Paper

Estimation of Three-dimensional Motion Information from Optical Flow Using Subspace Method

Non-member Yang Chunke (University of Tokushima)
Member Shunichiro Oe (University of Tokushima)
Member Kenji Terada (University of Tokushima)

Optical flow, a two-dimensional(2D) motion field on image plane, is essential for such tasks as the visual guidance of locomotion through the environment, the manipulation and recognition of objects. However, recovering three-dimensional(3D) motion information from optical flow, is a difficult problem because the relationship between the optical flow field and 3D motion parameters of the observer along with the depth of the environment, is nonlinear. In this paper, we propose a new method for estimating 3D motion information from optical flow. Considering an observer moving through a static environment, we intend to recover observer's 3D motion parameters and environment's relative depth map. Based on motion perspective, the estimation is carried out in three steps using three sets of equations derived from the nonlinear equation of motion perspective. First, direction of the translation components is recovered by searching a candidate over a discrete sampled space to minimize a residual function. Once the translation has been recovered, the rotation components of observer's 3D motion can be resolved from the second set of equations by using least square optimization. Finally, the estimation of relative depth map of the environment is straightforward using the third set of equations, given the recovered 3D motion parameters.

Keywords: motion analysis, 3D motion information, optical flow, subspace theory

1. Introduction

With the advent of advances in computer hardware, it is not only possible to carry out electronic processing on such simple data as characters or symbols, but also to build complex systems interacting with real world environments. These computer systems, which are generally known as *robots*, have a wide range of applications such as autonomous navigation or working under special environments (e.g., heights, underground, underwater, volcanoes, space, reactors). A fundamental issue for such systems is how to enable the robot to understand the 3D environment around. For a robot that only performs fixed operations under a fixed situation, a pre-inputed 3D map will be enough. However, this is just a simple mechanical instrument. A real robot should have flexibility that enables it to work under various situation. Therefore, it is necessary for a robot to discern objects and obstacles in the environment as well as its own 3D motion.

There are several ways for robots to measure 3D environment. Range finders measure distance by projecting ultrasound, laser, or pattern lights (e.g., slit, chess pattern, etc.) onto objects; tactile sensors measure positions and shapes by touching the objects. Especially, there is a widely applied method that measures 3D environment by analyzing image sequences obtained through video cameras. The method, which provides *vision* to robots, is called as *computer vision* or *robot vision*.

The main issue of motion image analysis, an important field of computer vision, is to estimate 3D motion information, such as motion parameters of objects and depth map of the environment, from a series of 2D motion image sequence. Such estimated 3D information can be further exploited for 3D shape recovery and 3D object recognition.

A typical approach for the above purpose is generally consisted of two main stages. The first stage is the measurement of 2D image motion field, which is also known as optical flow. The second stage is the recovery of 3D motion information in the real world from optical flow. As an observer moves with respect to a static environment, the 3D motion information need to be recovered includes 3D translation parameters of the observer, 3D rotation parameters of the observer, and distance from the observer to each point in the environment.

In this paper, we focused on the second stage and proposed a new method for estimating 3D motion information from optical flow. Considering a moving observer in a static environment, we intended to estimate 3D motion parameters of the observer and depth map of the environment. As will be explained later in this paper, it is impossible to determine translation components and depth in terms of absolute value, therefore, only the direction of translation and a relative depth map can be estimated.

Based on motion perspective, which projects 3D motion and structure to 2D optical flow, our method decomposed the nonlinear equation of motion perspective

into three sets of equations. The first set is a residual function in terms of only the direction of translation, image location on the image plane and optical flow, the direction of translation velocity can be solved for independently by searching a candidate, which can minimize the residual function, over a discretely sampled space of directions of translation. Once the direction of translation has been estimated, a large number of the second sets can be used to compute the rotation combining with least-square optimization. Finally, the relative depth of the environment can be estimated straightforward using the third set of equations, given the computed 3D motion parameters.

The rest parts of this paper are organized as follows. In section 2, we will review the relationship between 3D motion and optical flow, and derive equation of motion perspective that relates optical flow to 3D motion parameters and layout of the environment. In section 4, we present a method based on subspace theory to recover the translation without other unknowns being estimated. Once the translation has been determined, rotation and the environment's layout are estimated in section 5. Several experiments will be carried out in section 6 to verify the validity of the presented method. Finally, conclusion will be drawn in section 7.

2. 3D Motion and Optical Flow

Motion perspective, first termed by Gibson [2], projects 3D motion and structure of the environment onto 2D image plane. The result of this is optical flow, a 2D image velocity field. There are many examples of motion perspective in everyday life. For an instance, when a man looks out of the window of a running train. Objects close to the train will have high apparent velocities while objects far away will have low apparent velocities. The optical flow that results will consist of velocities of which the magnitudes are proportional to the distance from the train.

The usual approach based on motion perspective solves the reconstruction problem in two phases. At the first phase, 2D motion field, which is generally approximated by optical flow, is computed from an image sequence; At the second phase, 3D motion information, such as the motion parameters of moving observer and the depth map of the environment, is estimated using computed 2D motion field [1]. In the paper, we will focus on the problem of estimating 3D motion the observer.

The remainder of this section consists of two subsections. In section 2.1, we review the basic and well-known equation of the perspective transformation in 3D. In subsection 2.2, we explain the induction of the equation of motion perspective, which describes the relation between the 3D motion and structure and the corresponding optical flow, assuming a perspective projection.

2.1 Perspective Projection in 3D Suppose the optic axis of our camera lens is along a line parallel to the z-axis. To obtain the image frame coordinates for a given point in 3D space, we first translate this point to a 3D coordinate system centered at the lens of the camera. Then we translate along the z-axis by a distance f to the desired location of the projection image plane, and finally we take the perspective transformation.

We do this by using a homogeneous coordinate system that assumes an arbitrary position of the lens. Let (X, Y, Z) be the original coordinates of a point in 3D space. Let (X_0, Y_0, Z_0) be the position of the lens (called the *center of perspective*), and let (x, y) be the coordinates of the perspective projection of (X, Y, Z) on the image projection plane. Then $x = X'/t'$ and $y = Y'/t'$, where

$$\begin{pmatrix} X' \\ Y' \\ t' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -f \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ (Z - Z_0)/f \end{pmatrix} \quad \cdots\cdots\cdots\cdots (1)$$

Thus

$$x = f\frac{X - X_0}{Z - Z_0} \quad \text{and} \quad y = f\frac{Y - Y_0}{Z - Z_0} \quad \cdots\cdots (2)$$

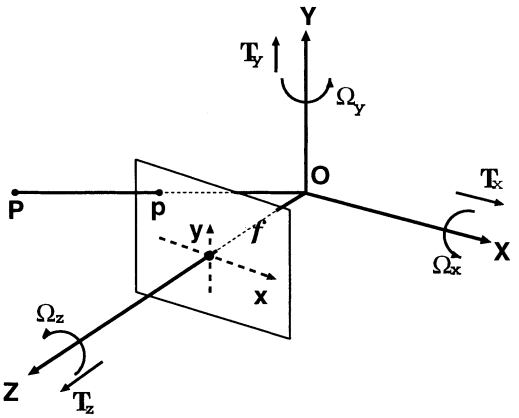


Fig. 1. A coordinate system (X, Y, Z) attached to the observer, and the corresponding image coordinates (x, y). The image position p is the perspective projection of the point P in the environment. $\vec{T} = (T_x, T_y, T_Z)$ and $\vec{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$ represent the translation and rotation of the observer.

2.2 motion perspective Let (X, Y, Z) represent a Cartesian coordinate system which is fixed with respect to the observer (Fig. 1), and let (x, y) represent a corresponding coordinate system of a planar image. The focal length, f, is assumed to be known. The observer's 3D coordinates (X_0, Y_0, Z_0) are $X_0 = 0, Y_0 =$

$0, Z_0 = 0$, thus the perspective projection (x, y) on the image of a point (X, Y, Z) in the environment is

$$x = \frac{fX}{Z}$$
$$y = \frac{fY}{Z} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (3)$$

The observer's 3D motion can be decomposed into two components: translation $\vec{T} = (T_x, T_y, T_z)$ and rotation $\vec{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$. Due to the motion of the observer, the relative motion of a surface point $P = (X, Y, Z)$ is:

$$\vec{V} = (\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt})^t = -(\vec{\Omega} \times \vec{P} + \vec{T})^t \quad \cdots\cdots (4)$$

Image velocity, $\vec{\theta}$, at the perspective projection $\vec{p} = (x, y)$ is defined as the derivatives of the x and y components with the respect to time. Taking derivatives of equation (3) with respect to time. and substituting from equation (4) gives:

$$\vec{\theta}(x, y) = \rho(x, y)\boldsymbol{A}(x, y)\vec{T} + \boldsymbol{B}(x, y)\vec{\Omega} \quad \cdots\cdots\cdots (5)$$

where $\rho(x, y) = \frac{1}{Z}$ is the inverse depth, and where:

$$\boldsymbol{A}(x, y) = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}$$

$$\boldsymbol{B}(x, y) = \begin{bmatrix} (xy)/f & -(f + x^2/f) & y \\ f + y^2/f & -(xy)/f & -x \end{bmatrix}.$$

Matrices $\boldsymbol{A}(x, y)$ and $\boldsymbol{B}(x, y)$ depend only on the image location, not on any of the unknowns.

Equation (5) describes the image velocity at each image location as a function of 3D motion and the inverse depth. An important observation about equation (5) is that it is bilinear; for a fixed ρ $\vec{\theta}$ is a linear function of $\vec{T}$ and $\vec{\omega}$, for a fixed $\vec{v}$ it is a linear function of ρ and $\vec{\Omega}$.

Since both $\rho(x, y)$(the inverse depth) and $\vec{T}$(the translation component of motion) are unknowns and since they are multiplied together in equation (5), they can each be determined only up to a scale factor; that is, only the direction of translation and the relative depth, not the absolute translation nor the absolute depth, can be solved for. For the rest of the paper, $\vec{T}$ denotes a unit vector translation direction($\| \vec{T} \|^2 = 1$) and $\rho(x, y)$ denotes the relative inverse depth.

3. Problem Statement

It is impossible to recover the 3D motion parameters, given the image velocity at only a single image location; there are six unknowns on the right-hand side of equation (5) and only two measurements (the two components of $\vec{\theta}(x, y)$ on the left-hand side. Generally, given the image velocity at N image locations, we have $2N$ equations for $N + 5$ unknowns, so image velocity measurements at 5 or more image locations are necessary to solve the problem.

For each of five image velocity vectors, a separate equation can be written in the form of equation (5). Following Heeger and Jepson [5], we collect these five equations into one matrix equation:

$$\vec{\Theta} = \boldsymbol{C}(\vec{T})\vec{q}, \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (6)$$

where $\vec{\Theta}$ (a 10-vector) is the image velocity at each of the five image locations, $\boldsymbol{C}(\vec{T})$ (a 10×8 matrix) is in terms of only the translational components of the observer's 3D motion, and $\vec{q}$ embodies the unknown inverse depths and the rotational components.

4. Estimating the Direction of Translation

We now present a method for estimating the observer's 3D translational velocity, $\vec{T}$. The depths and rotational velocity need not be known or estimated prior to solving for $\vec{T}$. We define a residual function, $\mathbf{R}(\vec{T})$, over the discretely sampled space of all candidate translation directions.

We use the residual function to assess how well each candidate translation accounts for the motion field. A residual value of zero for a particular candidate translation indicates that the motion field is consistent with that 3D motion.

The residual function, $\mathbf{R}(\vec{T})$, is defined to yield a least-squares estimate for translation, such that $\mathbf{R}(\vec{T}_0)$ is minimized for $(\vec{T}_0)$ equal to the actual translation. In the following parts of this section, we will show that the residual function can be defined as

$$\mathbf{R}(\vec{T}) = \| \vec{\Theta}^t \mathbf{C}^{\perp}(\vec{T}) \|^2. \quad \cdots\cdots\cdots\cdots\cdots\cdots (7)$$

$\mathbf{C}^{\perp}(\vec{T})$is a 10x2 matrix that is computed from $\mathbf{C}(\vec{T})$.

In equation (6) the matrix, $\boldsymbol{C}(\vec{T})$, divides $\vec{\theta}$-space into two subspace; the 8-dimensional subspace that is spanned by the columns of $\mathbf{C}(\vec{T})$, and the left over orthogonal 2-dimensional subspace. The columns of $\mathbf{C}(\vec{T})$ are guaranteed to span the full 8-dimensions for almost all choices of five sample locations and almost any $(\vec{T})$. The 8-dimensional subspace is called the range of $\mathbf{C}(\vec{T})$, and the 2-dimensional subspace is called the orthogonal complement of $\mathbf{C}(\vec{T})$.

Let $\mathbf{C}^{\perp}(\vec{T})$ be an orthonormal basis for the 2-dimensional orthogonal complement of $\mathbf{C}(\vec{T})$. It is straightforward, using techniques of numerical linear algebra(Strang [8]), to choose a $\mathbf{C}^{\perp}(\vec{T})$ matrix given $\mathbf{C}(\vec{T})$. The residual function, equation (7), is defined in terms of this basis for the orthogonal complement.

Given the image velocity, $\vec{\theta}$, and the correct translational velocity, $\vec{T}_0$, the following statement can be of validity:

$$\mathbf{R}(\vec{T}_0) = \| \vec{\Theta}^t \mathbf{C}^{\perp}(\vec{T}_0) \|^2 = 0.$$

Since $\vec{\Theta}$ is in the column space of $\mathbf{C}(\vec{T}_0)$ ($\vec{\Theta} = \mathbf{C}(\vec{T}_0)\vec{q}$, for some $\vec{q}$), and since $\mathbf{C}^{\perp}(\vec{T}_0)$ is orthogonal to $\mathbf{C}(\vec{T}_0)$, it is clear that $\mathbf{R}(\vec{T}_0) = 0$.

The residual function can be computed in parallel for each candidate $\vec{T}$, and residual surfaces can be computed in parallel for different sets of velocity vectors from different patches of the motion field. The resulting residual surfaces are then summed, to give a global least-squares estimate for $\vec{T}$.

5. Estimating the Rotation and Depth

Once $\vec{T}$ has been computed, we can solve for the rotational velocity as well. We now proceed to eliminate the depth, $\rho(x,y)$, from equation (5), leaving us with a linear constraint for $\vec{\Omega}$. To this end, we define a unit vector, $\vec{d}(x,y)$, perpendicular to the direction of translation.

$$\vec{d}^{\,t}(x,y)\boldsymbol{A}(x,y)\vec{T} = 0$$

$$\| \vec{d}(x,y) \|^2 = 1.$$

Multiplying by $\vec{d}^{\,t}(x,y)$ on the both side of the equation (5):

$$\vec{d}^{\,t}(x,y)\vec{\theta}(x,y) = \vec{d}^{\,t}(x,y)\boldsymbol{B}(x,y)\vec{\Omega}. \quad \cdots\cdots\cdots (8)$$

equation (8) is a linear constraint on the rotation, $\vec{\Omega}$, given the translation, $\vec{T}$, and a velocity vector, $\vec{\theta}(x,y)$. However, it is impossible to solve for the rotation given just one velocity vector, since $\vec{\Omega}$ has three unknown components and equation (8) provides only one constraint.

Several flow vector may be utilized in concert to solve for the rotation. If there is no error in the input motion field then two velocity vectors are sufficient. Using a large number of velocity vectors yields a least-squares estimate for $\vec{\Omega}$. For each velocity vector, we write an equation in the form of equation (8) (dropping the arguments x and y for simplicity):

$$\vec{d}_i^{\,t}\vec{\theta}_i = \vec{d}_i^{\,t}\boldsymbol{B}_i\vec{\Omega},$$

where i indexes over the velocity vectors at different image locations. The least squares solution for $\vec{\Omega}$ is obtained by minimizing:

$$\sum_i \| \vec{d}_i^{\,t}\boldsymbol{B}_i\vec{\Omega} - \vec{d}_i^{\,t}\vec{\theta}_i \|^2.$$

The estimate is given by:

$$\vec{\Omega} = \left(\sum_i \boldsymbol{B}_i^t \vec{d}_i \vec{d}_i^{\,t} \boldsymbol{B}_i \right)^{-1} \left(\sum_i \boldsymbol{B}_i^t \vec{d}_i \vec{d}_i^{\,t} \vec{\theta} \right) \quad \cdots\cdots (9)$$

The first factor on the right-hand side of equation (9) is a 3×3 matrix that does not depend on the input motion field, but it does depend on the recovered value of $\vec{T}$. The second factor on the right-hand side of equation (9) is a linear combination (a weighted sum) of the input image velocities. The coefficients in the weighted sum depend on the recovered value of $\vec{T}$. As new motion field measurements (and new estimates of $\vec{T}$) become available from incoming images, $\vec{\Omega}$ is computed as a linear combination of the image velocities.

Finally, once the direction of translation and rotation are both known, equation (5) provides two linear constraints to solve for the unknown relative distance at each image point:

$$\rho(x,y) = (\vec{\theta}(x,y) - \boldsymbol{B}(x,y)\vec{\Omega})(\boldsymbol{A}(x,y)\vec{T})^{-1}. \quad (10)$$

6. Experimental Results

In this section, we present several experiments for examining the efficacy of our proposed method.

6.1 various translation direction Optical flows were synthesized from a random depth map, given various translation directions. Error (E) between the estimated direction ($\vec{T}$) and the true direction ($\vec{T_0}$) is defined as:

$$E = \arccos\left(\frac{|\vec{T}||\vec{T_0}|}{\vec{T} \cdot \vec{T_0}} \right). \quad \cdots\cdots\cdots (11)$$

Figure 2 shows the error in the estimate of the translation direction as the function of the true translation direction. It can be seen from the figure that the proposed method is able to recover different direction of translation with relatively low error level.

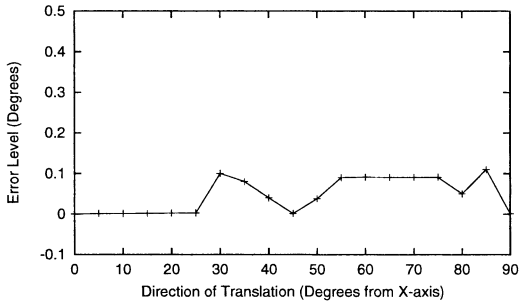


Fig. 2. Error level of estimated direction of translation. Focal length was 96 (in unit pixels). Depth from point to point varied randomly from 128 to 256 (in unit pixels). Entire image was 64×64. The translation direction varied from sideways, $\vec{T} = (1,0,0)$, to straight ahead, $\vec{T} = (0,0,1)$. Rotation was set to $\Omega = (0^\circ, 0^\circ, 0^\circ)$.

6.2 various rotation angles Optical flows were synthesized from a random depth map, given various rotation angles around Z-axis. Error (E) between the degrees of estimated rotation angles Ω) and those of the true angles (Ω_0) is defined as:

$$E = \arccos\left(\frac{|\Omega||\Omega_0|}{\Omega \cdot \Omega_0} \right). \quad \cdots\cdots\cdots (12)$$

Figure 3 shows the error level in the degrees of estimated rotation angles as the function of the true rotation angles. It can been seen from the figure that the error level is low over the estimation of small rotation angles, but becomes higher when the true rotation angles increase. The reason is that our method is based on an instantaneous approximation of motion perspective, where the optical flow vectors are defined as the derivatives, with respect to time, of the x- and y-components of the image point. Therefore, it is valid only if rotation angles are small enough, and usually introduces higher error levels when rotation angles are over the limit of the approximation. However, as can been seen from the figure, the error introduced by the instantaneous approximation is quite small if the rotation angle is less than 3 degrees per frame.

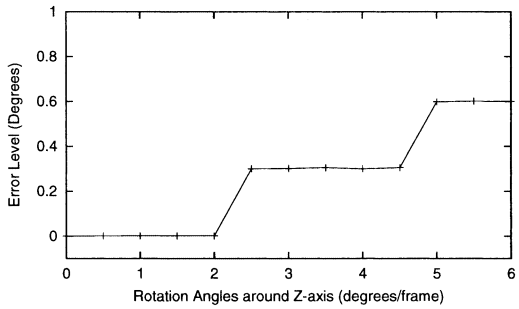


Fig. 3. Error level of estimated rotation angles. Depth from point to point varied randomly from 128 to 256 (in unit pixels). Entire image was 64×64. Rotation angles varied from $\vec{\Omega} = (0^\circ, 0^\circ, 0^\circ)$, $\vec{\Omega} = (0^\circ, 0^\circ, 0.5^\circ)$, ..., to $\vec{\Omega} = (0^\circ, 0^\circ, 6^\circ)$. Translation direction was set to $\vec{T} = (0, 0, 1)$.

6.3 recovery of simulated environment In this section, we tried to recover 3D information about a simulated environment using several synthesized optical flows. Figure 4 shows the simulated 3D environment, in this environment, two plates are separated by a ball and four columns, there is also an ellipse on one of the plate. Synthesized optical flows are shown with needle-grams in figure 5, the length of each needle represents the amplitude of the motion vector.

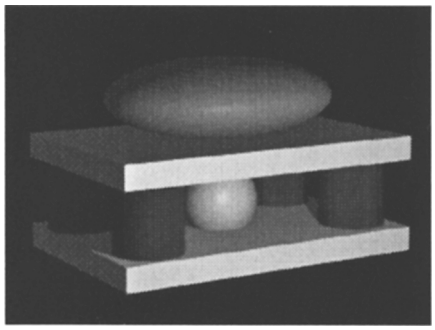
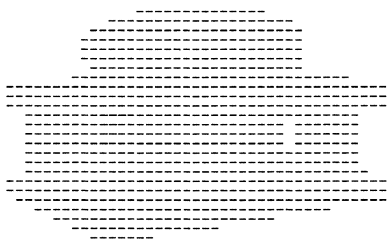


Fig. 4. Simulated 3D environment. 3D model was created using 3D modeler SCED [7]; Image and depth map were created using rendering program RAYSHADE [6].

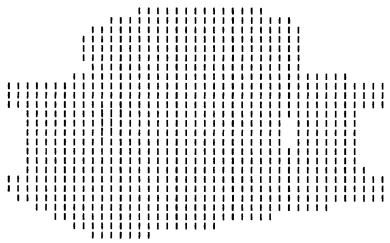
From these synthesized optical flows, we carried out the recovery of 3D motion information using the proposed method.

Table 1 shows the computed 3D motion parameters. It can been seen that 3D motion parameters are accurately recovered when the observer is under motion along X, Y-axis, or on the $X-Y$ plane, but small errors arise over the estimation for observer moving along Z-axis.

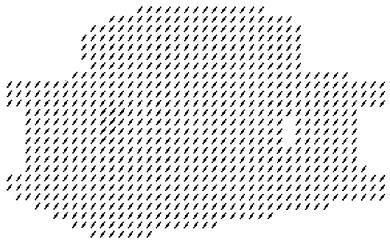
Depth maps of the environment, recovered from the estimated 3D motion parameters, are shown in figure 6. We notice that although the recovered depth maps are generally consistent with those of the simulated environment, significant errors exist along x- and y-axis in the case that the observer is moving along Z-axis, or forward. The reason is that in our method, the process



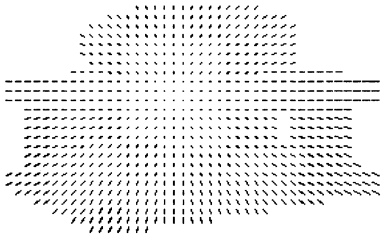
(a) $\vec{T} = \{1, 0, 0\}$



(b) $\vec{T} = \{0, 1, 0\}$



(c) $\vec{T} = \{0.7, 0.7, 0\}$



(d) $\vec{T} = \{0, 0, 0.1\}$

Fig. 5. Synthesized optical flows. Rotation was set to $\Omega = \{0, 0, 0\}$.

of recovering relative depth is based on the assumption that both x- and y-components of motion vector are valid at a given image location. However, in the case that the observer is moving forward, y-component of motion vectors are unavailable for image locations along x-axis, and vice versa. Therefore, the method tends to fail in recovering depth map along x- and y-axis when

the observer is moving forward only.

Table 1. Computed 3D motion parameters. T_x, T_y and T_z represent estimated 3D translation parameters; Ω_x, Ω_y and Ω_z represent estimated 3D rotation parameters.

	T_x	T_y	T_z	Ω_x	Ω_y	Ω_z
a	1.0	0.0	0.0	0.0	0.0	0.0
b	0.0	1.0	0.0	0.0	0.0	0.0
c	0.707	0.707	0.0	0.0	0.0	0.0
d	0.085	0.023	0.996	-0.0	0.001	-0.0

6.4 recovery of real scene Figure 7 shows several sample frames from a real image sequence. The image sequence was taken while a camera moving parallel to a cluster of trees on the ground.

This image sequence is known to be difficult to analyze because of the relatively poor resolution, the amount of occlusion, and the low contrast. Before the optical flows being computed, the sequence was smoothed using a three-dimensional Gaussian-filter. Then, optical flows were computed for the smoothed sequence using multiple gradient constraints method [9], and the computed optical flow is shown in figure 8.(a).

We continued to estimate 3D motion information about the real scene from the computed optical flows.

Table 2 shows the computed 3D motion parameters. Direction of translation was estimated with the sampling spaces' resolution being set to 0.015. To improve the speed of computation, we selected several key patches instead of all the patches over the whole image plane. The selection of key patches were determined by examining the numerical reliability of each patch in the image plane, only those patches with high numerical reliability are used for the estimation. A detailed explanation of numerical reliability examination can be found in our another paper [9].

Table 2. Computed 3D motion parameters

T_x	T_y	T_z	Ω_x	Ω_y	Ω_z
1.0	0.0	0.0	0.0	0.0	0.0

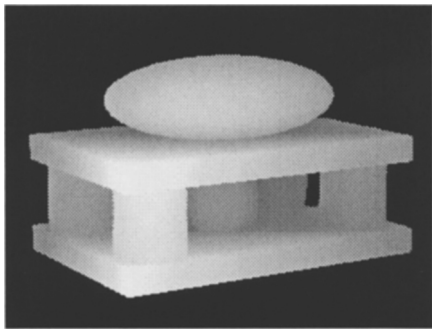
Figure 8.(b) shows the recovered depth map. We can see that the area of trees and ground which are close to the camera has the highest gray value, and the bushes behind has lower gray value.

It can thus be concluded that the recovered 3D motion parameters and depth map are consistent with that of the real scene.

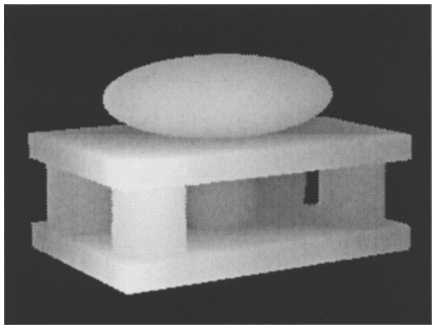
7. Conclusion

In this paper, based on motion perspective, we proposed a new method for estimating 3D motion information from 2D optical flow. Considering an observer moving through a static environment, we tried to recover observer's 3D motion parameters and environment's relative depth map.

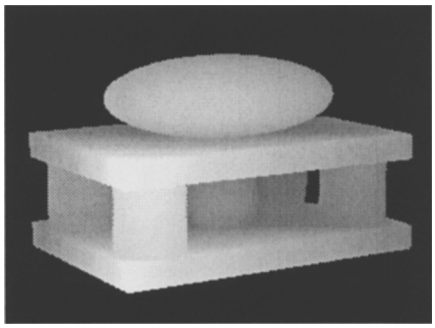
Our proposed method solved the recovery issue in three steps after decomposing the nonlinear equation of motion perspective into three sets of linear equations. Firstly, a residual function was derived from the first set of equations using subspace theory, and direction of translation was then recovered by searching a candidate over a discrete sampled space, which would minimize the residual function. Secondly, rotation was estimated from the second set of equations by using least square optimization. Finally, relative depth map of the envi-



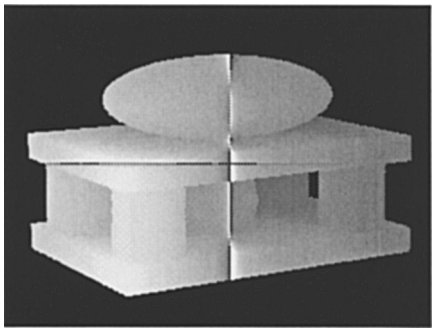
a: $\vec{T} = \{1, 0, 0\}$



b: $\vec{T} = \{0, 1, 0\}$



c: $\vec{T} = \{1, 1, 0\}$



d: $\vec{T} = \{0, 0, 1\}$

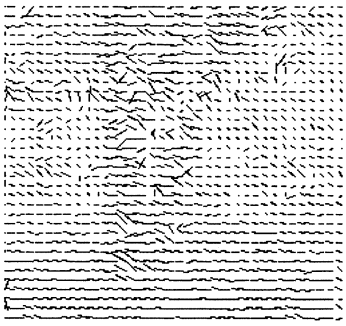
Fig. 6. Recovered depth map. Gray-scale level at each image point represents the inverse relative depth from the corresponding 3D location to the observer.

frame 01

frame 06

frame 19

Fig. 7. Real image sequence. The image size was 256×230.



(a) computed optical flow

(b) recovered depth map

Fig. 8. Computed optical flow and the recovered depth map for the real scene.

ronment was recovered using the third set of equations, given the recovered 3D motion parameters.

To verify the efficacy of the new method, we carried out several experiments both on simulated and real scene.

Experiment 6.1 and 6.2 were intended to verify the validity of our method to various 3D motion components. it could be seen from the results that our method was able to provide measurements of various 3D translation and rotation with relatively low error level. Although significant errors arise for large rotation angles because of instantaneous approximation of motion perspective, the error introduced by the instantaneous approximation is quite small if the rotation angle is less than 3 degrees per frame.

Experiment 6.3 was to examine our method's efficacy of recovering relative depth information from synthetic optical flows. We can see from the results (figures 6.(a), 6.(b), and 6.(c)) that our method could provide fairly accurate recovery of the simulated depth when the camera was under simple translations.

In experiment 6.4, we tried to recover 3D motion information about a real image scene. As can be seen in table 2 and 8.(b), given relatively accurate optical flows, our method could provide valid measurements of camera's 3D motion parameters and environment's relative depth map.

However, there are still some problems in our method need to be considered. At the phase of recovering translation components, there exists a searching process over a discrete sampling space. The searching process will be very computational costing when the resolution of the sampling space is high. And the computational accuracy depends on sampling space's resolution. The tradeoff between computation cost and accuracy will be an obstacle to implementing our method to real-time systems. Another problem is that in our method, the depth recovery step is very sensible to errors in estimated 3D motion components. As shown in figure 6.d, although the error level of estimated 3D motion components was very low, there still existed several apparent errors in recovered depth map from the optical flow synthesized while the camera was moving straight forward.

Our future work will involve in several issues. To implement our method to real-time systems, a substitute of the searching process at the phase of estimating translation components will be necessary. The robustness of recovering depth map need to be improved so that our method will be adaptive to more complex situation. We are also planning to extending our method to deal with scenes in which there are multiple motions.

Acknowledgment

The authors are grateful the reviewers for their constructive comments and advice. The first author would also like to thank Prof. Oe for his long-term guidance and kind help.

(Manuscript received April 27, 2000, revised February 21, 2001)

References

(1) J. Barron, "A Survey of Approaches for Determining Optic Flow, Environment Layout and Egomotion", *Technical Report(Dept. of Computing Science, University of Toronto)* RBCV-TR-84-5, 1984.

(2) J.J. Gibson, *The Perception of the Visual World*, Houghton Mifflin, Boston, 1950.

(3) R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision, VOLUME I*, Addison-Wesley Publishing Company, Inc., 1993.

(4) R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision, VOLUME II*, Addison-Wesley Publishing Company, Inc., 1993.

(5) D.J. Heeger and A.D. Jepson, "Subspace Methods for Recovering Rigid Motion I : Algorithm and Implementation", *Technical Report(Dept. of Computing Science, University of Toronto)* RBCV-TR-90-35, 1990.

(6) RAYSHADE - *a System for Generating Ray-traced Images*, http://www-graphics.stanford.edu/c̃ek/rayshade

(7) SCED - *a Constraints Based Scene Designer*, http://http.cs.berkeley.edu/s̃chenney/sced/sced.html

(8) G. Strang, *Linear Algebra and its Applications*, Academic Press, New York, 1980.

(9) C. Yang and S. Oe, *A New Multiple Gradient Constraints Method to Compute Optical Flow*, The Journal of the Institute of Image Electronics Engineers of Japan, vol. 28 no.4 pp. 387, 1999.

Yang Chunke (Non-member) received the B.Eng. degree in computer science from Sichuan University, China, in 1993, and the M.Eng. degree and Ph.D. in information science from University of Tokushima in 1997 and 2000, respectively. His main research interests are in computer vision and motion image analysis. He is a member of IEEE and IEICE.

Shunichiro Oe (Member) was born in Shiga Prefecture in 1943. He received the B.Eng. and M.Eng. degree from Tokushima University in 1967 and 1969, respectively, and the Ph.D. from the University of Osaka Prefecture in 1980. From 1969 to 1974 he was research assistant at computer center of the University of Tokushima, from 1974 to 1995 he was a lecturer and associate professor at the Department of Information Science and Intelligent Systems, Faculty of Engineering, the University of Tokushima, and from 1995 he has been a professor at the same department. His current research interests include time series analysis, pattern recognition, neural networks, genetic algorithms, and image processing, especially texture segmentation, industrial image processing, three dimensional image processing and remote sensing. He is a member of IEICE, SICE and etc.

Kenji Terada (Member) received the Ph.D degree in Electrical Engineering from Keio University in 1995. In 1995, he joined the Faculty of Engineering, the University of Tokushima, where he is now an Associate Professor. His research areas include image processing, computer vision and image sensing. He is a member of IEEE, SICE of Japan, IEICE of Japan, ISCIE of Japan.