated the radial distortion using the same model as \[13\]. The overall number of parameters describing a camera was 14 (five for intrinsics, four for camera orientation, three for camera center location and two for radial distortion) for SBA and 20 (as in \[8\] plus two for radial distortion) for XBA. Because two intrinsic were kept fixed, the numbers of parameters adjusted were 12 and 18 for SBA and XBA respectively. The XBA parameters were described in section \[3\].

4.2. Datasets

We examined the performance of XBA on four real world dataset captured using a panoramic camera setup. Pictures were acquired using a Nikon D60 and D3100 DSLR cameras. First dataset was simulating a panoramic observation of environment, using a camera rotating on a shaft mounted on a tripod. Three panoramic observations were made on three different locations, each consisting of 18 images of an urban environment. The data was then reconstructed using the SfM pipeline \[13\] and estimates for camera and structure parameters were obtained. The rest of the real world datasets was obtained using a turntable with two objects each with different shape properties. The turntable was used to capture the model from various viewpoints, while keeping the circular trajectory of the camera with respect to the object.

To evaluate our method, we needed some ground truth information about the scenes. For that purpose we took the outcome of the SfM pipeline, placed the cameras on circular trajectories (as they were in reality) and projected the 3D points into new image projections. Therefore, we obtained a perfect ground truth information in a real world scenarios.

In all datasets, the 3D points were perturbed by a uniformly distributed error \( e \in [0; d/20] \), where \( d \) is the distance from the camera centers. The image projections were perturbed by a Gaussian noise with mean 0 and standard deviation of 0.7.

<table>
<thead>
<tr>
<th>Table 1. Datasets</th>
<th>cameras</th>
<th>points</th>
<th>projections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dino1</td>
<td>40</td>
<td>14038</td>
<td>42241</td>
</tr>
<tr>
<td>Dino2</td>
<td>20</td>
<td>3080</td>
<td>7343</td>
</tr>
<tr>
<td>Ship</td>
<td>13</td>
<td>8851</td>
<td>29259</td>
</tr>
<tr>
<td>Street</td>
<td>54</td>
<td>34443</td>
<td>93348</td>
</tr>
</tbody>
</table>
4.3. Analysis

In measurements we analyzed the following aspects:

1. Error in reconstructed structure with respect to ground truth
2. Error in reconstructed camera positions with respect to ground truth
3. Evolution of the reconstruction error during iterations
4. Evolution of the mean reprojection error

To express the reconstruction error, we used a least square fit of either the resulting 3D point positions or the camera center positions onto the ground truth data. Camera fit means that a transformation was found between the reconstructed camera centers and the ground truth camera centers and this transformation was then applied to reconstructed camera centers as well as 3D points. Points fit applies the same principle but the transformation between reconstructed and ground truth 3D points was found in this case. After the fit, error in both camera centers and 3D points was measured as the Euclidean distance from the correct positions. These measurements express how accurately were the camera positions and 3D points reconstructed and also how accurately can we determine the camera positions knowing the 3D structure parameters and vice versa. This evaluation was also carried out for each individual BA step to show how the accuracy of the model develops during the process. For each dataset, the tests were run multiple times with different random initialization.

Results of reconstruction error of camera centers are in figure 4 which shows, how precisely we estimated the surrounding structure relatively to the camera positions, e.g. their relative scale. The precision of reconstructed structure with respect to the ground truth can be read from figure 5. Analogically to the camera centers’ case, figure 5 now show how precisely we can determine the camera centers relatively to the structure.

To investigate the behavior of XBA versus SBA deeper, we show for each iteration on the dataset Dino2 the mean reprojection error and the reconstruction error in Figure 6.

5. Discussion

As shown in figure 3 more dimensions of freedom cause SBA to achieve lower image reprojection error,
but the constraints ensure that the 3D reconstruction error becomes lower using XBA.

The results, summarized in table 2, show that XBA can in certain scenarios outperform SBA in terms of reconstruction precision. Particularly, in the dataset Street the reconstruction errors of XBA were much smaller than of SBA. As can be seen in the results, the difference between XBA and SBA varies with different initializations. On rare occasions, SBA achieves similar results as XBA or even slightly better. However, on average, the reconstruction using

Figure 3. **Real PANCAM.** (a) The image reprojection error during iterations of SBA and XBA on the dataset Street. (b) Development in 3D point and camera position error during iterations of SBA and XBA on the dataset Street.

XBA was more correlated with the ground truth.

This could be useful in many applications, such as

Figure 4. **Camera position error.** Error in camera center positions after reconstructing and fitting through 3D point positions for datasets (a) Dino1, (b) Dino2, (c) Ship and (d) Street.
Table 2. Summary of the results for all datasets

<table>
<thead>
<tr>
<th></th>
<th>SBA</th>
<th>XBA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cams</td>
<td>pts</td>
</tr>
<tr>
<td>Dino1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dino2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ship</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Street</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5. **Camera position error.** Error in camera center positions after reconstructing and fitting through 3D point positions for datasets (a) Dino1, (b) Dino2, (c) Ship and (d) Street.

localization and mapping. If one knows information about the dimension of either the structure or camera positions (for example the diameter of panoramic rotation), one can better estimate the dimension of the other.

For the datasets acquired using a turntable the performance of XBA was not always better than SBA as in the case of the Street dataset. In dataset Dino1 XBA handled better the reconstruction of the scene with respect to the points and in dataset Dino2 with respect to the cameras. Since Dino1 has far less cameras, points and projections, this could be interpreted as the ability of constrained BA to maintain the correct positions of cameras even with less observations and interconnectivity of the cameras and points in the scene.

Another situation occurs in the case of the Ship dataset. The points fit slightly better to the ground truth cameras using SBA, but points with respect to points and cameras with respect to either cameras or points are reconstructed better using XBA by a large margin. In this scenario, the camera circle is purposely incomplete and, as expected, XBA handled the camera positions much better. This could be useful if we are interested in precise poses of cameras, such as in the case of dense stereo reconstruction.

However, it can be concluded, that XBA does not bring such an improvement in the case of turntable scenarios, where all cameras are pointing towards a single object and the scene is heavily interconnected, i.e. cameras have high overlaps and each point is observed by many cameras. The potential of XBA seems to be more in the outdoor scenarios, where cameras are pointing outwards of the axis of rotation and have smaller overlap.
6. Conclusion

We proposed and implemented a new method of constrained BA (XBA) for cameras undergoing circular motion into Google optimization software Ceres. This method incorporates circular constraints for systems such as panoramic camera or a turntable. The method was validated on several real datasets and the reconstruction error in camera positions and 3D points was measured. The results show for certain scenarios improved quality of the reconstruction over the previous method. The advantages of this method and its possible applications were discussed.

Acknowledgements

The authors were supported by Grant Agency of the CTU Prague project SGS12/191/OHK3/3T/13.

References


