www.icgst.com

Estimation of Camera Positions over Long Image Sequences

Ming Yan, Robert Laganière, Gerhard Roth
VIVA Research lab,
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5
laganier@site.uottawa.ca,
`http://www.site.uottawa.ca/research/viva/`

Abstract

In this paper, we present an iterative algorithm that computes the camera path of long image sequences. It consists in applying successive bundle adjustment phases on different segments of the image sequence. The local models thus obtained are merged together into a common reference frame. The procedure is then repeated on a new grouping of the cameras, until the reconstruction error has reached a given error tolerance. The main objective is to ensure the scalability of the reconstruction and the good convergence of the bundle adjustment process by imposing a limit on the number of views for which the structure and motion parameters have to be simultaneously optimized. Error accumulation is also prevented by exploiting the presence of loopbacks and intersections in the camera path. We show results obtained over different camera paths, in particular a spiral path and snake-like path.

Keywords: *3D Reconstruction, Bundle Adjustment, Feature matching, Structure from motion.*

1 Introduction

The estimation of camera position from image sequences is an important problem in computer vision. Several pose estimation methods have been proposed that most often deal with relatively short image sequences [8, 11, 16, 17]. This problem is generally solved using a *Bundle adjustment* procedure that provides a true maximum likelihood estimation of 3D pose and structure from observed data [10]. In short, bundle adjustment is an iterative method that proposes an optimized solution that minimizes the overall error between the measured 2D feature points and the projected 3D feature points. It does this by simultaneously adjusting the camera parameters and the 3D scene structure as a bundle. It is a general method that can be used to solve many reconstruction problems. Unfortunately, bundle adjustment has intrinsic drawbacks [19, 10]: i) it requires a good initialization; ii) it is a time-consuming process; and iii) it does not always converge. These problems become severe when dealing with long image sequences that contain hundreds of images [1].

This paper proposes to compute the positions of a moving camera in a long image sequences by first separating it into overlapping segments. Feature points are detected in each frame and then matched across the images of each segment. A 3D reconstruction of the segments is obtained using bundle adjustment which lead to an initial estimate of the entire image sequence after registering all the segments. Next a new partition is produced based on the relative positions among the reconstructed cameras. Matches are extracted from the images in a group and are sent to the bundle adjuster. Then the reconstructed groups will be registered and a better reconstruction can be found.

The main objective of the approach we proposed is to ensure the scalability of the reconstruction and the good convergence of the bundle adjustment process by imposing a limit on the number of views for which the structure and motion parameters have to be simultaneously optimized. Indeed, the method does not require a global bundle adjustment phase on the full set of images. Such reconstruction scheme is however subject to error accumulation [4]. Drifting is here prevented by exploiting the presence of loopbacks, intersections and common field of views in the camera path.

This paper is organized as follows. An overview of bundle adjustment is given in Section 2. Section 3 explains the proposed approach to camera path reconstruction from long image sequences. Section 4 shows our experimental results on a spiral path and a snake-like path. Section 5 is a conclusion.

2 Bundle Adjustment

Bundle adjustment is the process by which globally visually consistent solutions are found for the structure and motion of a scene viewed by multiple cameras. If an initial estimate of the structure and motion is available, bundle adjustment is able to find a solution with minimal errors for all 2-D, 3-D points and projection matrices. The bundle adjustment procedure has been described by many authors [10, 6, 20]. The problem is usually formulated as follows:

Given $\mathbf{x}_{ij}$, the i^{th} 2-D point of the j^{th} image, find the maximum likelihood camera projection matrix $\mathbf{P}'_j$ and the maximum likelihood 3-D point $\mathbf{X}'_i$ simultaneously such that the reprojected image point $\mathbf{x}'_{ij}$ is as close as possible to the given image point $\mathbf{x}_{ij}$. In general, the reprojected image point $\mathbf{x}'_{ij}$ is not identical to the measured image point $\mathbf{x}_{ij}$ because of the noise. Bundle adjustment tries to minimize the overall error between the given 2-D points and the reprojected points by adjusting all the camera projection matrices and the 3-D points:

$$min \sum_{i,j} d(\mathbf{x}'_{ij} \ , \ \mathbf{x}_{ij})^2 = min \sum_{i,j} d(\mathbf{P}'_j \cdot \mathbf{X}'_i \ , \ \mathbf{x}_{ij})^2 \ . \tag{1}$$

Bundle adjustment is an iterative process that continues until the final error tolerance is reached. In practice, bundle adjustment does not always return a correct answer. It is a non-linear minimization process and it relies heavily on the initial estimate of the camera position and the 3-D scene points [19]. A bundle adjustment method either diverges or return a wrong estimation if the initial values were not close enough to the real values.

Another issue is that bundle adjustment is computationally expensive due to the large number of input frames and features [19]. It requires to solve of a very large minimization problem [10].

Different approaches that deals with these issues have been proposed.

2.1 Hierarchical Bundle Adjustment

The hierarchical bundle adjustment is somewhat like a recursive algorithm where the original problem is recursively separated into small pieces until each piece can be easily solved. The solution will then be propagated back to solve the entire problem.

Royer et al. [18] presented such a hierarchical bundle adjustment. The original long image sequence is recursively subdivided into two parts with two overlapping frames until there are only three frames in each final segment. The initial estimate of the first triplet was obtained by computing an essential matrix. Making use of the overlappings, they deduce the first two frames of the second triplet from the previous triplet. The pose estimation algorithm was used to compute the third camera position. Local estimations are done by running the bundle adjustment over all the triplet frames. These triplet frames are then merged and a global bundle adjustment is performed to find the reconstruction.

Another hierarchical method was presented by Shum et al. [19] from which the bundle adjustment was exploited efficiently with virtual key frames. First, they divide the sequence into small segments. The first frame and the last frame within a segment are used to solve the structure from motion problem, and all the in-between frames are interpolated for the initialization. Bundle adjustment is applied on the segments and partial 3-D reconstructions are obtained. These partial reconstructions are then merged and a final global bundle adjustment is then required to optimize the results.

2.2 Incremental Bundle Adjustment

In their approach, Mouragnon et al. [15] run new bundle adjustment phases whenever a new key frame and 3-D points are detected and added to the system. Camera poses are obtained using the five-point relative pose algorithm, and 3-D points are obtained using the standard triangulation. Key frames are identified such that they are as far apart as possible and at the same time have enough common correspondences. New points are identified as those observed in the last three key frames. These points are then added to the system and a fresh bundle adjustment is performed to minimize the overall errors.

Another incremental bundle adjustment approach was proposed by Zhang and Shan [21] that applies a sliding window on image triplets. The first two camera motions are obtained using two-view structure from motion techniques. The third camera motion is determined by applying the three-view partial bundle adjustment to the triplet. They used feature points from both two views and three views within a triplet. The window slides to the next image triplet when a new frame is added, thus reconstructing the scene incrementally. This approach is especially suitable for sparse image sequences where the difference between consecutive images is quite large.

2.3 Degenerate Bundle Adjustment

Several degenerated bundle adjustment approaches can also be found in the literatures. In [3], a rotation-free bundle adjustment is proposed. The camera rotation parameters were eliminated through algebraic manipulation based on invariant theory. A new structure from motion equation is created that introduces a rotation matrix free cost function for the bundle adjustment. A rotation-free bundle adjustment is more robust to errors arising from the initial estimation and this is especially useful for translational motions [9].

Malis and Bartoli [14] proposed an intrinsic free

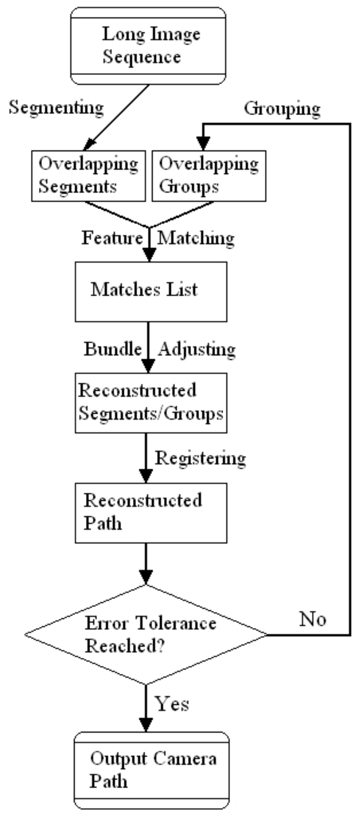

Figure 1: A block diagram of the proposed approach to 3D path reconstruction.

bundle adjustment. They do not consider the unknown camera internal parameters in the optimization process. Instead, they proposed a bundle adjustment approach that utilizes the camera external parameters and the 3-D structure to parameterize the reconstruction problem with no internal camera parameters being considered.

3 Proposed Approach to Path Reconstruction

We now introduce our approach for the 3-D camera pose estimation from long image sequences. The goal was to devise a scalable method that iteratively processes a large number of images while at the same time taking advantage of the loop backs among the images. This path reconstruction approach is composed of two major steps: i) camera segmentation and ii) camera registration. The block diagram of Figure 1 illustrates the proposed procedure.

In the segmentation step, the images of the sequence are divided into short overlapping groups. Grouping is accomplished such that the images in a group correspond to pictures of the scene taken from nearby locations. Consequently, the disparity between adjacent images of a group is relatively small, which means that correspondences can be easily established. However, at the same time, a sufficiently large baseline must exist within the group in order to ensure that the reconstruction process remains sufficiently accurate.

It is also necessary to have a significant amount of overlap between the groups. The connected groups must therefore share a certain number of common images. This redundancy will make possible to connect the different groups together during the registration step. Since the accuracy of the resulting representation partly depends on the level of overlap, the groups are built to ensure that at least $t\%$ of the images in a group are shared with at least one other group (we used 50%).

The complete path of the sequence is therefore reconstructed by iteratively processing each group, merge the camera together through registration and then re-group the camera set based on the new estimated positional information.

3.1 Segmentation

In the segmentation process, the goal is to group together spatially neighboring cameras; however for the first iteration, the pose of the cameras is unknown. Consequently, the groups are initially built based on the ordering of the image sequence. The assumption is that the images have been taken in sequence while moving the camera across the scene. As it will be shown, this is sufficient to obtain an acceptable initial estimate of the scene and to detect the potential loops in the camera path.

To obtain the initial match set that will be used by the bundle adjuster to reconstruct the scene, the image sequence is processed following its natural order. The resulting match set is sent to the bundle adjuster [5] to find camera positions as well as the 3-D reconstruction. The reconstruction of all the segments of the sequence is obtained the same way. Registration (see Sect. 3.2) is then required to merge these segments to obtain an initial 3-D model.

The segmentation process will then have to be repeated on the reconstructed cameras in order to form new groups. This new grouping aims at taking into consideration the possible loops in the camera sequence that connects together non-consecutive image sub-sequences because of their spatial proximity. This grouping is realized by using the available 3-D camera pose estimates obtained from the previous iteration.

3.1.1 Camera grouping

The objective is to create a new partition of the cameras, from their estimated spatial locations, such that the full group set will be connected and that a good level of overlap between the groups will be obtained.

We have N cameras $C_1, \cdots, C_N$, and we want to create a partition made of groups $\{G_i\}$, each containing L cameras. Each camera must belong to at least one group and, to ensure good overlap, at least $t\%$ of the cameras in a group must belong to at least one other group. To create these groups, we proceed by iteratively adding cameras to the groups. To ensure that these groups form compact clusters in which all cameras are as close as possible to all other cameras in the group, we proceed as follows. The distances between an unassigned camera and all cameras in a group are computed, and the maximum distance is retained. This procedure is repeated for all the unassigned cameras and the one that has the minimum maximum-distance is selected. This camera is indeed the one that is the closest to its farthest group member thus is the one that should produce the most compact cluster. The complete grouping algorithm is given below.

Grouping algorithm

1. Create two disjoint sets $\mathcal{A}$ and $\mathcal{U}$, where $\mathcal{A}$ is the assigned camera set and $\mathcal{U}$ is the ungrouped camera set. Initially, set $\mathcal{A}$ to empty while $\mathcal{U}$ contains all cameras to be processed.

2. Start with $n = 1$, randomly selected an image from $\mathcal{U}$ as the starting point; the corresponding camera C is assigned to G_n, added to $\mathcal{A}$ and removed from $\mathcal{U}$.

3. For each C_i in $\mathcal{U}$ find $d_{max}(C_i, G_n)$ by computing the distance between C_i and all cameras in G_n, where $d_{max}(C_i, G_n) = \max_{C_j \in G_n} d(C_i, C_j)$.

4. Get C_{min} that is the camera with the smallest $d_{max}(C_i, G_n)$. C_{min} is assigned to G_n, added to $\mathcal{A}$ and removed from $\mathcal{U}$.

5. Repeat step 3 and step 4 until the group size is reached or $\mathcal{U} = \emptyset$; then $n = n + 1$.

6. For each C_i in $\mathcal{U}$ find $d_T(C_i, \mathcal{A})$ with $T = tK$ (e.g. with $t = 50\%$, $T = K/2$). $d_T(C_i, \mathcal{A})$ is defined as the distance between camera C_i and its T^{th} nearest neighbor in $\mathcal{A}$.

7. Get C_{min} that is the camera with the smallest $d_T(C_i, \mathcal{A})$. C_{min} is assigned to G_n, added to $\mathcal{A}$ and removed from $\mathcal{U}$.

8. Get the T closest camera to C_{min} in $\mathcal{A}$. All these cameras are assigned to G_n. (They constitute the overlapping cameras in the group G_n). Go to step 3.

3.1.2 Multi-view correspondence

Once the groups formed, valid correspondences within each group must be found. Since a group is generally made of distinct image sub-sequences, some correspondences have already been established from the previous step. The sub-sequences are then connected together using a multi-view correspondence strategy [16].

Fundamental matrices are computed from the matches between all possible pairs. For a K-image group, a total of $K(K-1)$ fundamental matrices will be found. Those matches that can generate valid fundamental matrices are kept and they are the support pair sets. Trilinear tensors are to be computed from the support pair sets and we expect a total of $K(K-1)(K-2)$ tensors. Again, those matches that can generate valid trilinear tensors are kept and they are the support triplet sets. Finally we chain the triple sets and a list of the correspondences from all the images in the group are obtained.

To obtain the initial match set that will be used by the bundle adjuster to reconstruct the scene, the image sequence is processed following its natural order. First the Scale Invariant Feature Transform (SIFT) features [13] are extracted. Then a RANdom SAmple Consensus RANSAC [7] strategy based on both fundamental matrix and tensor estimations is used to find reliable correspondences between images [10]. The resulting triplets of matches are then chained together across the sequence segment to get multi-view correspondences. These steps can be performed with the help of the Projective Vision Toolkit(PVT) [17]. In order to stabilize the result, a fixed number of features are returned using an automatically selected threshold. Dealing with false matches is also an important issue in any correspondence process. The original matches must pass a symmetry test and a consistency test in order to reduce some false matches. The symmetry test requires that feature A in the first image also has the largest correlation with feature B in the second image. The consistency test examines the disparity gradient [12] of the correspondences and keeps only those correspondences with a disparity gradient less than a certain amount.

3.1.3 Reliable bundle adjustment

The automatic correspondence method produces more matches than the bundle adjustment requires. The match set is therefore subsampled by a factor of 2 and then sent to the bundle adjuster for testing. If no convergence to a solution with a sufficiently large support is found, then the match set is further subsampled by another factor of 2. This process is repeated until one of the subsampled set converges with good support. As the original match set is expected to contain very few false matches, two pass of subsampling are generally sufficient to obtain good convergence. It also has been observed that the bundle adjuster has a better stability when it starts with a smaller number of matches, with more matches being added as the solution improves in accuracy.

Feature points that are matched over a large number of images are also preferable for the estimation of the global 3-D structure of a group. We therefore scan the matches list and keep only those matches that exist in more than a certain number of images.

3.2 Registration

Registration is the process by which two adjacent 3-D reconstructions of points are merged into a single reference frame. This is possible because adjacent groups always exhibit a high degree of overlap. The registration process consists in finding the similarity transform that will bring two corresponding 3-D points and 3-D camera positions to the same location. Although registration on the overlapping 3-D points is possible, we found that it was more reliable to register the groups based on camera positions only.

The relative position of a camera in a group G_n can be extracted from the normalized camera matrix $Q_i^n = [R_i^n|T_i^n]$ obtained as a result of the bundle adjustment step. The i^{th} camera center as computed in the reference frame of G_n is given by:

$$C_i^n = (R_i^n)^T(-T_i^n) \qquad (2)$$

where $(R_i^n)^T$ is the transpose of the matrix R_i^n.

Since the segments are independent and they were processed separately, the scaling factors in the reconstructed segments are different. A consistent scaling factor is required for all the segments before they are registered. This is done by applying a scaling ratio on each segment. The scaling ratio is computed as the average ratio of the adjacent camera distances in the first segment to the adjacent camera distances in the second segment, as be shown below.

$$SR_j = \frac{\sum_{i=0}^{N-1} \frac{d(C_1^i, C_1^{i+1})}{d(C_{j+1}^i, C_{j+1}^{i+1})}}{N-1} \ . \qquad (3)$$

where:
$d(a,b)$ computes the distance between a and b;
SR_j is the scaling ratio of the segment.

A maximum likelihood rotation and translation is computed [2] in order to minimize

$$\sum{}^2 = \sum_{i=1}^{T} \|C_i^m - (R_{mn} \cdot (S_{mn} \cdot C_i^n) + T_{mn})\|^2 \quad (4)$$

where R_{mn} is a 3 by 3 rotation matrix representing the orientation difference between two 3-D sets, T_{mn} is a 3-vector representing the translation between two 3-D sets.

The complete registration is then obtained by iteratively connecting each group to the registered set of cameras in this fashion; every new segment being registered to the already registered segments. A complete estimate of the camera positions is thus obtained. Initially, this estimate will be approximate, but sufficient to form new groups, taking into account the potential loopbacks and intersections in the sequence as detected by the grouping procedure. The positional estimates are then refined through a few iterations of the grouping, bundle adjustment and registration procedures.

4 Experiments

Figure 2: The first a few images from the spiral path.

We have tested the proposed algorithm on two specific camera paths: a spiral path and a snake-like path. These two paths contain a number of loopbacks and intersections which are exploited by our reconstruction process. Over two hundred images were taken to generate the spiral path made of about two complete turns. The first a few images of the spiral path are displayed in Figure 2.

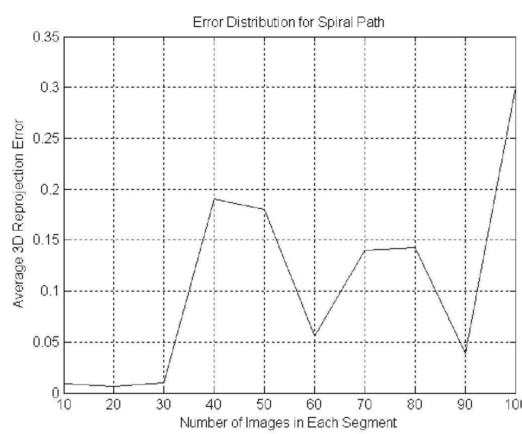

Figure 3: Reprojection error of different length of segments.

A trade off exists between the size of the groups and the precision of the result. Larger groups are preferred because fewer registrations are needed for the same long image sequence but large groups contain more unknowns which makes the bundle adjustment process less reliable. We tested the selected bundle adjuster [5] on different length of segments to find an appropriate segment length. Figure 3 shows that segments with less than 30 images are stable and that 20

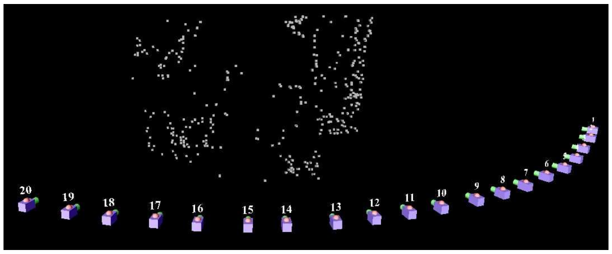

Figure 4: The reconstruction of the first segment.

images in a segment seems to be a good choice. Different bundle adjustment libraries would most probably give different results but similar trends would be observed and be used to guide the choice of an optimal segment length. The reconstruction of one segment is shown in Figure 4; error accumulation will become apparent when the different reconstructed segments are incrementally registered one with respect to the other.

In order to register the reconstructed segments, these ones need to overlap. Although three images are enough to compute the rotation and translation, involving more images stabilizes the registration process. The number of overlapping images has then been set to be half of the total images in a group to ensure both stability and efficiency. The complete initial 3-D reconstruction is shown in Figure 5. The box-like objects in the graphs are the cameras and the small dots are 3-D feature points. A complete circle contains 95 images. Starting from camera 1, the first loopback is camera 96, and the second loopback is camera 191. We enlarged these three particular cameras in Figure 5 for better viewing. The corresponding three images are in Figure 6.

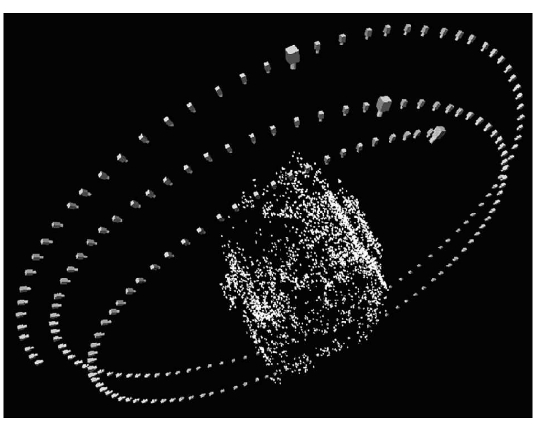

Figure 5: Initial reconstruction. Camera 1, camera 96 and camera 191 have been enlarged; these ones should be aligned according to views shown in Figure 6.

The three major problems arising from the initial 3-D reconstruction are:

1. Drifting errors. Camera 96 (The enlarged one on the middle circle) is supposed to be aligned to camera 1 (The enlarged one on the inner circle). Camera 191 (The enlarged one on the outer circle) drifted even farther.

2. Off path errors. The distance between the inner and outer circles is not constant while it was constant when the images were taken.

3. Off plane errors. The reconstructed camera path is not in the same plane while the actual motion path was planar.

Figure 6: Loop back images: from left to right are image 1, image 96 and image 191.

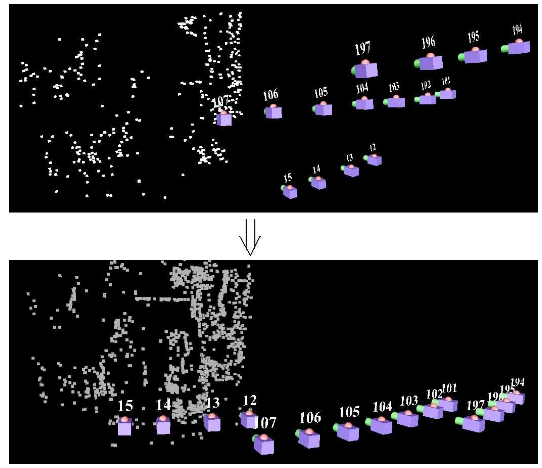

Figure 7: The lower figure is the reconstructed group, the upper figure is the corresponding group camera positions in the initial estimation.

However, this initial estimate is sufficient to have neighboring cameras included in the same group at the next iteration. More iterations are performed until the reprojection error falls below the error tolerance. Figure 8 shows the final 3D reconstruction of the spiral camera path, from which we can see that the drifting errors, the off path errors and the off plane errors have all been greatly reduced.

Our algorithm has also been applied to a snake-like path. After applying the bundle adjustment and registration on the segments, we obtain the initial reconstruction of the snake-like path that is shown in Figure 9. Again, the initial reconstruction is used to form groups. Bundle adjustment and registration are applied on these groups and a refined reconstruction can be found. We display the reconstructed camera positions along with the 3-D feature points in Figure 10.

Finally, we compare the reconstruction errors of

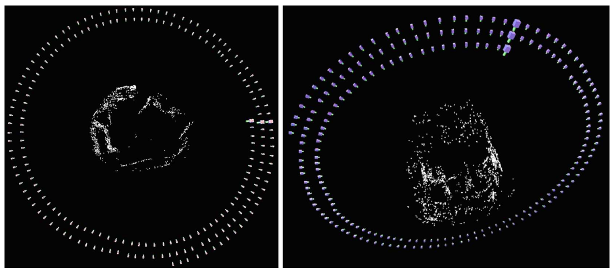

Figure 8: Top view and side view of the final reconstructed camera paths.

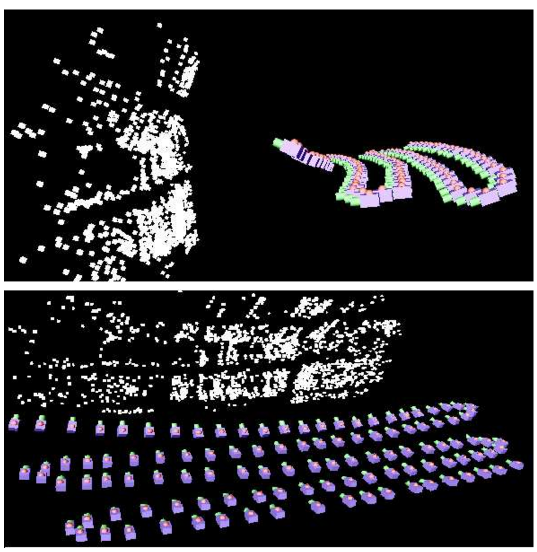

Figure 9: Side view and top view of the snake like path.

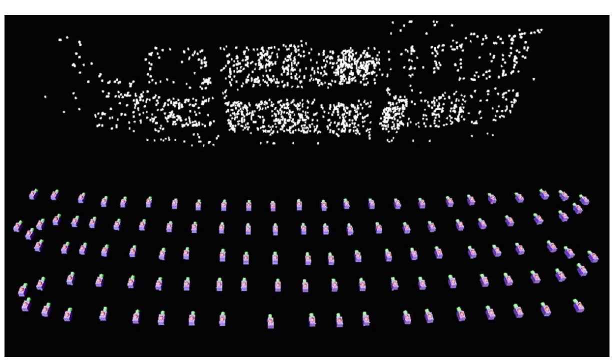

Figure 10: The reconstruction of the snake like path.

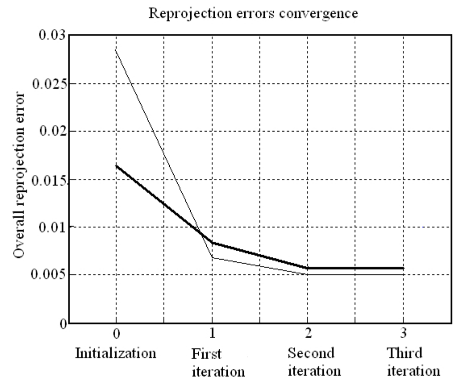

Figure 11: Evolution of the reprojection error over few iterations of the reconstruction process. The thin line represents the spiral path error trend, the thick line represents the snake path error trend.

the spiral path and the snake-like path. Typically there was 30 to 50 correspondences in an image. We randomly select 30 correspondences from each image to compute the reprojection errors. The obtained camera matrices and 3-D points are reprojected through:

$$\mathbf{x}'_{ij} = \mathbf{K}\mathbf{Q}'_j \cdot \mathbf{X}'_i \tag{5}$$

where $\mathbf{K}'$ is the camera internal calibration matrix, $\mathbf{Q}'_j$ is the j^{th} obtained camera matrix, $\mathbf{X}'_i$ is the i^{th} obtained 3-D point, and $\mathbf{x}'_{ij}$ is the projected 2-D feature point. We then compute the square distances between the measured correspondences and the reprojected correspondences. Results are displayed in Figure 11. The thin line shows how the overall reprojection error of the spiral path is reduced, and the thick line shows how the overall reprojection error of the snake path is reduced. The large initial errors are all greatly reduced after the first iteration when most loopbacks and intersections have been identified. The following few iterations slightly improve the results.

5 Conclusion

We have presented an iterative algorithm to compute long camera paths by breaking them into small segment of fixed length. By doing this, we limit the complexity of the bundle adjustment phase which ensure both scalability of the reconstruction process and makes bundle adjustment more likely to converge to the global minimum. Assuming intersections and loopbacks are present in the path, we showed that these ones can be automatically identified and advantageously exploited in order to increase the accuracy of the 3D reconstruction. This approach is particularly adapted to long image sequences taken with hand held cameras or from vehicle-mounted cameras to capture large urban environment.

References

[1] S. Agarwal, N. Snavely, I. Simon, S.M. Seitz, and R. Szeliski. Building rome in a day. In *International Conference on Computer Vision*, Kyoto, Japan, September 2009.

[2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, September 1987.

[3] P.L. Bazin and M. Boutin. Structure from motion: A new look from the point of view of invariant theory. *SIAM Journal on Applied Mathematics*, 64(4):1156–1174, 2004.

[4] K. Cornelis, F. Verbiest, and L. Van Gool. Drift detection and removal for sequential structure from motion algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1249–1259, October 2004.

[5] EOS. Photomodeler pro by eos systems inc. http://www.photomodeler.com.

[6] O. Faugeras and Q. Luong. *The Geometry of Multiple Images*. The MIT Press, Cambridge, Massachusetts, 2001.

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–395, 1981.

[8] A. W. Fitzgibbon, G. Cross, and A. Zisserman. Automatic 3d model construction for turn-table sequences. In *Proceedings of European Conference on Computer Vision*, Freiburg, Germany, June 1998.

[9] P. Hammarstedt and A. Heyden. Euclidean reconstruction from translational motion using multiple cameras. In *Proceedings of Fifth International Conference on 3-D Digital Imaging and Modeling*, pages 352–359, June 2005.

[10] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision, 2nd Edition*. Cambridge University Press, 2003.

[11] G. Jiang, Y. Wei, L. Quan, H. Tsui, and H. Y. Shum. Outward-looking circular motion analysis of large image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):271–277, February 2005.

[12] R. Klette, K. Schluns, and A. Koschan. *Computer vision : three-dimensional data from images*. Springer, New York, 1998.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[14] E. Malis and A. Bartoli. Euclidean reconstruction independent on camera intrinsic parameters. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 2313–2318, September 2004.

[15] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P.sayd. Real time locolization and 3d reconstruction. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, June 2006.

[16] G. Roth. Automatic correspondences for photogrammetric model building. In *International Society for Photogrammetry and Remote Sensing*, pages 713–720, Istanbul, Turkey, July 2004.

[17] G. Roth and Anthony Whitehead. Using projective vision to find camera positions in an image sequence. In *Proceedings of International Conference on Vision Interface*, pages 87–94, Montreal, Quebec, May 2000.

[18] E. Royer, M. Lhuillier, M. Dhome, and T. Chateau. Towards an alternative gps sensor in dense urban environment from visual memory. In *Proceedings of the fifteenth British Machine Vision Conference*, London, United Kingdom, 2004.

[19] H. Y. Shum, Q. Ke, and Z. Zhang. Efficient bundle adjustment with virtual key frames: A hierarchical approach to multi-frame structure from motion. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[20] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment–a modern synthesis. In *Prococeedings of Workshop Vision Algorithms: Theory and Practice*, pages 298–372, 1999.

[21] Z. Zhang and Y. Shan. Incremental motion estimation through modified bundle adjustment. In *Prococeedings of International Conference on Image Processing*, pages 343–346, Barcelona, Catalonia, Spain, September 2003.