6DOF Stereo Visual Servoing by Photo-Model-Based Pose Estimation

指 導 教 員　　見 浪 護 教 授

岡 山 大 学 大 学 院 自 然 科 学 研 究 科

博 士 後 期 課 程 産 業 創 成 工 学 専 攻

（令 和 元 年 度）

学 籍 番 号　51429306　　田 宏 志

6DOF Stereo Visual Servoing by Photo-Model-Based Pose Estimation

Intelligent Robotics and Control Laboratory

Hongzhi Tian (51429306)

Abstract:

Since robots have higher reliability and accuracy than humans, they have been used extensively in production factories to perform a wide variety of tasks instead of human workers. Moreover, humans have mobilized robots to perform repetitive and dangerous jobs that are required to be conducted in exceptional environments such as outer space—the universe beyond the earth's atmosphere—or the bottom of the sea. However, until now, robots cannot entirely replace humans. Humans can perform the intended tasks in a specific environment, and automated robots do not reach human adaptability. Therefore, researchers have been trying to improve the adaptive capabilities of autonomous robots. Concerning autonomous robots, a robot control technology using visual information obtained from cameras in the feedback loop, which is named as visual servoing, is expected to be able to allow the robot to adapt to changing or unknown environments. However, for a robot with vision sensors, such as cameras, it has been difficult till now to accurately detect the 3D pose (position and orientation) of the target object, especially if the target object cannot be predefined since the shape is arbitrary.

A model-based method is a way used to realize the visual servoing. Even though it enables a monocular vision to detect the distance of the target object from a single image, its accuracy is not enough. Many studies have used RGB-D camera, composed of one RGB camera and depth sensor with infrared light, to improve the distance detection capabilities of monocular vision. These studies still rely on the target detection in a single image or segmentation. Although the RGB-D camera generates a depth point cloud corresponding to the single image the pose estimation accuracy should be improved for practical applications. Therefore, some studies use

deep learning methods for target detection. However, this requires a lot of pictures and pre-training time. Some studies use a model-based approach to simplify preparatory work. Still, it cannot avoid the disadvantage that its depth distance measurement is inaccurate. Unlike the RGB-D method, stereo vision is another possible approach to estimate 3D pose.

The author has proposed a stereo vision visual servoing system that uses a 3D model for the target's pose detection. The adoption of 3D model to estimate the target's pose enables to improve the 3D pose estimation accuracy. However, the process to construct 3D model in programming was complicated. For producing industrial parts, some other researches use CAD models because they are readily available from their design phase. Commercial CAD packages can help shorten the conversion process from CAD model to a model for pose detection. However, for general objects appearing in people's daily life, e.g., deformable clothes, it is impossible to describe them in the CAD model. In order to respond to new requirements and overcome the disadvantages encountered in constructing 3D models, photo-model-based clothes processing robot has been developed for picking and placing clothes. It simplified the model making process since this process does not need to predefine the object's size, shape, color, pattern, and design in programming language. And more importantly, it can deal with deformable goods.

This thesis proposes a real-time 6DOF photo-model-based pose estimation method used for 6DOF visual servoing purposes. This method can detect the full pose of a 3D target object. To the best of the author's knowledge, no studies have yet been conducted on 3D pose visual servoing with only 2D photo-model of an object in the real world. What the author wants to certify by real experiments in this paper is whether a 2D photo-model generated from one photo of a 3D target can estimate the full 6DOF pose of the 3D target, and whether the estimated pose can be used for 3D pose visual servoing. And the results have shown that the full pose of an arbitrary 3D target can be estimated in real-time by using only a 2D photo and it also enables 3D visual servoing to the target. The above results have been confirmed by real experiments that use a 6DOF manipulator with stereo vision at the end-effector.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since robots have higher reliability and accuracy than humans, they have been used extensively in production factories to perform a wide variety of tasks instead of human workers. Moreover, humans have mobilized robots to perform repetitive tasks or dangerous jobs that are required in exceptional environments, such as outer space or the bottom of the sea. However, robots cannot entirely replace humans because they lack human adaptability. Therefore, researchers have been trying to improve the adaptive capabilities of autonomous robots.

Visual information is useful for an autonomous robot to perceive its surrounding environment. The field of robot vision researches how to use visual information to enable robots to perform some given tasks [1]–[5]. In this field, robot control technology uses visual information obtained from cameras in a feedback loop known as "visual servoing" to allow robots to adapt to changing or unknown environments [6]–[13]. However, a robot with vision sensors, such as cameras, has difficulties detecting the 3D pose of target objects accurately, especially if the target object cannot be predefined since the shape is arbitrary.

A model-based method is a way to meet the above challenges and used to realize visual servoing [14]–[16]. The author's research results in the field of model-based visual servoing will be introduced in this thesis.

1.1 Background and motivation

For the development of a robot vision system, it is important to choose a suitable camera configuration. Even though the model-based method enables a monocular vision to detect the distance of the target object from a single image, its accuracy is not enough [17]–[19]. Many studies have used an RGB-D camera, composed of one RGB camera and a depth sensor with infrared light, to improve distance detection capabilities to monocular vision [12],[20]–[22]. However, the RGB-D camera has a major disadvantage: missing depth data caused by the depth sensor. Some pixels do not have corresponding depth data [23]. What's more, bright ambient illumination can affect the contrast of infrared images in active light sensors, resulting in outliers or holes in the depth map [24]. Unlike optical infrared and electric-field sensing, stereo vision perceives a greater variety of target material properties and light conditions [25]. It is not dependent on capacitance, reflectivity, or other material properties, as long as the target surface has some visible features. For the above reasons, the research proposed in this paper is based on stereo vision, i.e., dual RGB cameras.

The author has proposed a model-based eye-vergence visual servoing system that uses a 3D model for the target's pose detection [26]. The adoption of a 3D model to estimate the target's pose enables the system to improve the 3D pose estimation accuracy. However, the process of constructing 3D models in programming was complicated. For producing industrial parts, some other studies use CAD models, because they are readily available from their design phase. Commercial CAD packages can help shorten the conversion process from CAD model to a model for pose detection [27]. However, for general objects appearing in people's daily lives, e.g., deformable clothes, it is impossible to describe them in the CAD model.

To overcome the disadvantages and difficulties of building 3D models, some studies based on 2D models have used QR-code or other artificial 2D markers for visual servoing tasks [28]. Some data-driven methods with deep learning techniques use 2D pictures to detect 3D poses of target objects, they require a large number of pictures and pre-training time [20],[21]. Their

application ranges are limited. To establish a more general and practical recognition approach, a photo-model-based clothes processing robot has been developed for picking and placing clothes using 4DOF detection results of the 2D photo-model [29]–[32]. This photo-model greatly simplified the model making process since this process did not need to define the object's size, shape, color, pattern, and design in the programming language. We hope that the photo-model-based technology will not be limited to clothes-handling, but have a broader range of applications, e.g., pursuing and catching aquatic animals that try to escape the visual servoing of underwater robots. Compared to static clothes, the poses of these or other animals will change when they are moving. The photo-model-based recognition method needs to be improved to be applicable to real-time pose tracking.

This thesis proposes a real-time 6DOF photo-model-based pose estimation method used for 6DOF visual servoing purposes. This method can detect the full pose of a 3D target object. To the best of the author's knowledge, no studies have been conducted on 3D pose visual servoing with only 2D photo-model of an object in the real world. The author wants to evaluate whether a 2D photo-model generated from one photo of a 3D target can estimate the full 6DOF pose of the 3D target and whether the estimated pose can be used for 3D pose visual servoing. Results show that the full pose of a 3D target can be estimated in real-time by using only a 2D photo, which enables 3D visual servoing of the target. The above results have been confirmed by real experiments that use a 6DOF manipulator with stereo vision at the end-effector.

In this paper, firstly, 6DOF model-based recognition method is introduced as necessary reference technology. Then, the proposed 6DOF photo-model-based recognition method is presented in detail. In the end, to confirm the tracking capability of the proposed recognition method, frequency response experiments to track a target have been conducted using a stereo vision hand-eye robot. The results show that the robot can track a given target in real-time with its photo-model and completed the visual servoing task. Furthermore, to verify this capability in a more realistic environment, the photo-model-based visual servoing system is used to track a marine creature toy floating on the pool surface without pose restrictions. It is confirmed from

the experimental results that the visual servoing robot can be used to capture a moving marine creature target and is not susceptible to partial occlusion conditions. Hence, these show that the 2D photo-model got from one photo can estimate the pose of the 3D target.

1.2 Aim and objectives

The overall aim of the research presented in this thesis is to develop a real-time 6DOF photo-model-based pose estimation method for visual servoing application. To achieve this aim, the following objectives should be fulfilled:

- to develop a real-time 3D pose estimation method with a 2D photo.
- to develop a visual servoing system using the proposed photo-model-based pose estimation method.
- to verify the real-time pose tracking capability of the visual servoing system by conducting frequency experiment.

1.3 Thesis contribution

Figure 1.1 shows the 4DOF photo-model-based handling robot (pick and place) introduced by our previous studies [29]–[32]. The proposed system aims at picking up clothes after a robot recognizes it and classifies the clothes into a collection box. The robot has been confirmed to be able to identify 12 different deformable clothes [31],[33].

Referring to the real-time pose estimation technology of the model-based eye-vergence visual servoing robot shown as Fig. 3.17 [34], this paper extends the past 4DOF photo-model-based method to a real-time 6DOF recognition method for detecting the full pose of a 3D solid target object. As shown in Fig. 1.2, a photo-model is used for visual servoing so that a visual servoing robot (VS-robot) can follow a 3D target whose motion is given by a target control robot (TC-robot), There is no communication between the two robots except vision information.

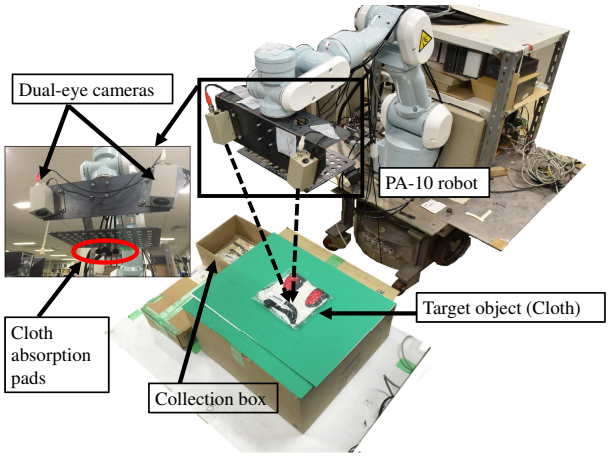


Fig. 1.1: A photo of the clothes handling robot system with dual-eye cameras: PA-10 robot is equipped with a vacuum unit and two cameras used as stereo vision, where four pads connected with the air compressor made the robot possible to perform the pick (absorption) and place of the clothes. In the test, the robot picked up 12 kinds of deformable clothes and classified them, and set them into the collection box.

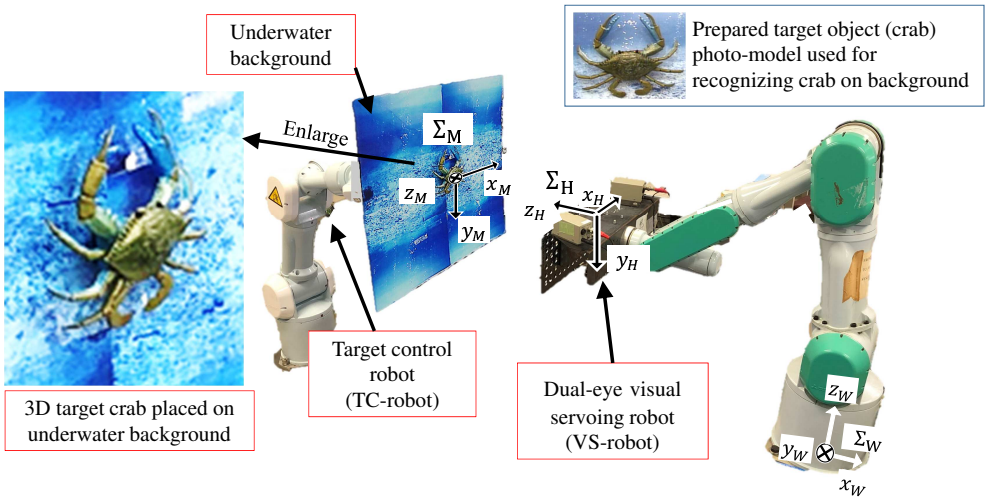


Fig. 1.2: The motion of the target animal, crab, is given by TC-robot, and the VS-robot moves to keep desired relative pose of the VS-robot against the crab attached on a panel with sea bottom backdrop whose motion is given by TC-robot. World coordinate system Σ_W, hand coordinate system Σ_H, and target coordinate system Σ_M are depicted in the figure.

In summary, the contributions of this paper are listed as follows.

- A method is proposed to estimate the pose of a 3D target object by using stereo vision and only one 2D photo.
- With the proposed pose estimation method a photo-model-based visual servoing robot (Fig. 1.2) is developed. Stereo vision cameras fixed at the end-effector of the VS-robot perform the real-time pose estimation of the 3D target based on its photo-model.
- The developed system's visual servoing abilities to a moving 3D target have been confirmed through frequency response experiments.

All above points have helped achieve the photo-model-based visual servoing.

1.4 Dissertation structure

This thesis is organized as follows:

Chapter 2 presents a literature review on robot vision, vision-based approaches, basic classification of visual servoing, and stereo vision.

Chapter 3 describes the model-based recognition method with detailed explanation on stereo vision geometry, 3D model-based matching and genetic algorithm (GA). And the developed eye-vergence visual servoing system is introduced.

Chapter 4 describes experiments of the model-based eye-vergence visual servoing system. Through these experiments, the tracking ability of the eye-vergence visual servoing system is verified. Comparing with the fixed camera vision, advantage of the eye-vergence vision is confirmed.

Chapter 5 presents the proposed 6DOF photo-model-based pose estimation method with 2D photo. In the early stage of development, it is developed based on the fixed camera

vision. In the future, it will be used on the eye-vergence vision. Photo-model generation and 3D matching will be introduced in detail.

Chapter 6 describes the real-time 3D pose estimation and visual servoing experimental results, followed by discussion and conclusion.

Chapter 7 concludes this thesis.

1.5 Publications

The research work presented in thesis has resulted in the following publications.

Journals

(1) "Evaluation of eye-vergence visual servoing by lateral frequency response," (in Japanese 横軸方向周波数応答実験による両眼転導ビジュアルサーボの評価), Hongzhi Tian, Yejun Kou, and Mamoru Minami, Transactions of the JSME, Vol.84, No.857, DOI: 10.1299/transjsme.17-00182 (2018)

(2) "Frequency Response Experiments of Eye-Vergence Visual Servoing in Lateral Motion with 3D Evolutionary Pose Tracking," Hongzhi Tian, Yu Cui, Mamoru Minami, Akira Yanou, Artificial Life and Robotics, Vol.22, No.1, pp.36-43 (2017)

International Conferences

(1) "Visual Servoing to Arbitrary Target with Photo-Model-Based Recognition Method," Hongzhi Tian, Yejun Kou, Mamoru Minami, 24th International Symposium on Artificial Life and Robotics, B-Con Plaza, (Beppu, Japan), pp.950-955 (2019)

(2) "Photo-Model-Based Stereo-Vision 3D Perception for Marine Creatures Catching by ROV," Hongzhi Tian, Yejun Kou, Takuro Kawakami, Renya Takahashi, Mamoru Minami, OCEANS 2019 Seattle, Washington State Convention Center, (Seattle, America) (MTE/IEEE), 55789150 (2019)

(3) “Robust Translational/Rotational Eye-Vergence Visual Servoing under Illumination Varieties,” Hongzhi Tian, Yejun Kou, Khaing Win Phyu, Daiki Yamada, Mamoru Minami, IEEE International Conference on Robotics and Biomimetics, (Macau, China), pp.2032-2037 (2017)

(4) “Tracking Performances of Eye-Vergence Visual Servoing System under Different Light Condition with Evolutionary Pose Tracking,” Hongzhi Tian, Yejun Kou, Ryuki Funakubo, Mamoru Minami, International Symposium on System Integration, (Sapporo, Japan) (IEEE/SICE), pp.568-573 (2016)

(5) “3D Evolutionary Pose Tracking Experiments of Eye-Vergence Visual Servoing in Lateral Motion and Arc Swing Motion,” Hongzhi Tian, Ryuki Funakubo, Yejun Kou, Mamoru Minami. In 2016 IEEE International Conference on Robotics and Biomimetics, (Qingdao, China), pp.577-582 (2016)

(6) “Visual Servoing Frequency Response of Eye-vergence System in Lateral Motion with Evolutionary Pose Tracking of 3D-Object,” Hongzhi Tian, Yu Cui, Mamoru Minami, Akira Yanou, 21st International Symposium on Artificial Life and Robotics, B-Con Plaza (Beppu, Japan), pp.658-663 (2016)

National Conferences and Poster Presentation

(1) “Visual Servoing to Arbitrary Target by Using Photo-Model Definition,” Hongzhi Tian, Yejun Kou, Khaing Win Phyu, Ryuki Funakubo, Mamoru Minami, JSME ロボティクス・メカトロニクス講演会 (ROBOMECH 2018), 2A1-M17 (2018)

(2) “Eye-Vergence に基づくビジュアルサーボシステム,” 田 宏志, 侯 森, 見浪 護, 于 福佳, 前田 耕一, 矢納 陽, SICE, 第 7 回 CI 研究会, 宮城県仙台市東北大学, pp.25-32 (2015)

(3) “Eye-Vergence を用いたビジュアルサーボシステムの 6 自由度を持つ 3 次元マーカへの追従特性,” 田 宏志, 崔 禹, 見浪 護, 新木 遼平, 矢納 陽, 第 58 回自動制御連合講演会, 1I2-2 (2015)

Chapter 2

Literature Review

A literature review on some background topics is introduced in this section, relating to this study and research for configurations of robot vision, visual servo, stereo vision, and object recognition techniques.

2.1 Computer vision, machine vision, and robot vision

Computer vision (CV) has a dual purpose. From a biological science perspective, computer vision aims to propose a computational model of the human visual system. From an engineering perspective, computer vision is designed to build an autonomous system that can perform specific tasks that human vision systems can achieve (in many cases even exceeding them) [35]. Computer vision aims to use cameras to analyze or understand scenes in the real world. This discipline studies methodological and algorithmic issues, as well as topics related to the realization of design solutions. In CV, people might want to know if a vehicle is driving in the center of the lane, how many people are in the scene, or may even want to identify a specific person - all of which can be answered based on recorded images or videos [36].

The differences between computer vision and machine vision are analyzed in detail in [37]. Machine vision (MV) is concerned with the engineering of integrated mechanical-optical-electronic software systems for examining natural objects and materials, human artifacts, and

manufacturing processes in order to detect defects and improve quality, operating efficiency, and the safety of both products and processes. It is also used to control machines used in manufacturing. [37].

To summarise, the division between MV and CV reflects the separation between engineering and science. Machine vision systems perform quality tests, guide machines, control processes, identify components, read codes, and deliver valuable data for optimizing production. [38]. Machine vision must involve the harmonious integration of mechanical handling, lighting, optics, video cameras, image sensors (including visible, infrared radiation, X-ray sensor arrays, or laser scanners), industrial engineering, human-computer interfacing, control systems, manufacturing, and quality assurance methods. Machine vision is not a scientific endeavor; it is a branch of systems engineering. [37].

The terms robot vision (RV) and machine vision are usually used interchangeably [2]. However, there are some subtle differences between them. Some machine vision applications, such as part inspections (in which parts are placed just in front of the vision sensor to look for faults), have nothing to do with robotics. [5]. Moreover, RV is not just an engineering field. It is a science with its particular area of research. Unlike pure computer vision research, robot vision must incorporate the aspects of robotics into its technologies and algorithms, such as kinematics and the physical impact of robots on the environment [39]. Visual servoing is a perfect example of technology that can only be called robot vision, not computer vision [5].

It must be understood that CV, MV, and RV share many terms, concepts, and algorithmic techniques, but they have different goals and have different priorities to deal with problems.

2.2 Visual tracking and visual servoing

2.2 Visual tracking, also named object tracking, is an important task within the field of computer vision [40],[41]. It can be applied to many domains, such as visual surveillance [42], human-computer interaction, and video compression [43].

In its simplest form, tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene. In other words, a tracker assigns consistent labels to the tracked objects in different frames of a video, either in the 2D image plane or in the 3D object space [44].

Visual tracking essentially deals with non-stationary data, both the target object and the background, that change over time [45]. Visual tracking of an object involves the detection of some known object features in the acquired images and the estimation of the object's position and orientation (pose) with these features [46]. The target object's pose estimation is very important for robot motion. Therefore, not only in CV, but also in RV, a visual tracking system is essential as a basis for visual servoing, autonomous navigation, path planning, robot-human interaction, and other robotic functions [47]. However, it does not involve robot control.

Visual servoing is a robot control technology that guides robots with real-time and continuous visual feedback [6],[46]. Visual servo control refers to the use of computer vision data to control the motion of a robot, and relies on techniques from image processing, computer vision, and control theory [48]. Visual data can be obtained from a camera mounted directly on the robot manipulator or mobile robot, in which case the movement of the robot will cause the camera to move [48]. Therefore, the nonlinear dynamic influence of the entire robotic arm will affect the stable tracking ability of the hand-eye visual servoing system [49],[50]. Visual servoing is a perfect example of robot vision as opposed to computer vision [5]. Due to the high requirements of visual servoing, a visual tracking technique may not be directly applicable to visual servoing.

2.3 Classification of visual servoing

This section introduces the basic classification of the visual servoing research based on control schemes and tracking approaches.

2.3.1 Visual servoing control schemes

The two archetypal visual servo control schemes are image-based and position-based visual servo control [51]. Some studies combine the image-based and position-based methods; therefore, visual servo control techniques are broadly classified into three major groups [9]:

1. position-based [52]–[54] (pose-based called in [51],[55]) visual servoing (PBVS),
2. image-based [56]–[59] visual servoing (IBVS),
3. hybrid visual servoing [60] (combining PBVS and IBVS).

In IBVS, the control law is based on the error between current and desired features on the image plane, and does not involve any estimate of the target's pose. The features may be the coordinates of visual features, lines, or moments of regions. This is servoing in 2D.

In PBVS, the pose of the target object is estimated with respect to the camera, and then a command is issued to the robot controller, which in turn controls the robot. In this case the image features are extracted as well, but are additionally used to estimate 3D information (the pose of the object in Cartesian space). A kinematic error is generated in the Cartesian space and mapped to actuators' commands [61]. This is servoing in 3D.

The advantages and drawbacks of each visual servoing method have been discussed in a significant amount of studies [51],[61]. Compared with image-based visual servoing, position-based visual servoing is more understandable, since the method of the visual servo is more similar to a human being; that is, it determines the object pose in the Cartesian coordinate frame and leads to Cartesian robot motion planning. Moreover, in position-based visual servoing, the robot controller and object pose recognition are separated as independent units.

2.3.2 Tracking approaches for visual servoing purpose

To highlight research characteristics, many studies are named after tracking techniques instead of control schemes (e.g., model-based visual servoing) [62],[63]. Most of the available tracking

techniques can be divided into two main classes [64]:

1. feature-based approaches [65]–[67],

2. model-based approaches [63],[68].

The feature-based approach focuses on tracking 2D features, such as geometrical primitives (e.g., points, segments, edges, circles) or an object's contours or regions of interest. The main idea of this method is to select a set of feature points, which are matched against the incoming video to update the estimation pose. Feature-based techniques are naturally less sensitive to occlusions, as they are based on local correspondences. Several kinds of research apply this method to head pose estimation by tracking small facial features, like the corners of the eyes or mouth. [69] presented a head tracking algorithm using stereo vision to overcome the occlusion problem. However, the tracker needs to know the initial head pose in order to start tracking, and this is determined by seven corresponding landmark points in each image that are selected manually.

The model-based approach explicitly uses a model of the target objects, which helps the robot estimate the target's pose precisely. The pose includes position and orientation. It then uses a model to search a target object in the image, and this model is composed based on how the target object can be seen in the input image [70]. Compared to feature-based methods, model-based methods have more information about the target object and usually provide a robust solution. For example, they can cope with partial occlusion of the objects. Our method is included in this category. The matched degree of the model to the target can be estimated by a fitness function, whose maximum value represents the best matching and can be solved by a GA (Genetic Algorithm). An advantage of our method is that we use a 3D solid model, which enables it to possess 6DOF (i.e., both the position and orientation). In other methods, such as feature-based recognition, the pose of the target object should be determined by a set of image points, which means it requires a very strict camera calibration. Moreover, searching the corresponding points in stereo vision camera images is also complicated and time-consuming

[71],[72]. However, a model-based matching method adopts a set-point-model form of thinking. All points on the solid model are projected as a group into a 2D image without the mispairing problem. As a result, all projections for each point are correct.

2.4 Camera configuration

From the viewpoint of how the cameras are used, Fig. 2.1 summarizes the camera-robot configurations [73]. Camera-robot configurations used in visual servoing control can be divided into VM1 monocular eye-in-hand, VM2 monocular eye-to-hand, VM3 binocular eye-in-hand, VM4 binocular eye-to-hand [74],[75], and VM5 redundant camera system [73],[76]. The eye-to-hand configuration is also termed stand-alone configuration [73]. In this paper, it is called eye-to-hand [9],[76]. In the eye-in-hand configuration [15], the camera(s) is(are) mounted on the robot's end-effector. In the eye-to-hand configuration, the camera(s) is(are) fixed in the workspace to see the robot's hand [76]. These methods can obtain multiple different views to observe an object by increasing the number of cameras. The eye-in-hand configuration has a partial but precise sight of the scene, since the camera can be placed near targets by a robot hand, whereas the eye-to-hand camera has a less accurate but global view of the robot and the targets. However, in the eye-to-hand configuration, a fixed camera position in the workspace reduces the adaptability of the system for a changing environment since it is fixed. After considering those factors, an eye-in-hand configuration was adopted in our approach. Although multi-view stereo with three or more cameras can give more details, it makes a system too complex and time-consuming [62],[77].

In this paper, the eye-in-hand configuration, i.e., Fig. 2.1 VM3, is used for the research.

2.4.1 Monocular vision and RGB-D camera sensing

A model-based method is an excellent way to solve the above problems using a model of a target object [14]. Even though it enables monocular vision to detect the distance of the target object

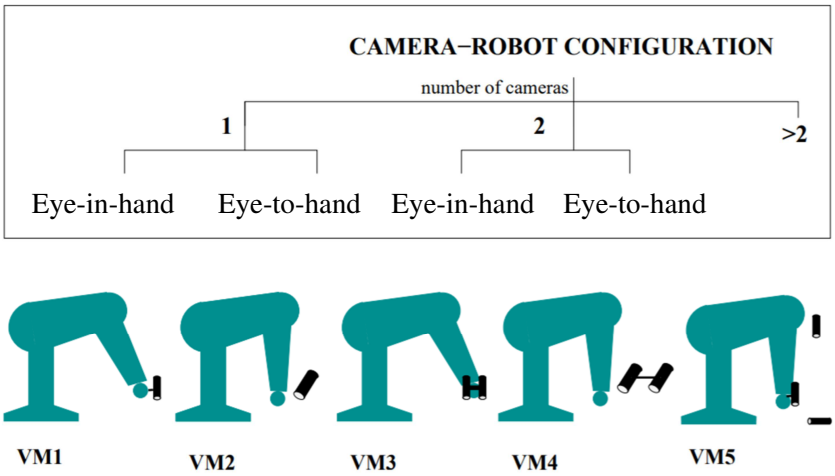


Fig. 2.1: Camera-robot configurations used in visual servoing control (from left to right): VM1 monocular eye-in-hand, VM2 monocular eye-to-hand, VM3 binocular eye-in-hand, VM4 binocular eye-to-hand and VM5 redundant camera system. The eye-to-hand configuration is also named stand-alone configuration. In this paper, it is called eye-to-hand.

from a single image, its accuracy is lower than that of stereo vision [17]–[19]. Moreover, stereo vision is more sensitive to an object's pose variation than monocular vision. Some researchers use an RGB-D camera, one RGB camera, and a depth sensor with infrared light to improve the distance detection of monocular vision, and conduct picking and placing or visual servoing tasks [12],[20]–[22]. RGB-D sensors such as the Microsoft Kinect, Inter RealSense, and the Asus Xtion are inexpensive 3D sensors. A depth image is computed by calculating the distortion of a known infrared light pattern, which is projected into the scene [78]. These studies still rely on the target detection or segmentation from a single image and cannot directly use the depth point cloud for target detection, although the RGB-D camera generates a depth point cloud corresponding to the single image. Therefore, many studies utilize deep learning methods for target detection [20]–[22]. However, this requires a large amount of pictures and pre-training time. Some other studies use model-based methods to simplify preliminary preparations [79]. But both of these methods cannot avoid the disadvantage of the RGB-D camera, i.e., missing depth data caused by the depth sensor. Some pixels do not have corresponding depth data [23]. Additionally, bright ambient illumination can affect the contrast of infrared images in active light sensors, resulting in outliers or holes in the depth map [24].

2.4.2 Stereo vision

Stereo vision extracts 3D information from digital images, such as information obtained by CCD cameras. 3D information is extracted by comparing the relative positions of objects at different positions in the scene information. Unlike optical infrared and electric-field sensing, stereo vision is more robust for varying target material properties and light conditions [25]. It is not dependent on capacitance, reflectivity, or other material properties, as long as the target surface has some visible features. Stereo vision can be divided into two different categories. One is a two-view stereo with two cameras. This is similar to the stereopsis of biological processes. The other is a multi-view stereo with three or more cameras, which is commonly used in 3D projection reconstruction [80]–[82]. Although multi-view stereo can provide more detail, it also makes the system too complicated and time-consuming. For the above reasons, the research proposed in this paper is based on two-view stereo vision, i.e., dual RGB cameras [83].

Chapter 3

Model-Based Recognition

In conventional methods of the stereo vision, the information of a target object is determined by a set of image points in different images. The object relative pose recognition process requires a time-consuming and complex search of the corresponding points. The Corresponding Points Identification Problem [84]–[87] has been pointed out as the difficulty existing 3D image reconstruction from 2D images input from stereo vision (2D-3D method).

In contrast, the author has employed a "Forward Projection," i.e., a 3D model has been projected into stereo vision image planes, and the projected 3D models are compared with the actual target that is also projected naturally onto the stereo vision image planes (3D-2D method). The merit of this method is that it can avoid the Corresponding Points Identification Problem, since the points on a 2D model projected to left and right camera images from points defined on a 3D model have no irregularities in the correspondence of the points in left and right images.

This chapter discusses the methodology of the proposed model-based recognition method with a detailed explanation of stereo vision geometry, 3D model-based matching, and genetic algorithm (GA). And the developed eye-vergence visual servoing system is introduced.

3.1 HSV color model

HSV (Hue, Saturation, Value) color model is an alternative representation of the RGB (Red, Green, Blue) color model. Unlike RGB, HSV separates the color type from the color intensity. It is convenient for color comparison. In this research, it does not need to compare the red, green, and blue three parameters of color. Only hue is used for object detection. And the HSV color model is similar to how humans perceive colors. The hue of a color changes little when the intensity of ambient light changes.

As shown in Fig. 3.1, H (Hue) [0$\sim$360) of the HSV color model represents different color types such as red or blue. S (Saturation) [0$\sim$1] represents a vividness of a color. And it is represented by the distance from the center of the hue circle. V(Value) [0$\sim$255] represents the intensity or brightness of a color. It is the axis orthogonal to the circle of the hue circle. The smaller the Value is, the darker the color is.

The conversion formula from RGB color model to HSV color model is described as below. r, g, and b represent the red, green, and blue components of the RGB color model. The V(Value) is calculated as

$$V = max\{r, g, b\}. \tag{3.1}$$

v is defined as

$$v = min\{r, g, b\}. \tag{3.2}$$

S (Saturation) is calculated as

$$S = \begin{cases} 0 & \text{if}(V = 0); \\ (V - v)/V & \text{if}(V \neq 0). \end{cases} \tag{3.3}$$

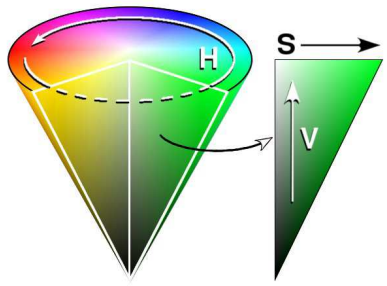


Fig. 3.1: HSV color model.

In the end, H (Hue) is calculated as

$$H = \begin{cases} \text{undefined}, & \text{if}(S = 0); \\ 60(g-b)/(V-v), & \text{if}(g \geqq b, V = r, \text{ and } S \neq 0); \\ 360 + 60(g-b)/(V-v), & \text{if}(g < b, V = r, \text{ and } S \neq 0); \\ 120 + 60(b-r)/(V-v), & \text{if}(V = g \text{ and } S \neq 0); \\ 240 + 60(r-g)/(V-v), & \text{if}(V = b \text{ and } S \neq 0). \end{cases} \tag{3.4}$$

3.2 Projective transformation matrix

From the relationship of the central projection as shown in Fig. 3.2, the focal length of the camera is f, the image center coordinate is $({}^{I}x_0, {}^{I}y_0)$. And the ratios of unit length on the x axis and the y axis in the camera coordinate system Σ_C with unit [mm] and that in the image coordinate system Σ_I with unit [pixel] is η_x, η_y[mm/pixel]. The distance between the origins of the two coordinate systems is a. An arbitrary point $({}^{C}x_i, {}^{C}y_i, {}^{C}z_i)$ in Σ_C is transmitted through the lens to Σ_I as image $({}^{I}x_i, {}^{I}y_i)$. From Fig. 3.2, when the thickness of the lens is not considered, the coordinate relation between arbitrary points and its image is expressed by the following relation.

$$\frac{({}^{I}x_i - {}^{I}x_0)\eta_x}{{}^{C}x_i} = \frac{a}{{}^{C}z_i} \tag{3.5}$$

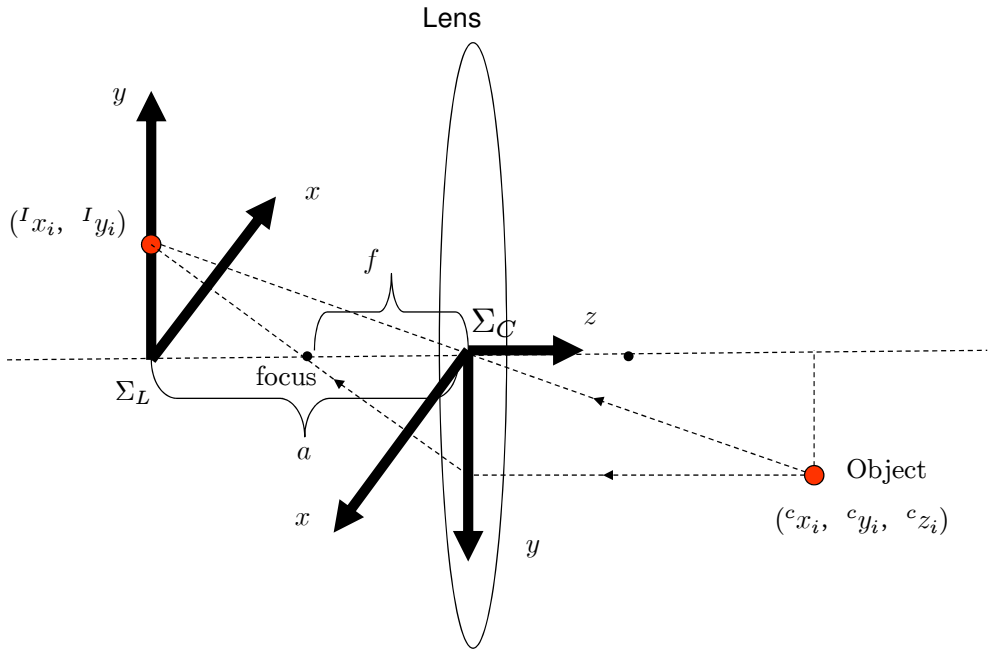


Fig. 3.2: Projection schematic diagram

$$\frac{({}^I y_i - {}^I y_0)\eta_y}{{}^C y_i} = \frac{a}{{}^C z_i} \tag{3.6}$$

The following equation is derived by summing up Eqs. (3.5) and (3.6).

$$\begin{bmatrix} {}^I x_i \\ {}^I y_i \end{bmatrix} = \frac{1}{{}^C z_i} \begin{bmatrix} a/\eta_x & 0 & {}^I x_0 & 0 \\ 0 & a/\eta_y & {}^I y_0 & 0 \end{bmatrix} \begin{bmatrix} {}^C x_i \\ {}^C y_i \\ {}^C z_i \\ 1 \end{bmatrix} \tag{3.7}$$

Because the distance between the object and the lens is a relatively large value as compared with the focal length, a can be approximated as the focal length f and expressed by the following

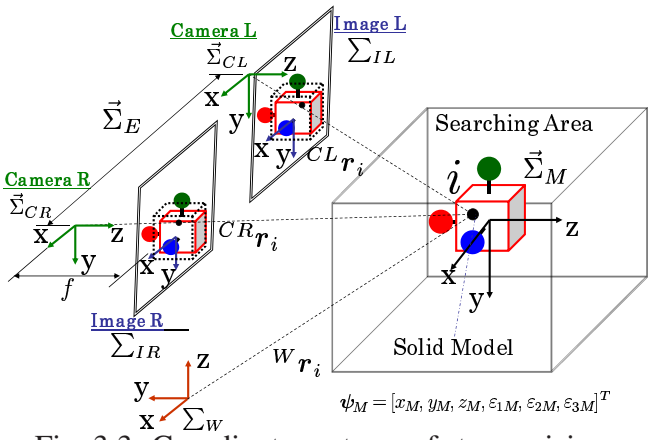


Fig. 3.3: Coordinate systems of stereo vision.

equation.

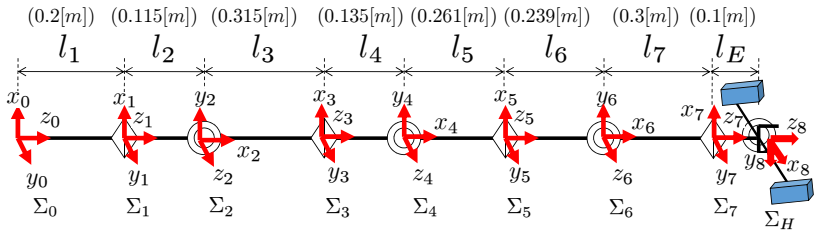
$$\begin{bmatrix} {}^{I}x_i \\ {}^{I}y_i \end{bmatrix} = \frac{1}{{}^{C}z_i} \begin{bmatrix} f/\eta_x & 0 & {}^{I}x_0 & 0 \\ 0 & f/\eta_y & {}^{I}y_0 & 0 \end{bmatrix} \begin{bmatrix} {}^{C}x_i \\ {}^{C}y_i \\ {}^{C}z_i \\ 1 \end{bmatrix} \tag{3.8}$$

From this, the projection transformation matrix is denoted to the camera as $\boldsymbol{P}$ and summarized as follows.

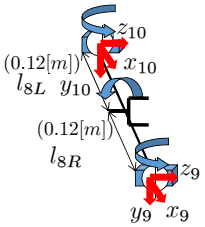
$$\boldsymbol{P} = \frac{1}{{}^{C}z_i} \begin{bmatrix} f/\eta_x & 0 & {}^{I}x_0 & 0 \\ 0 & f/\eta_y & {}^{I}y_0 & 0 \end{bmatrix} \tag{3.9}$$

3.3 Stereo vision geometry

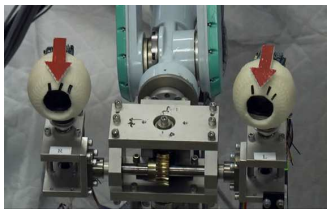
Figure 3.3 shows the coordinate system of the stereo vision system. The target object's coordinate system is represented by $\vec{\Sigma}_M$ and image coordinate systems of the left and right cameras are represented by Σ_{IL} and Σ_{IR}. The structure of the manipulator and the cameras are shown in Fig. 3.4. The coordinates of the target object and the manipulator in the experiment are shown in Fig. 3.17. The difference between $\vec{\Sigma}$ and Σ is explained in Section 3.6.2. A point i on the target can be described using these coordinates and homogeneous transformation matrices. At first, a homogeneous transformation matrix from right camera coordinate system $\vec{\Sigma}_{CR}$ to $\vec{\Sigma}_M$ is



(a) Sketch map of the eye-vergence system



(b) Sketch map of the cameras



(c) Physical map of the cameras

Fig. 3.4: Frame structure of manipulator

defined as ${}^{CR}\boldsymbol{T}_M$. And an arbitrary point i on the target object in $\vec{\Sigma}_{CR}$ and $\vec{\Sigma}_M$ is defined as ${}^{CR}\boldsymbol{r}_i$ and ${}^{M}\boldsymbol{r}_i$. Then ${}^{CR}\boldsymbol{r}_i$ is,

$$ {}^{CR}\boldsymbol{r}_i = {}^{CR}\boldsymbol{T}_M \, {}^{M}\boldsymbol{r}_i. \tag{3.10} $$

Where ${}^{M}\boldsymbol{r}_i$ is predetermined fixed vectors. Using a homogeneous transformation matrix from world coordinate system Σ_W to the right camera coordinate system $\vec{\Sigma}_{CR}$, i.e., ${}^{W}\boldsymbol{T}_{CR}$, then ${}^{W}\boldsymbol{r}_i$ is got as,

$$ {}^{W}\boldsymbol{r}_i = {}^{W}\boldsymbol{T}_{CR} \, {}^{CR}\boldsymbol{r}_i. \tag{3.11} $$

The position vector of i point in right image coordinates, ${}^{IR}\boldsymbol{r}_i$ is described by using projection matrix $\boldsymbol{P}$ of camera as,

$$ {}^{IR}\boldsymbol{r}_i = \boldsymbol{P} \, {}^{CR}\boldsymbol{r}_i. \tag{3.12} $$

By the same way as above, using a homogeneous transformation matrix of fixed values defining the kinematical relation from the left camera coordinate system $\vec{\Sigma}_{CL}$ to the right camera coordinate system $\vec{\Sigma}_{CR}$, ${}^{CL}\boldsymbol{T}_{CR}$, ${}^{CL}\boldsymbol{r}_i$ is

$$ {}^{CL}\boldsymbol{r}_i = {}^{CL}\boldsymbol{T}_{CR} \, {}^{CR}\boldsymbol{r}_i. \tag{3.13} $$

${}^{IR}\boldsymbol{r}_i$, ${}^{IL}\boldsymbol{r}_i$ is described by the following Eq. (3.14) through projection matrix $\boldsymbol{P}$.

$$ {}^{IL}\boldsymbol{r}_i = \boldsymbol{P}\ {}^{CL}\boldsymbol{r}_i = \boldsymbol{P}^{CL}\boldsymbol{T}_{CR} \cdot {}^{CR}\boldsymbol{r}_i \tag{3.14} $$

Then position vectors projected in the Σ_{IR} and Σ_{IL} of arbitrary point i on target object can be described ${}^{IR}\boldsymbol{r}_i$ and ${}^{IL}\boldsymbol{r}_i$. Here, position and orientation of $\vec{\Sigma}_M$ based on $\vec{\Sigma}_{CR}$ has been defined as $\boldsymbol{\psi}_M$, which means ${}^{CR}\boldsymbol{T}_M$ in Eq. (3.10) is determined by $\boldsymbol{\psi}_M$. Then Eqs. 3.12 and 3.14 are rewritten as,

$$ \begin{cases} {}^{IR}\boldsymbol{r}_i = \boldsymbol{f}_R(\boldsymbol{\psi}_M,\ {}^{M}\boldsymbol{r}_i) \\ {}^{IL}\boldsymbol{r}_i = \boldsymbol{f}_L(\boldsymbol{\psi}_M,\ {}^{M}\boldsymbol{r}_i). \end{cases} \tag{3.15} $$

3.4 Orientation representation by quaternion

There are several representations used to describe the orientation of a rigid body. Euler angle is a well-known one that includes a set of three angles rotating around three coordinates, e.g., z, y, z successively. A drawback of the Euler angle is the occurrence of representation singularities (for a manipulator, the Jacobian matrix is singular for some orientation).

An alternative representation is angle/axis, describing the general orientation of a rigid body as a displacement of an angle around an axis. A general angle/axis representation is not unique because a rotation by an angle $-\theta$ around an axis $-\boldsymbol{k}$ can not be distinguished from a rotation by θ around $\boldsymbol{k}$.

A unit quaternion is different from angle/axis representation. It can represent the orientation of a rigid body without singularities [88]. For the reader's convenience, a few basic concepts regarding the use of a unit quaternion to describe the orientation of a rigid body are summarized hereafter [89]. As shown in Fig. 3.5, the unit quaternion is defined as

$$ \boldsymbol{Q} = [\eta, \boldsymbol{\varepsilon}], (\eta = cos\frac{\theta}{2}, \boldsymbol{\varepsilon} = sin\frac{\theta}{2}\boldsymbol{k}), \tag{3.16} $$

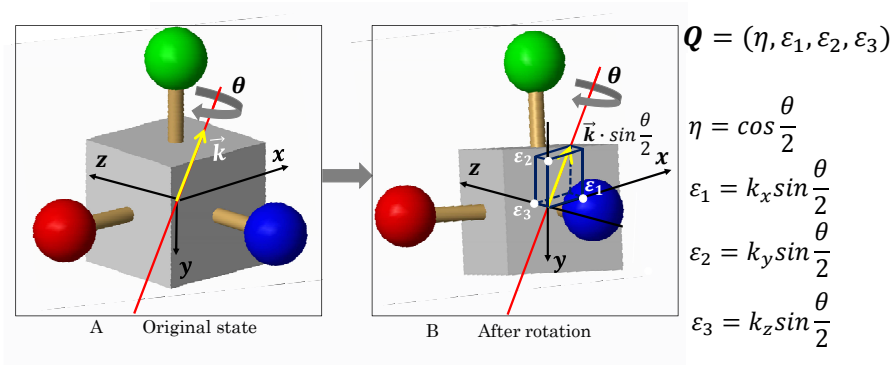


Fig. 3.5: Definition of quaternion in the proposed system.

where $\boldsymbol{k}(\|\boldsymbol{k}\| = 1)$ is the rotation axis and θ is the rotation angle around $\boldsymbol{k}$. η is called the scalar part of the quaternion and $\boldsymbol{\varepsilon}$ is called the vector part of the quaternion. They are constrained by

$$\eta^2 + \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} = 1. \tag{3.17}$$

It is worth to remark that, unlike the angle/axis representation, a rotation by an angle $-\theta$ around an axis $-\boldsymbol{k}$ have the same quaternion as a rotation by θ around $\boldsymbol{k}$. Therefore, quaternion can solve the nonuniqueness problem of angle/axis representation. If the position and orientation of an object $\vec{\Sigma}_M$ based on the end-effector $\vec{\Sigma}_E$ is ${}^{\boldsymbol{E}}\boldsymbol{\psi}_M = [{}^E x_M, {}^E y_M, {}^E z_M, \varepsilon_1, \varepsilon_2, \varepsilon_3]^T$, the homogeneous transformation matrix can be calculated as Eq. (3.18) [90]–[93].

$${}^E\boldsymbol{T}_M = \begin{bmatrix} 1-2\varepsilon_2^2-2\varepsilon_3^2 & 2(\varepsilon_1\varepsilon_2-\eta\varepsilon_3) & 2(\varepsilon_1\varepsilon_3+\eta\varepsilon_2) & {}^E x_M \\ 2(\varepsilon_1\varepsilon_2+\eta\varepsilon_3) & 1-2\varepsilon_1^2-2\varepsilon_3^2 & 2(\varepsilon_2\varepsilon_3-\eta\varepsilon_1) & {}^E y_M \\ 2(\varepsilon_1\varepsilon_3-\eta\varepsilon_2) & 2(\varepsilon_2\varepsilon_3+\eta\varepsilon_1) & 1-2\varepsilon_1^2-2\varepsilon_2^2 & {}^E z_M \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.18}$$

3.5 Model-based recognition with Real-Time Multi-Step Genetic Algorithm (RM-GA)

As shown in Fig. 3.6, a 3D-ball-object named as 3D marker is used as 3D target object whose size and color are known. The sizes of the balls projected into 2D images of left and right cameras from the 3D model are different because the camera depth distance of each ball is different.

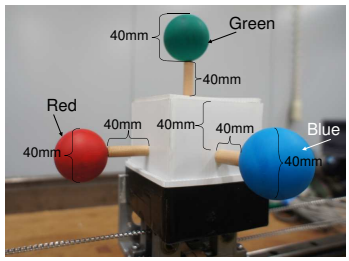


Fig. 3.6: 3D marker

As shown in Fig. 3.7 (a), the dotted line block named $\mathbb{R}$ means a searching space that will be described in detail in Section 4.2.1. Σ_j is a model searching for the pose of the 3D marker. The models have the same 3D structure as the 3D marker. The part of inner circle is named as S_{in}, and the part between S_{in} and outer circle is named as S_{out}. Through the projection transformation, S_{in} and S_{out} are projected onto the 2D coordinates of the left image and right image shown in (b). Then we take the sampling points on the images like (c) and calculate the fitness $F({}^{E}\boldsymbol{\psi}^{j}_{\widehat{M}})$ to evaluate the overlap degree between the model and the object in images. Through the fitness function, 3D marker searching problem can be changed to an optimization problem. For real-time pose estimation, the optimization problem has been solved by Real-Time Multi-Step Genetic Algorithm (RM-GA). The pose of j-th 3D model based on the end-effector $\vec{\Sigma}_E$ is defined by chromosome

$$\overbrace{10\cdots10}^{x}\overbrace{11\cdots01}^{y}\overbrace{01\cdots10}^{z}\overbrace{11\cdots10}^{\varepsilon_1}\overbrace{10\cdots10}^{\varepsilon_2}\overbrace{10\cdots01}^{\varepsilon_3}.$$

12[bit] 12[bit] 12[bit] 12[bit] 12[bit] 12[bit]

Firstly, fitness function is explained. Each model has three small circles S_{in} and two rings

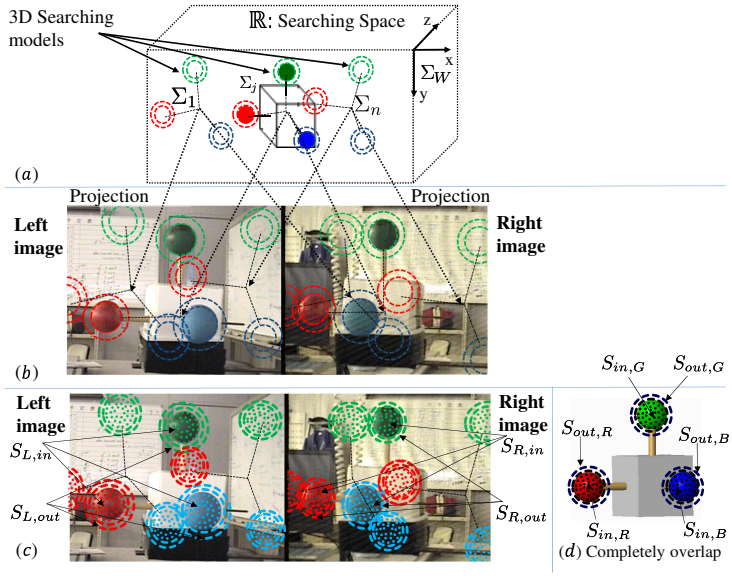


Fig. 3.7: (a) shows the searching space. And searching models are projected into 2D images (b). Sampling points are shown in two images (c). When a model completely overlap the object (d), its fitness function gets the maximum.

S_{out}. The relative positions of circles and rings in 3D space are unchanged. Each pair of circle and ring corresponds with a color, and three pairs of circles and rings are corresponding to red, green, and blue.

As shown in Fig. 3.7 (d), inner portions of a model corresponding to three balls are $S_{in,R}$, $S_{in,G}$ and $S_{in,B}$. Similarly, the three outer portions are $S_{out,R}$, $S_{out,G}$ and $S_{out,B}$. Each pair of circle and ring corresponds with a color, and three pairs of circles and rings are corresponding to red, green, and blue. $S_{in,R}$ of a model has 36 sampling points. $S_{out,R}$ of a model has 24 sampling points. The green and blue portions of a model have the same number of points as that of red portion. Therefore, the sum of sampling points of a model is

$$\Sigma_s = \Sigma_{sin} + \Sigma_{sout} = 3 \times 36 + 3 \times 24 = 180. \tag{3.19}$$

To determine which solid model is closest to the real target, a correlation function, i.e., fitness function is defined for evaluation. As shown in Eq. (3.21), the hue of each sampling point $\boldsymbol{r}_i$ is compared with a fixed hue H_u corresponding to the color of ball. For red, green, and blue balls,

their fixed hue value H_u is set as $H_R = 0, H_G = 120$, and $H_B = 240$ respectively. The hue value of a pixel of a captured image overlapped by $\boldsymbol{r}_i$ is h_u. If h_u is near to H_u, the calculate value of $\boldsymbol{r}_i$ is $p(\boldsymbol{r}_i) = 1$. Otherwise, $p(\boldsymbol{r}_i) = -1$. For the concision of Eq. (3.21), about red ball $u = R$, the judgement condition $h_R \in [H_R - 20, H_R + 20]$ of $p(\boldsymbol{r}_i) = 1$ is an abberation of $h_R \in ([0, H_R + 20] \cup [H_R - 20 + 360, 360])$, i.e., $h_R \in ([0, 20] \cup [340, 360])$. The judgement condition of $p(\boldsymbol{r}_i) = -1$ is the complement of that of $p(\boldsymbol{r}_i) = 1$. As shown of Eq. (3.22), the sum of $p(\boldsymbol{r}_i)$ of all the sampling points in a model Σ_j is defined as fitness $F({}^E\boldsymbol{\psi}^j_{\widehat{M}})$. The higher coincidence degree between the inner portion of a model and the corresponding ball of the image is, the higher fitness is. Conversely, the higher coincidence degree between the outer portion and the corresponding ball of the image is, lower fitness will be. The fitness function is defined by Eq. (3.22). When the searching model Σ_j completely overlaps to the target object like (d), then the fitness function gives maximum value

$$F_{max}({}^E\boldsymbol{\psi}^j_{\widehat{M}}) = \Sigma_s/\Sigma_{sin} = 180/108 = 1.67. \tag{3.20}$$

$$p(\boldsymbol{r}_i) = \begin{cases} 1 \ (h_u \in [H_u - 20, H_u + 20], u = R, G, B), \\ -1 \ (h_u \notin [H_u - 20, H_u + 20], u = R, G, B). \end{cases} \tag{3.21}$$

$$\begin{aligned} F({}^E\boldsymbol{\psi}^j_{\widehat{M}}) = & \left\{ \Big(\sum_{\substack{{}^{IR}\boldsymbol{r}_i \in \\ S_{R,in}({}^{CR}\boldsymbol{\psi}^j_{\widehat{M}})}} p({}^{IR}\boldsymbol{r}_i) - \sum_{\substack{{}^{IR}\boldsymbol{r}_i \in \\ S_{R,out}({}^{CR}\boldsymbol{\psi}^j_{\widehat{M}})}} p({}^{IR}\boldsymbol{r}_i) \Big) / n_{R,in} \right. \\ & \left. + \Big(\sum_{\substack{{}^{IL}\boldsymbol{r}_i \in \\ S_{L,in}({}^{CL}\boldsymbol{\psi}^j_{\widehat{M}})}} p({}^{IL}\boldsymbol{r}_i) - \sum_{\substack{{}^{IL}\boldsymbol{r}_i \in \\ S_{L,out}({}^{CL}\boldsymbol{\psi}^j_{\widehat{M}})}} p({}^{IL}\boldsymbol{r}_i) \Big) / n_{L,in} \right\} /2 \\ = & \left\{ F({}^{CR}\boldsymbol{\psi}^j_{\widehat{M}}) + F({}^{CL}\boldsymbol{\psi}^j_{\widehat{M}}) \right\} /2 \end{aligned} \tag{3.22}$$

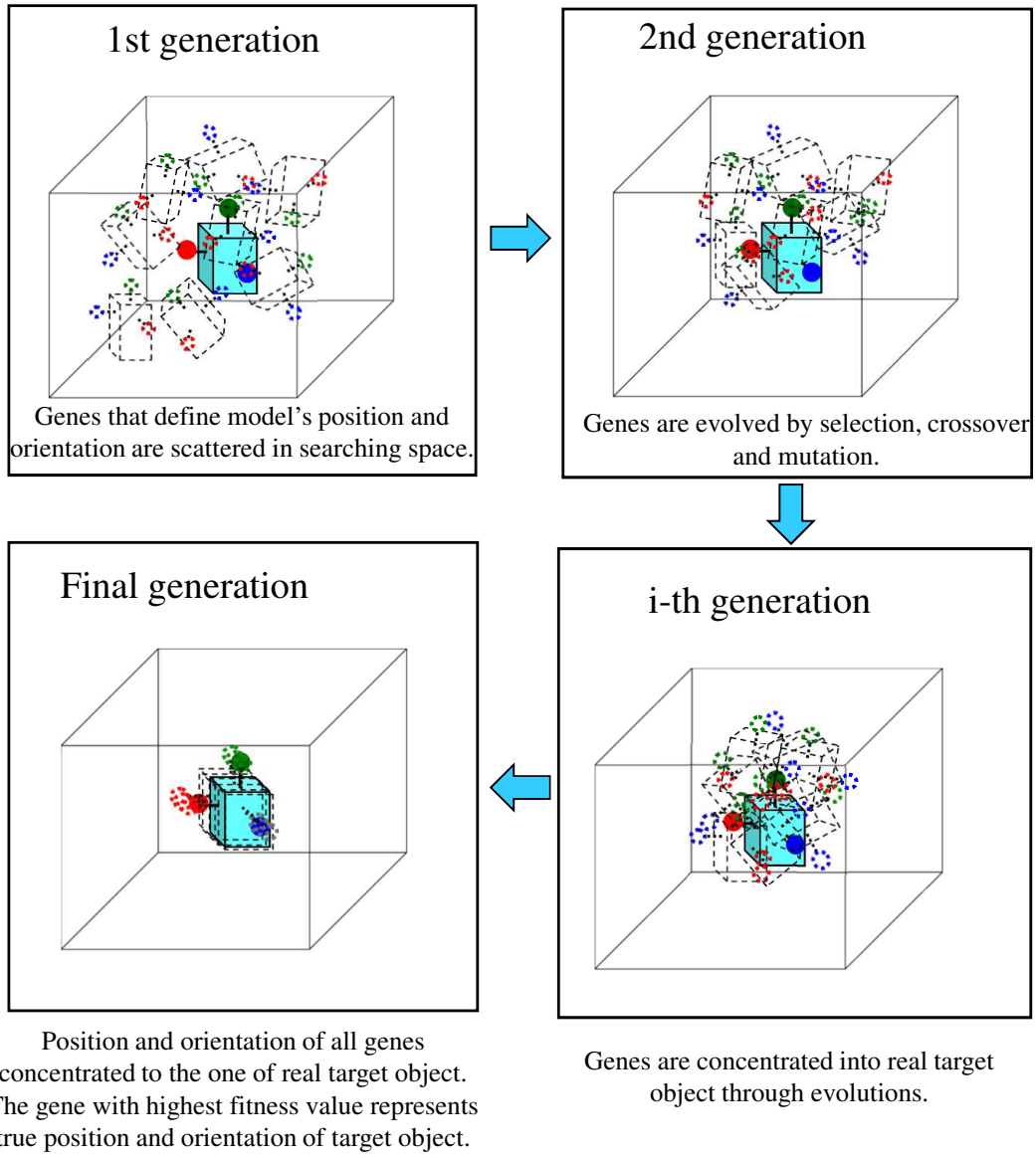


Fig. 3.8: Evolution process, 3D models converge into the real target object.

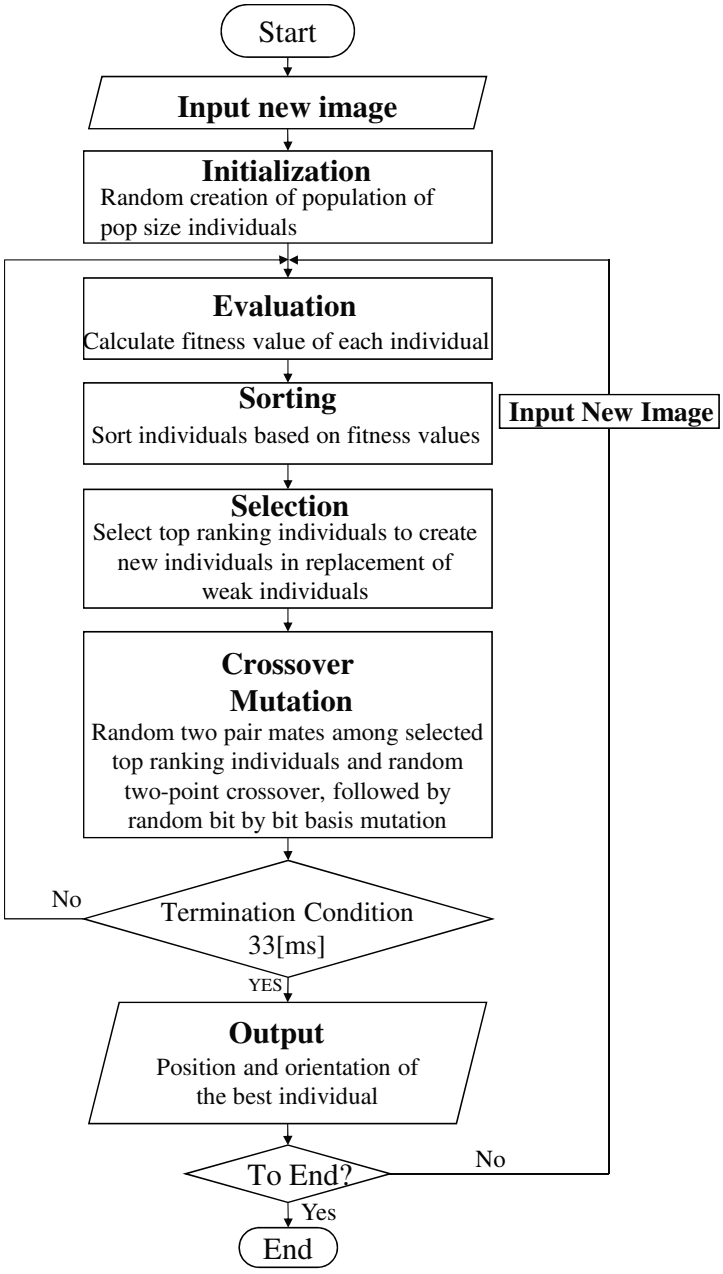


Fig. 3.9: Flow chart of RM-GA.

The evolution process of RM-GA is shown in Fig. 3.8. At first, models of the target object whose poses are represented by genes are scattered in the search space. Then the fitness of each model is calculated. Through selection, crossover, and mutation, a new generation with the same number of models as the last generation is generated. The models converge to the real target though evolution progressing. In the final generation, the gene ${}^{E}\boldsymbol{\psi}_{\widehat{M}}$ that gives the highest fitness value can be thought true pose of the real target.

The following is summarized explanation about the real-time pose (position and orientation) tracking method as shown in Fig. 3.9. About this method,

1. genes are randomly generated to define the poses of the models.

2. These 3D models are projected onto the left and right camera images.

3. The correlation degrees between the 3D object captured by the left and right cameras and the projected models are calculated through the fitness function.

4. The correlation degrees are utilized to evolve genes representing position/orientation.

5. Because the time for transferring one frame of video to the memory is 9.2[ms], the remaining time within the video rate 33[ms] is $33 - 9.2 = 23.8$[ms]. During this time the genes can be evolved nine times by GA.

6. The position/orientation of the model giving the highest fitness among the genes evolved at the time of 23.8[ms] is taken as the position/orientation measurement result of the 3D object at that time.

By repeating the above steps $2 \sim 6$, it is measured that the position/orientation of the object in the moving image continuously input at the video rate.

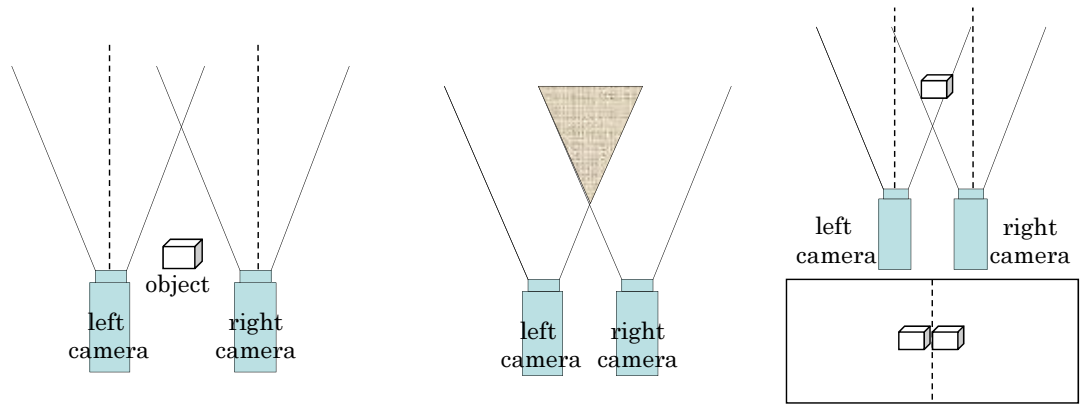
3.6 Eye-vergence visual servoing system

3.6.1 Advantage of eye-vergence visual servoing system

Comparing with a fixed eye system that cameras are fixed to the hand, there are advantages in the eye-vergence system, where cameras can rotate. As shown in Fig. 3.10 (a), in terms of the fixed eye system when an object is near to the cameras, it may not be recognized. In addition, in (b), the common visible region of the two cameras is narrow. In (c), even if the object is in a visible region, it is not captured to the center of the camera image field. There is an aberration problem because an object is easily affected by lens distortion, which causes the object in the periphery of the lens to become larger. To avoid lens distortion problems, in this paper, both cameras have the flexibility to change the angle to capture the object in the center of the image. Because it is possible to change the orientation of the camera, as shown in Fig. 3.11 (a)-(c), it improves the performance of observing the object. Corresponding to the problems in Fig. 3.10 (a)-(c), as shown in Fig. 3.11 (a)-(b), the proposed system expands the binocular view zone. And as shown in (c), it becomes possible that observing an object at the center of the lens. It can be avoided that the distortion of the input image generated by the lens aberration.

In the eye-vergence system shown in Fig. 3.11, because it is possible to control line-of-sight direction of cameras for catching object in the center of images, the cameras can continue gazing the object in the visual field center [94]–[96].

In the application of visual servoing it is necessary to keep stability so that the object should stay in vision field of cameras. Figure 3.12 (a) shows that cameras can catch an object. (b) shows that the cameras are fixed to the hand. When an object moves fast, it disappears from the view field of the camera, and the control system may fall into a dangerous state to move aimlessly. Thus, in the visual servoing system, it is important to raise the trackability that cameras can continue catching the object in the camera field of vision. In addition, it is thought that eye-vergence system has better tracking property than the fixed eye system because the mass and the moment of inertia of the camera are relatively smaller than the whole manipulator.

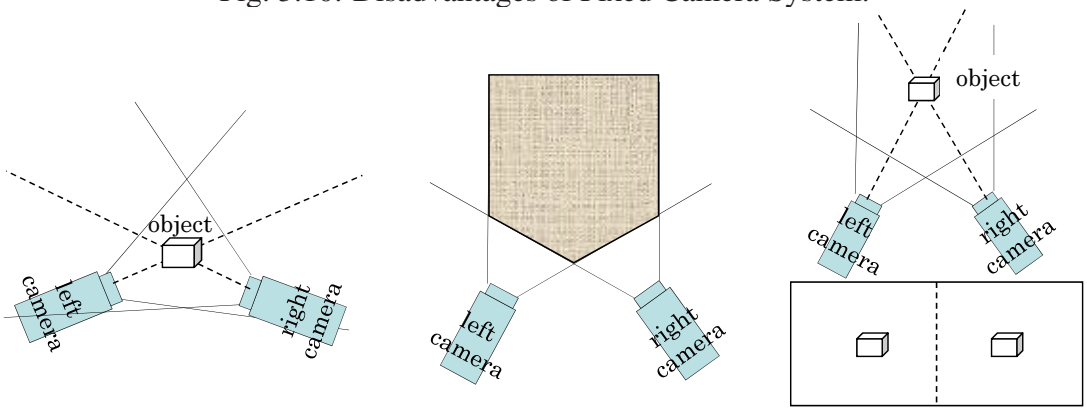


(a) Cannot be seen when the object near to the cameras

(b) Smaller possible sight area

(c) Not in the center of the sight

Fig. 3.10: Disadvantages of Fixed Camera System.



(a) Can be seen when the object near to the cameras

(b) Bigger possible sight area

(c) In the center of the sight

Fig. 3.11: Advantages of eye-vergence system.

As shown in Fig. 3.12 (c), the trackability can be raised by adding eyes' controller.

3.6.2 Symbol meaning

M represents the object and $\widehat{M}$ represents the estimated object. Then $\vec{\Sigma}_M$ denotes the coordinate system that moves along with the object. The relationships between coordinate systems such as the actual pose of the hand $\vec{\Sigma}_E$ or the recognized pose of the object $\vec{\Sigma}_{\widehat{M}}$ are shown in Fig. 3.18. $\vec{\Sigma}$ represents a coordinate system moving in the world coordinate system Σ_W. The coordinate system represented by Σ keeps fixed in Σ_W. $\vec{\Sigma}_E$, $\vec{\Sigma}_{Ed}$, $\vec{\Sigma}_M$ and $\vec{\Sigma}_{\widehat{M}}$ are all moving in Σ_W. Σ_{EO}, Σ_{EC} and Σ_{MO} keep fixed in the world coordinate system Σ_W.

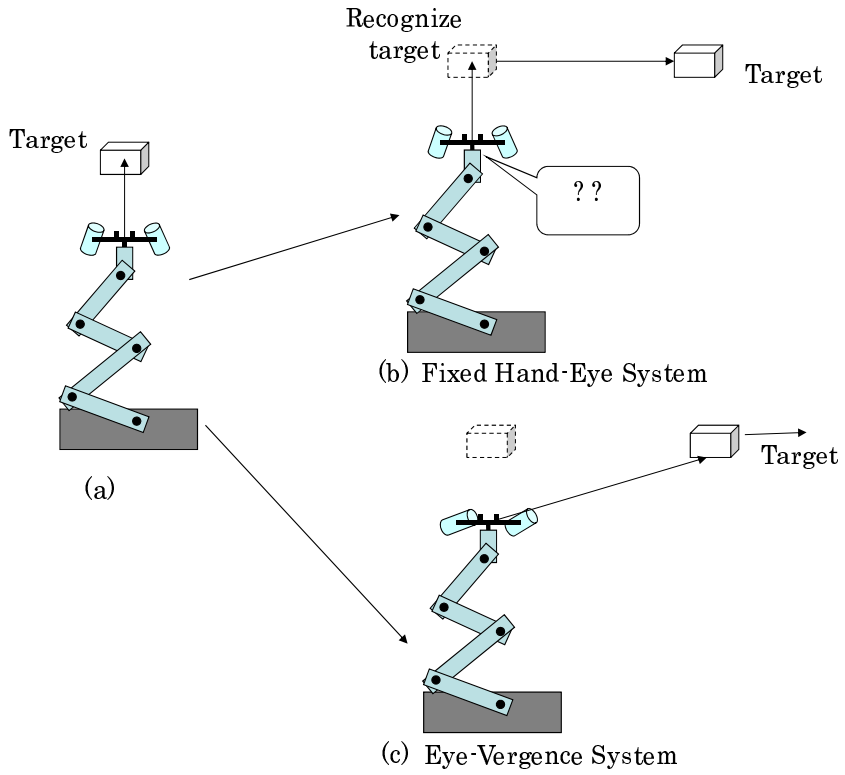


Fig. 3.12: Dynamical advantage of eye-vergence system.

3.6.3 Generation of desired-trajectory

Fig. 3.13 shows the relationship between the hand and the object. Σ_W is the world coordinate system, and $\vec{\Sigma}_M$ is the coordinate system fixed on the object. Furthermore, the coordinate system of the actual hand and its target coordinate system are represented by $\vec{\Sigma}_E$, $\vec{\Sigma}_{Ed}$. The relative position/orientation relationship between the target state of the hand and the object is represented by the homogeneous transformation matrix ${}^{Ed}\boldsymbol{T}_M$. And the relationship between the actual hand and the object is represented by ${}^{E}\boldsymbol{T}_M$. At this time, the difference between $\vec{\Sigma}_E$ and $\vec{\Sigma}_{Ed}$ is expressed as ${}^{E}\boldsymbol{T}_{Ed}$. And ${}^{E}\boldsymbol{T}_{Ed}$ can be described as follows.

$$ {}^{E}\boldsymbol{T}_{Ed}(t) \;=\; {}^{E}\boldsymbol{T}_M(t)\,{}^{Ed}\boldsymbol{T}_M^{-1}(t) \tag{3.23} $$

(3.23) includes an arbitrary motion ${}^{E}\boldsymbol{T}_M(t)$ of the object represented by $\vec{\Sigma}_E$ and the relative time-varying visual servo target trajectory ${}^{Ed}\boldsymbol{T}_M(t)$ represented by the arbitrary target posi-

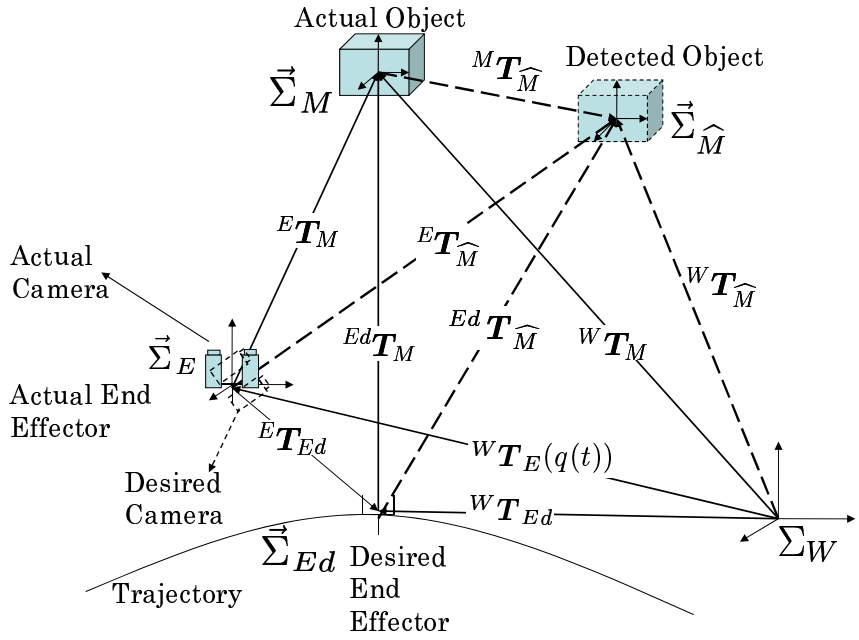


Fig. 3.13: Motion of the end-effector and object.

tion/orientation of the robot hand $\vec{\Sigma}_{Ed}$. ${}^{E}\boldsymbol{T}_{M}(t)$ is measured by online model-based recognition method [15],[52] that uses the velocity/angular velocity information of the hand as feedforward information and a moving image recognition method GA [97] to recognize moving image sequence input at video rate. When the estimated object is represented by $\vec{\Sigma}_{\widehat{M}}$, it is general that an error ${}^{M}\boldsymbol{T}_{\widehat{M}}$ exists between the actual object $\vec{\Sigma}_{M}$ and the detected object $\vec{\Sigma}_{\widehat{M}}$. Here, we reconstruct the position/orientation error ${}^{E}\boldsymbol{T}_{Ed}(t)$ of the hand represented by Eq. (3.23) based on the object $\vec{\Sigma}_{\widehat{M}}$ estimated as follows.

$$ {}^{E}\boldsymbol{T}_{Ed}(t) \;=\; {}^{E}\boldsymbol{T}_{\widehat{M}}(t)\,{}^{\widehat{M}}\boldsymbol{T}_{Ed}(t) \tag{3.24} $$

When Eq. (3.24) is differentiated with respect to time, the following equation is obtained.

$$ {}^{E}\dot{\boldsymbol{T}}_{Ed}(t) = {}^{E}\dot{\boldsymbol{T}}_{\widehat{M}}(t)\,{}^{\widehat{M}}\boldsymbol{T}_{Ed}(t) + {}^{E}\boldsymbol{T}_{\widehat{M}}(t)\,{}^{\widehat{M}}\dot{\boldsymbol{T}}_{Ed}(t). \tag{3.25} $$

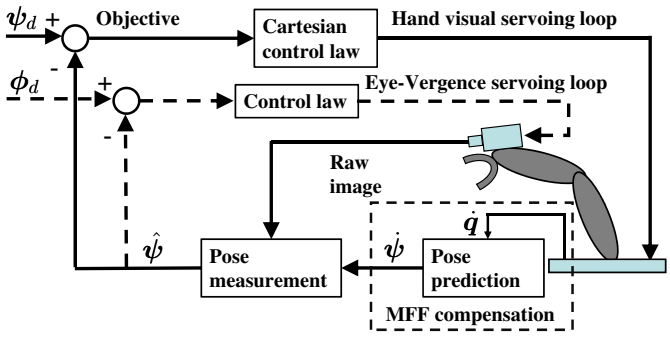


Fig. 3.14: Hand & Eye-vergence Visual Servoing System. MFF is motion-feedforward.

Here, ${}^{\widehat{M}}\boldsymbol{T}_{Ed}$, ${}^{\widehat{M}}\dot{\boldsymbol{T}}_{Ed}$ is given in advance as the target trajectory of the visual servo, and ${}^{E}\boldsymbol{T}_{\widehat{M}}$, ${}^{E}\dot{\boldsymbol{T}}_{\widehat{M}}$ is observed by RM-GA. ${}^{E}\boldsymbol{T}_{Ed}(t)$ and ${}^{E}\dot{\boldsymbol{T}}_{Ed}(t)$ are the position/orientation error between $\vec{\Sigma}_E$ and $\vec{\Sigma}_{Ed}$ and its time differentiation, which is necessary when constructing the controller. As shown in Fig. 3.13, there are two errors that should be 0 in the visual servo process. One is the recognition error between the actual object and the detected object ${}^{M}\boldsymbol{T}_{\widehat{M}}$, and the other is the error of the motion control given by the target state of the hand and the actual hand ${}^{E}\boldsymbol{T}_{Ed}$.

3.6.4 Hand visual servoing controller

The proposed visual servo controller consists of two portions, hand position/orientation controller and sight line controller. Its block diagram is shown in Fig. 3.14. The hand visual servoing is the outer loop. In the figure, the notation "MFF" in the block marked "Pose Prediction" is the abbreviation of "Motion Feedforward." When the moving speed of the object in the image is calculated, it is necessary to think about the influence of the position/orientation changes of the hand because of its velocity/angular velocity. "MFF" is a method that advances the evolution of the GA using the moving speed of the object to predict the object position after 33[ms] of the video rate [15].

$$ {}^{W}\dot{\boldsymbol{r}}_d = \boldsymbol{K}_{PP}{}^{W}\boldsymbol{r}_{E,Ed} + \boldsymbol{K}_{VP}{}^{W}\dot{\boldsymbol{r}}_{E,Ed}, \tag{3.26} $$

Using the motion trajectory of the hand discussed in the previous section, the target speed ${}^{W}\dot{\boldsymbol{r}}_d$ of the hand is calculated by PD control law as Eq. (3.26). ${}^{W}\boldsymbol{r}_{E,Ed}$ is the position vector from the origin of $\vec{\Sigma}_E$ to the origin of $\vec{\Sigma}_{Ed}$ expressed in Σ_W in Fig. 3.13. And ${}^{W}\dot{\boldsymbol{r}}_{E,Ed}$ is its time differentiation. ${}^{W}\boldsymbol{r}_{E,Ed}$, ${}^{W}\dot{\boldsymbol{r}}_{E,Ed}$ are obtained from ${}^{E}\boldsymbol{T}_{Ed}$ and ${}^{E}\dot{\boldsymbol{T}}_{Ed}$ using the coordinate transformation from $\vec{\Sigma}_E$ to Σ_W. $\boldsymbol{K}_{PP} = \mathrm{diag}(0.4, 0.4, 0.4)$[1/s] is a spring constant. $\boldsymbol{K}_{VP} = \mathrm{diag}(0.1, 0.1, 0.1)$[dimensionless] is a positive definite diagonal matrix representing viscous damping coefficient. The target position/orientation of the hand is ${}^{W}\boldsymbol{\psi}_d^{\mathrm{T}} = [{}^{W}\boldsymbol{r}_d^{\mathrm{T}}, {}^{W}\boldsymbol{\varepsilon}_d^{\mathrm{T}}]^{\mathrm{T}}$. ${}^{W}\boldsymbol{\varepsilon}_d$ is the target orientation expressed by quaternion, and the target angular velocity vector ${}^{W}\boldsymbol{\omega}_d$ of the hand is calculated by the following equation.

$$ {}^{W}\boldsymbol{\omega}_d = \boldsymbol{K}_{PO}{}^{W}\boldsymbol{R}_E{}^{E}\Delta\boldsymbol{\varepsilon} + \boldsymbol{K}_{VO}{}^{W}\boldsymbol{R}_E{}^{E}\boldsymbol{\omega}_{E,Ed}, \tag{3.27} $$

Here, ${}^{E}\Delta\boldsymbol{\varepsilon}$ is the deviation of the quaternion, which is the orientation error of the object represented by $\vec{\Sigma}_E$, and is obtained directly from the recognition result ${}^{E}\boldsymbol{T}_{Ed}$ by RM-GA. Because $\boldsymbol{S}(\boldsymbol{\omega}){}^{E}\boldsymbol{R}_{Ed} = {}^{E}\dot{\boldsymbol{R}}_{Ed}$ is established [98], therefore,

$$ \boldsymbol{S}(\boldsymbol{\omega}) = {}^{E}\dot{\boldsymbol{R}}_{Ed}{}^{E}\boldsymbol{R}_{Ed}^{-1}, \tag{3.28} $$

where $\boldsymbol{S}(\boldsymbol{\omega})$ is skew-symmetric matrix and described as Eq. (3.29)

$$ \boldsymbol{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{3.29} $$

Therefore, ${}^{E}\boldsymbol{\omega}_{E,Ed} = [\omega_x, \omega_y, \omega_z]^{\mathrm{T}}$ can be derived from Eq. (3.28) with the rotation matrix ${}^{E}\boldsymbol{R}_{Ed}$ contained in ${}^{E}\boldsymbol{T}_{Ed}$ and its time derivative ${}^{E}\dot{\boldsymbol{R}}_{Ed}$ contained in ${}^{E}\dot{\boldsymbol{T}}_{Ed}$.

$\boldsymbol{K}_{PO} = \mathrm{diag}(0.4, 0.4, 0.4)$[1/s] is spring constant. $\boldsymbol{K}_{VO} = \mathrm{diag}(0.1, 0.1, 0.1)$[dimensionless] is a positive definite diagonal matrix representing viscous damping coefficient.

The arm robot PA-10 (Mitsubishi Heavy Industries, Ltd.) used in this research has one redundancy degree of freedom. Because this paper does not consider redundancy, it is removed by setting the target angle q_{1d} of the first link to 0. Therefore, the target angle of each link from the target position of the hand can be determined by inverse kinematics. And the target joint angle $\boldsymbol{q}_{Hd} = [0, q_{2d}, ..., q_{7d}]^{\mathrm{T}}$ of the robot hand and the angular velocity $\dot{\boldsymbol{q}}_{Hd}$ are defined as follows.

$$\boldsymbol{q}_{Hd} = \boldsymbol{f}^{-1}({}^{W}\boldsymbol{\psi}_d) \tag{3.30}$$

$$\dot{\boldsymbol{q}}_{Hd} = \boldsymbol{K}_{PQ}(\boldsymbol{q}_{Hd} - \boldsymbol{q}_H) + \boldsymbol{J}^{+}(\boldsymbol{q}) \begin{bmatrix} {}^{W}\dot{\boldsymbol{r}}_d \\ {}^{W}\boldsymbol{\omega}_d \end{bmatrix} \tag{3.31}$$

Here, $\boldsymbol{f}^{-1}({}^{W}\boldsymbol{\psi}_d)$ represents inverse kinematics, and $\boldsymbol{K}_{PQ} = \mathrm{diag}(0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2)$[1/s] is a spring constant. The first term on the right side of Eq. (3.31) is feedback, and the second term gives speed and angular velocity as feedforward. $\boldsymbol{J}^{+}(\boldsymbol{q})$ is a pseudo inverse matrix of the Jacobian matrix obtained by partially differentiating the hand position/orientation with the joint angle vector.

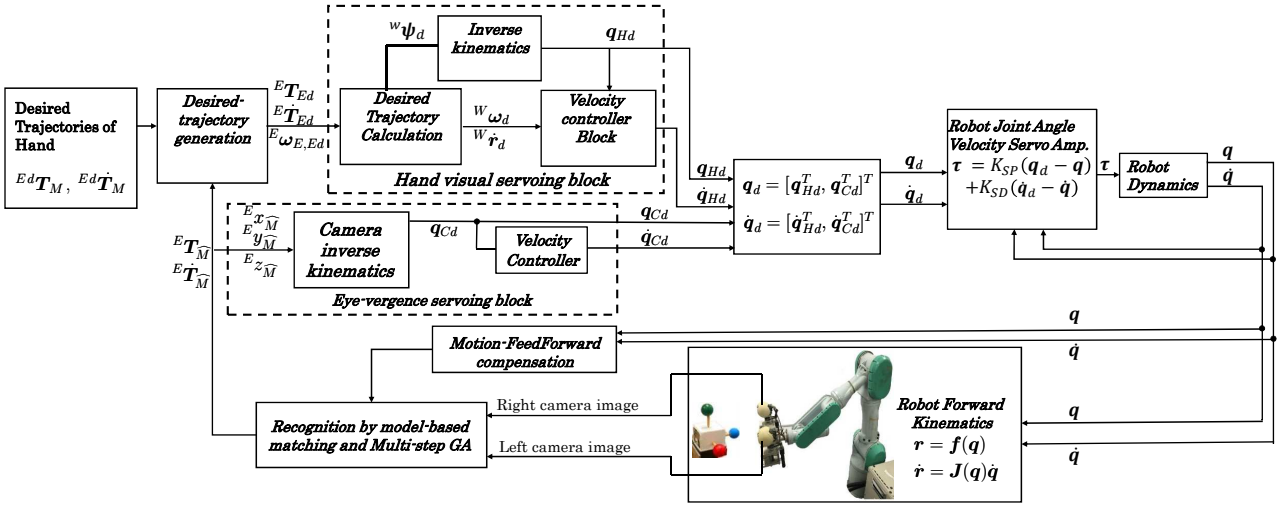


Fig. 3.15: Block diagram of the hand and eye-vergence visual servoing system.

3.6.5 Eye-vergence visual servoing controller

As shown in Fig. 3.14 eye-vergence visual servoing controller is the inner loop shown by the broken line. In this paper, two pan-tilt cameras are used for eye-vergence visual servoing. The cameras are attached to the hand and can rotate. q_8 represents the tilt angle common to the left and right cameras. And q_9 and q_{10} represent the pan angles. As shown in Fig. 3.16, ${}^{E}x_{\widehat{M}}$, ${}^{E}y_{\widehat{M}}$ and ${}^{E}z_{\widehat{M}}$ represent the position of the object detected in the hand coordinate $\vec{\Sigma}_E$. The block that controls eye-vergence is shown in the center of Fig. 3.15. Desired camera angle $\boldsymbol{q}_{Cd} = [q_{8Cd}, q_{9Cd}, q_{10Cd}]^{\mathrm{T}}$ is calculated by using the length defined in Fig. 3.16.

$$q_{8Cd} = atan2({}^{E}y_{\widehat{M}}, {}^{E}z_{\widehat{M}}) \tag{3.32}$$

$$q_{9Cd} = atan2(l_{8R} - {}^{E}x_{\widehat{M}}, {}^{E}z_{\widehat{M}}) \tag{3.33}$$

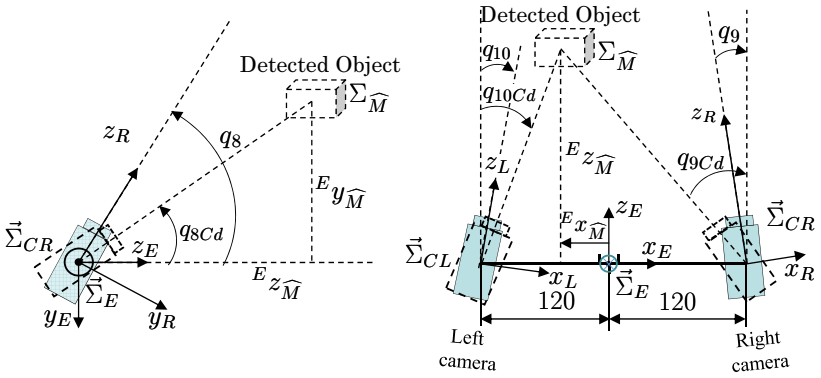
$$q_{10Cd} = atan2(l_{8L} + {}^{E}x_{\widehat{M}}, {}^{E}z_{\widehat{M}}) \tag{3.34}$$

Here, $l_{8L} = l_{8R} = 120$[mm] represents the position of the camera from the origin of $\vec{\Sigma}_E$, and the center line in the sight line of the camera is the z axis of left and right camera coordinates. The target joint angular velocity $\dot{\boldsymbol{q}}_{Cd} = [\dot{q}_{8Cd}, \dot{q}_{9Cd}, \dot{q}_{10Cd}]$ of eye-vergence is

$$\dot{q}_{iCd} = K_P(q_{iCd} - q_i) \qquad (i = 8, 9, 10) \tag{3.35}$$

$\dot{q}_{iCd}$ is input to the pulse motor for camera angle control as a pulse train. Here, $K_P = 1$ represents the spring constant. Further, $\boldsymbol{q}_d = [\boldsymbol{q}_{Hd}^{\mathrm{T}}, \boldsymbol{q}_{Cd}^{\mathrm{T}}]^{\mathrm{T}}$ and $\dot{\boldsymbol{q}}_d$ are constructed from $\boldsymbol{q}_{Hd}$, $\dot{\boldsymbol{q}}_{Hd}$ in equations Eq. (3.30), Eq. (3.31) and $\boldsymbol{q}_{Cd}$, $\dot{\boldsymbol{q}}_{Cd}$ in equations Eq. (3.32)$\sim$(3.35) Input torque to the robot using $\boldsymbol{\tau} = [\boldsymbol{\tau}_H^{\mathrm{T}}, \boldsymbol{\tau}_C^{\mathrm{T}}]^{\mathrm{T}}$ is determined by the following equation.

$$\boldsymbol{\tau} = \boldsymbol{K}_{SP}(\boldsymbol{q}_d - \boldsymbol{q}) + \boldsymbol{K}_{SD}(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) \tag{3.36}$$



(a) Tilt angle q_8 (b) Pan angle q_9 and q_{10}

Fig. 3.16: Definition of tilt and pan angles with relation of detected object

$\boldsymbol{K}_{SP}$ in the above equation is a spring constant and $\boldsymbol{K}_{SD}$ is a matrix representing viscous resistance. (3.36) is an operation within the servo amplifier of the robot, and the output of the controller is $\dot{\boldsymbol{q}}_{Hd}$ in Eq. (3.31) and $\dot{q}_{iCd}(i = 8, 9, 10)$ in Eq. (3.35). The control formula Eq. (3.36) of the robot's hand and camera gaze direction is located in the right block on the top of Fig. 3.15. The block that controls the position/orientation of the hand is shown on the left side of the upper row, and the control output torque $\boldsymbol{\tau}$ is determined together with the outputs $\boldsymbol{q}_{Cd}$, $\dot{\boldsymbol{q}}_{Cd}$ of the eye-vergence control block and the outputs $\boldsymbol{q}_{Hd}$, $\dot{\boldsymbol{q}}_{Hd}$ of the hand section doing.

The bottom row of Fig. 3.13 shows the forward kinematics of the robot. After correcting [52],[99] by MFF (Motion Feedforward) of the position/orientation of the marker captured in the camera image by this relation and the mapping to the left and right cameras, the moving image is real-time recognized by RM-GA.

3.6.6 Definition of gazing point

In order to evaluate whether the directions of the sight lines of cameras are controlled so as to take photos of the object at the center of the camera image, the gazing point of the camera is defined. As shown in Fig. 3.16 the point of intersection of the sight lines of the left and right cameras is defined as the gazing point of the cameras. As shown in Fig. 3.16 (a), because the two cameras are installed on a common plate and are rotated by q_8, the sight lines of the cameras have always an intersection in three-dimensional space. Because the gazing direction of the two

cameras is scanned in the $x_E - z_E$ plane of $\vec{\Sigma}_E$ which is fixed to the hand, the y coordinate of the gazing point represented by $\vec{\Sigma}_E$ is always -100[mm] .

The motion of object M, end-effector E,detected object $\widehat{M}$, and gazing point G in the x-axis of Σ_{EC} are represented by ${}^{EC}x_M$, ${}^{EC}x_E$, ${}^{EC}x_G$ and ${}^{EC}x_{\widehat{M}}$, as shown in Fig. 4.3. The distance between object and end-effector (hand) is expressed as

$$\Delta i_{ME} = {}^{EC}i_M - {}^{EC}i_E, (i = x, y, z). \tag{3.37}$$

Tracking error of detection is

$$\Delta i_{M\widehat{M}} = {}^{EC}i_M - {}^{EC}i_{\widehat{M}}, (i = x, y, z). \tag{3.38}$$

Tracking error of end-effector(hand) is

$$\Delta i_{EdE} = {}^{EC}i_{Ed} - {}^{EC}i_E, (i = x, y, z). \tag{3.39}$$

Tracking error of the gazing point is

$$\Delta i_{MG} = {}^{EC}i_M - {}^{EC}i_G, (i = x, y, z). \tag{3.40}$$

As shown in Fig. 3.18 and Eq. (4.4), the desired value between object and end-effector is $\Delta x_{ME} = 0$, $\Delta y_{ME} = -100$[mm], $\Delta z_{ME} = 545$[mm]. And of cause, the desired tracking error between gazing point and end-effector is 0, i.e., $i_{M\widehat{M}} = 0$ and $i_{EdE} = 0$.

As shown in Fig. 3.20, the gazing point based on $\vec{\Sigma}_E$ is represented by q_9 and q_{10}. And $0 < q_9, q_{10} < \pi/2$. The following equation can be obtained.

$$\frac{{}^{E}z_G}{120 + {}^{E}x_G} = tan(\frac{\pi}{2} - q_{10}) \tag{3.41}$$

$$\frac{{}^{E}z_G}{240 - {}^{E}x_G} = tan(\frac{\pi}{2} - q_9) \tag{3.42}$$

Therefore,

$$^{E}x_G = \frac{240tan(\frac{\pi}{2} - q_9)}{tan(\frac{\pi}{2} - q_{10}) + tan(\frac{\pi}{2} - q_9)} - 120, \tag{3.43}$$

$$^{E}z_G = \frac{240tan(\frac{\pi}{2} - q_{10})tan(\frac{\pi}{2} - q_9)}{tan(\frac{\pi}{2} - q_{10}) + tan(\frac{\pi}{2} - q_9)}]^{\mathrm{T}}. \tag{3.44}$$

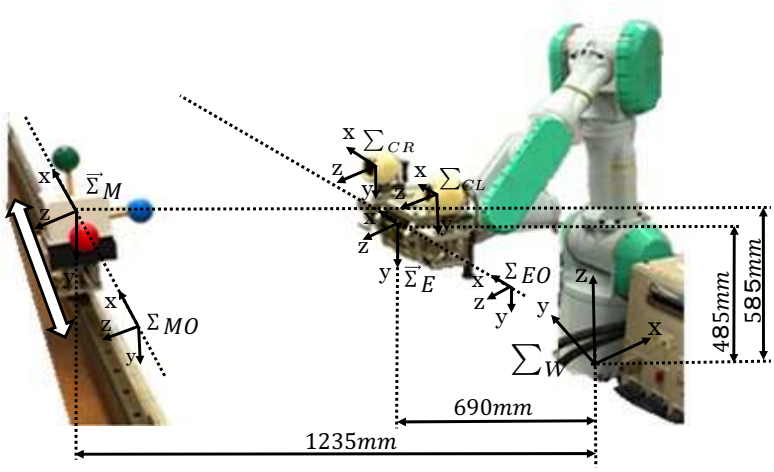


Fig. 3.17: The 3D marker and the eye-vergence visual servoing system

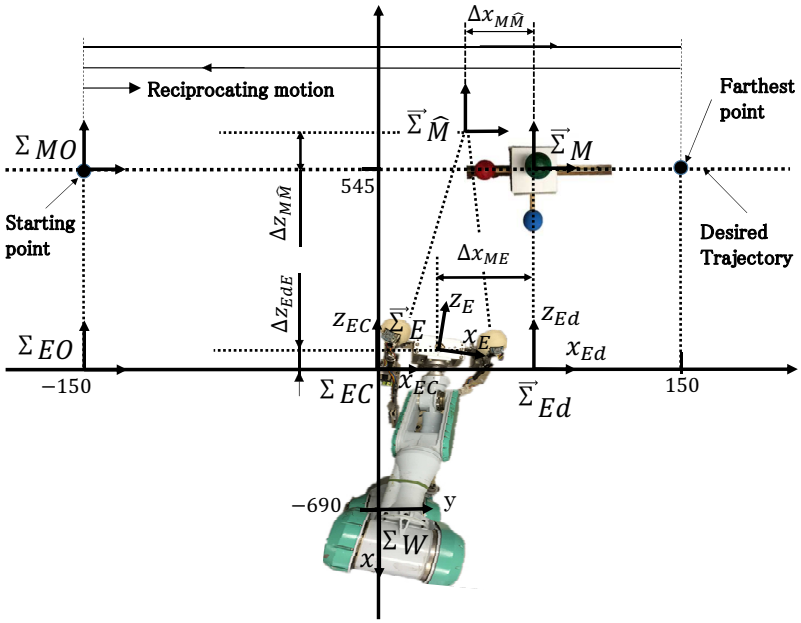


Fig. 3.18: Target object and definition of coordinates depicted in the x-z plane of Σ_{EC}. Initial position of the object is represented by Σ_{MO}; actual object $\vec{\Sigma}_M$; detected object $\vec{\Sigma}_{\widehat{M}}$; initial position of the hand Σ_{EO}; actual end effector $\vec{\Sigma}_E$; and desired end-effector $\vec{\Sigma}_{Ed}$. At this moment orientation ${}^{E}\Delta\varepsilon$ in Eq. (3.27) is not zero vector.

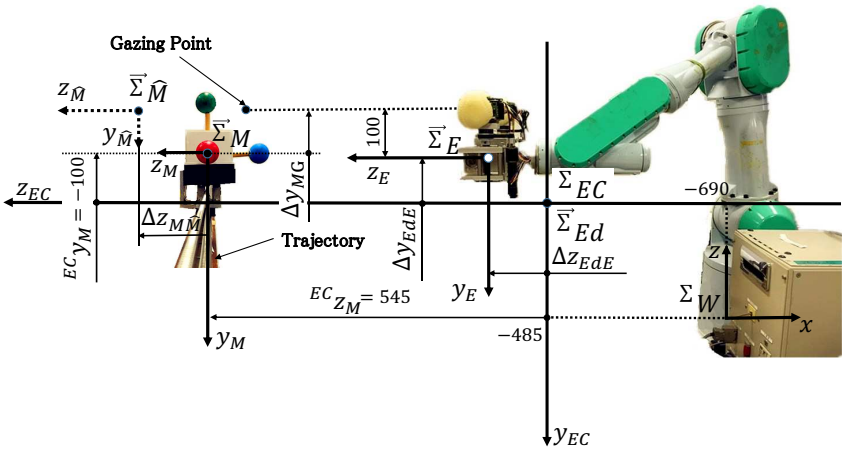


Fig. 3.19: 3D maker and coordinates in the y-z plane of Σ_{EC}.

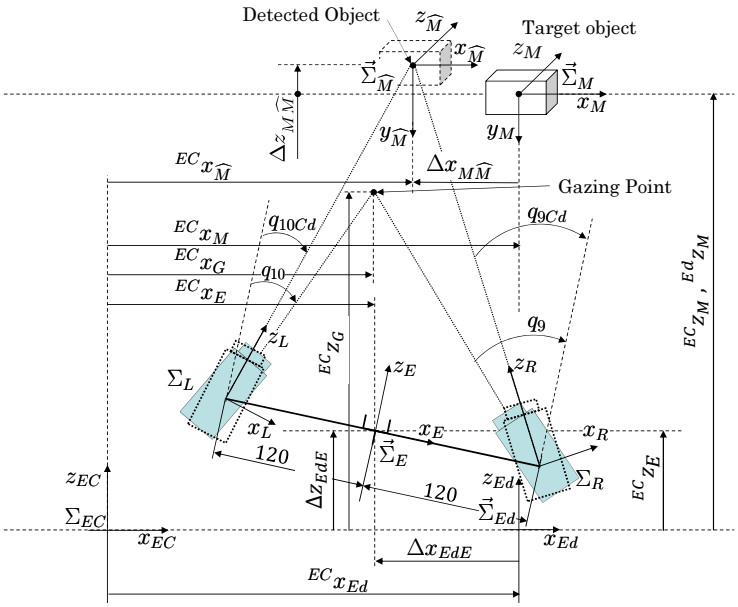


Fig. 3.20: Enlarged drawing of Fig. 3.18 with gazing point. As shown in Eq. (4.4) ${}^{Ed}\boldsymbol{\psi}_M$ is desired pose relationship between object and end-effector with respect to coordinate frame $\vec{\Sigma}_{Ed}$.

Chapter 4

Experiment of Model-Based Eye-Vergence Visual Servoing System

4.1 Lateral visual servoing

In order to confirm the tracking ability of the proposed eye-vergence visual servoing system, tracking experiments have been conducted in which the target object reciprocates along a straight lateral trajectory. Experiments were conducted to verify the effectiveness of the hand & eye-vergence visual servoing system through PA-10 robot arm manufactured by Mitsubishi Heavy Industries, LTD. And two rotational cameras manufactured by Sony Industries are mounted on the end-effector. The resolution of dynamic images is 640×480 [pixel]. The frame frequency of stereo cameras is set as 30[fps]. These experiments are divided in to three groups, i.e., x-position tracking, 3DOF position visual servoing, and 6DOF position/orientation visual servoing. Each group includes several experiments in which the angular velocity of the object are set as ω=0.314, 0.628, and 1.256[rad/s] separately.

4.1.1 Experiment condition

As shown in Figs. 3.18 and 3.17, EO, MO and EC represent initial hand pose, initial object pose and midpoint of round-trip tracking movement of the end-effector respectively. Therefore their coordinate systems are defined as Σ_{EO}, Σ_{EC} and Σ_{MO} separately.

The homogeneous transformation matrix from Σ_W to Σ_{EC} and Σ_{MO} are:

$$ {}^{W}\boldsymbol{T}_{EC} = \begin{bmatrix} 0 & 0 & -1 & -690[\text{mm}] \\ 1 & 0 & 0 & 0[\text{mm}] \\ 0 & -1 & 0 & 485[\text{mm}] \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1} $$

$$ {}^{W}\boldsymbol{T}_{MO} = \begin{bmatrix} 0 & 0 & -1 & -1235[\text{mm}] \\ 1 & 0 & 0 & -150[\text{mm}] \\ 0 & -1 & 0 & 585[\text{mm}] \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.2} $$

Target object motion function is

$$ {}^{MO}z_M(t) \;=\; 150 - 150\cos(\omega t)[\text{mm}]. \tag{4.3} $$

Target position and orientation relationship between the object and the end-effector is set as:

$$ {}^{Ed}\boldsymbol{\psi}_M = [0, -100[\text{mm}], 545[\text{mm}], 0, 0, 0]. \tag{4.4} $$

The object is subjected to reciprocating motion of the sine wave in orbit. Pose relationship of the coordinate system of the object and the visual servoing system are shown in Fig. 3.17.

4.1.2 Tracking experiment without recognition

The tracking of the manipulator by visual servoing includes both time lag of recognition and motion delay of the robot. In this section the motion trajectory of the object is given to to robot. The robot do not need to detect the pose of 3D marker. In this case, Eq. (3.38) satisfies $\Delta i_{M\widehat{M}} = 0$. This is equivalent to assuming a situation where there is no delay in recognition and no recognition error, and it is possible to consider only the motion control performance of the robot.

The experimental results in the case of directly giving the position/orientation indication value are shown in Fig. 4.1. (a) is the situation of $\omega = 0.314[\mathrm{rad/s}]$(Period 20[s]), (b) is 0.628[rad/s](Period 10[s]) and (c) is 1.256[rad/s](Period 5 [s]). ${}^{EC}x_M$ is the x coordinate of the object, and the dotted line marked ${}^{EC}x_E$ represents the tracking result of the end-effector. The solid line with ${}^{EC}x_G$ represents the trackability of the eye-vergence system. In (a) $\sim$ (c), all ${}^{EC}x_G$ match ${}^{EC}x_M$, and the gazing point follow the object position without delay. As shown in Fig. 4.1(d) and (e), each experiment is carried out for 60[s]. Furthermore, the gain curves and the phase curves are depicted by the experimental data in the periods of 30[s] (0.209[rad/s]), 25[s] (0.251[rad/s]), 20[s] (0.314[rad/s]), 15[s] (0.419[rad/s]), 10[s] (0.628[rad/s]) and 5[s] (1.256[rad/s]). The solid line ① in (d) is $20log(a_{xG}/a_{xM})$ calculated with a_{xM} and a_{xG} defined in (a). Furthermore, the dashed line ② in (d) shows $20log(a_{xE}/a_{xM})$. In the phase curve of (e), the solid line ① represents $\angle^{EC}x_G - \angle^{EC}x_M$, the dashed line ② represents $\angle^{EC}x_E - \angle^{EC}x_M$. The gain and phase curves of the eye-vergence system are represented by ① in (d) and (e) respectively. On the other hand, the gain and the phase curves of the hand are represented by ② in (d) and (e) respectively. Up to $\omega = 0.24[\mathrm{rad/s}]$, the x coordinate position of the hand can follow the object. Moreover, it can be seen that the gain characteristic of the hand represented by the broken line ② in (d) can be approximated by the first order lag system. The break frequency is $\omega = 0.24[\mathrm{rad/s}]$ indicated by Ⓐ in (d). It can be seen that the gain decreases due to the increase of the frequency of the motion of the tracked object in (d). The amplitude of the hand gradually becomes smaller than that of the gazing point of the camera. The phase

of the hand shown in ② of (e) is also delayed. However, the gain characteristic of the eye-vergence system represented by $20log(a_{xG}/a_{xM})$ in ① in (d) is almost 0[dB]. Therefore, in the range up to $\omega = 1.256[\mathrm{rad/s}]$, the system can constantly capture objects in the center of the cameras' field of vision. There is hardly delay in the phase of eye-vergence shown in ① of (e). In summary, the motion characteristic of eye-vergence is superior to that of the hand. And it is expected that the tracking performance to the object is good.

4.1.3 Position 3DOF visual servoing experiment

Figure 4.2 shows the x, y and z coordinates of the object recognized by RM-GA and also shows the time response waveforms of the object, end-effector and gazing point when visual servoing control is performed. For orientation, the correct value $\varepsilon = 0$ is always given. From Eq. (4.2), the target orientation of the object is $\varepsilon_d = [0, 0, 0]$. And in Eq. (3.27) ${}^{E}\Delta\varepsilon = 0$. In this case, only the position is controlled by visual feedback, so the orientation of the end-effector is consistent with ε_d, and there is no error.

Comparing (a), (b) and (c) of Fig.4.2 with those of Fig. 4.1, it can be seen that the derivation between the position of the gazing point ${}^{EC}x_G$ and that of the object ${}^{EC}x_M$ is larger than the corresponding derivation in Fig. 4.1. That is, recognition error exists. The data of (d) and (e) of Fig.4.2 are the maximum amplitude of the tracking experiment result of the object, gazing point and end-effector of the periods 30[s], 25[s], 20[s], 15[s], 10[s] and 5[s]. And it is shown in the same way in Fig. 4.1. The track characteristic of the end-effector shown in ② of Fig.4.2 is not much different from that at the time of Fig. 4.1.

In addition, as shown in ① of (e) of Fig.4.2, the track characteristic of the phase of the eye-vergence system is slightly delayed as the motion speed of object increases. Therefore, although the system is slightly delayed due to the recognition process, it can be seen from ① of (d) the tracking performance of the object represented by the gain curve does not substantially affect.

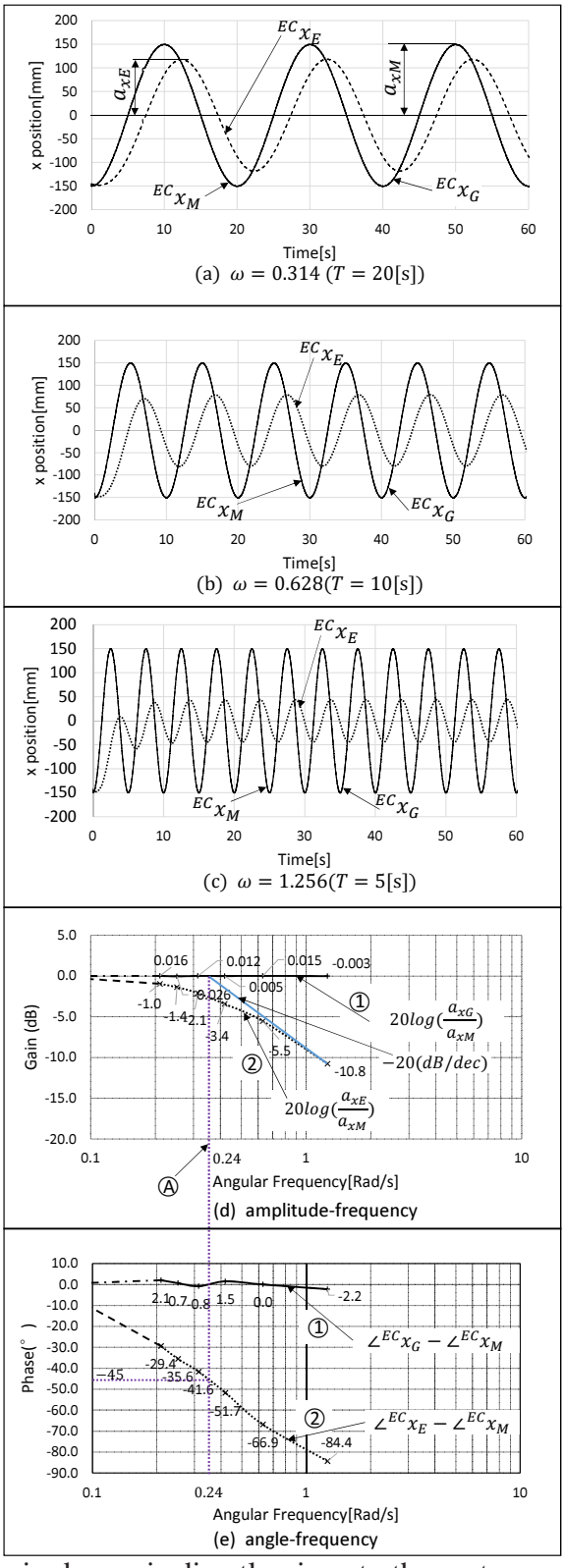


Fig. 4.1: True object's desired pose is directly given to the system, which guarantees the pose tracking recognition error to be zero. So in this figure, only the delays made by dynamic influences is evaluated. And the figure shows the camera can track the object much better than the end-effector.

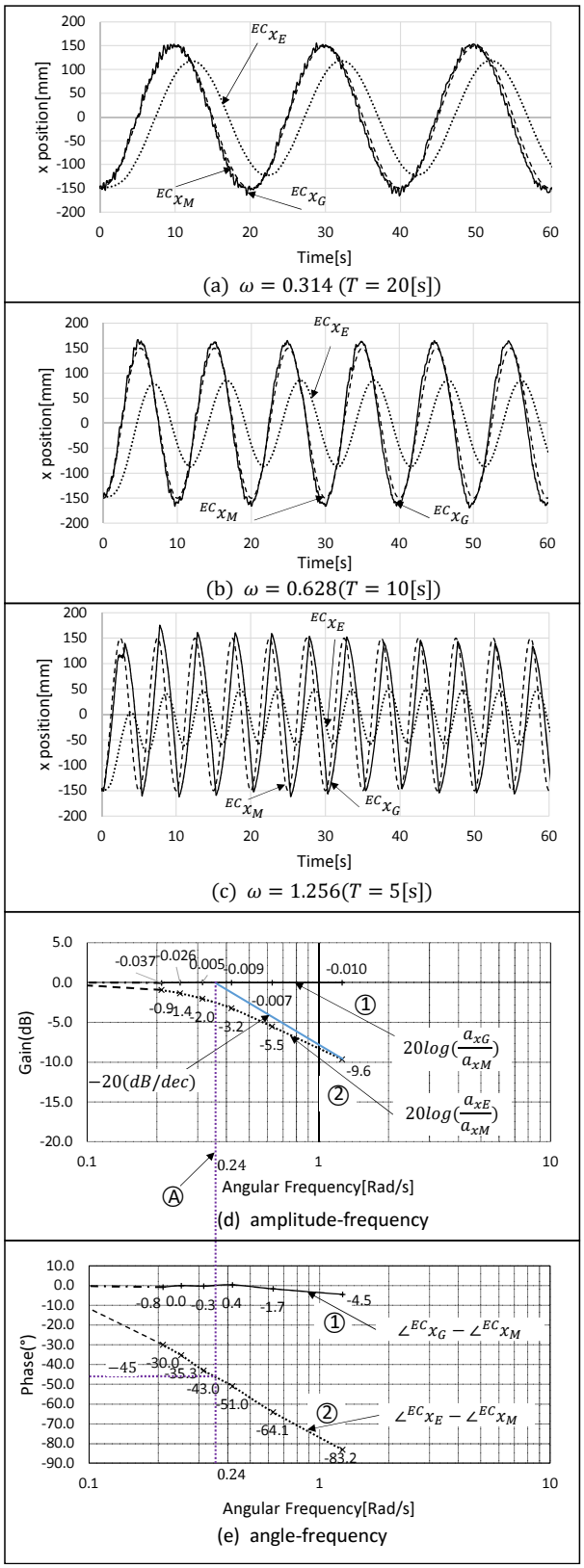


Fig. 4.2: The object's pose ε_1, ε_2 and ε_3 are assumed to be given to servoing controller and the object's pose x, y and z are recognized by camera.

4.1.4 Pose 6DOF visual servoing experiment

Real-time recognition of the six variables of position/orientation using RM-GA and control of hand position/orientation and Gazing Point have been performed. Figure 4.3 shows position x response examining results with $\omega = 0.314, 0.628, 1.256$ [rad/s].

In Fig. 4.3, (a.1) shows data ${}^{EC}x_M$, ${}^{EC}x_E$, ${}^{EC}x_G$ of $\omega = 0.314$ (T=20[s]), (b.1) shows their data of $\omega = 0.628$ (T=10[s]), (c.1) shows their data of $\omega = 1.256$ (T=5[s]). (a.2), (b.2), and (c.2) show Δx_{EdE}, which means the delay of the hand in the x-axis direction, and Δx_{MG}, which means the delay of the eye-vergence. Comparing ${}^{EC}x_G$ shown in Fig. 4.2 (a) $\sim$ (c) and that shown in Fig. 4.3 (a) $\sim$ (c) , the error of ${}^{EC}x_G$ in Fig. 4.3 (a) and (b) increases. In particular, it can be clearly understood by comparing the time around 20 [s] and 40 [s] in both figures.

As shown in (c.1) of Fig. 4.3, the amplitude of ${}^{EC}x_G$ is almost the same as the amplitude of ${}^{EC}x_M$ (150[mm]). It can be seen that the position of the gazing point on the x axis ${}^{EC}x_G$ follows the object ${}^{EC}x_M$. However, there is a delay in the phases of ${}^{EC}x_G$ comparing with ${}^{EC}x_M$. As shown in Fig. 4.3 (c.1), the end-effector amplitude ${}^{EC}x_E$ is reduced to approximately 50[mm], while ${}^{EC}x_M$ is in amplitude 150[mm] periodic motion. It can be seen that although the RM-GA is identifying, the tracking range of the end-effector reduces significantly. But ${}^{EC}x_G$ can still follow, the fact indicates that the inertia of the entire robot is large, and the inertia of the eye itself is small, which is conducive to continuous tracking. In (a.2), Δx_{MG} is very small, and Δx_{EdE} vibrates about 80 [mm].

Therefore, in the T=20[s] experiment, the end-effector tracking has been slow, but there is no delay in the tracking of gazing point. Under the condition of (b.2) T=10[s], the amplitude of Δx_{EdE} further increases to 110 [mm]. The effect of eye-vergence can be seen from (a.2) and (b.2). In (c.2), Both Δx_{MG} and Δx_{EdE} have large fluctuations, and gazing point cannot track 3D marker.

Figure 4.3 evaluates the tracking characteristics of the 3D marker represented by Σ_{EC} in the x-axis direction. Figures 4.4 and 4.5 are evaluated in the y-axis and z-axis directions result. They differ from the frequency response in the x-axis direction, because the 3D marker does not

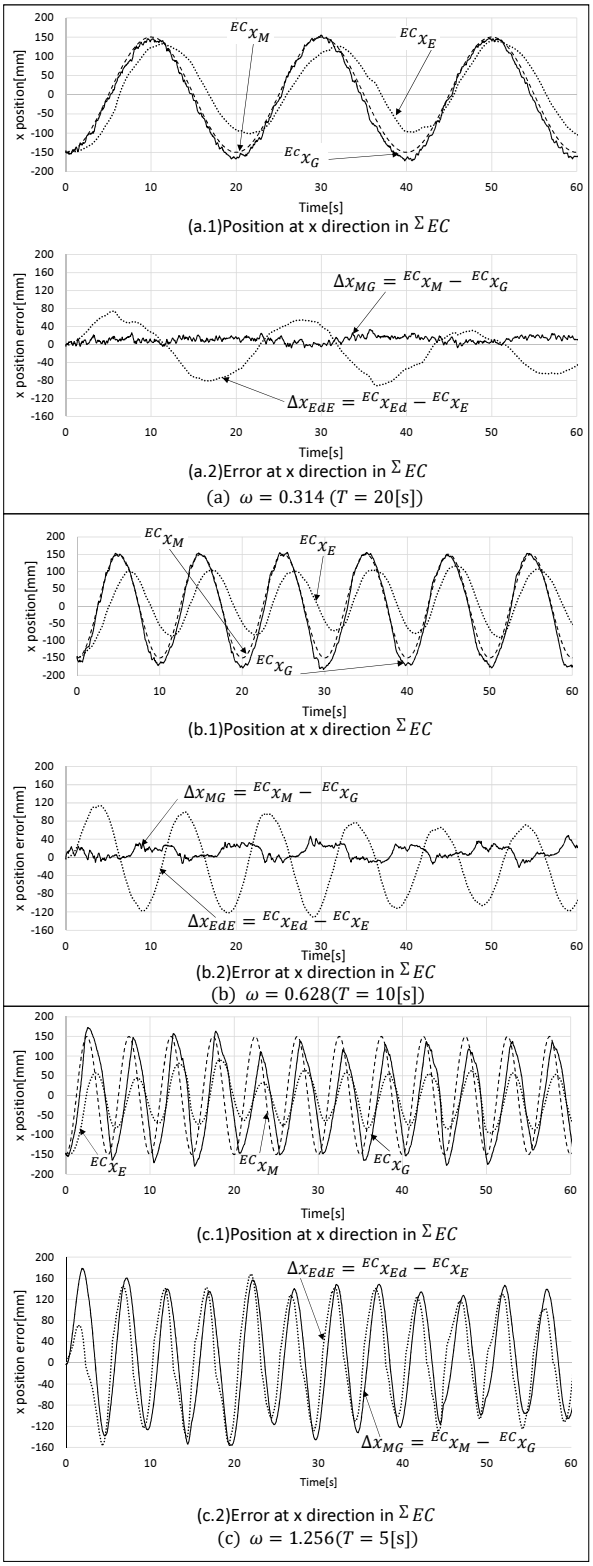


Fig. 4.3: Movements of end effector $^{EC}x_E$ and gazing point $^{EC}x_G$ on the x-axis direction in the center coordinate system of hand Σ_{EC}. On condition that the object's pose x, y, z, ε_1, ε_2 and ε_3 are recognized RM-GA.

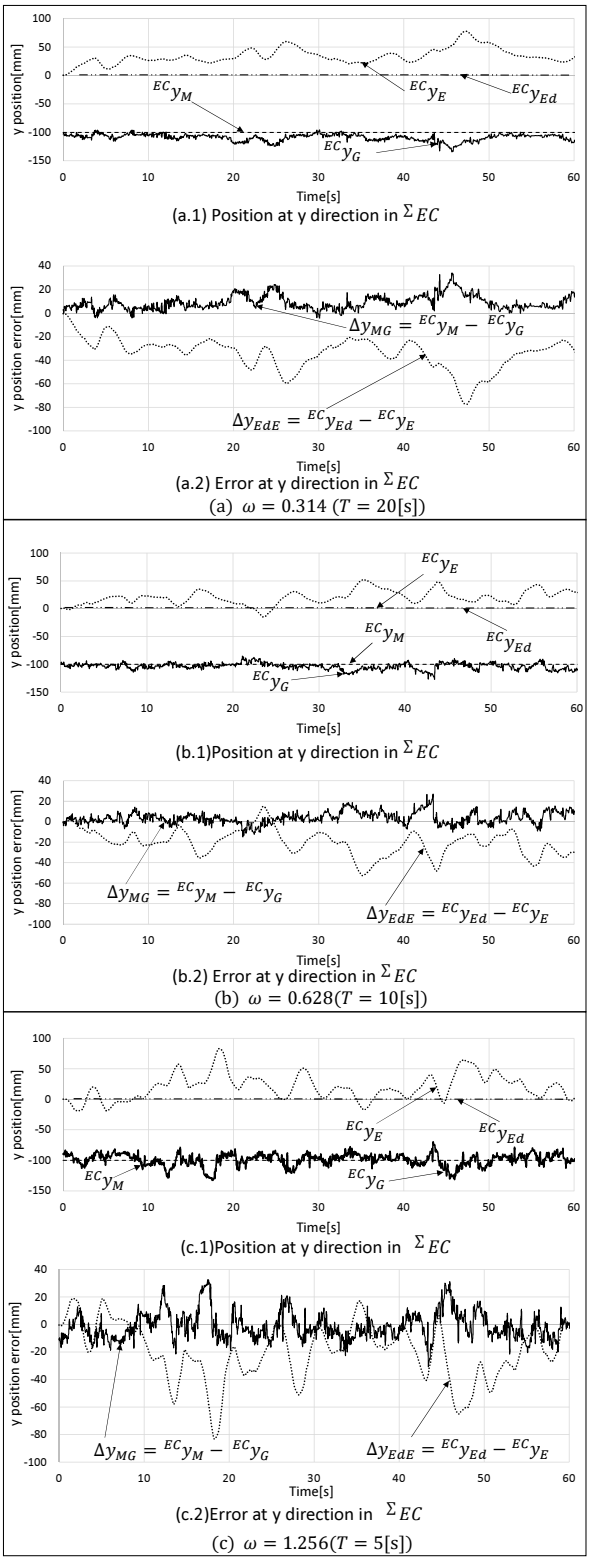


Fig. 4.4: Movements of actual object $^{EC}y_M$, end effector $^{EC}y_E$, gazing point $^{EC}y_G$ and desired end effector position $^{EC}y_{Ed}$ on the y-axis direction in the center coordinate system of hand Σ_{EC} on condition that the object's pose x, y, z, ε_1, ε_2 and ε_3 are recognized by RM-GA.

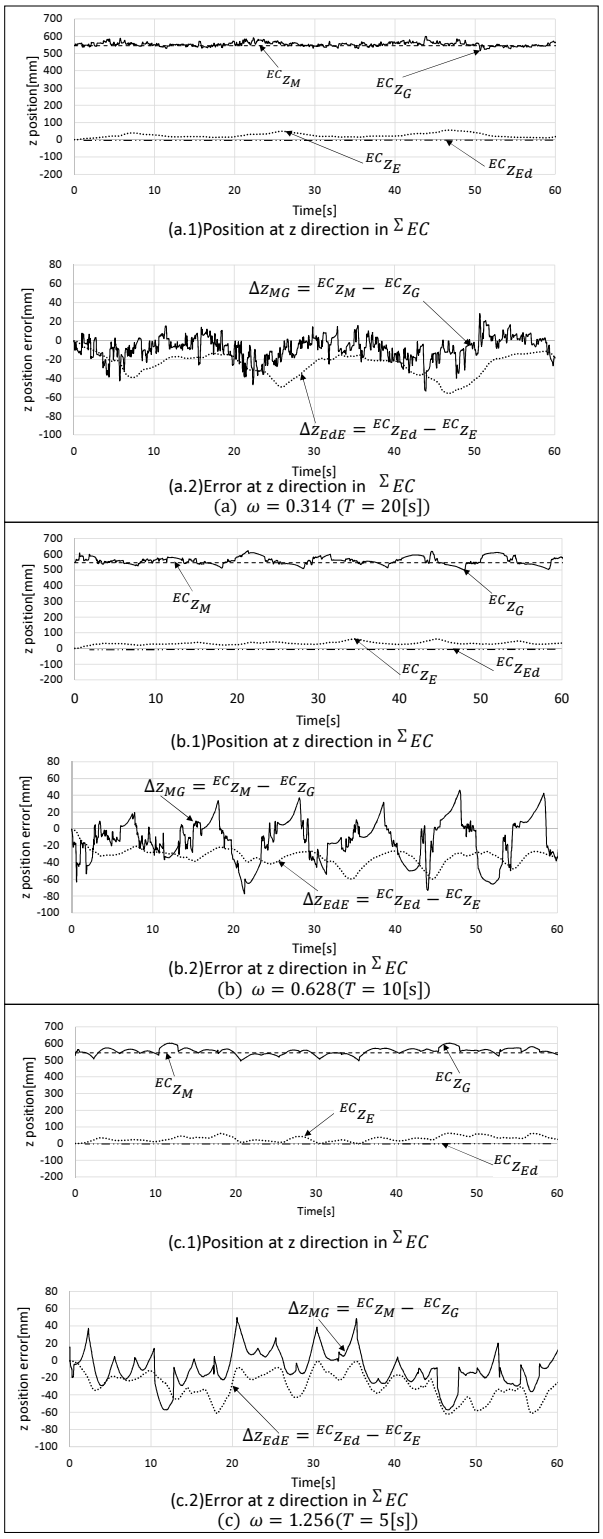


Fig. 4.5: Movements of actual object $^{EC}z_M$, end effector $^{EC}z_E$, gazing point $^{EC}z_G$ and desired end effector position $^{EC}z_{Ed}$ on the z-axis direction in the center coordinate system of hand Σ_{EC} on condition that the object's pose x, y, z, ε_1, ε_2 and ε_3 are recognized by RM-GA.

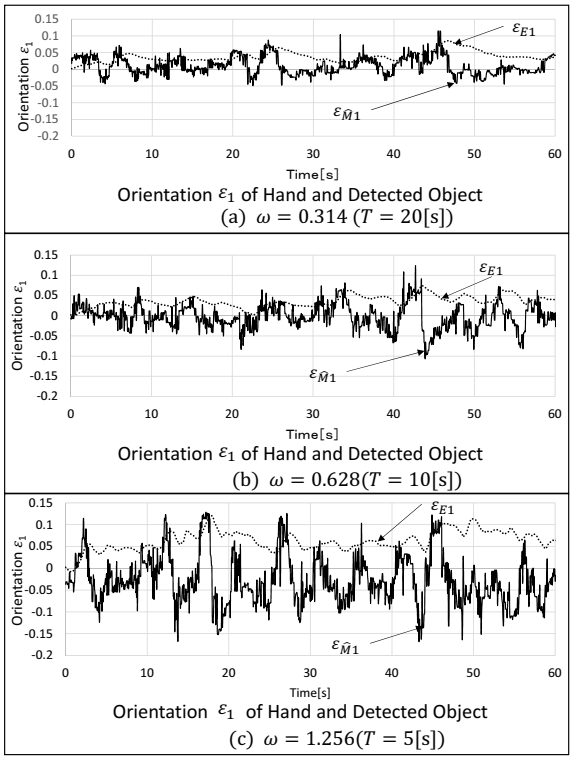


Fig. 4.6: Changes of orientation ε_1 of hand and detected object during tracking movement. Target values are ${}^{EC}\varepsilon_{E1} = 0$ and ${}^{EC}\varepsilon_{M1} = 0$ respectively.

move in the y- and z-axis directions, so the position target value is always zero. In the frequency response experiment shown in Fig. 4.3, the y-axis motion result of each variable is shown in Fig. 4.4. As shown in Fig. 3.19, the camera is 100[mm] higher than the hand in the y-axis direction, and the positive direction of the y-axis is downward, so the target position of the hand shown in Fig. 4.4 is ${}^{EC}y_{Ed} = 0$, the target position of the gazing point is -100[mm].

Comparing (a.1), (b.1) and (c.1) in Fig. 4.4, it is shown that no periodic oscillation occurs. It can be seen that the gazing point ${}^{EC}y_G$ oscillates around -100 [mm], the same height as the object. In addition, the magnitude of the gazing point deviation Δy_{MG} is smaller than the hand error Δy_{EdE}. As the frequency of marker movement increases, the fluctuation of the motion trajectory of the hand and the gazing point gradually increases. In addition, like ${}^{EC}y_E$, from (a) to (b) to (c), as the marker's movement frequency increases, the fluctuation of the gazing point ${}^{EC}y_G$ increases.

(a.1), (b.1), and (c.1) of Fig. 4.5 represent response curves related to the z-axis direction

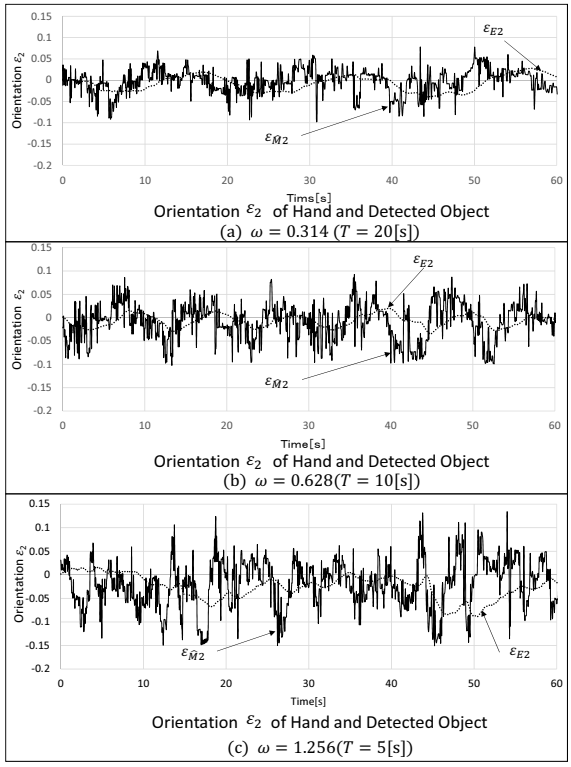


Fig. 4.7: Changes of orientation ε_2 of hand and detected object during tracking movement. Target values are ${}^{EC}\varepsilon_{E2} = 0$ and ${}^{EC}\varepsilon_{M2} = 0$ respectively.

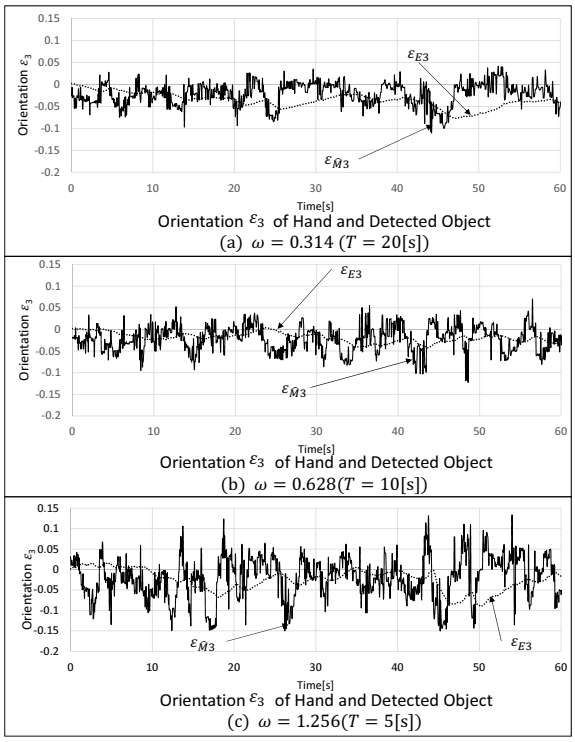


Fig. 4.8: Changes of orientation ε_3 of hand and detected object during tracking movement. Target values are ${}^{EC}\varepsilon_{E3} = 0$ and ${}^{EC}\varepsilon_{M3} = 0$ respectively.

of the hand and gazing point with $\omega = 0.314, 0.628, 1.256$[rad/s] respectively. It can be seen from the figure that the system can continuously detect the object and can maintain a set distance from the object to track. However, as shown in (a.2), (b.2), and (c.2), recognition error Δz_{MG} increases as the frequency of movement of the target object increases. Therefore, the tracking stability of the end-effector in (c.2) reduces. Because Δz_{MG} has less vibration than Δz_{EdE}, compared with the fixed camera system, the eye-vergence system has better tracking performance and stability.

Figures 4.6, 4.7, and 4.8 are orientation tracking results of the hand and the detection results of the object corresponding to Figs 4.3 ∼4.5. About the orientation of the recognized object $\boldsymbol{\varepsilon}_{\widehat{M}} = [\varepsilon_{\widehat{M}1}, \varepsilon_{\widehat{M}2}, \varepsilon_{\widehat{M}3}]^{\mathrm{T}}$ and the end-effector $\boldsymbol{\varepsilon}_E = [\varepsilon_{E1}, \varepsilon_{E2}, \varepsilon_{E3}]^{\mathrm{T}}$, Fig. 4.6 shows $\varepsilon_{\widehat{M}1}$and ε_{E1}, Fig. 4.7 shows $\varepsilon_{\widehat{M}2}$ and ε_{E2}, and Fig. 4.8 shows $\varepsilon_{\widehat{M}3}$and ε_{E3}. In the experiment, because the object does not rotate, the actual pose of the object is ${}^{EC}\boldsymbol{\varepsilon}_M = \mathbf{0}$. As shown in Eq.(4.5), the target pose of the hand is ${}^{EC}\boldsymbol{\varepsilon}_{Ed} = \mathbf{0}$. Therefore, all response curves represent both tracking and detection errors.

Figure 4.6 represents the recognized orientation $\varepsilon_{\widehat{M}1}$ of the object rotating around the x-axis of Σ_{EC} and the orientation response of the hand ε_{E1}. The recognition results $\varepsilon_{\widehat{M}1}$ at each frequency fluctuates around the true value of 0. Comparing these three graphs, it can be seen that the error of the recognition $\varepsilon_{\widehat{M}1}$ increases as the frequency of movement of the target object increases. Therefore, the hand tracking error also increases.

Figure 4.7 represents the recognized orientation $\varepsilon_{\widehat{M}2}$ of the object rotating around the y-axis of Σ_{EC} and the orientation response of the hand ε_{E2}. (a), (b), (c) show the experimental data ε_{E2}, $\varepsilon_{\widehat{M}2}$ at periods T = 20, 10, 5[s]. These results show that the periodic movement in the x-axis direction of Σ_{EC} will affect the orientation in the y-axis direction. Since the hand movement of the robot has a delay in the direction of ${}^{EC}x$, according to the function of eye-vergence, the camera will rotate around y_E axis of $\vec{\Sigma}_E$ to observe the target object . The author think that caused an orientation detection error of $\varepsilon_{\widehat{M}2}$ around the y-axis.

Figure 4.8 represents the recognized orientation $\varepsilon_{\widehat{M}3}$ of the object rotating around the y-axis

of Σ_{EC} and the orientation response of the hand ε_{E3}. Same as the Fig. 4.6, the recognition result of each frequency vibrates slightly. These results do not indicate the actual hand vibration, but indicate that the convergence of the RM-GA vibrates near the detected object.

Through the frequency response experiments, the eye-vergence system's tracking performance, keeping the target in the field of view, is verified. Through detailed comparison and analysis of the visual servoing results of the six pose variables, it can be confirmed that the tracking performance of the eye-vergence system is better than that of the end-effector.

4.2 Arc swing motion tracking experiment under different light conditions

4.2.1 Fitness distribution under different illumination

For practical application, the light condition is an important effect element for the visual servoing system to recognize the target object. Using still pictures at an instant moment, the fitness value is calculated with model's pose varied as parameters. We call it "fitness distribution." It is a way to verify whether the RM-GA can detect the true pose of a target object at that moment. Therefore, to verify the detection capability of RM-GA under different illumination conditions, the fitness distribution experiment in ${}^{E}x - {}^{E}z$ plane is conducted.

Figure 4.9 shows the searching area of RM-GA that is defined based on the range of motion of the object. Target position and orientation relationship between the object and the end-effector is set as:

$$ {}^{E}\boldsymbol{\psi}_{M} = [0, -100[\text{mm}], 545[\text{mm}], 0, 0, 0]. \tag{4.5} $$

According to pre-set tracking conditions of Eq. (4.5) and a number of tests, we set the search area of RM-GA as

$$ {}^{E}x_{\widehat{M}} \in [-200, 200], {}^{E}y_{\widehat{M}} \in [-195, 5], {}^{E}z_{\widehat{M}} \in [350, 750], (\text{unit:[mm]}). \tag{4.6} $$

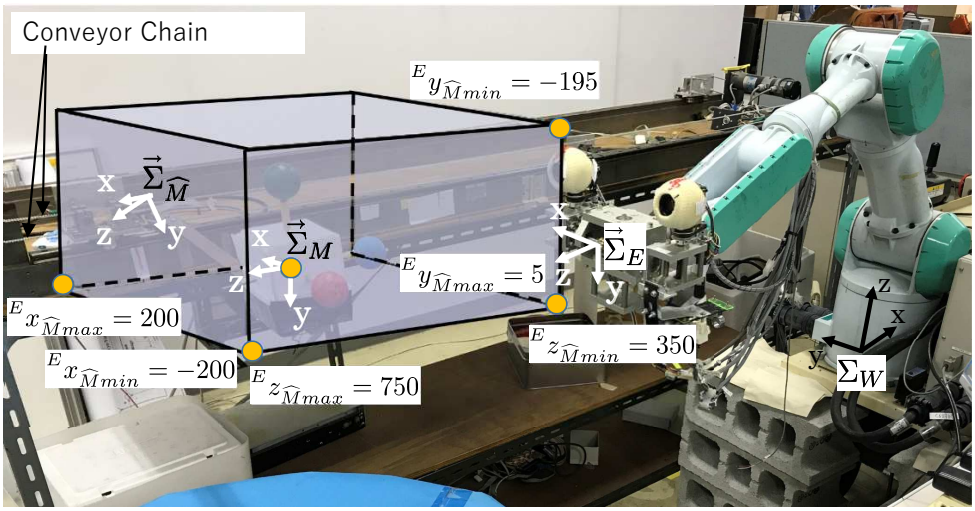


Fig. 4.9: Searching area of GA. The origins of models generated by GA are in a cuboid space. Its range of the target object is ${}^{E}x_{\widehat{M}} \in [-200, 200], {}^{E}y_{\widehat{M}} \in [-195, 5], {}^{E}z_{\widehat{M}} \in [350, 750]$, unit:[mm].

The object $\vec{\Sigma}_M$ and the end-effector $\vec{\Sigma}_E$ do not move in the experiment. And the relative pose of $\vec{\Sigma}_M$ and $\vec{\Sigma}_E$ is the same as Eq. (4.5).

Figure 4.10 shows the results of fitness distribution in different illumination conditions. The images on the right side of each row are taken by the two cameras in different experimental conditions. In (a)∼(d), only the illumination is changed, the object and the arm are fixed. The left two columns (a1∼ d1 and a2∼ d2) of Fig. 4.10 are fitness distributions in ${}^{E}x - {}^{E}z$ plane under different experimental conditions. Fitness is calculated by Eq. (3.22). The distribution of the middle column (a2∼ d2) is the 2D display of a1∼d1.

For example, in the row (b), the two images are taken at 500[lx]. In the case of given true values ${}^{E}\boldsymbol{\varepsilon}_{\widehat{M}} = {}^{E}\boldsymbol{\varepsilon}_{M} = \mathbf{0}, {}^{E}y_{\widehat{M}} = {}^{E}y_{M} = -100$[mm], the target object is searched on the ${}^{E}x - {}^{E}z$ plane as shown in Fig. 3.18. And the fitness distribution is shown as Fig. 4.10 (b1). The (b2) is a 2D figure of (b1). There are two highest points (vertex), i.e., the peak of the mountain of the distribution. One is $({}^{E}x_{1\widehat{M}}, {}^{E}z_{1\widehat{M}}) = (4, 540)$[mm]. And the other is $({}^{E}x_{2\widehat{M}}, {}^{E}z_{2\widehat{M}}) = (4, 546)$[mm]. That means, according to the model-based matching method, object is most likely to be in either $({}^{E}\boldsymbol{r}_{1\widehat{M}}, {}^{E}\boldsymbol{\varepsilon}_{1\widehat{M}}) = (4, -100, 540, 0, 0, 0)$ or $({}^{E}\boldsymbol{r}_{2\widehat{M}} {}^{E}\boldsymbol{\varepsilon}_{2\widehat{M}}) =$

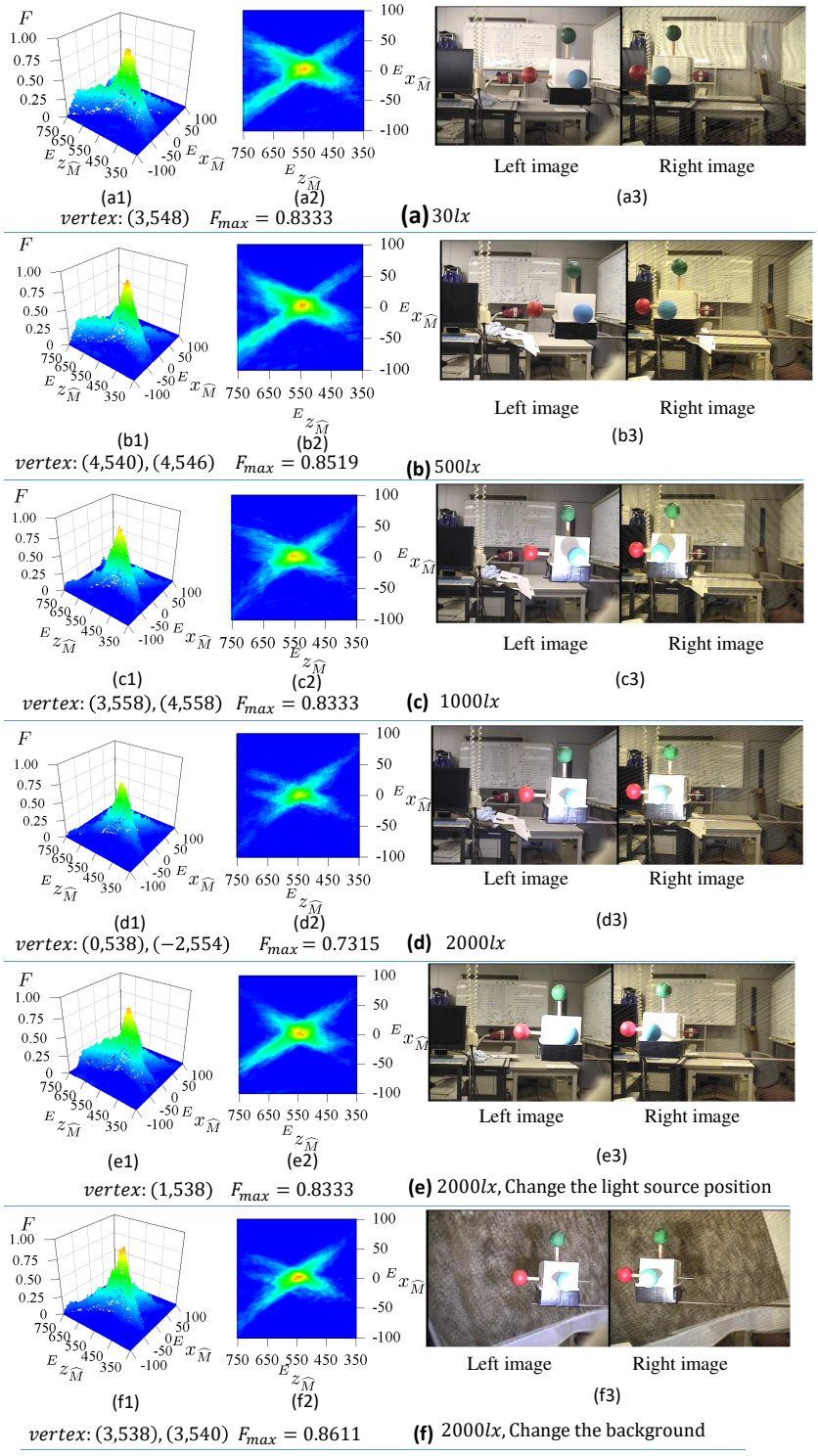


Fig. 4.10: Fitness distribution under different illumination. (a)∼(d) show the results of experiments with different illumination. In (e), the light source position changes. In (f), the background changes and the illumination is same with (d). (a3)∼(f3) show the left and right images in each experiment. (a1)∼(f1) show the distribution of fitness on each point on ${}^{E}x - {}^{E}z$ plane in search area. Exploration interval is 1[mm], i.e. ${}^{E}x = -100, -99, ..., 99, 100; {}^{E}z = 350, 351, ..., 749, 750$[mm]. (a2)∼(f2) are the 2D figure of (a1)∼(f1). In each experiment, "vertex" show the position $({}^{E}x_{\widehat{M}}, {}^{E}z_{\widehat{M}})$ with maximum fitness F_{max}.

$(4, -100, 546, 0, 0, 0)$ with fitness value $F_1 = F_2 = F_{max,b} = 0.8519$. It can be seen that they are near to the true value shown as as Eq. (4.5).

In (a3), because the environment with the illumination 30[lx] is very dark, there are more black points in the three balls in the images than that in (b3). Therefore, the $F_{max,b} > F_{max,a}$. From (b) to (d), the illumination is gradually increasing. And there are more and more white points in the images. They influence the fitness calculation. Therefore, $F_{max,b} > F_{max,c} > F_{max,d}$.

By the fitness distribution experiment, it is verified that the fitness function Eq. (3.22) can transform the target position and orientation estimation problems into optimization problems. And it has the robustness against the illumination changing.

4.2.2 Content of arc swing motion experiment

In order to study the tracking performance of the system in orientation under different light conditions, arc swing motion tracking experiment have been conducted. In this light changing experiment, the illumination condition is divided into 80[lx], 500[lx], 900[lx], and 2200[lx] four cases. As shown in Fig. 4.12 and Eq. (6.2) with the same set of lateral tracking experiments the desired value of distance between object $\vec{\Sigma}_M$ and end-effector $\vec{\Sigma}_E$ is ${}^E x_M = 0$, ${}^E y_M = -100$[mm], ${}^E z_M = 545$[mm]. And the relative orientation between object and end-effector is $\boldsymbol{\varepsilon} = \mathbf{0}$, i.e. in the process of tracking, always keep the x-y plane in $\vec{\Sigma}_E$ parallel to the x-y plane in $\vec{\Sigma}_M$. Σ_B is the coordinate system of the turntable. And the turntable takes $\pm 20^\circ$ reciprocal uniform rotation movement around y-axis of Σ_B. Equation (4.7) shows the periodic function with period $T = 4 \times 4.44 = 17.76$[s]. Therefore, $\theta(t) = \theta(t + 17.76)$. In the first period, $\theta(t)$ is shown as follow.

$$\theta(t) = \begin{cases} -4.5t & t \in [0, 4.44)s \quad (4.7a) \\ 4.5t - 40 & t \in [4.44, 13.32)s \quad (4.7b) \\ -4.5t + 80 & t \in [13.32, 17.76)s \quad (4.7c) \end{cases}$$

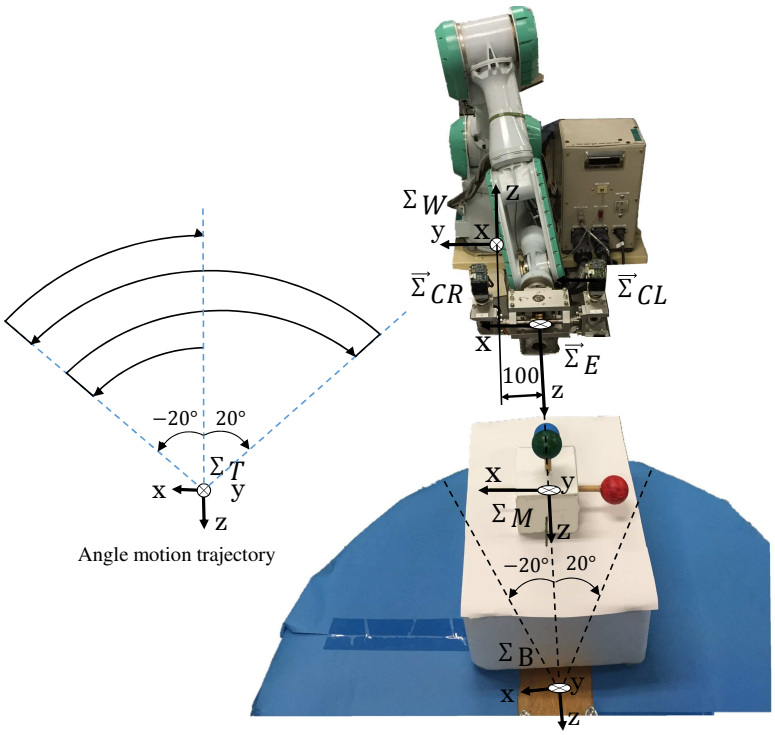


Fig. 4.11: The initial state of each coordinate system and the angle motion trajectory of the turntable.

At this speed the 3D marker will take 80[s] to rotate one cycle, that means the angular velocity $\omega = \pm 2\pi/T = \pm 2\pi/80 \approx \pm 0.079$[rad/s]$\approx \pm 4.5$[°/s].

4.2.3 Experimental Result

As shown in Fig. 4.14, (a) is fitness value during the tracking process calculated by Eq. (3.22). It shows at each time the degree of matching between the object and the best individual evolved from GA. As described in Section 3.5, the maximum of fitness is $F_{max} = 1.67$. The fitness can be affected by many factors, e.g., the quality of the captured images, the motion of manipulator or the changing of light. The fitness in Fig. 4.14 (a) takes dramatic fluctuations. It can be seen that changes in light illumination affect the object recognition.

In Fig. 4.14 (b), (c) and (d), the dashed lines represent the orientation ε_M of real target Σ_M.

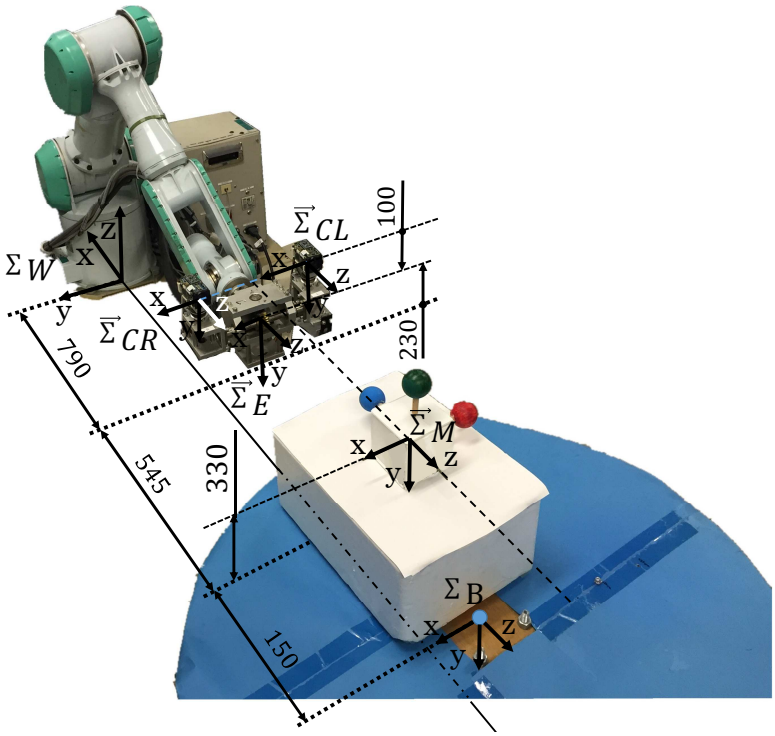


Fig. 4.12: The initial state of object and visual servoing system. The relative position relationship between different coordinate systems is marked. Unit:[mm].

Orientation tracking result of the detected object $\vec{\Sigma}_{\widehat{M}}$ and end-effector $\vec{\Sigma}_E$ are shown as the solid line $\boldsymbol{\varepsilon}_{\widehat{M}}$ and dotted line $\boldsymbol{\varepsilon}_E$ respectively. As shown in Fig. 4.11, the target takes the arc swing motion on a turntable in a horizontal plane. Therefore, as shown in Fig. 4.14, the orientation of the target object ε_{M1} and ε_{M3} are all 0.

As shown in Fig. 4.14 (b), it can be seen that the recognition result $\varepsilon_{\widehat{M}1}$ is near to 0. However, the tracking result of hand had always a small deviation about 0.05. Similar to (b), in (d) it can be seen that the detection result $\varepsilon_{\widehat{M}3}$ is near to the true value ε_{M3}. And the motion of manipulator is also near to the target.

As shown in Fig. 4.14 (c), the true value of the object is ε_{M2}, i.e., the triangular wave in dashed line. It can been seen that the detected orientation $\varepsilon_{\widehat{M}2}$ can continually track the object and is closer to the truth value than ε_{E2} of the end-effector. (e) shows the tracking error of hand $\Delta\varepsilon_{EM2}$ and detection error of stereo vision $\Delta\varepsilon_{M\widehat{M}2}$. At about 32[s], there is a transient error in orientation detection. Because the time is short and the system responds slowly, this error does not have much effect on the tracking motion.

Through the results, it can be confirmed that although the fitness is changed a lot because of the illumination changing, the recognition and the motion of manipulator was not influenced so much. It shows that the system can overcome some illumination change and track the orientation of the target object continually.

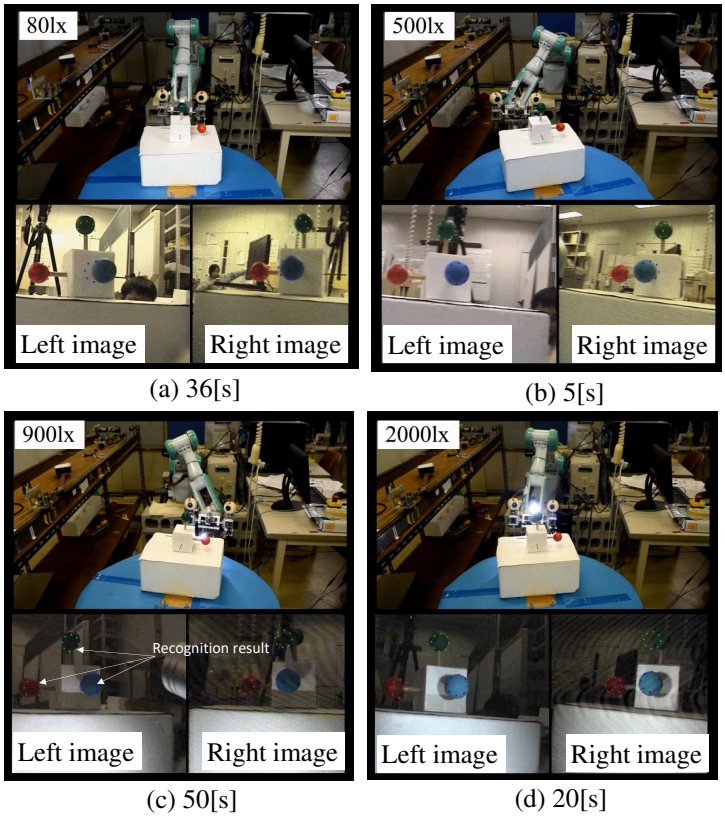
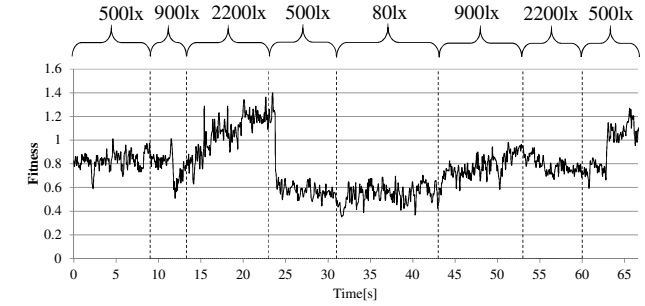
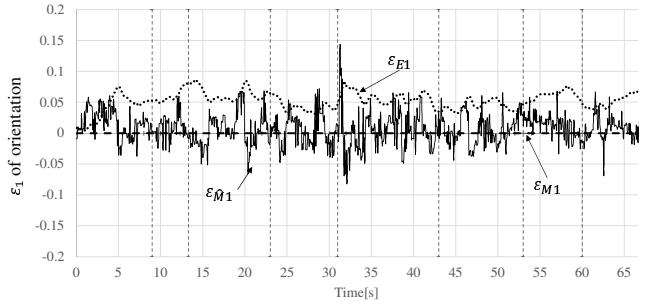


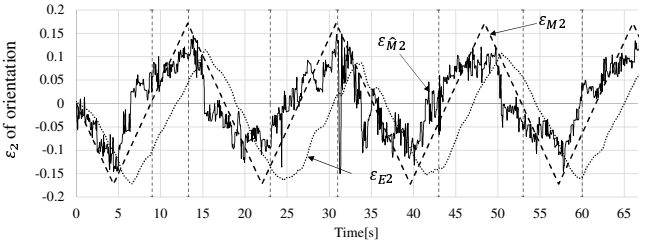
Fig. 4.13: The experimental status and dual-eye images under different illuminations. The upper left corner of each picture is marked with the current illumination. And the subtitle of each picture is the photography time corresponding to the time in Fig.4.14



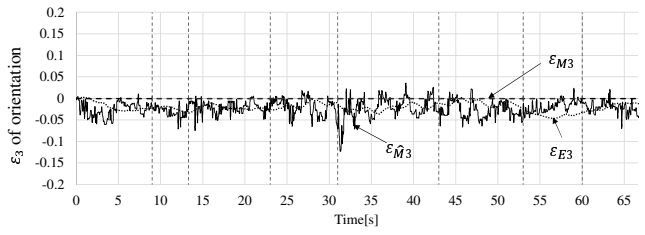
(a)Fitness value change in the tracking process



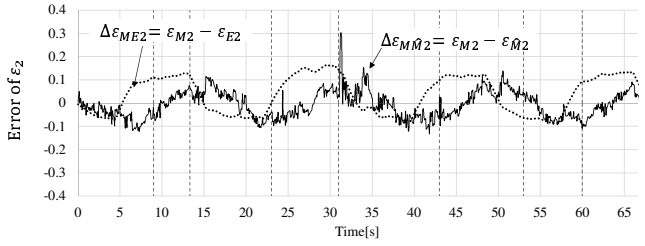
(b) ε_1 of Orientation of End Effector $^{\Sigma}E$, 3D Marker $^{\Sigma}M$ and Detected Object $^{\Sigma}\hat{M}$



(c) ε_2 of Orientation of End Effector $^{\Sigma}E$, 3D Marker $^{\Sigma}M$ and Detected Object $^{\Sigma}\hat{M}$



(d) ε_3 of Orientation of End Effector $^{\Sigma}E$, 3D Marker $^{\Sigma}M$ and Detected Object $^{\Sigma}\hat{M}$



(e) Tracking Error in ε_2 of End Effector $^{\Sigma}E$ and Detected Object $^{\Sigma}\hat{M}$

Fig. 4.14: Tracking results under different illumination. $\boldsymbol{\varepsilon}_M = [\varepsilon_{M1}, \varepsilon_{M2}, \varepsilon_{M3}]^T$ is the actual orientation of the target object. $\boldsymbol{\varepsilon}_E = [\varepsilon_{E1}, \varepsilon_{E2}, \varepsilon_{E3}]^T$ is the orientation of the end-effector. And detected orientation is $\boldsymbol{\varepsilon}_{\widehat{M}} = [\varepsilon_{\widehat{M}1}, \varepsilon_{\widehat{M}2}, \varepsilon_{\widehat{M}3}]^T$. In ε_2 direction, tracking error of end-effector (hand) is $\Delta\varepsilon_{ME2}$ and detection error is $\Delta\varepsilon_{M\widehat{M}2}$.

Chapter 5

Photo-Model-Based Recognition

This chapter discusses the methodology of the proposed photo-model-based recognition method. Firstly, the geometry of a stereo vision system and symbol definition will be described to make it easy to understand the recognition method. Secondly, the generation and matching of a photo-model are introduced. Then, an evaluation function is designed to convert the object recognition problem into an optimization problem. In the end, a genetic algorithm is chosen as a solution to the optimization problem to ensure that the recognition method can detect an object in real-time. Although the eye-vergence vision has better tracking ability than the fixed camera vision, it is more complex than the later one. In this study, as the initial stage of the development, the real-time photo-model-based 6DOF pose estimation method is developed based on the fixed camera system. In the future, it will be used in the eye-vergence visual system.

5.1 Stereo vision geometry and definition of each symbol

Figure 1.2 shows the developed photo-model-based visual servoing system, VS-robot. Each coordinate system is as follows:

- Σ_W: world coordinate system,
- Σ_H: end-effector (hand) coordinate system,

- Σ_M: object coordinate system.

The world coordinate system Σ_W is fixed on the floor. The homogeneous transformation matrices from Σ_W to Σ_H and Σ_M are ${}^W\boldsymbol{T}_H$ and ${}^W\boldsymbol{T}_M$ respectively. ${}^W\boldsymbol{T}_H$ can be calculated with the joint angles of VS-robot. Based on Σ_W, the pose of end-effector ${}^W\boldsymbol{\phi}_H$ and target ${}^W\boldsymbol{\phi}_M$ are represented as

$$ {}^W\boldsymbol{\phi}_k = [{}^W\boldsymbol{r}_k^{\mathrm{T}}, {}^W\boldsymbol{\varepsilon}_k^{\mathrm{T}}]^{\mathrm{T}} = [{}^Wx_k, {}^Wy_k, {}^Wz_k, {}^W\varepsilon_{1k}, {}^W\varepsilon_{2k}, {}^W\varepsilon_{3k}]^{\mathrm{T}}, \quad (k = H, M). \tag{5.1} $$

Figure 5.1 shows a perspective projection of the stereo vision system.

- Σ_{CL}, Σ_{CR}: left and right camera coordinate systems,
- Σ_{IL}, Σ_{IR}: left and right image coordinate systems,
- Σ_{Mj}: j-th model coordinate system,
- $\Sigma_{\widehat{M}}$: the coordinate system of RM-GA searching result that is not shown in Fig. 5.1.

The position vectors of an arbitrary i-th point of the j-th 3D model coordinate Σ_{Mj} based on different coordinate systems are as follows:

- ${}^W\boldsymbol{r}_i^j$: 3D position of an arbitrary i-th point on j-th 3D model based on Σ_W,
- ${}^M\boldsymbol{r}_i^j$: 3D position of an arbitrary i-th point on j-th 3D model in Σ_{Mj},
- ${}^{CR}\boldsymbol{r}_i^j$ and ${}^{CL}\boldsymbol{r}_i^j$: 3D position of an arbitrary i-th point on j-th 3D model based on Σ_{CR} and Σ_{CL},
- ${}^{IL}\boldsymbol{r}_i^j$ and ${}^{IR}\boldsymbol{r}_i^j$: 2D projected position on Σ_{IL} and Σ_{IR} of an arbitrary i-th point on j-th 3D model.

The pose of the j-th 3D model, including three position variables and three orientation variables in quaternion based on Σ_H, is represented as

$$ {}^H\boldsymbol{\phi}_M^j = [{}^Hx_M^j, {}^Hy_M^j, {}^Hz_M^j, {}^H\varepsilon_{1M}^j, {}^H\varepsilon_{2M}^j, {}^H\varepsilon_{3M}^j]^{\mathrm{T}}. \tag{5.2} $$

For simplicity, the ${}^{H}\boldsymbol{\phi}_M^j$ is written as $\boldsymbol{\phi}_M^j$ hereafter.

As the searching result of RM-GA explained in Section 5.4, the detected result is defined as,

$$ {}^{H}\boldsymbol{\phi}_{\widehat{M}} = [{}^{H}x_{\widehat{M}}, {}^{H}y_{\widehat{M}}, {}^{H}z_{\widehat{M}}, {}^{H}\varepsilon_{\widehat{1M}}, {}^{H}\varepsilon_{\widehat{2M}}, {}^{H}\varepsilon_{\widehat{3M}}]^{\mathrm{T}}. \tag{5.3} $$

Based on Σ_W, the homogenous transformation matrix ${}^{W}\boldsymbol{T}_{\widehat{M}}$ is calculated by Eq. (5.5). The j-th model's pose based on Σ_H and Σ_W are represented as Eqs. (5.2) and (5.6), respectively, including three position variables and three orientation variables in quaternion. The pose ${}^{W}\boldsymbol{\phi}_M^j$ is derived from ${}^{W}\boldsymbol{T}_{Mj}$ that is calculated in Eq. (5.4) [93].

$$ {}^{W}\boldsymbol{T}_{Mj} = {}^{W}\boldsymbol{T}_{H}\,{}^{H}\boldsymbol{T}_{Mj}(\boldsymbol{\phi}_M^j) \tag{5.4} $$

$$ {}^{W}\boldsymbol{T}_{\widehat{M}} = {}^{W}\boldsymbol{T}_{H}\,{}^{H}\boldsymbol{T}_{\widehat{M}}({}^{H}\boldsymbol{\phi}_{\widehat{M}}) \tag{5.5} $$

$$ {}^{W}\boldsymbol{\phi}_M^j = [({}^{W}\boldsymbol{r}_M^j)^{\mathrm{T}}, ({}^{W}\boldsymbol{\varepsilon}_M^j)^{\mathrm{T}}]^{\mathrm{T}} = [{}^{W}x_M^j, {}^{W}y_M^j, {}^{W}z_M^j, {}^{W}\varepsilon_{1M}^j, {}^{W}\varepsilon_{2M}^j, {}^{W}\varepsilon_{3M}^j]^{\mathrm{T}} \tag{5.6} $$

About stereo vision, position ${}^{CL}\boldsymbol{r}_i^j$ can be calculated by using Eq. (5.7),

$$ {}^{CL}\boldsymbol{r}_i^j = {}^{CL}\boldsymbol{T}_M(\boldsymbol{\phi}_M^j, \boldsymbol{q})\ {}^{M}\boldsymbol{r}_i^j, \tag{5.7} $$

where ${}^{M}\boldsymbol{r}_i^j$ is predetermined as a fixed vector since Σ_{Mj} is fixed on the j-th model. Similarly, ${}^{CR}\boldsymbol{r}_i^j$ is calculated by using ${}^{CR}\boldsymbol{T}_M(\boldsymbol{\phi}_M^j, \boldsymbol{q})$. Since $\boldsymbol{q}$ can be measured by robot's joint sensors, it could be thought to have been known, then $\boldsymbol{q}$ is omitted hereafter.

The proposed system is an eye-in-hand system with dual-eye stereo-vison cameras. Camera model is pinhole model. Eq. (5.8) represents the projective transformation matrix $\boldsymbol{P}_k$,

$$ \boldsymbol{P}_k = \frac{1}{{}^{k}z_i} \begin{bmatrix} f/\eta_x & 0 & {}^{I}x_0 & 0 \\ 0 & f/\eta_y & {}^{I}y_0 & 0 \end{bmatrix}, \tag{5.8} $$

where,

- k = CL, CR,
- ${}^{k}z_i$: z-axis position of the i-th point in the camera sight direction in Σ_{CR} and Σ_{CL},
- f: focal length,
- η_x, η_y: [mm/pixel] in x-axis, and y-axis,
- ${}^{I}x_0, {}^{I}y_0$: [pixel] offset of origin of Σ_I.

The 2D position vector of the i-th point in the left camera image coordinates ${}^{IL}\boldsymbol{r}_i^j$ can be described by using projective transformation matrix $\boldsymbol{P}_{CL}$ as,

$$ {}^{IL}\boldsymbol{r}_i^j = \boldsymbol{P}_{CL}{}^{CL}\boldsymbol{r}_i^j = \boldsymbol{P}_{CL}{}^{CL}\boldsymbol{T}_M(\boldsymbol{\phi}_M^j){}^{M}\boldsymbol{r}_i^j. \tag{5.9} $$

Then, ${}^{IL}\boldsymbol{r}_i^j$ can be conceptually described by function $\boldsymbol{f}_L$ as,

$$ {}^{IL}\boldsymbol{r}_i^j(\boldsymbol{\phi}_M^j) = \boldsymbol{f}_L(\boldsymbol{\phi}_M^j, {}^{M}\boldsymbol{r}_i^j). \tag{5.10} $$

Like the description of ${}^{IL}\boldsymbol{r}_i^j$, ${}^{IR}\boldsymbol{r}_i^j$ can also be calculated as the same manner.

In the visual servoing experiments of Section 6.2.1 and Section 6.2.2, Σ_M will move along predetermined trajectories. Therefore, ${}^{W}\boldsymbol{T}_M$ is a known time-varying matrix. The goal of the visual servoing experiment is to control the end-effector to maintain the relative pose with the target object as

$$ {}^{M}\boldsymbol{T}_{Hd} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -500[\mathrm{mm}] \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.11} $$

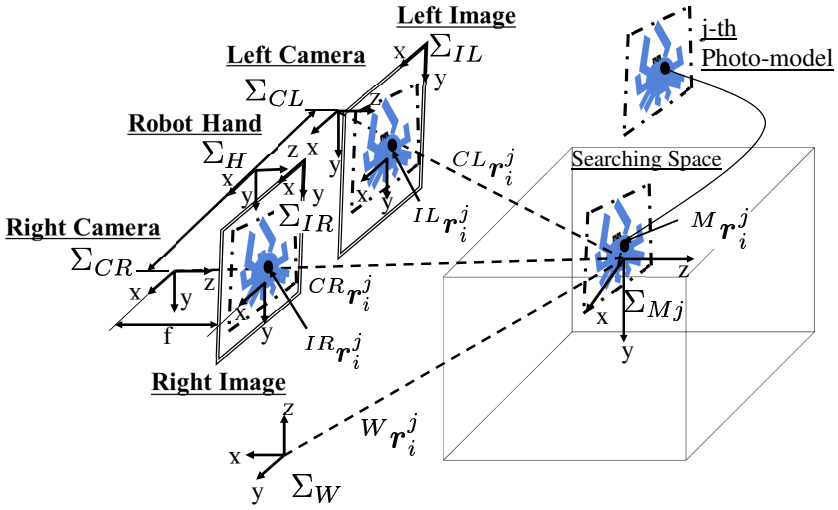


Fig. 5.1: Perspective projection of stereo vision system. In the searching space, a j-th 3D solid model is represented by the picture of crab, which is defined by j-th model coordinate system Σ_{Mj}. The distance between Σ_{CL} and Σ_{CR}, i.e. baseline, is 323[mm].

where Σ_{Hd} is the desired pose of Σ_H. Based on Σ_W, the pose of Σ_{Hd},

$$ {}^{W}\boldsymbol{\phi}_{Hd} = [{}^{W}\boldsymbol{r}_{Hd}^{\mathrm{T}}, {}^{W}\boldsymbol{\varepsilon}_{Hd}^{\mathrm{T}}]^{\mathrm{T}} = [{}^{W}x_{Hd}, {}^{W}y_{Hd}, {}^{W}z_{Hd}, {}^{W}\varepsilon_{1Hd}, {}^{W}\varepsilon_{2Hd}, {}^{W}\varepsilon_{3Hd}]^{\mathrm{T}}, \tag{5.12} $$

is derived from ${}^{W}\boldsymbol{T}_{Hd}$ that is calculated by Eq. (5.13) [93].

$$ {}^{W}\boldsymbol{T}_{Hd} = {}^{W}\boldsymbol{T}_{M}\,{}^{M}\boldsymbol{T}_{Hd}. \tag{5.13} $$

Figure 1.2 shows the initial poses of Σ_M and Σ_H at $t = 0$[s]. They are also defined as Σ_{M0} and Σ_{H0} respectively. The directions of Σ_{M0} and Σ_{H0} are different from Σ_W. Their orientations in quaternion are ${}^{W}\boldsymbol{\varepsilon}_{M0} = {}^{W}\boldsymbol{\varepsilon}_{H0} = [-0.5, -0.5, 0.5]^{\mathrm{T}}$ [34]. It is difficult to directly imagine. Therefore, when we talk about the experiment results, the orientations of Σ_M and Σ_H are calculated as Eq. (5.14), i.e., relative orientations to the initial status.

$$ \boldsymbol{\varepsilon}_k = {}^{W}\boldsymbol{\varepsilon}_k - {}^{W}\boldsymbol{\varepsilon}_{k0}, (k = M, H) \tag{5.14} $$

			
C01 Seahorse 13 × 4.5 × 2.7[cm]	C02 Coelacanth 7.0 × 14.5 × 6.0[cm]	C03 Moray eel 3.0 × 14.2 × 2.3[cm]	C04 Dolphin 8 × 6 × 4.5[cm]
			
C05 Bigfin reef squid 21 × 8.25 × 4.5[cm]	C06 Jellyfish 9 × 9 × 11[cm]	C07 Leatherback sea turtle 10.5 × 13.2 × 3[cm]	C08 Octopus 14.3 × 12.5 × 3.5[cm]
			
C09 Anemonefish 12 × 3.6 × 5[cm]	C10 Mobula 10 × 8 × 2[cm]	C11 Bluespotted ribbontail ray 8.5 × 15.0 × 1.5[cm]	C12 Crab 17.5 × 14 × 4[cm]

Fig. 5.2: Twelve marine biological creature models. The code name is from C01 to C12. The second line of each frame shows the English name. And the last line shows the size of each 3D toy (unit: [cm]).

Similarly, the desired orientation of the end-effector is

$$\varepsilon_{Hd} = {}^{W}\varepsilon_{Hd} - {}^{W}\varepsilon_{H0}, \tag{5.15}$$

and ε_{Hd} means ε_M because the control goal is shown as Eq. (5.11). Because the orientation of the target object does not change in the position visual servoing experiment, as shown in Fig. 6.4 (1.d) $\sim$ (1.f),

$$\varepsilon_{Hd} = \varepsilon_M = \mathbf{0}. \tag{5.16}$$

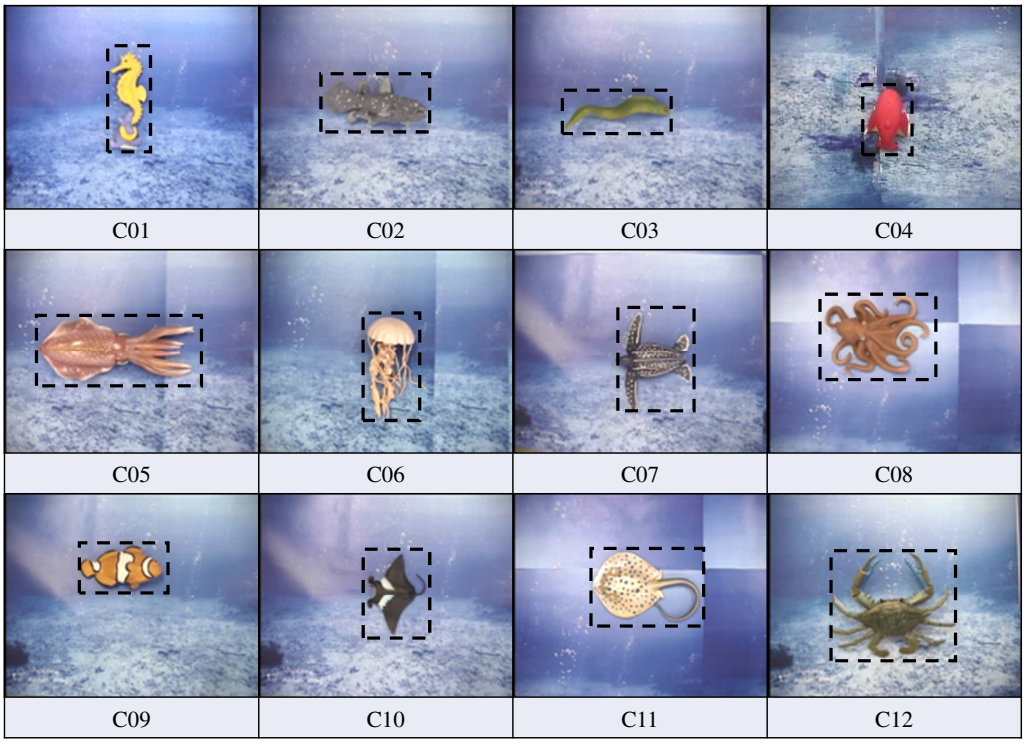


Fig. 5.3: Twelve pictures of marine biological creature models are shown with blue sea background corresponding to Fig. 5.2. The code name is from C01 to C12. The size of each picture is 640×480 [pixel]. Each dashed line rectangle indicates a photo-model.

5.2 Photo-model generation

There are two main portions of the proposed pose estimation method. The first portion is 2D model generation and the latter is relative pose estimation using the generated 2D model. This subsection is a description of the first portion before an explanation of a matching method. As shown in Fig. 5.2, 12 different sea creature toys are prepared as 3D target objects whose code names are from C01 to C12. The table includes the English name and the size of each 3D toy. Figure 5.3 shows photo-models with blue sea background. The size of each picture is 640×480 [pixel]. Each dashed line rectangle indicates a photo-model used in pose estimation of 3D toy targets. The photo-model is only part of a picture including a target shape as shown by the rectangles in Fig. 5.3.

The model generation process is represented as Fig. 5.4. It should be noted that the photo-model is only part of a picture including target shape. Firstly, a background image is captured by the camera and the average hue value of the background image is calculated as shown in Fig. 5.4 (a). Then, the solid crab target object is put on the background. A 640×480 [pixel] picture is captured at a distance of 400[mm] from the object as shown in (b). In (c), a photo-model composed of dots with color information of hue is set as S_{in}. Finally, the outside space S_{out} of the model is generated by enveloping S_{in} as shown in (d).

5.3 3D photo-model-based matching

Figure 5.5 shows a generated photo-model placed in the 3D searching space with assumed pose of $\boldsymbol{\phi}_M^j$ (sub figure on the top of Fig. 5.5) and the left and right 2D searching models that are projected from photo model with the pose being assumed to be $\boldsymbol{\phi}_M^j$ (sub figures on the left and right bottom of Fig. 5.5) respectively. In Fig. 5.5, a generated 2D photo-model is projected from the 3D space onto the left and right 2D searching planes. The sub figures on the top of Fig. 5.5 shows a generated 3D solid photo model $S(\boldsymbol{\phi}_M^j)$ composed of $S_{in}(\boldsymbol{\phi}_M^j)$ (inner dotted points) and the outside space enveloping $S_{in}(\boldsymbol{\phi}_M^j)$ denoted as outer dotted line $S_{out}(\boldsymbol{\phi}_M^j)$. The sub figures on

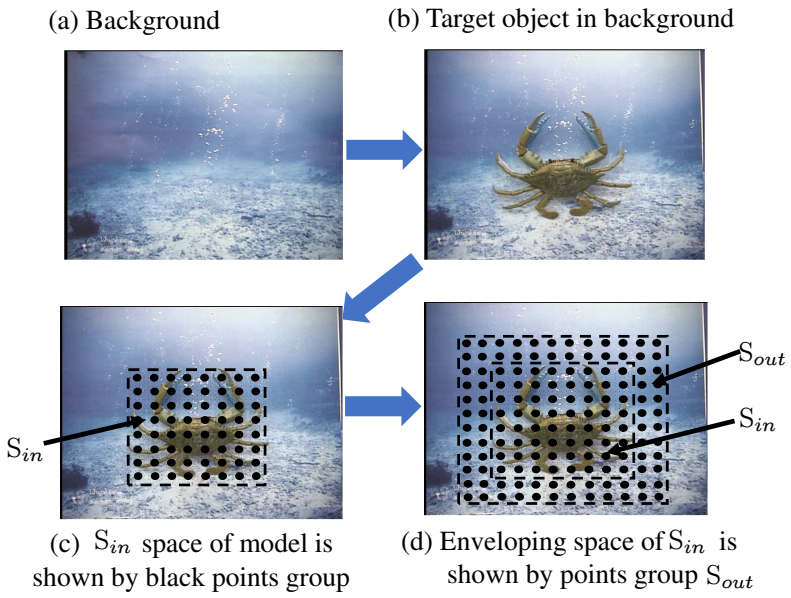


Fig. 5.4: (a) shows a photograph of background image, (b) shows a photograph of the target object, the crab, in background, (c) represents a photograph of surface space model S_{in} by inner points group and (d) represents an outer points group of outside space of model S_{out} that enveloping S_{in}.

the left/right bottom of Fig. 5.5 show the left/right projected 2D searching models $S_L(\boldsymbol{\phi}_M^j)$ and $S_R(\boldsymbol{\phi}_M^j)$ respectively. Both $S_L(\boldsymbol{\phi}_M^j)$ and $S_R(\boldsymbol{\phi}_M^j)$ consist of inner and outer portions $S_{L,in}(\boldsymbol{\phi}_M^j)$, $S_{L,out}(\boldsymbol{\phi}_M^j)$ and $S_{R,in}(\boldsymbol{\phi}_M^j)$, $S_{R,out}(\boldsymbol{\phi}_M^j)$.

The evaluation of the correlation between the projected model and the images including real target object that are input from the dual-eye cameras is defined as a fitness function.

5.3.1 Definition of the fitness function

An overlap degree, that means correlation degree, between a projected model and the target in images captured by the dual-eye cameras is used as a fitness [100]. The highest fitness value represents the best pose of the model $\widehat{\boldsymbol{\phi}}$ among $\boldsymbol{\phi}_M^j$ that coincides with the crab's pose in 3D space as depicted at top of Fig. 5.5.

A model is composed of some sampling points. The number of them is "N." After forward projection, as shown in Fig. 5.1, each point coordinate in left image Σ_{IL} is ${}^{IL}\boldsymbol{r}_i^j$. And evaluation

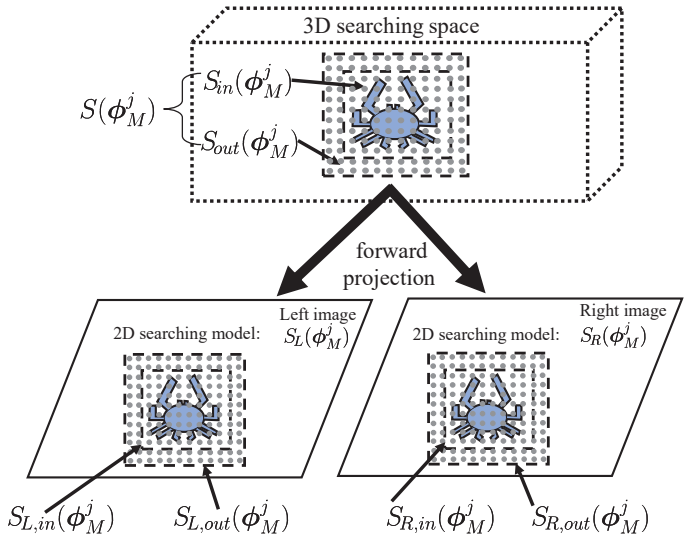


Fig. 5.5: A photo model $S(\phi_M^j)$ in the 3D searching space on the top of this figure is a 2D model but it has 3D pose information ϕ_M^j. The left and right 2D searching models represented as $S_L(\phi_M^j)$ and $S_R(\phi_M^j)$ on the left/right bottom, are calculated by forward projection from the 2D photo-model $S(\phi_M^j)$.

value of each point ${}^{IL}\boldsymbol{r}_i^j$ in inner portion of the model (${}^{IL}\boldsymbol{r}_i^j \in S_{R,in}(\boldsymbol{\phi}_M^j)$) is $p_{L,in}({}^{IL}\boldsymbol{r}_i^j)$ calculated by Eq. (5.17). The one of outer portion (${}^{IL}\boldsymbol{r}_i^j \in S_{L,out}(\boldsymbol{\phi}_M^j)$) is $p_{L,out}({}^{IL}\boldsymbol{r}_i^j)$ calculated by Eq. (5.18)

$$p_{L,in}({}^{IL}\boldsymbol{r}_i^j) = \begin{cases} 2, & \text{if}(|H_{IL}({}^{IL}\boldsymbol{r}_i^j) - H_{ML}({}^{IL}\boldsymbol{r}_i^j)| \leq 30); \\ -0.005, & \text{else if}(|\bar{H}_B - H_{IL}({}^{IL}\boldsymbol{r}_i^j)| \leq 30); \\ 0, & \text{otherwise;} \end{cases} \tag{5.17}$$

$$p_{L,out}({}^{IL}\boldsymbol{r}_i^j) = \begin{cases} 0.1, & \text{if}(|\bar{H}_B - H_{IL}({}^{IL}\boldsymbol{r}_i^j)| \leq 20); \\ -0.5, & \text{otherwise;} \end{cases} \tag{5.18}$$

where

- $H_{IL}({}^{IL}\boldsymbol{r}_i^j)$: the hue value of the left camera image at the point ${}^{IL}\boldsymbol{r}_i^j$ (i-th point in j-th photo model, lying in $S_{L,in}$),
- $H_{ML}({}^{IL}\boldsymbol{r}_i^j)$: the hue value of photo model at the point ${}^{IL}\boldsymbol{r}_i^j$ (i-th point in $S_{L,in}$),
- $\bar{H}_B$: the average hue value of the background image, i.e., Fig. 5.4 (a).

The $p_{R,in}({}^{IR}\boldsymbol{r}_i^j)$ and $p_{R,out}({}^{IR}\boldsymbol{r}_i^j)$ are defined as the same above manner. The fitness $F(\boldsymbol{\phi}_M^j)$ of a model is calculated as Eq. (5.19), and its abbreviated form is Eq. (5.20),

$$F(\boldsymbol{\phi}_M^j) = \Big[\Big(\sum_{\substack{{}^{IR}\boldsymbol{r}_i^j \in \\ S_{R,in}(\boldsymbol{\phi}_M^j)}} p_{R,in}({}^{IR}\boldsymbol{r}_i^j) + \sum_{\substack{{}^{IR}\boldsymbol{r}_i^j \in \\ S_{R,out}(\boldsymbol{\phi}_M^j)}} p_{R,out}({}^{IR}\boldsymbol{r}_i^j) \Big) / N + \Big(\sum_{\substack{{}^{IL}\boldsymbol{r}_i^j \in \\ S_{L,in}(\boldsymbol{\phi}_M^j)}} p_{L,in}({}^{IL}\boldsymbol{r}_i^j) + \sum_{\substack{{}^{IL}\boldsymbol{r}_i^j \in \\ S_{L,out}(\boldsymbol{\phi}_M^j)}} p_{L,out}({}^{IL}\boldsymbol{r}_i^j) \Big) / N \Big] / 2 \tag{5.19}$$

$$= [F_R(\boldsymbol{\phi}_M^j) + F_L(\boldsymbol{\phi}_M^j)]/2. \tag{5.20}$$

The fixed values in Eqs. (5.17) and (5.18) have been tuned experimentally to provide a peak in the fitness value distribution at the true pose. Figure 5.7 (a) shows j-th model, the evaluation points of hue value, $\cdots {}^{IL}\boldsymbol{r}_{i-1}^j$, ${}^{IL}\boldsymbol{r}_i^j$, ${}^{IL}\boldsymbol{r}_{i+1}^j \cdots$, are indicated by white dots in inside area $S_{L,in}$, and those in outside strip $S_{L,out}$. Figure 5.7 (b) shows another situation that the overlapping

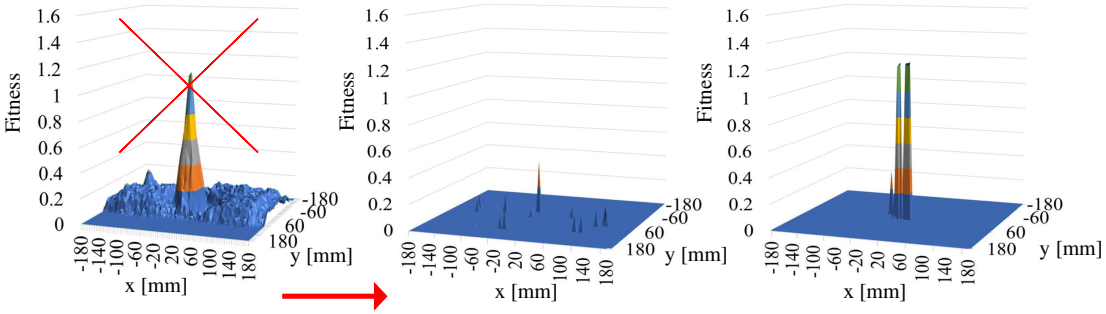
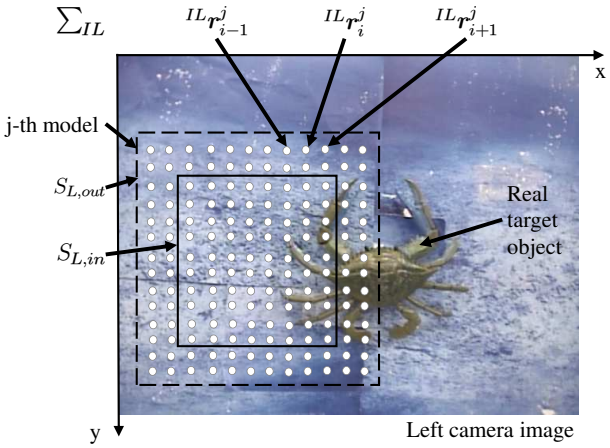
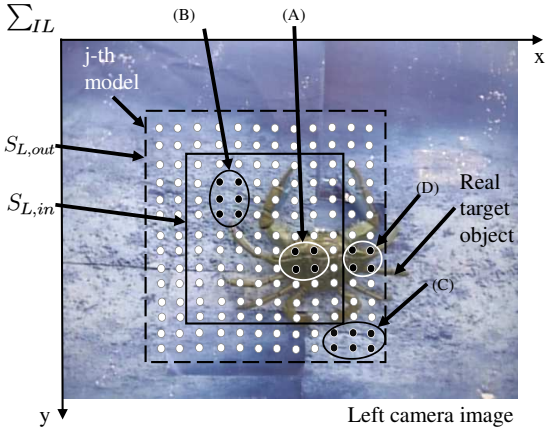


Fig. 5.6: We have proposed Real-time Multi-step Genetic Algorithm (RM-GA) for searching the pose of target object in real-time.



(a) Evaluation position $^{IL}\boldsymbol{r}_i^j$, that is i-th point of j-th model, which is projected on left image whose pose $\boldsymbol{\phi}_M^j$ is given by evolutionary process of GA.



(b) Classification of evaluation points (A)∼(D) on the photo model is explained. (A) represents points that satisfy the first case of Eq. (5.17), $|H_{IL}(^{IL}\boldsymbol{r}_i^j) - H_{ML}(^{IL}\boldsymbol{r}_i^j)| \leq 30$, representing that inner model $S_{L,in}$ overlaps with the real target. (B) does $|\bar{H}_B - H_{IL}(^{IL}\boldsymbol{r}_i^j)| \leq 30$, representing that inner model $S_{L,in}$ overlaps with background. (C) does $|\bar{H}_B - H_{IL}(^{IL}\boldsymbol{r}_i^j)| \leq 20$, meaning that the outer model $S_{L,out}$ overlaps with background, and (D) shows $S_{L,out}$ overlaps with the real target.

Fig. 5.7: Calculation of the matched degree of each point in model space ($S_{L,in}$ and $S_{L,out}$).

area of real crab and the model increased than the one depicted in (a). The hue value of the left camera input image at the point ${}^{IL}\boldsymbol{r}_i^j$ is represented by $H_{IL}({}^{IL}\boldsymbol{r}_i^j)$. The i-th point of j-th model in $S_{L,in}$ and $S_{L,out}$ and the hue value of the same point ${}^{IL}\boldsymbol{r}_i^j$ on the model is defined as $H_{ML}({}^{IL}\boldsymbol{r}_i^j)$. The average hue value of background calculated from Fig. 5.4 (a) is defined as $\bar{H}_B$.

In Eq. (5.17), if the hue value of each point on 3D target in left images, $H_{IL}({}^{IL}\boldsymbol{r}_i^j)$, which lies inside the surface model frame $S_{L,in}$, and the hue value of corresponding same point in a model, $H_{ML}({}^{IL}\boldsymbol{r}_i^j)$, have similar values with a tolerance less than 30, that is $|H_{IL}({}^{IL}\boldsymbol{r}_i^j) - H_{ML}({}^{IL}\boldsymbol{r}_i^j)| \leq 30$ then this means that model's hue value and input target crab's hue value have close hue distance at the same checking point of ${}^{IL}\boldsymbol{r}_i^j$. This represents photo model overlaps to the real crab projected in left camera image in S_{in}, which is represented by dots designated by (A) in Fig. 5.7 (b). In this case the fitness value would be increased with the voting value of "+2." The fitness value will decrease with the value of "-0.005" for every point ${}^{IL}\boldsymbol{r}_i^j$ in S_{in} by the condition, $|\bar{H}_B - H_{IL}({}^{IL}\boldsymbol{r}_i^j)| \leq 30$ in Eq. (5.17), when model's crab area overlaps with blue background. This represents that the model does not overlap precisely the target in the input image, which are represented by (B) in Fig. 5.7 (b). In this case, "-0.005" is given as a penalty to decrease F. Otherwise, the evaluation value will be "0."

Similarly, in Eq. (5.18), if the hue value of each point in the left camera image lying in $S_{L,out}$ has similar value to the average hue value of background $\bar{H}_B$ calculated from Fig. 5.4 (a) with the tolerance of 20, the fitness value will be increased with the value of "+0.1." This means $S_{L,out}$ strip area surrounding $S_{L,in}$ overlaps the background, expressing the model and the crab overlap rather correctly as (C) in Fig. 5.7 (b). Since this situation means that the model's position and orientation matches to the real crab, plus points "0.1" is given to the function $p_{L,out}$, which is described in Eq. (5.18). Otherwise, the fitness value will decrease with the penalty value of "-0.5." This represents points on $S_{L,out}$ overlaps with the real crab as (D) in Fig. 5.7 (b).

5.4 Improved Real-Time Multi-Step Genetic Algorithm (RM-GA)

The main problem of identifying the pose of the object can be converted into an optimization problem if the fitness function has been designed to give the maximum value only in the case that the model whose pose coincides with the target object in the 3D space. Several optimization methods can search the maximum value of the evaluation function. For real-time recognition in dynamic images input with frame rate 30[fps], we have proposed a Real-time Multi-step Genetic Algorithm (RM-GA) [101],[102]. RM-GA evaluation process is applied to find the maximum value as an optimal solution because of its simplicity and effectiveness. The 20 individuals of RM-GA are used in this experiment, where the chromosome of an individual consists of 72[bit] with six variables. Each variable is coded by 12[bit] as shown in Eq. (5.21), the first three variables of a model (${}^{H}x_{M}^{j}, {}^{H}y_{M}^{j}, {}^{H}z_{M}^{j}$) represents the position in 3D space and the last three variables (${}^{H}\varepsilon_{1M}^{j}, {}^{H}\varepsilon_{2M}^{j}, {}^{H}\varepsilon_{3M}^{j}$) represents the orientation. The genes of RM-GA representing possible pose solution is defined as below:

$$\overbrace{\underbrace{01\cdots01}_{12[\text{bit}]}}^{{}^{H}x_{M}^{j}}\overbrace{\underbrace{00\cdots01}_{12[\text{bit}]}}^{{}^{H}y_{M}^{j}}\overbrace{\underbrace{11\cdots01}_{12[\text{bit}]}}^{{}^{H}z_{M}^{j}}\overbrace{\underbrace{01\cdots01}_{12[\text{bit}]}}^{{}^{H}\varepsilon_{1M}^{j}}\overbrace{\underbrace{01\cdots11}_{12[\text{bit}]}}^{{}^{H}\varepsilon_{2M}^{j}}\overbrace{\underbrace{01\cdots10}_{12[\text{bit}]}}^{{}^{H}\varepsilon_{3M}^{j}}. \tag{5.21}$$

As the searching result of RM-GA, the output best individual is defined as,

$${}^{H}\boldsymbol{\phi}_{\widehat{M}} = [{}^{H}x_{\widehat{M}}, {}^{H}y_{\widehat{M}}, {}^{H}z_{\widehat{M}}, {}^{H}\varepsilon_{\widehat{1M}}, {}^{H}\varepsilon_{\widehat{2M}}, {}^{H}\varepsilon_{\widehat{3M}}]^{\mathrm{T}}. \tag{5.22}$$

Figure 5.8 (a) shows the process flow in the RM-GA in which 3D models converge into the real 3D solid target object. In Fig. 5.8 (a), a target object is a crab, and each 3D model is depicted as a rectangle with dotted lines including the same shape and same color information of the target. But models have different poses $\boldsymbol{\phi}_{M}^{j}$(j=1, 2, $\cdots$, 20) as shown at the top of Fig. 5.8 (a) whose poses have been defined by the chromosomes, Eq. (5.21). Note that the system

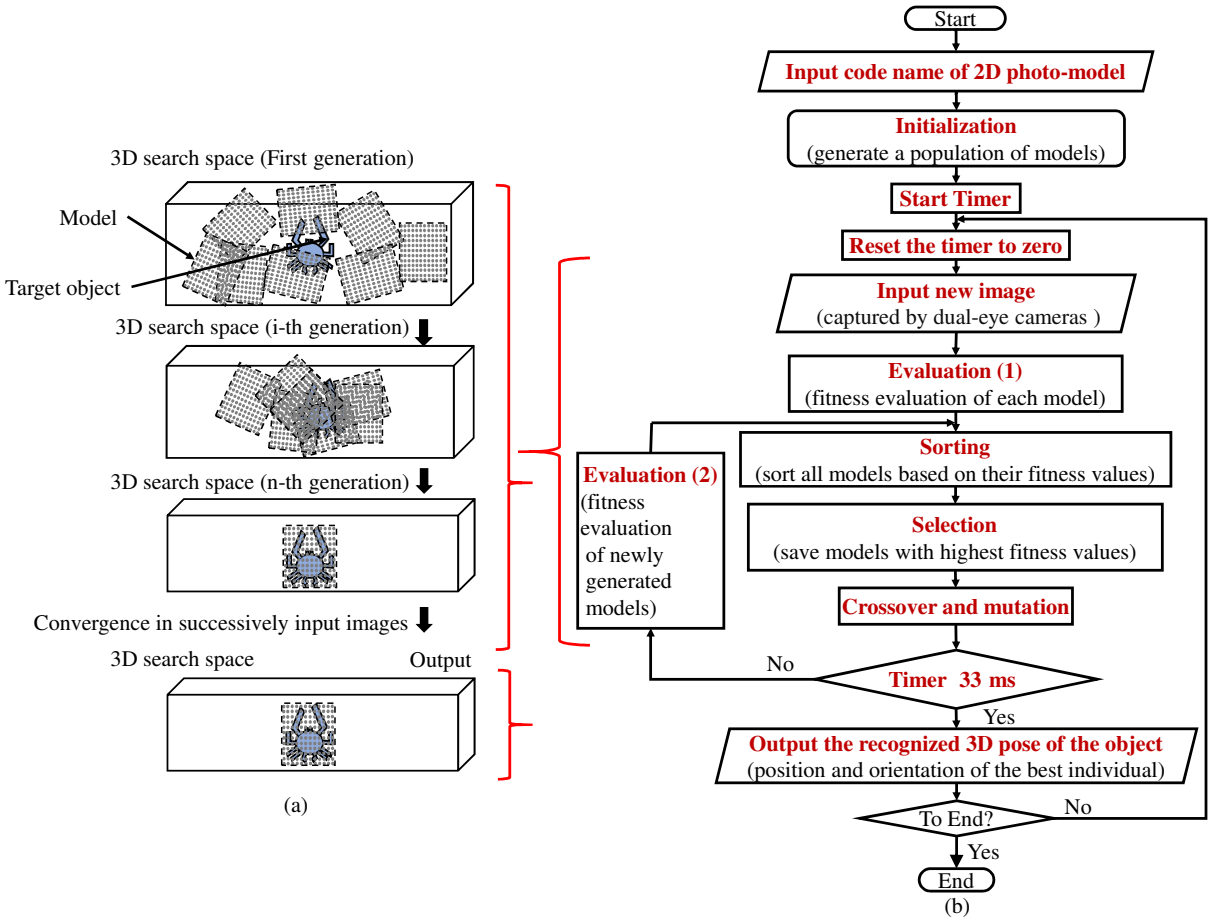


Fig. 5.8: RM-GA evolution process in which 3D models with random poses converge to the real 3D solid target object in 3D space. The pose of the model with the highest fitness value represents the estimated pose of the target object at that instant: (a) schematic diagram of the evolutionary process and (b) flowchart of RM-GA process during each 33[ms] control period, from "Input new image" to "Output."

performs the evaluation process in the left and right 2D image planes. And the convergence of searching models occurs in 3D searching space. The fitness function value evaluates the overlap degree between an individual and the target object. The fitter individuals are selected to regenerate the next genes. Thus, the genes converge to the real target after some transient period of evolution. Then, the gene that gives the highest fitness value stands for the most trustful pose as shown in the bottom part in Fig. 5.8 (a).

Figure 5.8 (b) shows a flowchart for the RM-GA evolution process for recognition and pose estimation:

(1) Firstly, the individuals are randomly generated in the 3D searching area as the first generation.

(2) New images captured by dual-eye cameras are input.

(3) The fitness value of every individual is calculated.

(4) Every individual's fitness value is sorted by the calculated fitness value.

(5) The best individual is selected from the current population, and the weak individuals are removed.

(6) Then, the individuals for the next generation are reproduced by making crossover and mutation between the selected individuals.

(7) Only new individuals in the next generation are evaluated by the fitness function, shown by "Evaluation (2)" block, because the right and left images do not change and top individuals with highest fitness do not need to calculate fitness again since the image is constant during 33[ms].

(8) And then, the above procedures (5)-(7) are repeated within 33 [ms]. Because the time needed for transferring one frame of video from image input board to the memory of main

CPU is 9[ms] , the remaining time within the video rate 33[ms] is $33 - 9 = 24$[ms]. Then 24[ms] remains for RM-GA to evolve.

(9) Finally, the RM-GA outputs the best individual. If the process is not ended, it will input new images and repeat the above procedures.

5.5 Fitness distribution

Fitness function Eq. (5.19) converts the target recognition and pose estimation problem into an optimization problem if variables to give the maximum peak represents the target's pose. To ensure whether this problem conversion about Eq. (5.19) is feasible, a way is a brute-force search or an exhaustive search. Using still pictures at an instant moment, the fitness value $F(\boldsymbol{\phi}_M^j)$ is calculated with its pose varied as parameters. We call it "fitness distribution." It is also a way to verify whether the RM-GA can detect the true pose of a target object at that moment. Even though the fitness distribution is made by an exhaustive search method, it is impossible to calculate all possibilities. This time, the position incremental distance of fitness value is set at 1.0[mm], and the orientation increment is 0.01[] (quaternion does not have the unit). Search ranges of fitness distribution are set as position: ${}^H x_M$ and ${}^H y_M \in [-180, 180]$[mm], ${}^H z_M \in [320, 680]$[mm]; orientation: ${}^H \varepsilon_{1M}$, ${}^H \varepsilon_{2M}$, and ${}^H \varepsilon_{3M} \in [-0.35, 0.35]$.

Figures 5.9 and 5.10 show left and right images captured by the stereo vision system and the fitness distribution of C04 dolphin and C12 crab in detail. Figure 5.9 (a) shows the left and right camera images of C04 dolphin, and (b), (c) show the $x - y$ and $y - z$ position fitness distribution respectively, and (d), (e) show the orientation fitness distribution. All the fitness distribution (b)$\sim$(e) have peaks. For example, in Fig. 5.9 (b), the $x - y$ position that gives maximum peak is $({}^H x_M, {}^H y_M) = (-3, 7)$[mm] and this result shows it is near the true position $(0, 0)$[mm] given by Eq. (6.2). About another object C12 crab, Fig. 5.10 (b) and (c) shows the position fitness distribution, and (d) and (e) show the orientation fitness distribution. All the fitness distribution (b)$\sim$(e) also have peaks near the true value. The results of other target

objects except of C04 and C12 are similar to Figures 5.9 and 5.10, then they are not listed in this paper. Each subfigure of the results has a main peak near the true value ${}^{H}\boldsymbol{\phi}_{M}$ given by Eq. (6.2). Therefore, it has been confirmed that fitness function Eq. (5.19) can convert the target recognition and pose estimation problem into an optimization problem. Furthermore, it has been confirmed that the proposed method can estimate 3D target pose by using stereo vision and 2D photo-model. But the gentle shapes of peaks given by (d) and (e) in Figs. 5.9 and 5.10 mean that the estimated orientations tend to include estimation errors than the positions whose fitness distributions have sharp peaks as shown in Figs. 5.9 and 5.10.

RM-GA searching experiments have been also conducted to compare with the fitness distribution. The results show that RM-GA can find the pose of all target objects from C01 to C12 in less than 10[s] by using the left and right still images. In this experiment, the optimization procedure is conducted by static still photographs not dynamic images, then the RM-GA process means usual GA process practically. For example, the left and right camera images shown at Fig. 5.9 (a) are used for the RM-GA searching experiment concerning C04 dolphin. And the detected pose by RM-GA ${}^{H}\boldsymbol{\phi}_{\widehat{M}} = [{}^{H}x_{\widehat{M}}, {}^{H}y_{\widehat{M}}, {}^{H}z_{\widehat{M}}, {}^{H}\varepsilon_{\widehat{1M}}, {}^{H}\varepsilon_{\widehat{2M}}, {}^{H}\varepsilon_{\widehat{3M}}]^{\mathrm{T}}$ is shown at the row of C04 in Table 5.1, which includes also results of other 3D toys shown in Fig. 5.2. The real pose that gives maximum peak is represented by ${}^{H}\boldsymbol{\phi}_{M} = [{}^{H}x_{M}, {}^{H}y_{M}, {}^{H}z_{M}, {}^{H}\varepsilon_{1M}, {}^{H}\varepsilon_{2M}, {}^{H}\varepsilon_{3M}]^{\mathrm{T}}$, and is shown in the same row. The detection errors $\Delta\boldsymbol{\phi} = {}^{H}\boldsymbol{\phi}_{M} - {}^{H}\boldsymbol{\phi}_{\widehat{M}} = [\Delta x, \Delta y, \Delta z, \Delta\varepsilon_1, \Delta\varepsilon_2, \Delta\varepsilon_3]^{\mathrm{T}}$ is also listed in the table. From the error values of C01$\sim$C12, it has been confirmed that the poses of all 3D toys could be estimated by GA evolutional procedures with the position error being less than 10[mm] and orientation error being less than 0.15 in quaternion.

In this section, by the fitness distribution experiment, it is verified that the fitness function Eq. (5.19) can transform the target position and orientation estimation problems into optimization problems. It is also confirmed that the proposed method can estimate the 3D target pose by using stereo vision and 2D photo-model. Since the estimated value of RM-GA is close to the peak result in the fitness distribution experiment, RM-GA can be used practically as a solution

Table 5.1: Peak coordinates ${}^{H}\boldsymbol{\phi}_M = [{}^{H}x_M, {}^{H}y_M, {}^{H}z_M, {}^{H}\varepsilon_{1M}, {}^{H}\varepsilon_{2M}, {}^{H}\varepsilon_{3M}]^{\mathrm{T}}$ of 12 target objects in the fitness distribution, RM-GA detection results ${}^{H}\boldsymbol{\phi}_{\widehat{M}} = [{}^{H}x_{\widehat{M}}, {}^{H}y_{\widehat{M}}, {}^{H}z_{\widehat{M}}, {}^{H}\varepsilon_{\widehat{1M}}, {}^{H}\varepsilon_{\widehat{2M}}, {}^{H}\varepsilon_{\widehat{3M}}]^{\mathrm{T}}$ and errors $\Delta\boldsymbol{\phi}_M = {}^{H}\boldsymbol{\phi}_M - {}^{H}\boldsymbol{\phi}_{\widehat{M}} = [\Delta x, \Delta y, \Delta z, \Delta\varepsilon_1, \Delta\varepsilon_2, \Delta\varepsilon_3]^{\mathrm{T}}$ are listed. Search range of fitness distribution, position: $x \in [-180, 180]$[mm], $y \in [-180, 180]$[mm], $z \in [320, 680]$[mm]; orientation: $\varepsilon_1, \varepsilon_2$, and $\varepsilon_3 \in [-0.35, 0.35]$. Search interval of fitness are 1.0[mm] in position; orientation: 0.01[]. True values given by TC-robot shown in Fig. 1.2 are ${}^{H}\boldsymbol{\phi}_M = [{}^{H}x_M, {}^{H}y_M, {}^{H}z_M, {}^{H}\varepsilon_{1M}, {}^{H}\varepsilon_{2M}, {}^{H}\varepsilon_{3M}]^{\mathrm{T}} = [0, 0, 500\text{[mm]}, 0, 0, 0]^{\mathrm{T}}$.

Pose / Target Number	Real pose that gives maximum peak						Detected pose by RM-GA						Error values					
	Position [mm]			Orientation (quaternion[])			Position [mm]			Orientation (quaternion[])			Position [mm]			Orientation (quaternion[])		
	${}^{H}x_M$	${}^{H}y_M$	${}^{H}z_M$	${}^{H}\varepsilon_{1M}$	${}^{H}\varepsilon_{2M}$	${}^{H}\varepsilon_{3M}$	${}^{H}x_{\widehat{M}}$	${}^{H}y_{\widehat{M}}$	${}^{H}z_{\widehat{M}}$	${}^{H}\varepsilon_{\widehat{1M}}$	${}^{H}\varepsilon_{\widehat{2M}}$	${}^{H}\varepsilon_{\widehat{3M}}$	Δx	Δy	Δz	$\Delta\varepsilon_1$	$\Delta\varepsilon_2$	$\Delta\varepsilon_3$
C01	-2.0	0.0	501.0	-0.03	-0.09	0.04	-2.25	0.39	497.62	-0.03	-0.15	0.04	0.25	-0.39	3.38	0.00	0.06	0.00
C02	5.0	-3.0	494.0	0.11	-0.18	-0.06	12.11	-2.64	495.57	0.10	-0.17	-0.05	-7.11	-0.36	-1.57	0.01	-0.01	-0.01
C03	-2.0	5.0	508.0	0.01	-0.1	0.03	-2.54	4.98	509.24	0.03	-0.07	0.02	0.54	0.02	-1.24	-0.02	-0.03	0.01
C04	-3.0	7.0	506.0	-0.02	-0.02	-0.01	-3.13	7.13	506.41	-0.03	-0.04	-0.04	0.13	-0.13	-0.41	0.01	0.02	0.03
C05	-1.0	-11.0	500.0	0.02	0.02	0.03	-0.39	-10.45	500.55	0.02	0.02	0.04	-0.61	-0.55	-0.55	0.00	0.00	-0.01
C06	-2.0	5.0	493.0	0.11	0.11	0.07	-3.13	5.66	494.30	0.16	0.04	0.02	1.13	-0.66	-1.30	-0.05	0.07	0.05
C07	12.0	4.0	517.0	0.03	-0.01	-0.01	9.38	4.79	514.22	-0.03	-0.08	-0.01	2.63	-0.79	2.78	0.06	0.07	0.00
C08	-6.0	-1.0	504.0	0.02	-0.02	-0.07	-6.05	-2.25	502.11	0.08	-0.07	-0.08	0.05	1.25	1.89	-0.06	0.05	0.01
C09	6.0	-6.0	502.0	-0.1	0.09	-0.04	3.81	-6.05	498.20	-0.06	0.09	-0.04	2.19	0.05	3.80	-0.04	0.00	0.00
C10	6.0	6.0	507.0	0.01	0.04	-0.04	6.35	4.00	504.45	-0.06	-0.10	-0.04	-0.35	2.00	2.55	0.07	0.14	0.00
C11	9.0	1.0	513.0	0.04	0.16	-0.06	7.62	1.27	509.14	0.05	0.09	-0.05	1.38	-0.27	3.86	-0.01	0.07	-0.01
C12	0.0	1.0	497.0	-0.09	0.06	0.01	0.29	1.76	498.50	0.06	-0.04	0.03	-0.29	-0.76	-1.50	-0.15	0.10	-0.02

to detect the pose of the 3D target objects by using 2D photo-model.

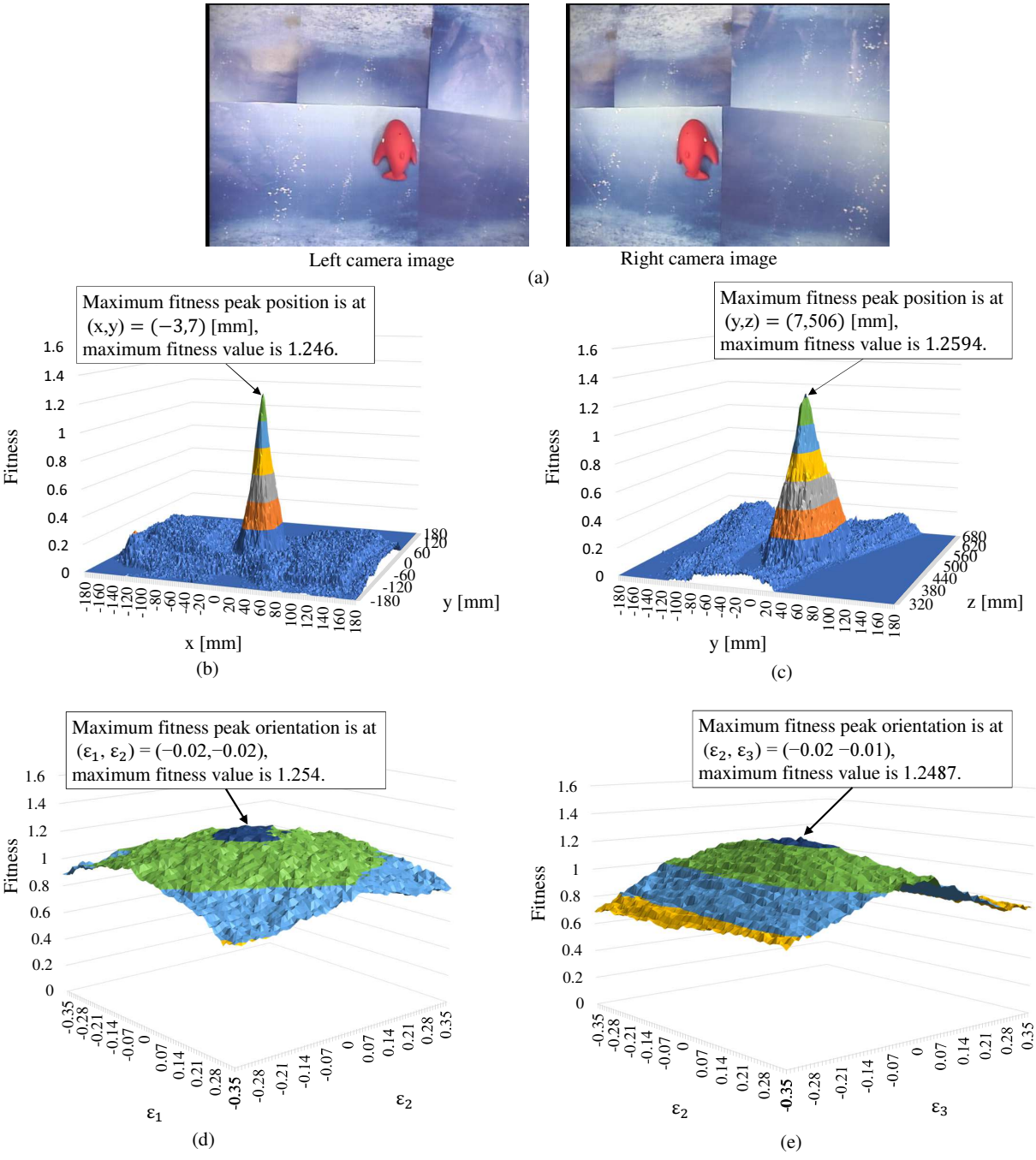


Fig. 5.9: Fitness distribution of C04 dolphin. (a) Left and right camera images, (b) fitness distribution in the x-y plane, (c) fitness distribution in the y-z plane, (d) fitness distribution of orientation in ε_1-ε_2, and (e) fitness distribution of orientation in ε_2-ε_3. In each subfigure of (b)∼(e), the maximum fitness and corresponding coordinate are shown in a text box.

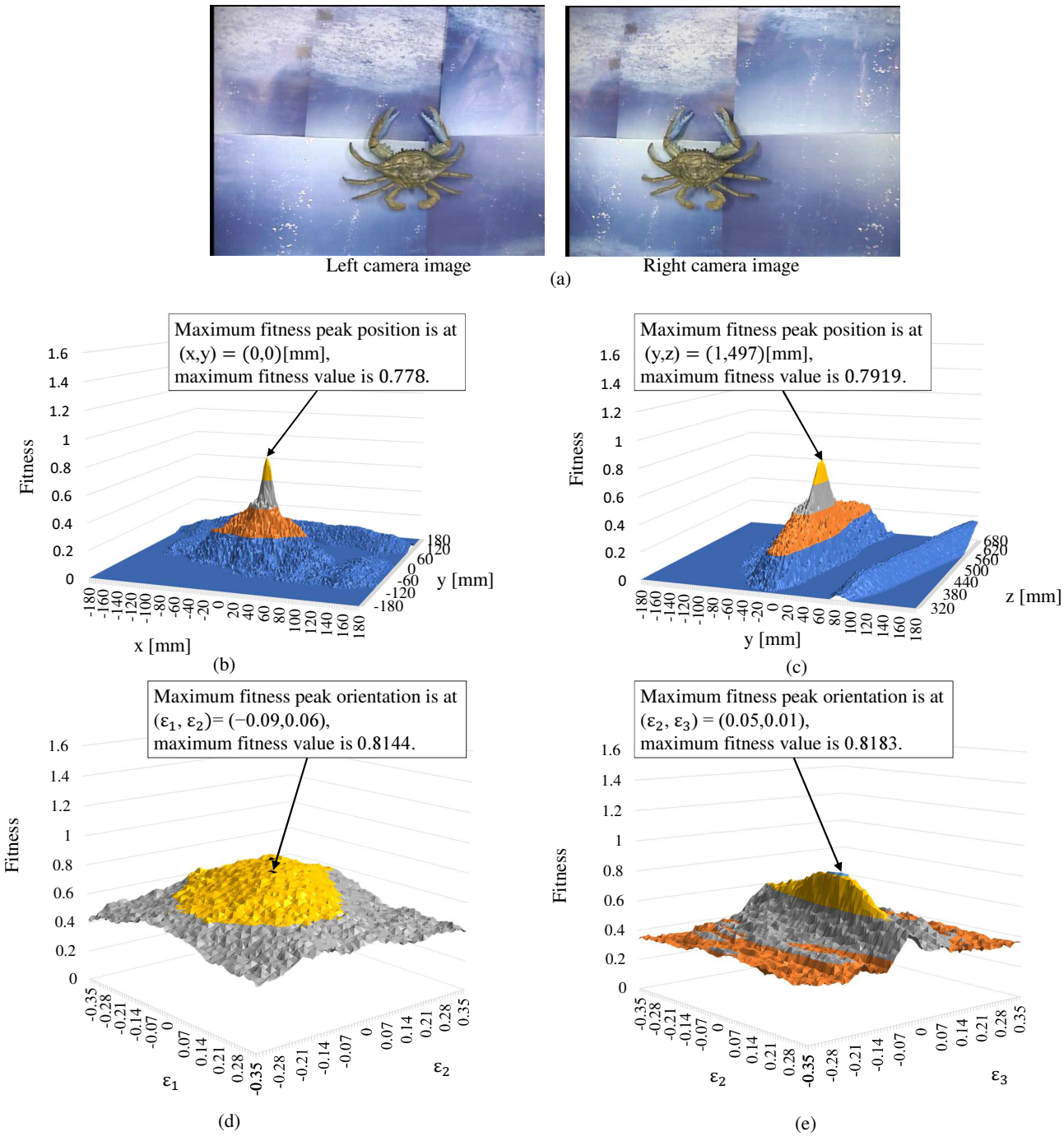


Fig. 5.10: Fitness distribution of C12 crab. (a) Left and right camera images, (b) fitness distribution in the x-y plane, (c) fitness distribution in the y-z plane, (d) fitness distribution of orientation in ε_1-ε_2, and (e) fitness distribution of orientation in ε_2-ε_3. In each subfigure of (b)∼(e), the maximum fitness and corresponding coordinate are shown in a text box.

Chapter 6

Experiments of Photo-Model-Based Visual Servoing

6.1 Pose 6DOF visual tracking

As introduced in Section 2.2, the guidance of robots through real-time and continuous visual feedback is generally known as visual servoing. Only the constant observation of the objects of interest is referred to as visual tracking, and it does not involve the robot control. The visual tracking is essential as a basis for visual servoing.In this section, the visual tracking experiment will be conducted to verify the tracking ability of the proposed stereo vision recognition method. In the Sections 6.2 and 6.3, visual servoing experiments will be conducted to confirm the feedback control ability of the visual servoing robot.

6.1.1 Experimental content

The initial condition of the pose real-time estimation experiment is shown at top subfigure (Step 0) in Fig. 6.1. The pose of the target object represented by Σ_M based on the end-effector Σ_H is set as Eq. (6.2). In Fig. 6.1, each subfigure, i.e., each (Step), is a state of the target object at a special time point in the experiment. For example, subfigure (Step 0) shows the state of

the target object at the beginning time point $t = 0$[s] of the experiment. And the target's pose of (Step 0) ${}^{H}\boldsymbol{\phi}_M = [{}^{H}x_M, {}^{H}y_M, {}^{H}z_M, {}^{H}\varepsilon_{1M}, {}^{H}\varepsilon_{2M}, {}^{H}\varepsilon_{3M}]^{\mathrm{T}} = [0, 0, 500[\mathrm{mm}], 0, 0, 0]^{\mathrm{T}}$ is also shown in Table 6.1. In this pose estimation experiment, the VS-robot in Fig. 1.2 does not move. The TC-robot controls the target object to move, with one of the elements of target pose of $({}^{H}x_M, {}^{H}y_M, {}^{H}z_M, {}^{H}\varepsilon_{1M}, {}^{H}\varepsilon_{2M}, {}^{H}\varepsilon_{3M})$ being changed and others being kept to be constant as shown in Table 6.1. The table lists the pose of TC-robot and the transition of the pose when the target pose is changed from (Step 0) to (Step 19).

For example, as shown in Fig. 6.1, from (Step 0) to (Step 1), ${}^{H}x_M$ that is x-coordinate of target pose, is changed from 0[mm] to -50[mm] by TC-robot based on the Σ_M. ${}^{H}y_M$, ${}^{H}z_M$, and orientation parameters ${}^{H}\boldsymbol{\varepsilon}_M$ are constant. In subfigure (Step 1) of Fig. 6.1, the arrow shows the moving direction along the x-axis from former (Step 0) to this (Step 1). And as shown in Table 6.1, the arrow between rows (Step 0) and (Step 1) in the column of ${}^{H}x_M$ has the same meaning and indicates that only ${}^{H}x_M$ is changed from 0[mm] at (Step 0) to -50[mm] at (Step 1). In Fig. 6.1, from (Step 1) to (Step 2), ${}^{H}y_M$ that is y-coordinate of target pose, is changed from 0[mm] to -50[mm] by TC-robot. ${}^{H}x_M$, ${}^{H}z_M$, and orientation parameters ${}^{H}\boldsymbol{\varepsilon}_M$ at (Step 2) are the same with the parameters at (Step 1). And the arrow in subfigure (Step 2) in Fig. 6.1 shows the moving direction along the y-axis. And as shown in Table 6.1, the arrow between rows (Step 1) and (Step 2) in the column of ${}^{H}y_M$ also represents that only ${}^{H}y_M$ is changed from 0[mm] at (Step 1) to -50[mm] at (Step 2) by TC-robot. The position of TC-robot is changed with the same manner from (Step 2) to (Step 10), which represents the same pose as (Step 0).

From (Step 11) to (Step 19), the position of the target is kept to be constant, but the orientation is changed. The TC-robot rotates the target to $\varepsilon_{1M} = 0.174$ around x_M axis at (Step 11) and then to $\varepsilon_{1M} = -0.174$ around x_M axis at (Step 12). And at (Step 13) the target is rotated back to the initial pose of (Step 10). From (Step 14) to (Step 16), the target object rotates only around y-axis and from (Step 17) to (Step 19) it does around z-axis. The poses of Σ_M at (Step 10), (Step 13), (Step 16), and (Step 19) are the same with the initial state (Step 0).

Table 6.1 shows the real pose of the target object at each step. The position trajectories of

Table 6.1: The target pose value ${}^{H}\boldsymbol{\phi}_{M} = [{}^{H}x_{M}, {}^{H}y_{M}, {}^{H}z_{M}, {}^{H}\varepsilon_{1M}, {}^{H}\varepsilon_{2M}, {}^{H}\varepsilon_{3M}]^{\mathrm{T}}$ of each motion step is listed with names of (Step 0) to (Step 19), corresponding to the target's motion trajectory in Fig. 6.1. Similar to Fig. 6.1, the arrows in this table show the changing parameters from the previous step to the next. For example, in this table, since from (Step 0) to (Step 1) ${}^{H}x_{M}$ is only changed, there is an arrow between row (Step 0) and (Step 1) in the column of ${}^{H}x_{M}$. And the arrow of subfigure (Step 1) in Fig. 6.1 also shows that the target moves along the x-axis.

Pose / Step	Position[mm]			Orientation(quaternion[])		
	${}^{H}x_{M}$	${}^{H}y_{M}$	${}^{H}z_{M}$	${}^{H}\varepsilon_{1M}$	${}^{H}\varepsilon_{2M}$	${}^{H}\varepsilon_{3M}$
(Step 0)	0	0	500	0	0	0
(Step 1)	-50	0	500	0	0	0
(Step 2)	-50	-50	500	0	0	0
(Step 3)	0	-50	500	0	0	0
(Step 4)	0	0	500	0	0	0
(Step 5)	0	0	550	0	0	0
(Step 6)	-50	0	550	0	0	0
(Step 7)	-50	-50	550	0	0	0
(Step 8)	0	-50	550	0	0	0
(Step 9)	0	0	550	0	0	0
(Step 10)	0	0	500	0	0	0
(Step 11)	0	0	500	0.174	0	0
(Step 12)	0	0	500	-0.174	0	0
(Step 13)	0	0	500	0	0	0
(Step 14)	0	0	500	0	0.174	0
(Step 15)	0	0	500	0	-0.174	0
(Step 16)	0	0	500	0	0	0
(Step 17)	0	0	500	0	0	-0.174
(Step 18)	0	0	500	0	0	0.174
(Step 19)	0	0	500	0	0	0

the target are shown at the subfigures from (Step 0) to (Step 10) in Fig. 6.1 and the time profiles of target pose given by TC-robot are depicted as (a) to (f) at the center of Fig. 6.1. Target's pose time profiles (a)∼(f) are enlarged and shown as dashed lines in Fig. 6.2. The solid lines in Fig. 6.2 show the pose estimation results. The 3D pose estimation error is shown as Fig. 6.3.

6.1.2 Results and discussion

Figure 6.2 (a)∼(f) show the pose estimation results $[{}^{H}x_{\widehat{M}}, {}^{H}y_{\widehat{M}}, {}^{H}z_{\widehat{M}}, {}^{H}\varepsilon_{\widehat{1M}}, {}^{H}\varepsilon_{\widehat{3M}}, {}^{H}\varepsilon_{\widehat{3M}}]^{\mathrm{T}}$ depicted with solid lines. (a)∼(c) are position recognition results. (d)∼(f) are orientation recog-

nition results. The true values ${}^{H}\boldsymbol{\phi}_M = [{}^{H}x_M, {}^{H}y_M, {}^{H}z_M, {}^{H}\varepsilon_{1M}, {}^{H}\varepsilon_{2M}, {}^{H}\varepsilon_{3M}]^{\mathrm{T}}$ are shown as dashed lines, which is enlarged from Fig. 6.1. The descriptions of (Step 0)∼(Step 19) in Fig. 6.2, where "Step" has been eliminated to save space, are the time points corresponding to those in Fig. 6.1. In the beginning period of recognition time $t = 0 \sim 6$[s], the detection results of RM-GA gradually converge to the true pose ${}^{H}\boldsymbol{\phi}_M$. Then, the estimation results are almost similar to the real pose. Even though the detection result ${}^{H}z_{\widehat{M}}$ in (c) have some fluctuations when the target moves along the x- or y-axis at (Step 1), (Step 4), (Step 6), and (Step 8), RM-GA can quickly converge to the true pose in the later. The position estimation results in (a)∼(c) shows that the proposed method can track the position of the moving target object. The orientation estimation results in the period of (Step 11)∼(Step 13) in (d), (Step 14)∼(Step 16) in (e), and (Step 17)∼(Step 19) in (f) show that this method can also track the changing orientation of the target in real-time.

Figure 6.3 shows the errors of the pose tracking results. And the detection errors of x and y coordinates in (a) and (b) are in the range of ± 20[mm] except at time (Step 3), (Step 6), and (Step 8), which represents the time that the target's motion includes accelerations. And about distance estimation in z coordinate, Fig. 6.3 (c) shows that the error is in the range of ± 30[mm] roughly. Some large fluctuations in (a)∼(c) show the time delay of position coordinate detection, e.g., (Step 1), (Step 6), and (Step 8) in (a). About the orientation estimation, the error of ε_3 in (f) is small and less than those of ε_1 in (d) and ε_2 in (e). In the period of (Step 1)∼(Step 10), it can be confirmed that the change of position of the target object interferes with tracking errors of the orientation estimation. And in the period of (Step 11) to (Step 19), even though the orientation of the target object has been changed, the position detection errors in (a)∼(c) are kept to be small.

Through the above analyses and discussions of experimental results, it has been confirmed that the proposed photo-model-based recognition method can detect an object's pose in real-time by using RM-GA.

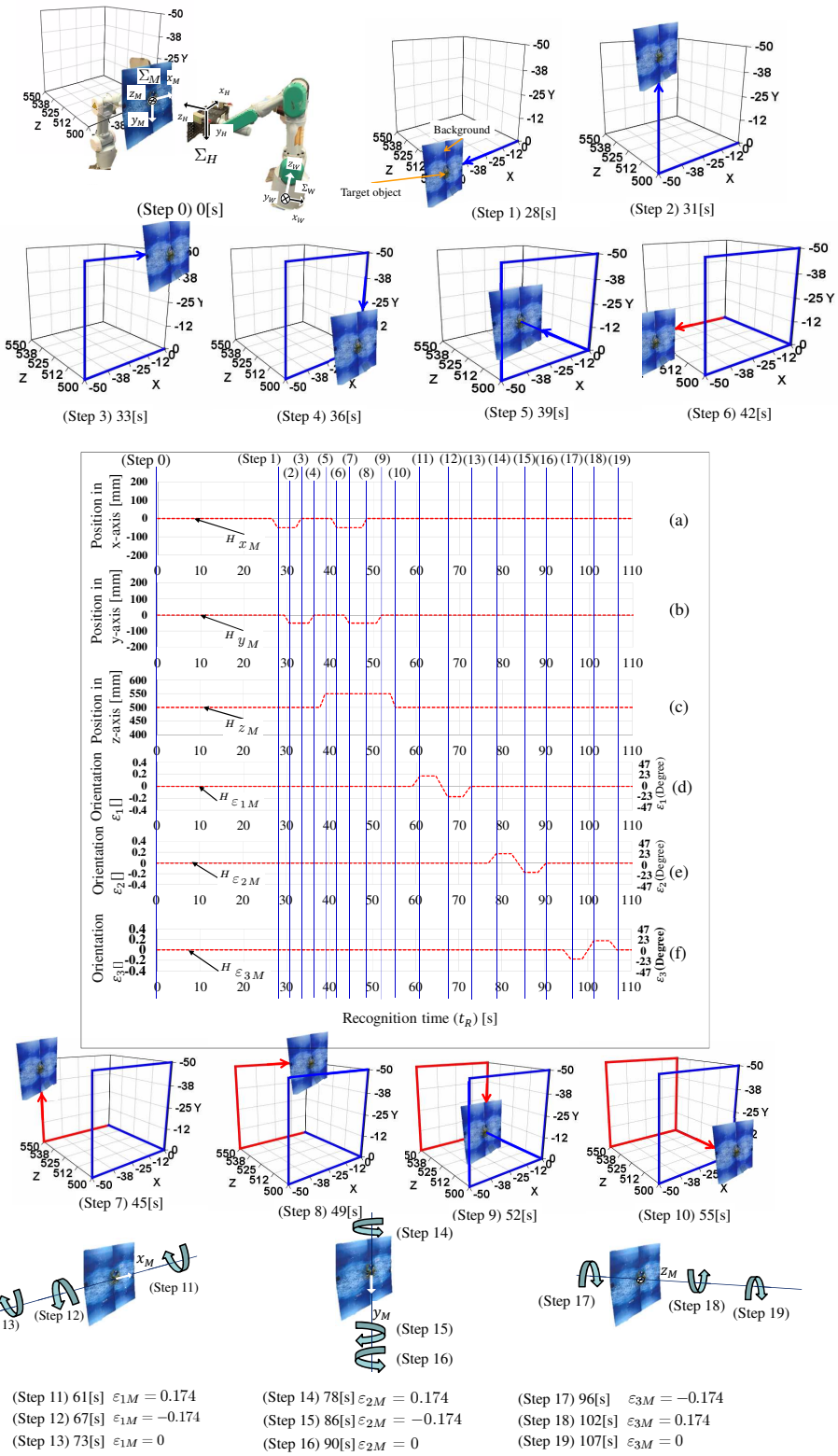


Fig. 6.1: The target in this pose tracking experiment is C12 crab shown in Figs. 5.2 and 5.3. The subfigures of (a)∼(f) show all the poses time profile of the target Σ_M based on Σ_H. The crab's position time profile is shown by (a), (b), and (c) based on the end-effector Σ_H. Orientation motion is shown by (d), (e), and (f). The subfigures (Step 0)∼(Step 19) shows the target motion schematically. The arrows in (Step 0)∼(Step 19) show the target's moving direction. Motion curves (a)∼(f) are enlarged and shown as dashed lines in Fig. 6.2. The poses of Σ_M at (Step 10), (Step 13), (Step 16), and (Step 19) are the same with the initial state (Step 0).

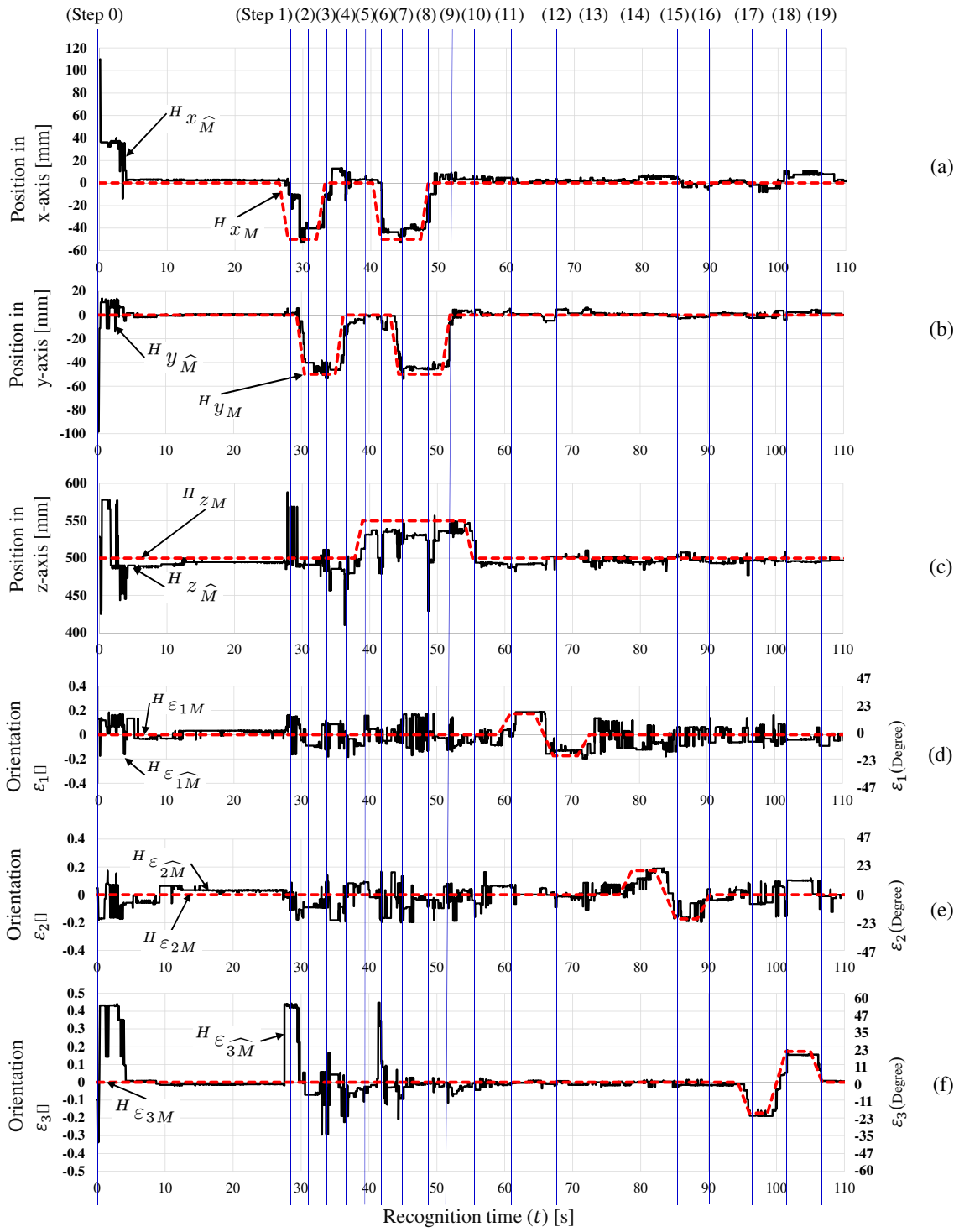


Fig. 6.2: The 3D pose estimation results of the target whose motions are displayed in Fig. 6.1. The target is C12 crab shown in Fig. 5.2 and 5.3. The crab's position detection results are shown in above (a), (b), and (c) as solid lines. Orientation detection results are shown in (d), (e), and (f) as solid lines. The dashed lines are enlarged from Fig. 6.1 and show the true pose of the target object. (Step 0)∼(Step 19) that are written at the top of this figure show the specific time points which are corresponding to the subfigures in Fig. 6.1. And from (Step 2) to (Step 19), "Step" has been eliminated to save space. The right side axes of (d)∼(f) indicate angles that are calculated from quaternion to degree.

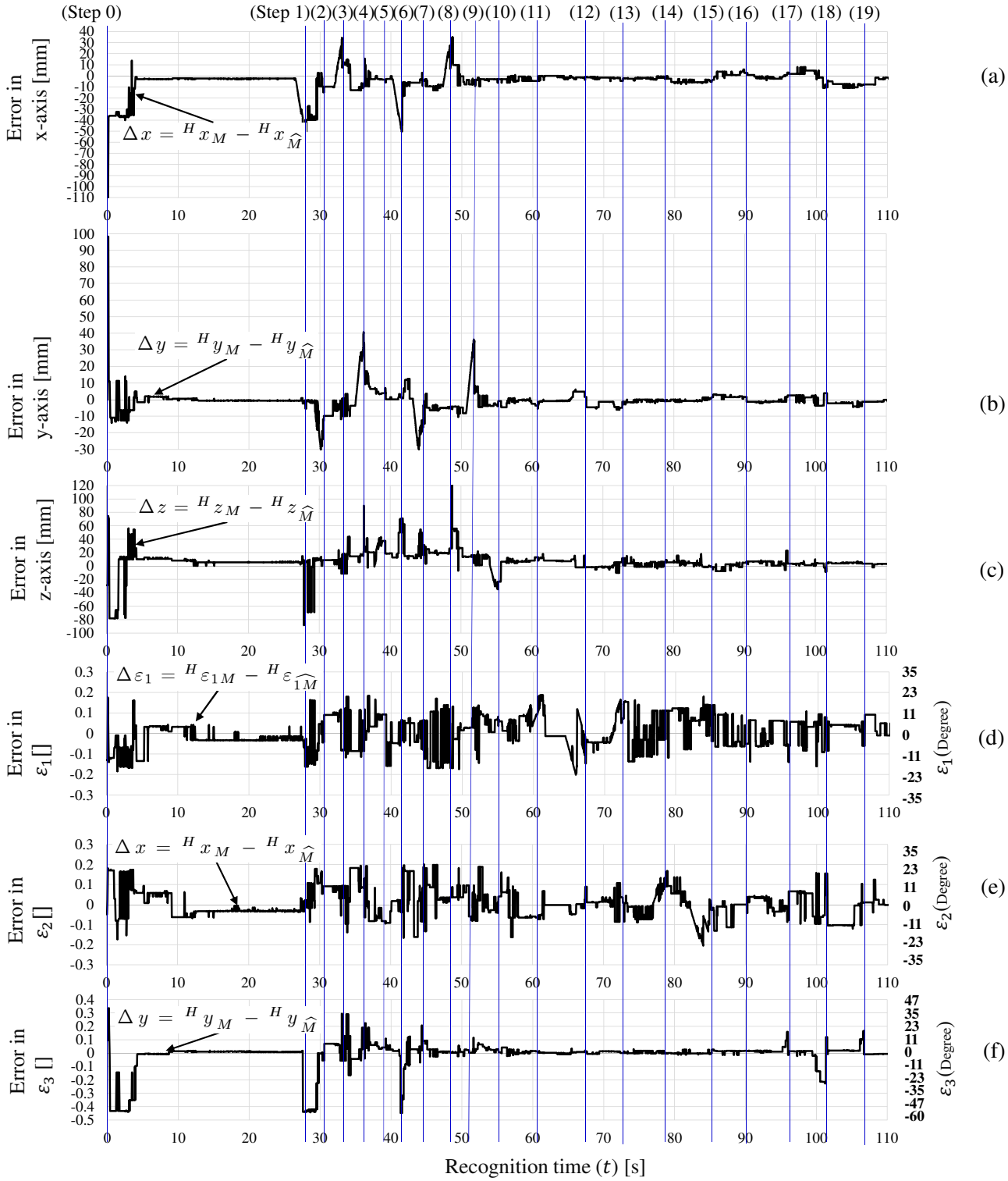


Fig. 6.3: The 3D pose estimation errors corresponding to Fig. 6.2. The crab's position detection errors are shown in (a), (b), and (c). Orientation detection errors are shown in (d), (e), and (f). (Step 0)∼(Step 19) at the top of this figure show the specific time points which are corresponding to the subfigures in Fig. 6.1. And from (Step 2) to (Step 19), "Step" has been eliminated to save space.

6.2 Visual servoing experiments with two manipulators

In this section, the target object's pose 6DOF visual servoing will be conducted to confirm the tracking ability of the visual servoing system. In these experiments, Σ_M will move along predetermined trajectories. Therefore, ${}^W\boldsymbol{T}_M$ is a known time-varying matrix. The goal of the visual servoing experiment is to control the end-effector to maintain the relative pose with the target object as

$$ {}^M\boldsymbol{T}_{Hd} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -500[\text{mm}] \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6.1} $$

where Σ_{Hd} is the desired pose of Σ_H.

At the beginning, VS-robot shown in Fig. 1.2 is set in front of TC-robot, and the 3D target's pose based on Σ_H is set as

$$ {}^H\boldsymbol{\phi}_M = [{}^H\boldsymbol{r}_M^{\mathrm{T}}, {}^H\boldsymbol{\varepsilon}_M^{\mathrm{T}}]^{\mathrm{T}} = [{}^Hx_M, {}^Hy_M, {}^Hz_M, {}^H\varepsilon_{1M}, {}^H\varepsilon_{2M}, {}^H\varepsilon_{3M}]^{\mathrm{T}} = [0, 0, 500[\text{mm}], 0, 0, 0]^{\mathrm{T}}. \tag{6.2} $$

6.2.1 Visual servoing with the object's position changing

In the position frequency response experiment, the target object is C12 crab. Its position trajectories are sine curves with an amplitude of 100 [mm] and a period of 20 [s] in the y_W and z_W-axes directions, and with amplitude of 100 [mm] and a period of 60 [s] in the x_W-axis direction. Its orientation does not change, i.e., $\boldsymbol{\varepsilon}_M = \boldsymbol{0}$.

The experiment results are shown in Fig. 6.4. The point curves ${}^W\boldsymbol{r}_M = [{}^Wx_M, {}^Wy_M, {}^Wz_M]^{\mathrm{T}}$ represents the predetermined movement trajectory of Σ_M along the x_W-, y_W-, and z_W-axes of Σ_W and orientation $\boldsymbol{\varepsilon}_M = [\varepsilon_{1M}, \varepsilon_{2M}, \varepsilon_{3M}]^{\mathrm{T}} = [0, 0, 0]^{\mathrm{T}}$. The solid lines ${}^W\boldsymbol{r}_H =$

$[{}^W x_H, {}^W y_H, {}^W z_H]^{\mathrm{T}}$ and $\boldsymbol{\varepsilon}_H$ are the visual servoing results of end-effector. Figure 1.2 shows the initial status of Σ_M and Σ_H at $t = 0$[s].

It can be seen that even though the tracking curves ${}^W\boldsymbol{r}_H$ delay somewhat in phase, the visual servoing system with photo-model-based recognition method can track the object ${}^W\boldsymbol{r}_M$ in time.

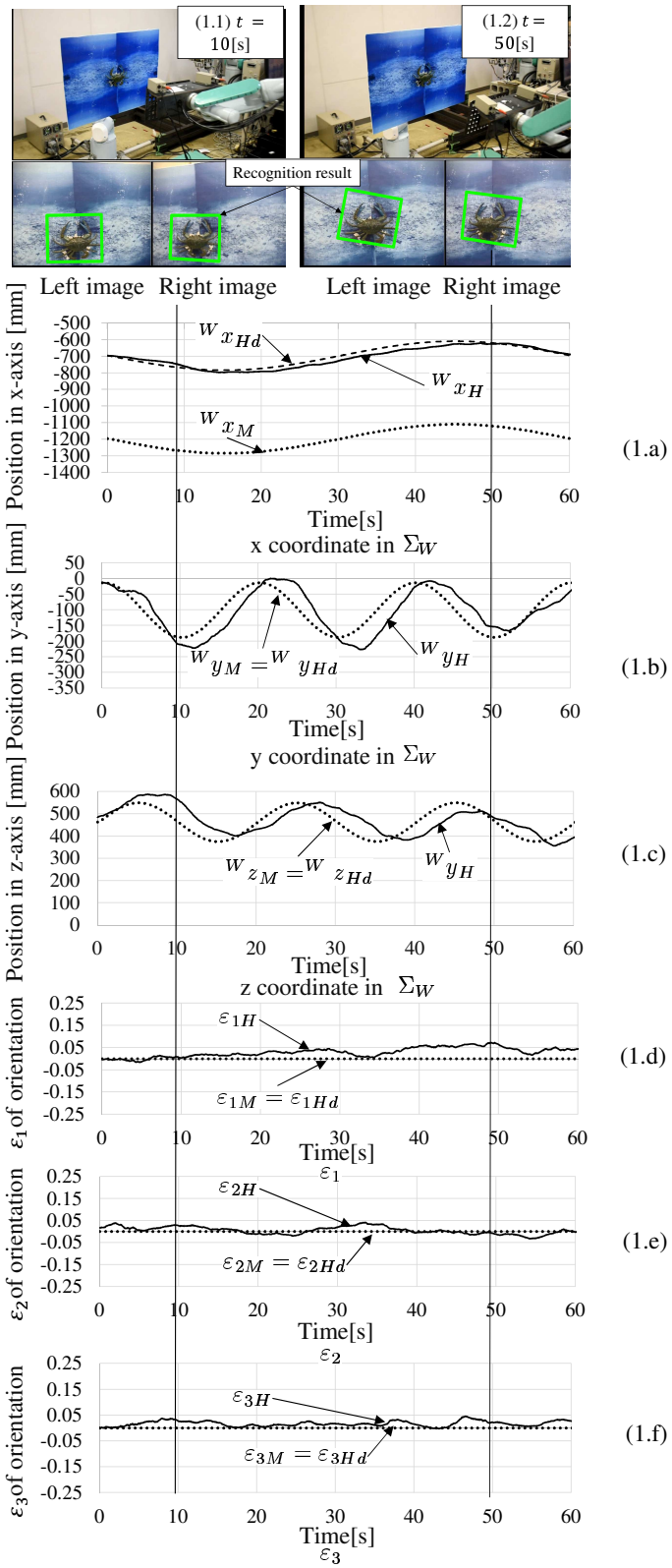
6.2.2 Visual servoing with the object's orientation changing

In this subsection, visual servoing experiment will conducted with the object's orientation changing. Its position ${}^W\boldsymbol{r}_M$ does not change. Σ_M rotates half period, i.e., 20[s], around z_M, x_M, and y_M axes with sine wave respectively. The maximum rotation degree is 25[°], i.e., 0.216 in quaternion [34]. The subfigures (2.1)∼(2.3) on the top of Fig. 6.5 show some states of target object and end-effector during the experiment. According to the visual servoing control goal Eq. (6.1), even though the target object only rotates without position changing, the end-effector needs to adjust both own position and orientation to face towards the target object. Therefore, the position/orientation visual servoing experiment is more complicated than the position one.

As shown in (2.a)∼(2.c), the broken lines show the desired position ${}^W\boldsymbol{r}_{Hd} = [{}^W x_{Hd}, {}^W y_{Hd}, {}^W z_{Hd}]^{\mathrm{T}}$ of the end-effector. In (2.d)∼(2.f), according to the control goal Eq. (6.1), end-effector's desired orientation $\boldsymbol{\varepsilon}_{Hd} = \boldsymbol{\varepsilon}_M$. The visual servoing results of the end-effector $({}^W x_H, {}^W y_H, {}^W z_H, \varepsilon_{1H}, \varepsilon_{2H}, \varepsilon_{3H})$ are shown as solid lines in Fig. 6.5.

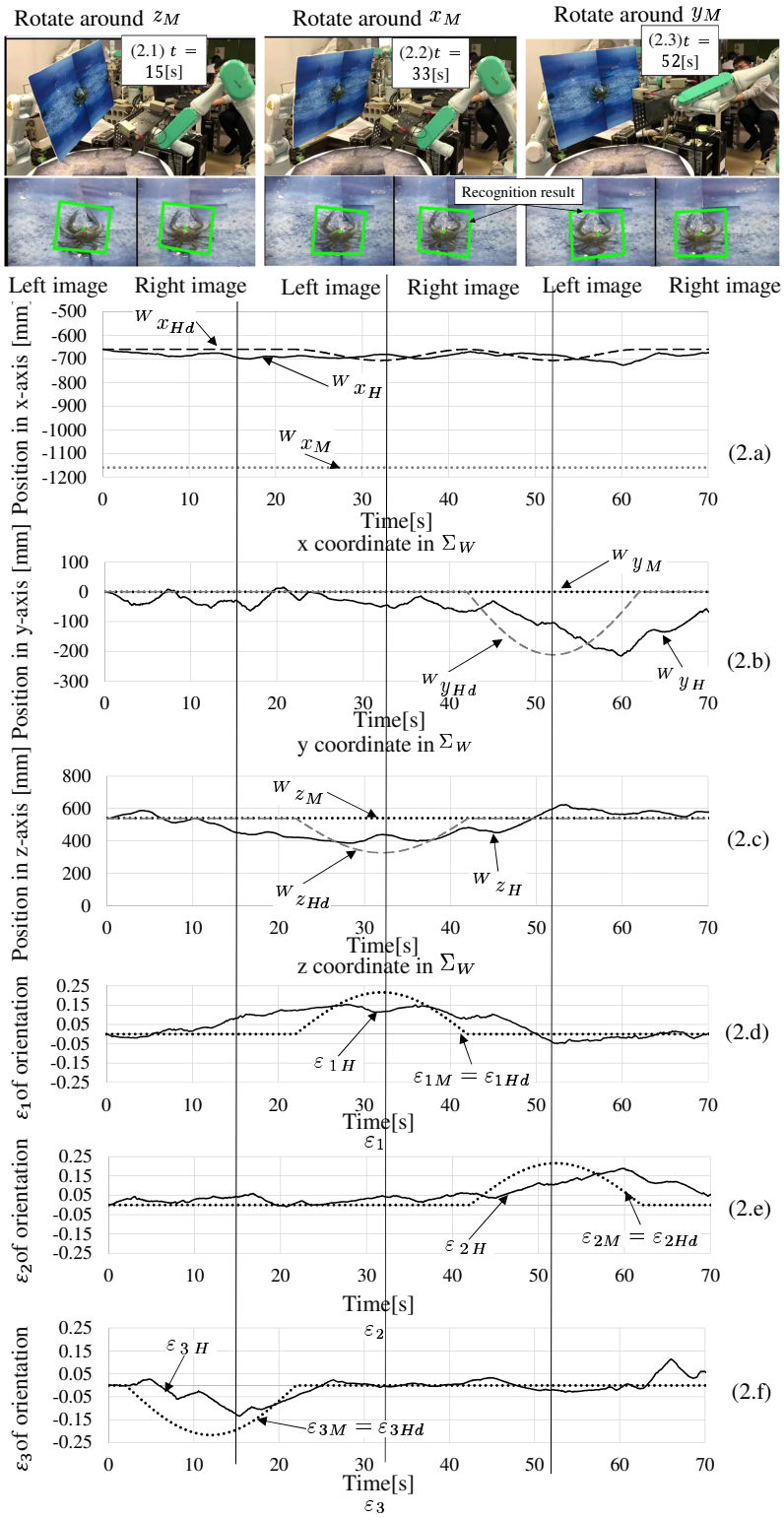
In (2.a)∼(2.c), the position curves ${}^W x_M$, ${}^W y_M$, and ${}^W z_M$ of Σ_M do not change. In (2.d)∼(2.f), the point curves $\boldsymbol{\varepsilon}_M = [\varepsilon_{1M}, \varepsilon_{2M}, \varepsilon_{3M}]^{\mathrm{T}}$ represents the rotation trajectories of Σ_M. Starting from the initial status $\boldsymbol{\varepsilon}_M = \mathbf{0}$, the target object rotates around z_M axis. After it rotates back to the initial status, it starts to rotate around x_M axis. Rotational motion is performed separately. For example, as shown in Fig. 6.5 (2.d), when the target object rotates around x-axis, only ε_{1M} has value and $\varepsilon_{2M} = \varepsilon_{3M} = 0$.

Through the results in (2.d)∼(2.f), on the overall trend, it can be seen that $\boldsymbol{\varepsilon}_H$ varies with the changing of $\boldsymbol{\varepsilon}_M$. And (2.a)∼(2.c) indicates that the real position ${}^W\boldsymbol{r}_H$ of the end-effector is also near to the desired one ${}^W\boldsymbol{r}_{Hd}$. Therefore, the visual servoing system with photo-model-based



(1) Position visual servoing

Fig. 6.4: Visual servoing with the object's position changing. Position ${}^W\boldsymbol{r}_M = [{}^W x_M, {}^W y_M, {}^W z_M]^{\mathrm{T}}$ of crab changes in sine wave. Object's orientation $\boldsymbol{\varepsilon}_M = [\varepsilon_{1M}, \varepsilon_{2M}, \varepsilon_{3M}]^{\mathrm{T}}$ does not change.



(2) Orientation visual servoing

Fig. 6.5: Visual servoing with the object's orientation changing. Position $^W\boldsymbol{r}_M$ does not change. Σ_M rotates half period around z_M, x_M, and y_M axes with sine wave respectively. The desired pose of the end-effector is $(^W x_{Hd},\ ^W y_{Hd},\ ^W z_{Hd}, \varepsilon_{1Hd}, \varepsilon_{2Hd},\ \varepsilon_{3Hd})$. The real pose of the end-effector is $(^W x_H,\ ^W y_H,\ ^W z_H, \varepsilon_{1H}, \varepsilon_{2H},\ \varepsilon_{3H})$.

recognition method can track the object's orientation in time. And the position can be detected correctly although the orientation ε_M changes. The results verified that the 2D photo-model of a 3D target is able to estimate the 3D target pose.

6.3 Position visual servoing with pool environment

6.3.1 Experimental environment and content

As shown in Fig. 6.6, a squid toy is a target object. And a marker pen is hung on the end-effector along the Z_H direction. In the experiment, the stereo vision detects the pose six parameters ${}^H\boldsymbol{\phi}_M$ of the object and the detection result is ${}^H\boldsymbol{\phi}_{\widehat{M}}$. The goal is based on ${}^H\boldsymbol{\phi}_{\widehat{M}}$ to control the end-effector to move to the top of the squid object and to maintain the relative position

$$ {}^{Hd}\boldsymbol{r}_M = [{}^{Hd}x_M, {}^{Hd}y_M, {}^{Hd}z_M]^{\mathrm{T}} = [0, 0, 600]^{\mathrm{T}}[\mathrm{mm}], \tag{6.3} $$

where Σ_{Hd} is the desired pose of the end-effector. The squid floats on the water in the pool without pose constraints. In the end, a marker pen is released and falls off to hit the squid to confirm the position visual servoing ability. The purpose of the experiment is to verify the availability of the proposed photo-model-based visual servoing system for catching a marine creature.

6.3.2 Results and discussion of the experiment

Figure 6.7 shows the states of the visual servoing in chronological order. At the beginning $t = 0$[s], in (a), the distance between Σ_H and Σ_M at the vertical direction was ${}^Hz_M = {}^Wz_H - {}^Wz_M = 680$[mm]. In other directions, ${}^Hx_M$ and ${}^Hy_M$ were unknown. Figure 6.7 (b) shows that at $t = 5$[s], the end-effector has been controlled to move near the target position with a height about ${}^Hz_M = 600$[mm]. Comparing with (a), the height of the end-effector had a significant drop in (b). In two camera images of (b), the squid became bigger than that in (a). As shown

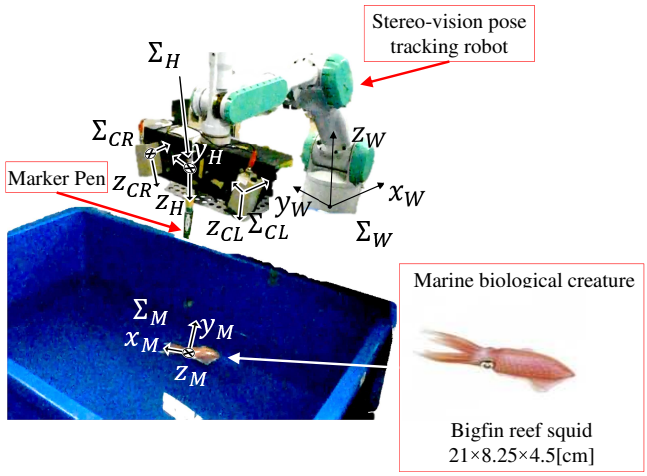


Fig. 6.6: Photo-model-based stereo vision system

in Fig. 6.7 (c) and (f), in the experiment, the moving direction of the squid target object was change by a stick randomly. And in (d), (e) and (g), the wave was made by the stick to mimic the natural situation. In the end, in (h), the marker pen was released and hit the squid.

Figure 6.8 shows the experimental results. The dotted line shows the pose of the end-effector. The solid line shows the pose of the object detected by RM-GA.

In the top subfigures (a), (b), and (c), ${}^{W}x_{H}$, ${}^{W}y_{H}$, and ${}^{W}z_{H}$ are the position tracking results of the end-effector. ${}^{W}x_{\widehat{M}}$, ${}^{W}y_{\widehat{M}}$, and ${}^{W}z_{\widehat{M}}$ are position recognition results of RM-GA and calculated by Eq. (5.5) for comparing with the position tracking results of the robot. They are all based on Σ_W.

In the bottom subfigures (d), (e), and (f), ε_{1H}, ε_{2H}, and ε_{3H} are the relative orientation of the end-effector to its initial status calculated by Eq. (5.14). Because in this experiment end-effector's orientation dose not change, they are all 0. ${}^{H}\varepsilon_{\widehat{1M}}$, ${}^{H}\varepsilon_{\widehat{2M}}$, and ${}^{H}\varepsilon_{\widehat{3M}}$ are orientation recognition results of RM-GA based on Σ_H.

In this experiment, the true pose of the target object is unknown. We only know the hight ${}^{H}z_{M} = 680$[mm] at the beginning $t = 0$[s]. In (c), at $t = 0$[s], the detection result ${}^{H}z_{\widehat{M}} = {}^{W}z_{H} - {}^{W}z_{\widehat{M}}$ was near to 680[mm]. It can be seen that the stereo vision can detect the distance correctly. Later, ${}^{W}z_{H}$ reduced. The end-effector moved down to track the squid at 600[mm] relative hight because of the control goal Eq.(6.3). Even if the orientation of the squid changed,

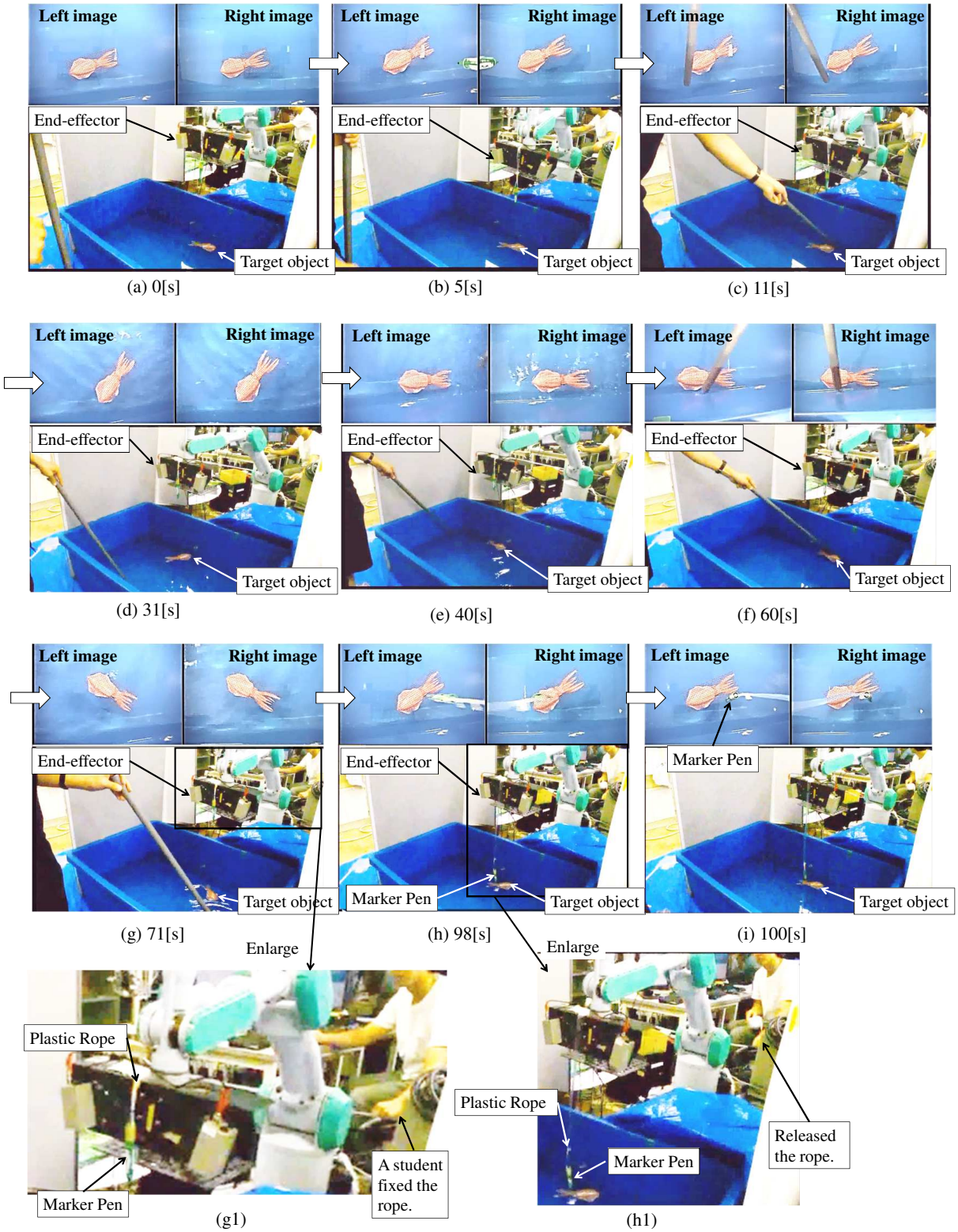


Fig. 6.7: Position 3DOF visual servoing experiment with pose (position/orientation) 6DOF estimation. The marker pen was tied on a rope and hung near the end-effector. From (a) to (g), the rope was fixed by a student. At (h), the rope is released, and the marker pen hit the squid. At the end (i), the squid drifted away due to the impact. (g1) and (h1) are enlarged views of a part of (g) and (h) respectively.

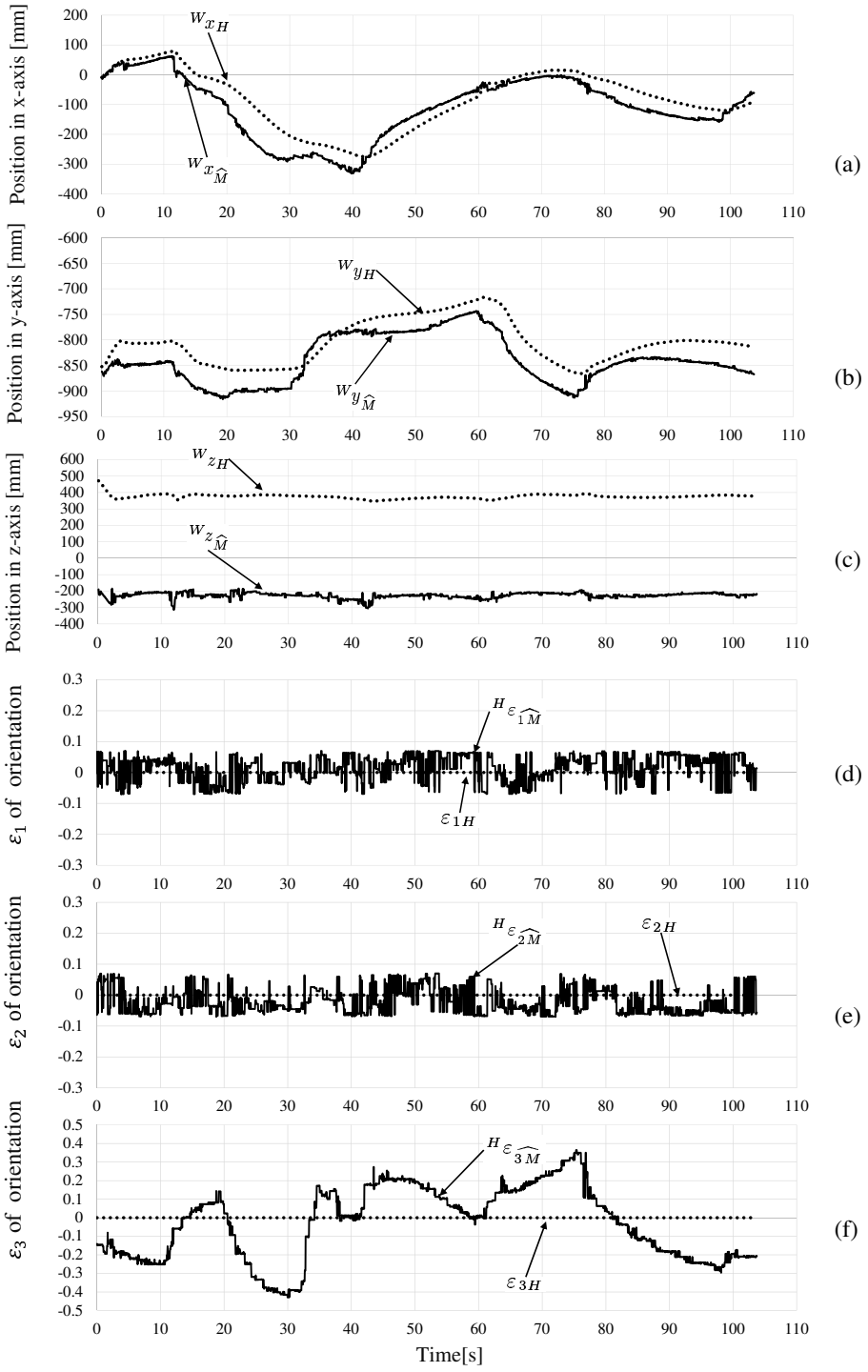


Fig. 6.8: Robot recognition and visual servoing results. $^{W}x_{H}$, $^{W}y_{H}$, and $^{W}z_{H}$ in (a), (b), and (c) are the position tracking results of the end-effector. ε_{1H},ε_{2H}, and ε_{3H} in (d), (e), and (f) are the relative orientation of the end-effector to its initial status calculated by Eq. (5.14). Because in this experiment end-effector's orientation dose not change, they are all 0. $^{W}x_{\widehat{M}}$, $^{W}y_{\widehat{M}}$, and $^{W}z_{\widehat{M}}$ are position recognition results of RM-GA. They are all based on Σ_W and calculated by Eq. (5.5). $^{H}\varepsilon_{\widehat{1M}}$, $^{H}\varepsilon_{\widehat{2M}}$, and $^{H}\varepsilon_{\widehat{3M}}$ are orientation recognition results of RM-GA based on Σ_H.

the end-effector could track the target continually. In the end, the marker pen was released and hit the squid.

According to the results, it can be seen that the photo-model-based pose estimation method is not susceptible to partial occlusion conditions. It can detect the object's pose and control the robot to track it continually, even though there were waves and light reflection on the water. It is confirmed that the system has an ability to conduct a visual servoing task for a moving target and has a robustness against external disturbances.

It is verified that the proposed pose estimation method can make the robot's hand-eye track the designated 3D target object by using its 2D photo-model. This means that the 3D target's pose can be estimated in real-time by 2D photo-model.

Chapter 7

Conclusion

This thesis proposed a real-time 6DOF photo-model-based pose estimation method for the purpose of 6DOF visual servoing. First, as a basic reference study, this paper introduced a model-based eye-vergence visual servoing system. Stereo vision geometry, 3D model-based matching, and RM-GA were explained in detail. Second, the experimental results of lateral and arc swing motion tracking proved that the system could detect the 6DOF of a target object and then continuously track it.

With reference to the above technology, the paper proposed a real-time 6DOF photo-model-based pose estimation method. It then introduced the generation and 3D matching of a photo-model. The fitness function was designed to convert the object recognition problem into an optimization problem. RM-GA was used as a solution to the optimization problem to ensure object detection in real-time. According to the results of the fitness distribution and real-time 3D pose visual servoing experiments, the full pose of a 3D target object was successfully estimated in real-time using only a 2D photo, thus enabling 3D visual servoing of the target. The above results were confirmed by real experiments using a 6DOF manipulator with stereo vision at the end-effector.

Because the eye-vergence vision system has better tracking ability than the fixed camera vision, the proposed photo-model-based 6DOF recognition method will be used in the eye-vergence vision system in future works.

Acknowledgement

I would like to extend my sincere gratitude to my supervisor, Professor Mamoru Minami. With his instruction and encouragement, I can get well along with this research. I have learned many things from Prof. Minami such as research direction and methods. These will be very helpful for my future. Thanks to the guidance and help of the professor, I can participate in many academic conferences. These have broadened my horizons. Then, I would like to thank Associate Professor Takayuki Matsuno and Assistant Yuuichirou Toda for their many suggestions for my research. Moreover, I wish to express my gratitude to all the people in the Visual Servo Group, who gave me valuable advice and helped me in experiments. And special thanks should go to my friends who have put considerable time and effort into their comments on the draft. Finally, I am indebted to my parents for their continuous support and encouragement.

January 2020

Hongzhi Tian

Bibliography

[1] “Model-Based Recognition in Robot Vision,” Roland T Chin, Charles R Dyer, ACM Computing Surveys (CSUR), Vol. 18, No. 1, pp. 67–108 (1986)

[2] “Robot vision,” Berthold Horn, Berthold Klaus, Paul Horn, MIT press (1986)

[3] “Subspace Methods for Robot Vision,” Shree K Nayar, Sameer A Nene, Hiroshi Murase, IEEE Transactions on Robotics and Automation, Vol. 12, No. 5, pp. 750–758 (1996)

[4] “Kalman Filter for Robot Vision: A Survey,” SY Chen, IEEE Transactions on Industrial Electronics, Vol. 59, No. 11, pp. 4409–4420 (2011)

[5] “Robot Vision vs Computer Vision: What’s the Difference?” Alex Owen-Hill, Robotics Tomorrow (2016)

[6] “A Tutorial on Visual Servo Control,” Seth Hutchinson, Gregory D Hager, Peter I Corke, IEEE transactions on robotics and automation, Vol. 12, No. 5, pp. 651–670 (1996)

[7] “Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System,” Peter K. Allen, Aleksandar Timcenko, Billibon H. Yoshimi, Paul Michelman, IEEE Trans. Robotics and Automation, Vol. 9, No. 2, pp. 152–165, DOI:10.1109/70.238279 (1993)

[8] “Visual Servoing by Partitioning Degrees of Freedom,” Paul Y. Oh, Peter K. Allen, IEEE Trans. Robotics and Automation, Vol. 17, No. 1, pp. 1–17, DOI:10.1109/70.917078 (2001)

[9] "Visual Servo Control. II. Advanced Approaches [Tutorial]," François Chaumette, Seth Hutchinson, IEEE Robotics & Automation Magazine, Vol. 14, No. 1, pp. 109–118 (2007)

[10] "2 1/2 D Visual Servoing," Ezio Malis, Francois Chaumette, Sylvie Boudet, IEEE Transactions on Robotics and Automation, Vol. 15, No. 2, pp. 238–250 (1999)

[11] "Visual Servoing by Partitioning Degrees of Freedom," Paul Y Oh, K Allen, IEEE Transactions on Robotics and Automation, Vol. 17, No. 1, pp. 1–17 (2001)

[12] "Real-Time Object Pose Recognition and Tracking With an Imprecisely Calibrated Moving RGB-D Camera," Karl Pauwels, Vladimir Ivan, Eduardo Ros, Sethu Vijayakumar, in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 2733–2740 (2014)

[13] "Dual ARM Manipulation—A Survey," Christian Smith, Yiannis Karayiannidis, Lazaros Nalpantidis, Xavi Gratal, Peng Qi, Dimos V Dimarogonas, Danica Kragic, Robotics and Autonomous systems, Vol. 60, No. 10, pp. 1340–1353 (2012)

[14] "Analysis of CAD Model-based Visual Tracking for Microassembly using a New Block Set for MATLAB/Simulink," Andrey V. Kudryavtsev, Guillaume J. Laurent, Cédric Clévy, Brahim Tamadazte, Philippe Lutz, International Journal of Optomechatronics, Vol. 9, No. 4, pp. 295–309, ISSN 1559-9612, DOI:10.1080/15599612.2015.1059532 (2015)

[15] "On-Line Evolutionary Head Pose Measurement by Feedforward Stereo Model Matching," Wei Song, Mamoru Minami, Yasushi Mae, Seiji Aoyagi, in Proceedings 2007 IEEE International Conference on Robotics and Automation, IEEE, pp. 4394–4400 (2007)

[16] "Visual Servoing to Fish and Catching Using Global/Local GA Search," Mamoru Minami, Hidekazu Suzuki, Julien Agbanhan, Toshiyuki Asakura, in 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings (Cat. No. 01TH8556), IEEE, Vol. 1, pp. 183–188 (2001)

[17] "Model-Referenced Pose Estimation Using Monocular Vision for Autonomous Intervention Tasks," Jisung Park, Taeyun Kim, Jinwhan Kim, Autonomous Robots, ISSN 0929-5593, DOI:10.1007/s10514-019-09886-9 (2019)

[18] "A Comparison of Monocular and Stereo Visual Fastslam Implementations," Riccardo Giubilato, Marco Pertile, Stefano Debei, in 2016 IEEE Metrology for Aerospace (MetroAeroSpace), IEEE, June, ISBN 978-1-4673-8292-2, pp. 227–232, DOI:10.1109/MetroAeroSpace.2016.7573217 (2016)

[19] "Visual Docking Against Bubble Noise With 3-D Perception Using Dual-Eye Cameras," Khin Nwe Lwin, Naoki Mukada, Myo Myint, Daiki Yamada, Akira Yanou, Takayuki Matsuno, Kazuhiro Saitou, Waichiro Godou, Tatsuya Sakamoto, Mamoru Minami, IEEE Journal of Oceanic Engineering, ISSN 0364-9059, DOI:10.1109/JOE.2018.2871651 (2018)

[20] "Cartman: The Low-Cost Cartesian Manipulator That Won the Amazon Robotics Challenge," Douglas Morrison, Adam W Tow, M McTaggart, R Smith, N Kelly-Boxall, S Wade-McCue, J Erskine, R Grinover, A Gurman, T Hunn, et al., in 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 7757–7764 (2018)

[21] "Robotic Pick-and-Place of Novel Objects in Clutter with Multi-Affordance Grasping and Cross-Domain Image Matching," Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R. Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, Nima Fazeli, Ferran Alet, Nikhil Chavan Dafle, Rachel Holladay, Isabella Morona, Prem Qu Nair, Druck Green, Ian Taylor, Weber Liu, Thomas Funkhouser, Alberto Rodriguez, in 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 3750–3757, DOI:10.1109/ICRA.2018.8461044 (2018)

[22] "Fast Object Learning and Dual-Arm Coordination for Cluttered Stowing, Picking, and Packing," Max Schwarz, Christian Lenz, Germán Martín García, Seongyong Koo,

Arul Selvam Periyasamy, Michael Schreiber, Sven Behnke, in 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 3347–3354 (2018)

[23] "Recovering Missing Depth Information From Microsoft's Kinect," Abdul Dakkak, Ammar Husain, in Proc. Embedded Vis. Alliance, pp. 1–9 (2012)

[24] Achuta Kadambi, Ayush Bhandari, Ramesh Raskar, "3d Depth Cameras in Vision: Benefits and Limitations of the Hardware," in Computer Vision and Machine Learning with RGB-D Sensors, Springer, pp. 3–26 (2014)

[25] "Using Near-Field Stereo Vision for Robotic Grasping in Cluttered Environments," Adam Leeper, Kaijen Hsiao, Eric Chu, J Kenneth Salisbury, in Experimental Robotics, Springer, pp. 253–267 (2014)

[26] "Robust Translational/Rotational Eye-Vergence Visual Servoing under Illumination Varieties," Hongzhi Tian, Yejun Kou, Khaing Win Phyu, Daiki Yamada, Mamoru Minami, in 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, pp. 2032–2037 (2017)

[27] "A CAD Model Based Tracking System for Visually Guided Microassembly," Kemal Berk Yesin, Bradley J. Nelson, Robotica, Vol. 23, No. 4, pp. 409–418, ISSN 0263-5747, DOI:10.1017/S0263574704000840 (2005)

[28] "A QR-Code Localization System for Mobile Robots: Application to Smart Wheelchairs," Luca Cavanini, Gionata Cimini, Francesco Ferracuti, Alessandro Freddi, Gianluca Ippoliti, Andrea Monteriù, Federica Verdini, in 2017 European Conference on Mobile Robots (ECMR), IEEE, pp. 1–6 (2017)

[29] "Recognition and Handling of Clothes with Different Pattern by Dual Hand-Eyes Robotic System," Ryuki Funakubo, Khaing Win Phyu, Hongzhi Tian, Mamoru Minami, in 2016 IEEE/SICE International Symposium on System Integration, SII 2016, Sapporo, Japan, December 13-15, 2016, pp. 742–747, DOI:10.1109/SII.2016.7844088 (2016)

[30] "Verification of Recognition Performance of Cloth Handling Robot with Photo-Model-Based Matching," Khaing Win Phyu, Ryuki Funakubo, Ikegawa Fumiya, Yasutake Shinichiro, Mamoru Minami, in 2017 IEEE International Conference on Mechatronics and Automation (ICMA), pp. 1750–1756, DOI:10.1109/ICMA.2017.8016082 (2017)

[31] "Verification of Illumination Tolerance for Photo-Model-Based Cloth Recognition," Khaing Win Phyu, Ryuki Funakubo, Ryota Hagiwara, Hongzhi Tian, Mamoru Minami, Artificial Life and Robotics, Vol. 23, No. 1, pp. 118–130 (2018)

[32] "Verification of Illumination Tolerance for Clothes Recognition," Ryuki Funakubo, Khaing Win Phyu, Ryota Hagiwara, Hongzhi Tian, Mamoru Minami, in The Twenty-Second International Symposium on artificial life and robotics (AROB), pp. 19–21 (2017)

[33] "Verification of Photo-Model-Based Pose Estimation and Handling of Unique Clothes under Illumination Varieties," Khaing Win Phyu, Ryuki Funakubo, Ryota Hagiwara, Hongzhi Tian, Mamoru Minami, Journal of Advanced Mechanical Design, Systems, and Manufacturing, Vol. 12, No. 2, DOI:10.1299/jamdsm.2018jamdsm0047 (2018)

[34] "Frequency Response Experiments of Eye-Vergence Visual Servoing in Lateral Motion with 3D Evolutionary Pose Tracking," Hongzhi Tian, Yu Cui, Mamoru Minami, Akira Yanou, Artificial Life and Robotics, Vol. 22, No. 1, pp. 36–43 (2017)

[35] "Computer Vision: Evolution and Promise," Thomas Huang, Cern (1996)

[36] "Concise Computer Vision," Reinhard Klette, Springer (2014)

[37] "Machine Vision Handbook," Bruce G Batchelor, Springer (2012)

[38] "Key Technology for Automation Solutions. Machine Vision 2017/18. Applications Products Suppliers[Online]," VDMA Verlag GmbH, http://www.vdma.com/vision (2018)

[39] "Active Vision in Robotic Systems: A Survey of Recent Developments," Shengyong Chen, Youfu Li, Ngai Ming Kwok, International Journal of Robotics Research, Vol. 30, No. 11, pp. 1343–1377 (2011)

[40] "Visual Tracking Decomposition," Junseok Kwon, Kyoung Mu Lee, in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, pp. 1269–1276 (2010)

[41] "Object Tracking: A Survey," Alper Yilmaz, Omar Javed, Mubarak Shah, Acm computing surveys (CSUR), Vol. 38, No. 4, pp. 13 (2006)

[42] "A Survey on Moving Object Detection and Tracking in Video Surveillance System," Kinjal A Joshi, Darshak G Thakore, International Journal of Soft Computing and Engineering, Vol. 2, No. 3, pp. 44–48 (2012)

[43] "A Survey of Appearance Models in Visual Object Tracking," Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, Anton Van Den Hengel, ACM transactions on Intelligent Systems and Technology (TIST), Vol. 4, No. 4, pp. 58 (2013)

[44] "Coupled Object Detection and Tracking From Static Cameras and Moving Vehicles," Bastian Leibe, Konrad Schindler, Nico Cornelis, Luc Van Gool, IEEE transactions on pattern analysis and machine intelligence, Vol. 30, No. 10, pp. 1683–1698 (2008)

[45] "Incremental Learning for Robust Visual Tracking," David A Ross, Jongwoo Lim, Ruei-Sung Lin, Ming-Hsuan Yang, International journal of computer vision, Vol. 77, No. 1-3, pp. 125–141 (2008)

[46] "Highly Precise Micropositioning Task Using a Direct Visual Servoing Scheme," Brahim Tamadazte, Guillaume Duceux, N Le-Fort Piat, Eric Marchand, in 2011 IEEE International Conference on Robotics and Automation, IEEE, pp. 5689–5694 (2011)

[47] “A Kinect-Based Real-Time Compressive Tracking Prototype System for Amphibious Spherical Robots,” Shaowu Pan, Liwei Shi, Shuxiang Guo, Sensors, Vol. 15, No. 4, pp. 8232–8252 (2015)

[48] “Handbook of Robotics, Cap. 24 (Visual Servoing and Visual Tracking),” François Chaumette, Seth Hutchinson, Springer-Verlag Berlin Heidelberg (2008)

[49] “Eye-Vergence Visual Servoing Enhancing Lyapunov-Stable Trackability,” Fujia Yu, Mamoru Minami, Wei Song, Akira Yanou, Artificial Life and Robotics, Vol. 18, No. 1-2, pp. 27–35 (2013)

[50] “Stability/Precision Improvement of 6-Dof Visual Servoing by Motion Feedforward Compensation and Experimental Evaluation,” Wei Song, Mamoru Minami, in 2009 IEEE International Conference on Robotics and Automation, IEEE, pp. 722–729 (2009)

[51] “Visual Servoing,” François Chaumette, Seth Hutchinson, Peter Corke, in B. Siciliano, O. Khatib, editors, Handbook of Robotics, 2nd edition, Springer, pp. 841–866 (2016)

[52] “3D Visual Servoing by Feedforward Evolutionary Recognition,” Wei Song, Yu Fujia, Mamoru Minami, Journal of Advanced Mechanical Design, Systems, and Manufacturing, Vol. 4, No. 4, pp. 739–755 (2010)

[53] “Hierarchical Featureless Tracking for Position-Based 6-Dof Visual Servoing,” Wolfgang Sepp, Stefan Fuchs, Gerd Hirzinger, in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 4310–4315 (2006)

[54] “Position Based Visual Servoing control of a Wheelchair Mounter Robotic Arm using Parallel Tracking and Mapping of task objects.” Alessandro Palla, Gabriele Meoni, Luca Fanucci, Alessandro Frigerio, ICST Trans. Ambient Systems, Vol. 4, No. 13 (2017)

[55] “Stability Analysis of Pose-Based Visual Servoing Control of Cable-Driven Parallel Robots,” Zane Zake, Stéphane Caro, Adolfo Suarez Roos, François Chaumette, Nicolò

Pedemonte, in International Conference on Cable-Driven Parallel Robots, Springer, pp. 73–84 (2019)

[56] "Image-Based Path Following Control of Mobile Robots with Central Catadioptric Cameras," Toshifumi Hiramatsu, Takanori Fukao, Keita Kurashiki, Koichi Osuka, in 2009 IEEE International Conference on Robotics and Automation, IEEE, pp. 4045–4050 (2009)

[57] "Image-Based Visual Servoing for Nonholonomic Mobile Robots Using Epipolar Geometry," Gian Luca Mariottini, Giuseppe Oriolo, Domenico Prattichizzo, IEEE Transactions on Robotics, Vol. 23, No. 1, pp. 87–100 (2007)

[58] "Image Based Visual Servo Application on Video Tracking with Monocular Camera Based on Phase Correlation Method," Yoshi Ri, Hiroshi Fujimoto, in The 3rd IEEJ international workshop on Sensing, Actuation, Motion Control, and Optimization (2017)

[59] "Direct image-based visual servoing of free-floating space manipulators," Javier Pérez Alepuz, M Reza Emami, Jorge Pomares, Aerospace Science and Technology, Vol. 55, pp. 1–9 (2016)

[60] "Eye-In-Hand Stereo Visual Servoing of an Assistive Robot Arm in Unstructured Environments," Dae-Jin Kim, Ryan Lovelett, Aman Behal, in 2009 IEEE International Conference on Robotics and Automation, IEEE, pp. 2326–2331 (2009)

[61] "A Comparison between Position-Based and Image-Based Dynamic Visual Servoings in the Control of a Translating Parallel Manipulator," Giacomo Palmieri, Matteo Palpacelli, Massimiliano Battistelli, Massimo Callegari, Journal of Robotics, Vol. 2012 (2012)

[62] "Multiple Camera Model-Based 3-D Visual Servo," Jay Stavnitzky, David Capson, IEEE transactions on robotics and automation, Vol. 16, No. 6, pp. 732–739 (2000)

[63] “A real-time Model Based Visual Servoing application for a differential drive mobile robot using Beaglebone Black embedded system,” Indrazno Siradjuddin, Subali P Tundung, Agustien S Indah, Supriyatna Adhisuwignjo, in 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), IEEE, pp. 186–192 (2015)

[64] “Feature Tracking for Visual Servoing Purposes,” Éric Marchand, François Chaumette, Robotics and Autonomous Systems, Vol. 52, No. 1, pp. 53–70 (2005)

[65] “A single 3-D feature-based visual servoing,” Christophe Doignon, in 2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR), IEEE, pp. 41–46 (2015)

[66] “Adaptive visual servoing of contour features,” Hesheng Wang, Bohan Yang, Jingchuan Wang, Xinwu Liang, Weidong Chen, Yun-Hui Liu, IEEE/ASME Transactions on Mechatronics, Vol. 23, No. 2, pp. 811–822 (2018)

[67] “Eye-in-hand tracking control of a free-floating space manipulator,” Hesheng Wang, Dejun Guo, Hao Xu, Weidong Chen, Tao Liu, Kam K Leang, IEEE Transactions on Aerospace and Electronic Systems, Vol. 53, No. 4, pp. 1855–1865 (2017)

[68] “Vision-Based Control and Stability Analysis of a Cable-Driven Parallel Robot,” Zane Zake, François Chaumette, Nicolò Pedemonte, Stéphane Caro, IEEE Robotics and Automation Letters, Vol. 4, No. 2, pp. 1029–1036 (2019)

[69] “Model-Based Head Pose Tracking with Stereovision,” Ruigang Yang, Zhengyou Zhang, in Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition, IEEE, pp. 255–260 (2002)

[70] “Pose Estimation from a Line Drawing Using Genetic Algorithm,” Fubito Toyama, Kenji Shoji, Juichi Miyamichi, Transactions of the Institute of Electronics, Information, and Communication Engineers D-II (Japan), Vol. J81D-II, No. 7, pp. 1584–1590 (1998)

[71] "Smooth Matching of Feature and Recovery of Epipolar Equation by Tabu Search," Y Maeda, G Xu, Vol. J83-D-2, No. 3, pp. 440–448 (1999)

[72] "Image Matching Using Structural Similarity and Geometric Constraint Approaches on Remote Sensing Images," Jian-hua Guo, Fan Yang, Hai Tan, Jing-xue Wang, Zhi-heng Liu, Journal of Applied Remote Sensing, Vol. 10, No. 4, pp. 045007 (2016)

[73] "Survey on Visual Servoing for Manipulation," Danica Kragic, Henrik I Christensen, Computational Vision and Active Perception Laboratory, Fiskartorpsv, Vol. 15 (2002)

[74] "Orthogonal Image Features for Visual Servoing of a 6-Dof Manipulator With Uncalibrated Stereo Cameras," Caixia Cai, Nikhil Somani, Alois Knoll, IEEE transactions on Robotics, Vol. 32, No. 2, pp. 452–461 (2016)

[75] "Implementation of a Stereo Vision Based System for Visual Feedback Control of Robotic ARM for Space Manipulations," GR Sangeetha, Nishank Kumar, PR Hari, S Sasikumar, Procedia computer science, Vol. 133, pp. 1066–1073 (2018)

[76] "One Click Focus with Eye-In-Hand/Eye-To-Hand Cooperation," Claire Dune, Eric Marchand, Christophe Leroux, in Proceedings 2007 IEEE International Conference on Robotics and Automation, IEEE, pp. 2471–2476 (2007)

[77] "3D Move to See: Multi-Perspective Visual Servoing for Improving Object Views With Semantic Segmentation," Chris Lehnert, Dorian Tsai, Anders Eriksson, Chris McCool, arXiv preprint arXiv:1809.07896 (2018)

[78] "IR Stereo Kinect: Improving Depth Images by Combining Structured Light with IR Stereo," Faraj Alhwarin, Alexander Ferrein, Ingrid Scholl, in Pacific Rim International Conference on Artificial Intelligence, Springer, pp. 409–421 (2014)

[79] "A Modular Framework for Model-Based Visual Tracking Using Edge, Texture and Depth Features," Souriya Trinh, Fabien Spindler, Eric Marchand, François Chaumette,

in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 89–96 (2018)

[80] “Multi-View Stereo: A Tutorial,” Yasutaka Furukawa, Carlos Hernández, et al., Foundations and Trends in Computer Graphics and Vision, Vol. 9, No. 1-2, pp. 1–148 (2015)

[81] “Multiple View Geometry in Computer Vision,” Richard Hartley, Andrew Zisserman (2003)

[82] “Tutorial: Overview of Stereo Matching Research,” RA Lane, NA Thacker, Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester (1998)

[83] “6D Image-Based Visual Servoing for Robot Manipulators With Uncalibrated Stereo Cameras,” Caixia Cai, Emmanuel Dean-León, Nikhil Somani, Alois Knoll, in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 736–742 (2014)

[84] “Computer Vision: Technical Review and Future View,” T Matsuyama, Y Kuno, A Imiya, New Technology Communications (in Japanese) (1998)

[85] “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms,” Daniel Scharstein, Richard Szeliski, International journal of computer vision, Vol. 47, No. 1-3, pp. 7–42 (2002)

[86] “Shape and the Stereo Correspondence Problem,” Abhijit S Ogale, Yiannis Aloimonos, International Journal of Computer Vision, Vol. 65, No. 3, pp. 147–162 (2005)

[87] “Accurate Multi-View Reconstruction Using Robust Binocular Stereo and Surface Meshing,” Derek Bradley, Tamy Boubekeur, Wolfgang Heidrich, in 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp. 1–8 (2008)

[88] "Interaction Control of Robot Manipulators: Six Degrees-Of-Freedom Tasks," Ciro Natale, Springer Science & Business Media, Vol. 3 (2003)

[89] "Robot Force Control," B. Siciliano, L. Villani, Springer US, Kluwer International Series in Engineering and Computer Science: Robotics: Vision, Manipulation, and Sensors, ISBN 9780792377337 (1999)

[90] "A Survey on the Computation of Quaternions from Rotation Matrices," Soheil Sarabandi, Federico Thomas, Journal of Mechanisms and Robotics, Vol. 11, No. 2, ISSN 19424310, DOI:10.1115/1.4041889 (2019)

[91] "Accurate Computation of Quaternions from Rotation Matrices," Soheil Sarabandi, Federico Thomas, in International Symposium on Advances in Robot Kinematics, Springer, pp. 39–46 (2018)

[92] "Matrix Animation and Polar Decomposition," Ken Shoemake, Tom Duff, in Proceedings of the conference on Graphics interface, Citeseer, Vol. 92, pp. 258–264 (1992)

[93] "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors," James Diebel, Matrix, Vol. 58, No. 15-16, pp. 1–35 (2006)

[94] "Hand-Eye Motion-Invariant Pose Estimation with Online 1-Step GA-3D Pose Tracking Accuracy Evaluation in Dynamic Hand-Eye Oscillation," Mamoru Minami, Wei Song, Journal of Robotics and Mechatronics, Vol. 21, No. 6, pp. 709–719 (2009)

[95] "3-D Hand & Eye-Vergence Approaching Visual Servoing with Lyapunouv-Stable Pose Tracking," Wei Song, Mamoru Minami, Fujia Yu, Yanan Zhang, Akira Yanou, in 2011 IEEE International Conference on Robotics and Automation, IEEE, pp. 5210–5217 (2011)

[96] "Hand & Eye-Vergence Dual Visual Servoing to Enhance Observability and Stability,"

Wei Song, Mamoru Minami, in 2009 IEEE International Conference on Robotics and Automation, IEEE, pp. 714–721 (2009)

[97] "Visual Servoing to Catch Fish Using Global/Local GA Search," Hidekazu Suzuki, Mamoru Minami, IEEE/ASME Transactions on Mechatronics, Vol. 10, No. 3, pp. 352–357 (2005)

[98] "Derivative of Rotation Matrix Direct Matrix Derivation of Well Known Formula," Fumio Hamano, arXiv preprint arXiv:1311.6010 (2013)

[99] "Feedforward On-Line Pose Evolutionary Recognition Based on Quaternion," W Song, M Minami, S Aoyagi, Journal of the Robot Society of Japan, Vol. 28, No. 1, pp. 55–64 (2010)

[100] "Evolutionary Scene Recognition and Simultaneous Position/Orientation Detection," Mamoru Minami, Julien Agbanhan, Toshiyuki Asakura, in Soft Computing in Measurement and Information Acquisition, Springer, pp. 178–207 (2003)

[101] "Visual Servoing for Underwater Vehicle Using Dual-Eyes Evolutionary Real-Time Pose Tracking." Myo Myint, Kenta Yonemori, Akira Yanou, Khin Nwe Lwin, Mamoru Minami, Shintaro Ishiyama, JRM, Vol. 28, No. 4, pp. 543–558 (2016)

[102] "Dual-Eyes Vision-Based Docking System for Autonomous Underwater Vehicle: An Approach and Experiments," Myo Myint, Kenta Yonemori, Khin Nwe Lwin, Akira Yanou, Mamoru Minami, Journal of Intelligent & Robotic Systems, Vol. 92, No. 1, pp. 159–186 (2018)