MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

3-D SHAPE RECONSTRUCTION FROM CAMERA MOTION WITH INEXACT MOTION PARAMETERS

Keisuke KINOSHITA and Koichiro DEGUCHI
Dept. of Mathematical Engineering and Information Physics,
Faculty of Engineering, University of Tokyo

7-3-1, Hongo, Bunkyo-ku, Tokyo 113, Japan

Abstract

When a sequence of images are obtained by a moving camera, if the exact camera velocity and corresponding points on the images are determined, the 3-D shape of the object can be reconstructed using stereo method. However, in general, it is difficult to determine both of them exactly. In this paper, the reconstruction of 3-D shape, using the sequence of images of the object is described, for the case when the camera velocity cannot be known exactly.

First, we calibrate the camera position and camera velocity at a time from image points whose 3-D positions are already calculated exactly using previous images in the sequence, then, using this camera velocity and the positions of corresponding points on the image whose 3-D positions are not determined yet, we calculate their 3-D exact positions.

Repeating these two phases on the sequence of images, we reconstruct the 3-D position of the object. In the later phase, to reduce the effect of quantization errors, Kalman filtering method is introduced.

Introduction

Recently, 3-D shape recognition from images obtained by moving camera is studied intensively [1][2]. But the calibration of cameras or searching corresponding points is not an easy problem. When a camera is installed on a moving robot, it is hard to determine the camera velocity or camera position exactly from a sensor on the robot. In such cases, we have to determine the camera velocity only from a sequence of images which are taken from the camera.

Here, we propose the method to reconstruct 3-D shape of objects from such a sequence of images by iterative uses of next two phases.

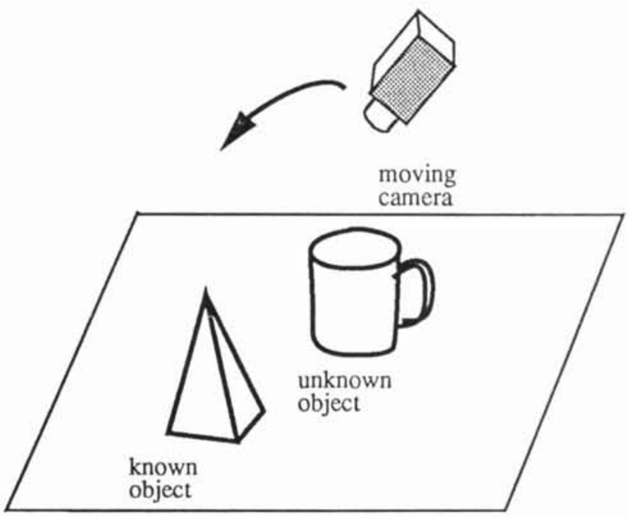

Figure 1: Moving camera and objects whose 3-D positions are already known and yet unknown in its view.

- Phase 1
  Determine the camera velocity from points whose 3-D positions are already given or calculated exactly using previous images in the sequence.

- Phase 2
  From the camera velocity obtained by phase 1 and the point correspondences in sequential two images, calculate the exact 3-D positions of points which are not determined yet.

(Figure 1, Figure 2).

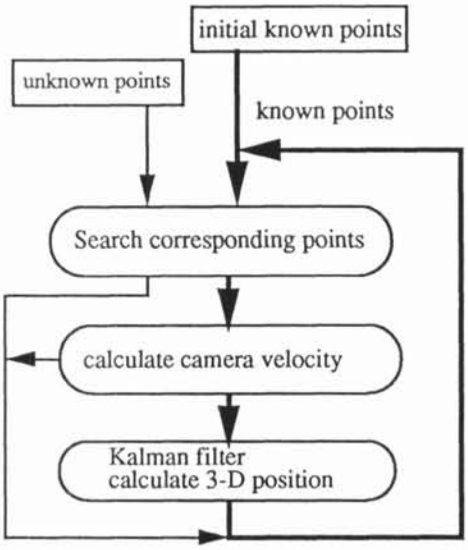

Figure 2: Flow of procedures to reconstruct 3-D position

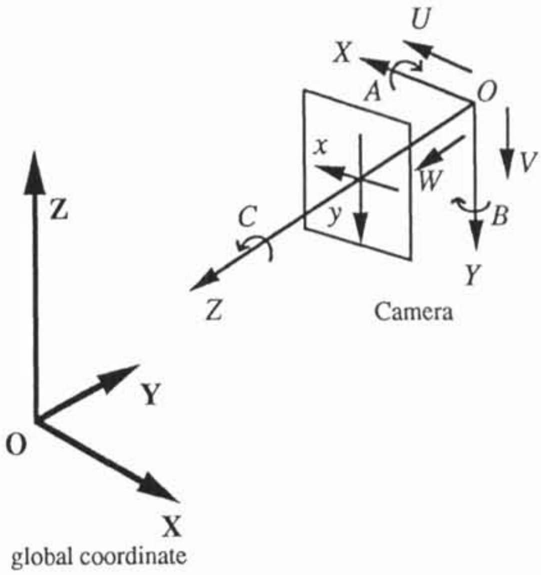

Figure 3: Camera coordinates and camera velocity

Determining Camera Velocity

To determine the camera velocity, corresponding points on the sequence of images are used. They can be determined by taking correlation between the images, assuming that the image positions don't change not so much when the time interval of the images is small.

In the followings, we use a coordinate system X, Y, Z fixed to a camera, to formalize the relation between the camera velocity and image correspondences.

Let rotational components of the camera velocity be A, B, C, and its translational components be U, V, W, respectively (Figure 3).

And let the 3-D position of point i in camera coordinate at time k be (X_{ik}, Y_{ik}, Z_{ik}), and whose position projected on the image are denoted by (x_{ik}, y_{ik}).

When camera moves, 3-D position of the point in the camera coordinate system changes from the time $k-1$ to k as,

$$\begin{aligned} X_{ik} &= X_{ik-1} - U_{k-1} - B_{k-1}Z_{ik-1} + C_{k-1}Y_{ik-1} \\ Y_{ik} &= Y_{ik-1} - V_{k-1} - C_{k-1}X_{ik-1} + A_{k-1}Z_{ik-1} \quad (1) \\ Z_{ik} &= Z_{ik-1} - W_{k-1} - A_{k-1}Y_{ik-1} + B_{k-1}X_{ik-1} \end{aligned}$$

These 3-D positions produce image positions of the points as,

$$\begin{aligned} x_{ik-1} &= f\frac{X_{ik-1}}{Z_{ik-1}} \quad & y_{ik-1} &= f\frac{Y_{ik-1}}{Z_{ik-1}} \\ x_{ik} &= f\frac{X_{ik}}{Z_{ik}} \quad & y_{ik} &= f\frac{Y_{ik}}{Z_{ik}} \quad (2) \end{aligned}$$

where, f is the focal length of the camera.

Substituting (1) to (2), using a notation $\mathbf{c}_k = (A_k B_k C_k U_k V_k W_k)^T$, linear simultaneous equations(3)(top of next page) for determining the camera velocity is obtained. We represent this form (3),

$$\mathbf{Q}_k \mathbf{c}_k = \mathbf{n}_k \quad (4)$$

By this equation, camera velocity can be determined from 3-D positions of three, as the minimum, points and their projected positions. But those calculated by a small number of points include much errors, and most of which are due to quantization errors on images. So we use much more points than three and employ the least square error method. In the method, the camera velocity $\mathbf{c}_k$ can be solved as follows.

$$\mathbf{c}_k = (\mathbf{Q}_k^T \mathbf{Q}_k)^{-1} \mathbf{Q}_k^T \mathbf{n}_k$$

When much more than three points are employed,the accuracy of $\mathbf{c}_k$ is expected to be improved.

Reconstruction of 3-D Shape

Once camera velocity is determined using its image correspondence in a sequence of images, the 3-D position of a point in camera coordinate system is determined as equation (5)(top of next page).

Then, the points whose 3-D positions were unknown at the time $k-1$ can be used in turn at

$$
\begin{pmatrix}
\vdots & & & & & \vdots \\
-x_{ik}Y_{ik-1} & x_{ik}X_{ik-1}+fZ_{ik-1} & -fY_{ik-1} & f & 0 & -x_{ik} \\
-y_{ik}Y_{ik-1}-fZ_{ik-1} & y_{ik}X_{ik-1} & fX_{ik-1} & 0 & f & -y_{ik} \\
\vdots & & & & & \vdots
\end{pmatrix} \mathbf{c} = \begin{pmatrix} \vdots \\ fX_{ik-1}-x_{ik}Z_{ik-1} \\ fY_{ik-1}-y_{ik}Z_{ik-1} \\ \vdots \end{pmatrix} \tag{3}
$$

$$
Z_{ik} = \frac{f x_{ik} W_{k-1} - f^2 U_{k-1}}{x_{ik}(f - A_{ik-1} y_{ik-1} + B_{k-1} x_{ik-1}) - f x_{ik-1} - C_{k-1} f y_{ik-1} + B_{k-1} f^2}
$$

or,

$$
Z_{ik} = \frac{f y_{ik} W_{k-1} - f^2 V_{k-1}}{y_{ik}(f - A_{ik-1} y_{ik-1} + B_{k-1} x_{ik-1}) - f y_{ik-1} - C_{k-1} f x_{ik-1} - A_{k-1} f^2} \tag{5}
$$

---

time k as the points whose positions are already known.

Reducing Quantization Errors

As is mentioned above, we have the effects of quantization errors of the image. The effects on the camera velocity determination is reduced by calculating with much more than three points just described.

But the effect on the 3-D position determination of points are still remained. We reduce these errors by using Kalman Filtering method [3][4].

Taking state vector $\mathbf{x}_{ik}$ as the 3-D position of the point i at time k, and taking its projected point on the image as $\mathbf{y}_{ik}$, the camera movements are expressed as,

$$
\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{w}_{k-1} \tag{6}
$$

(point number i is omitted here).

As the perspective transformation (2) is not a linear system, we approximate it as a linear form using the Taylor expansion. Then representing the quantization error by $\mathbf{v}_k$, we obtain the observation equation as,

$$
\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{7}
$$

where

$$
\begin{aligned}
\mathbf{F}_k &= \begin{pmatrix} 1 & -C_{k-1} & B_{k-1} \\ C_{k-1} & 1 & -A_{k-1} \\ -B_{k-1} & A_{k-1} & 1 \end{pmatrix} \\
\mathbf{G}_k &= \frac{\partial \mathbf{x}_k}{\partial(A,B,C,U,V,W)} \\
&= \begin{pmatrix} 0 & Z_{k-1} & -Y_{k-1} & -1 & 0 & 0 \\ -Z_{k-1} & 0 & X_{k-1} & 0 & -1 & 0 \\ Y_{k-1} & -X_{k-1} & 0 & 0 & 0 & -1 \end{pmatrix} \\
\mathbf{H}_k &= \begin{pmatrix} \frac{f}{Z_{k-1}} & 0 & -f\frac{X_{k-1}}{Z_{k-1}^2} \\ 0 & \frac{f}{Z_{k-1}} & -f\frac{Y_{k-1}}{Z_{k-1}^2} \end{pmatrix}
\end{aligned}
$$

The means and variances of the vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ are given as,

$$
\begin{aligned}
\bar{\mathbf{w}}_k &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ U_k \\ V_k \\ W_k \end{pmatrix} \\
\mathbf{W}_k &= Var(\mathbf{w}_k) \\
\bar{\mathbf{v}}_k &= \begin{pmatrix} f\frac{X_{k-1}}{Z_{k-1}} \\ f\frac{Y_{k-1}}{Z_{k-1}} \end{pmatrix} \\
\mathbf{V}_k &= Var(\mathbf{v}_k) = \begin{pmatrix} {\sigma_x}^2 & 0 \\ 0 & {\sigma_y}^2 \end{pmatrix}
\end{aligned}
$$

To reduce the effect of quantization errors, the Kalman filtering is adopted to this system as followings.

$\tilde{\mathbf{x}}$: state prediction

$\hat{\mathbf{x}}$: state update

- Prediction Phase is

$$
\tilde{\mathbf{x}}_k = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{G}_{k-1}\bar{\mathbf{w}}_{k-1} \ ,
$$

the variance of state prediction $\tilde{\mathbf{x}}_k$ is

$$
\mathbf{M}_k = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{G}_{k-1}\mathbf{W}_k\mathbf{G}_{k-1}^T \ ,
$$

the variance of state update $\hat{\mathbf{x}}_k$ is

$$
\mathbf{P}_k = ({\mathbf{M}_k}^{-1} + \mathbf{H}^T\mathbf{V}_k\mathbf{H})^{-1}
$$

- Update Phase is

$$
\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{P}_k{\mathbf{H}_k}^T\mathbf{V}_k^{-1}(\mathbf{y}_k - (\mathbf{H}_k\tilde{\mathbf{x}}_k + \bar{\mathbf{v}}_k)) \ .
$$

Starting initial value $\tilde{\mathbf{x}}_0$ and $\mathbf{M}_0$, $\hat{\mathbf{x}}$ is calculated by repeating the prediction phase and the update phase. In these equations, $\mathbf{W}_k$ and $\mathbf{V}_k$ are the variance of camera velocity and quantization errors, respectively, then,

$$
\begin{aligned}
\mathbf{W}_k &= Var(\mathbf{c}_k) \\
&= (\mathbf{Q}_k^T\mathbf{Q}_k)^{-1}\mathbf{Q}_k^T Var(\mathbf{n}_k)\mathbf{Q}_k((\mathbf{Q}_k^T\mathbf{Q}_k)^{-1})^T)
\end{aligned}
$$

And, the quantization errors of x_k and y_k can be assumed to distribute uniformly, the variance of $\mathbf{n}_k$ becomes,

$$Var(\mathbf{n}_k) = diag(\cdots, Z_{ik}^2/12, Z_{ik}^2/12, \cdots)$$

Using this, $\mathbf{V}_k$ becomes,

$$\mathbf{V}_k = \begin{pmatrix} 1/12 & 0 \\ 0 & 1/12 \end{pmatrix}$$

Beside this value, the observation error $\mathbf{v}_k$ must include linearization error to represent the perspective transformation as the linear form (7). To compensate this error, this variance $\mathbf{V}_k$ must be multiplied by severals.

Experimental Results

A simple case was simulated as the basic experiments. In the simulation, 128 points are used for the initial points whose 3-D positions are known, and 128 more points are used as unknown object points. The points are located randomly inside the cube at the origin in the global coordinates whose edges are 125mm. The image simulated has 512 x 512 pixels. The true camera velocity is fixed to $(0.0, 0.0, 0.0, 0.0, 0.0, -25.0)$, the initial camera position is $(0.0, 0.0, 300.0)$ and the initial camera angle is $(0.1, 3.0, 0.1)$ in the global coordinates (Figure 4).

The camera parameters obtained for every time in the sequence of 16 images are shown (Figure 5). It is shown that camera parameters are reconstructed almost exactly. The 3-D positions of points were also reconstructed, but the depth errors are slightly larger than those expected.

Conclusion

The method to reconstruct the 3-D shape of an object when camera velocity is inexact or unknown was proposed.

In the Kalman filtering phase, we approximated the observation equation by linear form. The experimental results shows this error could not completely compensated with the proposed method. We have to investigate further about this problem.

Acknowledgments

We wish to thank Prof. Iwao Morishita for his instructive suggestions.

References

[1] A. Bandyopadhyay and J.Aloimonos : "Active Vision", Int. J. Computer Vision, Vol. 1, pp. 333/356, 1988.

[2] G. Sandini and M. Tistarelli : "Active Tracking Strategy for Monocular Depth Inference Over Multiple Frames", IEEE Trans. **PAMI**, Vol. 12, pp. 12/27, 1990.

[3] N.Matsuda and Y.Ohta : "Correspondence Search on a Sequence of Stereo Images", IPSJ SIG Reports, **CV** 59-2, 1989.

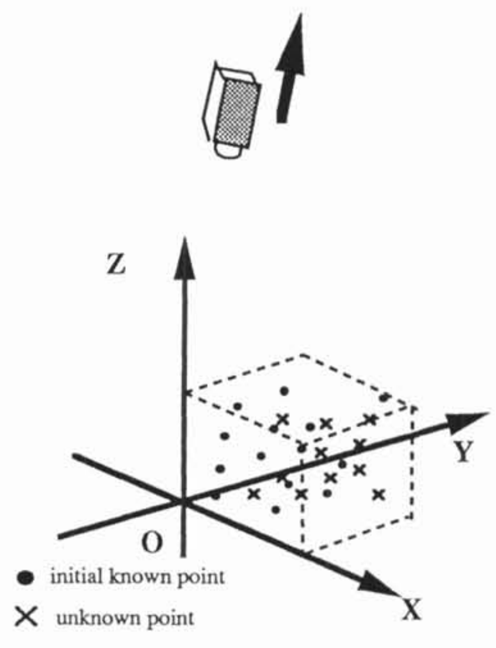

Figure 4: Sample points used in the experiment

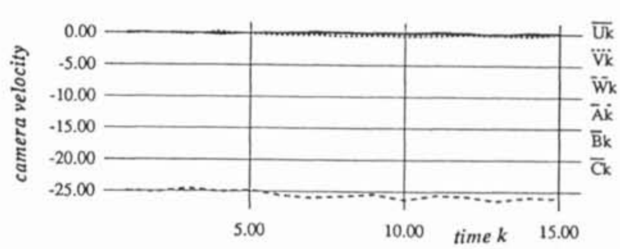

Figure 5: Obtained camera velocity