

Trajectory Generation for Obstacle Avoidance of Uncalibrated Stereo Visual Servoing without 3D Reconstruction

Koh HOSODA, Kenji SAKAMOTO and Minoru ASADA

Dept. of Mechanical Engineering for Computer-Controlled Machinery, Faculty of Engineering, Osaka University, Suita, Osaka 565, Japan

Abstract

In this paper, a trajectory generator for a visual servoing system is proposed to make the system accomplish obstacle avoidance tasks in unknown environments. Using an estimated epipolar constraint, the proposed scheme can generate trajectories for the visual servoing system on the 2D image planes by a simple obstacle avoidance method without reconstructing 3D geometry. The proposed scheme is based on the idea, "as long as one of projected trajectories does not intersect with projected obstacles, the trajectory in 3D space can avoid the obstacles." An experimental result is shown to demonstrate the validity of a combination of the proposed trajectory generator and the visual servoing control scheme.

1 Introduction

There have been demands on autonomous robot systems to accomplish given tasks in unknown and/or dynamic environments. The use of visual information is indispensable for such systems. The previous works on computer vision, however, have been adopting deliberative approaches in order to reconstruct 3D scene structure from 2D images, which are very time consuming, and therefore utilizing these methods to real robot applications seems hard. It is needed to design a vision-based real-time controller for such systems which is robust for disturbance, change of parameters, and unknown environments.

Recently, there have been several studies on visual servoing, using visual information in the dynamic feedback loop to increase robustness of the closed loop system (we can find a summary in [1]). The visual servoing scheme is effective to accomplish tasks in such unknown/dynamic environments. However, most of the previous papers on visual servoing study reactive behaviors, and few study purposive ones. Hager had proposed a trajectory planner for a visual servoing system[2] to accomplish alignment tasks such as a floppy insertion in a simple environment of which structure is described in terms of lines[3].

We have proposed a visual servoing controller with an on-line estimator for uncalibrated camera-manipulator systems which does not need any a priori knowledge on the system nor the environment[4]. The proposed controller has mechanisms to accomplish tasks that can be done by purposive behaviors. To accomplish such tasks, one has to generate trajectories on the image planes and feed them to the visual

servoing controller.

In this paper, a trajectory generator for the visual servoing controller without a priori knowledge on the camera-manipulator system nor the environment is proposed to accomplish the obstacle avoidance task. The robot is assumed to have two cameras whose relative position and orientation are fixed, so as to utilize the epipolar constraint[5]. The proposed trajectory generation is based on the idea, "As long as one of projected trajectories does not intersect with projected obstacles, the trajectory in 3D space can avoid the obstacles." Using the trajectory generator, a complicated obstacle avoidance task in 3D space can be reduced to a simple obstacle avoidance task on the 2D image planes.

This paper is organized as follows. First, an uncalibrated visual servoing controller is reviewed. Then, a trajectory generator is proposed. An experimental result is shown to demonstrate how to generate the desired trajectory to realize the obstacle avoidance and how the obstacle avoidance is achieved by the uncalibrated visual servoing control scheme along the generated trajectories.

2 Task and assumptions

A given task for the visual servoing system in this paper is to avoid certain obstacles (see Figure 1). The system has two cameras although the following discussion is applicable for multiple ones. We assume that

- a start point and an end point are given in 3D space, which are projected onto the both image planes,
- the obstacles can be detected from the background on the image planes, and
- the relative position and orientation between two cameras are fixed.

In this paper, we will not mention on the interference of links of the robot with the obstacles, but on the interference of the arm-tip with the obstacles.

3 Visual servoing controller for uncalibrated camera-manipulator system[4]

In this section, a visual servoing controller for uncalibrated camera-manipulator systems with a Jacobian matrix estimator[4] is reviewed.

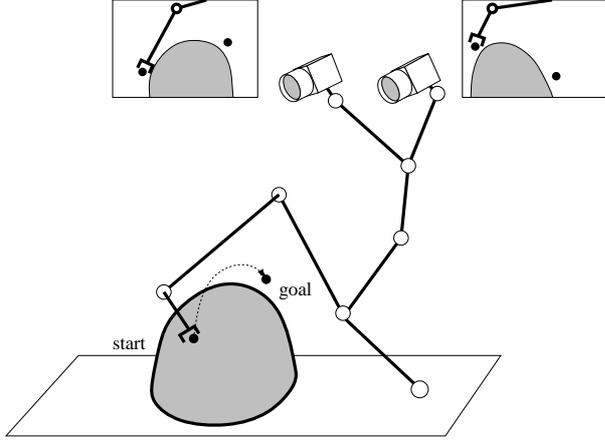


Figure 1: Obstacle avoidance using visual servoing

3.1 System description

A camera-manipulator system consists of manipulators and cameras, as shown in Figure 2. From cameras, one can observe quantities of image features such as position, line length, contour length, and/or area of certain image patterns. A set of measurable variables (which we call system describing variables in the rest of this paper) can describe the state of the camera-manipulator system. Let $\theta \in \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^m$ denote the vectors of the system describing variables and the image features extracted from visual data, respectively. A relation between θ and \mathbf{x} is

$$\mathbf{x} = \mathbf{x}(\theta), \quad (1)$$

because we assume that the system describing variables can describe the state of the system. Differentiating eq.(1), we get a velocity relation,

$$\dot{\mathbf{x}} = \mathbf{J}(\theta)\dot{\theta}, \quad (2)$$

where $\mathbf{J}(\theta) = \partial \mathbf{x} / \partial \theta^T \in \mathbb{R}^{m \times n}$ is a Jacobian matrix of time-derivatives of the quantities of image features with respect to those of system describing variables. The vector \mathbf{x} consists of image feature points of the arm-tip in this paper.

3.2 Jacobian estimator

To realize an uncalibrated controller, a Jacobian estimator is proposed. Assuming that movement of the camera-manipulator system is slow enough to be able to regard the Jacobian matrix \mathbf{J} as constant during the sampling time, we get

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \mathbf{J}(k)\mathbf{u}(k), \quad (3)$$

as a discrete model of the system, where $\mathbf{J}(k)$ and $\mathbf{u}(k)(= T\dot{\theta})$ denote the constant Jacobian matrix and a control input vector in the k -th step during sampling rate T , respectively. From eq.(3), i -th row vector of the matrix \mathbf{J} , \mathbf{j}_i^T , satisfies

$$\{\mathbf{j}_i(k+1)^T - \mathbf{j}_i(k)^T\}\mathbf{u}(k+1) = \{\mathbf{x}(k+2) - \mathbf{x}(k+1) - \mathbf{J}(k)\mathbf{u}(k+1)\}_i. \quad (4)$$

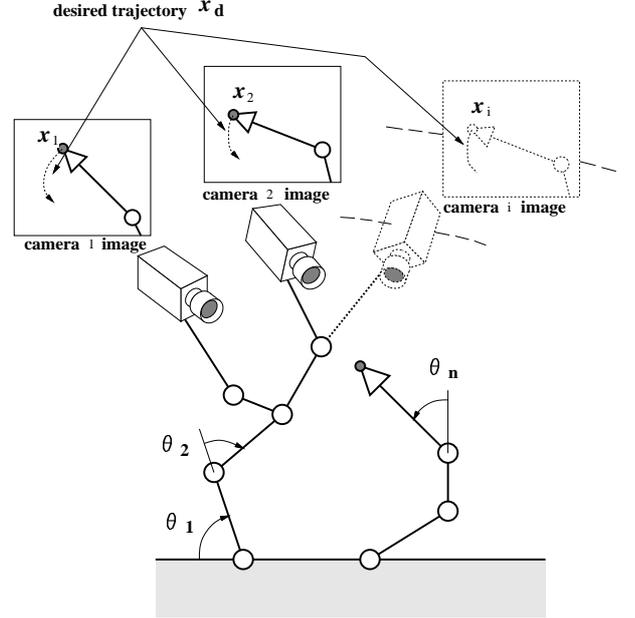


Figure 2: Variables of camera-manipulator system and image features

A solution of eq.(4) which makes the norm of weighted time-derivatives of $\hat{\mathbf{j}}_i$ as small as possible is given by the iteration,

$$\hat{\mathbf{j}}_i(k+1) - \hat{\mathbf{j}}_i(k) = \frac{\{\mathbf{x}(k+2) - \mathbf{x}(k+1) - \hat{\mathbf{J}}(k)\mathbf{u}(k+1)\}_i}{\mathbf{u}(k+1)^T \mathbf{W}_i(k+1)\mathbf{u}(k+1)} \mathbf{W}_i(k+1)\mathbf{u}(k+1), \quad (5)$$

where $\mathbf{W}_i(k+1)$ denotes a weighting matrix. We have to measure $\mathbf{x}(k+2)$ to estimate $\mathbf{J}(k+1)$ if we use eq.(5) for the estimation, which is impossible in real situations. Therefore, $\hat{\mathbf{J}}$ is extrapolated by

$$\hat{\mathbf{j}}_i(k+1) - \hat{\mathbf{j}}_i(k) = \frac{\{\mathbf{x}(k+1) - \mathbf{x}(k) - \hat{\mathbf{J}}(k)\mathbf{u}(k)\}_i}{\mathbf{u}(k)^T \mathbf{W}_i(k)\mathbf{u}(k)} \mathbf{W}_i(k)\mathbf{u}(k). \quad (6)$$

Theoretically, the right-hand side of eq.(6) does not tend to infinity when $\|\mathbf{u}\|$ tends to 0, because $\{\mathbf{x}(k+1) - \mathbf{x}(k) - \hat{\mathbf{J}}(k)\mathbf{u}(k)\}_i$ also tends to 0 at the same or faster speed. In real situations, however, the right-hand side is prone to diverge because of disturbances. To increase the stability of eq.(6), particularly when $\|\mathbf{u}\|$ tends to 0, the estimating law is modified as

$$\hat{\mathbf{j}}_i(k+1) - \hat{\mathbf{j}}_i(k) = \frac{\{\mathbf{x}(k+1) - \mathbf{x}(k) - \hat{\mathbf{J}}(k)\mathbf{u}(k)\}_i}{\rho_i + \mathbf{u}(k)^T \mathbf{W}_i(k)\mathbf{u}(k)} \mathbf{W}_i(k)\mathbf{u}(k), \quad (7)$$

where ρ_i is an appropriate positive constant that ensures stability of eq.(7).

3.3 Visual servoing control

By using the estimated $\hat{\mathbf{J}}$, a visual servoing control scheme is proposed to track desired trajectories $\mathbf{x}_d(k)$:

$$\begin{aligned} \mathbf{u}(k) = & \hat{\mathbf{J}}(k)^+ \{\mathbf{x}_d(k+1) - \mathbf{x}_d(k)\} \\ & + \{\mathbf{I}_n - \hat{\mathbf{J}}(k)^+ \hat{\mathbf{J}}(k)\} \mathbf{k}_r \\ & - \mathbf{K} \hat{\mathbf{J}}(k)^T \{\mathbf{x}_d(k+1) - \mathbf{x}(k)\}, \end{aligned} \quad (8)$$

where $\hat{\mathbf{J}}(k)^+$, \mathbf{I}_n , and \mathbf{K} denote a pseudo-inverse matrix of $\hat{\mathbf{J}}(k)$, an $n \times n$ identity matrix, and a positive definite gain matrix, respectively. The second term on left hand side denotes the redundancy of the system and the vector \mathbf{k}_r is an arbitrary vector.

As far as the Jacobian matrix is estimated appropriately, this controller can track the given desired trajectory making use of feedforward terms without *reactive* feedback terms. These feedforward terms are different from those in traditional tracking control because it based on the *estimated* Jacobian matrix, not on true parameters, to realize given tasks on the image planes. In this sense, the proposed controller is *purposive* to accomplish given visual tasks.

4 Trajectory generation for uncalibrated visual servoing

When the visual servoing controller (8) is applied to real robot applications, desired trajectories on the image planes appropriate for the task must be generated. For such a purpose, a trajectory generator without a priori knowledge of the system nor environment is needed. In this section, a trajectory generator in order to avoid obstacles is proposed.

4.1 Epipolar constraint

There is a constraint between four coordinates of two camera images of one point, so called *epipolar constraint* [5]. Let \mathbf{p}_1 and \mathbf{p}_2 be homogeneous image coordinates of one point in camera 1 and 2, respectively. An epipolar constraint is given as

$$\mathbf{p}_1^T \mathbf{E} \mathbf{p}_2 = 0, \quad (9)$$

where \mathbf{E} is a matrix consisting of the relative position and orientation between two cameras and internal camera parameters. If the relative position and orientation between two cameras do not change, the matrix \mathbf{E} is constant. In such a case, we can calibrate \mathbf{E} by observing the projections of at least eight points. By moving the manipulator of the system and tracking the tip of the manipulator, one can obtain these eight projections in eight different postures of it.

4.2 Generation of obstacle avoidance trajectory

The main idea for generating a trajectory is ‘‘as long as one of projected trajectories does not intersect with projected obstacles, the trajectory in 3D space can avoid the obstacles’’. In this subsection, a trajectory generator according to this idea is proposed.

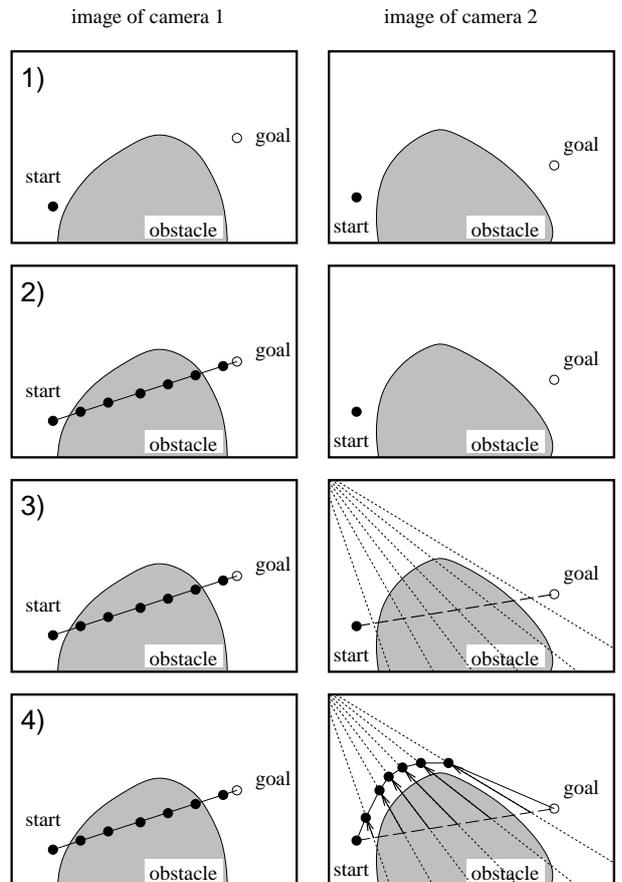


Figure 3: Trajectory generation algorithm to avoid obstacles

The algorithm to generate an obstacle avoidance trajectory is shown in Figure 3. The epipolar constraint matrix \mathbf{E} is estimated in the previous procedure.

1. The obstacles are detected by certain image processes.
2. A trajectory is given as a straight line segment between the start and end points on the image plane 1. Points are placed along the line at intervals of predefined distance.
3. By using the estimated epipolar constraint, epipolar lines corresponding to the sampled points in the image 1 are projected onto the image 2.
4. Searching trajectory points of which the distance to the obstacle area is greater than a threshold along the epipolar lines, the avoiding trajectory in image 2 is obtained.

5 Experiment

In this section, experimental results are shown to demonstrate how to generate the desired trajectory



Figure 4: Camera-manipulator system used for experiments

to realize the obstacle avoidance and how the obstacle avoidance is achieved by the visual serving control scheme along the generated trajectory.

5.1 Camera-manipulator system used for experiments

In figure 4, the experimental system is shown. The system consists of a robot manipulator and two cameras. The robot manipulator is a puma type 6 d.o.f. robot manipulator, Js-5 by Kawasaki Heavy Industry, which is used as a 3 d.o.f. robot in this experiment. The stereo camera system consists of two color CCD cameras, ELMO UN401, whose focal length is 11[mm].

5.2 Experimental equipment

Figure 5 shows the experimental equipment. Video signals from two cameras are sent to an image processing board MV200 (DataCube). The size of original video images from the cameras is 512×480 [pixel] and by using this board, two images are compressed into 256×480 [pixel] images and combined into one 512×480 [pixel] image. This board is also used for thresholding operation to detect the obstacles. The signal from MV200 is sent to the tracking module (Fujitsu). The tracking module has a function of block correlation to track some pre-memorized patterns. The tracking module feeds coordinates of the reference pattern to the controller, MVME167 (CPU:68040, 33MHz, motorola). The controller calculates the desired position of the manipulator by the proposed scheme and sends it to the manipulator controller via VME-VME bus adapter. The sampling time is 33[ms]. The two cameras and the manipulator are not calibrated, so one need not know the parameters of the system completely.

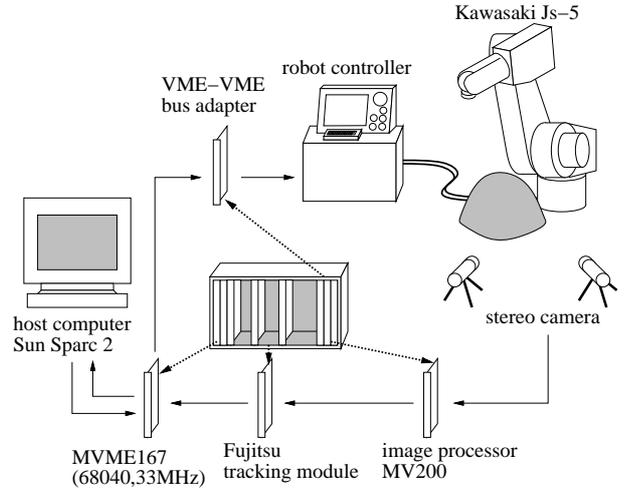


Figure 5: Experimental equipment

5.3 Experimental results

5.3.1 Estimation of epipolar constraint

First, we get eight projected points to estimate the the epipolar constraint matrix E . We move the manipulator in eight different postures, tracking its tip by the tracking module. The manipulator takes eight random postures, and we have got the following eight projected points: $[x_1, y_1, x_2, y_2] = [170, 168, 185, 162], [126, 134, 157, 142], [96, 100, 128, 122], [95, 111, 96, 141], [96, 126, 68, 160], [41, 181, 46, 220], [68, 277, 33, 297], [106, 389, 273, 391]$. From these values, we can estimate matrix E as

$$E = \begin{bmatrix} -5.2 \times 10^{-5} & 1.0 \times 10^{-5} & -1.1 \times 10^{-2} \\ 1.2 \times 10^{-4} & 2.5 \times 10^{-5} & 4.1 \times 10^{-2} \\ 5.8 \times 10^{-3} & -3.3 \times 10^{-2} & 1.0 \end{bmatrix}. \quad (10)$$

5.3.2 Trajectory generation

In this experiment, we can detected obstacle region from background by using a simple thresholding operation. By using the proposed method, we can generate the desired trajectories shown in Figure 6. Desired image points are moving along the generated trajectories in 33[s].

5.3.3 Visual servoing

Feeding the generated desired trajectory to the uncalibrated visual servoing scheme, obstacle avoidance is achieved. Figures 7 and 8 show how the manipulator avoids an obstacle by the proposed scheme. During the control period, the arm did not interfere with the obstacle. In this experiment, the forgetting factor ρ and the weighting matrix W are 0.08 and an identity matrix, respectively.

The manipulator is considered as a 3 d.o.f. one, and the dimension of image features is 4. Therefore,

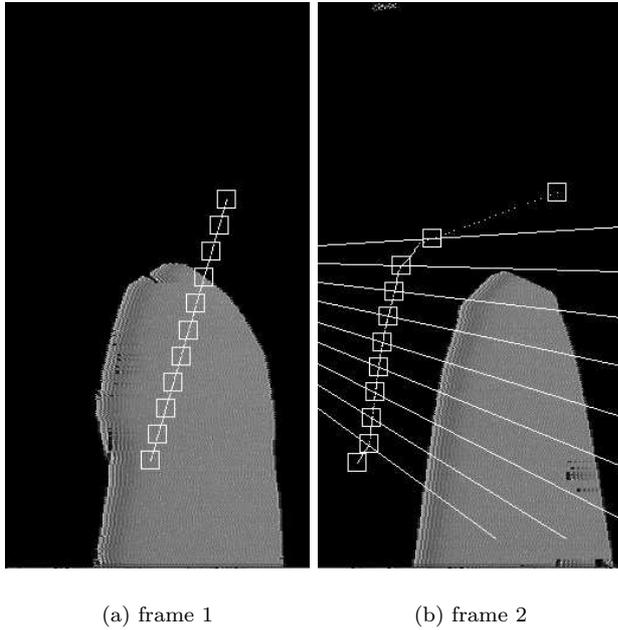


Figure 6: Generated desired trajectories on the image planes

the second term in eq.(8) does not exist, because there is no redundancy in this system.

In Figures 9 and 10, we show the error norm between the desired trajectories and the tracked trajectories, on image planes of camera 1 and 2, respectively. At the beginning of control period, the controller cannot suppress the tracking error because the system does not know the true Jacobian matrix nor any a priori knowledge on the structure of itself. But in some steps, the system can estimate \hat{J} , an appropriate Jacobian to track the trajectory, and therefore the tracking error becomes small.

On the image plane of camera 1, the trajectory is designed at constant velocity, but in the real 3D space, the resultant velocity may not be constant because the proposed trajectory generator does not reconstruct the 3D information but generates trajectory on the 2D image planes. By such a reason, around $t = 33.0[s]$ error norm becomes large.

From these results, we can show that the proposed trajectory generator and the uncalibrated visual servoing controller are valid for achieving an obstacle avoidance task in unknown environments as long as obstacle regions are detected.

6 Discussion and future works

In this paper, a trajectory generator for the visual servoing controller has been proposed to accomplish an obstacle avoidance task. The effectiveness of the control system consisting of the trajectory generator and the visual servoing controller in unknown environment has been shown by the experiment.

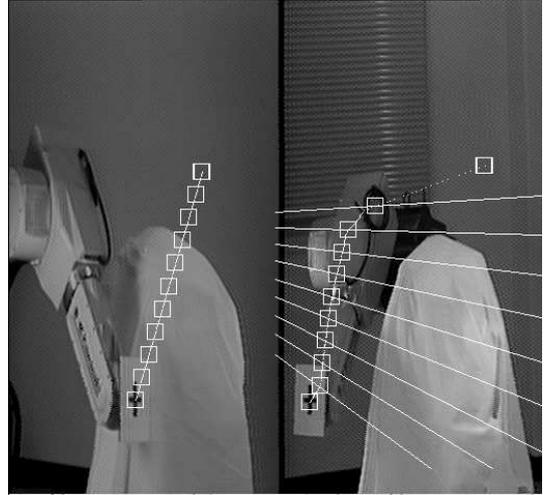


Figure 7: Start posture of the manipulator

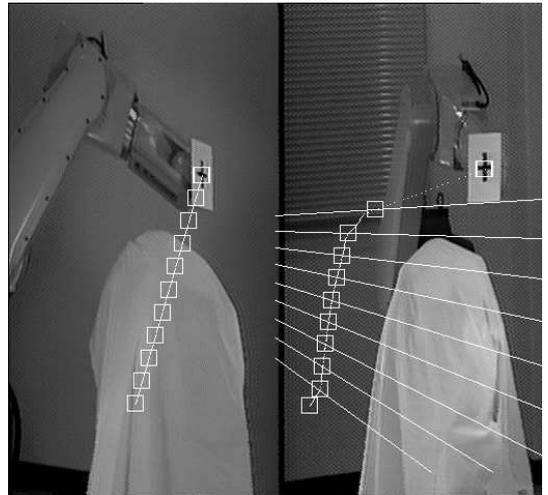


Figure 8: Goal posture of the manipulator

6.1 Modification of trajectory generation

In the case that the distance between two cameras is rather large compared to the distance between the cameras and the obstacles, the proposed method can generate avoiding trajectories. In such a case, a simple detection of obstacle areas is sufficient to realize an obstacle avoidance task, as shown in the experiment in this paper.

In the case that such a condition does not hold, however, obstacle avoiding trajectories cannot be generated in the same manner. In such a case, images of obstacles have to be divided into small patches, assuming that the correspondence of these patches between two images has been established by certain image processing (see Figure 11). Applying the proposed trajectory generating scheme to each corresponding patch pair, one can obtain avoiding trajectories. The smaller the distance between two cameras becomes compared to the distance between the cameras and

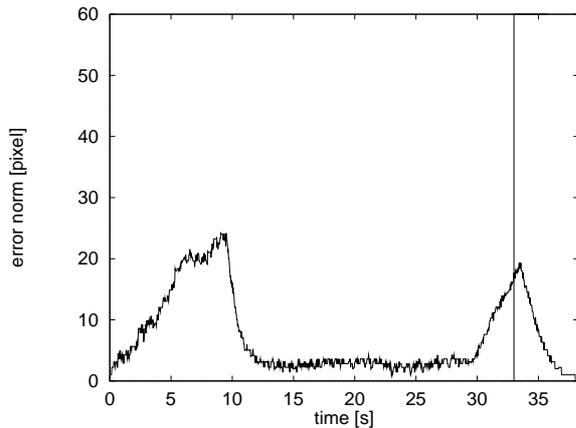


Figure 9: Error norm on image plane 1

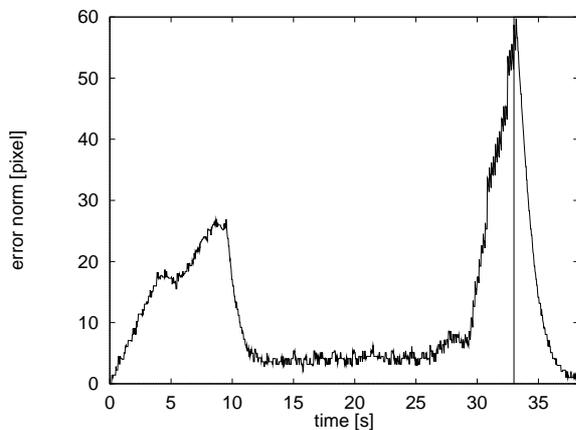


Figure 10: Error norm on image plane 2

the obstacles, the smaller patches becomes to accomplish the given task. In this sense, one has to change resolution of image processing according to the given situation, which might be called “situation-oriented vision.” By using such situation-oriented vision, computational cost may be reduced comparing to the deliberative approaches to reconstruct 3D scene structure, which are very time consuming[6].

6.2 Future works

There are several problems to be solved.

- A certain image processing must be developed to obtain corresponding patches of the images of the obstacles.
- An algorithm must be developed to determine size of the patches.
- Experiments must be done using the modified trajectory generator.
- The trajectory is generated on the image planes, so velocity in 3D space may become large. A certain framework for learning to correct the trajectory from sensor information of the robot must be developed.

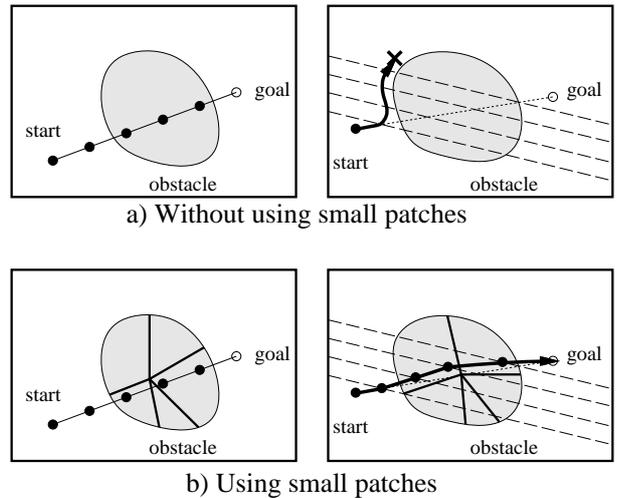


Figure 11: Obstacle avoiding trajectory using small patches

References

- [1] P. I. Corke. Visual control of robot manipulators – a review. In *Visual Servoing*, pages 1–31. World Scientific, 1993.
- [2] G. D. Hager, W.-C. Cang, and A. S. Morse. Robot feedback control based on stereo vision: Towards calibration-free hand-eye coordination. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 2850–2856, 1994.
- [3] G. D. Hager. Real-time feature tracking and projective invariance as a basis for hand-eye coordination. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 533–539, 1994.
- [4] K. Hosoda and M. Asada. Versatile visual servoing without knowledge of true jacobian. In *Proc. of the 1994 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 186–193, 1994.
- [5] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.
- [6] Y. Aloimonos. Introduction: Active vision revisited. In Y. Aloimonos, editor, *Active Perception*, chapter 0. Lawrence Erlbaum Associates, Publishers, 1993.