Simultaneous Optimization of Structure and Motion in Dynamic Scenes Using Unsynchronized Stereo Cameras

Akihito Seki[†] and Masatoshi Okutomi
Graduate School of Science and Engineering,
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo, Japan
a-seki@ok.ctrl.titech.ac.jp,mxo@ctrl.titech.ac.jp

Abstract

In this paper, we propose a simultaneous estimation method of structure and motion in dynamic scenes. Usual methods for obtaining structure and motion using stereo cameras require two kinds of operations: stereo correspondence and tracking. Therefore, we must separately determine the correspondence between stereo images and sequential images. This necessity complicates the algorithm and increases the possibility of mismatches because of the object's motion and visibility change in the images. Our proposed method makes two contributions. The first contribution is the method of corresponding all stereo images and sequential images at once. Therefore, we can obtain the structure and motion simultaneously and more accurately. On the other hand, most stereo correspondence algorithms are limited to use under a synchronized status. In a stereo rig using unsynchronized cameras, as are most commercially available cameras, the structure cannot be obtained by stereo correspondence and triangulation because of the unknown time offset between cameras. Therefore, our second contribution is a method of estimating structure, motion, and time offset simultaneously using unsynchronized stereo cameras. This latter task is accomplished by taking advantage of the first contribution scheme. Additionally, our method requires no preprocessing such as motion segmentation for separating identical-motion objects and advance calibration of the time offset. Finally, we present the experimental results using both synthetic and real images.

1. Introduction

Estimation of the three-dimensional (3D) position and motion given by a moving stereo rig, such as that of a robot and vehicle-mounted cameras in scenes are available for 3D reconstruction, ego-motion estimation, segmentation using spatial information, visual navigation, local 3D map generation, obstacle detection and avoidance, etc.

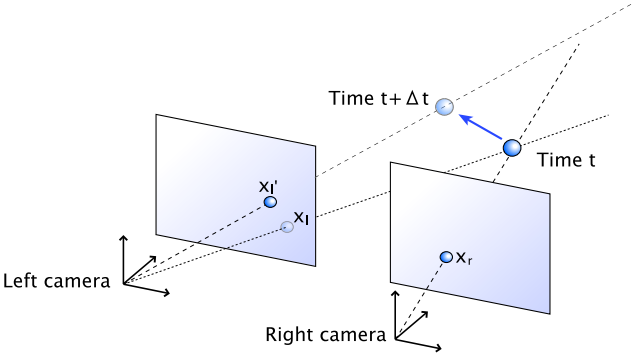

Figure 1. Unsynchronized stereo images and 3D measurement. x_r is the projected point at time t in the right image, x'_l shows the arrangement at time t+Δt in the left image. The rays of x_r and x'_l do not intersect because of the time offset Δt.

The problems of estimating 3D positions in a scene and its 3D motion are closely intertwined. If one or the other of the two were known (e.g. 3D position), other unknown parameters (i.e. 3D motion) could be estimated easily. Usually, however, one or the other of those two is not known accurately: typically, both parameters must be estimated.

In a general environment, two cases are assumed. The first case is that unique 3D motion is observed in stereo images. For example, the cameras are moving in static scenes. Unique 3D motion in stereo images and 3D positions of every point are estimated [2, 4, 5, 7, 11]. However, some different 3D motions exist in real environments and applications of these methods are limited.

The second case is that some different 3D motions are included in the images. The simplest way to estimate the 3D position and 3D motion is that we first obtain the 3D position of a point through conventional stereo correspondence

[†]He belongs to Toshiba Corporate Research & Development Center, now.

1-4244-1180-7/07/$25.00 ©2007 IEEE

between stereo images, and then track the point between sequential images. Then stereo correspondence is performed for the tracked point to obtain the 3D motion. A method developed by Hao et al. estimates motion and depth for every color-based segmented region [3]. Depth and motion are assumed as unique within the same color region.

These methods assume the use of a synchronized stereo camera system. In contrast, commercially available stereo camera systems constructed of commercially available devices, such as web cameras and home video cameras, take stereo images at different timing: they are unsynchronized stereo camera systems. Such unsynchronized systems are not applicable with these methods because, if the target object moves, the spatial localizations of the object differ among stereo images. As shown in Fig. 1, this presents the issue that stereo-correspondent points are not on the epipolar line and the estimated spatial localization is not accurate, even if the points are correctly matched between stereo images. If the 3D position and its 3D motion were estimated using unsynchronized cameras, many applications would thereby be useful with any camera.

Svedman et al. proposed a method for estimation of 3D positions in feature points using an unsynchronized stereo camera [12]. A method developed by Shimizu et al. [9] obtains the 3D position at the frequency of more than the frame rate of one camera by delaying the timing of the stereo camera (as an unsynchronized camera). However, those methods presuppose knowledge of the time offset between stereo cameras.

Zhou et al. [13] introduced a method to estimate the time offset using four-point correspondence between two pairs (i.e. right image at time t and at time t+1 and left image at time t+Δt and t+Δt+1) of unsynchronized stereo images and epipolar geometry. Next, optical flows between sequential images are estimated. A virtual synchronized image is generated by blending the pixel position and its brightness using the estimated time offset and optical flows. Depth is estimated using a conventional stereo correspondence algorithm with the generated virtual synchronized image. This method is applicable in dynamic scenes. However, the four points in the time offset estimation step are separately corresponded; therefore, they do not include depth and motion coherence at the points in the stereo image sequence.

In our proposed method, correspondence of all stereo images and sequential images is done at once. For that reason, we can obtain structure and motion simultaneously. By taking advantage of this correspondence scheme, our method estimates structure, motion, and the time offset simultaneously using unsynchronized stereo cameras. Additionally, the 3D position and 3D motion can be estimated with each point in stereo images. Therefore, our method is applicable to dynamic scenes without preprocessing, such as motion segmentation, for separating identical-motion objects.

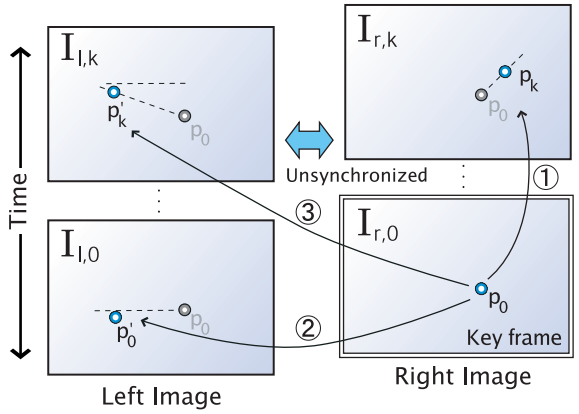

Figure 2. Correspondence between sequential stereo images.

In this paper, we first explain the correspondence formulation between unsynchronized stereo images and define the cost function. Next, we explain minimization of the cost function to estimate depth, 3D motion, and the time offset simultaneously. Finally, we present the experimental results to demonstrate the effectiveness of our method.

2. Correspondence formulation

Sequential stereo images are captured using unsynchronized cameras in dynamic scenes. These scenes typically contain multiple moving objects. Figure 2 shows stereo images $\mathbf{I}_{r,0}$, $\mathbf{I}_{l,0}$ at 0-th frame and subsequent stereo images $\mathbf{I}_{r,k}$, $\mathbf{I}_{l,k}$ at k-th frame. For this study, we assume the following:

- The sequential stereo images are taken at a uniform frame rate within a short time period. Therefore, the captured timing offset (time offset) of image $\mathbf{I}_{r,0}$ and $\mathbf{I}_{l,0}$ is the same as that of image $\mathbf{I}_{r,1}$ and $\mathbf{I}_{l,1}$.
- The extrinsic and intrinsic camera parameters are calibrated.
- The relative 3D motion of an object is assumed to be locally linear uniform motion in 3D space for a short time.

Next, we explain the correspondence relation between the images.

2.1. Relation between identical-camera images

We consider the correspondence relation between images taken at different timing with an identical camera (① in Fig. 2). Let a point $\mathbf{p}_0 = [x, y]^T$ in the key frame have depth Z and 3D motion $\mathbf{T}_m = [T_{mx}, T_{my}, T_{mz}]^T$ per image. The corresponding point $\mathbf{p}_k$ in the k-th frame is repre-

sented as

$$\mathbf{p}_k = \mathbf{p}_0 + \frac{k\mathbf{A}_x\mathbf{T}_m}{Z - kT_{mz}}, \quad (1)$$

where

$$\mathbf{A}_x = \begin{bmatrix} -f & 0 & \hat{x} \\ 0 & -af & \hat{y} \end{bmatrix},\ \hat{x} = x - u_0,\ \hat{y} = y - v_0, \quad (2)$$

and the intrinsic camera parameter $\mathbf{A}_{intr}$ is

$$\mathbf{A}_{intr} = \begin{bmatrix} f & 0 & u_0 \\ 0 & af & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

The epipolar constraint exists between the two images. The constraint is underspecified for unknown 3D motion, but if the motion is known, the epipolar constraint is determined uniquely.

2.2. Relation between different-camera images

Regarding the correspondence relation between unsynchronized stereo cameras (② and ③ in Fig. 2), we first explain the relation of ②. Let the position $\mathbf{p}'_0$ in image $\mathbf{I}_{l,0}$ be the projected point of $\mathbf{p}_0$. Then, $\mathbf{p}'_0$ is represented as

$$\mathbf{p}'_0 = \mathbf{p}_0 + \frac{\mathbf{A}_x\{\lambda\mathbf{T}_m + \mathbf{T}_s\}}{Z - \lambda T_{mz}}, \quad (4)$$

where λ is the time offset between stereo images and $\mathbf{T}_s$ is the camera translation vector.

Next, ③ is explained. The position $\mathbf{p}'_k$, which is in the k-th image after (or before) is given as

$$\mathbf{p}'_k = \mathbf{p}_0 + \frac{\mathbf{A}_x\{(k+\lambda)\mathbf{T}_m + \mathbf{T}_s\}}{Z - (k+\lambda)T_{mz}}. \quad (5)$$

3. Simultaneous estimation algorithm

In this section, the cost function for depth, motion, and time offset is defined. Then, the solution of the cost function is explained.

3.1. Cost function

The corresponding positions of $\mathbf{p}_0$ in key frame $\mathbf{I}_{r,0}$ are determined by minimizing the difference between intensities within a window area W around $\mathbf{p}_0$ and those in sequential stereo images. Therefore, the cost function $E(\mathbf{m})$ is defined below. Minimum E gives unknown parameters $\mathbf{m}$, which consist of depth Z, 3D motion $\mathbf{T}_m$, and time offset λ at point $\mathbf{p}_0$,

$$E(\mathbf{m}) = \sum_{k=0}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{I}_{l,k}(\mathbf{x}+\mathbf{u}_k) - \mathbf{I}_{r,0}(\mathbf{x})\right]^2 + \sum_{k=1}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{I}_{r,k}(\mathbf{x}+\mathbf{u}'_k) - \mathbf{I}_{r,0}(\mathbf{x})\right]^2, \quad (6)$$

where n is a number of sequential frame, vector $\mathbf{m}$ is $[T_{mx}, T_{my}, T_{mz}, Z, \lambda]^T$, $\mathbf{I}_*(\mathbf{x})$ represents the image intensity at $\mathbf{x} = [x, y]^T$, and $\mathbf{u}_k$ and $\mathbf{u}'_k$ are

$$\mathbf{u}_k = \frac{\mathbf{A}_x\{(k+\lambda)\mathbf{T}_m + \mathbf{T}_s\}}{Z - (k+\lambda)T_{mz}},\ \ \mathbf{u}'_k = \frac{k\mathbf{A}_x\mathbf{T}_m}{Z - kT_{mz}}. \quad (7)$$

The first half of eq.(6) indicates SSSD (sum of SSDs) between sequential images of the different cameras; the last half means SSSD between sequential images of the identical camera. Equation (6) is considered to be an application of the multi-baseline stereo concept[6].

3.2. Minimization of the cost function

For minimizing the cost function, we first approximate the equation (6). First, we consider that $\tilde{\mathbf{x}}'' \sim (\mathbf{I} + \mathbf{D})\tilde{\mathbf{x}}$[1], where $\mathbf{I}$ is identity matrix and $\mathbf{D}$ is a 3×3 matrix with small values. First-order Taylor expansion is applied to $\tilde{\mathbf{I}}_{l,k}(\mathbf{x}'')$ [2] around $\mathbf{x}$; therefore, we get

$$\mathbf{I}_{l,k}(\mathbf{x}+\mathbf{u}_k) - \mathbf{I}_{r,0}(\mathbf{x}) = \tilde{\mathbf{I}}_{l,k}(\mathbf{x}'') - \mathbf{I}_{r,0}(\mathbf{x}) \approx \mathbf{g}^T_{l,k}\mathbf{J}^T_{l,k}\Delta\mathbf{m} + e_{lr,k}, \quad (8)$$

where $\mathbf{g}^T_{l,k}$ is the intensity gradient of $\tilde{\mathbf{I}}_{l,k}$, $\mathbf{J}^T_{l,k}$ is the Jacobian matrix, and $e_{lr,k}$ is the intensity difference between $\tilde{\mathbf{I}}_{l,k}$ and $\mathbf{I}_{r,0}$. The detailed derivation is shown in Appendix A.

Identically, we obtain

$$\mathbf{I}_{r,k}(\mathbf{x}+\mathbf{u}'_k) - \mathbf{I}_{r,0}(\mathbf{x}) = \tilde{\mathbf{I}}_{r,k}(\mathbf{x}'') - \mathbf{I}_{r,0}(\mathbf{x}) \approx \mathbf{g}^T_{r,k}\mathbf{J}^T_{r,k}\mathbf{T}^T\Delta\mathbf{m} + e_{rr,k}, \quad (9)$$

where $\tilde{\mathbf{I}}_{r,k}(\mathbf{x})$, $\mathbf{g}^T_{r,k}$, $\mathbf{J}^T_{r,k}$, and $e_{rr,k}$ are the symbols for which subscript l are transposed to r in eq.(8). Also, the Jacobian $\mathbf{J}^T_{r,k}$ and $\mathbf{T}$ are described in Appendix B.

3.2.1 Solution using one point

If the observation point is considered as one, then the cost function E is represented using eqs. (8), (9) as

$$E(\Delta\mathbf{m}) = \sum_{k=0}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{g}^T_{l,k}\mathbf{J}^T_{l,k}\Delta\mathbf{m} + e_{lr,k}\right]^2 + \sum_{k=1}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{g}^T_{r,k}\mathbf{J}^T_{r,k}\mathbf{T}^T\Delta\mathbf{m} + e_{rr,k}\right]^2. \quad (10)$$

To minimize E, eq. (10) is differentiated partially and we obtain the following.

$$\frac{\partial E}{\partial \Delta\mathbf{m}} = \sum_{k=0}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{J}_{l,k}\mathbf{g}_{l,k}\mathbf{g}^T_{l,k}\mathbf{J}^T_{l,k}\Delta\mathbf{m} + e_{lr,k}\mathbf{J}_{l,k}\mathbf{g}_{l,k}\right] + \sum_{k=1}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{T}\mathbf{J}_{r,k}\mathbf{g}_{r,k}\mathbf{g}^T_{r,k}\mathbf{J}^T_{r,k}\mathbf{T}^T\Delta\mathbf{m} + e_{rr,k}\mathbf{T}\mathbf{J}_{r,k}\mathbf{g}_{r,k}\right] = 0 \quad (11)$$

[1] $\tilde{\mathbf{x}}$ is a homogenous coordinate of $\mathbf{x}$

[2] $\tilde{\mathbf{I}}_{l,k}(\mathbf{x})$ yields the image made by sub-sampling $\mathbf{I}_{l,k}$ at the position of $(\mathbf{x} + \mathbf{u}_k)$.

The upper equation is put into simple symbols $\mathbf{A}_o$, $\mathbf{b}$, and $\Delta\mathbf{m}$. Therefore, the equation is written as

$$\mathbf{A}_o \Delta\mathbf{m} = -\mathbf{b}, \tag{12}$$

where

$$\begin{aligned} \mathbf{A}_o &= \sum_{k=0}^{n}\sum_{\mathbf{x}\in W}\mathbf{J}_{l,k}\mathbf{g}_{l,k}\mathbf{g}_{l,k}^T\mathbf{J}_{l,k}^T \\ &\quad + \sum_{k=1}^{n}\sum_{\mathbf{x}\in W}\mathbf{T}\mathbf{J}_{r,k}\mathbf{g}_{r,k}\mathbf{g}_{r,k}^T\mathbf{J}_{r,k}^T\mathbf{T}^T \\ \mathbf{b} &= \sum_{k=0}^{n}\sum_{\mathbf{x}\in W} e_{lr,k}\mathbf{J}_{l,k}\mathbf{g}_{l,k} \\ &\quad + \sum_{k=1}^{n}\sum_{\mathbf{x}\in W} e_{rr,k}\mathbf{T}\mathbf{J}_{r,k}\mathbf{g}_{r,k}. \end{aligned} \tag{13}$$

In addition, $\mathbf{m}$ is given by iteratively solving eq. (12) for minimizing cost function E. In doing this iteration, updating rules of $\mathbf{m}$ and $\Delta\mathbf{m}$ are

$$\mathbf{m} \leftarrow \mathbf{m} + \Delta\mathbf{m}. \tag{14}$$

In that process, correspondence of all stereo images and sequential images are done at once. Therefore, structure and motion are obtained simultaneously.

3.2.2 Solution using multiple points

The previous section describes the solution using one observation point. In this section, we will explain the multiple points' case. If depth, motion, and time offset are estimated independently with each point using the previous approach, it is not good because the time offset is unique within the image sequence. For that reason, we estimate the cost function E to optimize depth and motion with each point and unique time offset within the image sequence.

The cost function is represented by summing of each point i($=[0,q]$)

$$E(\Delta\mathbf{M}) = \sum_{i=0}^{q}\left\{\sum_{k=0}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{g}_{l,k,i}^T\mathbf{J}_{l,k,i}^T\Delta\mathbf{m}_i + e_{lr,k,i}\right]^2 + \sum_{k=1}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{g}_{r,k,i}^T\mathbf{J}_{r,k,i}^T\mathbf{T}^T\Delta\mathbf{m}_i + e_{rr,k,i}\right]^2\right\}, \tag{15}$$

where

$$\Delta\mathbf{M} = \left[\Delta\mathbf{T}_{m,0}, \Delta Z_0, \cdots, \Delta\mathbf{T}_{m,q}, \Delta Z_q, \lambda\right]^T. \tag{16}$$

Next, $\Delta\mathbf{m}_i = [\Delta T_{mx,i}, \Delta T_{my,i}, \Delta T_{mz,i}, \Delta Z_i, \Delta\lambda]^T$ is written using $\Delta\mathbf{M}$.

$$\Delta\mathbf{m}_i = \mathbf{K}_i^T\Delta\mathbf{M} \tag{17}$$

$$\mathbf{K}_i^T = \begin{bmatrix} & & & 4i & 4i+1 & 4i+2 & 4i+3 & & & \\ 0 & \cdots & 0 & 1 & 0 & & \cdots & & & 0 \\ 0 & \cdots & & 0 & 1 & 0 & & \cdots & & 0 \\ 0 & & \cdots & & 0 & 1 & 0 & & \cdots & 0 \\ 0 & & & \cdots & & 0 & 1 & 0 & \cdots & 0 \\ 0 & & & & \cdots & & & & 0 & 1 \end{bmatrix}$$

As these equations are inserted into eq. (15), the cost function E is written as

$$E(\Delta\mathbf{M}) = \sum_{i=0}^{q}\left\{\sum_{k=0}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{g}_{l,k,i}^T\mathbf{J}_{l,k,i}^T\mathbf{K}_i^T\Delta\mathbf{M} + e_{lr,k,i}\right]^2 + \sum_{k=1}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{g}_{r,k,i}^T\mathbf{J}_{r,k,i}^T\mathbf{T}^T\mathbf{K}_i^T\Delta\mathbf{M} + e_{rr,k,i}\right]^2\right\}. \tag{18}$$

To minimize the cost function, the upper equation is partially differentiated by $\Delta\mathbf{M}$. Accordingly,

$$\begin{aligned} \frac{\partial E}{\partial\Delta\mathbf{M}} &= \sum_{i=0}^{q}\mathbf{K}_i\left\{\sum_{k=0}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{J}_{l,k,i}\mathbf{g}_{l,k,i}\mathbf{g}_{l,k,i}^T\mathbf{J}_{l,k,i}^T\right]\right. \\ &\quad \left. + \sum_{k=1}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{T}\mathbf{J}_{r,k,i}\mathbf{g}_{r,k,i}\mathbf{g}_{r,k,i}^T\mathbf{J}_{r,k,i}^T\mathbf{T}^T\right]\right\}\mathbf{K}_i^T\Delta\mathbf{M} \\ &\quad + \sum_{i=0}^{q}\mathbf{K}_i\left\{\sum_{k=0}^{n}\sum_{\mathbf{x}\in W}\left[e_{lr,k}\mathbf{J}_{l,k,i}\mathbf{g}_{l,k,i}\right]\right. \\ &\quad \left. + \sum_{k=1}^{n}\sum_{\mathbf{x}\in W}\left[e_{rr,k,i}\mathbf{T}\mathbf{J}_{r,k,i}\mathbf{g}_{r,k,i}\right]\right\} = 0. \end{aligned} \tag{19}$$

The upper equation is put into simple symbols of $\mathbf{A}_m$ and $\mathbf{b}$. Consequently, the equation is written as

$$\mathbf{A}_m\Delta\mathbf{M} = -\mathbf{b}, \tag{20}$$

where

$$\begin{aligned} \mathbf{A}_m &= \sum_{i=0}^{q}\mathbf{K}_i\left\{\sum_{k=0}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{J}_{l,k,i}\mathbf{g}_{l,k}\mathbf{g}_{l,k,i}^T\mathbf{J}_{l,k,i}^T\right]\right. \\ &\quad \left. + \sum_{k=1}^{n}\sum_{\mathbf{x}\in W}\left[\mathbf{T}\mathbf{J}_{r,k,i}\mathbf{g}_{r,k,i}\mathbf{g}_{r,k,i}^T\mathbf{J}_{r,k,i}^T\mathbf{T}^T\right]\right\}\mathbf{K}_i^T \\ \mathbf{b} &= \sum_{i=0}^{q}\mathbf{K}_i\left\{\sum_{k=0}^{n}\sum_{\mathbf{x}\in W}\left[e_{lr,k,i}\mathbf{J}_{l,k,i}\mathbf{g}_{l,k,i}\right]\right. \\ &\quad \left. + \sum_{k=1}^{n}\sum_{\mathbf{x}\in W}\left[e_{rr,k,i}\mathbf{T}\mathbf{J}_{r,k,i}\mathbf{g}_{r,k,i}\right]\right\} \end{aligned} \tag{21}$$

Optimized $\mathbf{M}$ is obtained by iteratively solving eq. (21) for minimizing cost function E. In doing this iteration, updating rules of $\mathbf{M}$ and $\Delta\mathbf{M}$ are

$$\mathbf{M} \leftarrow \mathbf{M} + \Delta\mathbf{M}. \tag{22}$$

In that process, correspondence of all stereo images and sequential images about multiple points are done at once. Therefore, the structure and motion of these points as well as unique time offset are estimated simultaneously.

In the case of synchronized cameras (i.e. time offset $\lambda = 0$) and unsynchronized cameras with known time offset λ, the structure and motion are obtained simultaneously using our framework. For that case, the derivations of eq.(8) and (9) are identical except the λ parameter. Thereby, Jacobian matrices are given by a partial differential with depth and 3D motion, and $\mathbf{T}$ becomes $\mathbf{I}$.

3.3. Implementation

As we described above, the structure, its motion, and time offset can be estimated. We summarize these estimation steps of multiple points' case (Fig. 3). First, select points [3] from the key frame. Next, estimate the unknown

[3] Shi et al.'s method[8] is employed to detect feature points.

for (iter=0 to iter max or convergence condition)
  if iter is 0
    Set initial value to $\mathbf{M}$.
  end if.
  for (k=0 to max image)
    for (i=0 to max point)
      Warp image $\mathbf{I}_{r,k,i}$ and get $\tilde{\mathbf{I}}_{r,k,i}$ using $\mathbf{m}_i$ $(k \neq 0)$.
      Warp image $\mathbf{I}_{l,k,i}$ and get $\tilde{\mathbf{I}}_{l,k,i}$ using $\mathbf{m}_i$.
      Calculate $\mathbf{g}_{r,k,i}$, $\mathbf{J}_{r,k,i}$, $e_{rr,k,i}$ in eq.(39) $(k \neq 0)$.
      Calculate $\mathbf{g}_{l,k,i}$, $\mathbf{J}_{l,k,i}$, $e_{lr,k,i}$ in eq.(33)
    end for loop.
  end for loop.
  Calculate $\mathbf{A}_m$ and $\mathbf{b}$ in eq.(21).
  Calculate $\Delta\mathbf{M}$ using eq.(20).
  Update $\mathbf{M} \leftarrow \mathbf{M} + \Delta\mathbf{M}$ in eq.(22).
end for loop.
3D MOTION, DEPTH for each point and TIME OFFSET parameters are given.

Figure 3. Optimization step of cost function in case of multiple points.

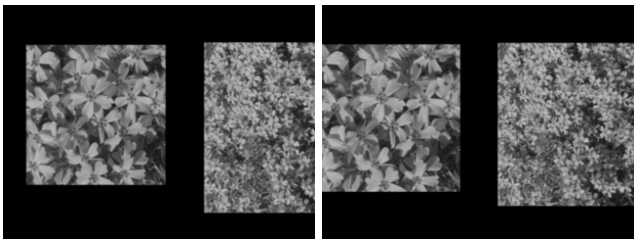

Figure 4. Unsynchronized stereo image pair.

vector $\mathbf{M}$ which contains depth, motion, and time offset according to Fig. 3.

We use a Gaussian pyramid for reducing the iteration number and are more robust about initial value. In this paper, we assume that stereo cameras are set up parallel. We also use the same intrinsic camera parameters for easy expression of formulation. Therefore, if the setting of stereo cameras is not parallel or intrinsic camera parameters are not the same, image rectification[1] is applied *apriori*. Consequently, the stereo cameras are virtually parallel and have the same intrinsic camera parameters.

4. Experiments

4.1. Synthetic images

Figure 4 shows input unsynchronized stereo images. A stereo camera is 0.77 [m] baseline and parallel layout. Therefore, the epipolar line of this stereo image is horizontal, but it is apparent that the same point between stereo images does not exist on the epipolar line because of the time offset. The layout and objects' motion are shown in Fig. 5. The two objects are flat and have different 3D motions and depths. The left object A is 10.6 [m] depth and the right object B is 14.4 [m].

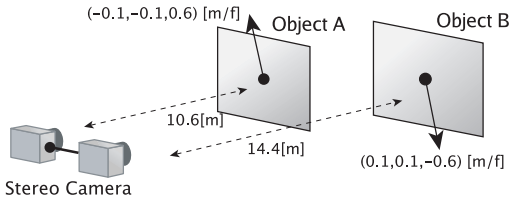

Figure 5. Layout and objects' motion.

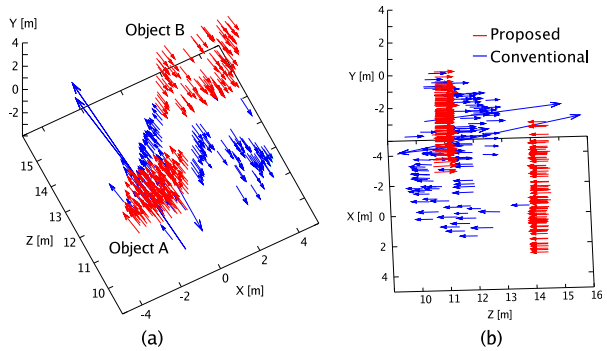

Figure 6. Estimated 3D positions and motions.

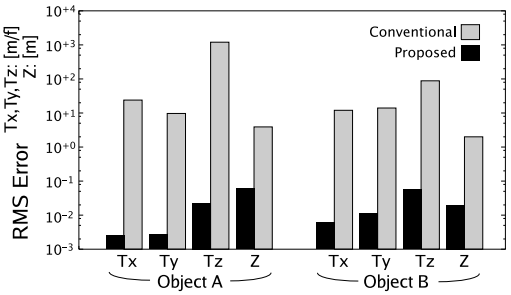

Figure 7. RMS Errors of depth and 3D motions.

Two pairs of unsynchronized sequential stereo images that are different by one frame are used in this experiment. Initial values of depth, 3D motion, and time offset are set uniformly as $Z = 12$ [m], $\mathbf{T}_m = [1e-5, 1e-2, 1e-1]^T$, and $\lambda = 0$, respectively.

The estimated time offset is -1.006 [frame lag], of which the true value is -1.0. Demonstrably, it accurately estimated the true value. We next compare the proposed method and the conventional method. The conventional method is that a point in $\mathbf{I}_{r,0}$ corresponds to $\mathbf{I}_{l,0}$ (unsynchronized stereo pair). Then this point is tracked to $\mathbf{I}_{r,1}$. Finally, the tracked point in $\mathbf{I}_{r,1}$ corresponds to $\mathbf{I}_{l,1}$. Therefore, an identical point has a 3D position at each time. The 3D mo-

tion is given as the difference between these estimated 3D positions.

Figure 6 shows the estimated 3D position and its 3D motion with an arrow. The direction and length of the arrows respectively represent the motion direction and its magnitude. Red and blue arrows respectively denote the proposed method's result and the conventional method's result.

Figure 7 shows the RMS error of depth and each axis of 3D motion, of which the vertical axis is a log scale. The RMS errors of the conventional method are larger than those of the proposed method in all results. The conventional method is difficult to correspond between the asynchronous stereo images. Even if correspondence were shown accurately, the estimated 3D position could be wrong because the locations of objects differ among stereo images.

4.2. Real images

The actual experiment described here used a Dragonfly camera (Point Grey Research Inc.) and a typical retail-purchased video camera (VX-2000; Sony Corp.). These cameras are unsynchronized. We cannot know the real time offset between these cameras. Figure 8 shows the input images; two pairs of unsynchronized images are used. A running model car [A] and a teddy bear [C] are shown. The stereo rig is fixed in tripod stand. Therefore, the carved ornament [B] and background are stationary.

The respective initial values of depth, motion, and time offset are 0.7 [m], 0.001 [m/f], and 0.0. The estimated time offset is 0.719 [frame lag]. Figure 9 shows estimated structures with textures observed from some viewpoints. The 3D positions of these objects are reconstructed well.

Figure 10 shows the estimated 3D position and its 3D motion using an arrow with direction and length. The lengths of the arrows are expanded two times for ease of view. The viewpoint of this figure is nearly the same as that in the middle of Fig. 9. The 3D motion of [A] is uniform in both direction and length, and is different from the motion of [B] and [C]. The object [B] has only a $+$ mark with no arrow, i.e. no motion for stationary status. By these results, it is apparent that the structure and motion are appropriately and accurately estimated.

5. Conclusions

We proposed the method of estimating structure, motion, and time offset using unsynchronized stereo cameras. The contribution of our method consists of two parts: The first is the method of corresponding all stereo images and sequential images simultaneously. Using that method, we can get structure and motion at the same time. Our second contribution is the method of estimating structure, motion, and time offset simultaneously using unsynchronized stereo

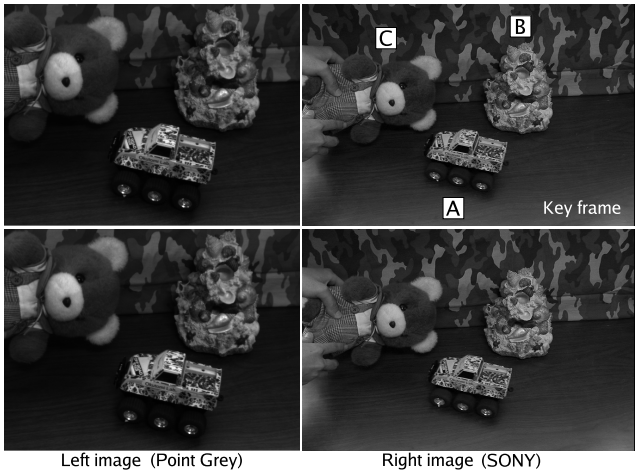

Figure 8. Unsynchronized stereo images taken by cameras of different types.

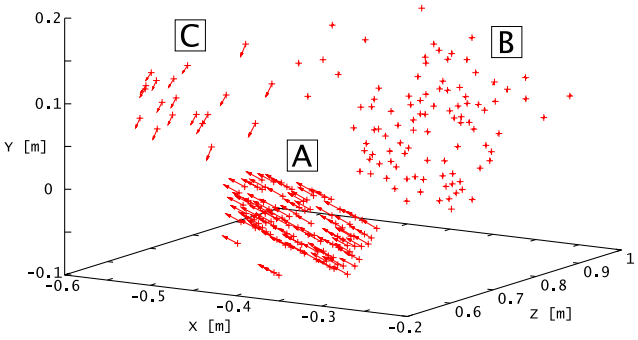

Figure 10. Estimated 3D positions and motions.

cameras. This is done by taking advantage of the first contribution scheme. Finally, we present the experimental results to show the effectiveness of our method.

Future work is intended to reduce the calculation time and to handle spatial and temporal occlusions.

References

[1] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.

[2] K. Hanna and N. Okamoto. Combining and motion analysis for direct estimation of scene structure. In *Proc.IEEE*, pages 357–365, 1993.

[3] H. Hao, H. Sawhney, and R. Kumar. Dynamic depth recovery from multiple synchronized video streams. In *Proc. IEEE CVPR*, volume 1, pages 118–124, 2001.

[4] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *Proc.IEEE CVPR*, pages 454–460, June 1994.

Figure 9. Estimated structure with texture mapping.

[5] R. Mandelbaum, G. Salgian, and H. Sawhney. Correlation-based estimation of ego-motion and structure from motion and stereo. In *Proc. ICCV*, pages 544–550, 1998.

[6] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, 1993.

[7] S. Park and I. Kweon. Robust and direct estimation of 3-d motion and scene depth from stereo image sequences. *Pattern Recognition*, 34(9):1713–1728, 2001.

[8] J. Shi and C. Tomasi. Good features to track. In *Proc. IEEE CVPR*, pages 593–600, June 1994.

[9] S. Shimizu, H. Fujiyoshi, Y. Nagasaka, and T. Takahashi. A pseudo stereo vision method for unsynchronized cameras. In *Proc.ACCV*, volume Vol.1, pages 575 – 580, 2004.

[10] H. Shum and R. Szeliski. Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, 2000.

[11] G. Stein and A. Shashua. Direct estimation of motion and extended scene structure from a moving stereo rig. In *Proc.IEEE CVPR*, pages 211–218, 1998.

[12] M. Svedman, L. Goncalves, N. Karlsson, M. Munich, and P. Pirjanian. Structure from stereo vision using unsynchronized cameras for simultaneous localization mapping. In *Proc. IROS*, pages 2 – 6, 2005.

[13] C. Zhou and H. Tao. Dynamic depth recovery from unsynchronized video streams. In *Proc.IEEE CVPR*, volume 2, pages 351–358, 2003.

A. Derivation of Jacobian Matrix for a Different-Camera Pair

Consider that the changed portions of $\mathbf{u}_k$ in eq.(7) are

$$\mathbf{u}_k^{\Delta} = \frac{\mathbf{A}_x\{(k+\lambda+\Delta\lambda)(\mathbf{T}_m+\Delta\mathbf{T})+\mathbf{T}_s\}}{Z+\Delta Z-\{(k+\lambda+\Delta\lambda)T_{mz}+\Delta T_z\}}. \quad (23)$$

The denominator of the upper equation is described as

$$\frac{1}{Z+\Delta Z-\{(k+\lambda+\Delta\lambda)T_{mz}+\Delta T_z\}} = \frac{1}{\dot{Z}_k}+\xi, \quad (24)$$

where

$$\begin{array}{rcl} \dot{Z}_k & = & Z-(k+\lambda)T_{mz} \\ \xi & = & \frac{-\{\Delta Z-(k+\lambda)\Delta T_z\ -\Delta\lambda T_{mz}\}}{\dot{Z}_k^2+\dot{Z}_k\{\Delta Z-(k+\lambda)\Delta T_z-\Delta\lambda T_{mz}\}}. \end{array} \quad (25)$$

Then, if the changed portions of $\mathbf{u}_k$ are assumed to be small, then

$$Z-(k+\lambda)T_{mz} >> \Delta Z-(k+\lambda)\Delta T_z-\Delta\lambda T_{mz}.$$

We obtain ξ as

$$\xi \approx \xi' = \frac{-\Delta Z+(k+\lambda)\Delta T_z+\Delta\lambda T_{mz}}{\dot{Z}_k^2}. \quad (26)$$

Equation (23) is described using ξ' as the following.

$$\mathbf{u}_k^{\Delta} = \left(\frac{1}{\dot{Z}_k}+\xi'\right)\mathbf{A}_x\{(k+\lambda+\Delta\lambda)(\mathbf{T}_m+\Delta\mathbf{T})+\mathbf{T}_s\} \quad (27)$$

Equation (27) is altered according to two considerations: one is disregard of the sections that contain Δ squared because of smallness; the other is $\mathbf{T}_s = [T_{sx},0,0]^T$ for a parallel setup.

$$\begin{array}{rcl} \mathbf{u}_k^{\Delta} & \approx & \left(\frac{1}{Z_k}+\frac{-\Delta Z+(k+\lambda)\Delta T_z+\Delta\lambda T_{mz}}{\dot{Z}_k^2}\right)\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \\ & & +\frac{1}{Z_k}\begin{bmatrix} -(k+\lambda)f\Delta T_x+(k+\lambda)\hat{x}_0\Delta T_z \\ \qquad +(-fT_{mx}+\hat{x}_0T_{mz})\Delta\lambda \\ -(k+\lambda)af\Delta T_y+(k+\lambda)\hat{y}_0\Delta T_z \\ \qquad +(-afT_{my}+\hat{y}_0T_{mz})\Delta\lambda \end{bmatrix} \end{array} \quad (28)$$

Therein,

$$\begin{array}{rcl} \alpha & = & (k+\lambda)(-fT_{mx}+\hat{x}_0T_{mz})-fT_{sx} \\ \beta & = & (k+\lambda)(-afT_{my}+\hat{y}_0T_{mz}). \end{array} \quad (29)$$

Using a homogeneous coordinate, the upper equation is described as:

$$\mathbf{x}+\mathbf{u}_k^{\Delta} = \begin{bmatrix}\gamma & 0 & \frac{\alpha'}{\dot{Z}_k}\\ 0 & \gamma & \frac{\beta'}{\dot{Z}_k}\\ 0 & 0 & 1\end{bmatrix}\underbrace{\begin{bmatrix}\frac{\kappa}{\gamma} & 0 & \frac{\zeta}{\gamma}\\ 0 & \frac{\kappa}{\gamma} & \frac{\eta}{\gamma}\\ 0 & 0 & 1\end{bmatrix}}_{\mathbf{I}+\mathbf{D}}\tilde{\mathbf{x}}, \quad (30)$$

where

$$\begin{array}{rcl}\tilde{\mathbf{x}} &=& [x,y,1]^T \\ \alpha' &=& -(k+\lambda)(fT_{mx}+u_0T_{mz})-fT_{sx} \\ \beta' &=& -(k+\lambda)(afT_{my}+v_0T_{mz}) \\ \gamma &=& 1+\frac{(k+\lambda)T_{mz}}{\dot{Z}_k} \\ \kappa &=& 1+\frac{1}{\dot{Z}_k}[(k+\lambda)(T_{mz}+\Delta T_z)+T_{mz}\Delta\lambda] \\ && +\xi'(k+\lambda)T_{mz} \\ \zeta &=& -\frac{1}{\dot{Z}_k}[(k+\lambda)(f\Delta T_x+u_0\Delta T_z) \\ && +(fT_{mx}+u_0T_{mz})\Delta\lambda]+\xi'\alpha' \\ \eta &=& -\frac{1}{\dot{Z}_k}[(k+\lambda)(af\Delta T_y+v_0\Delta T_z) \\ && +(afT_{my}+v_0T_{mz})\Delta\lambda]+\xi'\beta'.\end{array} \tag{31}$$

Next, we consider $\tilde{\mathbf{x}}'' \sim (\mathbf{I}+\mathbf{D})\tilde{\mathbf{x}}$, in which $\mathbf{I}$ is the identity matrix and $\mathbf{D}$ is a 3×3 matrix with small values. First-order Taylor expansion can be applied to $\tilde{\mathbf{I}}_{l,k}(\mathbf{x}'')$ [4] around $\mathbf{x}$, we get

$$\tilde{\mathbf{I}}_{l,k}(\mathbf{x}'')-\mathbf{I}_{r,0}(\mathbf{x}) \approx \mathbf{g}_{l,k}^T\mathbf{J}_{l,k}^T\Delta\mathbf{m}+e_{lr,k}, \tag{32}$$

where

$$\begin{array}{rcl}\mathbf{g}_{l,k}^T &=& \nabla\tilde{\mathbf{I}}_{l,k}(\mathbf{x}) \\ \mathbf{J}_{l,k}^T &=& \frac{\partial\mathbf{x}''}{\partial\Delta\mathbf{m}} \\ \Delta\mathbf{m} &=& [\Delta T_{mx},\Delta T_{my},\Delta T_{mz},\Delta Z,\Delta\lambda]^T \\ e_{lr,k} &=& \tilde{\mathbf{I}}_{l,k}(\mathbf{x})-\mathbf{I}_{r,0}(\mathbf{x})\end{array}. \tag{33}$$

The context of the Jacobian matrix is written as

$$\frac{\partial\mathbf{x}''}{\partial\Delta\mathbf{m}}=\frac{\partial\mathbf{d}}{\partial\Delta\mathbf{m}}\frac{\partial\mathbf{x}''}{\partial\mathbf{d}}. \tag{34}$$

$\mathbf{d}$ is the vector which aligns $\mathbf{D}$ line by line. The last half of the right side in eq. (34) is represented [10] as

$$\frac{\partial\mathbf{x}''}{\partial\mathbf{d}}=\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x^2 & -xy & -x \\ 0 & 0 & 0 & x & y & 1 & -xy & -y^2 & -y\end{bmatrix}^T. \tag{35}$$

In addition, $\frac{\partial\mathbf{d}}{\partial\Delta\mathbf{m}}$ is given by partially differentiating $\mathbf{D}$ in eq. (30) with $\Delta\mathbf{m}$. Therefore, the Jacobian matrix is given by substituting eq. (35) and $\frac{\partial\mathbf{d}}{\partial\Delta\mathbf{m}}$ to eq. (34). Consequently, we get

$$\mathbf{J}_{l,k}^T=\frac{\partial\mathbf{x}''}{\partial\Delta\mathbf{m}}=\frac{1}{\gamma}\begin{bmatrix} j_{11} & 0 \\ 0 & j_{22} \\ \mu & \nu \\ j_{41} & j_{42} \\ j_{51} & j_{52}\end{bmatrix}. \tag{36}$$

In those equations,

$$\begin{array}{l} j_{11}=-\frac{(k+\lambda)f}{\dot{Z}_k}, \quad j_{22}=-\frac{(k+\lambda)af}{\dot{Z}_k} \\ j_{41}=-\frac{\alpha'+(k+\lambda)xT_{mz}}{\dot{Z}_k^2}, \quad j_{42}=-\frac{\beta'+(k+\lambda)yT_{mz}}{\dot{Z}_k^2} \\ j_{51}=-\frac{fT_{mx}-(x-u_0)T_{mz}}{\dot{Z}_k}+\frac{\alpha' T_{mz}+(k+\lambda)xT_{mz}^2}{\dot{Z}_k^2} \\ j_{52}=-\frac{afT_{my}-(y-v_0)T_{mz}}{\dot{Z}_k}+\frac{\beta' T_{mz}+(k+\lambda)yT_{mz}^2}{\dot{Z}_k^2} \\ \mu=\frac{1}{\dot{Z}_k}(k+\lambda)(x-u_0)+\frac{1}{\dot{Z}_k^2}(k+\lambda)\{\alpha'+(k+\lambda)T_{mz}x\} \\ \nu=\frac{1}{\dot{Z}_k}(k+\lambda)(y-v_0)+\frac{1}{\dot{Z}_k^2}(k+\lambda)\{\beta'+(k+\lambda)T_{mz}y\}.\end{array}$$

[4] $\tilde{\mathbf{I}}_{l,k}(\mathbf{x})$ to represent the image that is generated by subsampling $\mathbf{I}_{l,k}$ at the position of $(\mathbf{x}+\mathbf{u}_k)$.

B. Derivation of Jacobian Matrix for an Identical-Camera Pair

The $\mathbf{u}_k'$ in eq.(7) is resolved in an identical manner as $\mathbf{u}_k$ in the Appendix A. Finally, the Jacobian is given as

$$\mathbf{J}_{r,k}^T=\frac{\partial\mathbf{x}''}{\partial\Delta\mathbf{m}'}=\frac{1}{\gamma''}\begin{bmatrix} -\frac{kf}{\ddot{Z}_k} & 0 \\ 0 & -\frac{kaf}{\ddot{Z}_k} \\ \mu'' & \nu'' \\ -\frac{\alpha''+kxT_{mz}}{\ddot{Z}_k^2} & -\frac{\beta''+kyT_{mz}}{\ddot{Z}_k^2}\end{bmatrix}, \tag{37}$$

where

$$\begin{array}{rcl}\Delta\mathbf{m}' &=& [\Delta T_{mx},\Delta T_{my},\Delta T_{mz},\Delta Z]^T \\ \ddot{Z}_k &=& Z-kT_{mz} \\ \alpha'' &=& -k(fT_{mx}+u_0T_{mz}) \\ \beta'' &=& -k(afT_{my}+v_0T_{mz}) \\ \gamma'' &=& 1+\frac{kT_{mz}}{\ddot{Z}_k} \\ \mu'' &=& \frac{1}{\ddot{Z}_k}k(x-u_0)+\frac{1}{\ddot{Z}_k^2}k(\alpha''+kT_{mz}x) \\ \nu'' &=& \frac{1}{\ddot{Z}_k}k(y-v_0)+\frac{1}{\ddot{Z}_k^2}k(\beta''+kT_{mz}y).\end{array}$$

In a similar manner, the difference of the image value is

$$\tilde{\mathbf{I}}_{r,k}(\mathbf{x}'')-\mathbf{I}_{r,0}(\mathbf{x}) \approx \mathbf{g}_{r,k}^T\mathbf{J}_{r,k}^T\Delta\mathbf{m}'+e_{rr,k}, \tag{38}$$

where

$$\begin{array}{rcl}\mathbf{g}_{r,k}^T &=& \nabla\tilde{\mathbf{I}}_{r,k}(\mathbf{x}) \\ \mathbf{J}_{r,k}^T &=& \frac{\partial\mathbf{x}''}{\partial\Delta\mathbf{m}'} \\ e_{rr,k} &=& \tilde{\mathbf{I}}_{r,k}(\mathbf{x})-\mathbf{I}_{r,0}(\mathbf{x})\end{array} \tag{39}$$

$$\Delta\mathbf{m}'=\mathbf{T}^T\Delta\mathbf{m}\,, \quad \mathbf{T}^T=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0\end{bmatrix}. \tag{40}$$

Equation (38) is expressed by considering eq. (40); thus, we obtain

$$\mathbf{g}_{r,k}^T\mathbf{J}_{r,k}^T\Delta\mathbf{m}'+e_{rr,k}=\mathbf{g}_{r,k}^T\mathbf{J}_{r,k}^T\mathbf{T}^T\Delta\mathbf{m}+e_{rr,k} \tag{41}$$