I^{2}MTC 2008 – IEEE International Instrumentation and
Measurement Technology Conference
Victoria, BC, Canada, May 12–15, 2008

Iterative Computation of Camera Paths

Ming Yan, Robert Laganière, Gerhard Roth

School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, CANADA, K1N 6N5
E–mail: laganier@site.uottawa.ca.

Abstract – *This paper presents a novel algorithm to iteratively compute camera paths of long image sequences. Scale Invariant Features are first extracted from the ordered set of images. These images are then matched pair-wise sequentially and correspondences are computed. An initial geometric path can be found after by applying a bundle adjustment algorithm on these correspondences. Distances between cameras can be computed from this initial estimation. The iteration process starts by grouping nearby cameras and then bundle adjusting the groups, and ends by merging the groups. This process is repeated until the reprojection errors fall into the preset tolerance. The key point in this algorithm is to take the advantages of loopbacks in the image sequences. We have obtained excellent results for two particular camera paths, namely the spiral path and the snake like path. Our algorithm achieves both precise and stable results.*

I. INTRODUCTION

Camera pose estimation has been explored for the past few decades and it still remains an active topic. Its applications involve but are not limited to image based rendering [1], robotics [2], photogrammetry [3] and virtual navigation [4]. Several successful pose estimation methods have been proposed for specific short image sequences [5], [3], [6]. Among them we can see that the bundle adjustment brings better results since it provides a true maximum likelihood estimation even when a few input data were missing [7]. Unfortunately, bundle adjustment has intrinsic drawbacks that prevent it from being used directly. These drawbacks include [7], [8]: i)Good initialization requirement; ii)Extremely time consuming process; and iii)Convergence problem. These become severe when dealing with long image sequences that contain hundreds of images.

The two practical methods that use bundle adjustment *indirectly* [7] are the hierarchical merging of sub-sequences and the incremental approach. Royer et al. [2] presented a hierarchical method. The original long image sequence is recursively subdivided into two parts with two overlapping frames until there are only three frames left in each final segment. Local estimations are done by running the bundle adjustment over all the triplet frames. These triplet frames are merged and then a global bundle adjustment is exploited to find the reconstruction. However, it is not efficient to run the global bundle adjustment on long image sequences because of the very large number of frames involved. Shum et al. [8] presented another hierarchical method that uses the bundle adjustment efficiently with virtual key frames. Instead of recursively subdividing the original image sequence, they only divided the sequence into small segments once and no further subdivision is performed. These small segments are merged to find a complete 3-D reconstruction after bundle adjustment is applied to each segment. Also, two virtual key frames are extracted from each segment. The final 3-D reconstruction is found by running bundle adjustment on the virtual key frames. This method significantly speeds up the bundle adjustment process for long image sequences. However, both these hierarchical methods suffer from the accumulated error built by the merging process. It results an initial reconstruction that has drifted away from the real location and the final bundle adjustment may not converge due to the poor initial reconstruction.

Mouragnon et al. [9] presented an accurate incremental method for reconstruction and localization. The bundle adjustment is run whenever a new key frame and 3-D points are detected and added to the system. Although this method works better than global bundle adjustment, it requires that the involved frames be relatively far from each other and that the camera be precisely calibrated. This may not be appropriate for a video sequence in which the frames are very close to each other.

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 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. Error accumulation is here prevented by exploiting the presence of loopbacks and common field of views in the camera path.

II. BUNDLE ADJUSTMENT AND CAMERA MATRIX

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. The bundle adjustment procedure has been described by many authors [10], [7], [11]. The problem is usually formulated as follows:

Given the i^{th} 2-D point of the j^{th} image, find the maximum likelihood camera projection matrix P_j' and the maximum likelihood 3-D point M_i' simultaneously such that the reprojected image point m_{ij}' is as close as possible to the given image point m_{ij}. Bundle adjustment tries to minimize the overall error of the complete given 2-D points and the reprojected points by adjusting all the camera projection matrices and the 3-D points. Several equivalent minimization equations of the bundle adjustment are shown below:

$$min \sum_{i,j} d(m_{ij}', m_{ij})^2 \tag{1}$$

$$min \sum_{i,j} d(P_j' \cdot M_i', m_{ij})^2 \tag{2}$$

$$min \sum_{i,j} d(KQ_j' \cdot M_i', m_{ij})^2 \tag{3}$$

Equation 3 shows that the output camera pose can be in the form of a normalized camera matrix Q assuming the camera calibration K is known. Different from the projection matrix P, Q does not contain any camera internal parameters. A normalized camera matrix is the combination of rotation and translation expressed as $Q = [R|T]$, where R is a rotation matrix and T is a translation vector. Camera pose is defined by the camera center(translation) and the camera orientation(rotation). A normalized camera matrix is enough to compute camera pose since it contains both the translation and the rotation. We will be focusing on the normalized camera matrix Q when applying the bundle adjuster on the image sequences and a roughly estimated calibration matrix is enough to obtain good results.

III. PATH RECONSTRUCTION

Our full path reconstruction approach is composed of two major steps: i) camera segmentation and ii) camera registration.

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 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 each group. The connected groups must therefore share a certain number of common images. This redundancy will make it possible to connect the different groups together during the registration step where the individual reconstruction results are merged in a common reference frame. Since the accuracy of the resulting representation depends on the level of overlap, the groups are built to ensure that at least half of the images in a group are shared with at least one other group.

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.

A. 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 feature points used are the Scale Invariant Feature Transform (SIFT) corners [12] and a Random Sample Consensus(RANSAC) [13] strategy based on both fundamental matrix and tensor estimations is used in order to extract reliable correspondences between images [7]. The resulting triplets of matches are then chained together across the sequence segment to get multi-view correspondences. These steps can be accomplished with the help of the Projective Vision Toolkit (PVT) [6].

The resulting match set is sent to the bundle adjuster [14] to find camera positions as well as the 3-D reconstruction. The reconstruction of all the segments of the sequence is obtained in the same way. Registration (see Sect. B) 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.

A.1 Camera grouping

The objective here 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 level of overlap between the groups will be obtained.

We have N cameras $C_1, \cdots, C_N$, we want to create a partition made of groups $\{G_i\}$, each containing K 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.

A.1.a 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 $d_{max}(C_i, G_n) = \max\limits_{C_j \in G_n} d(C_i, C_j)$.
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.

A.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 subsequences are then connected together using a multi-view correspondence strategy [3].

A.3 Reliable bundle adjustment

Bundle adjustment is a complex multi-variable optimization process that is not always guaranteed to converge. It is highly affected by the presence of outliers and since the automatic correspondence process tend to produce large number of matches, the presence of such outliers in the set is difficult to avoid. A good initial estimate of the 3-D camera positions is an important factor in obtaining reliable solutions.

B. Registration

Registration is the process by which two adjacent 3-D reconstructions of points and camera positions are merged into a single reference frame. This is possible because the adjacent groups 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:

$$C_i^n = (R_i^n)^T(-T_i^n) \tag{4}$$

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

Since each bundle adjustment procedure were applied independently on each group, the scales in the reconstructed camera sets are different. A consistent scaling factor must therefore be identified. This is done using the cameras that belong to more than one group. Let's consider C_i and C_j that both belong to G_n and G_m. The ratio of the distance between these two cameras, as computed in each reference is then equal to the scale factor that exists between the two groups, that is:

$$S_{mn} = \frac{d(C_i^m, C_j^m)}{d(C_i^n, C_j^n)} \tag{5}$$

In practice, we use the mean of the scale factors computed from all pairs that are common to groups G_m and G_n.

A maximum likelihood rotation and translation is to be computed [15] in order to minimize

$$\sum{}^2 = \sum_{i=1}^{T} \| C_i^m - (R_{mn} \cdot (S_{mn} \cdot C_i^n) + T_{mn}) \|^2 \tag{6}$$

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 registration is done by applying R_{mn} and T_{mn} to the cameras of group G_n. The complete registration is then obtained by iteratively connecting each group to the registered set of cameras in this fashion. 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 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 procedure.

IV. EXPERIMENTS

We have tested the proposed algorithm on two specific camera paths: a spiral path and a snake like path. These two particular paths contain a number of loopbacks, which are critical for the iteration process to converge.

Images were taken by moving the camera in the scene. The first a few images of the spiral path are displayed in figure 1. We will focus more on the spiral path hereafter since the snake like path can be processed similarly.

Fig. 1. The first a few images from the spiral path.

Over two hundred images were taken to generate the spiral path made of about two complete turns. There is always a trade off 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. The difficulty is that the longer the sequence, the less likely it is that the bundle adjustment will converge. We tried the bundle adjuster on different length of segments to find a appropriate segment length. Figure 2 shows that segments with less than 30 images are stable and that 20 images in a segment seems to be a good choice. We also need to determine the number of overlapping images in the segments for registration. 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. Following the steps described in section A we can find the reconstructions of all the segments. These reconstructed segments are then registered incrementally until we reach the last segment.

The complete initial 3-D reconstruction is shown in figure 3. 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 3 for better viewing and comparing purposes. The corresponding three images are in figure 4.

Three major problems arise from the initial 3-D reconstruction:

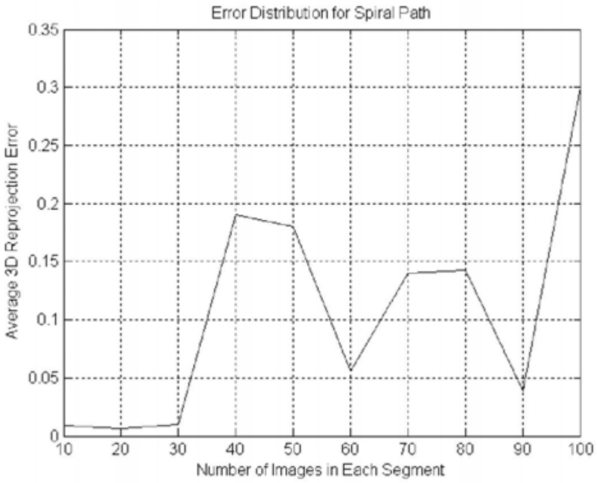

Fig. 2. Reprojection error of different length of segments.

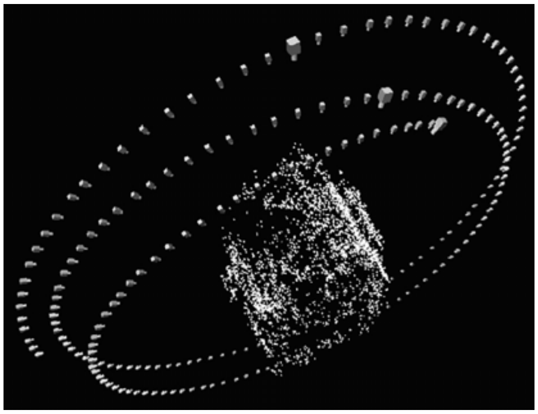

Fig. 3. Initial reconstruction. Camera 1, camera 96 and camera 191 have been enlarged; these ones should be aligned according to views shown in figure 4.

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). In figure 3, camera 96 is drifting away and camera 191 (The enlarged one on the outer circle) is drifting 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 path is in the same plane.

However, this initial estimate is sufficient to have these cameras included in the same group at the next iteration. More iterations are performed until the reprojection error falls below the error tolerance. Figure 5 shows the final 3- D 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 can be applied to the snake like path and

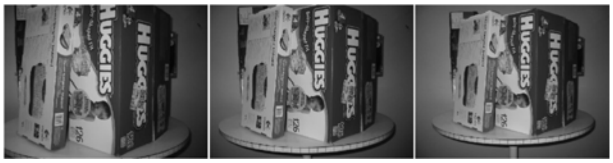

Fig. 4. Loop back images: from left to right are image 1, image 96 and image 191.

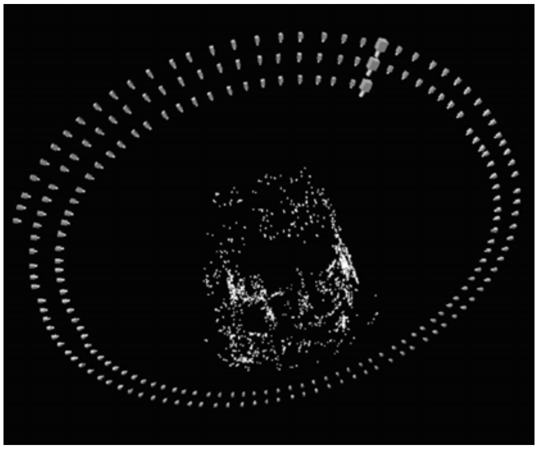

Fig. 5. Top view and side view of the final reconstructed camera paths.

other long image sequences similarly as long as loopbacks exist in the path. We process the snake like path using the same algorithm and display the reconstructed cameras along with the 3-D feature points in figure 6. When taking the images of the snake like path, we deliberately move the camera with unequal paces to demonstrate that our algorithm is also capable of unevenly separated paths.

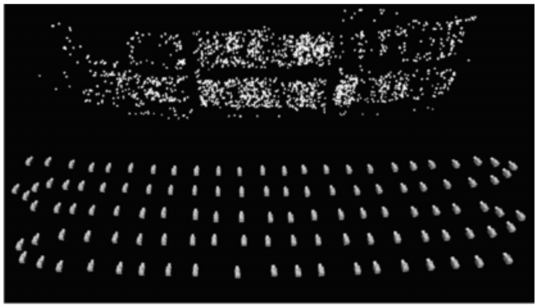

Fig. 6. Snake like path reconstruction.

V. CONCLUSION

We have presented an iteration algorithm to compute long camera paths. This system deals with long image sequences with loopbacks. We limit the bundle adjustment to only local reconstructions and this ensure precise reconstruction of ordered segments and unordered groups. Errors introduced by registration are reduced by the iteration process and a precise complete reconstruction can be expected. Furthermore, our system does not require that the distance and angle between images to be constant. In fact, it tolerates large difference of distances and angles between images. This is critical for videos taken with hand held cameras or vehicle mounted cameras instead of cameras being controlled by smoothly moving motors.

Our objective was to propose a reconstruction method that can scale to a very large number of images while ensuring the good convergence of the bundle adjustment process. This was achieved by imposing a limit on the number of views for which the structure and motion parameters have to be simultaneously optimized and by exploiting the presence of loopbacks and common field of views in the camera path. Feature matching and bundle adjustment are the two main time consuming steps. SIFT features are used here to ensure a match set of good quality but these are more complex to extract; the matching step can however be executed in matter of seconds. Bundle adjustment is an optimization process that can take few minutes to converge depending on the number of views and features. However, since a fixed number of views are simultaneously optimized, the computing time grows linearly with the number of frames in the image sequence.

[1] H. Y. Shum, S. B. Kang, and S. C. Chan, "Survey of image-based representations and compression techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1020–1037, Nov. 2003.

[2] 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.

[3] G. Roth, "Automatic correspondences for photogrammetric model building," in *International Society for Photogrammetry and Remote Sensing*, Istanbul, Turkey, July 2004, pp. 713–720.

[4] J. Dehmeshki, X.Y. Lin, M. Siddique, X. Ye, F. Dehmeshki, and M. Roddie, "An innovative path planning and camera direction calculation method for virtual navigation," in *Proceedings of Biomedical Engineering*, Innsbruck, Austria, Feb. 2004.

[5] 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*, vol. 27, no. 2, pp. 271–277, Feb. 2005.

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

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

[8] 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.

[9] 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.

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

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

[12] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] 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*, vol. 24, pp. 381–395, 1981.

[14] EOS, “Photomodeler pro by eos systems inc.,” http://www.photomodeler.com.

[15] 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*, vol. 9, no. 5, pp. 698–700, Sept. 1987.