Incremental Motion Estimation Through Local Bundle Adjustment

Zhengyou Zhang and Ying Shan

May 2001

Technical Report
MSR-TR-01-54

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

`zhang@microsoft.com`
`http://research.microsoft.com/˜zhang`

Incremental Motion Estimation through Local Bundle Adjustment

Zhengyou Zhang and Ying Shan
Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
zhang@microsoft.com

Contents

Incremental Motion Estimation through Local Bundle Adjustment

Abstract

We propose a new incremental motion estimation algorithm to deal with long image sequences. It applies to a sliding window of triplets of images, but unlike previous approaches, which rely on point matches across three or more views, we also use those points shared only by two views. This is important because matches between two views are more common than those across more views. The problem is formulated as a series of local bundle adjustments in such a way that the estimated camera motions in the whole sequence are consistent with each other. Two implementations are described. The first is an exact one, which, based on the observation of the sparse structure of the adjustment network, embeds the optimization of 3D structure parameters within the optimization of the camera pose parameters. This optimization embedding considerably reduces the minimization complexity. The second is a mathematical procedure which transforms the original problem involving both 3D structure and pose parameters into a much smaller one, a minimization over just the camera's pose parameters. This leads to an even higher computational gain. Because we make full use of local image information, our technique is more accurate than previous incremental techniques, and is very close to, and considerably faster than, global bundle adjustment. Experiments with both synthetic and real data have been conducted to compare the proposed technique with other techniques, and have shown our technique to be clearly superior.

Keywords: Motion analysis, incremental estimation, bundle adjustment, first order approximation, optimization embedding.

1. Introduction

In this paper, we deal with the problem of motion and structure estimation from long image sequences, assuming that feature correspondences (point matches in our case) across all images have been established. Sequential matchers have proven to be the most successful (see e.g. [17, 3, 10, 23]), but this is not the topic of this paper. It is sufficient to say here that we do not assume that a point feature appears in all images. The length of a point track can be any value equal or larger than 2.

The optimal way to recover motion and structure from long sequences is to use bundle adjustment which involves minimization of reprojection errors [6, 12, 10]. The reader is referred to [20] for an excellent survey of the theory of bundle adjustment as well as many implementation strategies. However, bundle adjustment does not give a direct solution; it is a refining process and requires a good starting point. The starting point can be obtained with some sub-optimal incremental approaches to be mentioned below. Incremental approaches usually are also preferable for time-critical applications because bundle adjustment is computationally more expensive.

If a parallel projection model is used (e.g., orthographic, weak perspective or affine cameras), a direct and optimal solution can be obtained through the factorization method [19, 14]. However, all features are assumed to be observed in every image throughout the sequence, although the missing point problem is tackled in [9]. Note that factorization-like techniques have also been developed for general perspective cameras [18, 8], but they only minimize an algebraic error and are thus not optimal.

There are mainly two categories of incremental techniques. The first is based on *Kalman filtering* [24, 2, 11, 21]. Because of the nonlinearity between motion-structure and image features, an extended Kalman filter is used. The final result then depends on the order in which the image features are supplied, and the error variance of the estimated motion and structure is usually larger than the bundle adjustment.

The work to be described in this paper falls into the second category which can be called *subsequence concatenation*, and our work is closely related to [1, 4]. Avidan and Shashua [1] proposed a "threading" operation that connects two consecutive fundamental matrices using the trifocal tensor [16]. The threading operation is applied to a sliding window of triplets of images, and the camera matrix of the third view is computed from at least 6 point matches across the three views and the fundamental matrix between the first two views. Because of use of algebraic distances, the estimated motion is not statistically optimal. Fitzgibbon and Zisserman [4] also proposed to use sub-sequences of triplets of images. The difference is that bundle adjustment is conducted for each triplet to estimate the trifocal tensor and successive triplets are stitched together into a whole sequence. A final bundle adjustment can be conducted to improve the result if necessary. Two successive triplets can share zero, one or two images, and the stitching quality depends on the number of common point matches across six, five or four images, respectively. The number of common point matches over a sub-sequence decreases as the length of the sub-sequence increases; this means that the stitching quality is lower when the number of overlapping images is smaller. Furthermore, with two overlapping images, there will be two inconsistent camera motion estimates between the two images, and they resort to an additional nonlinear minimization to maximize camera consistency.

As mentioned above, the two pieces of work on subsequence concatenation rely on point matches across three or more views. Point matches between two views, although they are more common, are ignored. In this paper, we propose a new incremental structure-from-motion algorithm which also works on a sliding window of triplets of images, but unlike [1], we also take into account those points that only match across two views. Furthermore, the motion is locally estimated in a statistically optimal way. We adapt three-view bundle adjustment to our incremental estimation problem, and call the new formulation *local bundle adjustment*. Experimental results show that our new incremental algorithm gives better results than previous incremental algorithms, and gives results very close to those obtained with the global bundle adjustment but in a fraction of time (in an experiment with 61 images, a gain of almost 700 times was observed).

An obvious alternative for incremental motion estimation is to apply two-view structure-from-motion recursively to a long sequence. As we will demonstrate experimentally, use of triplets of images rather than pairs of images gains considerable robustness. This is because a point match across three views provides *three* constraints on camera motions if they are considered integrally, but only *two* if they are considered as two consecutive pairs.

A good incremental motion estimation algorithm is very important for many applications:

- An obvious application is to supply a good starting point for the global bundle adjustment which is a highly nonlinear refining process.
- In many time-critical applications such as visual navigation, we are not afford to a global bundle adjustment.
- When an image sequence is dominated by short feature tracks (i.e., overlap between successive images is small), the global optimization degenerates into several weakly correlated local processes. The local bundle adjustment is close to be optimal.
- In some computer graphics applications, local consistency is more important than global consistency [15]. For example, due to errors in calibration and feature detection, a global 3D model may be not good enough to render photorealistic images. Approaches such as "view-dependent geometry" which rely on local 3D model may be preferable. Our algorithm should be very useful in this area.

The remaining paper is organized as follows. Section 2 states the problem we want to solve and describes the local bundle adjustment. Section 3 describes two techniques to speed up the local bundle adjustment by taking advantage of the sparse structure of the adjustment network [20]. The first, called *optimization embedding*, is to embed the optimization of structure parameters in the optimization of the camera's motion/pose parameters. The second is to eliminate all structure parameters through linearization. Both techniques transform the original local bundle adjustment involving both 3D structure and pose parameters into a much smaller problem. We provide the experimental results in Section 4.

2. Problem Statement and Local Bundle Adjustment

In this section, we first introduce the notation, provide an overview of our incremental motion estimation algorithm, and finally focus our effort on local bundle adjustment for three views.

2.1. Notation

An image point is denoted by $\mathbf{p} = [u, v]^T$, and a point in space is denoted by $\mathtt{P} = [X, Y, Z]^T$. For homogeneous coordinates, we use $\widetilde{\mathbf{x}} = [\mathbf{x}^T, 1]^T$ for any vector $\mathbf{x}$.

Image I_i is taken by a camera with unknown pose parameters $\mathtt{M}_i$ which describes the orientation (rotation matrix $\mathbf{R}_i$) and position (translation vector $\mathbf{t}_i$) of the camera with respect to the world coordinate system in which 3D points are described. The relationship between a 3D point $\mathtt{P}$ and its projection $\mathbf{p}_i$ in image i is described by a 3×4 projection matrix $\mathbf{P}_i$, and is given by

$$s\widetilde{\mathbf{p}}_i = \mathbf{P}_i \widetilde{\mathtt{P}} \ , \tag{1}$$

where s is a non-zero scale factor. In general, $\mathbf{P}_i = \mathbf{A}_i[\mathbf{R}_i \ \mathbf{t}_i]$, where the 3×3 matrix $\mathbf{A}_i$ contains the camera's internal parameters. If the camera is calibrated ($\mathbf{A}_i$ is known), we can work with normalized image coordinates and set $\mathbf{A}_i$ to the identity matrix. In the following discussion, it is sometimes more convenient to describe the nonlinear projection (1) by function $\boldsymbol{\phi}_i$ such that

$$\mathbf{p}_i = \boldsymbol{\phi}(\mathtt{M}_i, \mathtt{P}) \ . \tag{2}$$

To be general, if some camera parameters such as focal length is unknown, then the pose parameter vector $\mathtt{M}_i$ should also include them, besides the 3 parameters for rotation and the 3 parameters for translation.

2.2. Overview of the Incremental Motion Estimation Algorithm

Given an image sequence $\{I_i | i = 0, \ldots, N-1\}$, points of interest are extracted from each image using for example Harris' corner detector [5]. Point matches between successive images are established with any technique mentioned for example at the beginning of the introduction. Our algorithm for motion estimation works as the follows:

1. Choose the camera coordinate system associated with I_0 as the world coordinate system, i.e., $\mathbf{R}_0 = \mathbf{I}$ and $\mathbf{t}_0 = \mathbf{0}$.
2. For I_1, compute the motion $\mathtt{M}_1$ using a two-view structure-from-motion technique based on minimizing reprojection errors.

3. For I_i ($i \geq 2$), determine the motion $\mathtt{M}_i$ by applying the local bundle adjustment, to be described below, to the triplet of (I_{i-2}, I_{i-1}, I_i). Point matches only shared by two views as well as those across three views are used.

As is clear, this algorithm only determines the motion parameters of one image (the last one) at each time instant through local bundle adjustment, thus guaranteeing that the estimated consecutive camera motions are consistent with a single 3D model defined in the first camera coordinate system.

Note that when the motion between I_0 and I_1 is small, the motion estimate $\mathtt{M}_1$ from step 2 may not be very accurate, and one can obtain better results by conducting a bundle adjustment on the first three views I_0, I_1 and I_2.

2.3. Local Bundle Adjustment

Let us consider three views (I_{i-2}, I_{i-1}, I_i). To simplify notation, we only consider $i = 2$, and the result extends naturally to $i > 2$.

We are given two sets of point matches[1]: the first contains point matches across all three views, and is denoted by $\Omega = \{(\mathbf{p}_{0,j}, \mathbf{p}_{1,j}, \mathbf{p}_{2,j}) | j = 1, \ldots, M\}$; the second contains point matches only between I_1 and I_2, and is denoted by $\Theta = \{(\mathbf{q}_{1,k}, \mathbf{q}_{2,k}) | k = 1, \ldots, N\}$. The camera matrices $\mathbf{P}_0$ and $\mathbf{P}_1$ (or equivalently $\mathtt{M}_0$ and $\mathtt{M}_1$) are already known, and we need to determine the camera matrices $\mathbf{P}_2$ (or equivalently $\mathtt{M}_2$).

Our objective is to solve $\mathtt{M}_2$ in an optimal way by minimizing some statistically and/or physically meaningful cost function. A reasonable assumption is that the image points are corrupted by independent and identically distributed Gaussian noise because points are extracted independently from images by the same algorithm. In that case, the maximum likelihood estimation is obtained by minimizing the sum of squared errors between the observed image points and the predicted feature points. More formally, the problem becomes

$$\min_{\mathtt{M}_2, \{\mathtt{P}_j\}, \{\mathtt{Q}_k\}} \left(\sum_{j=1}^{M} \sum_{i=0}^{2} \|\mathbf{p}_{i,j} - \boldsymbol{\phi}(\mathtt{M}_i, \mathtt{P}_j)\|^2 + \sum_{k=1}^{N} \sum_{i=1}^{2} \|\mathbf{q}_{i,k} - \boldsymbol{\phi}(\mathtt{M}_i, \mathtt{Q}_k)\|^2 \right) , \tag{3}$$

where $\mathtt{P}_j$ is the 3D point corresponding to triple point match $(\mathbf{p}_{0,j}, \mathbf{p}_{1,j}, \mathbf{p}_{2,j})$, and $\mathtt{Q}_k$ is the 3D point corresponding to double point match $(\mathbf{q}_{1,k}, \mathbf{q}_{2,k})$.

This is a minimization problem over a large dimensional space; the number of dimensions is equal to $6 + 3(M + N)$. In the next section, we will describe two techniques to speed up the minimization.

3. Reducing the Local Bundle Adjustment

Examining problem (3) carefully, we can find an important property: the unknown 3D point structures are independent from each other. This reflects in the sparse structure of the Jacobian and Hessian of the objective function (3), and one can realize very great time savings by explicitly taking advantage

[1]Strictly speaking, there are two more sets of point matches for three views. One is those only between I_0 and I_1, but they do not contribute to the estimation of motion $\mathtt{M}_2$. Another is those only between I_0 and I_2. However, as we only consider sequential matching, this set of matches is not available. In case it is available, we can treat it exactly in the same way as for the set of matches between I_1 and I_2.

of the sparseness during minimization [20, 7]. However, the sparse minimization algorithm is quite complex, and we have not yet implemented it. In this section, we describe two alternative techniques.

3.1. Optimization Embedding

Because of independence between the 3D point structures, problem (3) is equivalent to

$$\min_{\mathtt{M}_2}\left(\sum_{j=1}^{M}\min_{\mathtt{P}_j}\sum_{i=0}^{2}\|\mathbf{p}_{i,j}-\boldsymbol{\phi}(\mathtt{M}_i,\mathtt{P}_j)\|^2 + \sum_{k=1}^{N}\min_{\mathtt{Q}_k}\sum_{i=1}^{2}\|\mathbf{q}_{i,k}-\boldsymbol{\phi}(\mathtt{M}_i,\mathtt{Q}_k)\|^2\right) . \tag{4}$$

The outer minimization is for estimating the camera pose parameters while the inner minimizations are for reconstructing 3D structures. That is, we can separate the structure parameters from the motion parameters such that the optimization of the structure parameters is conducted, independently for each point, in each optimization iteration for the motion parameters. Therefore, a problem of minimization over $6 + 3(M + N)$ dimensional space becomes a problem of minimization over 6D space, in the latter each iteration contains $(M + N)$ independent optimizations of 3 structure parameters. Recall that the computational complexity of minimization is usually n^3 in the number of parameters n. The computation is thus considerably reduced by optimization embedding. Note that problem (4) is exactly the same as problem (3); there is no approximation. This is different from the approximate bundle algorithms which alternate steps of resection (finding the camera poses from known 3D points) and intersection (finding the 3D points from known camera poses).

3.2. Linearizing Local Bundle Adjustment

Further computational gain can be achieved if we can eliminate the inner $(M + N)$ independent minimizations in (4). There is, however, no closed-form solution to 3D reconstruction based on this geometric errors. Fortunately, as to be shown in Appendix A, by linearization we can eliminate all the unknown structure parameters $\mathtt{P}_j$ and $\mathtt{Q}_k$ to obtain very close approximations to the geometric errors in (4), i.e.,

$$\mathcal{J}'_j(\{\mathbf{p}_{i,j},\mathtt{M}_i|i=0,1,2\}) \approx \min_{\mathtt{P}_j}\sum_{i=0}^{2}\|\mathbf{p}_{i,j}-\boldsymbol{\phi}(\mathtt{M}_i,\mathtt{P}_j)\|^2$$

$$\mathcal{L}'_k(\{\mathbf{q}_{i,k},\mathtt{M}_i|i=1,2\}) \approx \min_{\mathtt{Q}_k}\sum_{i=1}^{2}\|\mathbf{q}_{i,k}-\boldsymbol{\phi}(\mathtt{M}_i,\mathtt{Q}_k)\|^2$$

where $\mathcal{J}'_j$ and $\mathcal{L}'_k$ are given by (30) and (35). They do not contain any structure parameters. Problem (4) now becomes the following minimization problem:

$$\min_{\mathtt{M}_2}\left(\sum_{j=1}^{M}\mathcal{J}'_j + \sum_{k=1}^{N}\mathcal{L}'_k\right) . \tag{5}$$

Notice that, using a first order approximation, we have transformed the original large problem into a minimization over just the six-dimensional camera pose space.

3.3. Initialization

We have implemented the above techniques using the Levenberg-Marquardt algorithm from Minpack [13]. The initial guess is obtained as follows. We only use point matches across three views. We first reconstruct them in 3D space using points between the first and second view because their camera poses are known. We then compute the camera pose for the third view using 3D-2D correspondences. In both stages, we start with a linear solution, followed by a refining based on the distances between the observed image points and the predicted feature points.

4. Experimental Results

In this section, we provide experimental results with both synthetic (Sect. 4.1) and real data (Sect. 4.2 to Sect. 4.4).

We consider a number of algorithms:

Method L: An implementation of the exact local bundle adjustment, i.e., the optimization embedding (4), in which the independency of point structures (or equivalently the sparseness in the Jacobian and Hessian matrices) has been taken into account.

Method O: Our reduced local bundle adjustment (5), where structure parameters have been eliminated.

Method I: The algorithm used to initialize our local bundle adjustment, as described in Sect. 3.2. Only matches shared by three views are used. This is equivalent to the approach reported in [1], except that we use the reprojection errors, instead of algebraic errors.

Method II: A two-view incremental algorithm. Motion is estimated only from point matches between the last two images, and the undetermined scale is computed using the common matches shared with those between the previous image pair.

Method B: The global bundle adjustment which gives the statistically optimal solution. The independency of point structures has been taken into account by optimization embedding. The result obtained with Method O is used to initialize this method.

The above algorithms have been implemented with VC++, and all experiments reported in this section were conducted on a Pentium III 850 machine.

The synthetic data was used to compare Method O and method L. It was generated from a real data of a 3-image subsequence (first 3 images) of the STN31 sequence (see below) in the following way. First, we reconstructed the 3D point for each track with the real feature points and camera motions. Second, original feature points in a track were replaced by the projected 2D points of the reconstructed 3D points. After this replacement, the camera motions and the 3D points in the real data became our ground truth.

Four real sequences named STN31, STN61, DYN40, and FRG21 were used in the real experiments. STN61 is a 61-image sequence of a stone sculpture on a turntable, of which the rotation angle can be accurately controlled. STN31 is a 31-image sequence subsampled from the original STN61 sequence. Both STN31 and STN61 are closed-loop sequences, meaning that the first image and the last images coincide (i.e., their motion is zero). DYN40 is a 40-image sequence taken with a multi-camera rig (provided by kind permission of Dayton Taylor). FRG21 is a 21-frame sequence taken by a hand held camera targeting a toy frog. The camera intrinsic parameters for STN31, STN61, and FRG21 were calibrated in advance, and those for DYN40 were self-calibrated and thus less accurate. While the algorithms tested in this section do not require any special motion sequences, we intentionally chose orbital or near orbital ones in order to make the experiment results easily interpretable.

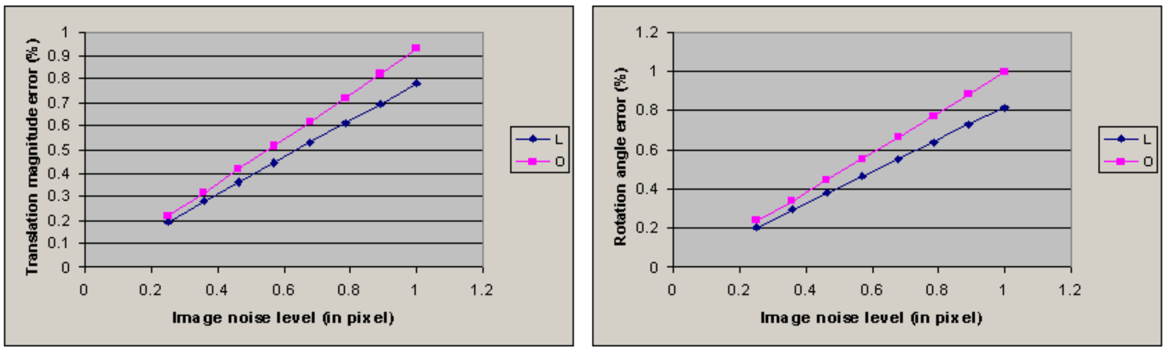

Figure 1: *Rotation and translation errors. Left: Errors of the translation magnitude vs. image noise level. Right: Errors of the rotation angle vs. image noise level*

	$\overline{\alpha}$	σ_α	$\overline{\mathbf{r}}$	$\overline{\mathbf{t}}$	$\sigma_{\mathbf{t}}$
O	12.510	0.384	-0.008, 0.999, 0.052	1.030, 0.002, 0.099	1.63e-3, 1.18e-2, 2.82e-2
B	12.253	0.228	0.001, 0.999, 0.054	1.021, -0.005, 0.093	4.06e-3, 1.17e-2, 1.59e-2
I	12.072	0.886	-0.017, 0.998, 0.065	1.012, 0.008, 0.095	8.29e-3, 1.97e-2, 7.46e-2
II	12.621	0.981	-0.004, 0.999, 0.048	1.002, 0.000, 0.097	8.31e-3, 1.43e-2, 9.24e-2

Table 1: *Ground truth comparisons*

With real data, we will compare methods O, I, II, and B. Section 4.2 gives detailed quantitative and visual comparisons of the four methods with help of some STN31's properties. Section 4.3 shows the visual comparisons of Method O and Method B with the other three sequences. Section 4.4 presents the quantitative comparisons of the four methods in terms of the projection errors and running time.

4.1. Synthetic Data

We compare Method L and Method O with the synthetic data. There are about 200 points and the interframe rotation angle is about 12 degrees. The two methods are used to compute the third camera motion under different image noise levels. We used the relative rotation angle and the translation magnitude with respect to the ground truth as the error metrics of the estimated motion. At each image noise level, we run both methods 30 times, and the mean differences between the computed motion and the ground truth motion were recorded. Average running time was recorded in a similar way. Figure 1 shows the rotation errors and the translation magnitude errors for both methods. We can see from the figure that the errors with Method O are only slightly higher than Method L. It is less than 0.2% larger even when noise with 1 pixel standard deviation was added. On the other hand, this accuracy has been achieved with 30 times less computational time as can be seen from Fig. 2. This shows that the mathematical procedure described in Sect. 3 to eliminate all structure parameters is indeed of benefit.

4.2. Ground Truths Comparisons

The STN31 sequence is used. There are several things we know for sure, within the control accuracy offered for our turntable, about the STN31 sequence. First, the camera is moving on a circle and the camera motion of the first image is overlapped with the last one. Second, the relative rotation angle and translation vector between two consecutive images are the same throughout the sequence. We

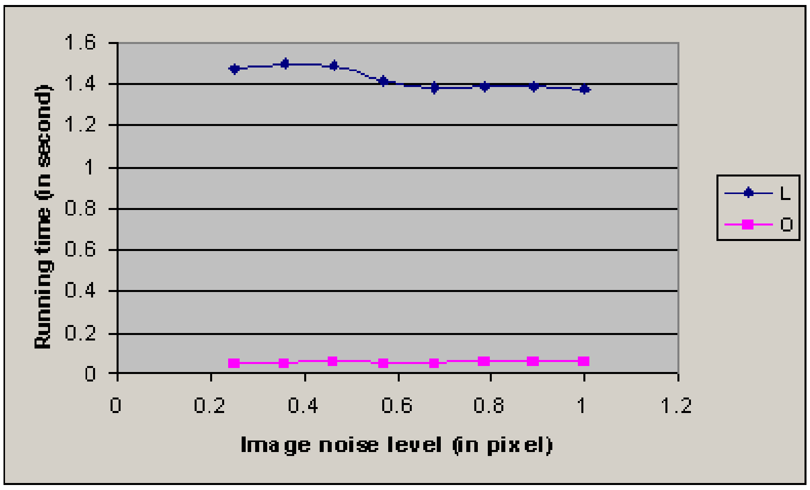

Figure 2: *Running time*

also roughly know that relative rotation angle is around 12.414°, and the rotation axis is the closed to $[0, 1, 0]$. The results are listed in the Table 1, where $\overline{\alpha}$ is the mean rotation angle between successive two frames (in degrees), σ_α is the sample deviation of α, $\overline{\mathbf{r}}$ is the mean rotation axis (in the order of x, y, and z), $\overline{\mathbf{t}}$ is the mean translation vector, and $\sigma_\mathbf{t}$ are the square roots of the diagonal elements of the covariance matrix of the translation. It can be seen from the table that the proposed method is superior to both Method I and Method II in terms of the accuracy of both rotation angle and translation vector, e.g., it has smaller σ_α and $\sigma_\mathbf{t}$. On the other hand, the results of our method are in general very close to those given by the global bundle adjustment method.

Figure 3 shows the visual comparison of the estimated camera motions for the STN31 sequence. Each plane in the figure represents a focal plane, of which the center is the optical center of the camera and the normal is the direction of the optical axis. The absolute size of the plane is not meaningful and has been properly adjusted for display purpose. According to the first ground truth mentioned earlier, the focal plane arrays should form a closed circle if the camera motions are computed correctly. This conforms quite good with the results of both our method and the bundle adjustment, with the exception that the overlapping between the first focal plane and the last one is a little bit overhead (see the small black images in Fig. 3 for details). This reveals that accumulation error is inevitable in both cases. (Note that we did not use the knowledge that the last image is the same as the first image.) Obviously, the results with the other two methods are much less accurate.

4.3. Visual Comparisons with Bundle Adjustment

This subsection concentrates on the visual comparison of our method against the bundle adjustment with the remaining three real image sequences. In brief, Fig. 4, Fig. 5, and Fig. 6 show the results of STN61, DYN40, and FRG21, respectively. Together with Fig. 3, these figures will be used as the complementary results for the quantitative comparisons in the next subsection.

4.4. Comparisons on Projection Errors and Running Time

We now compare the different methods in terms of the projection errors and running time. The projection errors were computed from a tracked feature table and the camera motions. The tracked feature

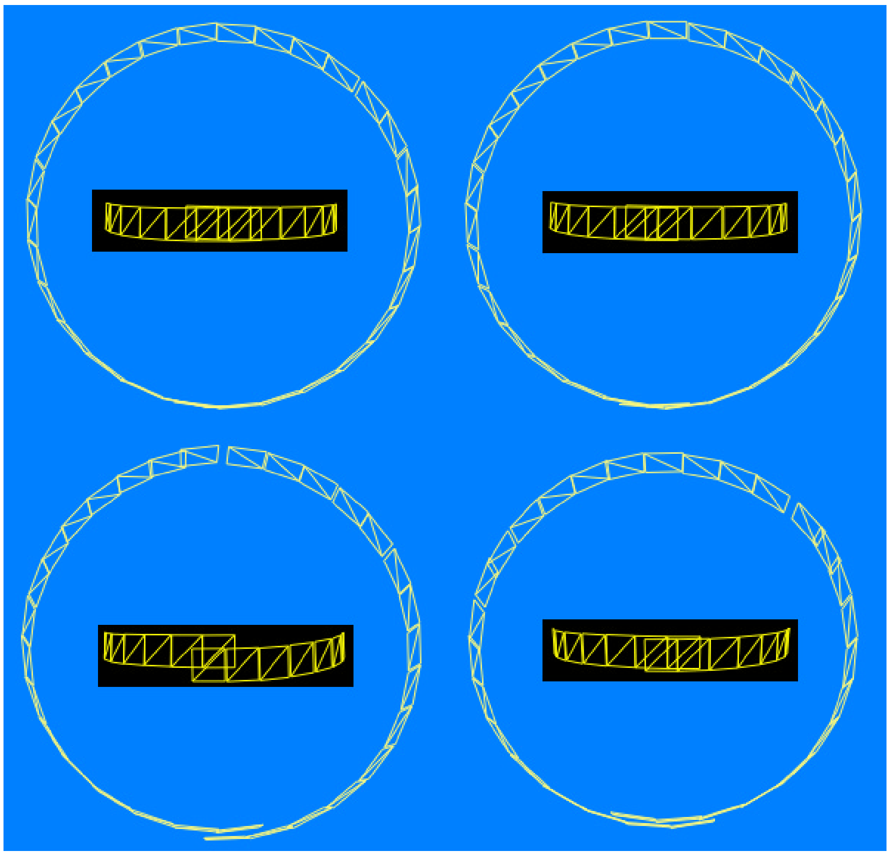

Figure 3: *Camera motions for the STN31 sequence. Top left: Method B; Top right: Method O. Bottom left: Method I. Bottom right: Method II. The small images with black backgrounds show, from the frontal direction, the portion where the starting and the ending of the sequence meet.*

table contains a list of feature tracks. Each track is a list of 2D point features that shared by different views. The camera motions were computed with the four different methods listed in the top row of Table 2. The camera motions were used to reconstruct optimal 3D points from the tracked feature table. These 3D points were then projected to the images to compute the projection errors for the whole sequence. Since the tracked feature table is the same for all methods, the method that produces good motions will have a small projection error. The number of 2D points in the tracked feature tables are 7990, 26890, 11307, and 6049 for the STN31, STN61, DYN40, and FRG21, respectively. The projection errors are displayed in the top of each cell of the table. The error used here is the *root mean square error* in pixels. The running time (in seconds) of each algorithm is displayed under each corresponding projection error. The numbers below the sequence names are the estimated average rotation angle between two successive frames. For example, the rotation angle between two successive frames in DYN40 is only about 2.89 degrees.

Several observations can be made from the Table 2. As compared with other two local processes, the proposed method is in general more accurate. It is slower than the Method I but is much faster than the Method II. For the DYN40 sequence, the relative rotation angle is small and camera intrinsic parameters are less accurate than in other sequences. Method I failed in the middle of the sequence because errors accumulated from the previous frames became too large to obtain usable 3D reconstruction for pose determination. Our method, however, can significantly improve each local initial

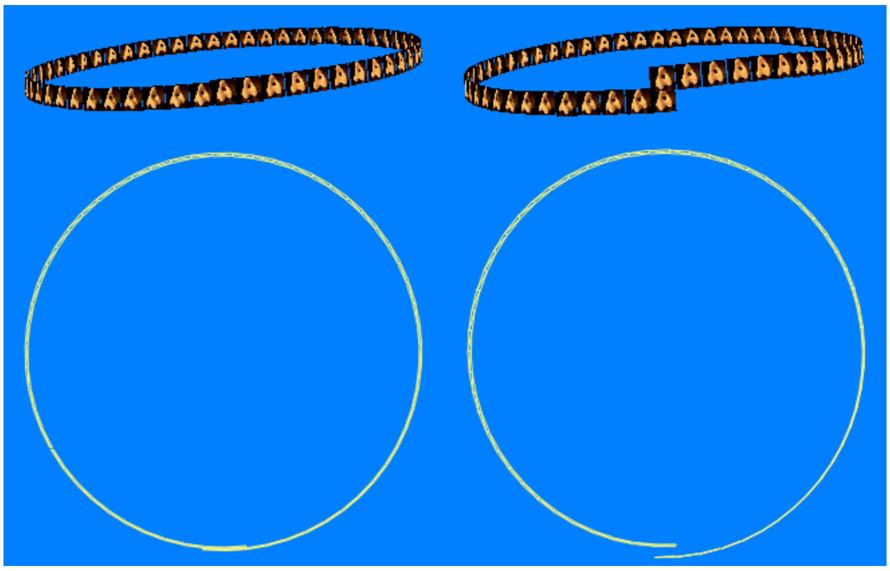

Figure 4: *Camera motions for the STN61 sequence. Left: side and top views of bundle adjustment results. Right: side and top views of our results*

guess (from three views) from the Method I and give a reasonable result for the whole sequence. This demonstrated the robustness of our method. When compared with the classical bundle adjustment algorithm, our method can give very similar results. This is true especially when the rotation angle is relatively large. Similar observations can also be made by inspecting the visual comparison results from Fig. 3, Fig. 4, Fig. 6, and Fig. 5. More importantly, our method has almost linear computational complexity w.r.t. the number of images inside the sequence. This can be seen from the STN31 and STN61 experiments, where the running time was doubled (5.250/2.578 = 2.04) when the number of frames increased from 31 to 61. On the contrary, the classical bundle adjustment algorithm was almost 8 times slower when dealing with a sequence only twice longer.

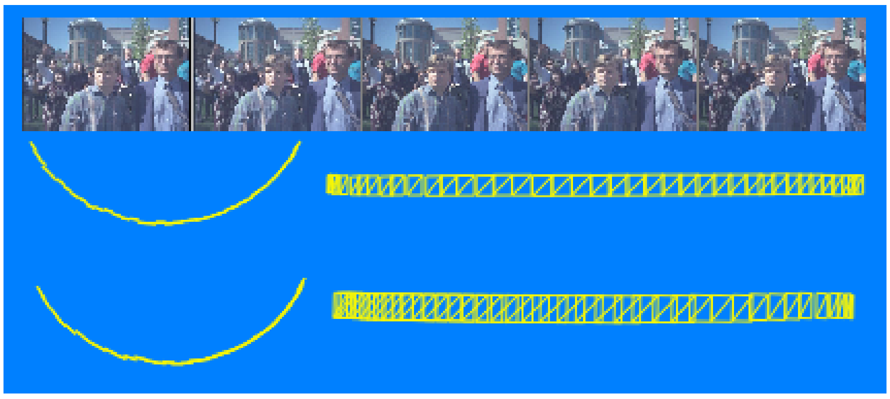

Figure 5: *Camera motions for the DYN40 sequence. Top: sample images from the sequence. Middle: top and front views of the bundle adjustment results. Bottom: top and front views of our results*

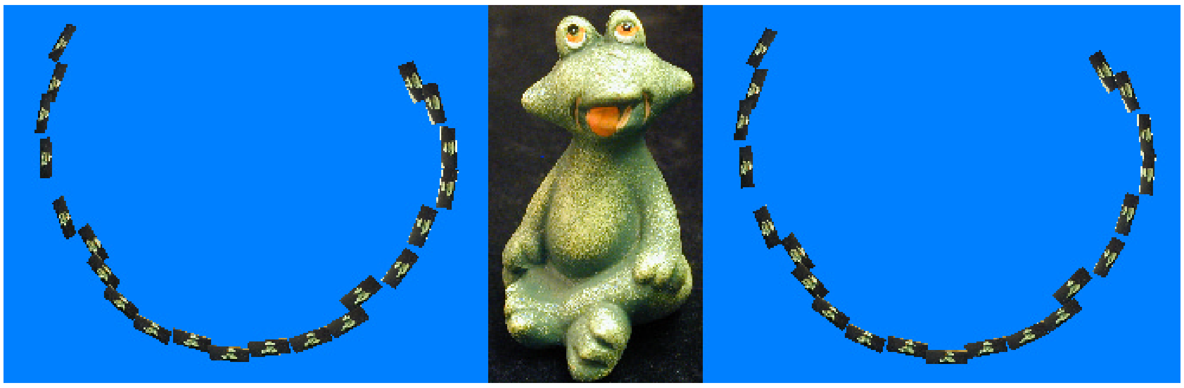

Figure 6: *Camera motions for the FRG21 sequence. From left to right: our results, an sample image from the sequence, bundle adjustment results*

	I	II	O	B
STN31 12.41	0.412 1.219	2.715 30.64	0.323 2.578	0.306 476.038
STN61 6.21	0.389 3.797	2.877 67.076	0.354 5.250	0.321 3739.193
DYN40 2.89	- -	- -	1.149 2.07	0.673 1250.163
FRG21 11.94	0.372 0.907	31.003 25.922	0.362 2.56	0.345 240.414

Table 2: *Projection errors and the running time*

5. Conclusion and Extension

We have proposed a new incremental motion estimation algorithm through a series of local bundle adjustments. Unlike previous approaches, we use not only point matches across three views, but also those points shared only by two views. This is important because matches between two views are more common than those across more views. The estimated camera motions in the whole sequence is guaranteed to be consistent with each other. Two implementations have been described: optimization embedding and structure elimination through linearization. Both techniques transform the original local bundle adjustment involving both 3D structure and pose parameters into a much smaller problem. The linearization achieves an even higher computational gain with only a little sacrifice in accuracy. Experiments with both synthetic and real data showed that our method is more accurate than other local processes, and is much faster (100 to 700 times) than the global bundle adjustment. Furthermore, the results produced by our method are very close to those by the global bundle adjustment, especially when motions between consecutive views are large.

Besides some obvious applications such as to supply a good starting point for the global bundle adjustment, we are expecting to integrate this method into some image-based rendering techniques, where local consistency and accuracy are preferred.

A. Approximating Geometric Errors through Linearization

In this section, we will show how to use a first order approximation to eliminate all the unknown structure parameters $\{\mathtt{P}_j\}$ and $\{\mathtt{Q}_k\}$ in (4).

A.1. Two-view and Three-view Geometry

Before we go further, we need to introduce some notation regarding two-view and three-view geometry. The reader is referred to a recent book [7] for a complete exposition.

Epipolar constraint. In order for a pair of points $(\mathbf{p}_i, \mathbf{p}_{i+1})$ between I_i and I_{i+1} to be matched (or in other words to correspond to a single point in space), they must satisfy the epipolar constraint:

$$\widetilde{\mathbf{p}}_{i+1}^T \mathbf{F}_{i,i+1} \widetilde{\mathbf{p}}_i = 0 \ , \tag{6}$$

where the fundamental matrix $\mathbf{F}_{i,i+1}$ is given by: (see [22])

$$\mathbf{F}_{i,i+1} = \left[\mathbf{P}_{i+1}\mathbf{c}_i\right]_\times \mathbf{P}_{i+1}\mathbf{P}_i^+ \ . \tag{7}$$

Here, $\mathbf{c}_i$ is a null vector of $\mathbf{P}_i$, i.e., $\mathbf{P}_i\mathbf{c}_i = \mathbf{0}$. Therefore, $\mathbf{c}_i$ indicates the position of the optical center of image I_i. $\mathbf{P}^+$ is the pseudo-inverse of matrix $\mathbf{P}$: $\mathbf{P}^+ = \mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}$. $[\mathbf{x}]_\times$ denotes the 3×3 antisymmetric matrix defined by vector $\mathbf{x}$ such that $\mathbf{x} \times \mathbf{y} = [\mathbf{x}]_\times \mathbf{y}$ for any 3D vector $\mathbf{y}$. When cameras' internal parameters are known and we work with normalized image coordinates, then the fundamental matrix becomes the essential matrix, and $\mathbf{F}_{i,i+1} = [\mathbf{t}_{i,i+1}]_\times \mathbf{R}_{i,i+1}$, where $(\mathbf{R}_{i,i+1}, \mathbf{t}_{i,i+1})$ is the relative motion between I_i and I_{i+1}.

Image point transfer across three views. Given the camera projection matrices $\mathbf{P}_i$ $(i = 0, 1, 2)$ and two points $\mathbf{p}_0$ and $\mathbf{p}_1$ in the first two images respectively, their corresponding point in the third image, $\mathbf{p}_2$, is then determined. This can be seen as follows. Denote their corresponding point in space by $\mathtt{P}$. According to the camera projection model (1), we have $s_0\widetilde{\mathbf{p}}_0 = \mathbf{P}_0\widetilde{\mathtt{P}}$ and $s_1\widetilde{\mathbf{p}}_1 = \mathbf{P}_1\widetilde{\mathtt{P}}$. Eliminating the scale factors yields

$$[\widetilde{\mathbf{p}}_0]_\times \mathbf{P}_0 \widetilde{\mathtt{P}} = \mathbf{0} \quad \text{and} \quad [\widetilde{\mathbf{p}}_1]_\times \mathbf{P}_1 \widetilde{\mathtt{P}} = \mathbf{0} \ .$$

We can easily solve $\widetilde{\mathtt{P}}$ from these equations. Then, its projection in I_2 is given by $s_2\widetilde{\mathbf{p}}_2 = \mathbf{P}_2\widetilde{\mathtt{P}}$. Combining the above two operations gives us the image transfer function which we will denote by $\boldsymbol{\psi}$, i.e.,

$$\mathbf{p}_2 = \boldsymbol{\psi}(\mathbf{p}_0, \mathbf{p}_1) \ . \tag{8}$$

Additional notation. Besides the notation defined in Sect. 2.1, we need to introduce a few more symbols. We define

$$\breve{\mathbf{p}} = \begin{bmatrix} \mathbf{p} \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{Z} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T \ . \tag{9}$$

We then have $\breve{\mathbf{p}} = \mathbf{Z}\mathbf{p}$, $\mathbf{p} = \mathbf{Z}^T\breve{\mathbf{p}}$, $\mathbf{Z}\mathbf{Z}^T = \text{diag}\,(1,1,0)$, and $\mathbf{Z}^T\mathbf{Z} = \text{diag}\,(1,1)$.

A.2. Simplifying the Three-View Cost Function

Consider

$$\min_{\mathtt{P}_j} \sum_{i=0}^{2} \|\mathbf{p}_{i,j} - \boldsymbol{\phi}(\mathtt{M}_i, \mathtt{P}_j)\|^2 \tag{10}$$

in (4). To simplify the notation, we will drop the subscript j. Instead of estimating the 3D point, we can estimate its projections in images. Problem (10) is equivalent to estimate $\widehat{\mathbf{p}}_i$ $(i = 0, 1, 2)$ by minimizing the following objective function

$$\mathcal{J} = \|\mathbf{p}_0 - \widehat{\mathbf{p}}_0\|^2 + \|\mathbf{p}_1 - \widehat{\mathbf{p}}_1\|^2 + \|\mathbf{p}_2 - \widehat{\mathbf{p}}_2\|^2 \tag{11}$$

$$\text{subject to (i)} \quad \widetilde{\widehat{\mathbf{p}}}_1^T \mathbf{F}_{0,1} \widetilde{\widehat{\mathbf{p}}}_0 = 0 \tag{12}$$

$$\text{and (ii)} \quad \widehat{\mathbf{p}}_2 = \boldsymbol{\psi}(\widehat{\mathbf{p}}_0, \widehat{\mathbf{p}}_1) \ , \tag{13}$$

where the fundamental matrix $\mathbf{F}_{0,1}$ and the image transfer function $\boldsymbol{\psi}$ are described in Sect. A.1. By applying the technique of Lagrange multiplier, we can convert the above constrained minimization problem into an unconstrained minimization problem with the new objective function given by

$$\mathcal{J}' = \mathcal{J} + \lambda_1 \mathcal{F} + \lambda_2 \mathcal{T} \tag{14}$$

where $\mathcal{F} = \widetilde{\widehat{\mathbf{p}}}_1^T \mathbf{F}_{0,1} \widetilde{\widehat{\mathbf{p}}}_0$ and $\mathcal{T} = \|\widehat{\mathbf{p}}_2 - \boldsymbol{\psi}(\widehat{\mathbf{p}}_0, \widehat{\mathbf{p}}_1)\|^2$.

Define $\Delta\mathbf{p}_i$ ($i = 0, 1, 2$) as $\Delta\mathbf{p}_i = \mathbf{p}_i - \widehat{\mathbf{p}}_i$, then $\widehat{\mathbf{p}}_i = \mathbf{p}_i - \Delta\mathbf{p}_i$, or $\widetilde{\widehat{\mathbf{p}}}_i = \widetilde{\mathbf{p}}_i - \Delta\breve{\mathbf{p}}_i$. Refer to (9) for $\breve{\mathbf{p}}$.

If we neglect the second order term, constraint (12) becomes $\mathcal{F} = 0$, with

$$\begin{aligned}\mathcal{F} &\approx \widetilde{\mathbf{p}}_1^T \mathbf{F}_{0,1} \widetilde{\mathbf{p}}_0 - \Delta\breve{\mathbf{p}}_1^T \mathbf{F}_{0,1} \widetilde{\mathbf{p}}_0 - \widetilde{\mathbf{p}}_1^T \mathbf{F}_{0,1} \Delta\breve{\mathbf{p}}_0 \\ &= \widetilde{\mathbf{p}}_1^T \mathbf{F}_{0,1} \widetilde{\mathbf{p}}_0 - \Delta\mathbf{p}_1^T \mathbf{Z}^T \mathbf{F}_{0,1} \widetilde{\mathbf{p}}_0 - \widetilde{\mathbf{p}}_1^T \mathbf{F}_{0,1} \mathbf{Z} \Delta\mathbf{p}_0 \ .\end{aligned} \tag{15}$$

Here, we have used the equation $\breve{\mathbf{p}} = \mathbf{Z}\mathbf{p}$.

Constraint (13) becomes

$$\mathcal{T} = \|\mathbf{p}_2 - \Delta\mathbf{p}_2 - \boldsymbol{\psi}(\mathbf{p}_0 - \Delta\mathbf{p}_0, \mathbf{p}_1 - \Delta\mathbf{p}_1)\|^2 \tag{16}$$

By applying Taylor expansion to $\boldsymbol{\psi}$ and keeping the first order terms, we have:

$$\boldsymbol{\psi}(\mathbf{p}_0 - \Delta\mathbf{p}_0, \mathbf{p}_1 - \Delta\mathbf{p}_1) \approx \boldsymbol{\psi}(\mathbf{p}_0, \mathbf{p}_1) - \boldsymbol{\Psi}_0 \Delta\mathbf{p}_0 - \boldsymbol{\Psi}_1 \Delta\mathbf{p}_1 \tag{17}$$

where $\boldsymbol{\Psi}_i = \partial\boldsymbol{\psi}(\mathbf{p}_0, \mathbf{p}_1)/\partial\mathbf{p}_i$. Equation (16) then becomes $\mathcal{T} = 0$, with

$$\begin{aligned}\mathcal{T} &\approx \|\mathbf{p}_2 - \Delta\mathbf{p}_2 - \boldsymbol{\psi}(\mathbf{p}_0, \mathbf{p}_1) + \boldsymbol{\Psi}_0 \Delta\mathbf{p}_0 + \boldsymbol{\Psi}_1 \Delta\mathbf{p}_1\|^2 \\ &\approx \mathbf{a}^T\mathbf{a} + 2\mathbf{a}^T \boldsymbol{\Psi}_0 \Delta\mathbf{p}_0 + 2\mathbf{a}^T \boldsymbol{\Psi}_1 \Delta\mathbf{p}_1 - 2\mathbf{a}^T \Delta\mathbf{p}_2\end{aligned} \tag{18}$$

where $\mathbf{a} = \mathbf{p}_2 - \boldsymbol{\psi}(\mathbf{p}_0, \mathbf{p}_1)$, and the second approximation in (18) is achieved by only keeping first order items.

To minimize $\mathcal{J}'$ (14), we let its first-order derivatives with respect to $\Delta\mathbf{p}_i$ be zero, which yields

$$\begin{aligned}\frac{\partial\mathcal{J}'}{\partial\Delta\mathbf{p}_0} &= 2\Delta\mathbf{p}_0 - \lambda_1 \mathbf{Z}^T \mathbf{F}_{0,1}^T \widetilde{\mathbf{p}}_1 + 2\lambda_2 \boldsymbol{\Psi}_0^T \mathbf{a} \\ \frac{\partial\mathcal{J}'}{\partial\Delta\mathbf{p}_1} &= 2\Delta\mathbf{p}_1 - \lambda_1 \mathbf{Z}^T \mathbf{F}_{0,1} \widetilde{\mathbf{p}}_0 + 2\lambda_2 \boldsymbol{\Psi}_1^T \mathbf{a} \\ \frac{\partial\mathcal{J}'}{\partial\Delta\mathbf{p}_2} &= 2\Delta\mathbf{p}_2 - 2\lambda_2 \mathbf{a} \ .\end{aligned}$$

This gives

$$\Delta\mathbf{p}_0 = \frac{1}{2}\lambda_1 \mathbf{Z}^T \mathbf{F}_{0,1}^T \widetilde{\mathbf{p}}_1 - \lambda_2 \boldsymbol{\Psi}_0^T \mathbf{a} \tag{19}$$

$$\Delta\mathbf{p}_1 = \frac{1}{2}\lambda_1 \mathbf{Z}^T \mathbf{F}_{0,1} \widetilde{\mathbf{p}}_0 - \lambda_2 \boldsymbol{\Psi}_1^T \mathbf{a} \tag{20}$$

$$\Delta\mathbf{p}_2 = \lambda_2 \mathbf{a} \ . \tag{21}$$

Substituting them into the linearized constraints (15) and (18) gives

$$g_1 - \lambda_1 g_2/2 + \lambda_2 g_3 = 0 \tag{22}$$

$$g_4 + \lambda_1 g_3/2 - \lambda_2 g_5 = 0 \ , \tag{23}$$

where

$$
\begin{aligned}
g_1 &= \widetilde{\mathbf{p}}_1^T \mathbf{F}_{0,1} \widetilde{\mathbf{p}}_0 && (24)\\
g_2 &= \widetilde{\mathbf{p}}_0^T \mathbf{F}_{0,1}^T \mathbf{Z}\mathbf{Z}^T \mathbf{F}_{0,1} \widetilde{\mathbf{p}}_0 + \widetilde{\mathbf{p}}_1^T \mathbf{F}_{0,1} \mathbf{Z}\mathbf{Z}^T \mathbf{F}_{0,1}^T \widetilde{\mathbf{p}}_1 && (25)\\
g_3 &= \mathbf{a}^T \mathbf{\Psi}_1 \mathbf{Z}^T \mathbf{F}_{0,1} \widetilde{\mathbf{p}}_0 + \widetilde{\mathbf{p}}_1^T \mathbf{F}_{0,1} \mathbf{Z} \mathbf{\Psi}_0^T \mathbf{a} && (26)\\
g_4 &= \frac{1}{2} \mathbf{a}^T \mathbf{a} && (27)\\
g_5 &= \mathbf{a}^T (\mathbf{I} + \mathbf{\Psi}_0 \mathbf{\Psi}_0^T + \mathbf{\Psi}_1 \mathbf{\Psi}_1^T) \mathbf{a} \,. && (28)
\end{aligned}
$$

Thus, the solution for λ_1 and λ_2 is

$$\lambda_1 = 2\frac{g_3 g_4 + g_1 g_5}{g_2 g_5 - g_3^2} \qquad \lambda_2 = \frac{g_2 g_4 + g_1 g_3}{g_2 g_5 - g_3^2} \,. \tag{29}$$

Substituting the obtained λ_1, λ_2, $\Delta\mathbf{p}_0$, $\Delta\mathbf{p}_1$ and $\Delta\mathbf{p}_2$ back into (14) finally gives

$$\mathcal{J}' = \frac{1}{4}\lambda_1^2 g_2 - \lambda_1 \lambda_2 g_3 + \lambda_2^2 g_5 = \frac{g_1^2 g_5 + g_2 g_4^2 + 2 g_1 g_3 g_4}{g_2 g_5 - g_3^2} \,. \tag{30}$$

This is our new cost functional for a point match across three views. It is a function of the motion parameters, but does not contain 3D structure parameters anymore.

A.3. Simplifying the Two-View Cost Function

Consider now

$$\min_{\mathtt{Q}_k} \sum_{i=1}^{2} \|\mathbf{q}_{i,k} - \phi(\mathtt{M}_i, \mathtt{Q}_k)\|^2 \tag{31}$$

in (4). To simplify the notation, we will drop the subscript k. Following the same idea as in the last subsection, problem (31) is equivalent to estimate $\widehat{\mathbf{q}}_1$ and $\widehat{\mathbf{q}}_2$ by minimizing the following objective function

$$\mathcal{L} = \|\mathbf{q}_1 - \widehat{\mathbf{q}}_1\|^2 + \|\mathbf{q}_2 - \widehat{\mathbf{q}}_2\|^2 \tag{32}$$

$$\text{subject to} \quad \widetilde{\widehat{\mathbf{q}}}_2^T \mathbf{F}_{1,2} \widetilde{\widehat{\mathbf{q}}}_1 = 0 \tag{33}$$

where the fundamental matrix $\mathbf{F}_{1,2}$ is defined as in section A.1.

Define $\Delta\mathbf{q}_1 = \mathbf{q}_1 - \widehat{\mathbf{q}}_1$ and $\Delta\mathbf{q}_2 = \mathbf{q}_2 - \widehat{\mathbf{q}}_2$. Linearize constraint (33) as in (15). Using the Lagrange multiplier, we can transform the above constrained minimization problem into an unconstrained one:

$$
\begin{aligned}
\mathcal{L}' = {} & \Delta\mathbf{q}_1^T \Delta\mathbf{q}_1 + \Delta\mathbf{q}_2^T \Delta\mathbf{q}_2 + \\
& \lambda(\widetilde{\mathbf{q}}_2^T \mathbf{F}_{1,2} \widetilde{\mathbf{q}}_1 - \Delta\mathbf{q}_2^T \mathbf{Z}^T \mathbf{F}_{1,2} \widetilde{\mathbf{q}}_1 - \widetilde{\mathbf{q}}_2^T \mathbf{F}_{1,2} \mathbf{Z} \Delta\mathbf{q}_1) \,.
\end{aligned} \tag{34}
$$

Letting the first-order derivatives be zero gives

$$
\begin{aligned}
\Delta\mathbf{q}_1 &= \frac{\lambda}{2}\mathbf{Z}^T \mathbf{F}_{1,2}^T \widetilde{\mathbf{q}}_2 \qquad \Delta\mathbf{q}_2 = \frac{\lambda}{2}\mathbf{Z}^T \mathbf{F}_{1,2} \widetilde{\mathbf{q}}_1\\
\lambda &= 2\frac{\widetilde{\mathbf{q}}_2^T \mathbf{F}_{1,2} \widetilde{\mathbf{q}}_1}{\widetilde{\mathbf{q}}_1^T \mathbf{F}_{1,2}^T \mathbf{Z}\mathbf{Z}^T \mathbf{F}_{1,2} \widetilde{\mathbf{q}}_1 + \widetilde{\mathbf{q}}_2^T \mathbf{F}_{1,2} \mathbf{Z}\mathbf{Z}^T \mathbf{F}_{1,2}^T \widetilde{\mathbf{q}}_2} \,.
\end{aligned}
$$

Substituting them back into (34), after some simple algebra, gives

$$\mathcal{L}' = \frac{(\widetilde{\mathbf{q}}_2^T \mathbf{F}_{1,2} \widetilde{\mathbf{q}}_1)^2}{\widetilde{\mathbf{q}}_1^T \mathbf{F}_{1,2}^T \mathbf{Z}\mathbf{Z}^T \mathbf{F}_{1,2} \widetilde{\mathbf{q}}_1 + \widetilde{\mathbf{q}}_2^T \mathbf{F}_{1,2} \mathbf{Z}\mathbf{Z}^T \mathbf{F}_{1,2}^T \widetilde{\mathbf{q}}_2} \,. \tag{35}$$

This is our new cost functional for a point match only between two views. It is a function of the motion parameters, but does not contain anymore 3D structure parameters.

References

[1] S. Avidan and A. Shashua. Threading fundamental matrices. In H. Burkhardt and B. Neumann, editors, *Proceedings of the 5th European Conference on Computer Vision*, volume I, pages 124–140, Freiburg, Germany, June 1998.

[2] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):545–561, 1994.

[3] P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In B. Buxton, editor, *Proceedings of the 4th European Conference on Computer Vision*, pages 683–695, Cambridge, UK, Apr. 1996.

[4] A. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In H. Burkhardt and B. Neumann, editors, *Proceedings of the 5th European Conference on Computer Vision*, pages 311–326, Freiburg, Germany, jun 1998.

[5] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conf.*, pages 189–192, 1988.

[6] R. Hartley. Euclidean reconstruction from uncalibrated views. In J. Mundy and A. Zisserman, editors, *Applications of Invariance in Computer Vision*, volume 825 of *Lecture Notes in Computer Science*, pages 237–256, Berlin, Germany, 1993. Springer-Verlag.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[8] A. Heyden and K. Åström. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 438–443, San Juan, Puerto Rico, June 1997. IEEE Computer Society.

[9] D. Jacobs. Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–121, San Juan, Puerto Rico, June 1997. IEEE Computer Society.

[10] P. McLauchlan and D. Murray. A unifying framework for structure and motion recovery from image sequences. In *Proc. Fifth International Conference on Computer Vision*, pages 314–320, Cambridge, Massachusetts, June 1995.

[11] P. McLauchlan, I. Reid, and D. Murray. Recursive affine structure and motion from image sequences. In J.-O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision*, volume I of *Lecture Notes in Computer Science*, pages 217–224, Stockholm, Sweden, May 1994. Springer-Verlag.

[12] R. Mohr, B. Boufama, and P. Brand. Accurate projective reconstruction. In J. Mundy and A. Zisserman, editors, *Applications of Invariance in Computer Vision*, volume 825 of *Lecture Notes in Computer Science*, pages 257–276, Berlin, 1993. Springer-Verlag.

[13] J. More. The levenberg-marquardt algorithm, implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, Lecture Notes in Mathematics 630. Springer-Verlag, 1977.

[14] C. J. Poelman and T. Kanade. A paraperspective factorization for shape and motion recovery. In J.-O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision*, volume B of *Lecture Notes in Computer Science*, pages 97–108, Stockholm, Sweden, May 1994. Springer-Verlag.

[15] P. Rademacher and G. Bishop. Multiple-center-of-projection images. In *Computer Graphics, Annual Conference Series*, pages 199–206. ACM SIGGRAPH, 1998.

[16] A. Shashua. Algebraic functions for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):779–789, 1995.

[17] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, WA, June 1994. IEEE.

[18] P. Sturm and W. Triggs. A factorization based algorithm for multi-image projective structure and motion. In B. Buxton, editor, *Proceedings of the 4th European Conference on Computer Vision*, pages 709–720, Cambridge, UK, Apr. 1996.

[19] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *The International Journal of Computer Vision*, 9(2):137–154, 1992.

[20] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment — a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, pages 298–372, Corfu, Greece, Sept. 1999.

[21] J. Weng, Y. Cui, and N. Ahuja. Transitory image sequences, asymptotic properties, and estimation of motion and structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):451–463, may 1997.

[22] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *The International Journal of Computer Vision*, 27(2):161–195, 1998.

[23] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87–119, Oct. 1995.

[24] Z. Zhang and O. D. Faugeras. Motion and structure from motion from a long monocular sequence. In V. Cantoni, M. Ferretti, S. Levialdi, R. Negrini, and R. Stefanelli, editors, *Progress in Image Analysis and Processing II*, pages 264–271, Como, Italy, Sept. 1991. World Scientific, Singapore.