

High Speed/Accuracy Visual Servoing based on Virtual Visual Servoing with Stereo Cameras

Takashi Nammoto, Koichi Hashimoto, Shingo Kagami and Kazuhiro Kosuge

Abstract—This paper presents high speed and high accuracy visual servoing system. The algorithm has three major improvements, which can be implemented in practical applications; Firstly high-accuracy pose estimation by using stereo cameras, secondly real time implementation issues with non-real-time image processing platform and thirdly a consideration for industrial position controller. To resolve the issues, position-based visual servoing (PBVS) is adopted and appearance model based virtual visual servoing (VVS) is applied for pose estimation. VVS approach does not compute the stereo matching but directly compares the OpenGL rendered image and camera image for each camera; estimate the position/orientation using VVS independently for each camera; and provides a theoretically optimal compromise among those estimates. To enhance estimation accuracy, a hybrid method of stereo trigonometry for position estimation and weighted least squares for orientation estimation is proposed to combine the information from the stereo cameras. Operation speed is increased by using graphic processing unit (GPU) acceleration and an on-line trajectory generator which can accommodate the variable cycle of the image processing and the fixed cycle of a common robot controller. Finally, some experimental results illustrate the effectiveness of the proposed framework.

I. INTRODUCTION

Product assembly has growing demand in consumer electronics. Because the lifetime of these products are very short, modification of the production line is frequent. Robots are the most promising solutions to make the line flexible and worker-friendly. Robot teaching is to store the desired positions in memory of the controller. This requires well-trained operator to achieve high repeated accuracy of the industrial robots. Therefore, to make robot teaching simple is the key to realize the flexible production system. For example, robot teaching is simplified by acquiring human skills and transferring these data to robotic systems [1]. A vision-guided control, in which the manipulator moves adaptively to its environment by using visual information, is another candidate of enabling method for the flexible production system. For example, when a visual servoing is applied as the vision-guided control, robot teaching is realized by preparing a reference image in which an object is located in certain position [2]. Therefore this paper presents a high speed/accuracy appearance model based visual servoing with stereo cameras for practical applications to realize the flexible production system.

T. Nammoto and K. Kosuge are with Department of Bioengineering and Robotics, K. Hashimoto and S. Kagami are with Graduate School of Information Sciences, Tohoku University, 6-6-01, Aoba, Aramaki, Aoba-ku, Sendai 980-8579, Japan {nammoto, kosuge}@irs.mech.tohoku.ac.jp, {koichi, swk}@ic.is.tohoku.ac.jp

II. RELATED WORKS

Hager proposed a pioneering work of stereo visual servoing without computing stereo matching [3]. The method is robust against camera positioning and robot kinematics. However, the task definition is feature dependent which is not desirable for industrial application because the feature selection is not trivial for the given tasks. In this sense, the authors prefer a appearance model based approach. It is much more intuitive for the user to define what should be controlled.

The real-time object pose estimation is one of the key components of the proposed control. The model-based approaches have been discussed widely in the past for the real-time object pose estimation. The approaches are classified according to the estimation technique and the feature selection. Error minimization by standard non-linear method is applied to edges in a camera image of an object and a model image of the object model in [4] and [5]. To enhance stability against degenerate point data sets, the singular value decomposition (SVD) is utilized to minimize errors between edges [9]. The extended Kalman filter (EKF) is applied to calculate the object pose from measurements of edges [6], texture in reference area [7], the end points of straight lines [8]. The virtual visual servoing (VVS) based on the visual servoing algorithm is applied to error minimization between the feature points, such as edges is presented in [10] and [11]. Features which are extracted from object model, such as edges, and model-free cues are integrated in the iterated extended Kalman filter (IEKF) to obtain unobservable rotations of spherical objects [12]. Appearance learning by hand-segmented images is applied to articulated object tracking [13]. Recently, the particle filter approach using feature points, such as edges, is proposed to realize robust estimation in [14], [15] and [16]. Robust photometric and mutual information approaches are also presented in [17] and [18].

Despite of a lot of research efforts, visual servoing systems that can be used for practical applications such as industrial use have seldom been realized in the past. Therefore, compared to the related works, in this paper, we focus on issues listed below.

- Enhancement of estimation accuracy of position in optical axis of camera and orientation using stereo cameras.
- Parallelism of estimation algorithm and interoperability between model image generation and GPU calculation to increase frame rates of image processing.
- Accommodation of variable cycle of image processing

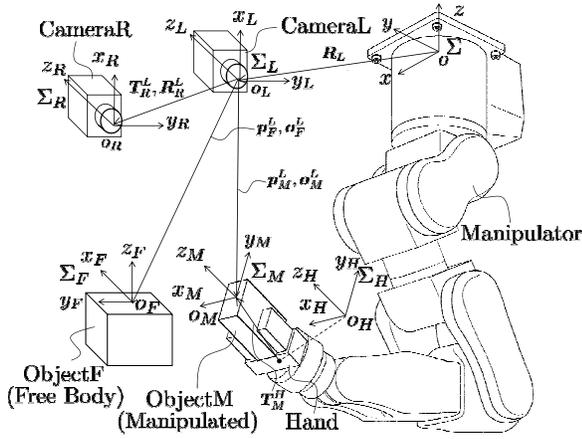


Fig. 1. System Configuration

and fixed cycle of a common robot controller.

The rest of the paper is organized as follows. In Section III, implementation of visual servoing with stereo cameras is described including proposals of GPU accelerated pose estimation and variable cycle on-line trajectory generator. Two experimental results regarding positioning accuracy and tracking speed are shown in Section IV. Finally, we conclude the paper in Section V.

III. POSITION-BASED VISUAL SERVOING WITH STEREO CAMERAS USING APPEARANCE MODEL BASED VIRTUAL VISUAL SERVOING

A. Visual Servoing Framework

The system configuration is depicted in Fig. 1. The system is composed of a manipulator, stereo cameras (CameraR and CameraL), a hand and two objects (ObjectM and ObjectF). The ObjectM is grasped by the hand and moved by the manipulator. While the ObjectF moves freely, the ObjectM is visually controlled so as to be in same relative pose depicted in a reference image. Σ , Σ_L , Σ_R , Σ_H , Σ_M and Σ_F are attached to a based of the manipulator, the CameraL, the CameraR, the hand, the ObjectM and the ObjectF, respectively. R_L is the rotation matrix which transforms a vector in Σ to that in Σ_L . T_R^L is the translation vector from the origin of Σ_L to that of Σ_R ; R_R^L is the rotation matrix which transforms a vector in Σ_L to that in Σ_R . T_M^H is the translation vector from the origin of Σ_H to that of Σ_M . $p_{M/F}^L$ is the position of the origin of $\Sigma_{M/F}$ in Σ_L ; $o_{M/F}^L$ is the orientation of $\Sigma_{M/F}$ in Σ_L .

The block diagram of position-based visual servoing (PBVS) is shown in Fig. 2. Considering compatibility to a common robot controller, we choose PBVS in our system. Therefore a variable cycle on-line trajectory generator is needed to accommodate the differences between the image processing cycle and the position control cycle. Speed and accuracy of object pose estimation are keys to enhance PBVS

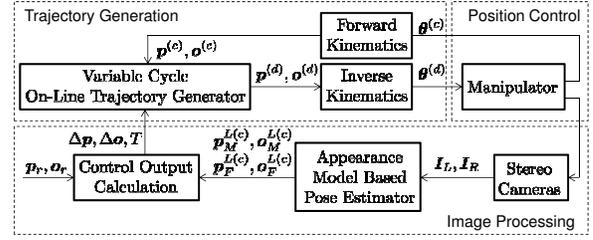


Fig. 2. Visual Servoing Block Diagram

performance. Control raw of PBVS is given as follows:

$$\begin{bmatrix} \Delta p \\ \Delta o \end{bmatrix} = -\lambda {}^t R_L \begin{bmatrix} I_3 & S(T_M^H) \\ \mathbf{0} & I_3 \end{bmatrix} (s - s^*) \quad (1)$$

$$s = {}^t \begin{bmatrix} {}^t p_M^{L(c)} & {}^t o_M^{L(c)} \end{bmatrix} \quad s^* = {}^t \begin{bmatrix} {}^t p_M^{L(g)} & {}^t o_M^{L(g)} \end{bmatrix}$$

$$\begin{bmatrix} p_M^{L(g)} \\ o_M^{L(g)} \end{bmatrix} = \begin{bmatrix} p_F^{L(c)} \\ o_F^{L(c)} \end{bmatrix} + \begin{bmatrix} p_r \\ o_r \end{bmatrix}. \quad (2)$$

The image processing starts with capturing images of the objects by the stereo cameras. The current poses of the objects are estimated from stereo camera images I_L and I_R by the appearance model based pose estimator which is implemented by using VVS. This is explained in Sec. III-B.1 and Sec. III-B.2 in detail. The outputs of the image processing block ${}^t \begin{bmatrix} \Delta p & \Delta o \end{bmatrix}$, which are relative displacements between the current and desired pose of the hand, are calculated from the differences between the current pose ${}^t \begin{bmatrix} {}^t p_M^{L(c)} & {}^t o_M^{L(c)} \end{bmatrix}$ and reference pose ${}^t \begin{bmatrix} {}^t p_M^{L(g)} & {}^t o_M^{L(g)} \end{bmatrix}$ of the ObjectM by multiplying the transpose of the rotation matrix R_L , the skew-symmetric matrix $S(T_M^H)$ and the gain λ . Here, $\mathbf{0}$ is 3x3 zero matrix. The reference pose of the ObjectM is obtained by adding the current pose of the ObjectF ${}^t \begin{bmatrix} {}^t p_F^{L(c)} & {}^t o_F^{L(c)} \end{bmatrix}$ and the relative position between the objects ${}^t \begin{bmatrix} {}^t p_r & {}^t o_r \end{bmatrix}$ which is defined in advance. The average cycle of the image processing T is also an output of the image processing block for the variable cycle on-line trajectory generator to generate trajectories which are divided into first acceleration/deceleration section and second constant velocity section considering the current pose ${}^t \begin{bmatrix} {}^t p^{(c)} & {}^t o^{(c)} \end{bmatrix}$ and the relative displacement ${}^t \begin{bmatrix} \Delta p & \Delta o \end{bmatrix}$ of the manipulator. The variable cycle on-line trajectory generator outputs the desired pose ${}^t \begin{bmatrix} {}^t p^{(d)} & {}^t o^{(d)} \end{bmatrix}$ with each cycles of the position control following the trajectories. After generation of the desired pose, the desired joint angles are calculated by the inverse kinematics and fed to the manipulator.

B. GPU accelerated Appearance Model Based Object Pose Estimation with Stereo Cameras based on Virtual Visual Servoing

In this section, two appearance model based estimation methods by using stereo cameras based on VVS are presented, one is weighted least squares and the other is the hybrid method of weighted least squares for orientation estimation and the stereo trigonometry for position estimation.

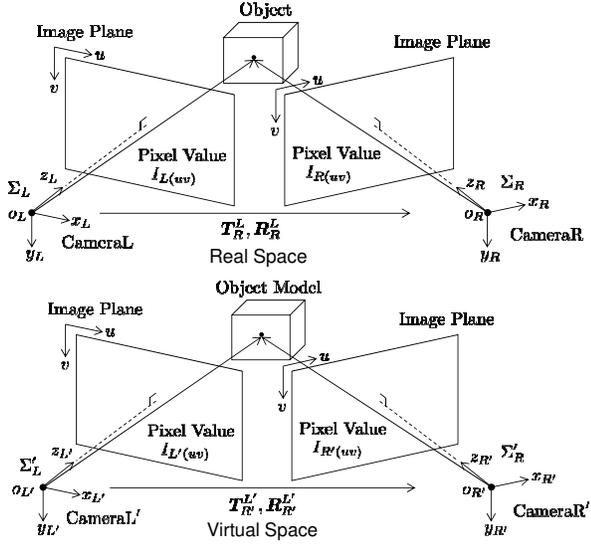


Fig. 3. Configuration of Real Space and Virtual Space of Stereo Cameras

Finally, GPU-accelerated SSD calculation is proposed to shorten the image processing cycle.

1) *Weighted Least Squares*: Configurations of real space and virtual space of stereo cameras are shown in Fig. 3. In the virtual space, stereo cameras and the coordinates frames are described as CameraL', CameraR', $\Sigma_{L'}$ and $\Sigma_{R'}$, respectively. To estimate an object pose in the real space by the appearance model based method, intrinsic parameters and relation between stereo cameras should be reproduced in the virtual space. Commonly-used calibration method, for example image processing software library OpenCV can be used to obtain intrinsic parameters of a camera [19]. Intrinsic parameters are described as follows:

$$\mathbf{A}_c = \begin{bmatrix} fk_u & 0 & u_c \\ 0 & fk_v & v_c \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

When an object model in the virtual space are rendered by using 3D graphics software library OpenGL [20], intrinsic parameters can be set by using glFrustum function. Arguments of glFrustum function are as follows:

$$\begin{aligned} \text{right} &= -\frac{u_c}{fk_u} z\text{Near} & \text{left} &= \frac{W-u_c}{fk_u} z\text{Near} \\ \text{top} &= \frac{H-v_c}{fk_v} z\text{Near} & \text{bottom} &= -\frac{v_c}{fk_v} z\text{Near}. \end{aligned} \quad (4)$$

Here, W and H are width and height of an image. The relative distance T_R^L and the rotation matrix R_R^L between stereo cameras can also be obtained by using OpenCV and these values are assigned to those in the virtual space $T_{R'}^{L'}$ and $R_{R'}^{L'}$. Therefore, the position and orientation of an object model in the virtual space correspond to those of the object in the real space. In the experimental system, relative distance and rotation matrix are measured as follows:

$$\begin{aligned} {}^tT_R^L &= \begin{bmatrix} -270.51 & -150.15 & 125.98 \end{bmatrix}^T \text{ (mm)} \\ R_R^L &= \begin{bmatrix} -0.82 & 0.10 & 0.57 \\ -0.31 & 0.90 & 0.29 \\ -0.48 & -0.41 & 0.77 \end{bmatrix}. \end{aligned}$$

Pose estimation by using weighted least squares based on the steepest descent method is described as follows:

$$\begin{bmatrix} \mathbf{p}_i^{L'}(k) \\ \mathbf{o}_i^{L'}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_i^{L'}(k-1) \\ \mathbf{o}_i^{L'}(k-1) \end{bmatrix} - \eta \nabla_{L'} f \quad (5)$$

$$\nabla_{L'} = {}^t \left[\frac{\partial}{\partial x_{L'}} \quad \frac{\partial}{\partial y_{L'}} \quad \frac{\partial}{\partial z_{L'}} \quad \frac{\partial}{\partial \alpha_{L'}} \quad \frac{\partial}{\partial \beta_{L'}} \quad \frac{\partial}{\partial \gamma_{L'}} \right]$$

$$\begin{aligned} f(\mathbf{p}_i^{L'}, \mathbf{o}_i^{L'}) &= w_L \left| \mathbf{I}_L - \mathbf{I}_{L'}(\mathbf{p}_i^{L'}, \mathbf{o}_i^{L'}) \right|^2 \\ &+ w_R \left| \mathbf{I}_R - \mathbf{I}_{R'}(\mathbf{p}_i^{R'}, \mathbf{o}_i^{R'}) \right|^2, \quad i \in \{M, F\} \end{aligned} \quad (6)$$

$$w_L = \frac{\sigma_R^2}{\sigma_L^2 + \sigma_R^2} \quad w_R = \frac{\sigma_L^2}{\sigma_L^2 + \sigma_R^2} \quad (7)$$

\mathbf{I}_L and \mathbf{I}_R are the object image vector of the CameraL and the CameraR. $\mathbf{I}_{L'}$ and $\mathbf{I}_{R'}$ are the model image vector of the CameraL' and the CameraR'. In this estimation method, the steepest descent method is utilized to minimize the cost function f which indicates difference between the object image and the model image. η is a tuning parameter of step size. k and $k-1$ denotes the current and the former status. The cost function is defined as a weighted summation of sum of squared differences (SSD) between the object image and the corresponding model image of stereo cameras. $\mathbf{p}_i^{R'}$ and $\mathbf{o}_i^{R'}$ in (6) are calculated from $\mathbf{p}_i^{L'}$ and $\mathbf{o}_i^{L'}$ by applying $R_{R'}^{L'}$ and translating $T_{R'}^{L'}$. Weights are calculated by using noise variance of each cameras. Assuming that object velocity is relatively slow compared to frame rate, variance of Gaussian white noise of each cameras can be obtained by calculating SSD of current and previous frames of camera images as follows:

$$|\mathbf{I}_j(k) - \mathbf{I}_j(k-1)|^2 \approx 2WH\sigma_j^2, \quad j \in \{L, R\}. \quad (8)$$

2) *Hybrid Method*: Appearance change when an object translates along z axis, which is parallel to optical axis, is relatively small compared to other axes. This causes estimation accuracy degradation of z axis. Therefore, in the hybrid method, z position is calculated from those of x and y by using trigonometry as described as follows:

$$\begin{bmatrix} x_i^{L'}(k) \\ y_i^{L'}(k) \end{bmatrix} = \begin{bmatrix} x_i^{L'}(k-1) \\ y_i^{L'}(k-1) \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial}{\partial x_{L'}} \\ \frac{\partial}{\partial y_{L'}} \end{bmatrix} g_{L'} \quad (9)$$

$$g_{L'}(\mathbf{p}_i^{L'}, \mathbf{o}_i^{L'}) = \left| \mathbf{I}_L - \mathbf{I}_{L'}(\mathbf{p}_i^{L'}, \mathbf{o}_i^{L'}) \right|^2 \quad (10)$$

$$\begin{bmatrix} x_i^{R'}(k) \\ y_i^{R'}(k) \end{bmatrix} = \begin{bmatrix} x_i^{R'}(k-1) \\ y_i^{R'}(k-1) \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial}{\partial x_{R'}} \\ \frac{\partial}{\partial y_{R'}} \end{bmatrix} g_{R'} \quad (11)$$

$$g_{R'}(\mathbf{p}_i^{R'}, \mathbf{o}_i^{R'}) = \left| \mathbf{I}_R - \mathbf{I}_{R'}(\mathbf{p}_i^{R'}, \mathbf{o}_i^{R'}) \right|^2 \quad (12)$$

$$\begin{aligned} z_i^{L'}(k) &= R_{R(31)}^L x_i^{R'}(k) + R_{R(32)}^L y_i^{R'}(k) \\ &+ R_{R(33)}^L z_i^{R'}(k) + T_{R(3)}^L \end{aligned} \quad (13)$$

$$\begin{aligned} z_i^{R'}(k) &= \left\{ x_i^{L'}(k) + y_i^{L'}(k) - T_{R(1)}^L - T_{R(2)}^L \right. \\ &\left. - \left(R_{R(11)}^L + R_{R(21)}^L \right) x_i^{R'}(k) - \left(R_{R(12)}^L + R_{R(22)}^L \right) y_i^{R'}(k) \right\} \\ &/ \left(R_{R(13)}^L + R_{R(23)}^L \right). \end{aligned} \quad (14)$$

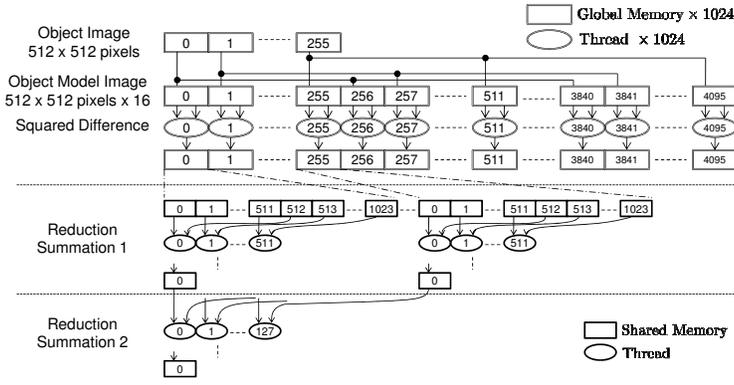


Fig. 4. Framework of GPU-accelerated SSD calculation

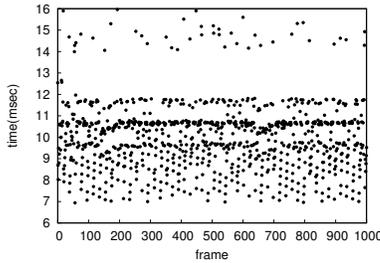


Fig. 5. Image Processing Cycle

Positions of x and y axes in the CameraL and the CameraR are calculated by the steepest descent method. $T_{R(i)}^L$ denotes a i -th row vector element of T_R^L and $R_{R(ij)}^L$ denotes a i -th row and j -th column matrix element of R_R^L , respectively.

3) *GPU-accelerated SSD calculation*: SSD calculation is implemented on GPU because cycle time can be shortened by calculating subtraction and square in parallel with each pixels. The framework of GPU-accelerated SSD calculation is shown in Fig. 4. Object images, that are captured by stereo cameras, are transferred from CPU to global memory on GPU. Sixteen model images are rendered at once and transferred through pixel buffer object (PBO) to global memory on GPU directly. Twelve images are used for calculation of partial differentials of the SSD in (5) as central difference. Another image is used to obtain the noise variance as described above. Other two images are mask images to extract the target objects of pose estimation. The other image is used for evaluation of the difference between the object image, in which the object is located the former position and orientation, and the object image. Squared differences are calculated in parallel with each pixels. Reduction summation is applied twice to sum up the squared differences in consideration of bank conflicts, loop unrolling and synchronization of threads. The calculation described above is applied to each of stereo cameras. CUDA is utilized to implement the framework on GeForce GTX 690. Measurements of the image processing cycle is shown in Fig. 5. The average image processing cycle is shortened to 10 ms by applying the framework.

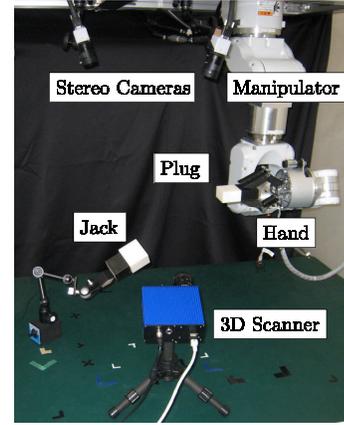


Fig. 6. Experimental System

C. Variable Cycle On-line Trajectory Generator

Cycle time of the image processing and display of the results on a PC monitor could be varied depending on the status of the PC as shown in Fig. 5. Therefore the variable cycle on-line trajectory generator is necessary for accommodation of the variable cycle of the image processing and the fixed cycle of a common robot controller. The on-line trajectory generator for a common robot controller is also discussed in [21].

Assuming the variations from an average cycle are around 50 % with reference to the measurement, a cycle is divided into two equal sections. Then a manipulator is accelerated or decelerated in the first half and moved at constant velocity in the second half. Besides, trajectory is configured so as that velocity changes as trapezoidal shape combined with sine curve for a manipulator to be accelerated or decelerated smoothly. The equation of the variable cycle on-line trajectory generation, which generates the desired pose in Fig. 2 in accordance with position control cycle, can be obtained by setting the boundary conditions of the velocity and position as follows:

$$\begin{cases} \begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix} = \frac{1}{3T + 4\Delta t} \left(4 \begin{bmatrix} \Delta p \\ \Delta o \end{bmatrix} - \begin{bmatrix} v_0 \\ \omega_0 \end{bmatrix} T \right) \\ \begin{bmatrix} p^{(d)}(t) \\ o^{(d)}(t) \end{bmatrix} = \begin{cases} \left[\frac{t^2}{T} - \frac{T}{8\pi^2} \left\{ 1 - \cos\left(\frac{4\pi t}{T}\right) \right\} \right] \cdot \left(\begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix} - \begin{bmatrix} v_0 \\ \omega_0 \end{bmatrix} \right) + \begin{bmatrix} v_0 \\ \omega_0 \end{bmatrix} t + \begin{bmatrix} p_0^{(c)} \\ o_0^{(c)} \end{bmatrix} & (t \leq \frac{T}{2}) \\ \begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix} t - \left(\begin{bmatrix} v_1 \\ \omega_1 \end{bmatrix} - \begin{bmatrix} v_0 \\ \omega_0 \end{bmatrix} \right) \frac{T}{4} + \begin{bmatrix} p_0^{(c)} \\ o_0^{(c)} \end{bmatrix} & (t > \frac{T}{2}) \end{cases} \end{cases} \quad (15)$$

v_1 , ω_1 , v_0 and ω_0 are the current and the former translational and angular velocity, respectively. $p_0^{(c)}$ and $o_0^{(c)}$ are the position and orientation of the hand when an output from the image processing is refreshed. T is an average cycle of the image processing, Δt is a position control cycle. Typically

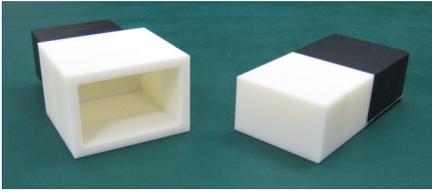


Fig. 7. Object Image of Plug and Jack

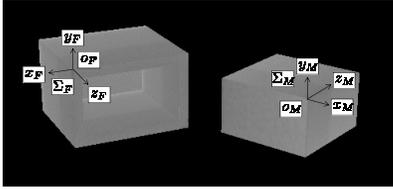
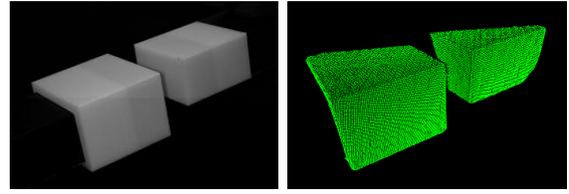


Fig. 8. Object Model Image of Plug and Jack



(a) Camera Image (b) 3D Scanner Image

Fig. 9. Camera Image and 3D Scanner Image after Positioning



Fig. 10. Relative Pose of CDs in Speed Measurement

Δt is 1 ms.

IV. EXPERIMENTS

In this section, two experimental results show the effectiveness of the proposed approach in terms of speed and accuracy.

A. Positioning Accuracy

Experimental system is shown in Fig. 6. The experimental system consists of a 7-DOF manipulator, to which the low degree of freedom multipurpose hand iGRIPP is attached [22], stereo cameras, a plug and a jack as objects to be positioned. Additionally, a 3D scanner is used to measure the relative object pose for evaluation purpose. An object image and an object model image of the plug and the jack are shown in Fig. 7 and Fig. 8, respectively. An analytical solution of the inverse kinematics is utilized to generate continuous motion of the manipulator in Cartesian space [23].

A camera image and a 3D scanner image after positioning are shown in Fig. 9. Position and orientation of the ObjectM coordinated frame in the ObjectF coordinate frame is calculated by fitting planes on to each surfaces of the plug and the jack. Positioning accuracy by using the weighted least squares and the hybrid method are shown in Table I. Reference pose of the plug is calculated just one time after pose estimation of the jack. The plug is positioned 5 times for each methods under the same conditions According to the results, the hybrid method is more accurate than weighted least squares generally.

B. Speed Measurement

In this section, the effectiveness of the proposed approach in aspect of speed is shown. In this experiment, CDs are utilized to validate speed and accuracy simultaneously. The relative pose of the CDs is defined so as that edges of the CDs are faced each other as shown in Fig. 10. The CD is grasped by the hand and moved by the manipulator so as to be same as the image shown in the reference image and the other is gripped by a fixing device and moved by an operator to measure tracking speed of the CD.

Extracted sequence images of the CDs are shown in Fig. 11. In this experiment, the CD which is gripped by the fixing device is moved for about 14 seconds. Average translational and angular velocity of the CD which is moved by the manipulator in the CameraL coordinate frame is shown in Table II. Frame rate of the image processing is around 100 frames per seconds (FPS). As shown in Fig. 11 and Table II, accurate high-speed tracking motion is realized by the proposed approach.

V. CONCLUSIONS

In this paper, high speed/accuracy appearance model based visual servoing with stereo cameras is presented. The hybrid method of stereo trigonometry for position estimation and weighted least squares for orientation estimation to enhance estimation accuracy is proposed and validated the effectiveness by comparing positioning accuracy. The effective GPU implementation of object pose estimation is proposed to

TABLE I
RELATIVE POSE MEASUREMENT RESULTS

		x (mm)	y (mm)	z (mm)	α (deg)	β (deg)	γ (deg)
Desired Value		10	10	10	0	0	0
Weighted Least Squares	Results(Average)	11.05	7.28	10.87	1.46	1.93	1.06
	Errors	1.05	2.72	0.87	1.46	1.93	1.06
Hybrid Method	Results(Average)	10.63	8.10	10.86	1.09	1.64	1.57
	Errors	0.63	1.90	0.86	1.09	1.64	1.57

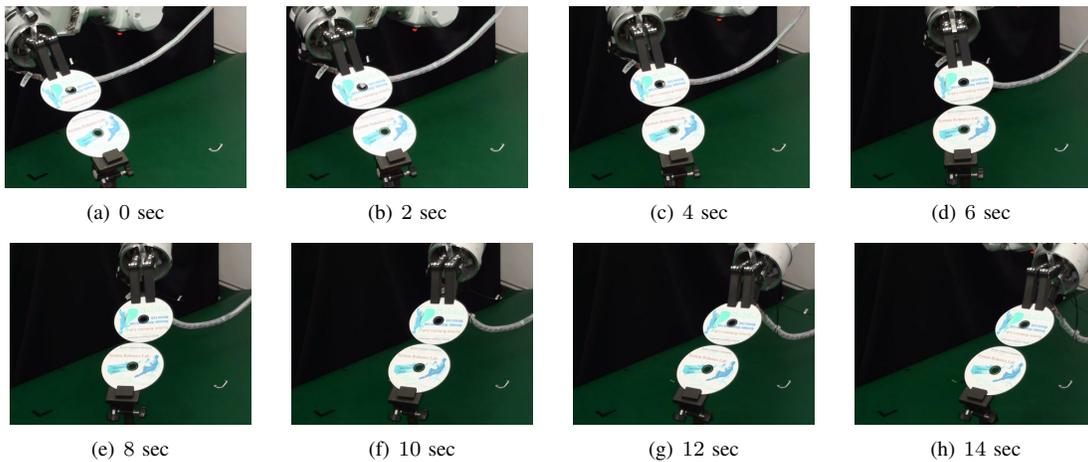


Fig. 11. Sequence Images of Speed Measurement

TABLE II
SPEED MEASUREMENT RESULT

	x (mm)	y (mm)	z (mm)	α (deg)	β (deg)	γ (deg)
Initial Pose	61	60	506	76	9	19
Final Pose	-64	-92	440	32	52	73
Displacement	-125	-152	-66	-44	43	54
	v_x (mm/s)	v_y (mm/s)	v_z (mm/s)	ω_α (deg/s)	ω_β (deg/s)	ω_γ (deg/s)
Average Velocity	-8.9	-10.9	-4.7	-3.1	3.1	3.9

increase frame rates of the image processing for the high-speed visual servoing. Besides the variable cycle on-line trajectory generator is proposed to accommodate the variable cycle of image processing and the fixed cycle of a common robot controller. The effectiveness of these approaches are validated by measuring positioning accuracy and tracking speed of an object when PBVS is working.

REFERENCES

- [1] K. Kosuge, T. Fukuda and H. Asada, "Acquisition of Human Skills for Robotic Systems," *Proc. of IEEE Int. Symp. on Intelligent Control*, pp. 496-474, 1991.
- [2] K. Hashimoto, editor, *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, World Scientific, October 1993.
- [3] G. D. Hager, "Modular System for Robust Positioning Using Feedback from Stereo Vision," *IEEE Trans. on Robotics and Automation*, Vol. 13(4), pp. 532-595, 1997.
- [4] T. Drummond and R. Cipolla, "Real-Time Visual Tracking of Complex Structures," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24(7), pp. 932-946, 2002.
- [5] P. Preisig and D. Kragic, "Robust Statistics for 3D Object Tracking," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2403-2408, 2006.
- [6] M. Tonko and H. Nagel, "Model-Based Stereo-Tracking of Non-Polyhedral Objects for Automatic Disassembly Experiments," *Int. J. of Computer Vision*, Vol. 37(1), pp. 99-118, 2000.
- [7] K. Nickels and S. Hutchinson, "Model-Based Tracking of Complex Articulated Objects," *IEEE Trans. on Robotics and Automation*, Vol. 17(1), pp. 28-36, 2001.
- [8] Y. Yoon, A. Kosaka and A. C. Kak, "A New Kalman-Filter-Based Framework for Fast and Accurate Visual Tracking of Rigid Objects," *IEEE Trans. on Robotics*, Vol. 24(5), pp. 1238-1251, 2008.
- [9] K. B. Yesin and B. J. Nelson, "Robust CAD Model Based Visual Tracking for 3D Microassembly Using Image Space Potentials," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1868-1873, 2004.
- [10] A. I. Comport, E. Marchand, M. Pressigout and F. Chaumette, "Real-Time Markerless Tracking for Augmented Reality: The Virtual Visual Servoing Framework," *IEEE Trans. on Visualization and Computer Graphics*, Vol. 12(4), pp. 615-628, 2006.
- [11] C. Zang and K. Hashimoto, "Camera Localization by CAD Model Matching," *Proc. of IEEE/SICE Int. Symp. on System Integration*, pp. 30-37, 2011.
- [12] V. Kyrki and D. Kragic, "Tracking rigid objects using integration of model-based and model-free cues," *Machine Vision and Applications*, Vol. 22, pp. 323-335, Springer Verlag, 2011.
- [13] Z. Pezzementi, S. Voros and G. D. Hager, "Articulated Object Tracking By Rendering Consistent Appearance Parts," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 3940-3947, 2009.
- [14] C. Teulière, E. Marchand and L. Eck, "Using multiple hypothesis in model-based tracking," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 4559-4596, 2010.
- [15] C. Choi and H. I. Christensen, "Robust 3D Visual Tracking Using Particle Filtering on the SE(3) Group," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 4384-4390, 2011.
- [16] J. A. Brown and D. W. Capson, "A Framework for 3D Model-Based Visual Tracking Using a GPU-Accelerated Particle Filter," *IEEE Trans. on Visualization and Computer Graphics*, Vol. 18(1), pp. 68-80, 2012.
- [17] A. Dame and E. Marchand, "Second-Order Optimization of Mutual Information for Real-Time Image Registration," *IEEE Trans. on Image Processing*, Vol. 21(9), pp. 4190-4203, 2012.
- [18] C. Collewet and E. Marchand, "Photometric Visual Servoing," *IEEE Trans. on Robotics*, Vol. 27(4), pp. 828-834, 2011.
- [19] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22(11), pp. 1330-1334, 2000.
- [20] www.opengl.org/
- [21] T. Kröger and J. Padiál, "Simple and Robust Visual Servo Control of Robot Arms Using an On-Line Trajectory Generator," *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 4862-4869, 2012.
- [22] H. Chiba and K. Kosuge, "Multipurpose Low Degrees of Freedom Robotic Hand," *Proc. of Int. Symp. on Robotics*, pp. 914-919, 2012.
- [23] T. Nammoto and K. Kosuge, "An Analytical Solution for a Redundant Manipulator with Seven Degrees of Freedom," *Int. J. of Automation and Smart Technology*, Vol. 2(4), pp. 339-346, 2012.