Proposal of Visual Servoing using Phase-Only-Correlation (POC)

Y. Ri*, H.Fujimoto**
The University of Tokyo
5-1-5, Kashiwanoha, Kashiwa, Chiba, 227-8561 Japan
Phone: +81-4-7136-3881
Email: ri14@hflab.k.u-tokyo.ac.jp*, fujimoto@hflab.k.u-tokyo.ac.jp**

***Abstract*—Image-based visual servo using coordinates of feature points has been well discussed and used in many applications. However, these methods rely on the perfect matching of feature points, which is often difficult in an actual situation.**

Recently, visual servoing methods based on region informations such as mutual information and an image transformation matrix are proposed. While these methods can achieve more robust visual servoing, there is still some difficulties in how to extract and apply such information.

In this paper, region based approach using Phase-Only-Correlation (POC), which is known as a subpixel image matching algorithm, is proposed. Although POC can be applied only in 4 DOF (degree of freedom) without X and Y axis rotation, it accomplishes robust and accurate positioning. We also reveal that the image jacobian, which is often difficult to estimate in realtime, can be written as time invariant matrix in 4 DOF with proposed control scheme.

I. Introduction

Visual servoing, which uses image information to control robots, plays an important role in autonomous robot control [1].

Approaches to visual servoing can be classified as following three types: image-based visual servo (IBVS), position-based visual servo (PVBS) and 2-1/2D visual servo [3], [4], [5].

Position-based methods desides a camera's move on 3D cartesian space and image-based methods desides a camera move from 2D image coordinates. 2-1/2D visual servo is a methods that decouples rotational and translational move and desides each move in 3D or 2D space [5].

These approaches, especially image-based methods using feature points coordinates, are well discussed starting at its stability and robustness [6]. But they are often assumed that the pairs of matched feature points between a desired and a current image can be found properly.

Generally, it is difficult to extract feature points from a natural image and find their matching. Although some kinds of robust feature extraction methods have been carried out such as SIFT (Scale Invariant Feature Transform) [7], ORB (Oriented-BRIEF) [8] and so on, it is still hard to find perfect matches.

To overcome this problem, some approaches using region imformation such as mutual information [9] and image transformation matrix from some feature points [10] are suggested. Those methods are based on region information thus have robustness to image complexities and some image distortions. But there is still some difficulties such as local minima problem or false detection by miss matching.

This paper proposes a region based visual servo method using Phase-Only-Correlation (POC) which is known as a subpixel image registration algorithm [11]. With POC, it is possible to detect image displacement parameters including translation, rotation and scaling paramters in high accuracy. These four image displacement parameters correspond with a 4 DOF camera's move on 3D cartesian space including translation and Z axis rotation. Hence 4 DOF visual servo control scheme using POC parameters is proposed in this paper.

The control scheme in proposed method follows conventional IBVS scheme [2] shown in Fig. 1. It shows that a desired pose and velocity refenrence is given in 2D image so that it can be robust to camera calibration error. Then, the reference camera velocity in 3D cartesian space is created through transformation matrix expressed as 'Jacob ' in Fig. 1. This matrix is called image jacobian matrix and represents the relationship between a camera's move and image feature. For example, considering coordinates of a feature point $\boldsymbol{\xi}_i = [x_i, y_i]^T$ $(i = 1, 2, 3..., n)$ and camera velocity $\boldsymbol{v} = [v_x\ v_y\ v_z\ \omega_x\ \omega_y\ \omega_z]^T$, the well known equation in IBVS [2] can be written as follows:

$$\begin{aligned} \dot{\boldsymbol{\xi}}_i &= \boldsymbol{L}_{\boldsymbol{x}_i}\boldsymbol{v} \\ \boldsymbol{L}_{\boldsymbol{x}_i} &= \begin{bmatrix} \frac{-1}{Z_i} & 0 & \frac{x_i}{Z_i} & x_i y_i & -(1+{x_i}^2) & y_i \\ 0 & \frac{-1}{Z_i} & \frac{y_i}{Z_i} & 1+{y_i}^2 & -x_i y_i & -x_i \end{bmatrix} \end{aligned} \quad (1)$$

Where Z_i is the distance between a camera and a image point. With the inverse or pseudoinverse of image jacobian, the proper 3D velocity can be obtained from 2D image reference. However, the distance between a camera and an object is a function of time and unable to be observed from camera. Thus, the distance is often approximated as a suitable constant value in conventional method [2], [3].

This paper reveals that an image jacobian matrix in 4 DOF can be written as a time invariant matrix in proposed method. After an introduction of POC and derivation of proposed control scheme, a simulation and an experiment are held to evaluate POC based method.

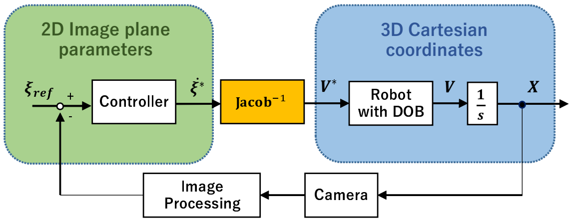


Fig. 1. IBVS system

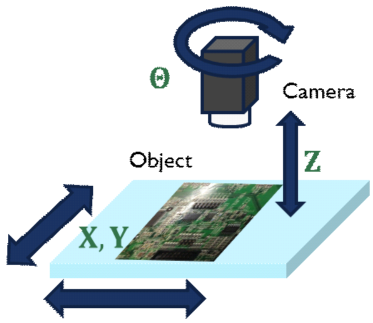


Fig. 2. 4 DOF camera pose and its parameters

II. Phase-Only-Correlation (POC) based estimation of image displacement parameters [11]

Phase-Only-Correlatation (POC) is one of the image registration techniques using 2D Fast Fourier Transformation (FFT), which enables to estimate image displacement paramters in subpixel order. With this technique, we can obtain 4 parameters of image displacement individually including translation, rotation and scaling. This method does not need any feature matching, so it is applied in biometric authentication.

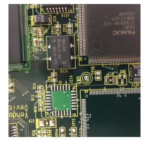
A. 2D translation estimation

Consider two $N_1 \times N_2$ images $f(n_1, n_2)$, $g(n_1, n_2)$. Then the frequency domains of these images transformed by 2D FFT are written as $F(k_1, k_2)$, $G(k_1, k_2)$. The cross spectrum $R(k_1, k_2)$ between $F(k_1, k_2)$ and $G(k_1, k_2)$ can be written as below

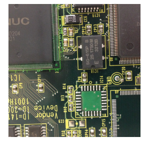
$$R(k_1, k_2) = \frac{F(k_1, k_2)\overline{G(k_1, k_2)}}{|F(k_1, k_2)\overline{G(k_1, k_2)}|} \tag{2}$$

where $\overline{G(k_1, k_2)}$ is a complex conjugate of $G(k_1, k_2)$. POC function $r(n_1, n_2)$ can be calculated by applying 2D IFFT to $R(k_1, k_2)$.

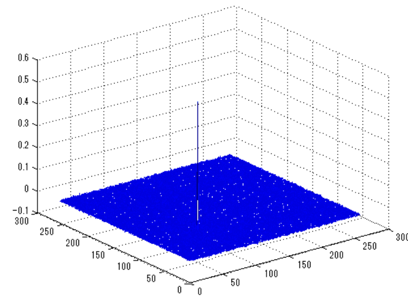
If the reference image and current image is similar, then POC function $r(n_1, n_2)$ has a sharp peak as shown in Fig. 3. The coordinate of the peak represents for the image displacement between the reference and the current image. Therefore, image displacement parameters can be detected by applying 2D IFFT to the multiples of phase information of two images. There are some additional work such as filtering and function fitting in order to estimate more accurate translation. Note that this method can detect image translation up to half size of the image.



(a) Reference image $f(n_1, n_2)$



(b) Current image $g(n_1, n_2)$



(c) POC function $r(n_1, n_2)$

Fig. 3. The appearance of POC function. The position of the peak represent for the image displacement between the reference and current images.

B. Rotation and scaling estimation

In order to estimate rotation and scaling factors, the amplitudes of frequency domains $|F(k_1, k_2)|$ and $|G(k_1, k_2)|$ are used. The rough detecting process of each displacement parameter are shown below and detailed one is explained in the cited paper.

Step1

Calculate 2D fourier spectrum of $f(n_1, n_2)$ and $g(n_1, n_2)$, and get $F(k_1, k_2)$ and $G(k_1, k_2)$.

Step2

Take the logarithm of spectrum amplitudes to get $\log|F(k_1, k_2)| + 1$ and $\log|G(k_1, k_2)| + 1$, then take a log poler mapping to get $|F_{LP}(l_1, l_2)|$ and $|G_{LP}(l_1, l_2)|$.

Step3

Described rotation and scaling error (θ, κ) between $f(n_1, n_2)$ and $g(n_1, n_2)$ by translational error δ_x, δ_y in $|F_{LP}(l_1, l_2)|$ and $|G_{LP}(l_1, l_2)|$ using $(\delta_x, \delta_y) = (N\theta/\pi, -N \log_N \kappa)$.

Step4

Create normalized image $g'(n_1, n_2)$ from $g(n_1, n_2)$ and (θ, κ). Finally, detect the translational error (ξ_x, ξ_y) from $f(n_1, n_2)$ and $g'(n_1, n_2)$.

The parameters including translation, scale and rotation displacements $(\xi_x, \xi_y, \kappa, \theta)$ are some kinds of affine transformation parameters. When define vector $\boldsymbol{x}$ as a coordinate of image feature in current image and $\boldsymbol{x_t}$ as that of reference image, these four parameters can be expressed as follows:

$$\boldsymbol{x} = \boldsymbol{A}(\boldsymbol{x}_t + \boldsymbol{b}) \tag{3}$$

$$\boldsymbol{A} = \kappa \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} \xi_x \\ \xi_y \end{pmatrix}$$

Compared with conventional affine transformation in (4), translation paramters $\boldsymbol{b}$ in (3) can be described on the desired image frame, which is important to make the constant image jacobian.

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{b} \tag{4}$$

In next section, we apply these image displacement parameters in (3) to visual servo.

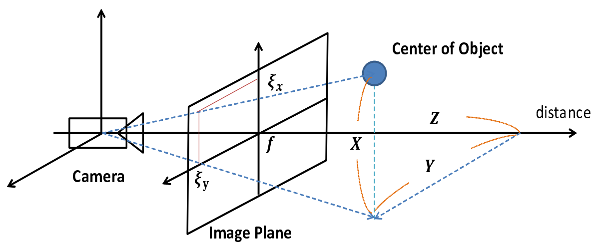


Fig. 4. Perspective model of camera

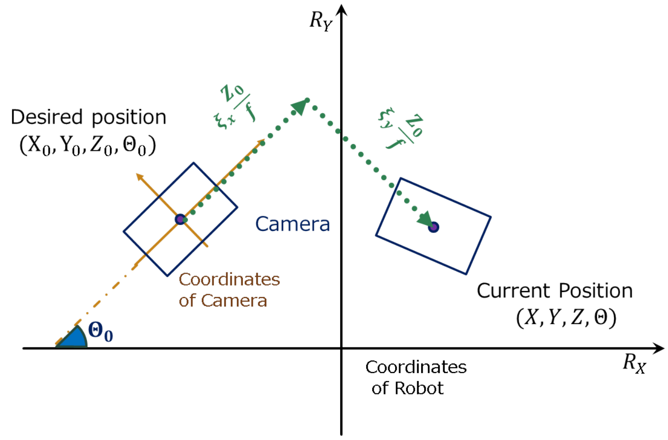


Fig. 5. Relationship between 3D camera position and 2D translation displacement in POC

III. POC based visual servoing

In this section, a new 4 DOF visual servoing scheme using POC parameters shown in (3) is described. Following the example of the conventional image based method, we derive an image jacobian matrix from the relationship between the POC parameters $(\xi_x, \xi_y, \kappa, \theta)$ and the 3D camera pose(X, Y, Z, Θ) shown in Fig. 2.

Without rotation in X and Y axis, POC parameters including x-y translation (ξ_x, ξ_y), scaling κ and rotation θ correspond with the 3D position (X, Y, Z) and the Z axis rotation Θ.

A. Perspective Model

The camera's perspective model in Fig. 4 shows the way that a 3D point at $\boldsymbol{X} = (X, Y, Z)$ in the 3D camera frame is projected into the image and its coordinates in image can be written as $\boldsymbol{x} = (\xi_x, \xi_y)$. The relation between these coordinates are shown below:

$$\begin{pmatrix} \xi_x \\ \xi_y \end{pmatrix} = \begin{pmatrix} f\dfrac{X}{Z} \\ f\dfrac{Y}{Z} \end{pmatrix} \tag{5}$$

where f is the focal length of camera. So, an object at a depth Z is projected on to the image plane magnified $\frac{f}{Z}$ times.

B. Image jacobian in POC based method

Suppose a camera has 4 DOF shown in Fig. 2, we define the camera pose in a fixed 3D cartesian space and set the origin at an intersection point between a robot rotation axis and a object plane. When the camera takes the same image as reference, then we define a desired pose as $(X_0, Y_0, Z_0, \Theta_0)$. Similarly, we also define a current pose as (X, Y, Z, Θ).

2D rotation θ and scaling κ simply correspond with 3D rotation Θ and depth Z, which equation can be written as follows:

$$\theta = \Theta_0 - \Theta \tag{6}$$

$$\kappa = \frac{Z_0}{Z} \tag{7}$$

On the other hand, the relationship between 2D translation ξ_x, ξ_y and 3D pose X, Y is shown in Fig. 5. Although the translation in 2D image is normally described in the current camera frame, due to the definition in (3), the translation parameters in proposed method can be described in the desired camera frame. While the current camera frame varies in realtime, the desired camera frame is fixed so that the relationship can be time invariant as follow:

$$\begin{pmatrix} \xi_x \\ \xi_y \end{pmatrix} = \frac{f}{Z_0} \boldsymbol{R}(-\Theta_0) \begin{pmatrix} X - X_0 \\ Y - Y_0 \end{pmatrix} \tag{8}$$

In (8), $\boldsymbol{R}(\phi)$ is the 2×2 rotation matrix of angle ϕ.

By calculating the time derivative of (6), (7) and (8), we obtain:

$$\begin{bmatrix} \dot{\xi}_x \\ \dot{\xi}_y \\ \dot{\kappa} \\ \dot{\theta} \end{bmatrix} = \boldsymbol{J}_1 \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{\Theta} \end{bmatrix} \tag{9}$$

Where $\boldsymbol{J}_1$ is a Jacobian matrix as follow:

$$\boldsymbol{J}_1 = \begin{bmatrix} \frac{f}{Z_0}\cos\Theta_0 & \frac{f}{Z_0}\sin\Theta_0 & 0 & 0 \\ -\frac{f}{Z_0}\sin\Theta_0 & \frac{f}{Z_0}\cos\Theta_0 & 0 & 0 \\ 0 & 0 & -\frac{\kappa}{Z} & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{10}$$

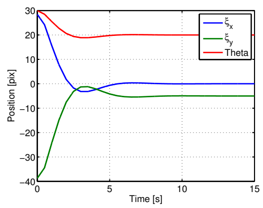
Matrix $\boldsymbol{J}_1$ in (10) has a time varying parameter Z. In the next step, the inverse of $\dot{\kappa}$, $(\frac{1}{\kappa})$ is used instead of itself. Then, these equations can be rewritten as follows:

$$\begin{bmatrix} \dot{\xi}_x \\ \dot{\xi}_y \\ (\dot{\frac{1}{\kappa}}) \\ \dot{\theta} \end{bmatrix} = \boldsymbol{J}_2 \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{\Theta} \end{bmatrix} \tag{11}$$

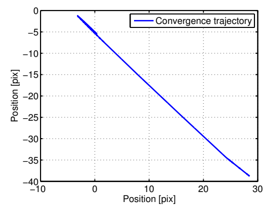
$$\boldsymbol{J}_2 = \begin{bmatrix} \frac{f}{Z_0}\cos\Theta_0 & \frac{f}{Z_0}\sin\Theta_0 & 0 & 0 \\ -\frac{f}{Z_0}\sin\Theta_0 & \frac{f}{Z_0}\cos\Theta_0 & 0 & 0 \\ 0 & 0 & \frac{1}{Z_0} & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{12}$$

TABLE I
SIMULATION PARAMETERS IN FIG. 6

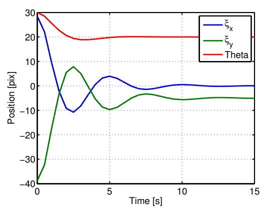
parameter	value
sampling time [ms]	500
desired camera pose $(X_0, Y_0, Z_0, \Theta_0)$	(0,0,15,20)
initial camera pose $(X_1, Y_1, Z_1, \Theta_1)$	(600,-400,20,30)
desired POC parameters $(\xi_{x_{ref}}, \xi_{y_{ref}}, (1/\kappa)_{ref}, \theta_{ref})$	(0,-5,1,0)
controller gains (K_p, K_i, K_d)	$(0.6, 0.001, 0.3)$



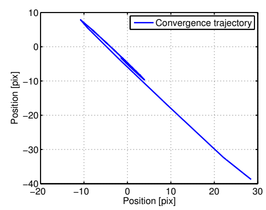
(a) POC parameters response with the true $\boldsymbol{J}_2$



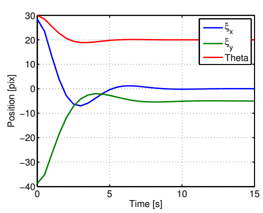
(b) Object trajectory in the camera frame with the true $\boldsymbol{J}_2$



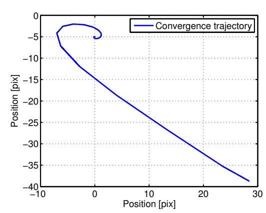
(c) POC parameters response with the larger Z_0 in $\boldsymbol{J}_2$



(d) Object trajectory in the camera frame with the larger Z_0 in $\boldsymbol{J}_2$



(e) POC parameters response with the false Θ_0 in $\boldsymbol{J}_2$



(f) Object trajectory in the camera frame with the false Θ_0 in $\boldsymbol{J}_2$

Fig. 6. Responses of POC parameters to the reference and the Object trajectories in the camera frame. Using (a)(b) true $\boldsymbol{J}_2$, (c)(d) $\boldsymbol{J}_2$ with 1.5 times larger Z_0, (e)(f) $\boldsymbol{J}_2$ with 20 degree larger Θ_0.

$\boldsymbol{J}_2$ shown in (12) is a constant matrix. The consistency of image jacobian matrix enable to control the camera velocity $\boldsymbol{V} = (\dot{X}, \dot{Y}, \dot{Z}, \dot{\Theta})^T$ from image parameters $\dot{\boldsymbol{\xi}} = (\dot{\xi}_x, \dot{\xi}_y, \dot{\kappa}, \dot{\theta})^T$ more properly. In visual servo control scheme, the inverse of a jacobian ${\boldsymbol{J}_2}^{-1}$ is used to make the reference of a camera velocity $\boldsymbol{V}$ from $\dot{\boldsymbol{\xi}}$.

Although there is already proposed 4 DOF control scheme based on affine transformation paramters in (4) [12], it does not consider the proper modeling and still needs some kinds of distance estimation.

There is a little problem that Z_0 and Θ_0 in (12) are unknown because the camera don't know the desired position. This problem is discussed in simulation below.

IV. SIMULATION

In this section, the pocess of positioning task with proposed control scheme is simulated. Considering a camera mounted on a 4 DOF robot manipulator and the robot is assumed to be able to move fast enough so that it can follow the reference velocity in no time. The sampling period of the visual servo controller is defined as 500 ms, which is almost the same with computation time of POC program used in an experiment with 512×512 pix image.

The estimation error of Z_0 and Θ_0 in $\boldsymbol{J}_2$ in (12) are evaluated in this simulation. Initial and desired camera pose are shown in Table I. Fig. 6(a) shows the time response of image displacement, and Fig. 6(b) shows a camera trajectory. Fig. 6(c) and Fig. 6(d) show the responce with larger estimation in Z_0. According to the shape of (12), the larger estimation in Z_0 is equal to the larger control gain with proper Z_0. So, the responce has overshoot. Fig. 6(e) and Fig. 6(f) shows the responce with wrong Θ_0 and it can be said that the wrong Θ_0 estimation may cause undesired camera trajectory.

Since Θ_0 can be detected easily with equation in (6). While it is difficult to know the desired distance Z_0 accurately, the error on Z_0 can be compensated because it is equal to the change in controller gain.

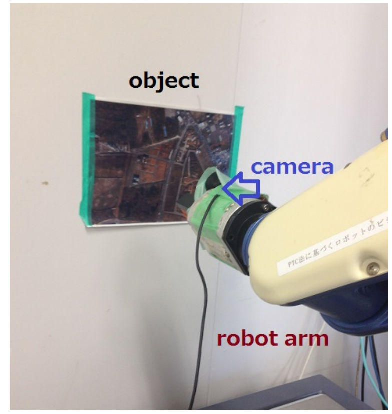


Fig. 7. Experiment settings

V. EXPERIMENTAL RESULTS

Then experiment is held with camera on 6 DOF robot manipulator shown in Fig. 7 and its move is fixed in 4 DOF.

The camera is on the tip of robot manipulator, so the camera pose is regarded to be same as the tip of robot hand. The manipulator is controlled with RT-Linux PC and image processing is computed in windows PC. Both PC is connected with LAN and communicated with socket programming. All programs used in the experiment including the stage control, image processing and the serial communication are written in C.

A. Feature points based method

For comparison, robust feature extraction methods SIFT [7] and ORB [8] are used to estimate the same image displacement parameters $(\xi_x, \xi_y, \kappa, \theta)$ defined in (3). SIFT has been used in many robot vision reserch [10] and ORB is a faster method proposed more recently.

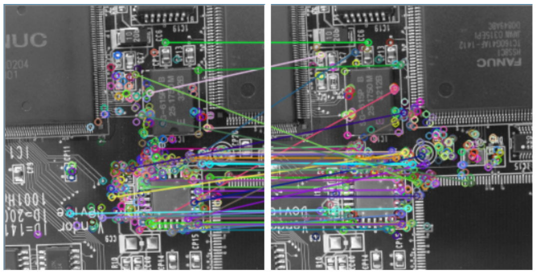


Fig. 8. A result of feature points matching between left image same as Fig. 3(a) and right one same as Fig. 3(b). Color points represent extracted feature points and color lines represent their matching. Since those two images have only translational error, there are should be only parallel matching lines. However there exist crossing lines which mean wrong matches.

Although these method are known to achieve robust matching, Fig. 8 shows its uncertainty in each feature points matching.

Let $\boldsymbol{x}_{\boldsymbol{t}i}$ and $\boldsymbol{x}_i$ ($i = 0, 1, 2, ...$) be the coordinate of each matching point and then $\boldsymbol{A}, \boldsymbol{b}$ in (3) can be estimated as follows:

$$[\boldsymbol{A}^*|\boldsymbol{b}^*] = \arg \min_{[\boldsymbol{A}|\boldsymbol{b}]} \sum_i ||\boldsymbol{x}_i - \boldsymbol{A}(\boldsymbol{x}_{\boldsymbol{t}i} + \boldsymbol{b})||^2 \tag{13}$$

$[\boldsymbol{A}^*|\boldsymbol{b}^*]$ in (13) means minimum mean-square value of $\boldsymbol{A}, \boldsymbol{b}$.

This least-squares method enables to estimate accurate image displacement parameters even if there are some wrong matching like the example shown in Fig. 8.

For the sake of convenience, these methods are reffered as 'SIFT based method' and 'ORB based method'.

B. Positioning task

Two case of positioning tasks are held in each three methods shown above. The methods includes SIFT based method, ORB based method and POC based method.

An initial image error in two case is shown in Table II: case1 has a small rotational error and the other has large one.

Fig. 10(a) shows the camera image in desired camera pose and Fig. 10(b) and Fig. 10(c) show the initial camera image in case1 and case2.

Fig. 10(d) and Fig. 10(e) shows the trajectories of the camera in each methods and each case. Since the trajectories in Fig. 10(d) and Fig. 10(e) are almost straight, decoupling between a rotation and a translation move mentioned in the simulation in Fig. 6 is confirmed.

The table in Fig. 10 shows a final positioning error in 3D cartesian space and calculation time measured in experiment. A final translational positioning error $\boldsymbol{D}$ is defined as $\boldsymbol{D} = \sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}$.

There is little difference in positioning accuracy among these three methods, except ORB based method becomes unstaple when rotational displacement become larger. It is caused by the increasing number of miss matches.

C. Evaluation of POC based method

The result shown in Fig. 10 states that POC based method is different from the othre two methods using feature points in computation time and robustness in paramters detection.

Fig. 9. An aerial photograph printed from GoogleMap

TABLE II
EXPERIMENT PARAMETER IN FIG. 10

parameter	value
initial pose error in case1 [mm,deg] $(X_0, Y_0, Z_0, \Theta_0)$	(10,10,10,30)
initial pose error in case2 [mm,deg]$(X_0, Y_0, Z_0, \Theta_0)$	(10,10,10,120)

The table in Fig. 10 indicates that the computation times in feature points based methods may fluctuate nearly twice. These fluctuations are caused by serch process in a feature potint matching and often considered as undesirable in control. POC based method, on the other hand, has an almost constant computation time because it does not have any serch process.

Furthermore, FFT used in POC can be fasten by hardware implementation so POC method can be more faster in such situation.

Compared to feature points based method, POC based method don't need to consider any wrong matches in robust parameter detection.

Although POC based method can cause false detection occasionally, such situation can be observed by the absence of the peak shown in Fig. 3. Hence, undesired camera's moves caused by false detection can be prevented.

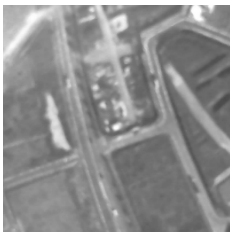
The maximum image displacement allowed in POC is checked experimentally: translational limit is up to the half of an image size, scaling limit is between 0.6 – 1.6 and rotation has no limitation. When image displacement exceeds this limit or there are rotational move on X or Y axis, the parameter detection will fail.

VI. CONCLUSION

In this paper, POC based visual servoing is proposed to overcome the fundemental problem in conventional methods using feature points. Then, improved control scheme in 4 DOF visual servo with the constant image jacobian matrix is derived from a perspective model. And the consistency of the image jacobian indicates that the 3D position information can be properly expressed by 2D image information.

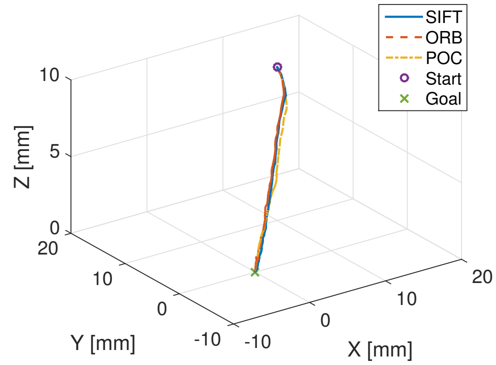
Simulation evaluates the stability toward parameter error in (12) in proposed control scheme, and shows the effectiveness of using image displacement paramter from the point of a camera trajectory.

(a) Desired image

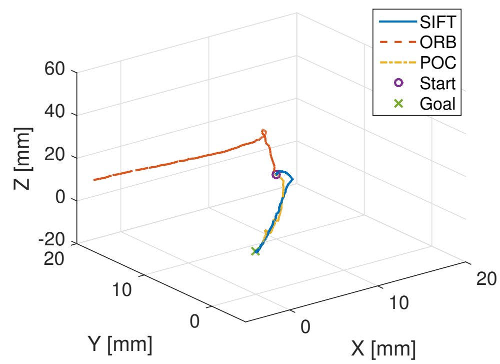


(b) Initial image (case1)

(c) Initial image (case2)



(d) The robot trajectory in case 1



(e) The robot trajectory in case2

	POC	SIFT	ORB
Final Error in Case1 D [mm]	0.1	0.1	0.1
$(\Delta X, \Delta Y, \Delta Z, \Delta\Theta)$ [mm],[deg]	(0.0, 0.0, 0.1, 0.0)	(0.0, 0.0, 0.1, 0.0)	(0.0, 0.1, 0.1, 0.0)
Final Error in Case2 D [mm]	0.2	0.1	–
$(\Delta X, \Delta Y, \Delta Z, \Delta\Theta)$ [mm],[deg]	(0.2, 0.1, -0.1, 0.0)	(0.0, 0.0, 0.1, 0.0)	–
Computation Time [ms]	110±10 ms	120–200ms	45–90ms

Fig. 10. Experiment results in each case and each methods.

Experiment compares the robustness and computation time of the paramters detection. POC is shown to achieve the robust and fixed time paramter detection.

The proposed method will provide the solution to make more proper moving path from image information.

References

[1] A. Goto, H. Fujimoto:"Proposal of 6 DOF Visual Servoing for Moving Object Based on Real-Time Distance Identification,"*SICE Annual Conference 2008, Aug. 20-22* (2008)

[2] S. Hutchinson, G. D. Hager, P. I. Corke,:"A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651670, (1996)

[3] F. Chaumette and S. Hutchinson.: "Visual servo control. Part 1: Basic Approaches." *IEEE Robot. Autom.Magazine* pp.82-90,(2006)

[4] S. Hutchinson, G. D. Hager and P. I. Corke." A tutorial on visual servo control." *IEEE Transactions on Robotics and Automation,* 12(5), pp.651-670.(1996)

[5] E. Malis, F. Chaumette and S. Boudet.:"2-1/2 D visual servoing,"*IEEE Transactions on Robotics and Automation.*(1999)

[6] F. Chaumette, D. Kriegman, G. Hager, S. Morse." Potential problems of stability and convergence in image-based and position-based visual servoing". *The Confluence of Vision and Control, LNCIS Series*, No 237, Springer-Verlag, pp.66-78, (1998).

[7] D. Lowe:"Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol. 60, no. 2, pp. 91-110, (2004).

[8] E. Rublee, V. Rabaud, K. Konolige, G. Bradski:"ORB: an efficient alternative to SIFT or SURF" *IEEE Intl. Conf. on Computer Vision (ICCV)*, volume 13, (2011).

[9] A. Dame and E. Marchand:" Mutual Information-Based Visual Servoing" *IEEE Transactions on Robotics*, 27(5), pp.958-969. (2011)

[10] F. Hoffmann, T. Nierobisch, T. Seyffarth, and G. Rudolph,: " Visual servoing with moments of SIFT features, " *in Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics,* (2007), vol. 5, pp. 4262–4267.

[11] K. Takita, T. Aoki, Y. Sasaki, T. Higuchi and K. Kobayashi. :"High-Accuracy Subpixel Image Registration Based on Phase-Only Correlation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* Vol. E86-A, pp. 1925-1934. (2003)

[12] N. R. Gans and S. A. Hutchinson,:"A switching approach to visual servo control," *Proc. IEEE Internatinal Symp. Intell. Control*,(2002).