2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

# A Goal Oriented Just-In-Time Visual Servoing for Ball Catching Robot Arm

Koichiro DEGUCHI, Hironari SAKURAI, and Shun USHIDA
Tohoku University, Japan

*Abstract*— **This paper proposes a novel concept for the image based visual control, which is called the goal-oriented just-in-time visual servoing. We apply this control to ball catching task by a robot arm. This approach contains the processes: 1. To achieve the control of robot arm 3D motion using visual observation, we employ no information to reconstruct 3D space from the 2D images, and only define the goal of the task in the image planes. 2. Then we apply the control to achieve all the 2D defined tasks simultaneously. 3. If all the respective goal defined in images are accomplished simultaneously, the 3D task can be interpreted to be achieved. Further more, in the proposed method, we estimate the Jacobian on-line to combine the image motion and 3D motion of the robot arm without the prior knowledge of the arm and camera structures for hand-eye and camera calibrations. The actual experiments show the feasibility of the proposed method for visual control of the ball catching robot.**

## I. INTRODUCTION

In this paper, we propose a novel concept for the image based visual control, which is called the goal-oriented just-in-time visual servoing. We apply this control to ball catching task by a robot arm.

Our approach is:

1) To achieve the control of robot arm 3D motion using observing visual information, instead to reconstruct 3D space from the 2D images, we define the goal state of the task in the image plane.
   For the example of the ball-catching task, we defined its goal so as the images of the arm head position just on the ball trajectories on the each respective more than one image planes simultaneously. It is because the coincidence of the images of arm head and ball only on one image plane does not mean the catching in 3D space. Cameras are placed at different points and observing the ball and the arm head from different view points. That is, the coincidences of them on all the respective images mean the robot arm just catches the ball.
2) Then we apply the control to achieve all the 2D defined tasks simultaneously.
   For the example of our task, we control the motion of the robot arm so that images of its head on the respective image planes move nearer to the ball trajectories which are estimated on the respective images.
3) If all the respective goal defined in images are accomplished simultaneously, the 3D task can be interpreted to be achieved.

In this strategy, we only consider to achieve the goal just in time. The problems are next two: Firstly, we need fine observations including the algorithm to read-out and estimate the ball trajectories and poses of the robot arm on the images. Secondly, we need the control strategy for efficient robot motion to the goal. We do not assume the precise parameters of the cameras and the robot arm. This point is important for easy construction of visual servoing. But, we can not guarantee the success of the catching in time, so that the efficient and precise observation and control are essential.

In our strategy, we employ on-line calibrations for the cameras and the robot arm. We only need rough parameters of the cameras and the robot arm at the start. We estimate the Jacobian which combines the image motion and 3D motion of the arm head in the process of the control. As the result, we do not require precise hand-eye and camera calibrations beforehand to construct the visual control system.

The flying ball catching task requires high speed and high accuracy for image processing and control within the very short time of ball flight. This is the task, in turn, good for the evaluation of the performances of the proposed control concept.

We also have the problem; how to interpret 3D control as the 2D tasks. Our solution is to replace the goal of the 3D ball catching with the condition that the images of ball just coincide the images of the arm head in all respective image planes simultaneously. We control the motion of the robot arm to let its images become nearer to the trajectories of the ball in the respective image plane, and iterate this process until their images meet the ball in all the image planes.

## II. VISUAL SERVOING FOR BALL-CATCHING TASKS

### A. Construction of the visual servoing system

The ball-catching in 3D space is interpreted in 2D images as follows; in the 2 or more camera images, the ball just goes into the catching cup mounted at the head of the robot arm at the same time. Only if in one image the ball meets the cup, it does not necessarily mean the success of the catching. We need 2 or more images seeing from different viewing points.

Here, we prepare two cameras to capture the images of flying ball and the arm. Then, we interpret that the task is accomplished when the positions of the ball and the catching cup have the same coordinates in those two images, respectively. The system configuration is schematically shown in Fig.1. It consists of a robot arm having 7 degrees of freedom and 2 cameras.

The robot arm catches the ball by the next procedures as shown in Fig. 2:

1) Read out image coordinates of the ball, and head and wrist of the robot arm in left and right images.

978-1-4244-2058-2/08/$25.00 ©2008 IEEE. 

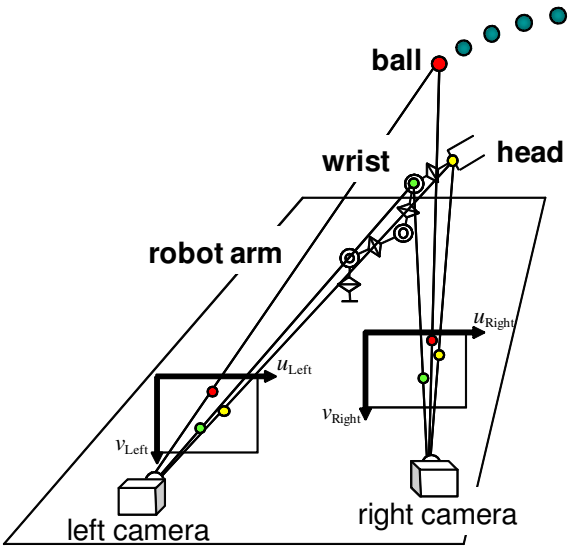



Fig. 1. Schematics of the system configuration for ball catching task by our strategy.

2) Estimate the trajectories of the flying ball in the respective images. Then, we set the target points, in both respective images, for the head and the wrist at the nearest points on the trajectory curves in the images.
3) Move the head and the wrist of the arm letting their images move nearer toward their respective target points in the images.
4) Iterate these processes.

By this iteration, if we successfully put the images of the head and the wrist on the trajectory curves in the respective images before the ball reaches the target points, we can expect to accomplish the task to catch the flying ball.

In the figures, $\boldsymbol{u}$ denotes image coordinates of the head, $\boldsymbol{u}_{\mathrm{ball}}$ denotes the image coordinates of the ball, $\boldsymbol{u}_{\mathrm{d}\cdot}$ is their desired values. We denote the set of joint angles of the robot arm with $\boldsymbol{\theta}$, and $\dot{\boldsymbol{\theta}}$ is the control input to the robot arm. Additionally, the desired position value of the head at the sample time $k$ is denoted by $\boldsymbol{u}_{\mathrm{dhead}}(k)$, which must be the nearest image point on the ball trajectory from $u_{\mathrm{head}}(k)$ on the image plane. The desired position value of the manipulator wrist $u_{\mathrm{dwrist}}(k)$ is also given similarly.

### B. Image based manipulator control

*1) Jacobian between image feature velocities and manipulator joint velocities:* In Fig. 1, we denoted the image coordinates of the head of the manipulator robot arm in the left camera image with $\boldsymbol{u}_{\mathrm{head_L}} = [u_{\mathrm{h_L}}, v_{\mathrm{h_L}}]^{\mathrm{T}}$, and those of the wrist with $\boldsymbol{u}_{\mathrm{wrist_L}} = [u_{\mathrm{w_L}}, v_{\mathrm{w_L}}]^{\mathrm{T}}$. For the right camera image, we also employ $\boldsymbol{u}_{\mathrm{head_R}}$ and $\boldsymbol{u}_{\mathrm{wrist_R}}$, similarly.

The total set of these image features will be represented by a vector $\boldsymbol{u}$ as

$$\boldsymbol{u} = [\boldsymbol{u}_{\mathrm{head_L}}{}^{\mathrm{T}}, \boldsymbol{u}_{\mathrm{wrist_L}}{}^{\mathrm{T}}, u_{\mathrm{h_R}}, u_{\mathrm{w_R}}]^{\mathrm{T}} \quad . \tag{1}$$

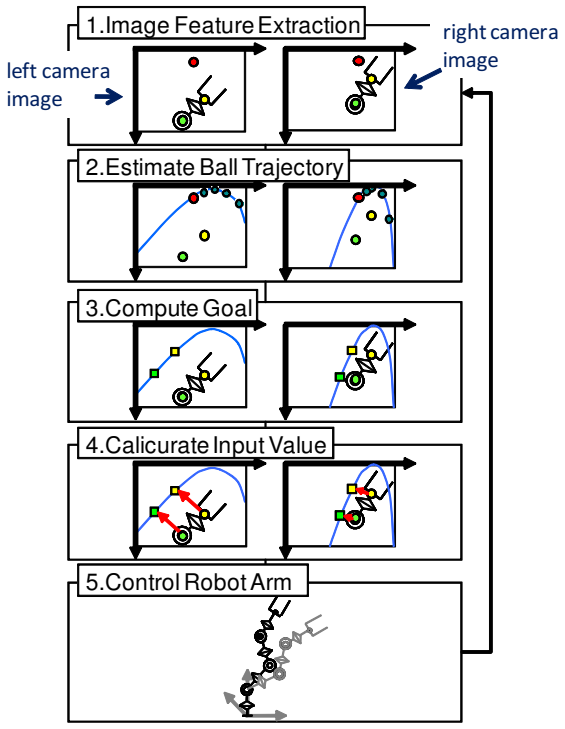



Fig. 2. Schematics of the control strategy of the ball-catching.

In this representation, considering the *epipolar constraints*, we only employ $u_{\mathrm{h_R}}$ and $u_{\mathrm{w_R}}$ of the right camera image.

The angle of the $i$-th arm joint (we used 7DOF manipulator, that is, $i = 1, \cdots, 7$) is denoted by $\theta_i$, and the set of these angles by $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_7]^{\mathrm{T}}$.

The relation between $\boldsymbol{\theta}$ and $\boldsymbol{u}$ will be determined with the structures of the arm and cameras and their relative positions. Generally, $\boldsymbol{u}$ is given as a function of $\boldsymbol{\theta}$ as

$$\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{\theta}) \tag{2}$$

By differentiating (2) in time, we have the relation between their velocities as

$$\dot{\boldsymbol{u}} = \mathbf{J}\dot{\boldsymbol{\theta}} \tag{3}$$

where $\mathbf{J}$ is the Jacobian give as $\mathbf{J} = (\partial \boldsymbol{u}/\partial \boldsymbol{\theta})^{\mathrm{T}} \in R^{6\times 7}$ .

*2) Controller represented with the Jacobian:* Now, we construct the visual control system represented with the Jacobian which produce the control input to let the image feature value $\boldsymbol{u}$ approach the desired value $\boldsymbol{u}_{\mathrm{d}}$. We need to move the robot arm to reduce deviations of the image feature values. That is to choose $\dot{\boldsymbol{\theta}}$ which makes

$$|(\boldsymbol{u} - \boldsymbol{u}_{\mathrm{d}}) - \dot{\boldsymbol{u}}| \rightarrow \min \tag{4}$$

and it gives

$$\dot{\boldsymbol{\theta}} = \lambda \mathbf{J}^{\mathrm{T}}(\mathbf{J}\mathbf{J}^{\mathrm{T}})^{-1}(\boldsymbol{u}_{\mathrm{d}} - \boldsymbol{u}) \tag{5}$$

Here, we set $\mathrm{rank}\mathbf{J} = 6$ and $\lambda(> 0)$ is given as a gain coefficient.

To avoid the singular point where the optimal control value $\dot{\boldsymbol{\theta}}$ cannot be computed because of its special pose where $\mathrm{rank}\mathbf{J} < 6$, we employ the SR-inverse method [5], [6] and (5) is replaced by

$$\dot{\boldsymbol{\theta}} = \lambda \mathbf{J}^{\mathrm{T}}(\mathbf{J}\mathbf{J}^{\mathrm{T}} + \varepsilon \mathbf{E}_6)^{-1}(\boldsymbol{u}_{\mathrm{d}} - \boldsymbol{u}) \tag{6}$$

where $\varepsilon(>0)$ is a small constant value, and $\mathbf{E}_6$ is the identity matrix of $6 \times 6$.

### C. Estimation of the Jacobian

In the proposed method, we do not derive the Jacobian from the actual configuration and dimensions, but estimate it only from the observation of $\dot{\boldsymbol{u}}$ and $\dot{\boldsymbol{\theta}}$ on-line. This means that we do not need the prior knowledge of the configuration and dimensions of the robot arm and the camera parameters. We employ the estimation rule basically proposed in [7].

We assume that the sampling period is sufficiently small so that the Jacobian $\mathbf{J}$ can be considered as constant within the period. Then the equation (3) is digitized as

$$\Delta \boldsymbol{u}(k) = \mathbf{J}(k)\Delta\boldsymbol{\theta}(k) \tag{7}$$

where $\Delta\boldsymbol{u}(k)$ and $\Delta\boldsymbol{\theta}(k)$ are velocities of the image features and angles, respectively, at the $k$-th sampling time, and $\mathbf{J}(k)$ is the Jacobian which is constant within the $k$-th sampling period. Then, we estimate $\mathbf{J}(k)$ by using the iterative exponentially weighted least square method as

$$\begin{aligned}\hat{\mathbf{J}}(k) - \hat{\mathbf{J}}(k-1) \\ = \frac{\left\{\Delta\boldsymbol{u}(k) - \hat{\mathbf{J}}(k-1)\Delta\boldsymbol{\theta}(k)\right\}\Delta\boldsymbol{\theta}(k)^{\mathrm{T}}\mathbf{W}(k)}{\rho + \Delta\boldsymbol{\theta}(k)^{\mathrm{T}}\mathbf{W}(k)\Delta\boldsymbol{\theta}(k)}\end{aligned} \tag{8}$$

where

$$\mathbf{W}(k) = \frac{1}{\rho}\left\{\mathbf{W}(k-1) - \frac{\mathbf{W}(k-1)\Delta\boldsymbol{\theta}(k)\Delta\boldsymbol{\theta}(k)^{\mathrm{T}}\mathbf{W}(k-1)}{\rho + \Delta\boldsymbol{\theta}(k)^{\mathrm{T}}\mathbf{W}(k-1)\Delta\boldsymbol{\theta}(k)}\right\} \tag{9}$$

and $0 < \rho \leq 1$, where the smaller $\rho$ makes the estimation of $\hat{\mathbf{J}}(k)$ less sensitive to the past.

### D. Estimating ball trajectories in the image planes

The motion of the robot is relatively slower than the ball flying speed. The camera frame rate is also as low as 30[fps] to track the ball trajectory. So, we need to estimate the ball trajectory and move the robot arm onto the trajectory just in time of the arrival of the ball. In our scheme of visual control, we do not utilize the 3D positional information for the robot arm, but control only based on the 2D image information. We estimate the ball trajectories, separately, on both of the 2D image planes.

The ball can be considered to fly along a parabola curve in 3D space, which is projected as a conic curve on the image plane by assuming a pin-hole camera optics as shown in Fig. 3. Therefore, the ball trajectory projected on the left image

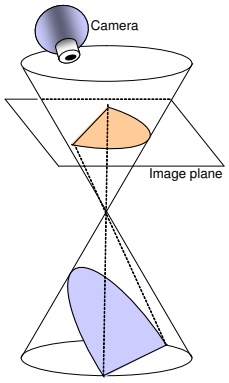



Fig. 3. Conic curve which is the image projection of the parabolic trajectory of the flying ball in 3D space.

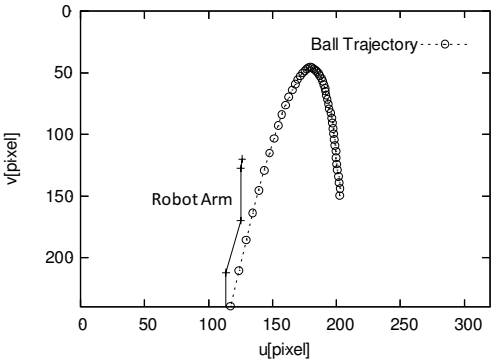



Fig. 4. Ball trajectory estimation using total images of the flying ball.

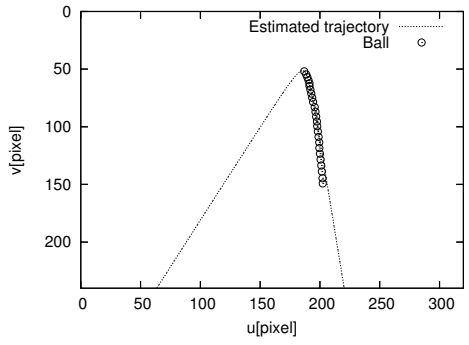
plane is represented as

$$a_L u(k)^2 + b_L u(k)v(k) + c_L v(k)^2 + d_L u(k) + e_L v(k) + f_L = 0 \tag{10}$$

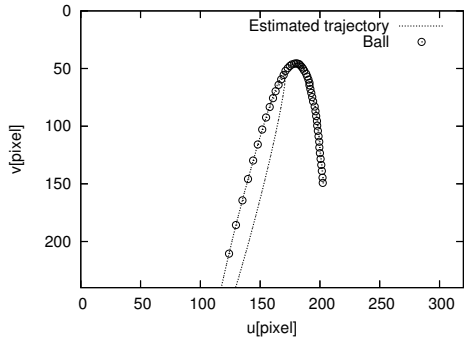
For the right image, we have same representation with coefficients $a_R, b_R, \cdots, f_R$ of the projection curve. We call these set of representations as model 1. In these representations, $(u(k), v(k))$ is the image coordinates of the ball at the $k$-th sampling time. The coefficients $a_L, b_L, \cdots, f_L$ and $a_R, b_R, \cdots, f_R$ are estimated from a set of past images in the sense of the least square errors by the singular value decomposition. We also employ weights for the observed data to reduce the contribution of a long past data.

An example of the estimation of the ball trajectory in the left image is shown in Fig. 4. In this case, the ball reached to the robot arm by about 1.5 sec. This figure shows, finally we obtain good estimation of the trajectories in the image planes by using all the images of the flying ball.

But, in the initial stage of the estimation, we have only small number of ball images, and the estimation will not be good. Fig. 5 shows the trajectory curves estimated from the images up-to 25-th and 50-th, respectively, with the image sampling rate period of 33[ms]. This figure shows that, in the initial stage of the observation, we could not obtain correct

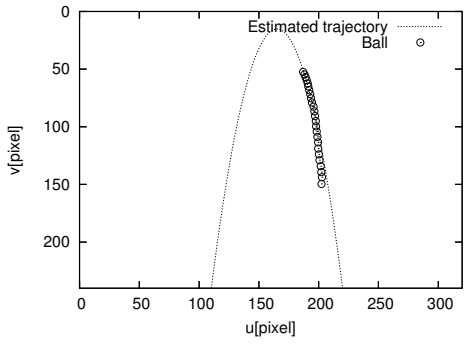



$[k = 25]$

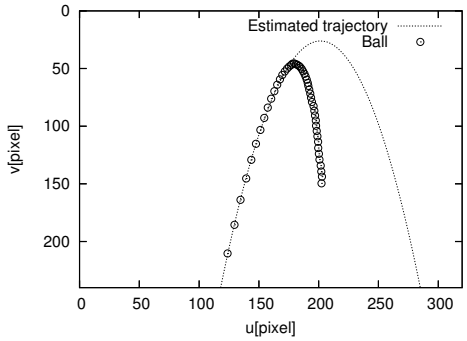



$[k = 50]$

Fig. 5. Estimated trajectories by model 1 using images up-to 25-th and 50-th, respectively.




$[k = 25]$




$[k = 50]$

Fig. 6. Estimated trajectories by model 2 using images up-to 25-th and 50-th, respectively.

estimation of the ball trajectory. This is because the number of images was not sufficiently many relatively to number of the coefficients to be estimated. So, it might be effective that, at the initial stage, the equation (10) of the curve was simplified into

$$a_L u(k)^2 + b_L u(k) + c_L + v(k) = 0 \tag{11}$$

We call this representation as the model 2.

The coefficients $a_L, b_L$, and $c_L$ of (11) were obtained by the iterative weighted least square error method. Fig. 6 shows the estimated ball trajectories using up-to 25-th and 50-th images, respectively, by the model 2 in the same way as for the model 1. This figure shows that, in the initial stage of the observation, the model 2 gave more correct estimation than the model 1.

Therefore, we employ the model 2 first, and switch to the model 1 for the estimation of the ball trajectories. By switching the models for the estimation, we reduce the effects of simplified model, and finally we obtained sufficiently accurate estimation of the trajectories to catch the ball.

On the $k$-th sample images, the target position $u_{\mathrm{dhead}}(k)$ of the robot head is set at the nearest point on the estimated trajectory from the current image position $u_{\mathrm{head}}(k)$. The target position of the arm wrist $u_{\mathrm{dwrist}}(k)$ is also set with the same way.

## III. Tasks for the ball catching

We evaluate the performances experimentally for the next two cases:

- Case 1 We construct the "*true*" Jacobian from the measurements of link lengths of the robot arm, camera positions and poses, and optical system parameters such as the focal lengths of the cameras, and then apply it to the control low of the visual servoing (6).
- Case 2 We estimate the Jacobian by the on-line method described in Sec. II-C, and then apply it.

The robot arm has 7 degrees of freedom. Two cameras observed the scene from right and left back. Camera image sampling period was 33[ms].

For the Case 2 above, to obtain more data about robot arm's dynamics, for 1.0[sec] before the ball was thrown, the arm was moved randomly.

### *A. Performance evaluation for the estimation of Jacobian matrix*

We compared the performances for the Case 1, where the system parameters were assume to be known beforehand, and the Case 2, where we used the on-line estimated Jacobian. Figures 7 and 8 shows the changes of the $u$-coordinate of the robot head on the image plane of right camera and its desired

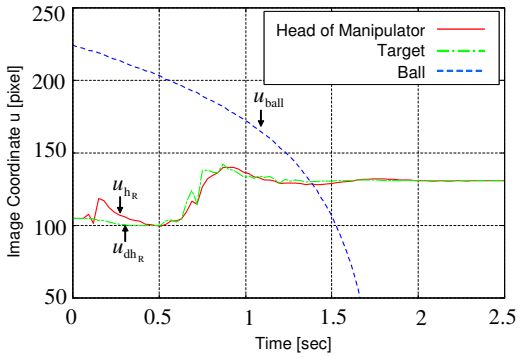



Fig. 7. The change of $u$-coordinates of the ball, target point and arm head in the right image for the Case 1.

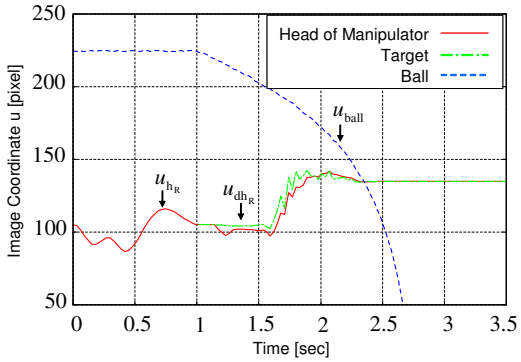



Fig. 8. The change of $u$-coordinates of the ball, target point and arm head in the right image for the Case 2.

value and the flying ball's $u$-coordinate of $\boldsymbol{u}_{\rm ball}$, respectively, with respect to time.

Both of Case 1 (Fig. 7) and Case 2 (Fig. 8) show that the motion of robot head (solid lines) could successfully follow the desired value curve (dotted lines) and converged at a point on the estimated ball trajectory before the ball arrived. We have the same results for the wrist in the right camera image and the head and wrist of the left camera image. This means also that the estimation of the Jacobian of the Case 2 has the same capability as of the Case 1. These results show the possibility of our proposed concept of new visual servoing method to apply on the example of the ball catching tasks.

Now, we show more details of the experimental results of the performances of our proposed method.

Figure 9 shows the configuration of the cameras and the robot arm in the real experiments. Fig. 10 shows the examples of images obtained by the left and right cameras.

The proposed method does not need prior calibration, so that we can place cameras and robot arm with arbitrary configuration if the ball trajectory and the catching head are included within obtained images. We did not make any special consideration for the placements of cameras and robot arm such as a parallel stereo and symmetric or equal distance

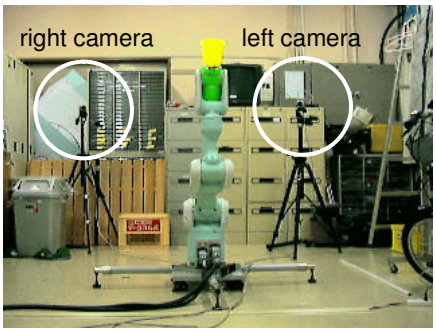



Fig. 9. The visual servo system for real experiments.


(a)An image from left camera.

(b)An image from right camera.

Fig. 10. Images from cameras. The robot arm is at the initial posture and trajectories of the flying ball are overlaid.

configurations.

The images obtained by the two CCD cameras, which have image size of $320 \times 240$ and frame rate of 30[fps], were applied with color-based ball extraction algorithm (details are omitted in this paper), and then the image coordinates of the ball and the head were read out in real time. From these data, the motion command for the 7 DOF robot arm (PA10-7C, Mitsubishi Heavy Ind.) was generated.

### *B. Experimental results*

Fig. 11 shows $u$-coordinates of the trajectories of the actual position of the head $u_{\rm h_R}$ and its desired position in the right image. This shows that the head successfully followed its desired position in the image. The positions of the wrist in the right image and the head and the wrist in the left image also shows same performances.

Fig. 12 shows the ball catching scene in the real experiment. These actual experiments show the feasibility of the proposed method for visual control of the ball catching robot, where we did not need precise prior calibrations.

## IV. Conclusions

We proposed a novel concept for the image based visual control, which is called the goal-oriented just-in-time visual servoing. We apply this control to ball catching task by a robot arm. This approach contains the processes:

1. To achieve the control of robot arm 3D motion using observing visual information, instead to reconstruct 3D space from the 2D images, we define the goal state of the task in the image plane.

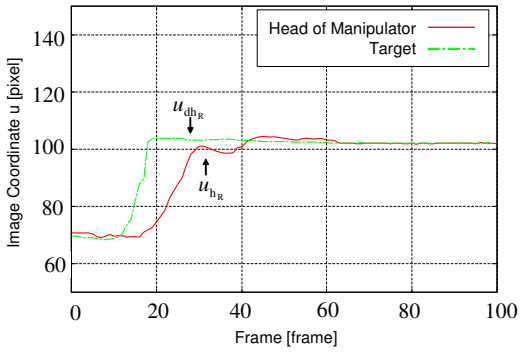



Fig. 11. The changes of $u$-coordinates of target and head of robot arm.

2. Then we apply the control to achieve all the 2D defined tasks simultaneously.

3. If all the respective goal defined in images are accomplished simultaneously, the 3D task can be interpreted to be achieved.

Further more, in the proposed method, we estimate the Jacobian to combine the image motion and 3D motion of the manipulator head without the prior knowledge of the manipulator and camera structures. As the result, we can construct the visual control system which does not need precise prior hand-eye and camera calibrations.

The actual experiments showed the feasibility of the proposed method for visual control of the ball catching robot.

The weakness of the proposed method is: we cannot guarantee the fulfillment of the given task completely in-time. It is because we do not estimate 3D dimensions, and all the performances will be evaluated in the image planes. Instead, we introduce simple concept of the goal oriented task definition and its control.

## References


[1] Seth Hutchinson, Gregory D. Hager, and Peter I. Corke: A Tutorial on Visual Servo Control, IEEE Transactions on Robotics and Automation, Vol. 12, No. 5, October (1996)

[2] R. L. Andersson: A Robot Ping-Pong Player: Experiment in Real Time Intelligent Control, MIT Press, Cambridge, Massachsetts (1998)

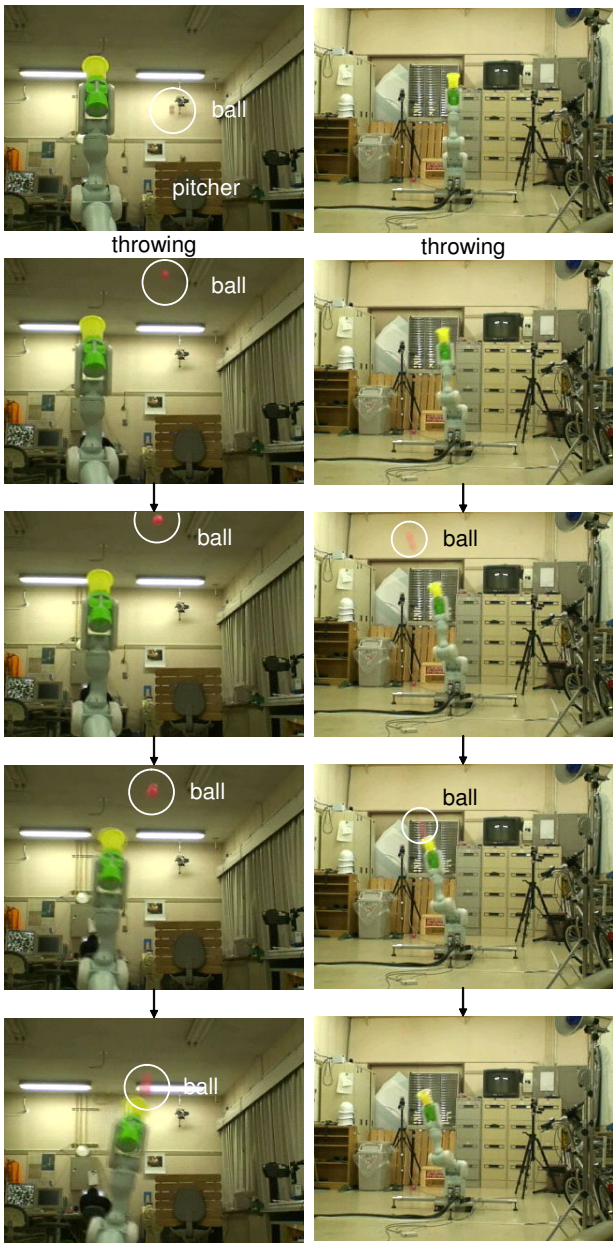
[3] Danica Kragić and Henrik I. Christen: Cue Integration for Visual Servoing, IEEE Transactions on Robotics and Automation, Vol. 17, No. 1, February (2001)

[4] Bernard Espiau, Francois Chaumette, and Patrick Rives: A New Approach to Visual Servoing in Robotics, IEEE Transactions on Robotics and Automation, Vol. 8, No. 3,June (1992)

[5] Y. Nakamura and H. Hanafusa. Inverse Kinematics Solutions with Singularity Robustness for Robot Manipulator Control. Journal of Dynamic Systems, Measurement, and Control, 108:163(171), 1986.

[6] Y. Nakamura, K. Yamane, Y. Fujita, and I. Suzuki. Smart Sensory Computation for Man-Machine Interface from Motion Capture Data and Musculoskeletal Human Model. IEEE Trans. on Robotics, 21(1), Feb. 2005.

[7] K. Hosoda and M. Asada. Versatile visual servoing without knowledge of true Jacobian. In Robots and Systems (IROS'94), 1994.


(a) View from back of the robot

(b) View from front of the robot

Fig. 12. Series of scenes in the real experiment.