FCV2004 ; The 10th Korea-Japan Joint Workshop on Frontiers of Computer Vision, pp.269-274 (2004)

Measuring Ball Spin by Image Registration

Toru Tamaki†, Takahiko Sugino‡, Masanobu Yamamoto‡
†: Graduate School of Science and Technology, Niigata University, JAPAN
‡: Dept. of Information Engineering, Fac. of Engineering, Niigata University, JAPAN
tamaki@ie.niigata-u.ac.jp

Abstract We propose a method for measuring a spin of a ball accurately by the image registration with a 3D model. The proposed method estimates the parameters of the transformation between two images containing a known shape object with depth information. The Gauss-Newton method is used to minimize the residuals of intensities of the two images, and the rotation and the translation are estimated. Experimental results using real images of a spinning ball demonstrate the robustness of the proposed method.

1 Introduction

Estimating the parameters of the transformation between images is an important process for computer vision. The transformation is caused by camera motion or movement of an object in a scene. Methods for estimating the parameters have been developed for stereo vision[1], motion estimation[2], and shape from motion[3].

Usually, such methods require a number of correspondences between a point in one image and the corresponding point in the other image, to estimate the parameters of the transformation between the corresponding points (for example, corners or edge intersections). The correspondences are established by manual operations or the marker detection, however, it is difficult to reconstruct a smooth curved surface from patches surrounded by the points.

Another way is the image registration technique[4] developed for image mosaicing[5] or motion estimation[2]. The image registration estimates the parameters of the transformation between two images by minimizing the difference in intensities, that is, the sum of the squares of the intensity residuals between the two images. Therefore, the image registration doesn't require to establish the point-to-point correspondence.

Registration methods for recovering arbitrary shape were proposed by adding extra term to the planar model[4], or by decomposing a scene into piecewise planar objects[2]. This can recover 3D depth map of a scene containing unknown shape object, however, it can not estimate the accurate motion parameters including rotation and translation.

But, there are some cases where we know the shape of the object, such as camera calibration using a cube[6], or motion tracking of a spinning ball. In such cases the problem can be solved by different way. Our approach is to use information of the shape of a known object to improve the estimation.

In this paper, we propose a method to measure a ball spin by using the image registration method for estimating motion parameters of a known shape object using depth information of the first image in order to register two images. We will show experiments of motion estimation using two successive frames with manually fitted model to generate depth information.

In sectoin 2, we explain models of the transformation between two images using depth information. We describe the algorithm of registration based on a nonlinear optimization in section 3. Experimental results for motion estimation of the proposed method is shown in section 4.

2 Registration between two frames

In this section, we describe the model of the transformation between two images using depth information.

As shown in Fig.1, given the first image I_1 the depth of a known object (a 3D model) is produced by known rotation R and translation $\boldsymbol{T}$. The object in the second image is slightly moved with unknown motion Q and $\boldsymbol{S}$.

2.1 Modeling the transformations

At first, we have a first image I_1 of the object with known shape. A point $\boldsymbol{P}_0$ on the object is represented with respect to the origin of the coordinate system (Fig.1(a)). Note that the transformations use the same coordinate system throughout the formulation. Then $\boldsymbol{P}_0$ is moved to $\boldsymbol{P}_1$ by the rotation matrix R and the translation vector $\boldsymbol{T}$ as

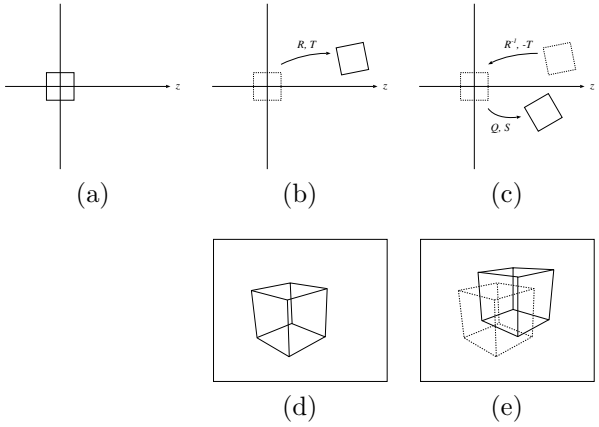


Fig. 1: Outline of the transformations. (a) The object (b) Moved object at the first image. (c) Moved object at the second image. (d) The first image I_1 of the object. (e) The second image I_2. The object at the first image is drawn in dotted line.

follows.

$$\boldsymbol{P}_1 = R\boldsymbol{P}_0 + \boldsymbol{T} \tag{1}$$

$$\begin{aligned} R &= R_z(a)R_y(b)R_x(c) \\ &= \begin{pmatrix} \cos a & -\sin a & 0 \\ \sin a & \cos a & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos b & 0 & \sin b \\ 0 & 1 & 0 \\ -\sin b & 0 & \cos b \end{pmatrix} \cdot \\ &\quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos c & -\sin c \\ 0 & \sin c & \cos c \end{pmatrix} \end{aligned} \tag{2}$$

$$\boldsymbol{T} = (t_x, t_y, t_z)^T \tag{3}$$

Then the point $\boldsymbol{P}_1 = (X_1, Y_1, Z_1)^T$ is projected to $\boldsymbol{p}_1 = (x_1, y_1)^T$ on the image I_1 (Fig.1(d)).

$$\boldsymbol{p}_1 = \left(f\frac{X_1}{Z_1}, f\frac{Y_1}{Z_1} \right)^T \tag{4}$$

where f is the focal length that is given as well as R and $\boldsymbol{T}$. Note that the depth Z_1 is known because the object is represented as a 3D model.

Then we consider on the second image I_2. We assume that I_2 is similar to but slightly different from I_1 (Fig.1(e)). The coordinates of the point $\boldsymbol{P}_2 = (X_2, Y_2, Z_2)^T$ is also transformed from $\boldsymbol{P}_0$ by the rotation matrix Q and the translation $\boldsymbol{S}$ as follows;

$$\boldsymbol{P}_2 = Q\boldsymbol{P}_0 + \boldsymbol{S} \tag{5}$$

$$Q = Q_z(\alpha)Q_y(\beta)Q_x(\gamma) \tag{6}$$

$$\boldsymbol{S} = (s_x, s_y, s_z)^T \tag{7}$$

then $\boldsymbol{P}_2$ is projected to $\boldsymbol{p}_2 = (x_2, y_2)^T$ on the image I_2.

$$\boldsymbol{p}_2 = \left(f\frac{X_2}{Z_2}, f\frac{Y_2}{Z_2} \right)^T \tag{8}$$

2.2 The transformation between two images

Figure 1(c) shows the transformation between $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ through $\boldsymbol{P}_0$. From $\boldsymbol{p}_1$ in the first image I_1 and the depth Z_1 (the depth is provided when I_1 is created), three dimensional coordinates $\boldsymbol{P}_1$ is calculated.

$$\boldsymbol{P}_1 = \left(\frac{x_1 Z_1}{f}, \frac{y_1 Z_1}{f}, Z_1 \right)^T \tag{9}$$

Then $\boldsymbol{P}_1$ is moved back to $\boldsymbol{P}_0$, and further moved to $\boldsymbol{P}_2$.

$$\boldsymbol{P}_2 = Q\boldsymbol{P}_0 + \boldsymbol{S} = QR^{-1}(\boldsymbol{P}_1 - \boldsymbol{T}) + \boldsymbol{S} \tag{10}$$

then $\boldsymbol{P}_2$ is projected to $\boldsymbol{p}_2$. Therefore, $\boldsymbol{p}_1$ in I_1 corresponds to $\boldsymbol{p}_2$ in I_2 through given $R, \boldsymbol{T}, f, Z_1$ and unknown $Q, \boldsymbol{S}$.

We estimate the unknown parameters by minimizing the difference between the intensity of $\boldsymbol{p}_1$ in I_1 and that of $\boldsymbol{p}_2$ in I_2.

3 Estimating parameters

In this section, we describe how to estimate the parameters of Q and $\boldsymbol{S}$. Image registration[6] seeks to minimize the residuals r_i of intensities of the two images, I_1 and I_2.

$$r_i = I_1(\boldsymbol{p}_{1i}) - I_2(\boldsymbol{p}_{2i}) \tag{11}$$

where $I_1(\boldsymbol{p}_1)$ is the intensity at the point $\boldsymbol{p}_1$ in the image I_1, and $I_2(\boldsymbol{p}_2)$ is the intensity at the point $\boldsymbol{p}_2$ in the image I_2. The function to be minimized is the sum of squares of the residuals over the image I_1.

$$\min_{\boldsymbol{\theta}} \sum_i {r_i}^2 \tag{12}$$

where $\boldsymbol{\theta} = (\alpha, \beta, \gamma, s_x, s_y, s_z)^T \equiv (\theta_1, \ldots, \theta_6)^T$ and the summation i is taken over the points of the object region in I_1 and is visible in I_2 (see section 3.2).

Estimating the parameters $\boldsymbol{\theta}$, the objective function is minimized by the Gauss-Newton method, a nonlinear optimization technique[7]. The parameters are updated with some initial value by the following rule: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\ \delta\boldsymbol{\theta}$. The decent direction $\delta\boldsymbol{\theta} = (\delta\theta_1, \ldots, \delta\theta_6)^T$ is calculated as follows[7]:

$$\delta\boldsymbol{\theta} = -(J^T J)^{-1} J^T \boldsymbol{r}, \quad J = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\theta}} \tag{13}$$

where $\boldsymbol{r} = (r_1, r_2, \ldots)^T$. This is the same as the least square formulation, that is, the system of linear equations[2] written as

$$\sum_i \sum_l \frac{\partial r_i}{\partial \theta_k} \frac{\partial r_i}{\partial \theta_l} \delta\theta_l = -\sum_i r_i \frac{\partial r_i}{\partial \theta_k} \tag{14}$$

for $k = 1, \ldots, 6$. The partial derivatives are derived by the chain rule of vector differentiation[7].

$$\frac{\partial r}{\partial \theta_k} = -\frac{\partial \boldsymbol{p}_2}{\partial \theta_k}\frac{\partial I_2}{\partial \boldsymbol{p}_2} = -\left(\frac{\partial x_2}{\partial \theta_k}, \frac{\partial y_2}{\partial \theta_k}\right)^T \nabla I_2(\boldsymbol{p}_2) \tag{15}$$

Once the direction is decided by solving the system of equations in Eq.(14), the step length α is optimized by line minimization[8]. The parameter update is repeated until it converges. At each iteration, the parameters estimated in the previous iteration are used for the current Jacobian.

3.1 Initial state

At the beginning of the iteration, we use the following initial value for each parameters: $Q = R$, $\boldsymbol{S} = \boldsymbol{T}$, because we assumes that the difference between the first and the second images is small.

3.2 Visible test

To remove a point which is not visible in I_2, we perform the following visible test. Let $\boldsymbol{P}$ be a visible point on the objet in I_1, a normal vector $\boldsymbol{N}$ of the surface at $\boldsymbol{P}$ is given by[9] $\boldsymbol{N} = \dfrac{\partial \boldsymbol{P}}{\partial x} \times \dfrac{\partial \boldsymbol{P}}{\partial y}$. Assuming that the camera center is identical to the origin, the angle between the normal and the viewing direction is given by $\cos^{-1}\left(\dfrac{|\boldsymbol{N}\cdot\boldsymbol{P}|}{|\boldsymbol{N}||\boldsymbol{P}|}\right)$. If the angle is larger than $\pm\frac{\pi}{4}$, then the point is excluded from the calculation of Eq.(12) and Eq.(14).

3.3 Depth from the z buffer

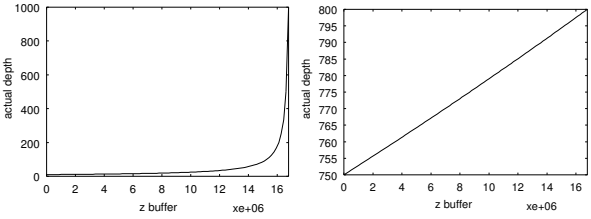
The proposed method uses the depth information to obtain the three dimensional coordinates Z_1 in Eq.(9). Actually, we use the depth information provided by a graphic library, for example, OpenGL[10], DirectX[11] and so on. However, a value stored in the z buffer is usually different from the actual depth. The reason is that for computer graphics the z buffer is used to decide the relative depth of objects in order to perform the hidden surface elimination.

Furthermore, the precision of the z buffer is not uniform over the depth range, and it becomes very poor if we deal with it carelessly. Let z_b be a value in the z buffer, and Z be the actual depth. The relation between z_b and Z is given by the following equation [12, 10].[1]

$$Z = \frac{f_z n_z}{\dfrac{z_b}{s}(f_z - n_z) - f_z} \tag{16}$$

where f_z, n_z are the distances from the view point to the front and the rear clipping planes; the z buffer holds the depth between f_z and n_z. s is the maximum of the z buffer (for example, $s = 2^{24} - 1$ for unsigned 24 bits z buffer).

[1]This equation is for OpenGL[10], however, actually DirectX has the same[11].



(a) (b)

Fig. 2: Examples of the precision of z buffer. (a) $f_z = 1000$, $n_z = 1$. (b) clipped between $f_z = 800$ and $n_z = 750$.

Usually there is more precision at the front of the z buffer, and the precision is affected by the ratio of f_z to n_z. When the ratio is large such as the example shown in Fig.2(a), the precision becomes pretty poor at the rear of the z buffer.

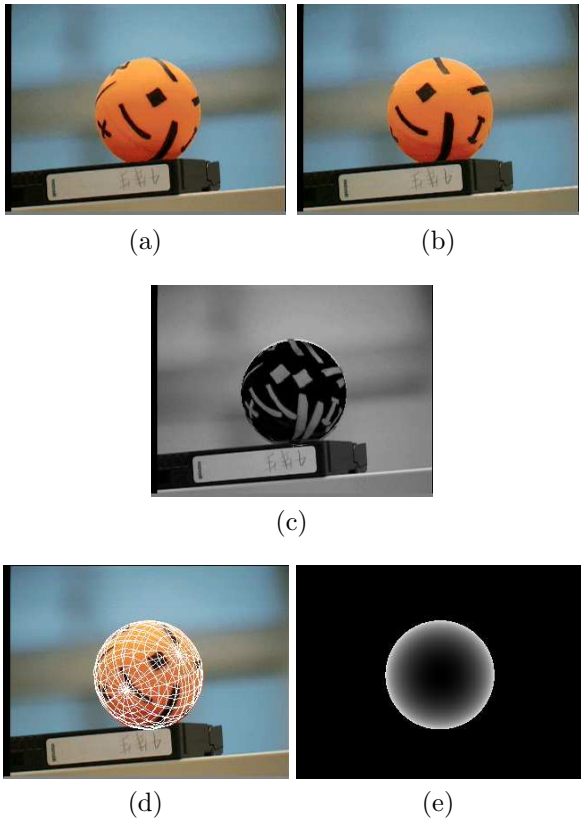
Therefore, we create an image with the z buffer twice. At first time, we have the ratio of f_z to n_z very large, and find the maximum and the minimum values in the z buffer. The values are transformed by Eq.(16) to obtain the corresponding actual depth. Then at the second time, we specify the actual depth as f_z and n_z in order to make the precision of the z buffer good (the example of the appropriately clipping is shown in Fig.2(b)).

4 Experimental results

Experimental results of the proposed method are shown. The parameters to be estimated for a spinning ball include the angular velocity and the translation. To estimate the parameters, we use a CG model carefully fitted to I_1 by a GUI tool, and the depth buffer is obtained for the first frame.

4.1 Preliminary experiment

At first, to demonstrate the correctness of our method, we conducted a preliminary experiment. We captured clearly an image of a colored table-tennis ball with randomly drawn marks as the first image I_1 (shown in Fig.3(a)). The ball was rotated although the location was unchanged, then the second image I_2 was taken (Fig.3(b)). Figure 3(c) show the difference between the first and the second images. The depth buffer shown in Fig.3(e) was obtained by fitting a CG sphere model to I_1. The CG sphere model is drawn as a wire frame as shown in the Fig.3(d). Unlike the cube model, the rotation matrix R for the sphere model is set to I (there is no rotation) because the sphere model has no information about the rotation.



(a) (b) (c) (d) (e)

Fig. 3: Preliminary experiment. (a) The first and (b) the second image of a table tennis ball. (c) Difference between two images. (d) Wire frame of the CG sphere model fitted on the fist image. (e) Visualized depth buffer.

To visualize the convergence of the proposed method, at each step of the optimization we produced a synthetic image which is transformed by the same way in the previous section. Each image of Fig.4 illustrates the difference between I_2 and the synthetic image. At 0th iteration, the two images were quite different. But the synthetic image changed drastically within a few iterations, and after 10 iterations the estimation had almost converged and the difference image had many dark pixels. It means that the synthetic image and I_2 became quite similar to each other (see Fig.5) and that the estimation result was good.

The estimated parameters are shown in Tab.1 and Fig.5(c). According to the estimates, the ball was turned from right to left about y axis (the vertical axis, its positive direction is downward) and this is correct as we see in Figs.5(a) and (b).

4.2 Experiment with real image sequence

Then we conducted an experiment using real images in which a table tennis ball bounces off a board.

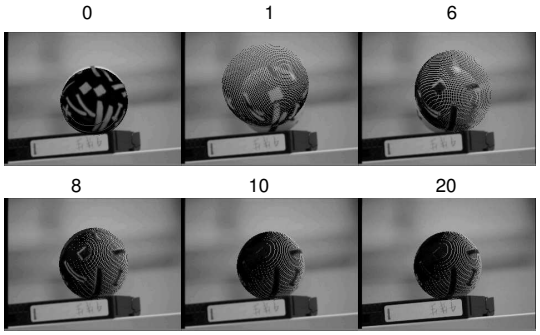
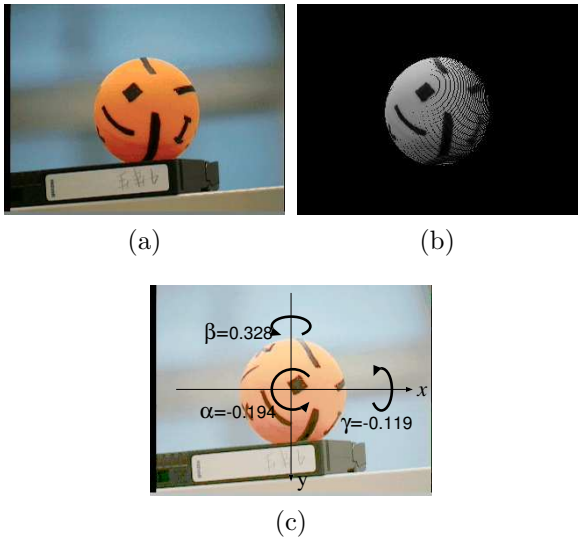


Fig. 4: Difference of the two images at each step of the optimization.



(a) (b) (c)

Fig. 5: (a) The second image. (b) The synthetic image. (c) Estimated parameters with direction superimposed on the first image.

Table 1: Initial parameters R,$\boldsymbol{T}$ and estimated parameters Q,$\boldsymbol{S}$ of a table tennis ball. Angles are represented in radian.

	a	b	c	t_x	t_y	t_z
initial	0	0	0	1	2	208
	α	β	γ	s_x	s_y	s_z
final	-0.194	0.328	-0.119	1.28	1.93	206.3

Figure 6 shows an image sequence of the table tennis ball (the same one in the previous experiment) taken by a high-speed camera (Redlake MASD, MotionMeter500) with a fixed focus lens (Canon PHF6 1.4). The interval between frames was set to 1/500 second and the shutter speed was 1/10000 second so that the motion blur does not occur. The high spin was imparted to the ball, and the ball rebounded off the board. The rebound was

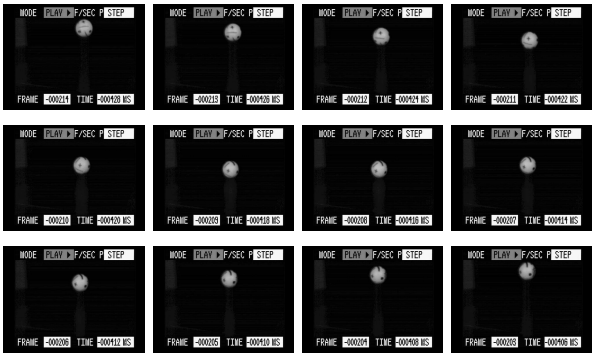


Fig. 6: Image sequence of a table tennis ball taken by a high-speed camera

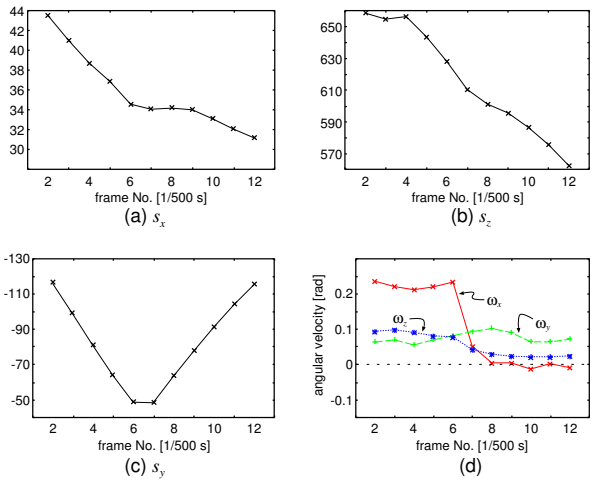


Fig. 7: Experimental result of motion tracking. The translations (a) s_x, (b) s_z, and (c) s_y and (d) the angular velocities about each axis.

captured in 16 frames as shown in Fig.6.

We performed the fitting of the sphere model to only the first frame. The estimated parameters in the previous frame is used to produce the depth buffer for each frame except the first frame, R is set to the identity matrix I at each frame so that the estimated rotation angles α, β and γ are ragard as the anglar velocities ω_x, ω_y and ω_z between two frames.

Figure 7 shows the results of the estimated parameters over 12 frames. As shown in Fig.7, the ball bounced at between the sixth and the seventh frame. The translation s_y changed its direction (from downward to upward). The angular velocity about the x axis was about 0.25 radians per 1/500 second $\approx$ 40 rps at the beginning, and it almost disappeared after the bounce.

4.3 Measuring a ball spin in a real rally

Here we show a result of an experiment for real table tennis rallies to see the difference of skills of an experienced player and a beginner. Two sequences of images in Figure 8 are taken under the same condition in the previous experiment except a varifocal lens (Canon V6X16-1.9 MACRO) is used. In the top two rows of Fig.8 an expert player hit a ball with top spin, and the bottom two rows of Fig.8 shows the impact of a biginner.

Estimated spins in the sequences of two players are shown in Fig.9 where the left is for the expert and right for the beginner. Besides angular velocities about the axes, the total angular velocity is plotted. We can see that the spin of the expert significantly increases after the impact, while the beginner's return did not give mush spin to the ball.

5 Conclutions

In this paper, we have proposed a method for measuring a spin of a spinning ball by the image registration with a 3D model. The proposed method estimates the parameters of the transformation of a known shape object using depth information and minimizes the residuals of intensities of the two images. Experimental results using real images demonstrated the robustness and the usefulness of the proposed method.

Since the proposed method does not estimate the focal length, another method is required for a complete calibration.

Acknowledgment

The authors would like to thank Yukihio Ushiyama for arrangements for and usefull comments on the experiments with table tennis rallies, and the players for their cooperations.

References

[1] O. D. Faugeras, ed., *Three-Dimensional Computer Vision : A Geometric Viewpoint*. MIT Press, 1993.

[2] H. S. Sawhney and S. Ayer, "Compact representations of videos through dominant and multiple motion estimation," *IEEE Trans. on PAMI.*, vol. 18, no. 8, pp. 814–830, 1996.

[3] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factrization method," *IJCV*, vol. 9, no. 2, pp. 137–154, 1992.

[4] R. Szeliski, "Image mosaicing for tele-reality applications," Tech. Rep. CRL 94/2, Digital Equipment Corporation, Cambridge Research Lab, 1994.

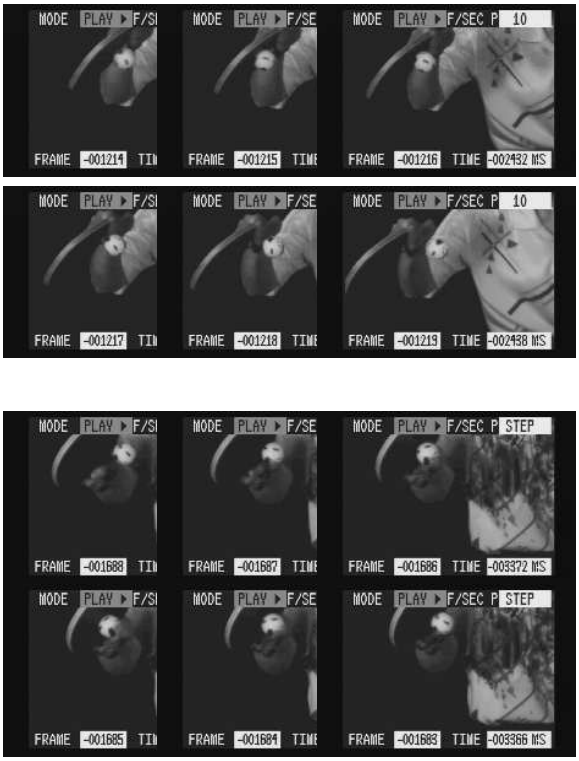


Fig. 8: Rally sequences. (upper) Experienced player. (lower) Beginner.

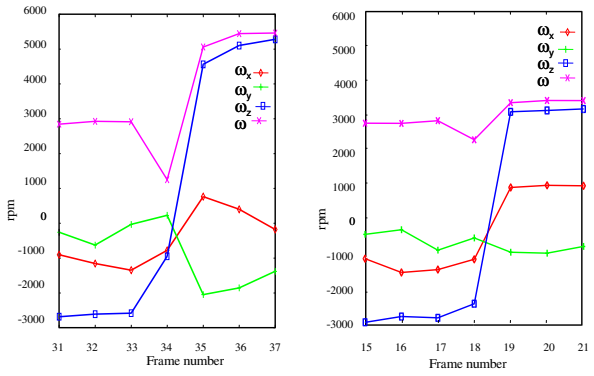


Fig. 9: Estimated spins. (left) Experienced player. (right) Beginner.

[5] R. Szeliski, "Video mosaics for virtual environment," *IEEE Computer Graphics and Applications*, vol. 16, no. 3, pp. 22–30, 1996.

[6] T. Tamaki and M. Yamamoto, "Calibration method by image registration with synthetic image of 3D model," *IEICE Transactions on Information & Systems*, vol. E86-D, no. 5, pp. 981–985, 2003.

[7] G. A. F. Seber and C. J. Wild, *Nonlinear Regression*. New York: Wiley, 1989.

[8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C*. Cambridge University Press, 1992.

[9] E. Kreyszig, *Advanced Engineering Mathematics*. Wiley, 8 ed., 1999.

[10] OpenGL.org, "OpenGL developer FAQ and troubleshooting guide," 2000. http://www.opengl.org/developers/faqs/technical/depthbuffer.htm.

[11] Microsoft, "DirectX C/C++ SDK documentation." http://msdn.microsoft.com/library/en-us/dx8_c/hh/dx8_c/graphics_using_8shf.asp.

[12] A. Watt, *3D Computer Graphics*. Addison-Wesley, 1993.

Tamaki, Toru: received his B.E., M.S. and Ph.D degrees in information engineering from Nagoya University, Japan, in 1996, 1998 and 2001, respectively. Currently, he is a research associate at the Graduate School of Science and Technology, Niigata University, Japan.

Sugino, Takahiko: He is a student at the Department of Information Engineering, Faculty of Engineering, Niigata University, Japan.

Yamamoto, Masanobu: received his B.E. from Faculty of Engineering, Kyushu Institute of Technology in 1973, and M.S. from Graduate School of Science and Engineering, Tokyo Institute of Technology in 1975, and Ph.D degrees in engineering from Tokyo Institute of Technology in 1988. Currently, he is a professor at the Department of Information Engineering, Faculty of Engineering, Niigata University, Japan.