Ingo Scholz, Heinrich Niemann

Globally Consistent 3-D Reconstruction by Utilizing Loops in Camera Movement

appeared in:
Pattern Recognition, 26th DAGM Symposium (DAGM 2004)
Tübingen, Germany
pp. 471–479

Globally Consistent 3-D Reconstruction by Utilizing Loops in Camera Movement

Ingo Scholz[⋆] and Heinrich Niemann

Lehrstuhl für Mustererkennung, Universität Erlangen-Nürnberg,
Martensstr. 3, 91058 Erlangen, Germany
{scholz,niemann}@informatik.uni-erlangen.de

Abstract. A common approach to 3-D reconstruction from image sequences is to track point features through the images, followed by an estimation of camera parameters and scene geometry. For long sequences, the latter is done by applying a factorization method followed by an image-by-image calibration. In this contribution we propose to integrate the tracking and calibration steps and to feed back already known camera parameters to both tracking and calibration. For loop-like camera motion, reconstruction can thus be optimized by using loop-closing algorithms known from robot navigation.

1 Introduction

Reconstructing 3-D scene geometry and camera parameters from a sequence of images is a common problem in many computer vision applications. One of these applications, for which the approach described in the following was developed, is the computation of light fields [5]. The light field is an image-based scene model where a set of original images is used to render new views of a scene from arbitrary camera positions. Beside image data and geometry information light fields require very accurately determined camera parameters for good rendering results.

If no information is available about the camera pose and internal parameters they are estimated by so-called structure-from-motion approaches [7]. Using feature detection and tracking algorithms point correspondences are established between the images of a sequence. These are used by a factorization algorithm to simultaneously determine the scene geometry (structure) and camera poses (motion) of multiple images. Usually there are not enough point correspondences to process the whole image sequence at once using one factorization, therefore the camera parameters of the rest of the sequence are computed image by image using camera calibration methods. This approach is described in detail in [3].

The main problem arising during this extension process is that, though errors may be small from one image to the next, they accumulate over a large number of images leading to inconsistencies in the geometry reconstruction. In the following we will consider the case that a hand-held camera is moved in loops around a scene, e. g. to view an object from every direction or to get a dense sampling. The

[⋆] This work was funded by the German Research Foundation (DFG) under grant SFB 603/TP C2. Only the authors are responsible for the content.

```
Initialize frame number: $i := 0$
    Track point features to frame $i$ and detect new ones
    $i := i + 1$
UNTIL min. number of features visible in all frames reached or $i = N - 1$
Apply factorization method to first $i$ frames
WHILE $i < N$
    Track point features to frame $i$ and detect new ones
    Triangulate 3-D points and calibrate frame $i$
    $i := i + 1$
Apply bundle adjustment to all frames and 3-D points
```

Fig. 1. Linear tracking and calibration over N images in two steps: factorization of initial subsequence and calibration of subsequent images

approach we will introduce was inspired by solutions in the field of simultaneous localization and mapping (SLAM) for robot navigation. Here, the goal is to generate a globally consistent map of the surroundings of a robot [6], while the data from the robot's sensors, e. g. odometry and a camera, are unreliable. Consistency of the map can be established when the robot returns to a previous position and recognizes landmarks it has seen before. The accumulated error can then be determined and the rest of the map corrected accordingly. For the case of 3-D reconstruction we will now use the occurrence of a loop in camera movement to update the pose of all previous cameras in the loop. The error introduced by this process is reduced by bundle adjustment.

The idea of using topology information to improve reconstruction was implemented before in [4], where a zigzag motion of the camera was utilized to track a feature in an increased number of images. In [1] the accumulated reconstruction error of a turntable image sequence is distributed to all camera position estimates by aligning several sub-sequences. A similar distribution of errors is done in [9] for image mosaics, although in this case the camera motion is constrained to rotations only.

A description of the linear, integrated structure-from-motion approach of tracking, factorization and frame-wise extension will be given in Section 2. The closing of loops by information feedback and optimization is the topic of Section 3, and its experimental evaluation is described in Section 4. A summary and outlook to the future are given in the conclusion.

2 Linear Calibration Process

The usual processing chain for a 3-D reconstruction of a scene is to first generate the required point correspondences for all images followed by the respective algorithms for structure-from-motion. In the work at hand we want to demonstrate the usefulness of feeding back information from the calibration step to the tracking and subsequent calibration. Therefore, tracking and calibration are first integrated into a linear processing chain as shown in Figure 1.

First, feature tracking is done until the number of tracked points reaches a lower bound and a factorization is performed for the images so far. In the second loop the features are tracked to the subsequent images and a camera calibration is applied for each. Thus the camera movement and 3-D points are recovered image by image. Last, the reconstruction is optimized by bundle adjustment on all camera positions and points.

The individual steps of this linear processing chain will be described in more detail in the following, whereas the extension to an iterative process, including information feedback, will be introduced in Section 3.

2.1 Feature Detection and Tracking

In order to get accurate point correspondences over a large number of images feature detection and tracking are performed using the gradient-based algorithm by Tomasi and Kanade [11] and the extension by Shi [8]. In the latter robustness is increased by considering affine transformations for each feature window.

This procedure has been further augmented by a hierarchical approach which computes a Gaussian resolution pyramid for each image, thus increasing the maximum disparity allowed between two images. A final improvement incorporates illumination compensation which solves for many problems occurring in environments which are not particularly lighted [14].

2.2 Factorization and Calibration Extension

For the images in the first block of Figure 1 structure and motion in the sequence are recovered using a factorization method assuming weak-perspective projection [7]. It yields the camera pose parameters for a set of images and the 3-D position of each feature visible in every image. In order to gain a perspective reconstruction of the camera poses perspective projection matrices are constructed from the result of the preceding factorization. Since the intrinsic parameters are unknown the principal point is assumed to be in the image center. For the focal length a rough approximation of the correct one is chosen as described in [3]. Camera parameters and 3-D points are then optimized using the Levenberg-Marquardt algorithm minimizing the back-projection error. Intrinsic parameters are assumed to be constant which results in a small but acceptable error due to the wrongly estimated focal length.

Once this initial reconstruction of the first subsequence is available, it can be used as a calibration pattern for calibrating the subsequent images. Features which are visible in the next image to be calibrated but whose 3-D positions are not yet available are triangulated using their projections in the already calibrated images. With these correspondences the camera position can be estimated using common calibration algorithms [12], and the result is optimized again by minimizing the back-projection error. In fact this optimization is accurate enough so that for small camera movements it can be initialized with the position of the last camera and the calibration step can be omitted entirely.

2.3 Bundle Adjustment

The optimization of the camera parameters and 3-D points in the steps before was always done for one camera after another and in turn with the point positions. In contrast to that the idea of bundle adjustment is to optimize all these

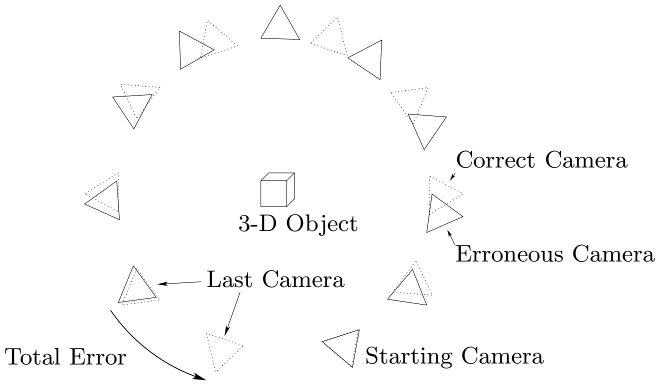


Fig. 2. Example reconstruction of a camera path around an object. Correct camera positions are denoted by dotted triangles, erroneous ones by solid triangles.

parameters at once to reduce the back-projection error globally. This straightforward approach, as used in [2] for scene reconstruction, has the disadvantage of a very large parameter space to be optimized. Therefore the less complex interleaved bundle adjustment [10] is used in the following.

Bundle adjustment is usually applied to an image sequence as a whole. For long sequences with more than 100 images it is very time consuming, especially if it is repeated every m images as explained later in Section 3.3. Therefore the method was adapted to support the optimization of only a few cameras at a time. The camera positions in such a subsequence are optimized jointly but without considering the rest of the sequence, while the 3-D points are optimized considering all cameras. Thus, back-projection error is only slightly increased for cameras outside the subsequence, while it is improved for those inside.

3 Feedback Loop

The main problem of the linear calibration process described in Section 2 is that small errors from one frame to the next accumulate over time and may thus lead to serious displacements of the camera positions. This is demonstrated in Figure 2, where a camera moves in a circle around an object taking 10 images in the process. The correct camera positions are equally spaced around the object, but an error of only about four degrees from each camera to the next adds up to more than 35 degrees. In order to get a correct reconstruction the circle must be closed again by removing this inconsistency. This situation is equivalent to a robot moving in a loop through some complex environment, and the approach introduced in the following is used similarly for mapping the robot's environment.

3.1 Closing Loops

Although in case of a hand-held camera it may be moved back to any earlier position, we assume here that N camera positions form a loop and that camera 0

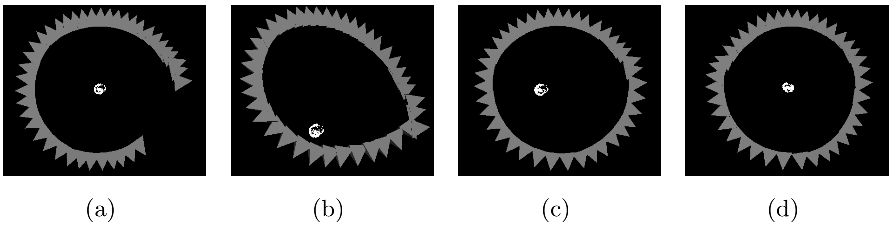


Fig. 3. (a) Linear, erroneous reconstruction. (b) Loop closed without considering rotation. (c) Loop closed considering rotation. (d) Final, optimized reconstruction.

follows again on camera $N-1$. In contrast to the linear calibration process used before features are now tracked from image $N-1$ to image 0, thus establishing a relationship between the two images. By applying the extension step of Section 2.2 the displacement between the last and the first camera position, $\varDelta\mathbf{t}_{N-1}$, is calculated with a much higher accuracy than before when the accumulated error was included. Going back to the image of a robot this is the equivalent of recognizing a formerly seen landmark.

Using this new information the task of closing the loop is again formulated as an optimization process. For now, only the translation vector of each camera, $\mathbf{t}_n$, is considered. The displacement vector between two cameras is denoted by $\varDelta\mathbf{t}_n = \varDelta\tilde{\mathbf{t}}_n = \mathbf{t}_{n+1} - \mathbf{t}_n$ for $0 \leq n < N-1$. Additionally, $\varDelta\tilde{\mathbf{t}}_{N-1}$ constitutes the current displacement vector between last and first camera while $\varDelta\mathbf{t}_{N-1}$ is the corresponding target displacement calculated above. Thus, for $0 \leq n < N$ the $\varDelta\mathbf{t}_n$ form the desired set of displacements while the $\varDelta\tilde{\mathbf{t}}_n$ are the displacements to be optimized. The residual vector is defined as

$$\epsilon = \left((\varDelta\mathbf{t}_0 - \varDelta\tilde{\mathbf{t}}_0)^T, (\varDelta\mathbf{t}_1 - \varDelta\tilde{\mathbf{t}}_1)^T, \ldots, (\varDelta\mathbf{t}_n - \varDelta\tilde{\mathbf{t}}_n)^T\right)^T \tag{1}$$

and using the Levenberg-Marquardt algorithm the camera positions $\mathbf{t}_n, n > 0$ are optimized by minimizing the residual $\epsilon^T\epsilon$. The first camera position $\mathbf{t}_0$ is kept unchanged.

The result of an erroneous, linear reconstruction of an example sequence is shown in Figure 3(a). Here, an object was placed on a turntable and rotated in 40 steps with one image taken for each. Applying the optimization above for closing this circle yields the reconstruction of Figure 3(b), which is obviously not satisfactory. The rotations between the displacement vectors $\varDelta\mathbf{t}_n$ do not sum up to a full circle, therefore the optimization does not yield a circle either.

The solution is to incorporate the missing rotation to a full circle into the computation of the residual vector. This rotation is calculated as the rotation difference between the last and the first camera pose, $\varDelta\mathbf{R} = \mathbf{R}_0\mathbf{R}_{N-1}^T$. Lacking any other knowledge we assume that the $\frac{n}{N}$th part of this rotation, $\varDelta\mathbf{R}_n$, is missing in each displacement vector. $\varDelta\mathbf{R}_n$ is computed using spherical linear interpolation [13] on a quaternion representation of $\varDelta\mathbf{R}$. Thus the new displacement vectors are computed as

$$\varDelta\hat{\mathbf{t}}_n = \varDelta\mathbf{R}_n\mathbf{R}_n(\mathbf{t}_{n+1} - \mathbf{t}_n). \tag{2}$$

Using these new target displacement vectors $\Delta\hat{\mathbf{t}}_n$ the result improves to that of Figure 3(c). The new camera positions were also rotated by $\Delta\mathbf{R}_n$ so that they now face in approximately the correct direction.

Usually an image sequence does not consist of exactly one revolution around an object. More circular camera movements may follow the first one, and in such cases it is not desired to change the camera positions in a loop already closed before. From there on, the position of a camera once adjusted is kept untouched, and the algorithm above is only applied to later cameras.

3.2 Optimizing Reconstruction

Changing the camera positions renders the 3-D point positions invalid, as seen in Figures 3(b) and 3(c), and they have to be recalculated. This is done by again minimizing the back-projection error during an optimization of the 3-D points.

Finally the result is again optimized globally using bundle adjustment as described in Section 2.3. The intrinsic parameters are assumed to be correct and bundle adjustment is only applied for the extrinsic parameters. The final result of such an optimization is shown in Figure 3(d). If only some cameras of a loop were adjusted in the closing step before, only those are optimized now, too.

3.3 Finding Loops

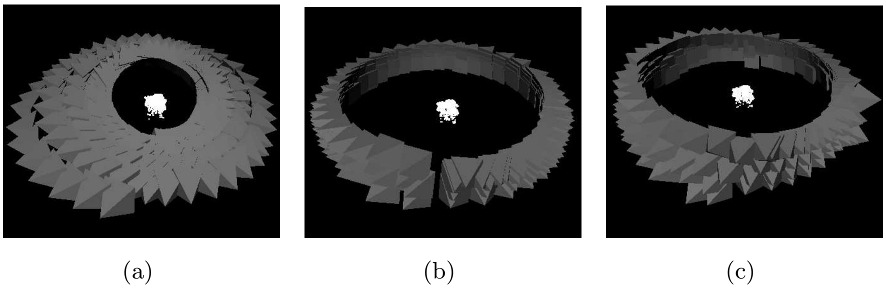
In a common application such as scene reconstruction from the images of a hand-held camera it is not known when a camera loop has been completed and the closing algorithm should be applied. The example of Figure 3 of an object on a turntable thus constitutes a special case since the end of the circle is known beforehand. For the general case a simple comparison scheme is used. A camera position is a neighbour of the current camera if its distance is smaller than k times the average distance between two consecutive camera positions and is not one of the m last positions. k and m are user-defined values. In order to assure that the corresponding images show approximately the same part of the scene a maximum viewing direction difference can be defined additionally.

An unsolved problem using this method is that large displacements, as in the example above, are not detected, while the closing algorithm makes the more sense the larger the accumulated error. This contradiction will be exemplified in the experiments in Section 4.

4 Experiments

Measuring the accuracy of a structure-from-motion reconstruction is a difficult problem especially for real scenes. The back-projection error is often used as a measure, but it depends highly on the quality of feature points, and a low back-projection error may still not give a satisfactory result.

Given ground-truth data for the camera positions a direct comparison to the reconstruction is possible and more meaningful. Therefore, two example sequences were chosen of an object being placed on a turntable and with a camera mounted on a robot arm above the table. Sequence 1 was already shown in Figure 3. It consists of 40 images of a coke can, taken during one revolution of the



(a) (b) (c)

Fig. 4. Reconstruction of the Santa Claus image sequence: (a) linear reconstruction, (b) only bundle adjustment on loops, (c) loops closed and bundle adjustment.

	back-projection error [pixel]			position difference		
	linear rec	only bundle	close+bundle	linear rec	only bundle	close+bundle
Sequence 1	1.28	1.75	2.06	11.7	11.5	5.18
Sequence 2	1.15	2.96	3.61	13.5	7.49	8.34

Table 1. Back-projection errors and camera position differences for the two example image sequences

turntable. Sequence 2 was taken from a Santa Claus figure with five revolutions of the turntable and 40 images each, where the robot arm was moved upward on a circle by 3 degrees after each revolution. The result of only a linear calibration is shown in Figure 4(a). For the improved calibration loops were detected automatically every tenth image after reconstruction of the first revolution, yielding the much improved results of Figures 4(b) and 4(c).

For comparison, the ideal camera positions were calculated from the turntable and robot arm positions. The reconstruction differs from the ideal one by a rotation, translation and scale factor. Using axis-angle notation for the rotation the 7 parameters of this transformation are estimated using (again) Levenberg-Marquardt to optimally register the two reconstructions with each other. The error value for the camera positions is calculated as the average distance of two corresponding cameras.

As mentioned before in Section 3.3 the closing of loops makes the more sense the larger the accumulated error. This issue is reflected in the experimental results of Table 1. Both the average back-projection errors and camera position differences are given for the reconstruction using only bundle adjustment on identified loops and for the whole process of closing loops of Section 3.1. The linear reconstruction of sequence 1 has a large accumulated error therefore closing loops has a great effect on the position difference while just applying bundle adjustment is insufficient to reduce this error. For sequence 2 on the other hand the accumulated error is rather low (the gap visible in Figure 4(b)) and thus, although this gap is closed for the reconstruction with closing in Figure 4(c), the camera position difference is still lower without the closing step. The inaccuracies introduced by closing, represented by the increased back-projection error in both sequences, were not compensated sufficiently by bundle adjustment.

5 Conclusion

In this contribution we proposed a method for creating a globally consistent scene reconstruction from an image sequence of a hand-held camera. Loops in the movement of the camera are detected and the accumulated error due to the linear calibration process is compensated by closing this loop. This approach is used similarly in robot navigation for simultaneous localization and mapping (SLAM). The results of each loop are optimized by bundle adjustment.

Since the closing introduces some error on each camera position it works well for the compensation of large errors, but for small displacements using only bundle adjustment may yield better results. Thus the main issues for future work are the identification of loops despite large errors and the reduction of errors introduced during the closing process.

References

1. A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *European Conference on Computer Vision (ECCV)*, volume 1, pages 311–326, 1998.
2. R. Hartley. Euclidean reconstruction from uncalibrated views. In *Lecture Notes in Computer Science*, pages 237–256. Springer-Verlag, 1994.
3. B. Heigl. *Plenoptic Scene Modeling from Uncalibrated Image Sequences.* ibidem-Verlag Stuttgart, January 2004.
4. R. Koch, M. Pollefeys, B. Heigl, L. van Gool, and H. Niemann. Calibration of hand-held camera sequences for plenoptic modeling. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 585–591, September 1999.
5. M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings SIGGRAPH '96*, pages 31–42, New Orleans, August 1996. ACM Press.
6. F. Lu and E. Milios. Globally consistent range scan alignment for environmental mapping. *Autonomous Robots*, 4:333–349, October 1997.
7. C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):206–218, March 1997.
8. J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, Seattle, Washington, 1994. IEEE Computer Society.
9. H.-Y. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *Sixth International Conference on Computer Vision (ICCV'98)*, pages 953–958, Bombay, January 1998.
10. R. Szeliski and P. Torr. Geometrically constrained structure from motion: Points on planes. In *European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (SMILE)*, pages 171–186, Freiburg, Germany, June 1998.
11. C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon University, April 1991.
12. E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision.* Addison–Wesley, Massachusets, 1998.
13. A. Watt and M. Watt. *Advanced Animation and Rendering Techniques.* Addison-Wesley, 1992.
14. T. Zinßer, C. Gräßl, and H. Niemann. Efficient feature tracking for long video sequences. In *DAGM '04: 26th Pattern Recognition Symposium*, August 2004. To appear.