MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

A Method for Estimating Rigid Object Motion Using Regularized Scene Flow

Hiroki Mizuno
Department of Computer Science
Chubu University
1200 Matsumotocho, Kasugai,
Aichi 487-5501 Japan.
hiroki@vision.cs.chubu.ac.jp

Hironobu Fujiyoshi
Department of Computer Science
Chubu University
1200 Matsumotocho, Kasugai,
Aichi 487-5501 Japan.
hf@cs.chubu.ac.jp

Abstract

In this paper, we propose a method for estimating object motion by three-dimensional scene flow using multiple cameras. The scene flow is regularized by applying subspace constraints and then object motion is estimated using RANSAC estimation. Regularizing the scene flow using subspace constraints results in highly accurate scene flow because it eliminates the effect of noise caused by computing of optical flow. Simulation and experimental results demonstrated that this method can be used to accurately estimate scene flow and object motion parameters for translation and rotation.

1 Introduction

Scene flow is the three-dimensional motion field of points in the world, just as optical flow is the two-dimensional motion field of points in an image. Vedula et al. have proposed a framework for computing dense, non-rigid scene flow from optical flow[5]. Possible applications of scene flow include dynamic rendering, from the generation of slow-motion replays to the measurement and modeling of human actions.

The approach proposed by Vedula can be used to estimate three-dimensional scene flow without any knowledge of the scene structure. Their approach uses multiple cameras, calibration information for each camera, and the optical flow in the image planes. However, estimating three-dimensional scene flow with high accuracy is difficult because optical flows have noise due to the so-called "aperture problem". For example, if the optical flows at the corresponding points between cameras contain noise, the orientation and magnitude of the estimated scene flow are inaccurate.

In this paper, we present a method for estimating three-dimensional scene flow with high accuracy using optical flows obtained from several cameras without reconstructing object shapes in three-dimensional space. It is based on the assumption that object motion is rigid in a short time intervals, so the set of scene flows in a sequence of frames resides in a low-dimensional linear subspace. The scene flow sequence can be represented as a $3M \times N$ measurement matrix, which is made up of the world coordinates of N points tracked through M frames. If the scene flow can be reconstructed without noise, the measurement matrix will have a rank of at most 4. However, if noise corrupts the scene flow, the measurement matrix will have a higher rank.

Linear subspace constraints have been used successfully. Tomasi and Kanade proposed a framework for recovering 3D information based on linear subspace constraints[3]. Their approach uses known 2D correspondence under orthography. Irani has proposed a method for estimating optical flow in a sequence of frames[6]. Her approach, which improves the accuracy of optical flow, is based on the assumption that scene structure dose not change in short time intervals. Our approach is to regularize the three-dimensional scene flow directly using singular value decomposition. It does not require point tracking over many frames. Next, we describe our proposed method for estimating object motion from a reconstructed scene flow. Our approach can estimate rigid-motion parameters with high accuracy because it combines regularization using subspace constraints with RANSAC estimation.

2 Three-dimensional scene flow

In the same way that optical flow describes an instantaneous motion field in an image, scene flow can be described as a three-dimensional flow field, $\frac{d\boldsymbol{x}}{dt}$, representing the motion at every point in the scene. In this section, we describe a method for reconstructing scene flow without having any knowledge of the scene structure using only the optical flow of several cameras, as proposed by Vedula et al. [5]. There are two steps. First, the location of the scene flow is computed. Second, the orientation and magnitude of the scene flow are computed. The details of each step are described as follows.

2.1 Computing location of scene flow

Since we know the projection matrix from the camera calibration, we can express the relationship between scene flow and optical flow:

$$\frac{d\boldsymbol{u}_k}{dt} = \frac{\partial \boldsymbol{u}_k}{\partial \boldsymbol{x}}\frac{d\boldsymbol{x}}{dt}, \tag{1}$$

where $\frac{d\boldsymbol{x}}{dt}$ is the scene flow at point $\boldsymbol{x}$ in the scene. The 2D motion of the kth camera image (i.e., the optical flow) projection is $\frac{d\boldsymbol{u}_k}{dt}$. The Jacobian $\frac{\partial \boldsymbol{u}_k}{\partial \boldsymbol{x}}$ describes the relationship between a small change at the 3D point and its image taken by camera k. Equation (1) can be written as:

$$\frac{d\boldsymbol{x}}{dt} = \left(\frac{\partial \boldsymbol{u}_k}{\partial \boldsymbol{x}}\right)^* \frac{d\boldsymbol{u}_k}{dt} + \mu \boldsymbol{r}_k(\boldsymbol{u}_k) \tag{2}$$

where $\left(\frac{\partial \boldsymbol{u}_k}{\partial \boldsymbol{x}}\right)^*$ is the pseudo-inverse of $\frac{\partial \boldsymbol{u}_k}{\partial \boldsymbol{x}}$, $\boldsymbol{r}_k(\boldsymbol{u}_k)$ is the direction of a ray through pixel $\boldsymbol{u}_k$, and μ is an unknown

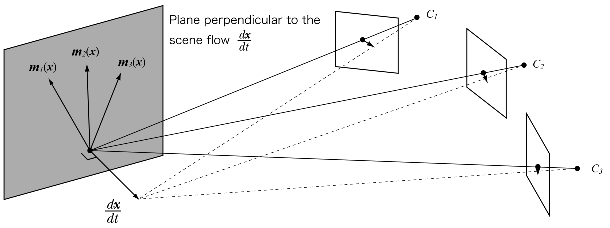


Figure 1: Property of $\boldsymbol{m}_k(\boldsymbol{x})$

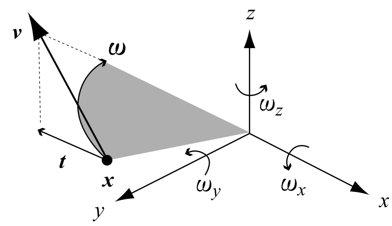


Figure 2: Motion model

constant. Therefore, we have the following constraint on the scene flow:

$$\boldsymbol{m}_k(\boldsymbol{x}) \cdot \frac{d\boldsymbol{x}}{dt} \equiv \left[\left(\frac{\partial \boldsymbol{u}_k}{\partial \boldsymbol{x}} \right)^* \frac{d\boldsymbol{u}_k}{dt} \times \boldsymbol{r}_k(\boldsymbol{u}_k) \right] \cdot \frac{d\boldsymbol{x}}{dt} = 0 \quad (3)$$

where $\boldsymbol{m}_k(\boldsymbol{x})$ is a vector perpendicular to the scene flow. If scene flow $\frac{d\boldsymbol{x}}{dt}$ exists at $\boldsymbol{x}$, the vectors $\boldsymbol{m}_k(\boldsymbol{x})$ obtained for each camera should all lie in a plane perpendicular to the scene flow $\frac{d\boldsymbol{x}}{dt}$, as shown in Figure 1. The measure of vector coplanarity, $\boldsymbol{m}_k(\boldsymbol{x})$, which is obtained from the camera k, is computed using

$$\boldsymbol{M}(\boldsymbol{x}) = \sum_k \hat{\boldsymbol{m}}_k \hat{\boldsymbol{m}}_k^T, \quad (4)$$

where $\hat{\boldsymbol{m}}_k$ is a unit vector normalized from $\boldsymbol{m}_k(\boldsymbol{x})$. If all $\boldsymbol{m}_k(\boldsymbol{x})$ is coplanarity, the smallest eigenvalue of $\boldsymbol{M}(\boldsymbol{x})$ is zero.

Computing the location of scene flow requires discretizing the scene into a three-dimensional array of voxels. The visibility of each point in the 3D space must be determined. We use the voxel coloring algorithm proposed by Seitz and Dyer [7] to search for the locations where the scene flow exists.

2.2 Computing orientation and magnitude of scene flow

If location $\boldsymbol{x}$ of an existing scene flow is observed by multiple cameras $(n > 2)$, we can extend Equation (1):

$$\begin{bmatrix} \frac{\partial u_1}{\partial t} \\ \frac{\partial v_1}{\partial t} \\ \vdots \\ \frac{\partial u_n}{\partial t} \\ \frac{\partial v_n}{\partial t} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_1}{\partial X_x} & \frac{\partial u_1}{\partial X_y} & \frac{\partial u_1}{\partial X_z} \\ \frac{\partial v_1}{\partial X_x} & \frac{\partial v_1}{\partial X_y} & \frac{\partial v_1}{\partial X_z} \\ & \vdots & \\ \frac{\partial u_n}{\partial X_x} & \frac{\partial u_n}{\partial X_y} & \frac{\partial u_n}{\partial X_z} \\ \frac{\partial v_n}{\partial X_x} & \frac{\partial v_n}{\partial X_y} & \frac{\partial v_n}{\partial X_z} \end{bmatrix} \frac{d\boldsymbol{x}}{dt} \quad (5)$$

where n is the number of cameras observing the scene flow. This gives us $2n$ equations in 3 unknowns, so for $n \geq 2$ we have an overconstrained system. Once the location of the scene flow is determined, its orientation and magnitude can be computed using the least-squares method and equation (5).

3 Method for reconstructing scene flow

To solve the problem of noise in optical flows affecting the accuracy of scene flow, we propose a method for reconstructing scene flow with high accuracy by regularizing the scene flow, and assuming that object motion is rigid in short time intervals.

3.1 Motion model and measurement matrix

Let $\boldsymbol{v}_{ij}$ denote the movement of an object at 3D point $\boldsymbol{x}_i$ at time j. If the object motion is rigid, $\boldsymbol{v}_{ij}$ is given by

$$\boldsymbol{v}_{ij} = \boldsymbol{t}_j + \boldsymbol{\omega}_j \times \boldsymbol{x}_i, \quad (6)$$

where $\boldsymbol{t}_j = [t_{x_j}, t_{y_j}, t_{z_j}]^T$ is a translation vector, and $\boldsymbol{\omega}_j = [\omega_{x_j}, \omega_{y_j}, \omega_{z_j}]^T$ is a rotation velocity vector, as shown in Figure 2. Equation (6) is rewritten as

$$\begin{bmatrix} v_{x_{ij}} \\ v_{y_{ij}} \\ v_{z_{ij}} \end{bmatrix} = \begin{bmatrix} t_{x_j} & 0 & 0 & 0 & -\omega_{z_j} & \omega_{y_j} \\ 0 & t_{y_j} & 0 & \omega_{z_j} & 0 & -\omega_{x_j} \\ 0 & 0 & t_{z_j} & -\omega_{y_j} & \omega_{x_j} & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ X_{x_i} \\ X_{y_i} \\ X_{z_i} \end{bmatrix}. \quad (7)$$

We rewrite Equation (7) as

$$\begin{bmatrix} v_{x_{ij}} \\ v_{y_{ij}} \\ v_{z_{ij}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_{x_j} \\ \hline \boldsymbol{s}_{y_j} \\ \hline \boldsymbol{s}_{z_j} \end{bmatrix}_{(3\times 6)} \boldsymbol{q}_{i(6\times 1)} \quad (8)$$

where

$$\begin{aligned} \boldsymbol{s}_{x_j} &= [t_{x_j}, 0, 0, 0, -\omega_{z_j}, \omega_{y_j}], \\ \boldsymbol{s}_{y_j} &= [0, t_{y_j}, 0, \omega_{z_j}, 0, -\omega_{x_j}], \\ \boldsymbol{s}_{z_j} &= [0, 0, t_{z_j}, -\omega_{y_j}, \omega_{x_j}, 0], \\ \boldsymbol{q}_i &= [1, 1, 1, X_{x_i}, X_{y_i}, X_{z_i}]^T. \end{aligned}$$

Equation (8) represents the motion at location $\boldsymbol{x}_k$. We extend it for N points by writing

$$\begin{bmatrix} \boldsymbol{v}_{x_j} \\ \hline \boldsymbol{v}_{y_j} \\ \hline \boldsymbol{v}_{z_j} \end{bmatrix}_{(3\times N)} = \begin{bmatrix} \boldsymbol{s}_{x_j} \\ \hline \boldsymbol{s}_{y_j} \\ \hline \boldsymbol{s}_{z_j} \end{bmatrix} \boldsymbol{Q}_{(6\times N)} \quad (9)$$

where

$$\boldsymbol{Q}_{(6\times N)} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N] = \begin{bmatrix} 1 & \ldots & 1 \\ 1 & \ldots & 1 \\ 1 & \ldots & 1 \\ X_{x_1} & \ldots & X_{x_N} \\ X_{y_1} & \ldots & X_{y_N} \\ X_{z_1} & \ldots & X_{z_N} \end{bmatrix}.$$

Additionally, we extend each vector $(\boldsymbol{s}_{x_j}, \boldsymbol{s}_{y_j}, \boldsymbol{s}_{z_j})$ in Equation (9) to describe multiple frames, M:

$$\begin{bmatrix} \boldsymbol{V}_x \\ \hline \boldsymbol{V}_y \\ \hline \boldsymbol{V}_z \end{bmatrix}_{(3M\times N)} = \begin{bmatrix} \boldsymbol{S}_x \\ \hline \boldsymbol{S}_y \\ \hline \boldsymbol{S}_z \end{bmatrix}_{(3M\times 6)} \boldsymbol{Q}_{(6\times N)} \quad (10)$$

where

$$\boldsymbol{S}_x = \begin{bmatrix} \boldsymbol{s}_{x_1} \\ \vdots \\ \boldsymbol{s}_{x_M} \end{bmatrix}, \quad \boldsymbol{S}_y = \begin{bmatrix} \boldsymbol{s}_{y_1} \\ \vdots \\ \boldsymbol{s}_{y_M} \end{bmatrix}, \quad \boldsymbol{S}_z = \begin{bmatrix} \boldsymbol{s}_{z_1} \\ \vdots \\ \boldsymbol{s}_{z_M} \end{bmatrix}.$$

We call the matrix $[\boldsymbol{V}_x/\boldsymbol{V}_y/\boldsymbol{V}_z]$ the "measurement matrix". All three rows of matrix $\boldsymbol{Q}$ contain 1, so, $\mathrm{rank}(\boldsymbol{Q}) \leq 4$. Since instantaneous object motion can be represented as a constant velocity, each matrix ($\boldsymbol{S}_x$, $\boldsymbol{S}_y$, $\boldsymbol{S}_z$) will have rank of at most 1. Therefore, $\mathrm{rank}([\boldsymbol{S}_\mathrm{x}/\boldsymbol{S}_\mathrm{y}/\boldsymbol{S}_\mathrm{z}]) \leq 3$. The matrix $[\boldsymbol{V}_x/\boldsymbol{V}_y/\boldsymbol{V}_z]$ of rank is thus limited to the smaller of $\boldsymbol{Q}$ and $[\boldsymbol{S}_x/\boldsymbol{S}_y/\boldsymbol{S}_z]$ because matrix $[\boldsymbol{V}_x/\boldsymbol{V}_y/\boldsymbol{V}_z]$ is the product of these matrices. Here, we assume that instantaneous object motion can be represented as a constant velocity, so we constrain the measurement matrix to have a rank of at most 3.

3.2 Regularizing scene flow using subspace constraints

If the 2D optical flows contain noise and outliers, the measurement matrix computed from the 3D scene flow will not have a rank of exactly 3. We thus regularize the scene flow applying subspace constraints to the measurement matrix. In this section, we introduce the notion of rank approximation, using the concept of singular value decomposition.

The matrix $[\boldsymbol{V}_x/\boldsymbol{V}_y/\boldsymbol{V}_z]$ can be decomposed to

$$\begin{bmatrix} \boldsymbol{V_x} \\ \hline \boldsymbol{V_y} \\ \hline \boldsymbol{V_z} \end{bmatrix}_{(3M\times N)} = \boldsymbol{U}_1 \boldsymbol{D} \boldsymbol{U}_2^T, \tag{11}$$

where $\boldsymbol{U}_1$ is a $3M \times 3M$ matrix, $\boldsymbol{U}_2$ is a $N \times N$ matrix, and both are orthogonal; $\boldsymbol{D}$ is a special diagonal matrix, and its diagonal component is singular value $\boldsymbol{d} = [d_1, d_2, \ldots, d_{3M}]^T$. To limit the rank of the measurement matrix to be at most 3, we change matrix $\boldsymbol{D}$ to the following equation using $\boldsymbol{d}' = [d_1, d_2, d_3, 0, \ldots, 0]^T$:

$$\boldsymbol{D}' = \begin{bmatrix} d_1 & 0 & \ldots & \ldots & \ldots & 0 \\ 0 & d_2 & 0 & \ldots & \ldots & 0 \\ 0 & 0 & d_3 & 0 & \ldots & 0 \\ 0 & \ldots & \ldots & 0 & \ldots & 0 \\ 0 & \ldots & \ldots & \ldots & \ddots & 0 \\ 0 & \ldots & \ldots & \ldots & \ldots & 0 \end{bmatrix}_{(3M\times N)}.$$

Next, a new measurement matrix is computed using $\boldsymbol{D}'$, $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$:

$$\begin{bmatrix} \boldsymbol{V_x}' \\ \hline \boldsymbol{V_y}' \\ \hline \boldsymbol{V_z}' \end{bmatrix}'_{(3M\times N)} = \boldsymbol{U}_1 \boldsymbol{D}' \boldsymbol{U}_2^T. \tag{12}$$

New measurement matrix $[\boldsymbol{V}'_x/\boldsymbol{V}'_y/\boldsymbol{V}'_z]'$ has a rank of at most 3. The scene flow, which can be computed from this matrix, is regularized using the motion model given by Equation (6).

4 Motion estimation from scene flow

In this section, we describe how the motion parameters are estimated from the reconstructed scene flow.

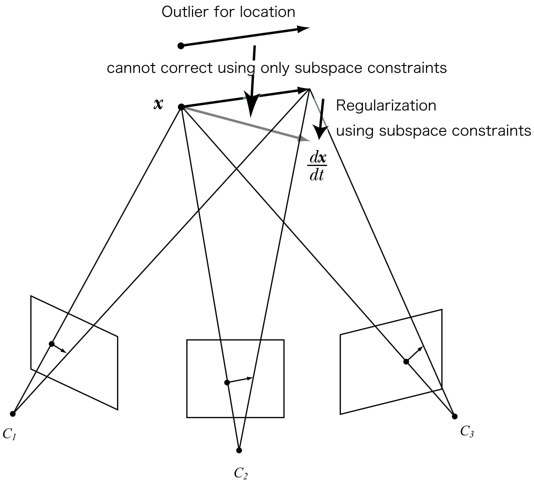


Figure 3: Outlier for location

4.1 Motion estimation

Since motion vector $\boldsymbol{v}_{ij}$ at point $\boldsymbol{x}_i$ in the 3D scene is known, the motion model (Equation (6)) can be expressed for time j as

$$\begin{aligned} \boldsymbol{v}_{ij} &= \begin{bmatrix} 0 & X_{z_i} & -X_{y_i} & 1 & 0 & 0 \\ -X_{z_i} & 0 & X_{x_i} & 0 & 1 & 0 \\ X_{y_i} & -X_{x_i} & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_j \\ \hline \boldsymbol{t}_j \end{bmatrix} \\ &= \begin{bmatrix} -[\boldsymbol{x}_i]_\times & \boldsymbol{I}_{(3\times3)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_j \\ \hline \boldsymbol{t}_j \end{bmatrix}, \end{aligned} \tag{13}$$

where $[\boldsymbol{x}_i]_\times$ is the skew-symmetric matrix of $\boldsymbol{x}_i$, and $\boldsymbol{I}_{(3\times3)}$ is the unit matrix of 3×3. This equation provides two linear constraints on the motion parameters because it contains a skew-symmetric matrix. Therefore, we can extend Equation (13) using N points:

$$\begin{bmatrix} \boldsymbol{v}_{1j} \\ \vdots \\ \boldsymbol{v}_{Nj} \end{bmatrix} = \begin{bmatrix} -[\boldsymbol{x}_1]_\times & \boldsymbol{I}_{1(3\times3)} \\ \vdots & \vdots \\ -[\boldsymbol{x}_N]_\times & \boldsymbol{I}_{N(3\times3)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_j \\ \hline \boldsymbol{t}_j \end{bmatrix}. \tag{14}$$

This gives us $2N$ equations with six unknown constants: $\omega_x, \omega_y, \omega_z$,and t_x, t_y, t_z. Since $N \geq 3$, we can estimate the motion parameters because the system is overconstrained.

4.2 RANSAC estimation

The regularized scene flow computed using the method described in section 3 is a model of rigid motion. Subspace constraints can be used to regularize the orientation and magnitude of the scene flow, but they cannot be used to reject the outliers not in the proper location. We thus use RANSAC estimation to reject the outliers by evaluating a median value of errors. (see Figure 3). We thus use RANSAC estimation to reject the outliers. There are seven steps in the process of estimating the parameters using RANSAC.

Step1 Select three points randomly from the of scene flow.

Step2 Estimate the motion parameters from these three points using the least-squares method.

Step3 Compute the scene flow from the estimated parameters.

Step4 Calculate errors of reconstructed scene flow and computed scene flow.

Step5 Identify a median value of errors.

Step6 Repeat **Step1** to **Step5**

Step7 Select the parameters which is minimum value in these median value of errors

5 Evaluation

We tested out method using simulation and experiment. In the simulation, we reconstructed a scene flow using subspace constraints and estimated the object motion parameters from the optical flows. In the experiment, we used three frames for an actual image: the current frame and the ones immediately before and after the current frames ($M = 2$).

5.1 Reconstructing scene flow using simulation model

In the simulation, we used produced scene flow of known shape objects such as cube, sphere, and arbitrary with rotation and translation. We added noise to orientation, magnitude, and location of each scene flow. Figure 4 (a) shows a example of scene flow with noise, and (b) shows a regularized one using our method. The orientation of the scene

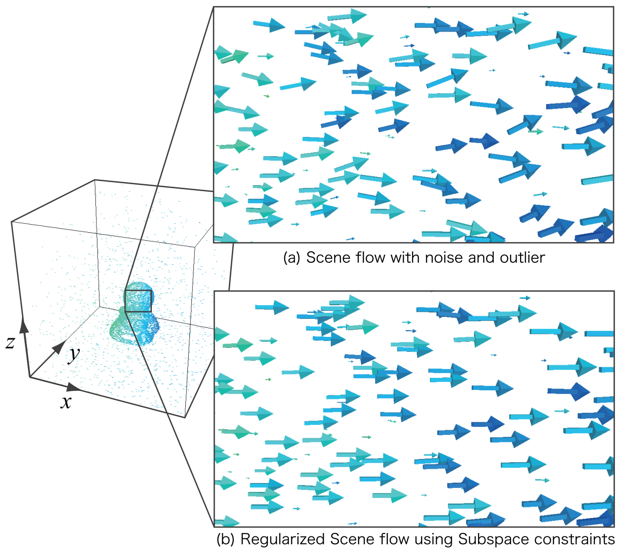


Figure 4: Example of simulation

flow in (a) is dispersed, while that in (b) is regular. This means that it is possible to regularize the scene flow with high accuracy by using subspace constraints.

The evaluation metric used to judge the accuracy of the reconstructed scene flow is the degree of similarity , $\cos\theta$, between reconstructed scene flow $\boldsymbol{v}$ and true scene flow $\boldsymbol{v}_t$:

$$\cos\theta = \frac{\boldsymbol{v} \cdot \boldsymbol{v}_t}{||\boldsymbol{v}||\ ||\boldsymbol{v}_t||}. \tag{15}$$

The other metric is the difference from true magnitude is computed as:

$$\frac{|\ ||\boldsymbol{v}|| - ||\boldsymbol{v}_t||\ |}{||\boldsymbol{v}_t||}. \tag{16}$$

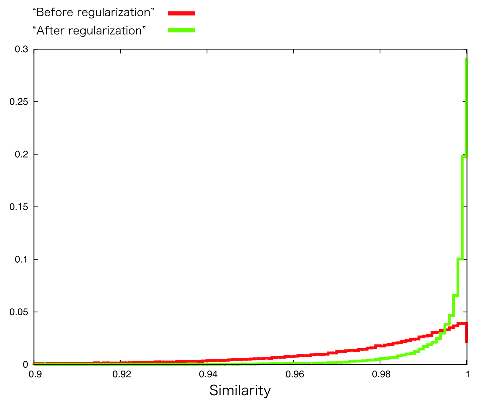


Figure 5: Histogram of similarity

Figure 5 shows a histogram of the degree of similarity for all objects. Before regularization, the ratio of high similarity, i.e., greater than 0.98, was 56.7%. After regularization, it was 94.2%. Figure 6 shows a histogram of the difference from true magnitude. Before regularization , the raio

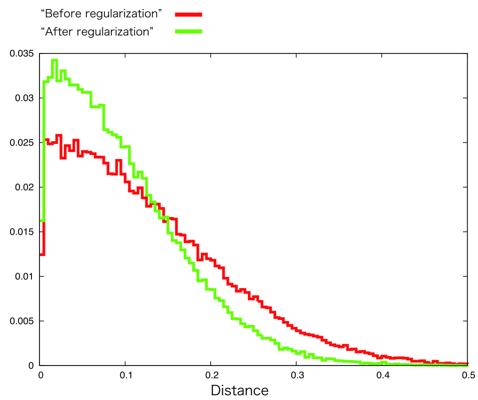


Figure 6: Histogram of difference from true magnitude

of high accuracy, i.e., greater than 0.1, was 46.3%. After regularization, it was 58.3%. Using subspace constraints is thus an efficient way to improve the accuracy of scene flow.

5.2 Motion estimation using simulation model

We estimated six parameters ($\omega_x, \omega_y, \omega_z$, and t_x, t_y, t_z) of object motion using the simulation model described above. The input was scene flow before and after regularization using subspace constraints. To determine the accuracy of the motion estimation, we estimated the motion using RANSAC and the least-squares method using all of the reconstructed scene flow. Figure 7 shows estimation errors for each motion ("rotation" and "translation + rotation"). Figure 7(a) and (b) shows that the scene flow regularized using RANSAC had the highest accuracy. Since our method uses RANSAC, it is possible to eliminate the effects of outliers. Our method can estimates the motion parameters with high accuracy because the scene flow is regularized.

5.3 Experimental result using real images

In our experiment, we mounted five cameras as shown in Figure 8. The shutters were synchronized to open simultaneously. The object was a person's hand and forearm

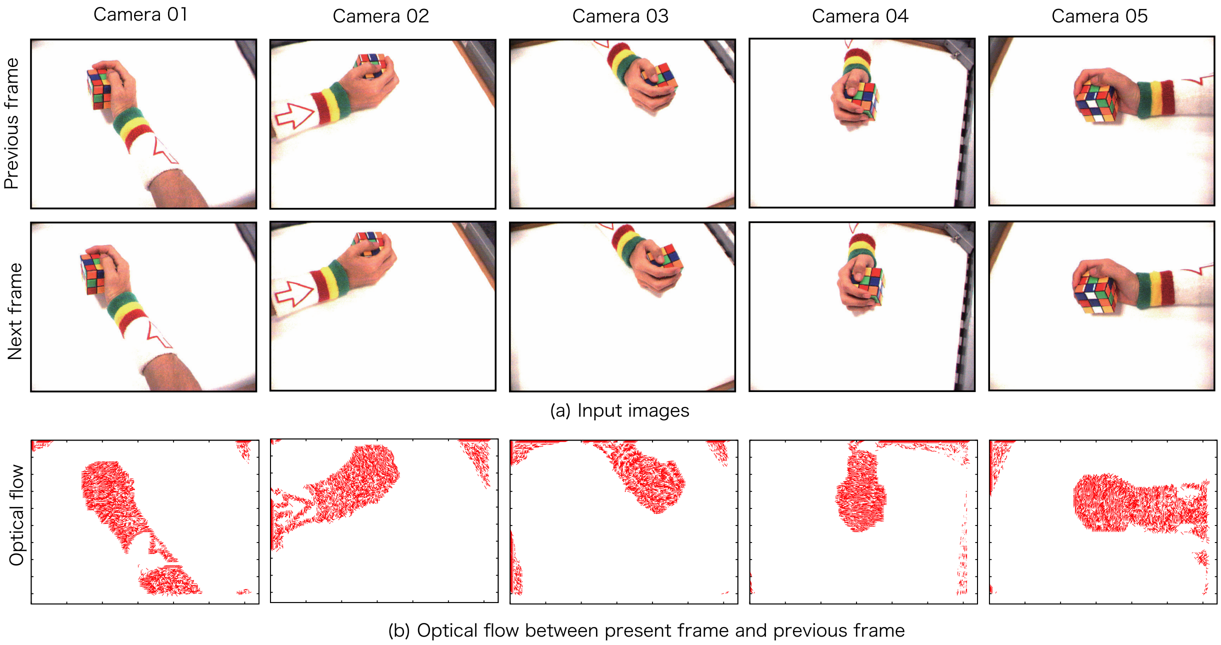


Figure 9: Captured images and optical flows (translation movement only)

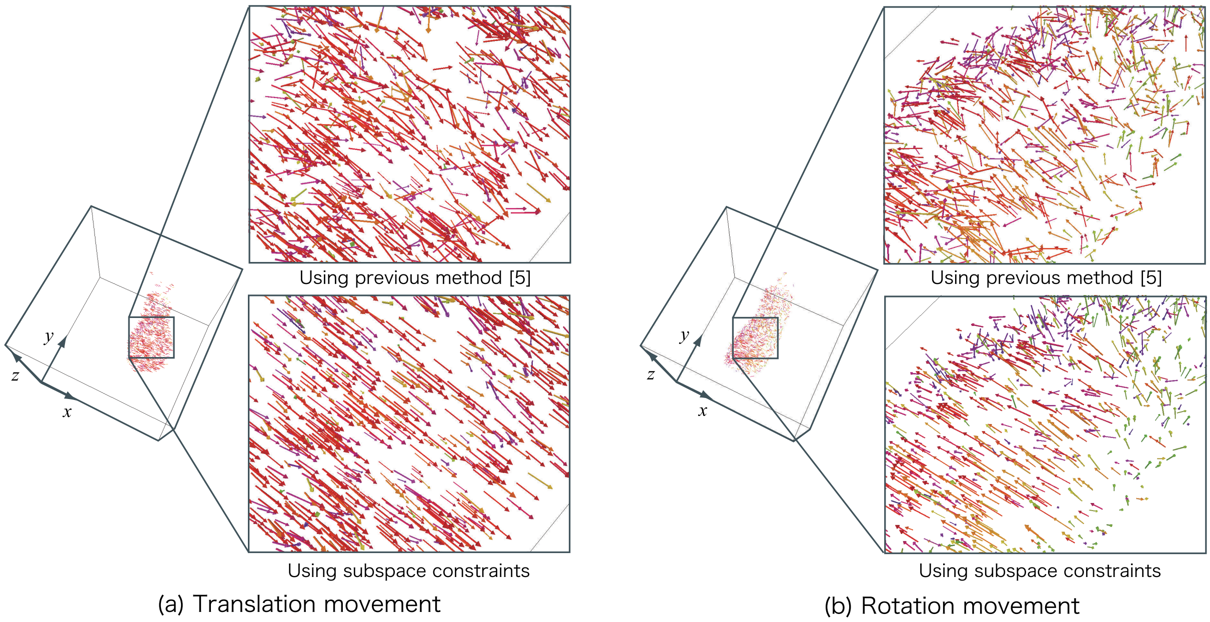
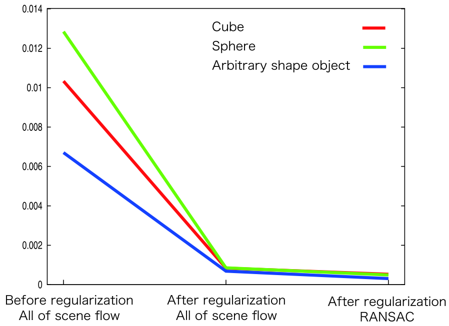


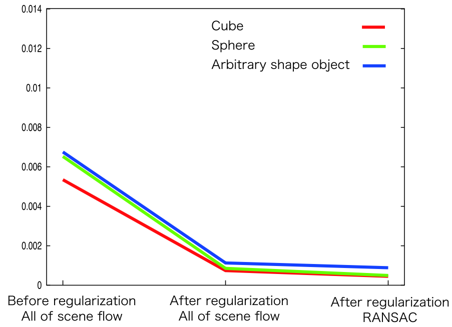
Figure 10: Example of reconstructed 3-D scene flows

Table 1: Estimated motion parameters for translation and rotation sequences

	ω_x [deg/frame]	ω_y [deg/frame]	ω_z [deg/frame]	t_x [mm/frame]	t_y [mm/frame]	t_z [mm/frame]
Translation	-0.494	0.380	-0.244	0.818	-4.523	-0.078
Rotation	-5.490	-0.114	0.978	-0.246	5.901	2.197



(a) Sequence of rotation



(b) Sequence of translation and rotation

Figure 7: Estimation errors of each motion

moving at about 150mm/s. The optical flows between the present and previous frames are shown in Figure 9. Figure 10 shows scene flows reconstructed from these images. Table 1 shows examples of estimated motion parameters for the translation and rotation sequences. For the translation sequence, all of the rotation velocity vectors $(\omega_x, \omega_y, \omega_z)$ were less than 1.0. This means that sequence was not rotated. The parameter with the highest value was t_y. It shows that the object moved 4.5 mm per frame, which corresponds to 135 mm per second. This means that high accuracy was achieved because the difference was only 10 $\%$. For the rotation sequence when the object was rotated 180 degrees around the x axis per second. The parameter with the highest value was ω_x. It shows that the object rotated 5.5 degrees around the x axis per frame, which corresponds to 165 degrees per second. This means that the motion parameters can be obtained with high accuracy because the difference was only about 8$\%$. Although this sequence was not translated, the value of t_y was high. This is why rotation around the x axis closely resembles translation to the y direction because the object is seen from the upper cameras. This problem can be solved by using more cameras mounted at different height.

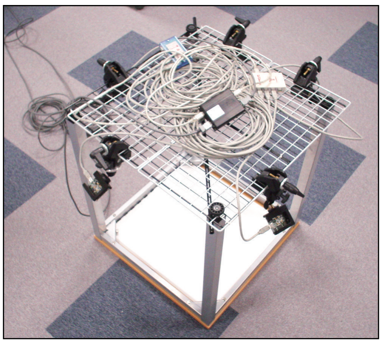


Figure 8: Experimental setup

6 Conclusion

The method we have described for regularizing three-dimensional scene flows uses subspace constraints when the object is assumed to have rigid motion. Regularizing the scene flow using subspace constraints results in highly accurate scene flow because it eliminates the effect of noise caused by computing of optical flow. With this approach, the number of frames, needed for regularization, is less than that of conventional methods. Moreover the parameters of object motion can be estimated with high accuracy using a regularized scene flow.

We plan to enhance the method to enable it to handle multiple moving objects.

Acknowledgement

This research was partly supported by a grant from the High-Tech Research Center Establishment Project from Ministry of Education, Culture, Sports, Science and Technology.

References

[1] B. D. Lucas and T. Kanade: "An iterative image registration technique with an application to stereo vision", *Imaging Understanding Workshop*, pp. 121-130, 1981.

[2] M. Fischer and R. Bolles: "Randoam sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM*, Vol. 24, No. 6, pp. 381-385, 1981.

[3] C. Tomasi and T. Kanade: "Shape and Motion from Image Streams under Orthography: a Factorization Method", *International Journal of Computer Vision*, Vol. 9, No. 2, pp. 137-154, 1992.

[4] J. L. Barron, D. J. Fleet, and S. S. Beauchemin: "Performance of Optical Flow Techniques", *International Journal of Computer Vision*, Vol. 12, No. 1, pp. 43-77, 1994.

[5] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade: "Three-Dimensional Scene Flow", *International Conference on Computer Vision*, Vol. 2, pp. 722 - 729, 1999.

[6] M. Irani: "Multi-Frame Optical Flow Estimation Using Subspace Constraints", *International Conference on Computer Vision*, Vol. 1, pp. 626 - 633. 1999.

[7] S. M. Seitz and C.R. Dyer: "Photorealistic scene reconstruction by Vvoxel Coloring", *International Journal of Computer Vision*, vol. 35, No. 2, pp. 151-173, 1999.

[8] J.Y. Bouguet: "Pyramidal Implementation of the Lucas Kanade Feature Tracker", In OpenCV Documentation, Microprocessor Research Labs, Intel Corp., 2000.

[9] R. Hartley and A. Zisserman: "Multiple View Geometry in Computer Vision", Cambridge University Press, 2000.

[10] S. M. Seitz and K. N. Kkutulakos: "Plenoptic Image Editing", *International Conference on Computer Vision*, vol. 48, No. 2, pp. 115-129, 2002.