A Robust Recursive Factorization Method for Recovering Structure and Motion from Live Video Frames

Takeshi Kurata†, Jun Fujiki†, Masakatsu Kourogi‡, Katsuhiko Sakaue†

†Electrotechnical Laboratory, 1–1–4 Umezono, Tsukuba, Ibaraki, 305–8568, JAPAN
‡School of Sci. & Eng., Waseda University, 3–4–1 Ohkubo, Shinjuku, Tokyo, 169–8555, JAPAN
E-mail: kurata@etl.go.jp http://www.etl.go.jp/~kurata/

Abstract

This paper describes a fast and robust approach for recovering structure and motion from video frames. It first describes a robust recursive factorization method for affine projection. Using the Least Median of Squares (LMedS) criterion, the method estimates the dominant 3D affine motion and discards feature points regarded as outliers. The computational cost of the overall procedure is reduced by combining this robust-statistics-based method with a recursive factorization method that can at each frame provide the updated 3D structure of an object at a fixed computational cost by using the principal component analysis. This paper then describes experiments with synthetic data and with real image sequences, the results of which demonstrate that the method can be used to estimate the dominant structure and the motion robustly and in real-time on an off-the-shelf PC.

1 Introduction

Recovering structure and motion from image streams is one of the most important topics in computer vision and the techniques are used for various applications such as image-based rendering, augmented reality, man-machine interface, and so on. In numerous studies on the field, the factorization method [13, 8] is stable in calculation and can treat all images uniformly. This method has been extended to various aims, for example, from point features to line features [10, 6] , and for the perspective projection model [12, 15]. Recursive factorization methods have also been proposed [5, 1]. These methods can provide an updated 3D structure at each frame and at the small and fixed computational cost.

In this paper, we focus on 1) robustness against outliers, which include false matches and other objects, and 2) the computational cost at each frame in order to apply the factorization method using point correspondences under affine projection to the real environment in real-time.

First, a robust recursive factorization method for affine projection is described in Section 3. This method can recover the dominant affine motion robustly by using the Least Median of Squares (LMedS) method [11], and it can discard feature points that are regarded as outliers. Therefore a measurement matrix is stably decomposed into the motion and shape matrices. Although some previous works [17, 14] have also used robust statistics for recovering the epipolar geometry and the multiple projective view relation, they haven't described how to cope with the increasing computational cost as the number of frames increases. In this method, discarding outliers based on the LMedS criterion is combined with the recursive factorization method in [1] for the real-time processing.

Section 4 describes experiments that use synthetic data and show that the method can, on an off-the-shelf PC, estimate the dominant structure and the motion robustly and in real-time. It also describes experiments using real images. In theory the LMedS method requires dominant data which account for more than 50% of all the data. To satisfy this requirement, we assume that features of one object exist locally. First, moving regions are roughly segmented by applying a global flow estimation in [3]. Next, features are detected and tracked in and around the moving regions, and then the 3D motion is estimated at each frame.

2 A Review of the Factorization Method for Affine Projection

For simplification, this paper assumes that an optical axis is orthogonal to an image plane, their intersection is known, the aspect ratio of a pixel is 1:1, and skew is absent.*

Using affine projection [7], the p-th 3D point $\boldsymbol{s}_p$ in the world coordinates system is related to the projection $\boldsymbol{x}_{fp}(=(x_{fp}, y_{fp}))$ on the f-th image in non-homogeneous coordinates by

$$\boldsymbol{x}_{fp} = \underset{(2\times3)}{A_f} \quad \underset{(3\times3)}{C_f} \quad \boldsymbol{s}_p + \boldsymbol{f}_f,$$

*Methods for self-calibration of an affine camera are described in [9] etc.

where A_f is the affine projection matrix, $C_f(=(\boldsymbol{i}_f, \boldsymbol{j}_f, \boldsymbol{k}_f)^{\mathrm{T}})$ is a rotation matrix,[†] and $\boldsymbol{f}_f$ is a factor related to translation.

By affine projection, $\boldsymbol{g}$, which is the centroid of all $\boldsymbol{s}_p$ $(p = 1, \cdots, P)$, is projected into $\boldsymbol{x}_{fc}(=(x_{fc}, y_{fc}))$, which is the centroid of all $\boldsymbol{x}_{fp}$ on the f-th image. Therefore, $\boldsymbol{s'}_p$ and $\boldsymbol{x'}_{fp}$, which are respectively the relative coordinates of $\boldsymbol{s}_p$ from $\boldsymbol{g}$ and the relative coordinates of $\boldsymbol{x}_{fp}$ from $\boldsymbol{x}_{fc}$, are related by

$$\boldsymbol{x'}_{fp} = A_f C_f \boldsymbol{s'}_p.$$

For all points and for the f-th image, these equations can be rewritten as

$$\underset{(2\times P)}{W'_f} = (\boldsymbol{x'}_{f1}, \cdots, \boldsymbol{x'}_{fP}) = \underset{(2\times 3)}{M_f} \; \underset{(3\times P)}{S'}$$

where $M_f = A_f C_f$ and $S' = (\boldsymbol{s'}_1, \cdots, \boldsymbol{s'}_P)$. Furthermore, for all frames $(f = 1, \cdots, F)$, these equations can be rewritten as

$$\underset{(2F\times P)}{W'} = \underset{(2F\times 3)}{M} \; \underset{(3\times P)}{S'}$$

where $W' = ({W'_1}^{\mathrm{T}}, \cdots, {W'_F}^{\mathrm{T}})^{\mathrm{T}}$ and $M = ({M_1}^{\mathrm{T}}, \cdots, {M_F}^{\mathrm{T}})^{\mathrm{T}}$. This equation shows that the registered measurement matrix W' can be factorized into the product of the motion matrix M and the shape matrix S'.

The factorization method realizes the decomposition by the following two-step procedure. In the first step, W' is decomposed as follows by singular value decomposition (SVD) and by the constraint that W' is rank three.

$$\underset{(2F\times P)}{W'} = \underset{(2F\times 3)}{\hat{M}} \; \underset{(3\times P)}{\hat{S}'}$$

where " $\hat{}$ " indicates that it is computed only up to an affine transformation. Therefore arbitrary 3×3 non-singular matrix D can be inserted such as $W' = \hat{M} D D^{-1} \hat{S}'$.

Thus, in the second step, D is determined by using metric constraints.[‡] A constraint that can be used to recover the Euclidean motion and shape is $C_f {C_f}^{\mathrm{T}} = \mathrm{I}_3$. Therefore the following constraints are obtained with regard to each affine projection model:

$$\hat{M}_f Q {\hat{M}_f}^{\mathrm{T}} = A_f {A_f}^{\mathrm{T}},$$

where $Q = DD^{\mathrm{T}}$. A_f for the scaled orthographic and paraperspective projection models are given by the following:

$$\text{Scaled Orthographic:} \quad A_f = \frac{l}{z_f}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\text{Paraperspective:} \quad A_f = \frac{1}{z_f}\begin{pmatrix} l & 0 & -x_{f\cdot} \\ 0 & l & -y_{f\cdot} \end{pmatrix}$$

Here z_f is the average depth of points and l is a focal length (More details about the metric constraints are given in [4]).

3 A Robust Recursive Factorization Method

3.1 Recovering the Affine Motion Using the LMedS Criterion

The estimate using the LMedS criterion is usually obtained by computing trial estimate repeatedly from randomly sampled data [11]. Since at least four non-coplanar points are required to determine the affine motion and shape, we randomly sample four points from all points and put them into a $2F \times 4$ measurement matrix $W^{[j]}$ for the j-th trial $(j = 1, \cdots, J)$. To obtain the trial affine motion, we decompose the registered measurement matrix $W'^{[j]}$ by SVD and by the constraint that $W'^{[j]}$ is rank three:

$$\underset{(2F\times 4)}{W'^{[j]}} = W^{[j]} - \begin{pmatrix} \boldsymbol{x}_{1c}^{[j]} \\ \vdots \\ \boldsymbol{x}_{Fc}^{[j]} \end{pmatrix} (1, \cdots, 1) = \underset{(2F\times 3)}{U^{[j]}} \; \underset{(3\times 3)}{\Sigma^{[j]}} \; \underset{(3\times 4)}{V^{[j]\mathrm{T}}}, \tag{1}$$

[†] $\{i_f, j_f\}$ is regarded as the normal orthogonal basis of the f-th image and k_f is regarded as unit vector along the optical axis.

[‡] It is well known that D has a reflection.

where $\boldsymbol{x}_{fc}^{[j]}$ is the centroid of those four points on the f-th image. Then we can regard $U^{[j]}$ as the affine motion for j-th trial and also can regard $U^{[j]\mathrm{T}}\boldsymbol{W}'_p$ as the 3D affine coordinates of the p-th point for the j-th trial. Here $\boldsymbol{W}'_p = (\boldsymbol{x}_{1p}{}^{\mathrm{T}}, \cdots, \boldsymbol{x}_{Fp}{}^{\mathrm{T}})^{\mathrm{T}} - (\boldsymbol{x}_{1c}^{[j]\mathrm{T}}, \cdots, \boldsymbol{x}_{Fc}^{[j]\,\mathrm{T}})^{\mathrm{T}}$.

The squared residual of each point is given by

$$\begin{aligned} r_p^{[j]^2} &= \left\| \boldsymbol{W}'_p - \underset{(2F\times 3)}{U^{[j]}} \left(\underset{(3\times 2F)}{U^{[j]\mathrm{T}}} \boldsymbol{W}'_p \right) \right\|^2 \\ &= \left\| \left(\underset{(2F\times 2F)}{I} - \underset{(2F\times 3)}{U^{[j]}} \underset{(3\times 2F)}{U^{[j]\mathrm{T}}} \right) \boldsymbol{W}'_p \right\|^2 . \end{aligned} \tag{2}$$

Computing $\mu^{[j]}$, which is the median of $r_p^{[j]^2}$ for each trial, we finally obtain the affine motion $U^{[j]}$ for which $\mu^{[j]}$ is minimal.

We can determine J, which is the required number of trials, as follows [11]. The probability R that at least one of J trials is good is given by $R = 1 - \{1 - (1-\epsilon)^4\}^J$ where ϵ is the ratio of outliers. Therefore, for example, when $R \geq 0.999$ and $\epsilon = 0.4$, J must be 50 or more.

It is computationally inefficient, however, to simply select four points because $W'^{[j]}$ must be rank three. Thus we compute $\mu^{[j]}$ only if $W'^{[j]}$ has the third singular value more than a threshold.

Since, as mentioned in [11, 17], the efficiency of the LMedS method is poor against Gaussian noise, the dominant affine motion is re-estimated using all inliers. The robust standard deviation $\hat{\sigma}$ can be estimated as

$$\hat{\sigma} = 1.4826\{1.0 + 5.0/(P-4)\}\sqrt{\mu^{[j]}}.$$

All feature points which satisfy $r_p^{[j]^2} \leq (2.5\hat{\sigma})^2$ are selected as inliers with the dominant 3D motion and the affine motion and the affine shape are obtained by the least squares method; that is, by SVD using all inliers.

3.2 Outline of the Recursive Factorization Method

The recursive factorization method in [1] compresses the motion matrix, the metric constraints, and the measurement matrix by using the principal component analysis (PCA) to reduce and to fix the computational cost at each frame. To fix the world coordinates through every image, we compute the orthogonal transformation between shape matrices of two frames instead of the one between motion matrices of two frames [1]. The outline of this recursive factorization method for each frame is described as follows.

(i) Recovering the affine shape and motion

The f-th recursive registered measurement matrix $W'_{[f]}$ is decomposed by SVD and by the constraint that $W'_{[f]}$ is rank three as follows:

$$\underset{(5\times P)}{W'_{[f]}} = \begin{pmatrix} \underset{(3\times P)}{\mathcal{W}'_{[f-1]}} \\ \underset{(2\times P)}{W'_f} \end{pmatrix} = \underset{(5\times 3)}{\hat{M}_{[f]}} \underset{(3\times P)}{\hat{S}'_{[f]}} .$$

$\mathcal{W}'_{[f-1]}$ is the $(f-1)$-th principal registered measurement matrix computed in Step (iv). Letting

$$\underset{(5\times 3)}{\hat{M}_{[f]}} = \begin{pmatrix} \underset{(3\times 3)}{\hat{\mathcal{M}}_{[f-1]}} \\ \underset{(2\times 3)}{\hat{M}_f} \end{pmatrix},$$

we can regard $\hat{\mathcal{M}}_{[f-1]}$ as the $(f-1)$-th principal affine motion matrix.

(ii) Obtaining Metric constraints

Obtain the Euclidean motion $\bar{M}_{[f]} (= \hat{M}_{[f]} D_{[f]})$ and shape $\bar{S}'_{[f]} (= D_{[f]}{}^{\mathrm{T}} \hat{S}'_{[f]})$ by using the following metric constraints. Here " $\bar{}$ " indicates that the coordinates is not yet fixed to those of the $(f-1)$-th frame.

For the principal motion:

$$\hat{\mathcal{M}}_{[f-1]} Q_{[f]} \hat{\mathcal{M}}_{[f-1]}{}^{\mathrm{T}} = \Lambda^2_{[f-1]}$$

For the new image:

$$\hat{M}_f Q_{[f]} \hat{M}_f{}^{\mathrm{T}} = A_f A_f{}^{\mathrm{T}}$$

Here $Q_{[f]} = D_{[f]} D_{[f]}{}^{\mathrm{T}}$ and $\Lambda_{[f-1]}$ is the diagonal matrix which has singular values of $M_{[f-1]}$, as described in Step (iv).

(iii) Fixation of the world coordinates

Obtain the orthogonal matrix $\mathcal{E}_f$ which transforms $\bar{S}'_{[f]}$ into $S'_{[f-1]}$, and fix the world coordinates as

$$M_f = \bar{M}_f \mathcal{E}_f{}^{\mathrm{T}}, \quad S'_{[f]} = \mathcal{E}_f \bar{S}'_{[f]}.$$

Here $W'_{[f]}$ is represented as

$$\underset{(5\times P)}{W'_{[f]}} = \begin{pmatrix} \hat{\mathcal{M}}_{[f-1]} D_{[f]} \mathcal{E}_f{}^{\mathrm{T}} \\ M_f \end{pmatrix} S'_{[f]} = \underset{(5\times 3)}{M_{[f]}} \; \underset{(3\times P)}{S'_{[f]}} .$$

(iv) Compression using PCA

Let $M_{[f]} = F_{[f]} \Lambda_{[f]} E_{[f]}$ be SVD of $M_{[f]}$.

The f-th principal motion matrix:

$$\underset{(3\times 3)}{\mathcal{M}_{[f]}} = F_{[f]}{}^{\mathrm{T}} M_{[f]} (= \Lambda_{[f]} E_{[f]})$$

The f-th principal registered measurement matrix:

$$\underset{(3\times P)}{\mathcal{W}'_{[f]}} = \mathcal{M}_{[f]} S'_{[f]} (= F_{[f]}{}^{\mathrm{T}} W'_{[f]})$$

Since $\mathcal{M}_{[f]} \mathcal{M}_{[f]}{}^{\mathrm{T}} = \Lambda^2_{[f]}$, we can use $\Lambda_{[f]}$ for the metric constraints, as described in Step (ii).

3.3 Combining the LMedS Method with the Recursive Factorization Method

To robustly provide updated shape and motion at each frame and at a small computational cost, we combine the procedure based on the LMedS method in Section 3.1 with Step (i) of the recursive factorization method in Section 3.2 as follows.

For each trial j, we randomly sample four points from the f-th recursive registered measurement matrix $W'_{[f]}$ and put them into the 5×4 measurement matrix $W^{[j]}$. The registered measurement matrix for the j-th trial is computed as $W'^{[j]} = W^{[j]} - (0,0,0,\boldsymbol{x}_{fc}^{[j]\mathrm{T}})^{\mathrm{T}}(1,\cdots,1)$. In the same way as in Eqs. (1) and (2), the affine motion $U^{[j]}$ is obtained by SVD of $W'^{[j]}$ and the squared residual $r_p^{[j]2}$ is given as follows.

$$r_p^{[j]2} = \left\| \left(\underset{(5\times 5)}{I} - \underset{(5\times 3)}{U^{[j]}} \; \underset{(3\times 5)}{U^{[j]\mathrm{T}}} \right) \boldsymbol{W}'_{[f]p} \right\|^2 ,$$

where $\boldsymbol{W}'_{[f]p}$ is the p-th column vector of $W'_{[f]}$. Then computing $\mu^{[j]}$, which is the median of $r_p^{[j]2}$ for each trial, we finally obtain the affine motion $U^{[j]}$ for which $\mu^{[j]}$ is minimal.

Although this strategy is very simple, it is very effective in reducing and fixing the computational cost with respect to the number of frames because the only procedures we have to perform for each trial are SVD of a 5×4 matrix $W'^{[j]}$ and the computation of the median $\mu^{[j]}$.

Some inliers are of course incorrectly regarded as outliers because of noise and so on. But because the 3D coordinates of those points are held in the previous frame, we can compensate those points. Then if the compensated points are regarded as the inliers at the next frame, the 3D coordinates of those points can be updated again.

4 Experiments

4.1 Initial Estimate

The robust recursive factorization method requires initial estimates of $S'_{[f]}$, $\Lambda_{[f]}$, and $\mathcal{W}'_{[f]}$. In our implementation, the number k of frames needed to obtain them is determined by the following procedure in order to prevent the method beginning to update the shape and motion before three or more distinct views are observed.

(i) The LMedS method is performed for the tracked features up to the k-th frame and the outliers are discarded as described in Section 3.1.

(ii) If the 3-rd and 4-th singular values, denoted by σ_3 and σ_4, of the registered measurement matrix without the outliers satisfy $\frac{\sigma_4}{\sigma_3} < \alpha$, go to Step (iii). Otherwise $k \leftarrow k + \delta$ and return to Step (i). Here δ is the number of skipped frames.

(iii) If the 3-rd eigenvalue λ_3 of the metric Q satisfies $\lambda_3 > \beta$, then $S'_{[k]}$, $\Lambda_{[k]}$, and $\mathcal{W}'_{[f]}$ are computed as the initial estimates, and they are updated from the $k+1$-th frame. Otherwise, $k \leftarrow k + \delta$ and return to Step (i).

In our implementation, we empirically set $\alpha = 0.2$, $\beta = 0.2$, and $\delta = 5$. To reduce the computational cost of the initial estimates, Step (i) of this method is performed using 5 frames which are sampled at even intervals out of k frames.

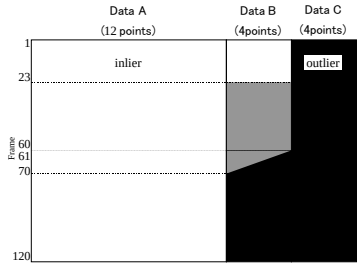


Fig. 1. Discarded outliers.

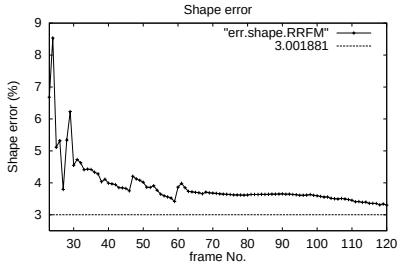


Fig. 2. Shape error.

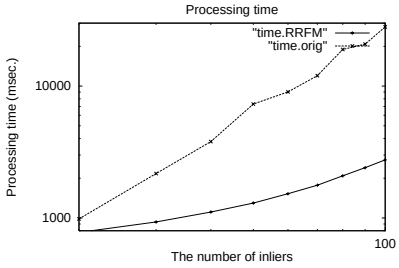


Fig. 3. Processing time.

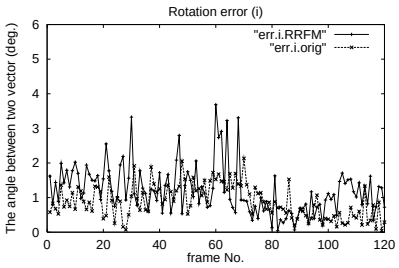


Fig. 4. Motion error. $(\boldsymbol{i}_f)$

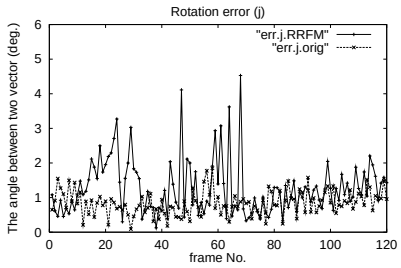


Fig. 5. Motion error. $(\boldsymbol{j}_f)$

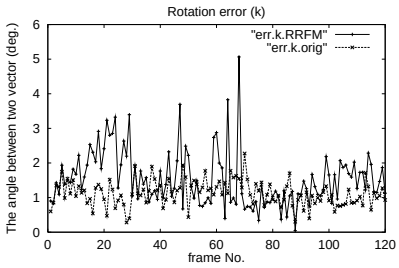


Fig. 6. Motion error. $(\boldsymbol{k}_f)$

4.2 Synthetic Data

In this experiment, the features correspond to 20 points randomly and uniformly distributed within a 200-mm cube. For 120 frames they are projected on image planes by a pinhole camera. The intrinsic parameters are known: pixels are square with 8-μm sides, the image resolution is 640×480, and the focal length is 13 mm. The camera pitch increases linearly to 80 degrees from the 1st frame to the 60th frame and decreases to 0 degrees from the 61th frame to the 120th frame. The camera roll increases linearly to 40 degrees for 120 frames. The average depth z_f changes linearly from $z_1 = 2000$mm to $z_{120} = 1600$mm and the distance of the projection center of the object from the center of the image plane, denoted u_f, changes from $u_1 = 115$ pixels to $u_{120} = 101$ pixels. The paraperspective projection model is used for recovering shape and motion and $J = 100$.

We divide 20 points into the following three groups.

Data A [12 points] Correct matches, which have Gaussian noise the standard deviation of which is 1 pixel, for 120 frames.

Data B [4 points] Correct matches, which have Gaussian noise the standard deviation of which is 3 pixels, from the 1st frame to the 60th frame, and false matches from the 61th frame to the 120th frame

Data C [4 points] False matches for 120 frames

In this experiment, the number k of frames used to obtain the initial estimates is chosen as 23 by the procedure in Section 4.1. The percentage of outliers is 20% at the 23th frame and 40% at the 108th frame. Fig. 1 shows that feature points in the black area are regarded as the outliers without fail, ones in the gray area are regarded as the outliers off and on, and ones in the white area are regarded as the inliers at almost all frames by our method. A few features of Data A are wrongly regarded as the outliers several times, but those points are compensated using the 3D coordinates of them at the previous frame and the 3D motion at the present frame. Those points can thus be updated again.

We compare the shape error and the motion error of our method with those of the original batch-type factorization method. The original method uses only Data A and our method uses Data A, B, and C. Fig. 2 shows the shape error, and Figs. 4, 5, and 6 show the motion error. The shape error of the original method with 120 frames is about 3% and the errors of $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ are each about 1 degrees. As mentioned above, our method can discard the outliers correctly and the shape and motion errors gradually decrease as the number of frames increases, even though both errors are relatively large at the beginning of the sequence.

We also compare the processing time, on a Pentium-II-450MHz PC, of our method with that of the original method. Fig. 3 shows the results for 120 frames and for $J = 100$. The number of features are varied from 20 to 100. Here "feature point" denotes "inlier (Data A)" because the original method uses only inliers and the processing time of our method depends on the number of inliers and not the number of outliers. Data A, B, and C is the same ratio as the above experiment. If we suppose that the number of frames needed to process in one second is 30 frames like NTSC, it takes only 2.8 seconds to recover the 3D coordinates of 100 features and the motion for 4.0 seconds (120 frames). Therefore the result shows that we can use our method in real-time applications.

4.3 Real Images

In this experiment, video of a stuffed koala held by a person was taken from a consumer-quality camera held by another person. In theory the LMedS method requires dominant data which account for more than 50% of all the data. To satisfy this requirement, we assume that features of one object exist locally. We first used a fast and robust method to roughly segment moving regions by using a global flow estimation in [3], where moving regions are regions that have a flow that differs from the global flow of the image.

In this method, the global flow is estimated by using a gradient-based method and the M-estimator [2]. Candidates for moving regions are detected by pixel-wise matching from two successive frames, votes for the pixels of candidates at each frame are accumulated sequentially, and then candidates with a large number of votes are selected as moving regions.

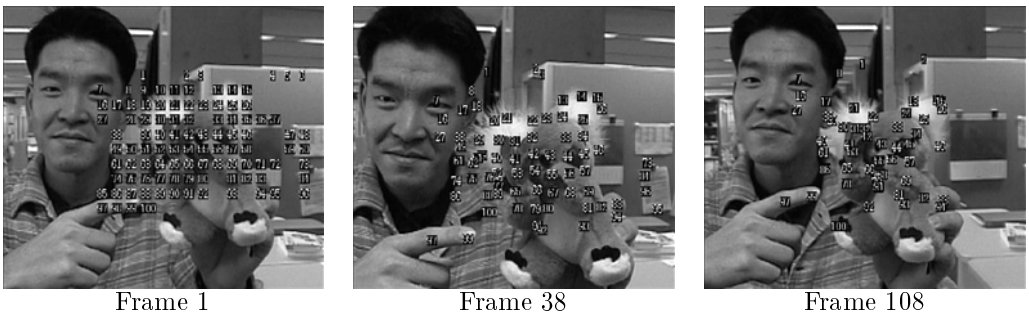
Features were then detected and tracked in a circumscribed rectangle of the region, and the 3D motion was estimated at each frame by using our method. In order to detect and track feature points, we use the normalized correlation method which is one of the functions of an image processing PCI board IP5000 manufactured by Hitachi.

In this example, a stuffed koala and a face were segmented. This paper describes the results on the stuffed koala because of the space constraint. The number k of frames used to obtain the initial estimates was chosen as 38. Fig. 7 (Movie 1) shows 100 features detected at the 1-st frame, 76 features tracked through 38 frames, and 56 features tracked through 108 frames. The percentage of outliers was 36% at the 38th frame and 30% at the 108th frame. Fig. 8 (Movie 2) shows the position of inliers retained at the 108th frame. To assess the result, we modeled the dense shape of the stuffed koala by using the recovered dominant motion. Three images (the 49th, 50th, and 51th frame) were rectified by the motion [16][4] as shown in Fig. 9, and the dense shape were recovered by a simple correlation-based stereo method as shown in Fig. 10 (Movie 3). Another example on the face is shown in Movies 4, 5, and 6.

Finally three tasks mentioned above, which are the method to roughly segment moving regions, the feature point detector/tracker, and our proposed method, were implemented on a PC-Cluster system consisting of two PCs (CPU: Dual Pentium II 450MHz, OS: Linux-2.2) connected by 100Base-TX. In this system, Video For Linux was used for capturing video frames and PVM library was used for network programming. An example of online experiments is shown in Movie 7. In this movie, large green rectangles indicate circumscribed rectangles of moving regions, small white squares indicate detected and tracked feature points, blue ones indicate initial inliers, and red ones indicate updated inliers. A throughput of the system was around 200 msec and the latency was around 1 sec because the network programming is not optimized currently.

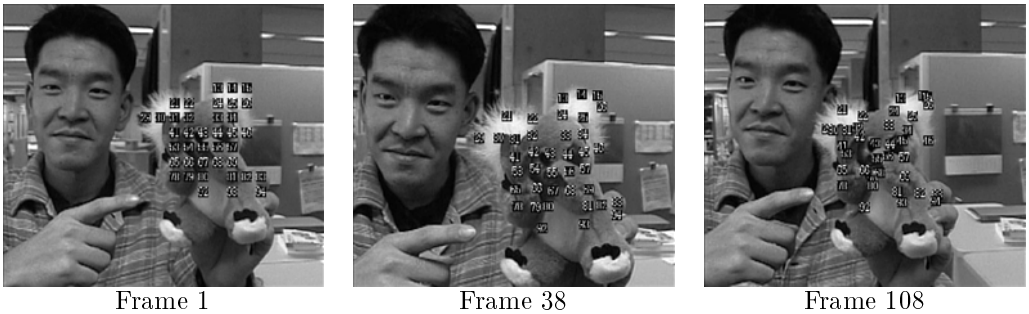
5 Conclusion

The robust recursive factorization method for recovering structure and motion from multiple views under affine projection described here compresses the motion matrix, the metric constraints, and the measurement matrix by using the PCA and recovers the dominant structure and the motion by using the LMedS method. Experiments



Frame 1 Frame 38 Frame 108

Fig. 7. Tracking results.



Frame 1 Frame 38 Frame 108

Fig. 8. Inliers selected by our method.

Frame 49 Frame 50 Disparity map

Fig. 9. Rectification and stereo matching.

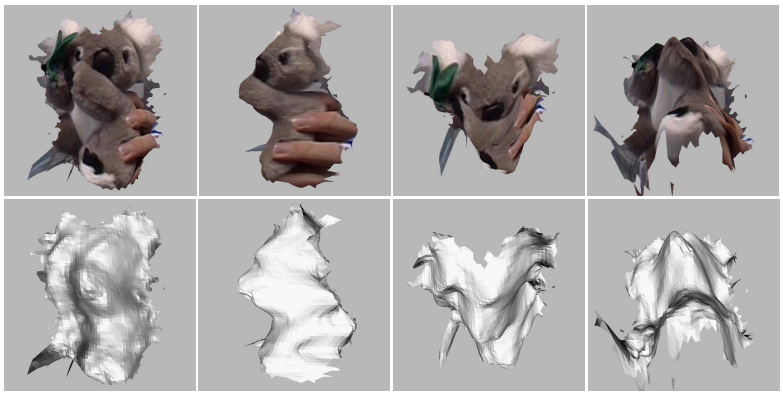


Fig. 10. Recovered dense shape.

with synthetic data have shown that the method is reliable and has a computational cost small enough that it can be executed in real-time on an off-the-shelf PC, and then the method has been applied to real images.

We have not yet evaluated the breakdown points rigorously nor have we considered adding new feature points. Future research will have to address these issues. We will also have to improve the performance of feature tracker by speeding it up and by using cooperative procedures such as feedback of 3D motion updated by our method.

Acknowledgments: This work was conducted as a part of the Real World Computing (RWC) Program.

References

[1] J. Fujiki, T. Kurata, and M. Tanaka. Iterative factorization method for object recognition. In *Proc. SPIE98*, vol. 3454, 192–201, 1998.

[2] P. J. Huber. *Robust Statistics*. John Wiley & Sons, NY, 1981.

[3] M. Kourogi, T. Kurata, J. Hoshino, and Y. Muraoka. Real-time image mosaicing from a video sequence. In *Proc. ICIP99*, 1999. (to appear).

[4] T. Kurata, J. Fujiki, and K. Sakaue. Affine epipolar geometry via factorization method. In *Proc. ICPR98*, vol. 1, 862–866, 1998.

[5] T. Morita and T. Kanade. A sequential factorization method for recovering shape and motion from image streams. *PAMI*, 19(8):858–867, 1997.

[6] D. D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Proc. ICCV98*, 696–702, 1998.

[7] J. L. Mundy and A. Zisserman. *Geometric Invariance from Motion*. MIT Press, Cambridge, MA, 1992.

[8] C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *PAMI*, 13(3):206–218, 1997.

[9] L. Quan. Self-calibration of an affine camera from multiple views. *IJCV*, 19(1):93–105, 1996.

[10] L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *PAMI*, 19(8):834–845, 1997.

[11] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, NY, 1987.

[12] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proc. ECCV96*, 709–720, 1996.

[13] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography. *IJCV*, 9(2):137–154, 1992.

[14] P. Torr and A. Zisserman. Robust computation and parametrization of multiple view relations. In *Proc. ICCV98*, 727–732, 1998.

[15] T. Ueshiba and F. Tomita. A factorization method for projective and euclidean reconstruction from multiple perspective views via iterative depth estimation. In *Proc. 5th ECCV*, vol. 1, 296–310, 1998.

[16] G. Xu and Z. Zhang. *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*, vol. 6 of *Computational Imaging and Vision*. Kluwer Academic Publishers, 1996.

[17] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *IJCV*, 27(2):161–195, 1998.