6

Online 3-D Trajectory Estimation of a Flying Object from a Monocular Image Sequence for Catching

Rafael Herrejon Mendoza, Shingo Kagami and Koichi Hashimoto
Tohoku University
Japan

1. Introduction

Catching a fast moving object can be used to describe work across many subfields of robotics, sensing, processing, actuation, and systems design. The reaction time allowed to the entire robot system: sensors, processor and actuators is very short. The sensor system must provide estimates of the object trajectory as early as possible, so that the robot may begin moving to approximately the correct place as early as possible. High accuracy must be obtained, so that the best possible catching position can be computed and maximum reaction time is available. 3D visual tracking and catching of a flying object has been achieved successfully by several researchers in recent years (Andersson; 1989)-(Mori et al.; 2004). There are two basic approaches to visual servo control: Position-Based Visual Servoing (PBVS), where computer techniques are used to reconstruct a representation of the 3D workspace of the robot, and actuator commands are computed with respect to the 3D workspace; and, Image-Based Visual Servoing (IBVS), where an error signal measured directly in the image is mapped to actuator commands.

In most of the research done in robotic catching using PBVS, the trajectory of the object is predicted with data obtained with a stereo vision system (Andersson; 1989)-(Namiki & Ishikawa; 2003), and the catching is achieved using a combination of light weight robots (Hove & Slotine; 1991) with fast grasping actuators (Hong & Slotine; 1995; Namiki & Ishikawa; 2003). A major difference exists between motion and structure estimation from binocular image sequences and that from monocular image sequences. With binocular image sequences, once the baseline is calibrated, the 3-D position of the object with reference with the cameras can be obtained.

Using IBVS, catching a ball has been achieved successfully in a hand-eye configuration with a 6 DOF robot manipulator and one CCD camera based on GAG strategy (Mori et al.; 2004).

Estimation of 3D trajectories from a monocular image sequence has been researched by (Avidan & Shashua; 2000; Cui et al.; 1994; Chan et al.; 2002; Ribnick et al.; 2009), among others, but to the best of our knowledge, no published work has addressed the 3-D catching of a fast moving object using monocular images with a PBVS system.

Our system (see Fig. 1) consists of one high speed stationary camera, a personal computer to calculate and predict the trajectory online of the object, and a 6 d.o.f. arm to approach the manipulator to the predicted position.

Source: Visual Servoing, Book edited by: Rong-Fong Fung,
ISBN 978-953-307-095-7, pp. 234, April 2010, INTECH, Croatia, downloaded from SCIYO.COM

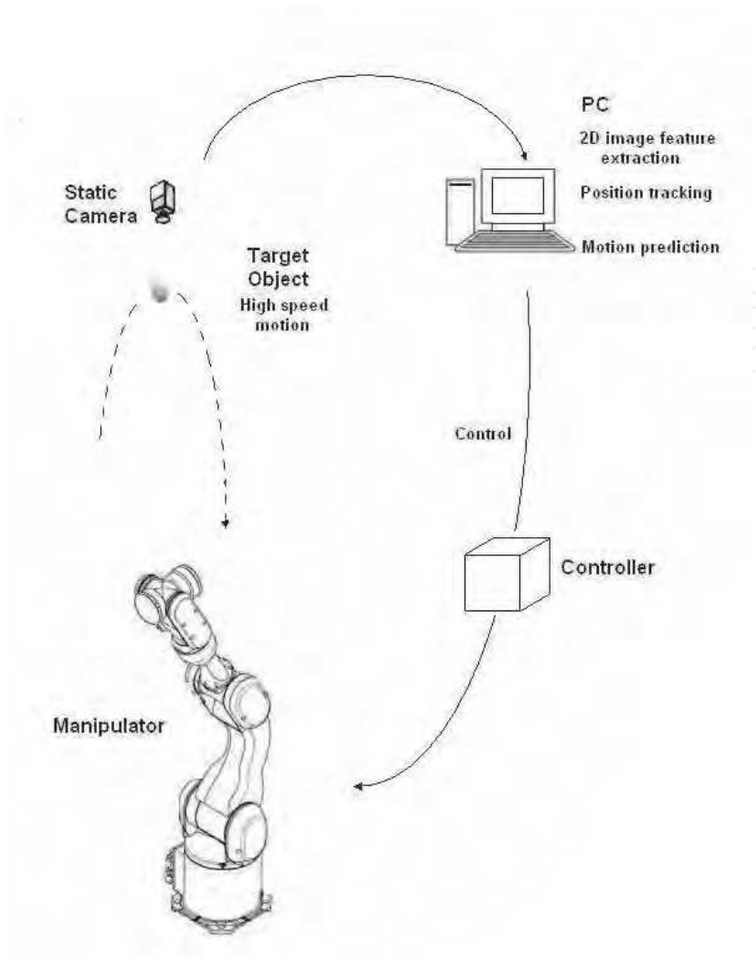


Fig. 1. System configuration used for catching.

The low level robot controller must be able to operate the actuator as close as possible to its capabilities, unlike conventional controllers. The robot must be able to be made to arrive not only at a specific place, but accurately at a specific time. The robot system must act before accurate data is available. The initial data is incorrect due to the inherent noise in the image, but if the robot waits until accurate data is obtained, there is very few time for motion.

2. Target trajectory estimation

Taking 3D points to a 2D plane is the objective of projective geometry. Due to its importance in artificial vision, work on this area has been used and developed thoroughly. The approach to determine motion consists of two steps: 1) Extract, match and determine the location of corresponding features, 2) Determine motion parameters from the feature correspondences. In this paper, only the second step is discussed.

2.1 Camera model

The standard pinhole model is used throughout this article. The camera coordinate system is assigned so as the x and y axis form the basis for the image plane, the z-axis is

perpendicular to the image plane and goes through its optical center (c_u, c_v). Its origin is located at a distance f from the image plane. Using a perspective projection model, every 3-D point $\mathbf{P} = [X,Y,Z]^T$ on the surface of an object is deflated to a 2D point $\mathbf{p} = [u,v]^T$ in the image plane via a linear transformation known as the projection or intrinsic matrix **A**.

$$\mathbf{A} = \begin{bmatrix} -f_u & 0 & c_u \\ 0 & -f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where f_u and f_v are conversion factors transforming distance units in the retinal plane into horizontal and vertical image pixels.
The projection of a 3D point on the retinal plane is given by

$$s\tilde{\mathbf{p}} = \mathbf{A}\tilde{\mathbf{P}} \tag{2}$$

where $\tilde{\mathbf{p}} = [u, v, 1]^T$ and $\tilde{\mathbf{P}} = [^c x, ^c y, ^c z, 1]^T$ are augmented vectors and s is an arbitrary scale factor. From this model, it is clear that any point in the line defined by the projected and original point produces the same projection on the retinal plane.

2.2 3D rigid-body motion

In this coordinate system, the camera is stationary and the scene is moving. For simplicity, assume that the camera takes images at regular intervals. As the rigid object move with respect to the camera, a sequence of images is obtained.
The motion of a rigid body in a 3D space has six degree of freedom. These are the three translation components of an arbitrary point within the object and the three rotation variables about that point. The translation component of the motion of a point at time t_i can be calculated with

$$x_i = C_1 + C_2 t_i + C_3 t_i^2 \tag{3}$$

$$y_i = C_4 + C_5 t_i + C_6 t_i^2 \tag{4}$$

$$z_i = C_7 + C_8 t_i + C_9 t_i^2 \tag{5}$$

where C_1,C_4,C_7 are initial positions, C_2,C_5,C_8 are velocities and C_3, C_6, C_9 are accelerations in the camera x, y, z axis respectively.

2.3 Observation vector

From (2), let the perspective of $\mathbf{P}_i$ be $\mathbf{p}_i = (u_i', v_i', 1)^T$. Its first two components u_i', v_i' represent the position of the point in image coordinates, and are given by

$$u_i' = -f_u \frac{x_i}{z_i} + c_u \tag{6}$$

$$v_i' = -f_v \frac{y_i}{z_i} + c_v. \tag{7}$$

If $u_i = u_i' - c_u$ and $v_i = v_i' - c_v$, (6, 7) can be expressed as

$$u_i = -f_u \frac{x_i}{z_i} \tag{8}$$

$$v_i = -f_v \frac{y_i}{z_i}. \tag{9}$$

Substituting (3, 4, 5) into (8) and (9) to obtain

$$u_i = -f_u \frac{C_1 + C_2 t_i + C_3 t_i^2}{C_7 + C_8 t_i + C_9 t_i^2} \tag{10}$$

$$v_i = -f_v \frac{C_4 + C_5 t_i + C_6 t_i^2}{C_7 + C_8 t_i + C_9 t_i^2}. \tag{11}$$

Reordering and multiplying (10) and (11) by a constant d such as $dC_9 = 1$, yields

$$\begin{aligned} d(C_7 u_i + C_8 u_i t_i + f_u C_1 + f_u C_2 t_i + f_u C_3 t_i^2) = \\ -dC_9 t_i^2 u_i, \end{aligned} \tag{12}$$

and

$$\begin{aligned} d(C_7 v_i + C_8 v_i t_i + f_v C_4 + f_v C_5 t_i + f_v C_6 t_i^2) = \\ -dC_9 t_i^2 v_i. \end{aligned} \tag{13}$$

We have the equation describing the state observation as follows

$$\mathbf{H}_i \mathbf{a}_i + \boldsymbol{\mu}_i = \mathbf{q}_i, \tag{14}$$

where $\boldsymbol{\mu}_i$ is a vector representing the noise in observation, $\mathbf{H}_i$ is the state observation matrix given by

$$\mathbf{H}_i = \begin{bmatrix} f_u & f_u t_i & f_u t_i^2 & 0 & 0 & 0 & u_i & u_i t_i \\ 0 & 0 & 0 & f_v & f_v t_i & f_v t_i^2 & v_i & v_i t_i \end{bmatrix}, \tag{15}$$

$\mathbf{a}_i$ is the state vector

$$\mathbf{a}_i = \begin{bmatrix} dC_1 & dC_2 & dC_3 & dC_4 & dC_5 & dC_6 & dC_7 & dC_8 \end{bmatrix}^T \tag{16}$$

and

$$\mathbf{q}_i = [-u_i t_i^2, -v_i t_i^2]^T \tag{17}$$

is the observation vector.

Considering one point in the space as the only feature to be tracked (the center of mass of an object), the issue of acquiring feature correspondences is dramatically simplified, but it is impossible to determine uniquely the solution. If the rigid object was n times farther away

from the image plane but translated at n times the speed, the projected image would be exactly the same.
In order to be able to calculate the motion, one constraint in motion has to be added. We consider the case of a not-self propelled projectile, in this case, the vector of acceleration is gravity.

$$C_3^2 + C_6^2 + C_9^2 = \frac{g^2}{4} \tag{18}$$

Equation 19 is a constraint given by the addition of the decomposition of the vector of gravity in its different components on each axis for a free falling object.
Substituting C_3,C_6 and C_9 from (14) and (17) into (19) yields

$$\frac{1}{d^2}a_3^2 + \frac{1}{d^2}a_6^2 + \frac{1}{d^2} = \frac{g^2}{4}. \tag{19}$$

From (20) the constant d can be calculated as

$$d = 2\sqrt{\frac{a_3^2 + a_6^2 + 1}{g^2}}. \tag{20}$$

2.4 Object trajectory estimation method

Recursive least squares is used to find the best estimate of the state from the previous state. The best estimate for time i is computed as

$$\hat{\mathbf{a}}_i = \hat{\mathbf{a}}_{i-1} + \mathbf{K}_i(\mathbf{q}_i - \mathbf{H}_i\hat{\mathbf{a}}_{i-1}). \tag{21}$$

where $\mathbf{K}_i$ is the gain matrix, $\mathbf{q}_i$ is the measurement vector for one point, and $\mathbf{H}_i$ is the projection matrix and given by the camera model and time.
The equation that describes the computation of the gain matrix is

$$\mathbf{K}_i = \mathbf{P}_i\mathbf{H}_i^T. \tag{22}$$

$\mathbf{P}_i$ is the covariance matrix for the estimation of the state i, and can be expressed mathematically as

$$\mathbf{P}_i = (\mathbf{P}_{i-1}^{-1} + \mathbf{H}_i^T\mathbf{H}_i)^{-1}. \tag{23}$$

The accuracy of the estimation depends of the number of points projected in the camera plane. Assuming we can observe enough points, the error from the calculated path and the projected path tends to zero.

2.5 Estimated trajectory accuracy

We evaluate the error by the sum of the squares of the 3-D euclidean distance between the simulated position (${}^c x(t)$, ${}^c y(t)$, ${}^c z(t)$) and the estimated position (${}^c\hat{x}(t)$, ${}^c\hat{y}(t)$, ${}^c\hat{z}(t)$), over the flying time interval i.e,

$$e_{xyz} = \sum_{t=0}^{T} ({}^{c}x(t) - {}^{c}\hat{x}(t))^2 + ({}^{c}y(t) - {}^{c}\hat{y}(t))^2 + ({}^{c}z(t) - {}^{c}\hat{z}(t))^2 \quad (24)$$

and in image coordinates, we evaluate the mean distance between the projected object trajectory and the reprojected estimated coordinate ($\hat{u}$, $\hat{v}$), which are functions of the estimated position, as follow

$$e_{uv}(N) = \frac{1}{N}\sum_{n=1}^{N}\sqrt{(u_n - \hat{u}_n)^2 + (v_n - \hat{v}_n)^2}. \quad (25)$$

3. Catching task

3.1 Constraints for the catching task

There are several constraints present for any robotic motion, but for catching there are some others that must be considered. Both of the types are included here for completion.

1. Initial Conditions. The initial arm position, velocity and acceleration are constrained to be their values at the time the target is first sighted
2. Catching Conditions. At the time of the catch, the end effector's position (x_r) has to match that of the ball. Thus at t_{catch}, $\mathbf{x}_r$ is constrained.
3. System Limits. The end effector's velocity and acceleration must stay below the limits physically acceptable to the system. The position, velocity and acceleration of the end effector must each be continuous. The end effector cannot leave the workspace
4. Freedom to change. When new vision information comes in, it should be possible to update the trajectory accordingly.

Two requirements are necessary for a particular trajectory matching solution have relevance to catching. One, the algorithm must require no prior knowledge of the trajectory such as starting position or velocity, and two, the algorithm can not be too computationally intensive.

3.2 Catching approaches

The processing of the computer images is time consuming, causing inherent delays in the information flow, when the position of a moving object is determined from the images, the computed value specifies the location of the object some periods ago. A time delay in the calculated position of the moving object is the main cause of difficulties in the visual-based implementation of the system. This problem can be avoided by predicting the position of the moving object.

There are two fundamental approaches to catching. One approach is to calculate an intercept point, move to it before the object arrives, wait and close at the appropriate time, the situation is analogous to a baseball catcher that positions the glove in the path of the ball, stopping it almost instantaneously. The other approach is to match the trajectory of the object in order to grasp the object with less impact and to allow for more time for grasping, like catching a raw egg, matching the movement of the hand with that of the egg. To be able to match the trajectory, it is expected that the robot end-effector can travel faster that the target within the robot's workspace.

3.3 Catch point determination

Depending on the initial angle and velocity, an object thrown at 1.7 meters from the base of the x-axis of the robot takes approximately 0.7-0.8 seconds to cover this distance. When the object enters the workspace of the robot, it typically travels at 3-6 m/s and the amount of time when the object is within the arm's workspace is about 0.20 seconds. Due to the maximum velocity of our arm (a six d.o.f industrial robot Mitsubishi PA10) being 1 m/s, it is physically impossible to match the trajectory of the object. Because of this constraint, we used move and wait approach for catching.

The catch point determination process begins by selection an initial prospective catch time. We assume that the closest point along the path of the object to the end-effector is when the z value of the object is equal to the robot's one.The time for the closest point is calculated solving 5 for t

$$t_{catch} = \frac{-C_8 + \sqrt{C_8^2 - (4C_9C_7 - Z_{r0})}}{2C_9}; \tag{26}$$

where Z_{r0} is the initial position of the robot's manipulator on camera coordinates, and C_7,C_8 and C_9 are the best estimated values obtained in 2.4. Note that the parabolic fit is updated with every new vision sample, therefore the position and time at which the robot would like to catch the changes during the course of the toss. Once t_{catch} has been obtained, it is just a matter of substituting its value in equations 3,4 to calculate the catching point.

3.4 Convergence criterion

When the mean square error e_{uv} in 26 is smaller than a chosen threshold (image noise + 1 pixel), and the error has been decreasing for the last 3 frames, we considered that the estimated path is close enough to the ground data and therefore the calculated rendez-vous point is valid. Prediction planning execution (PPE) strategy is started to move the robots end effector to the rendez-vous point.

3.5 Simulation results

Simulations for the task of tracking and catching a three dimensional flying target are described. At the initial time (t = 0), the initial position of the center of the manipulator end-effector is at (0.35, 0.33, 0.81) of the world coordinate frame. The speed of the manipulator is given by

$$\dot{x}_{robot} + \dot{y}_{robot} = 1m/s. \tag{27}$$

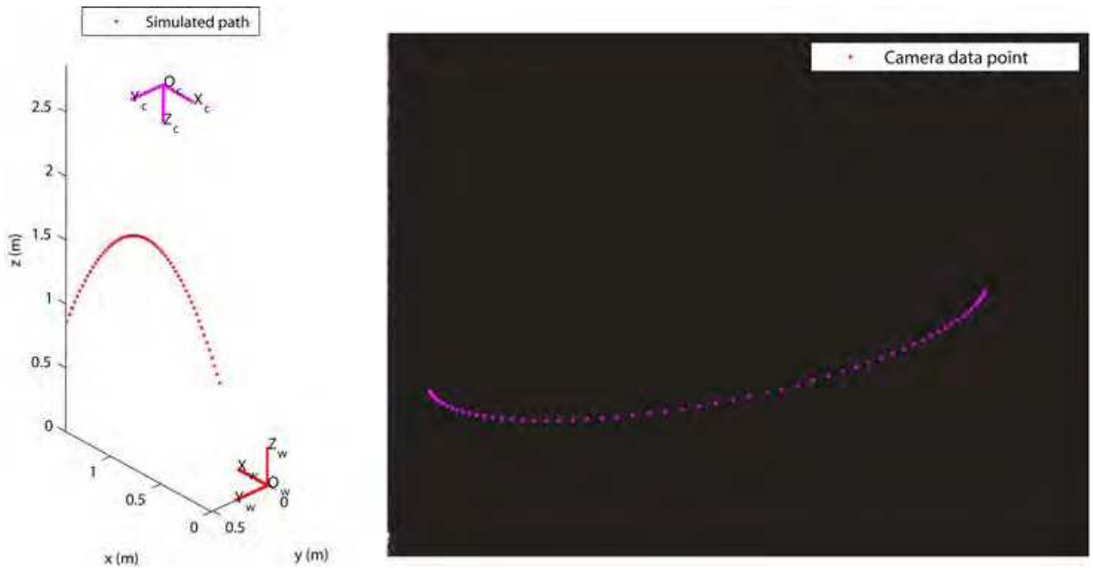
The object motion in world coordinates considered for this simulation (Fig. 2.a) is given by

$$^{w}x(t) = 1.465 - 1.5t \tag{28}$$

$$^{w}y(t) = 0.509 - 0.25t \tag{29}$$

$$^{w}z(t) = 0.8 + 4.318t + \frac{1}{2}gt^2 \tag{30}$$

The coordinates of the object with respect to the camera can be calculated by



(a) (b)

Fig. 2. Path of the object in a) world and b) image coordinates

$$^{c}\mathbf{X}(t) = {}^{c}\mathbf{R}_{w}\,{}^{w}\mathbf{X}(t) + {}^{c}\mathbf{t}_{w} \tag{31}$$

where ${}^{c}\mathbf{R}_{w}$ is the rotation matrix from world to camera coordinates. First, a rotation about the x-axis, then about the y-axis, and finally the z-axis is considered. This sequence of rotations can be represented as the matrix product $\mathbf{R} = R_z(\phi)R_y(\theta)R_x(\psi)$.

The camera pose is given by rotating ψ = 3.1806135, θ = –0.0123876 and ϕ = .0084783 radians in the order previously stated. The translation vector is given by **t** = [0.889;–0.209;–2.853] meters. Substituting these parameters in (32), the object's motion in camera coordinates is given by

$$^{c}x_{sim}(t) = -0.599 + 1.374t + 0.137t^2 \tag{32}$$

$$^{c}y_{sim}(t) = 0.317 - 0.299t + 0.030t^2 \tag{33}$$

$$^{c}z_{sim}(t) = 2.091 - 4.356t + 4.898t^2. \tag{34}$$

The image of the simulated camera is a rectangle with a pixel array of 480 rows and 640 columns. The number of frames used is 60 at a sampling rate of 69 MHz, which accounts for a flying time of 0.87 seconds. The image coordinates (u,v) obtained using focal lengths f_u = 799, f_v = 799 and centers of image c_u = 267, c_v = 205 in (6,7) are shown in Fig. 2.b.

The object passes the catching point (0.42, 0.065, 0.81) at time t = 0.806. If the center of the manipulator end-effector can reach the catching point at the catching time, catching of the object is considered successful. Because the start of the actuation of the robot depends on the convergence criterion stated in 3.4, the success of the catching task is studied for image noise levels of 0, 0.5, 1 and 2 pixels.

Selection of an optimal convergence criteria to begin the robot motion is a difficult task. In Fig. 3, we can see that e_{uv} converges approximately 10 frames earlier than e_{xyz}, for all the

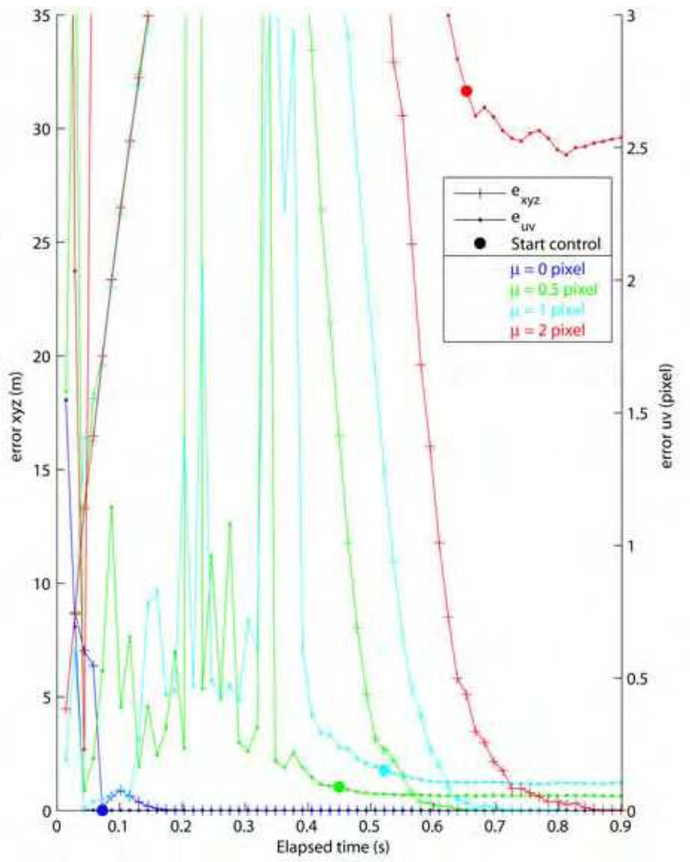


Fig. 3. Errors e_{xyz}, e_{uv} and trigger for servoing.

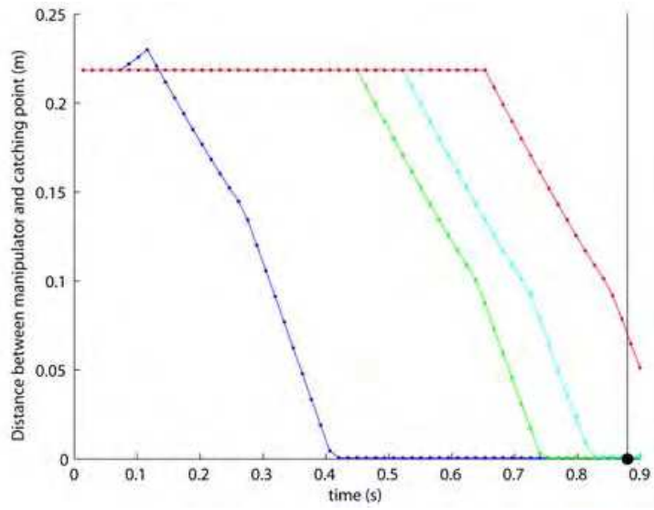


Fig. 4. Distance between manipulator and catching point

noise levels, but because e_{xyz} has not converged yet, triggering the start control flag at this moment would result in an incorrect catching position and the manipulator most probably lose valuable time back-tracking. It could be possible to wait until e_{xyz} is closer to convergence, but that would shorten the time for moving the manipulator. Our convergence

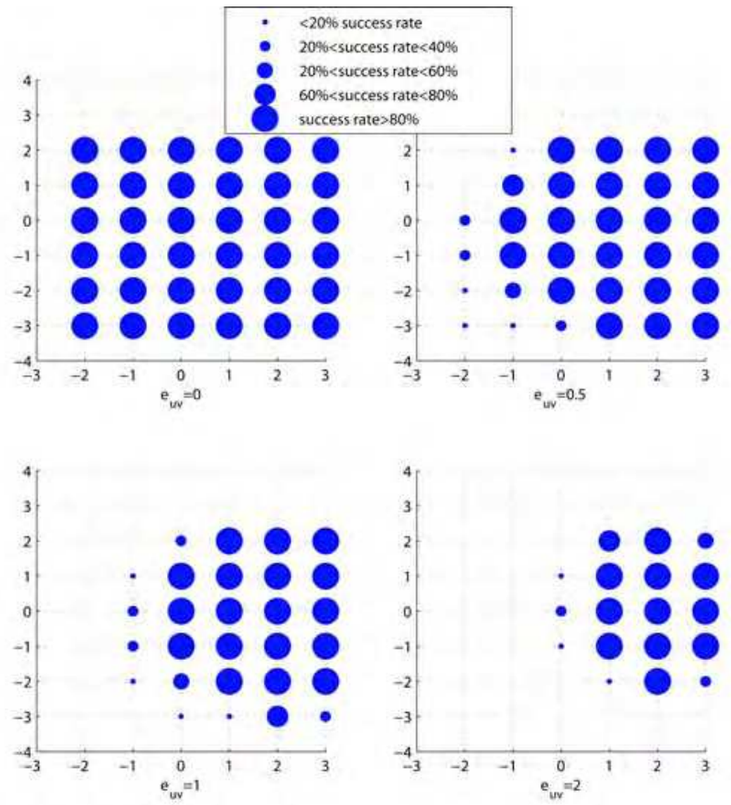


Fig. 5. Target catchable regions

criteria shows a reasonable balance within both stated situations. Fig. 4 shows the distance from the manipulator to the catching point. Judging from these results, we can see that the manipulator reaches the catching point at the catching time, when image noise is smaller than 2 pixels.

3.6 Target catchable region

In this section, we describe the target catchable region for the manipulator for each of the noise levels obtained by simulation. We consider several trajectories, landing grid points at time t = 0.806 from different initial positions. In these figures, the catching rate of the object is shown by size of the circle in the grid. As expected, the smaller the image noise is, the wider is the catching region. It was also found that trajectories that show a relative small change from their initial to final y-coordinates tend to converge faster than those with higher change rates.

4. Experimental results

Implementation of our visual servo trajectory control method was implemented to verify our simulation results. For this experiment, 58 images were taken with a Dragonfly Express Camera at 70 fps, the center of gravity of the object (a flipping coin) in the image plane (u,v) is used to calculate the trajectory. Camera calibration to obtain the intrinsic parameters was realized. Because the coin is turning, the calculated center of gravity varies accordingly to the image obtained, missing data is due to the observed coin projection in the image does

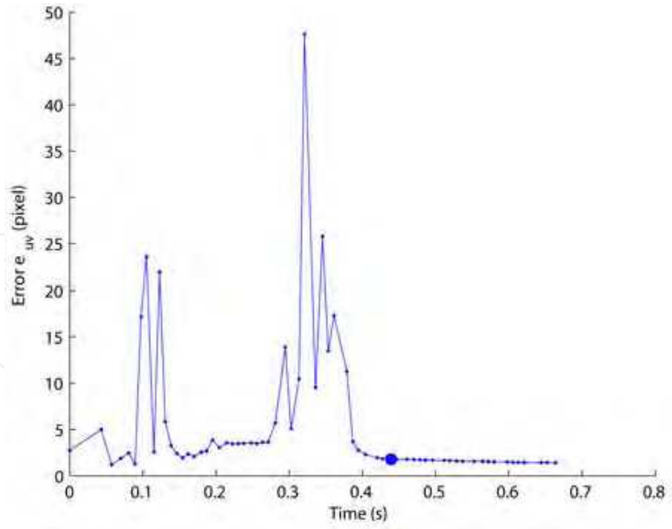


Fig. 6. Error e_{uv} and trigger for servoing.

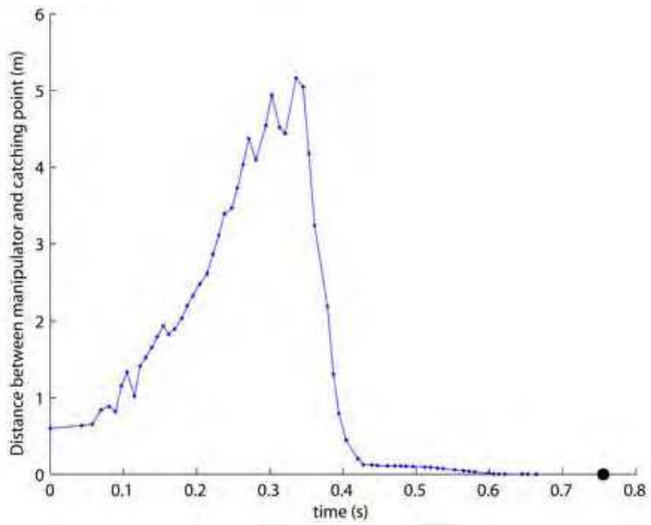


Fig. 7. Distance between manipulator and predicted catching point

not fulfill a minimum specified area, also, blur in the image causes errors in the calculation of the center of the coin. Error e_{uv} was found to be 1.4 pixels, we know from simulations the approximate catching range for this noise level. Experimental results are shown in Fig. 9 and Fig. 10, where it can be seen the movement of the robot to the catching point. Judging from these results, the robot performed the object catching task successfully. From Fig. 6 and Fig. 7, it is visible that the predicted catching point has already converged when control starts.

5. Conclusions and future work

This paper presented an implementation of ball catching task using a robot manipulator. We demonstrated that the robot can catch an object flying in three-dimensional space using recursive least squares (RLS) algorithm to extract and predict the position of the object from one feature correspondence from only a monocular vision system. The object trajectory path

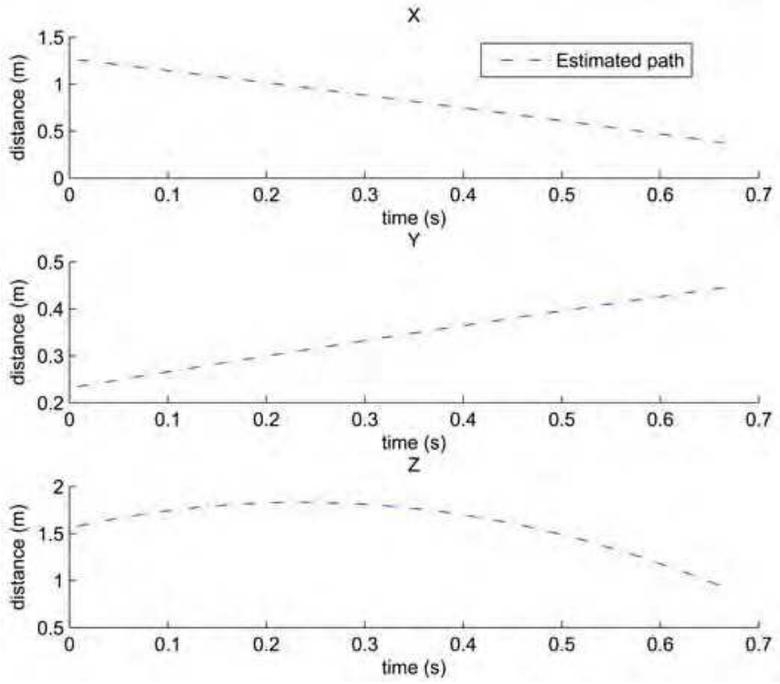


Fig. 8. Calculated path in each axis

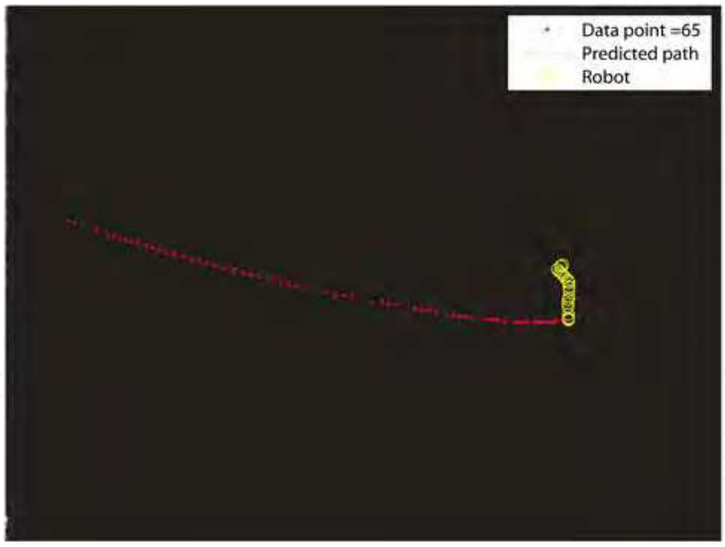


Fig. 9. The center of mass as observed by the camera

was obtained successfully even under high noise images. The recursive estimation technique presented in this paper has numerous advantages over other methods currently in use. First, using only one feature point, the issue of feature points correspondence is simplified. Another advantage is the recursive nature of the computations makes it suitable for real-time applications. Results on simulation and real imagery illustrate the performance of the estimator, and the feasibility of our estimation method for the catching task. Convergence of the path under image noise was studied and a satisfactory criteria was determined

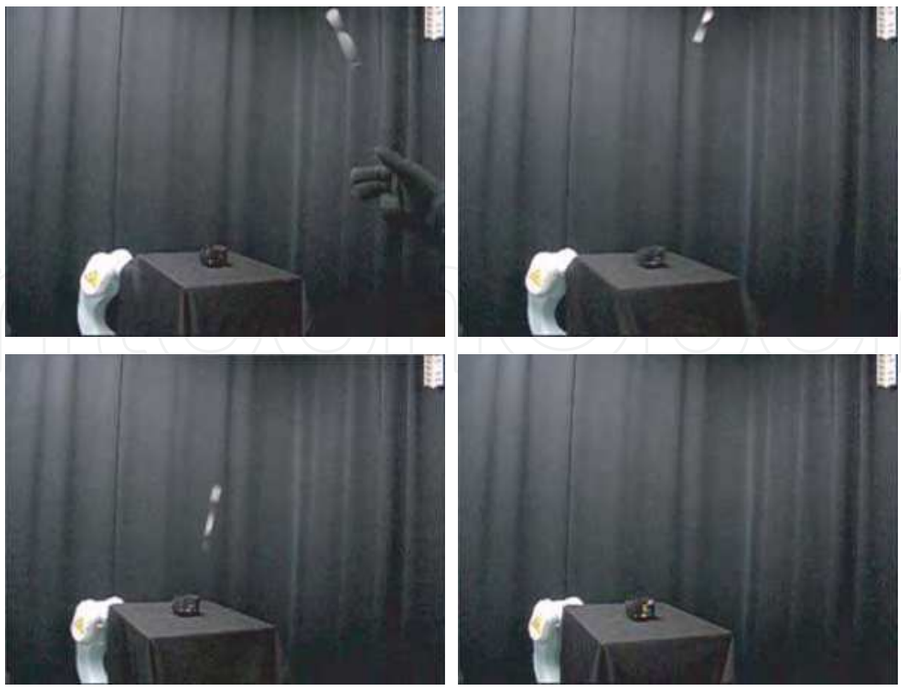


Fig. 10. Sequence of images of object catching

successfully for both simulations and experiments. Current research is directed towards the study of different control approaches to increase the catching range of the manipulator under noisy images.

6. References

Andersson, R. L. (1989). *A robot ping-pong player: Experimental in Real-Time Intelligent Control*, ATT Bell Laboratories, MIT Press.

Avidan, S. & A. Shashua, A. (2000). Trajectory Triangulation: 3D Reconstruction of Moving Points from a Monocular Image Sequence, *IEEE. Trans of Pat, An. and Mac. Int*, Vol. 22, pp. 348-357, 2000.

Chan, C.; Guesalaga, A.; &Obac, V. (2002). Robust Estimation of 3D Trajectories from a Monocular Image Sequence, *Int. journal of imaging sys. and tech.*, Vol. 12, pp. 128-137, 2002.

Cui, N.; Weng, J. J. & Cohen, P. (1994). Recursive-Batch Estimation of Motion and Structure from Monocular Image Sequences, *CVGIP: Image Understanding*, Vol. 59, pp. 154-170, 1994.

Frese, U.; Bauml, B.; Haidacher, S.; Schreiber, G.; Schaefer, I.; Hahnle, M. & Hirzinger, G. (2001). Off-the-Shelf Vision for a Robotic Ball Catcher, *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Maui, 2001.

Hove, B. M. & Slotine, J.J.E. (1991). Experiments in Robotic Catching, *Proc. of American Control Conf*, Vol (1), pp. 380 - 385, Boston, MA, 1991.

Hong,W. & Slotine, J.J.E. (1995). Experiments in Hand-Eye Coordination Using Active Vision, *Proc. 4th Int. Symposium on Experimental Robotics*, Stanford, CA, 1995.

Namiki, A. & Ishikawa, M. (2003). Vision-Based Online Trajectory Generation and Its Application to Catching, *Control Problems in Robotics*, Springer-Verlag, pp. 249-264, Berlin, 2003.

Namiki, A. & Ishikawa, M. (2003). Robotic Catching Using a Direct Mapping from Visual Information to Motor Command, *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2400-2405, Taipei, Taiwan, 2003.

Mori, R.; Hashimoto,K. & Miyazaki, F. (2004). Tracking and Catching of 3D Flying Target based on GAG Strategy, *Proc. Int. Conf. Robotics and Automation*, pp. 4236-4241, 2004.

Ribnick, E.; Atev, S. & Papanikolopoulos, N. P. (2009). Estimating 3D Positions and Velocities of Projectiles from Monocular Views, *Trans. Pat. An. and Mach. Int.* Vol. 31(5), pp. 938-944, 2009.

Visual Servoing

Edited by Rong-Fong Fung

ISBN 978-953-307-095-7

Hard cover, 234 pages

Publisher InTech

Published online 01, April, 2010

Published in print edition April, 2010

The goal of this book is to introduce the visional application by excellent researchers in the world currently and offer the knowledge that can also be applied to another field widely. This book collects the main studies about machine vision currently in the world, and has a powerful persuasion in the applications employed in the machine vision. The contents, which demonstrate that the machine vision theory, are realized in different field. For the beginner, it is easy to understand the development in the vision servoing. For engineer, professor and researcher, they can study and learn the chapters, and then employ another application method.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Rafael Herrejon Mendoza, Shingo Kagami and Koichi Hashimoto (2010). Online 3-D Trajectory Estimation of a Flying Object from a Monocular Image Sequence for Catching, Visual Servoing, Rong-Fong Fung (Ed.), ISBN: 978-953-307-095-7, InTech, Available from: http://www.intechopen.com/books/visual-servoing/online-3-d-trajectory-estimation-of-a-flying-object-from-a-monocular-image-sequence-for-catching

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen