2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Visual tracking with occlusion handling for visual servo control

Yasushi Iwatani, Kei Watanabe, and Koichi Hashimoto

Abstract— **This paper proposes a visual tracking method which is robust to occlusion. This paper also integrates the visual tracking method and visual servo control into a vision-based control method with occlusion handling. The proposed method chooses a set of correctly extracted image features, and it then obtains an estimate of all the image features from the correctly extracted image features. The estimation procedure makes it possible to track image features even when occlusion occurs. The method has low computational complexity, since the image Jacobian is used for the image feature selection and estimation. In addition, even when the controller fails to track a moving image feature, it can find the failed image feature without a global search over the entire image plane. As a result, it can track the failed image feature again quickly.**

I. Introduction

Visual servo control [1], [2] is widely used in the area of robotics. Visual servo control uses image features in visual feedback. A typical example of image features is a marker position in the image plane. Each image feature is usually extracted in a window in the image plane, which reduces the computation time and improves accuracy. In order to make a window for each image feature at each time step, each image feature has to be tracked in the image plane, that is, visual tracking is required. Integration of visual servo control and visual tracking is one of major problems in the area of visual servo control [3].

It is assumed in standard visual servo control that all image features are visible and they can be tracked. The assumption sometimes fails in practice. In particular, occlusion occurs and the assumption does not hold in the following cases:

- A tracked target moves out of view.
- An object passes between a camera and a tracked target.
- The background color becomes similar to the color of a tracked target.

Once visual tracking fails due to occlusion, a point which is not a target is tracked. This may generate crucial errors in 3D reconstruction and in visual feedback control.

Recently some effort has been devoted to vision-based control with occlusion handling. Visual servo control methods proposed in [4], [5], [6] always keep tracked targets in the camera field of view, but they can not be allied to general occlusion cases as mentioned above. A vision-based control method which predicts occlusion has been presented in [7]. The method is effective only against self occlusion which can be predicted through the Binary Space Partitioning Tree model [8], [9]. Weighted image feature approaches have been presented in [10], [11]. If an image feature is far from the median value or the reference, then the image feature is not used in feedback control. The weighted approaches can not determine whether visual tracking fails or not. Thus once a point which is not a target is tracked, visual tracking may continue to fail.

This paper proposes a visual tracking method which includes an image feature selection algorithm and an image feature estimation algorithm. The method chooses a set of correctly extracted image features in the selection step. It then estimates all the image features from the correctly extracted image features. The estimation procedure enables us to track image features even when non-predicted occlusion occurs. This paper also integrates the image feature selection, the image feature estimation and visual servo control into a vision-based control method. Advantages of the proposed visual servo control with occlusion handling are summarized as follows:

- It has low computational complexity, since the image Jacobian is used for the image feature selection and estimation. The image Jacobian usually appears in visual servo control.
- The proposed method can determine whether visual tracking fails or not. For each image feature, an error between the measurement and an estimate is computed. If the error is bigger than a given tolerance, it is seen that visual tracking fails.
- The proposed method estimates all the image features at every time step. Thus it can find a failed target again, if the target is visible.

II. Image Jacobians

Let m image features be given. The image features are labeled from 1 to m. The vector of the i-th image feature is denoted by $\boldsymbol{\xi}_i \in \mathbb{X}_i \subset \mathbb{R}^{n_i}$, where $\mathbb{X}_i$ is the domain of $\boldsymbol{\xi}_i$. If $\boldsymbol{\xi}_i$ is the position of a marker in the image plane, then $\mathbb{X}_i \subset \mathbb{R}^2$. We set

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}_1^\top & \boldsymbol{\xi}_2^\top & \dots & \boldsymbol{\xi}_m^\top \end{bmatrix}^\top . \tag{1}$$

In addition, we define

$$\boldsymbol{\xi}_{\mathbb{I}} = \begin{bmatrix} \boldsymbol{\xi}_{\sigma_1}^\top & \boldsymbol{\xi}_{\sigma_2}^\top & \cdots & \boldsymbol{\xi}_{\sigma_{|\mathbb{I}|}}^\top \end{bmatrix}^\top \tag{2}$$

where $\mathbb{I} \subset \{1, 2 \dots, m\}$, $|\mathbb{I}|$ is the size of the set $\mathbb{I}$, $\sigma_j \in \mathbb{I}$ $(j = 1, 2, \dots, |\mathbb{I}|)$ and

$$\sigma_1 < \sigma_2 < \dots < \sigma_{|\mathbb{I}|}. \tag{3}$$

This work was supported in part by the Grant-in-Aid for Scientific Research (B) No. 18360110 of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

Y. Iwatani, K. Watanabe, and K. Hashimoto are with the Department of System Information Sciences, Tohoku University, Aramaki Aza Aoba 6-6-01, Sendai, Japan {iwatani, watanabe, hashimoto}@ic.is.tohoku.ac.jp

As we shall see later, $\mathbb{I}$ implies a set of correctly extracted image features. The vector $\boldsymbol{\xi}_{\mathbb{I}}$ is used to generate a control input, when $\boldsymbol{\xi}_i$ can be extracted correctly for $i \in \mathbb{I}$ and when $\boldsymbol{\xi}_i$ can not be extracted due to occlusion for $i \notin \mathbb{I}$. Details will be discussed in Section IV.

Next, let $\boldsymbol{q} \in \mathbb{R}^n$ denote the vector of generalized coordinates of a given control object. Using a nonlinear function $\boldsymbol{\alpha}_i$ of $\boldsymbol{q}$, we can write

$$\boldsymbol{\xi}_i = \boldsymbol{\alpha}_i(\boldsymbol{q}) \tag{4}$$

(see for instance [12]). We define

$$\boldsymbol{\alpha}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{\alpha}_1^\top(\boldsymbol{q}) & \boldsymbol{\alpha}_2^\top(\boldsymbol{q}) & \ldots & \boldsymbol{\alpha}_m^\top(\boldsymbol{q}) \end{bmatrix}^\top \tag{5}$$

and we have

$$\boldsymbol{\xi} = \boldsymbol{\alpha}(\boldsymbol{q}). \tag{6}$$

Differentiating the above equation yields

$$\dot{\boldsymbol{\xi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{7}$$

where

$$\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{\alpha}(\boldsymbol{q})}{\partial \boldsymbol{q}}. \tag{8}$$

The matrix function $\boldsymbol{J}$ defined by (8) is called the image Jacobian for $\boldsymbol{\xi}$. Similarly, we define the image Jacobian for $\boldsymbol{\xi}_{\mathbb{I}}$ by

$$\boldsymbol{J}_{\mathbb{I}}(\boldsymbol{q}) = \frac{\partial \boldsymbol{\alpha}_{\mathbb{I}}(\boldsymbol{q})}{\partial \boldsymbol{q}} \tag{9}$$

where

$$\boldsymbol{\alpha}_{\mathbb{I}}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{\alpha}_{\sigma_1}^\top(\boldsymbol{q}) & \boldsymbol{\alpha}_{\sigma_2}^\top(\boldsymbol{q}) & \ldots & \boldsymbol{\alpha}_{\sigma_{|\mathbb{I}|}}^\top(\boldsymbol{q}) \end{bmatrix}^\top. \tag{10}$$

III. Image feature estimation and selection

This section discusses image feature estimation and selection. Section III-A proposes a method to obtain an estimate of $\boldsymbol{\xi}$ from $\boldsymbol{\xi}_{\mathbb{I}}$. Section III-B presents a selection method to have a set of correctly extracted image features.

A. Image feature estimation

We first describe a reconstruction algorithm of generalized coordinates $\boldsymbol{q}$ by using the image Jacobians. Let two vectors $\boldsymbol{\xi}^{\mathrm{o}}$ and $\boldsymbol{q}^{\mathrm{o}}$ which satisfy

$$\boldsymbol{\xi}^{\mathrm{o}} = \boldsymbol{\alpha}(\boldsymbol{q}^{\mathrm{o}}) \tag{11}$$

be given. A linear approximation of (6) near $\boldsymbol{q}^{\mathrm{o}}$ is given by

$$\boldsymbol{J}(\boldsymbol{q}^{\mathrm{o}})(\tilde{\boldsymbol{q}} - \boldsymbol{q}^{\mathrm{o}}) = \boldsymbol{\xi} - \boldsymbol{\xi}^{\mathrm{o}} \tag{12}$$

or

$$\tilde{\boldsymbol{q}} = \boldsymbol{J}^{+}(\boldsymbol{q}^{\mathrm{o}})(\boldsymbol{\xi} - \boldsymbol{\xi}^{\mathrm{o}}) + \boldsymbol{q}^{\mathrm{o}} \tag{13}$$

where $\tilde{\boldsymbol{q}}$ is an approximate value of $\boldsymbol{q}$, and $\boldsymbol{J}^{+}(\boldsymbol{q}^{\mathrm{o}})$ the Moore-Penrose inverse of $\boldsymbol{J}$ at $\boldsymbol{q}^{\mathrm{o}}$. An approximate value of $\boldsymbol{q}$ is also obtained by

$$\boldsymbol{J}_{\mathbb{I}}(\boldsymbol{q}^{\mathrm{o}})(\tilde{\boldsymbol{q}} - \boldsymbol{q}^{\mathrm{o}}) = \boldsymbol{\xi}_{\mathbb{I}} - \boldsymbol{\xi}_{\mathbb{I}}^{\mathrm{o}} \tag{14}$$

or

$$\tilde{\boldsymbol{q}} = \boldsymbol{J}_{\mathbb{I}}^{+}(\boldsymbol{q}^{\mathrm{o}})(\boldsymbol{\xi}_{\mathbb{I}} - \boldsymbol{\xi}_{\mathbb{I}}^{\mathrm{o}}) + \boldsymbol{q}^{\mathrm{o}} \tag{15}$$

where $\boldsymbol{\xi}_{\mathbb{I}}^{\mathrm{o}}$ is defined by replacing $\boldsymbol{\xi}$ with $\boldsymbol{\xi}^{\mathrm{o}}$ in (2). To simplify notation, we write $\boldsymbol{J}$, $\boldsymbol{J}^{+}$, $\boldsymbol{J}_{\mathbb{I}}$ and $\boldsymbol{J}_{\mathbb{I}}^{+}$ instead of $\boldsymbol{J}(\boldsymbol{q}^{\mathrm{o}})$, $\boldsymbol{J}^{+}(\boldsymbol{q}^{\mathrm{o}})$, $\boldsymbol{J}_{\mathbb{I}}(\boldsymbol{q}^{\mathrm{o}})$ and $\boldsymbol{J}_{\mathbb{I}}^{+}(\boldsymbol{q}^{\mathrm{o}})$, respectively, in this section.

We next obtain an estimate of $\boldsymbol{\xi}$ from $\boldsymbol{\xi}_{\mathbb{I}}$. Substituting (15) into (12), we have

$$\hat{\boldsymbol{\xi}} = \boldsymbol{J}\boldsymbol{J}_{\mathbb{I}}^{+}(\boldsymbol{\xi}_{\mathbb{I}} - \boldsymbol{\xi}_{\mathbb{I}}^{\mathrm{o}}) + \boldsymbol{\xi}^{\mathrm{o}}. \tag{16}$$

The vector $\hat{\boldsymbol{\xi}}$ implies an estimate of all of the image features from correctly extracted image features, if $\mathbb{I}$ is a set of correctly extracted image features.

We here define

$$\begin{bmatrix} \hat{\boldsymbol{\xi}}_1^\top & \hat{\boldsymbol{\xi}}_2^\top & \ldots & \hat{\boldsymbol{\xi}}_m^\top \end{bmatrix}^\top = \hat{\boldsymbol{\xi}}, \tag{17}$$

$$\hat{\boldsymbol{\xi}}_i \in \mathbb{R}^{n_i}, \quad i \in \{1, 2, \ldots, m\}. \tag{18}$$

B. Image feature selection

Similar to (16), we can estimate $\boldsymbol{\xi}_{\mathbb{I}}$ from itself by

$$\tilde{\boldsymbol{\xi}}_{\mathbb{I}} = \boldsymbol{J}_{\mathbb{I}}\boldsymbol{J}_{\mathbb{I}}^{+}(\boldsymbol{\xi}_{\mathbb{I}} - \boldsymbol{\xi}_{\mathbb{I}}^{\mathrm{o}}) + \boldsymbol{\xi}_{\mathbb{I}}^{\mathrm{o}}. \tag{19}$$

We here set

$$\begin{bmatrix} \tilde{\boldsymbol{\xi}}_{\sigma_1}^\top & \tilde{\boldsymbol{\xi}}_{\sigma_2}^\top & \ldots & \tilde{\boldsymbol{\xi}}_{\sigma_{|\mathbb{I}|}}^\top \end{bmatrix}^\top = \tilde{\boldsymbol{\xi}}_{\mathbb{I}}, \tag{20}$$

$$\tilde{\boldsymbol{\xi}}_{\sigma_j} \in \mathbb{R}^{n_{\sigma_j}}, \quad \sigma_j \in \mathbb{I}, \quad j \in \{1, 2, \ldots, |\mathbb{I}|\} \tag{21}$$

The vector $\tilde{\boldsymbol{\xi}}_{\sigma_j}$ implies an estimate of $\boldsymbol{\xi}_{\sigma_j}$ from $\boldsymbol{\xi}_{\mathbb{I}}$. If $\tilde{\boldsymbol{\xi}}_{\sigma_j}$ is far from $\boldsymbol{\xi}_{\sigma_j}$, then we may determine that the σ_j-th image feature is not extracted correctly due to occlusion. We therefore obtain a better candidate for a set of correctly extracted image features by the following procedure:

$$\ell = \arg\max_{\sigma_j \in \mathbb{I}} \|\boldsymbol{\xi}_{\sigma_j} - \tilde{\boldsymbol{\xi}}_{\sigma_j}\|, \tag{22}$$

$$\mathbb{I} \leftarrow \mathbb{I} - \ell. \tag{23}$$

If $\|\boldsymbol{\xi}_{\sigma_j} - \tilde{\boldsymbol{\xi}}_{\sigma_j}\|$ is within a given tolerance for all $\sigma_j \in \mathbb{I}$, then $\mathbb{I}$ implies a set of correctly extracted image features

IV. Visual servo control with occlusion handling

This section integrates image feature estimation, image feature selection and visual servo control into vision-based control with occlusion handling.

A. Preliminaries

The vector of the generalized coordinates at time k is denoted by $\boldsymbol{q}(k)$, and the vector of all the image features at time k is written by $\boldsymbol{\xi}(k)$. We use $\boldsymbol{q}(k-1)$ as $\boldsymbol{q}^{\mathrm{o}}$ at time k. We thus use $\boldsymbol{\alpha}(\boldsymbol{q}(k-1))$ as $\boldsymbol{\xi}^{\mathrm{o}}$ at time k, which satisfies (11). The arguments of image Jacobians and Moore-Penrose inverse of them are $\boldsymbol{q}(k-1)$, but the arguments are omitted to simplify notations.

B. Control algorithm

It is assumed here that all the image features are the positions of given markers in the image plane. This assumption is standard in visual servo control. We will discuss other types of image features in Section IV-C.

Our proposed visual servo control algorithm consists of image feature extraction, image feature selection, update of the state, control input calculation and image feature estimation as illustrated in Fig. 1, where the initial state $\boldsymbol{q}(0)$ is given or set to $\boldsymbol{0}$.

Let us explain details of the proposed algorithm.

a) Image feature extraction: Image features at time k are extracted by using an estimate of image features at time $k-1$. For example, $\hat{\boldsymbol{\xi}}_i(k-1)$ is used as the center of a window to extract $\boldsymbol{\xi}_i(k)$. The extraction algorithm can be chosen by the user.

b) Image feature selection: Let the minimum size of $\mathbb{I}$, say m_{s}, and a tolerance ε (> 0) be given. The following algorithm determines the set $\mathbb{I}$:

```
$\mathbb{I} = \{1, 2, \ldots, m\}$
until $|\mathbb{I}| \geq m_{\mathrm{s}}$
    $\tilde{\boldsymbol{\xi}}_{\mathbb{I}}(k) = \boldsymbol{J}_{\mathbb{I}}\boldsymbol{J}_{\mathbb{I}}^{+}\{\boldsymbol{\xi}_{\mathbb{I}}(k) - \boldsymbol{\alpha}_{\mathbb{I}}(\boldsymbol{q}(k-1))\} + \boldsymbol{\alpha}_{\mathbb{I}}(\boldsymbol{q}(k-1))$
    if $\max_{i\in\mathbb{I}} \|\boldsymbol{\xi}_i(k) - \tilde{\boldsymbol{\xi}}_i(k)\| > \varepsilon$
        $\ell = \arg\max_{i\in\mathbb{I}} \|\boldsymbol{\xi}_i(k) - \tilde{\boldsymbol{\xi}}_i(k)\|$
        $\mathbb{I} \leftarrow \mathbb{I} - \{\ell\}$
    else if
        break
    end if
end until
```

c) Update of the state: The state $\boldsymbol{q}(k)$ is given by

$$\boldsymbol{q}(k) = \boldsymbol{J}_{\mathbb{I}}^{+}\{\boldsymbol{\xi}_{\mathbb{I}}(k) - \boldsymbol{\alpha}_{\mathbb{I}}(\boldsymbol{q}(k-1))\} + \boldsymbol{q}(k-1). \quad (24)$$

d) Control input calculation: A control input signal $\boldsymbol{u}(k)$ is derived by using $\boldsymbol{q}(k)$. The concrete algorithm can be chosen by the user. In particular, if

$$\boldsymbol{u} = \dot{\boldsymbol{q}}, \quad (25)$$

then one of standard methods is

$$\boldsymbol{u}(k) = -\lambda\{\boldsymbol{q}(k) - \boldsymbol{q}(k-1)\} \quad (26)$$

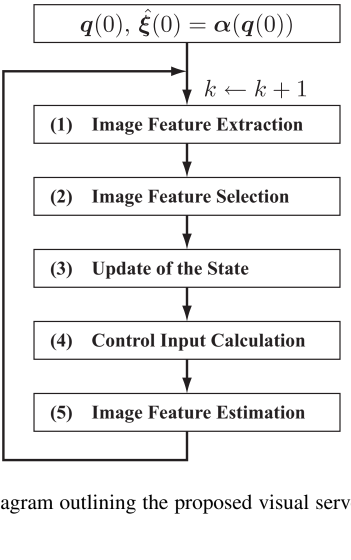
$\boldsymbol{q}(0)$, $\hat{\boldsymbol{\xi}}(0) = \boldsymbol{\alpha}(\boldsymbol{q}(0))$
$k \leftarrow k+1$
(1) Image Feature Extraction
(2) Image Feature Selection
(3) Update of the State
(4) Control Input Calculation
(5) Image Feature Estimation

Fig. 1. A flow diagram outlining the proposed visual servo control system.

where λ is a gain. In the next section, a PID controller is implemented.

e) Image feature estimation: Using (16), image features are estimated by

$$\hat{\boldsymbol{\xi}}(k) = \boldsymbol{J}\boldsymbol{J}_{\mathbb{I}}^{+}\{\boldsymbol{\xi}_{\mathbb{I}}(k) - \boldsymbol{\alpha}(\boldsymbol{q}(k-1))\} + \boldsymbol{\alpha}(\boldsymbol{q}(k-1)). \quad (27)$$

C. discussion

We first discuss the image feature selection.

The selection step eliminates the index i such that the error between the measurement $\boldsymbol{\xi}_i$ and the estimate $\tilde{\boldsymbol{\xi}}_i$ is biggest until the biggest error is within the given tolerance. The set $\mathbb{I}$ obtained in the method satisfies

$$\|\boldsymbol{\xi}_i - \tilde{\boldsymbol{\xi}}\| \leq \varepsilon, \quad \forall i \in \mathbb{I} \quad (28)$$

or

$$\|\boldsymbol{\xi}_i - \tilde{\boldsymbol{\xi}}\| \leq \|\boldsymbol{\xi}_i - \tilde{\boldsymbol{\xi}}\|, \quad \forall i \in \mathbb{I}, \quad j \notin \mathbb{I}. \quad (29)$$

We can make additional pre-selections before the image feature selection in order to improve the accuracy of the set $\mathbb{I}$. Three typical examples are shown below: (a) If $\tilde{\boldsymbol{\xi}}_i(k-1) \notin \mathbb{X}_i$, then

$$\mathbb{I} \leftarrow \mathbb{I} - \{i\}. \quad (30)$$

(b) If windows for extracting the i-th and the j-th image features overlap with each other in the image plane, then

$$\mathbb{I} \leftarrow \mathbb{I} - \{i, j\}. \quad (31)$$

(c) If an image moment such as the area of an image feature is without a tolerance, then

$$\mathbb{I} \leftarrow \mathbb{I} - \{i\}. \quad (32)$$

Using many selections increases computational complexity. This leads to large time delay, which have a considerable influence in real-time control systems. We have to implement appropriate selections taking into account the tradeoff between accuracy and time delay.

We then discuss the type of image features.

In the previous section, it is assumed that all the image features are the positions of given markers. In this case, the estimated image features are used as the centers of windows for extracting image features. Thus, the proposed method can find a failed marker again, after the marker is visible.

Let us consider the case where an image feature is not the position of a marker. In this case, we require two types of image features for the marker. One of the image features is the original image feature, and the other is the position of the marker in the image plane. Using the position for visual tracking, we obtain the set $\mathbb{I}$ at each time k. We can update the state $\boldsymbol{q}(k)$ by using the obtained $\mathbb{I}$ and the original image features.

V. Experimental result

This section demonstrates the proposed visual servo control with occlusion handling.

A. Experimental setup

The system considered in this section consists of a small helicopter, four stationary cameras, three computers, a DA converter and a transmitter as illustrated in Fig. 2 (see [13] for a single-camera and wired version of this system). The helicopter does not have any sensors which measure the position or posture. It has four small black balls, and they are attached to rods connected to the bottom of the helicopter. The black balls are labeled from 1 to 4. Their positions in the image planes are image features. The four cameras are placed on the ground and they look upward. The cameras are labeled from 1 to 4. Each two of four cameras are connected to a computer. The two computers linked to cameras are called client computers. The computer linked to the DA converter is called server computer.

Client computers execute image feature extraction in Fig. 1. Each camera captures a gray-scale image, and each client computer binarizes two of the images at each time. The image features are extracted in windows whose centers are set to the estimate derived from (27). The estimate is calculated in server computer, and it is transfered to client computers. Client computes take 8.5 milli-seconds to extract image features. This follows from the use of fast IEEE 1394 cameras, Dragonfly Express, developed by Point Grey Research Inc. The values of the extracted image features are transfered to server computer.

Server computer executes image feature selection, update of the state, control input calculation and image feature estimation. Details of control input calculation are described in Section V-C. The obtained control signals are supplied from server computer to the helicopter through the DA converter and the transmitter. It takes 8.5 milli-seconds to update the control signals.

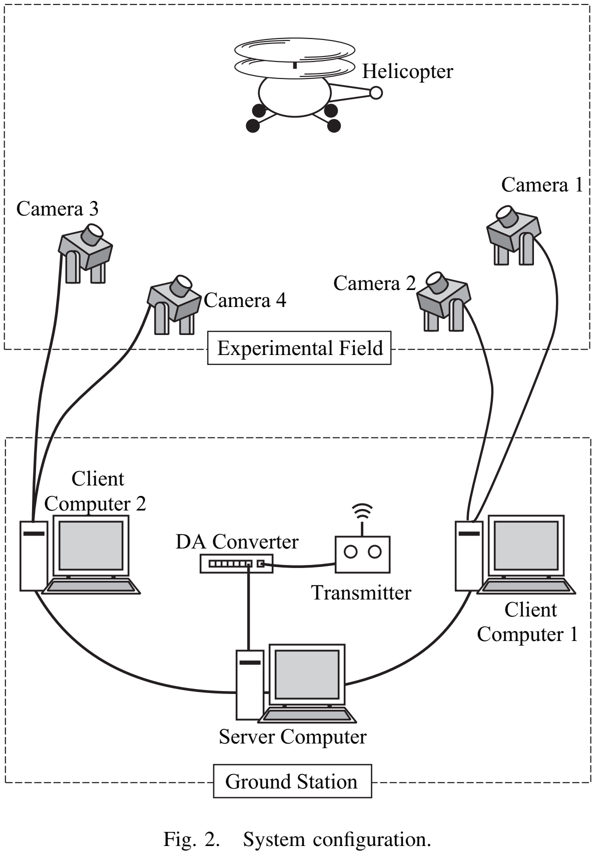

Fig. 2. System configuration.

The small helicopter used here is X. R. B SR SKY ROBO Shuttle developed by HIROBO (see Fig. 3). It has a coaxial rotor configuration. The two rotors share the same axis, and they rotate in opposite directions. The tail is a dummy. A stabilizer is installed on the upper rotor head. It mechanically keeps the posture horizontal.

Table I summarizes specifications of the system.

B. Coordinate frames

Let Σ^{w} be the world reference frame. The z^{w} axis is directed vertically downward. A coordinate frame Σ^{b} be attached to the helicopter body as illustrated in Fig. 4. The $(x^{\mathrm{b}}, y^{\mathrm{b}}, z^{\mathrm{b}})$ positions of the four black balls in the frame Σ^{b} are given by

$$\begin{bmatrix} 0.1 \\ 0.1 \\ 0.04 \end{bmatrix}, \begin{bmatrix} -0.1 \\ 0.1 \\ 0.04 \end{bmatrix}, \begin{bmatrix} 0.1 \\ -0.1 \\ 0.04 \end{bmatrix}, \begin{bmatrix} -0.1 \\ -0.1 \\ 0.04 \end{bmatrix} \qquad (33)$$

Let $\boldsymbol{\xi}_{4(i-1)+j}$ denote the j-th ball position in the image plane of the i-th camera.

The helicopter position relative to the world reference frame Σ^{w} is denoted by (x, y, z). The roll, pitch and yaw angles are denoted by ψ, θ, ϕ, respectively. Recall that the helicopter has the horizontal-keeping stabilizer. Both the angles θ and ψ converge to zero fast enough even when the body is inclined. Thus we may suppose that

$$\theta(k) = 0, \ \psi(k) = 0, \ \forall \ k \geq 0, \qquad (34)$$

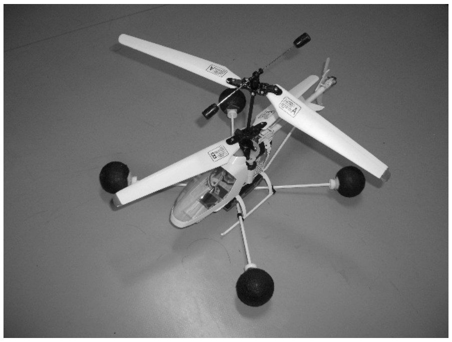

Fig. 3. X.R.B. with four black balls.

TABLE I

SPECIFICATIONS OF THE SYSTEM.

Length of the helicopter,	0.40 [m].
Height of the helicopter,	0.21 [m].
Rotor length of the helicopter,	0.34 [m].
Weight of the helicopter,	0.21 [kg].
Focal length of the lens,	4 [mm].
Camera resolution,	640 × 480 [pixels].
Pixel size,	7.4 [μm] × 7.4 [μm].

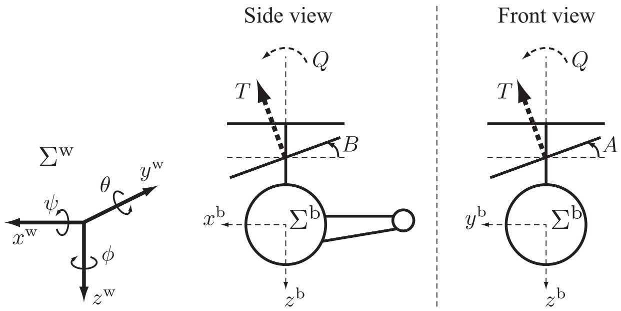

Fig. 4. Coordinate frames.

in practice. The vector of generalized coordinates of the helicopter is given by

$$\boldsymbol{q} = \begin{bmatrix} x & y & z & \phi \end{bmatrix}^{\top}. \tag{35}$$

The following four variables are individually controlled by control signals (see Fig. 4):

B : Elevator. Pitch angle of the lower rotor.

A : Aileron. Roll angle of the lower rotor.

T : Throttle. Resultant force of the two rotor thrusts.

Q : Rudder. Difference of the two torques generated by the two rotors.

The corresponding control signals are denoted by u_1, u_2, u_3 and u_4. Note that x, y, z and ϕ are controlled by applying u_1, u_2, u_3 and u_4, respectively.

C. Controller design

The aim here is that $\boldsymbol{q}(k) \to \boldsymbol{0}$ as $k \to \infty$. We define

$$\begin{bmatrix} \bar{q}_1(k) & \bar{q}_2(k) & \bar{q}_3(k) & \bar{q}_4(k) \end{bmatrix} = \boldsymbol{R}(\phi(k))\boldsymbol{q}(k) \tag{36}$$

where

$$\boldsymbol{R}(\phi) = \begin{bmatrix} \cos\phi & \sin\phi & 0 & 0 \\ -\sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{37}$$

The control signals are given by a set of PID controllers of the form

$$\begin{aligned} u_i(k) = b_i + P_i\bar{q}_i(k) + I_i \sum_{0}^{k} \bar{q}_i(k) \\ + D_i(\bar{q}_i(k) - \bar{q}_i(k-1)), \end{aligned} \tag{38}$$

where b_i, P_i, I_i and D_i are constants for $i = 1, \ldots, 4$. They are heuristically tuned to the values in Table II. The signs of the gains depend on the property of the transmitter.

D. Result

The altitude of the origin of the world reference frame Σ^{w} is 1.0 [m] above the ground. The $x^{\mathrm{w}}y^{\mathrm{w}}$ plane is horizontal. The cameras are located as shown in Table III (We first set camera i at the origin of Σ^{w} temporarily. The optical axis lies along x^{w} axis. We then move camera i in accordance with Table III.) We set $m_{\mathrm{s}} = 4$ and $\varepsilon = 15$ [pixels].

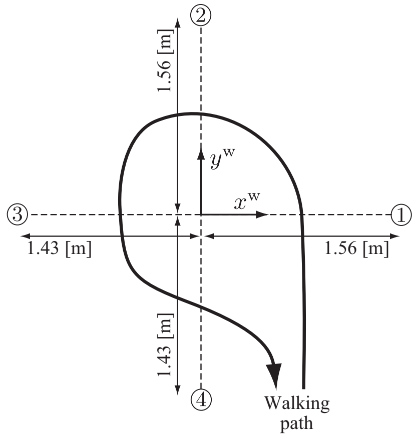

Fig. 5. Camera locations and a walking path. The symbol ⓘ stands for the position of camera i.

After hovering, the proposed visual servo control was implemented. A person walked in the experimental field as illustrated in Fig. 5. Occlusion in the experiment can not be predicted.

The top figure in Fig. 6 shows time profiles of elements of $\mathbb{I}$. It is seen that $\boldsymbol{\xi}_i \notin \mathbb{I}$ for $i = 1, \ldots, 4$ at around 1 [sec.]. Fig. 6 also illustrates that occlusion was detected in camera 2, 3 and 4 at around 4, 7 and 10 [sec.], respectively. It is seem from Figs. 5 and 6 that occlusion was detected in camera i when the person walked between the camera and the helicopter. In addition, image features were tracked again, when he left from the front of the camera.

The bottom two figures in Fig. 6 are closeups of the top figure. The figures show that occlusion was detected in camera 2 unrelated to the motion of the walking person. This follows from the background color in the image. The image features $\boldsymbol{\xi}_6$ and $\boldsymbol{\xi}_8$ were over a room light in the image captured by camera 2. Thus client computer sometimes fails to extract the image features correctly. Our proposed visual servo control is robust to such image outliers.

Fig. 7 shows time profiles of the generalized coordinates $\boldsymbol{q}$. The helicopter hovered in a neighborhood of $\boldsymbol{0}$ even when occlusion occurred. In particular, the z position was within 2 [cm] for all time. The helicopter was controlled well.

TABLE II

PID GAINS.

	b_i	P_i	I_i	D_i
u_1	2.02	2.00	0.10	1.5
u_2	2.76	−2.00	−0.10	−1.5
u_3	1.85	−2.40	−1.50	−1.8
u_4	2.53	−1.50	−0.80	−0.1

TABLE III

CAMERA CONFIGURATION.

	x	y	z	ϕ	θ
camera 1	0.00	1.00	1.13	$\pi/2$	$5\pi/18$
camera 2	0.80	−1.20	1.21	$4\pi/3$	$\pi/6$
camera 3	−1.20	−0.70	1.12	$11\pi/6$	$\pi/9$
camera 4	−1.20	0.00	1.13	0	$2\pi/9$

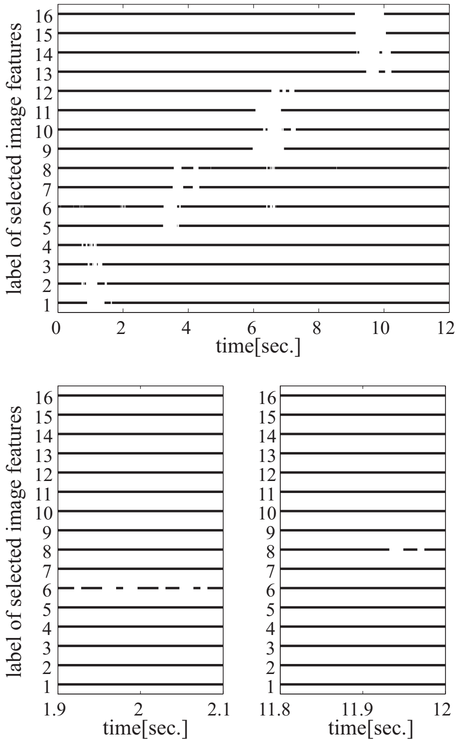

Fig. 6. Experimental result: The label of the selected image features. Lines are drawn for i when $i \in \mathbb{I}$ at each time. The bottom two are closeups of the top figure.

VI. Conclusion

This paper has proposed a visual tracking method with occlusion handling for visual servo control. In particular, image feature selection, image feature estimation and visual servo control have been integrated into a vision-based control algorithm. The proposed control algorithm chooses a set of correctly extracted image features, and it then estimates all image features from the correctly extracted image features. The algorithm has low computational complexity, since the image Jacobian is used for image feature extraction and estimation. In addition, even when the algorithm fails to track a moving image feature, it does not need a global search over the entire image plane to find the failed image feature. As a result, it can track the failed image feature again quickly. The validation of the proposed algorithm has been demonstrated in an experiment. Occlusion in the experiment can not be predicted. Several movies can be seen at [14].

References

[1] F. Chaumette and S. Hutchinson, “Visual servo control, Part I: Basic approaches,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[2] ——, “Visual servo control, Part II: Advanced approaches,” *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[3] E. Malis and S. Benhimane, “A unified approach to visual tracking and servoing,” *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 39–52, 2005.

[4] S. Benhimane and E. Malis, “Vision-based control with respect to planar and non-planar objects using a zooming camera,” in *IEEE International Conference on Advanced Robotics*, 2003.

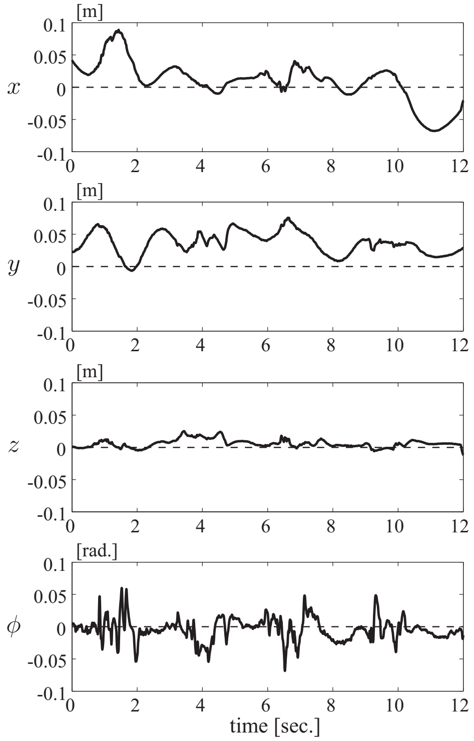

Fig. 7. Experimental result: Time profile of the generalized coordinates.

[5] G. Chesi, K. Hashimoto, D. Prattichizzo, and A. Vicino, “Keeping features in the field of view in eye-in-hand visual servoing: a switching approach,” *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 908–914, 2004.

[6] Y. Mezouar and F. Chaumette, “Optimal camera trajectory with image-based control,” *The International Journal of Robotics Research*, vol. 22, no. 10, pp. 781–804, 2003.

[7] V. Lippiello, B. Siciliano, and L. Villani, “An occlusion prediction algorithm for visual servoing tasks in a multi-arm robotic cell,” in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2005.

[8] V. Lippiello and L. Villani, “Managing redundant visual measurements for accurate pose tracking,” *Robotica*, vol. 21, no. 5, pp. 511–519, 2003.

[9] V. Lippiello, “Real-time visual tracking based on BSP-tree representations of object boundary,” *Robotica*, vol. 23, no. 3, pp. 365–375, 2005.

[10] A. Comport, M. Pressigout, E. Marchand, and F. Chaumette, “A visual servoing control law that is robust to image outliers,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[11] N. García-Aracil, E. Malis, R. Aracil-Santonja, and C. Pérez-Vidal, “Continuous visual servoing despite the changes of visibility in image features,” *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1214–1220, 2005.

[12] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley, 2005.

[13] K. Watanabe, Y. Yoshihata, Y. Iwatani, and K. Hashimoto, “Image-based visual PID control of a micro helicopter using a stationary camera,” *Advanced Robotics*, in press.

[14] http://www.ic.is.tohoku.ac.jp/E/research/helicopter/.