

# Sparse Bundle Adjustment Speedup Strategies

Christian Lipski  
TU Braunschweig,  
Germany  
lipski@cg.tu-bs.de

Denis Bose  
TU Braunschweig,  
Germany  
bose@tu-bs.de

Martin Eisemann  
TU Braunschweig,  
Germany  
eisemann@cg.tu-bs.de

Kai Berger  
TU Braunschweig,  
Germany  
berger@cg.tu-bs.de

Marcus Magnor  
TU Braunschweig,  
Germany  
magnor@cg.tu-bs.de

## ABSTRACT

Over the past years, Structure-from-Motion calibration algorithms have become widely popular for many applications in computer graphics. From an unordered set of photographs, they manage to robustly estimate intrinsic and extrinsic camera parameters for each image. One major drawback is the quadratic computation time of existing algorithms. This paper presents different strategies to overcome this problem by only working on subsets of images and merging the results. A quantitative comparison of these strategies reveals the trade-off between accuracy and computation time.

**Keywords:** Camera Calibration, Sparse Bundle Adjustment, Structure-from-Motion.

## 1 INTRODUCTION

Many of today's vision and graphics applications are based on well-calibrated cameras. The camera calibration process has been widely explored in the past years and many methods have been proposed - ranging from classical checkerboard recordings to calibration without a priori known patterns [PGV<sup>+</sup>04, SSS08]. These recent methods require the recorded images only to obtain a multitude of feature points (e.g. SIFT-features) for a proper *self-calibration*. Especially image-based modeling and rendering applications benefit from the development: The camera setup can be freely chosen and a calibration recording session has become obsolete. Furthermore, the camera setup does not need to be fixed during the recording anymore. Scenes recorded with multiple handheld cameras can nowadays be reconstructed by employing the self-calibration methods. The method most widely used in the research community is the Sparse Bundle Adjustment, or *Bundler* for short, introduced by Snavely et al. [SSS08]. The recorded images are searched for feature points, e.g. SIFT-features. Feature points, that are shared between any two images are considered as correspondence points. After an initial estimate of camera parameters, these points are triangulated and reprojected to the images. The reprojection error, i.e. the euclidean distance between the original feature locations and their reprojections on the image plane is minimized during the so-called bundle adjustment. Being considered as a milestone in the community,

this tool, however, has serious issues regarding the computation time.

In this paper we examine the reasons for these issues and propose new methods to significantly reduce the computation time whilst keeping the reprojection error minimal. The paper is outlined as follows. We give a brief overview to recent advances in calibration methods in Section 2, also focussing on Bundler's runtime issues. Afterwards, we introduce two strategies to tackle these problems in Section 3. We justify our methods with a quantitative analysis in Section 4 and conclude in Section 5.

## 2 RELATED WORK

While our work mainly improves Bundler by Snavely et al. [SSS08], a renowned tool for 3D object reconstruction from uncalibrated multicamera footage used by many other scientists [WMC04, Sna08, JB09], we also relate to the following previous work in the field of multicamera calibration.

A good overview of calibration algorithms can be found in the paper by Triggs et al. [TMHF99]. The commercial tool *Boujou* [vic09] reconstructs 3D models from moving uncalibrated cameras. Hasler et al. [HRT<sup>+</sup>09, THWS08] calibrate multiple moving unsynchronized cameras by first finding each camera's trajectory (using KLT-tracking and RANSAC-fitting). An approach based on geometric dissimilarity measurement is described by Denzler et al. [BBD09]. They rely on a less restrictive matching method compared to [SSS08].

However, most calibration approaches, including the Sparse Bundle Adjustment [SSS08], suffer from long computation times. Schwartz et al. [SK09] investigate the preconditions of multicamera calibration and suggest to merge connected components for an initial estimate to achieve computation speedup. Byrod et al. suggest an iterative adjusting approach by solving the problem with a conjugate gradient method. They pre-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

condition the matrix with a multiscale Gauss-Seidel approach. He et al. [HQH08] try to improve the computation time by propagating matches between camera pairs.

Our approaches, instead, address the computation time problem by applying Bundler to a limited selection of images, and incorporating the other images at a later stage.

## 2.1 Bundler: Sparse Bundle Adjustment

As our work is based on the work of Snavely et al. [SSS08], we will give a brief introduction into the Bundler Calibration pipeline. Bundler accepts an unordered set of photographs as input, along with an initial estimate of the focal lengths of the cameras that took these images. A calibration of the images is the output of the algorithm which provides the relative rotations  $R$  and translations  $t$  of all cameras along with the intrinsic parameters (focal length and radial lens distortion). The first part of the Pipeline is an image feature extraction. Snavely et al. proposed to use SIFT features [Low04] for this task. This step runs in linear time. A pairwise feature matching phase matches the key features of all images pairs. This step runs in quadratic time. The two most promising images are chosen for an initial calibration. After calibration, an initial set of 3D points is obtained via triangulation of the corresponding points. The bundle adjustment step refines the calibration by minimizing the reprojection errors of the obtained points. The remaining cameras are added one by one: If at least six correspondences to the already reconstructed 3D points are known, an initial estimate of its parameters is calculated via Direct Linear Transformation. A bundle adjustment step refines the initial parameters of the camera, new reconstructed 3D points may be added and a global bundle adjustment step is performed. This final phase runs in quadratic time. We can see that both the key feature matching and the bundle adjustment run in quadratic time with respect to the amount  $m$  of input images. The overall computational complexity of Bundler is therefore  $O(m^2)$ .

## 3 SPEEDUP STRATEGIES

Data sets containing just a few hundred images may lead to run-times of several days on a single CPU. Instead of focussing on algorithmic techniques to tackle this problem, our approaches reduce the number of images used as an input to the sparse bundle adjustment. We developed two different strategies that let Bundler only run on subsets of images, thus decreasing the overall run-time.

### 3.1 Merge Images Approach

We partition the set of images into  $n$  subsets of equal size. Given an (arbitrarily chosen) order of images, the

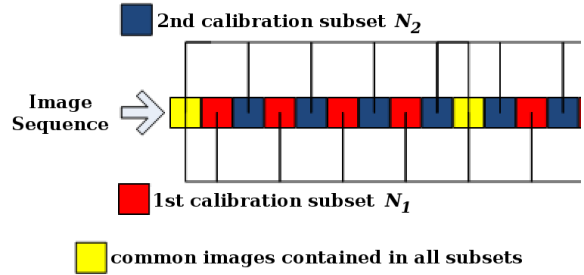


Figure 1: Merge Images Approach for  $n = 2$  and  $k = 9$ . Two subsets are created and separated independently (blue and red boxes). All subsets contain a set of common images (yellow boxes). Both subsets are merged via a Procrustes transformation.

first, the  $n + 1$ st, the  $2n + 1$ st, etc... image are put in subset  $N_1$ . The second,  $n + 2$ nd,  $2n + 2$ nd, etc... image are placed in subset  $N_2$  and so forth, see Fig. 1. Afterwards, we make sure that the image subsets also contain some common images. We select each  $k$ th image from the original image set and add it to each subset if it is not already present in that set. Each subset is calibrated with Bundler independently. We are now faced with the problem that we obtained  $n$  calibrations of the same scene. We arbitrarily pick the first subset to be our reference set and merge the other calibration results into this reference system. The subset's reference systems differ in their location  $z_n$ , their rotation  $R_n$  and their scale  $b_n$ . So, a Procrustes transformation  $\Phi$  has to be obtained for each subset to align it with the reference set. When this transformation is known, new rotation matrices  $R_{new}$  and translation vectors  $t_{w_{new}}$  are obtained. We recall that the position  $p$  of a camera can be derived from its rotation matrix  $R$  and its translation vector  $t$ .

$$p = -R^T t. \quad (1)$$

We can obtain a set of common points for all subsets of images when we compute the camera positions for the common images in each set. For each image subset, we obtain the transformation  $\phi$  that maps the set of common camera locations to the one of the reference calibration. We make use of the matlab implementation of the Procrustes Analysis. The same transformation can be used to obtain the camera locations  $p_{new}$ , the rotation matrices  $R_{new}$  and the translation vectors  $t_{new}$ . The new camera locations and rotation matrices can be derived by directly applying  $\phi$ . The translation vectors are computed as follows:

$$t_{new} = -R_{new}^T p_{new} \quad (2)$$

The speedup caused by this strategy can be formalized by a reduction of the complexity from  $O(m^2)$ , where  $m$  is the total number of images, to  $O(n \cdot (m/n + m/k)^2)$ . As we will show in Section 4, an adequate se-

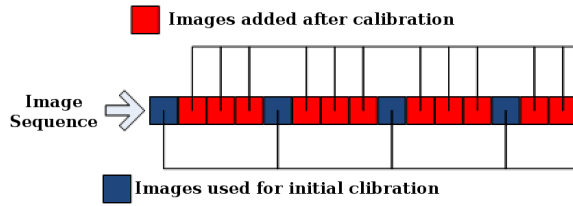


Figure 2: Add Images approach with  $n = 4$ . Only each  $n$ th image is used for the initial calibration (red boxes). The other images are added using via Direct Linear Transformation.



Figure 3: A representative frame of the test sequence.

lection of  $k$  and  $m$  can cause a dramatic speedup, while preserving a high accuracy, i.e. a low reprojection error.

### 3.2 Add Images Approach

The original implementation of bundler provides the opportunity to add images to an already calibrated set of images. We exploit this feature and determine a subset of images that is calibrated instead of the complete set of images. We add every  $n$ th image into the subset, calibrate the subset and add all remaining images via Bundler's *Add Images* feature, Fig 2. When adding images to the calibrated set of images, no new bundle adjustment iteration is performed. I.e., only the optimal rotation matrix and translation vector for the new image is determined, no new 3D points are inserted and no optimization of the camera parameters is performed. Therefore, adding images runs in linear time. Instead of the original computational complexity of  $O(m^2)$ , the Add Images Approach has a complexity of  $O((m/n)^2 + (m - m/n))$ , which is even faster than the Merge Image Approach.

## 4 RESULTS

Our speedup strategies are tested on the graffiti image sequence, Fig. 3. This test sequence contains the recordings of 5 non-stationary camcorders, all pointed towards a juggler in front of a highly textured wall. Each camera recorded 40 video frames, resulting in a total size of 200 images. The image size is 480px  $\times$  270px. We calibrate the set of 200 images with the original bundler algorithm, the Merge Images Approach and the Add Images Approach. Several calibration runs

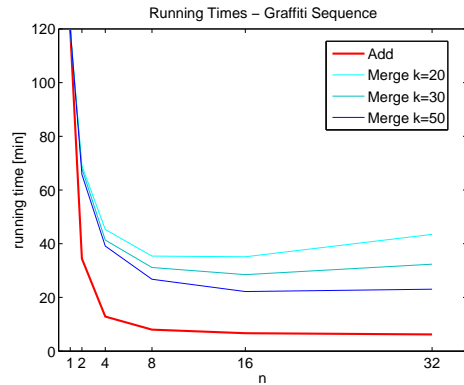


Figure 4: Runtimes for Bundler using both speedup strategies with different parameters  $n$  and  $k$ . We obtained results for  $n = 1, 2, 4, 8, 16, 32$  and  $k = 20, 30, 50$ . Please note that  $n = 1$  is identical to a calibration without speedup. Compared to the original Bundler calibration ( $n = 1$ ), a significant speedup can be achieved in all cases.

with different parameters quantitatively determine the tradeoff between computation time and accuracy.

As an error measure, we use the reprojection error of the reconstructed 3D points. In order to make all speedup scenarios comparable, we have to make a slight alteration to the Add Image approach. When using this approach, the reconstructed point sets tend to be much smaller with increasing  $n$ . Because not all images are used for Bundle Adjustment, less reconstructed points are added. It is also quite likely that only these points will be incorporated into that set that have a low reprojection error: Bundler either optimizes or discards points. Therefore, we store a list of reconstructed 3D points and their image locations when running Bundler without a speedup strategy. When evaluating the reprojection error with the Add Images method, we reconstruct the full set of 3D points by triangulation of the previously stored image locations. We then measure the reprojection error of the full set of 3D points. For both speedup methods, we calibrate with  $n = 2, 4, 8, 16, 32$ . In the case of the Merge Images method, we did individual test runs for each different  $n$  with  $k = 20, 30, 50$ .

The computation times, Fig. 4, reveal that the Add Images Approach outperforms the Merge Images Approach in terms of speed. For  $n = 32$  it takes just 6 instead of 120 minutes to perform the calibration. This is not surprising, as the Merge Images method does run  $n$  separate calibrations instead of only a single one. With computation times as low as 22 minutes, the Merge Images method still achieves a remarkable result. When choosing  $k > n$ , the runtimes start to increase again, as a lot of redundant frames are incorporated into the calibrations. All calibration runs are performed on a 2.66 Ghz Intel CPU using a single core. In defense of the Merge Images method one must admit that the Merge

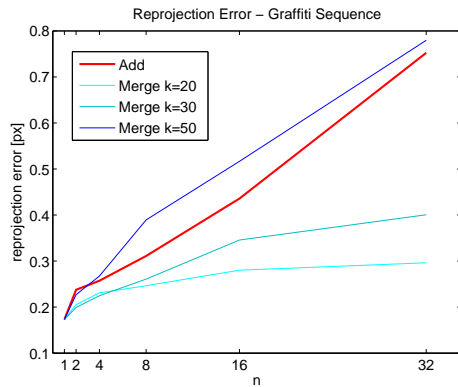


Figure 5: Average reprojection error for both the Add Images and the Merge Images approach. Please note that  $n = 1$  is identical to a calibration without speedup.

Images method can be easily parallelized. In contrast, the Add Images approach runs a consecutive algorithm.

When we look at the reprojection error, one can see that for low  $k$  ( $k = 20, 30$ ) values, the Merge Images Method achieves much better results, Fig. 5. With higher  $k$  ( $k = 50$ ) the merging of data sets seems to become unstable. The Add Images method's reprojection error increases linear with  $n$ . Although, for  $n = 32$  the mean error still stays below 0.8 px.

When we look at the mean deviation of the error, we see that it keeps low in all scenarios where the Merge Approach is used, Fig. 6. On the other hand, the deviation of the error climbs up to a value of 1.6 px when using the Add Images Approach. This can be explained by the fact that many of these points were not considered for bundle adjustment and that a few large outliers exist. The shown quantitative results lead to the interpretation that both approaches succeed in their task to speed up the computation while maintaining a low reprojection error. When a very high speedup is required, the Add Images approach is the first choice, especially for high values of  $n$ , drastic speedups are achieved. When accuracy is crucial, the Merge Images approach is the more advisable choice. One should pick  $n < k$  when using the Merge Images method, otherwise the speedup will significantly diminish.

## 5 CONCLUSION

We introduced two methods, i.e., the Merge Images and the Add Images approach, to speed up the computation in the camera calibration tool Bundler. We found that both methods achieve comparably fair results, i.e. minimal reprojection error.

In the future we want to examine, if clustering of images will lead to further speedup. I.e., if instead of picking images arbitrarily for our calibration subsets, a more considerate preselection of images can be used to further improve the accuracy of the calibration.

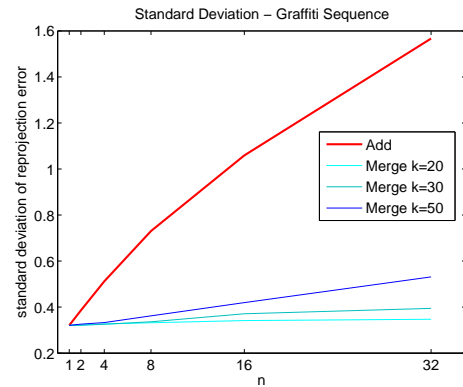


Figure 6: Standard deviation of the reprojection error for both speedup strategies.

## REFERENCES

- [BBD09] M. Brückner, F. Bajramovic, and J. Denzler. Geometric and probabilistic image dissimilarity measures for common field of view detection. *Proc. of CVPR*, pages 2052–2057, 2009.
- [HQH08] S. He, Y. Qi, and F. Hou. Photo Traveler: A System for Exploring Photos in 3D. *Proc. of the JCSIS*, 2008.
- [HRT<sup>+</sup>09] N. Hasler, B. Rosenhahn, T. Thormählen, M. Wand, J. Gall, and H.-P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *Proc. of CVPR*, pages –, Miami, USA, June 2009. IEEE Computer Society. (to appear).
- [JB09] K. Josephson and M. Byrod. Pose estimation with radial distortion and unknown focal length. *Proc. of CVPR*, pages 2419–2426, 2009.
- [Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 60:91–110, 2004.
- [PGV<sup>+</sup>04] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, and R. Koch J. Tops. Visual modeling with a hand-held camera. *Intl. Journal of Computer Vision*, pages 207–232, 2004.
- [SK09] C. Schwartz and R. Klein. Improving initial estimations for structure from motion methods. In *Proc. of the CESC*, April 2009.
- [Sna08] N. Snavely. *Scene Reconstruction and Visualization from Community Photo Collections*. PhD in computer science, University of Washington, Computer Science and Engineering, University of Washington, Box 352350, Seattle, WA 98195-2350, USA, 2008.
- [SSS08] N. Snavely, S.M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *Intl. Journal of Computer Vision*, 80(2):189–210, 2008.
- [THWS08] T. Thormählen, N. Hasler, M. Wand, and H.-P. Seidel. Merging of feature tracks for camera motion estimation from video. In *Proc. of the CVMP*, London, UK, November 2008.
- [TMHF99] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment—a modern synthesis. *Lecture Notes in Computer Science*, pages 298–372, 1999.
- [vic09] vicon. boujou. <http://www.vicon.com/boujou/>, 2009.
- [WMC04] K. H. Wong, M. Ming, and Y. Chang. 3d model reconstruction by constrained bundle adjustment. In *Proc. of ICPR*, pages 902–905, Washington, DC, USA, 2004. IEEE Computer Society.