

# Visual Servoing Control of Baxter Robot with Obstacle Avoidance using Kinematic Redundancy

C. Yang<sup>1,2</sup>, S.Amarjyoti<sup>1</sup>, X. Wang<sup>1,3</sup>, Z. Li<sup>2</sup>, H. Ma<sup>3</sup> and C.-Y. Su<sup>2,4</sup>

<sup>1</sup>Center for Robotics and Neural Systems, Plymouth University, UK

<sup>2</sup>College of Automation Science and Engineering, South China University of Technology, Guangzhou, China.

<sup>3</sup>School of Automation, Beijing Institute of Technology, Beijing, China

<sup>4</sup>Department of Mechanical and Industrial Engineering, Concordia University, Canada

**Abstract.** In this paper, a visual servoing control enhanced by an obstacle avoidance strategy using kinematics redundancy has been developed and tested on Baxter robot. A Point Grey Bumblebee2 stereo camera is used to obtain the 3D point cloud of a target object, which is then utilized to manipulate the closely coupled dual arm Baxter robot. The object tracking task allocation between two arms has been developed by identifying workspaces of dual arms and tracing the object location in a convex hull of the workspace. By employment of a simulated artificial robot as a parallel system as well as a task switching weighting factor, the actual robot is able to restore back to the natural pose smoothly in the absence of the obstacle. Two sets of experiments have been carried out demonstrate the effectiveness of the developed servoing control method.

**Keywords:** Visual servoing, Point cloud, Obstacle Avoidance

## 1 Introduction

Visual sense is one of the most essential markers of intelligence and contributes immensely to the interactions with our environment. Visual feedback plays an important role for an anthropomorphic robot to operate in human surroundings. Especially, the advancements in 3D Vision Guided Robots (VGRs) have led to accurate non-contact geometrical measurements and reduction of workspace ambiguity. Therefore, strides made in robotic and machine vision are of paramount importance for engineering, manufacturing and design processes [1]. Visual servoing(VS) is defined as the use of visual feedback mechanisms for the kinematic control of a robot. Based on the positioning of the camera on the link and control techniques, VS ramifies into several types. Eye-in-hand and eye-to-hand VS are represented by the position of the camera on the robotic manipulator. Being attached on the robot arm, eye-in-hand VS provides a narrower field of view as compared to eye-to-hand servoing. Many control schemes use either a direct visual servoing or a dual loop system [6]. A visual processing method for hand gesture recognition is developed in [8] for control a simulated iCub robot

on YARP platform. In [7], a noncontacting vision-based method of robot tele-operation is presented, whereas a human operator communicate simultaneous six-degree-of-freedom motion tasks to a robot manipulator by having the operator perform the three-dimensional human hand-arm motion that would naturally be used to complete an object manipulation task.

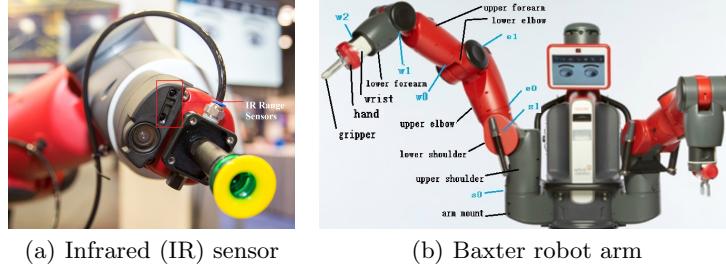
These schemes are robust but come at the price of reduced performance and speed variables. The drawbacks of time delay between image acquisition and its relation to latency in robot controller can be solved by the use of an open loop based VS for computing the reference input of the joint controller from the targets projection once [3]. It is practically useful to integrate visual sensing with other sensing technologies to develop a hybrid sensing system for better control performance. A combination of force sensor, CCD cameras and laser projector is used for accurate drilling on a 3D surface of unknown pose [2]. Eye-to-hand pose estimation and eye-in-hand object grasping in a hybrid control loop have been studied in [11][9][13]. In this paper, we use a similar approach that subsumes stereo image processing for pose estimation and the range data from IR sensors near the end effectors for precise gripping.

Similar to our human arm, the Baxter robot manipulator is of 7 joints. When the degree of freedom (DOF) in joint space is larger than that required in the task space, there is redundancy. At most 6DOF is required for manipulation in the Cartesian space, and the redundancy can be used to generate motion for obstacle avoidance. In [10, 14], the decomposition of the solution into a particular and a homogeneous component effectively illustrates the priority of the multiple goals that is exact end-effector control with redundant degrees of freedom maximizing the distance to obstacles. In this work, the robot end effector is controlled to follow the reference trajectory generated according to the detected target object's position, while at the same time the joint space redundancy is utilized to avoid obstacle in the environment. Using our proposed method, the redundant Baxter robot arm will avoid obstacle without sacrificing the tracking performance of the end effector. In addition, the develop control method also enables the robot arm to restore its natural pose when the obstacle is absent, by employment of a simulated artificial robot as a parallel system based on kinematic model of the robot.

## 2 Baxter® Robot and its Kinematics

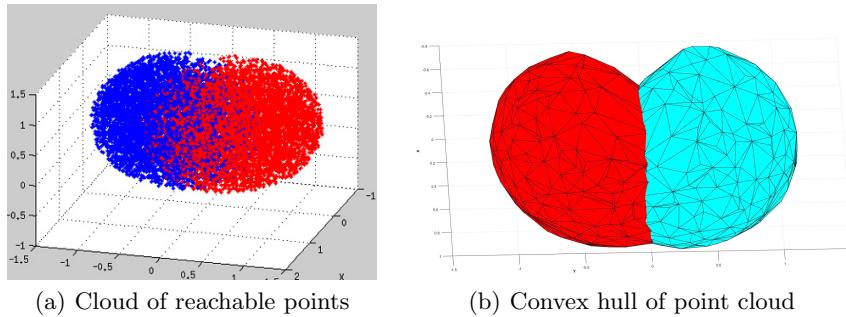
### 2.1 Workspace identification of the Baxter® robot arms

Baxter® robot is a semi-humanoid robot consisting of two arms and a rotational head on its torso, which can be installed on a movable pedestal. Several attributes such as the presence of a CCD camera and an IR range sensor at the end effector as shown in Fig. 1(a), make Baxter the definable option. As illustrated in Fig. 1(b), each arm of Baxter robot constitutes of 8 links and 7 rotational joints with different interchangeable grippers, e.g., electric gripper or a vacuum cup. The kinematics model of a Baxter robot arm was built in our previous work [5], where the details of DH parameters as well as joint rotation



**Fig. 1.** Left: IR Range sensor; Right: Illustration of the 7DOF Baxter robot arm (Modified from [4]).

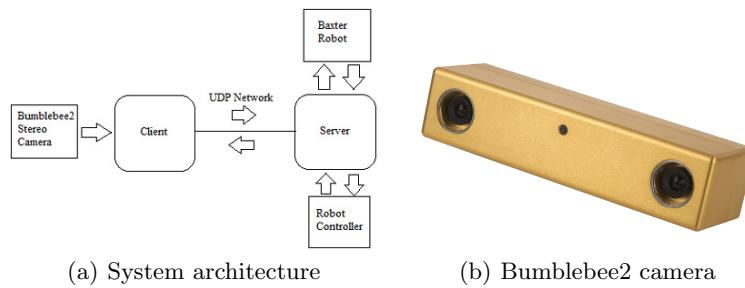
limits are provided. Robot manipulator workspace boundary estimation is essential for improvising of robotic algorithms and optimization of its overall design and analysis. In this paper, we first extend the Monte Carlo method used on a single arm [5] to dual arms of Baxter robot, in order to identify the reachable workspace of two arms. Homogenous radial distribution is used to generate 6000 points in the joint space for each arm separately. The joint angle values are chosen randomly and forward kinematics is implemented to evaluate the end effector points, creating a point cloud of the reachable workspace for dual arms. The generated point cloud of the workspace for both the manipulators is shown in Fig. 2(a). Next, Delaunay triangulation is applied to the points in the 3D space to generate a set of points with a circumcircle without any points in its interior. This facilitates the creation of a convex hull of the joint space. It is used to constrain the workspace and to distinguish the ranges of control between the left and right arms for efficient manoeuvrability.



**Fig. 2.** Workspace Identification

### 3 Visual Sensor and its Preprocessing

A client-server UDP network as shown in Fig. 3(a) is employed to integrates various system components and confers parallel processing. The machine vision processing is done in the client computer connected with a Point Grey® Bumblebee2 stereo camera with IEEE-1394 Firewire connection. The Point Grey Bumblebee2 stereo camera, as shown in Fig. 3(b), is a 2 sensor progressive scan CCD camera with fixed alignment between the sensors. Video is captured at a rate of 20 fps with a resolution of  $640 \times 480$  to produce dense colored depth maps to assist in tracking and a viable pose estimation of the object. The resolution-speed trade-off has to be managed concisely as an increased frame speed gives a smooth robot trajectory whereas enhances the processing time. And an increased resolution provides a denser, more accurate point cloud for feature extraction but with increased latency.



**Fig. 3.** Left: Point Grey® Bumblebee2 camera; Right: communication network

#### 3.1 Camera calibration

Camera calibration is necessary as the use of lenses introduces nonlinearities and deviates from the simple pin-hole model such as lens distortion, namely radial and tangential distortion. The camera parameters, namely the intrinsic, extrinsic and distortion are evaluated by the use of a 2D checker-board pattern. 3D reference models were avoided due to computational complexity and high cost of precise calibration objects. In this work, 20 checkerboard images were fed to the calibrator algorithm encompassing differential angles in the projection space. This provides enough values to estimate the camera geometry parameters, including 4 intrinsic parameters of  $f_x$ ,  $f_y$ ,  $c_x$ ,  $c_y$ , 5 distortion parameters of  $k_1$ ,  $k_2$ ,  $k_3$  (radial) plus  $p_1$ ,  $p_2$  (tangential) and the camera extrinsic parameters of  $\Phi$ ,  $\Psi$ ,  $\Theta$  (rotation) together with  $T_x$ ,  $T_y$ ,  $T_z$  (translation). The point “Q” on the object plane is related with the image plane point “q” by the following equations. A summary to the definition of these parameters is provided in Table 1.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = sA[r_1 \ r_2 \ r_3 \ T] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}, \quad x = f \frac{X}{Z} + c_x, \quad y = f \frac{Y}{Z} + c_y, \quad q = sHQ \quad (1)$$

The camera parameters are finally obtained using Closed Form solution and Maximum Likelihood estimation. Further the distortion is nullified from the image points by introducing the following equations

$$x_{corrected} = x(1 + k_1r^2 + k_2r^4 + k_3r^6), y_{corrected} = y(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

**Table 1.** Definition of variables

$X, Y$	World co-ordinates of the point	$c_x, c_y$	Projection displacement parameter
$x, y$	Image plane co-ordinates of the point	$H, A$	Homography matrix, and Intrinsic matrix
$f$	Focal length of the camera lens	$r_1, r_2, r_3$	spatial rotational matrices
$s$	scaling factor	$T$	translational matrix

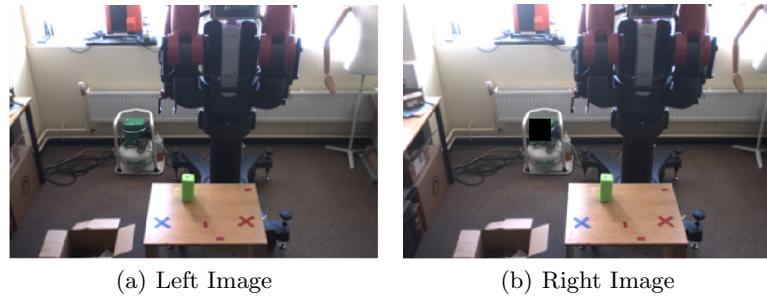
The co-ordinate transformation of the detected feature points from the Bumblebee2 co-ordinates to Baxter co-ordinates can be achieved by a homogeneous transformation matrix to be obtained as

$$\mathbf{T} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ Z_1 & Z_2 & Z_3 & Z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \quad (2)$$

where  $(x_i, y_i, z_i)$  and  $(X_i, Y_i, Z_i)$ ,  $i = 1, 2, 3, 4$  are four non-collinear points measured from the robot coordinate and the Bumblebee2 coordinate, respectively.

## 4 Object Detection and Localization

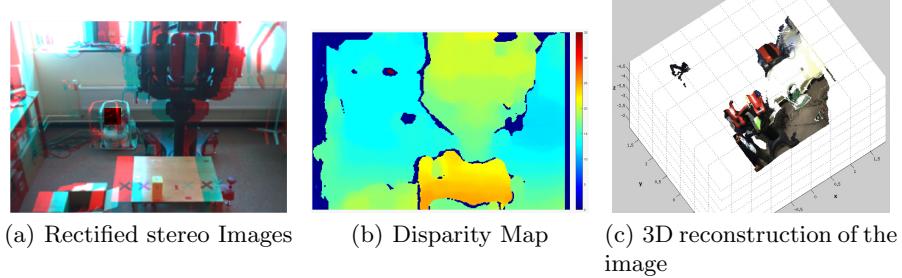
### 4.1 3D reconstruction



**Fig. 4.** Images captured from Bumblebee2

The images captured by the Bumblebee2 stereo camera in active ambient lighting are shown in Fig. 4. Both the images are calibrated using the camera

intrinsics and are corrected for distortion. Subsequently, the undistorted images are stereo rectified in order to align the epipolar lines of both the projection planes and ensure the presence of similar pixels in a specified row of the image. The images obtained are then frontal parallel and are ready for correspondence estimate. The Essential and the Fundamental matrix are calculated by using Epipolar geometry. Essential matrix is a  $3 \times 3$  matrix with 5 parameters; two for translation and three for the rotation values between the camera projection planes. The Fundamental matrix on the other hand, represents the pixel relations between the two images and has seven parameters, two for each epipole and three for homography that relates the two image planes. Bouguets algorithm is then implemented to align the epipolar lines and shift the epipoles to infinity. Fig. 5(a) depicts the results of stereo rectification where the red and cyan colors represent the left and right images with row aligned pixels.



**Fig. 5.** Stereo images and 3D reconstruction.

**Table 2.** Definition of variables

1	$x^l$	column value of left image pixel	5	$f$	focal length
2	$x^r$	column value of right image pixel	6	$d$	disparity
3	$D$	Depth	7	$Q$	Projection matrix
4	$T$	Baseline	8	$X/W, Y/W, Z/W$	3D world co-ordinates

The definition of variables used below is provided in Table 2. Stereo correspondence is a method of matching pixels with similar texture across two co-planar image planes. The distance between the columns of these perfectly matched pixels is defined below:

$$d = x^l - x^r \quad (3)$$

Block matching is implemented for evaluating the correspondence between the images. Block sizes of 15 pixel window are used to find the matches by the use of SAD (sum of absolute differences). The disparity range is kept low as [0 40] in order to match the indoor low texture difference and taking into account computational speed. Semi Global method is used to force the disparity values to the neighboring pixels for a more comprehensive result. The disparity output

is shown in Fig. 5(b). Disparity is inversely proportional to the depth of the pixel and is related by the Triangulation equation (4). Triangulation refers to the estimation of depth of an object by visualizing its location from two different known points.

$$D = T \frac{f}{d} \quad (4)$$

The reconstruction of the image in the Cartesian co-ordinates is obtained by the use of projection matrix evaluated using Bouguets algorithm (5). Fig. 5(c) depicts the 3D reconstruction of the robots workspace.

$$Q [x, y, d, 1]^T = [X, Y, Z, W]^T \quad (5)$$

#### 4.2 Object detection

Color based segmentation is used in order to separate a single color object from the background. The image is converted into L\*a\*b\* color space and the Euclidean distance between red-green and yellow-blue opponent components of the object and a and b matrices calculated. The minimum value gives the most accurate estimate of the object. Further, the corners of the object are calculated by Harris corner detector and the centroid calculated by intersection of the diagonals. The depth value of the centroid is then extracted from reconstructed point cloud of the task space. Fig. 6 shows the calculated robot co-ordinates of the centroid of the object after co-ordinate transformation.



**Fig. 6.** Object detection results

#### 4.3 Switching scheme for dual arms

The depth of the object to be manipulated is mapped to the convex hull to expedite the process of decision making, because a point cloud matching would be hinder the processing speed. This is done by checking the presence of the point's respective co-ordinates in the convex-hull projection on the three Cartesian planes. Hence, if  $(X_1, Y_1, Z_1)$  be the point representing the object, its presence in the 3D hull is detected by following five steps:

The above procedure ensures an efficient methodology to reach the object in the robot workspace. It mimics the human intuition of using the nearest

**Algorithm 1** Object detection in the 3D workspace hull

- 
- 1: Check if the  $XY$  plane projection of the hull contains the point  $(X_1, Y_1)$ ,  $YZ$  plane contains point  $(Y_1, Z_1)$  and  $XZ$  plane contains point  $(X_1, Z_1)$  using Ray Casting Algorithm;
  - 2: Obtain the presence decision in the workspace of both arms;
  - 3: If the point is present in both of the manipulator workspace, give the control priority to the arm with the smallest Euclidean distance from manipulator;
  - 4: Divide the control based on the detection of the point in left or right workspaces;
  - 5: If the point lies outside both the workspaces, stop arm movement to avoid singularity.
- 

possible arm for grabbing in order to avoid the use of excessive body movements and minimize the use of energy.

#### 4.4 Detection of collision points

The collision points,  $\mathbf{p}_{cr}$  and  $\mathbf{p}_{co}$ , are the two points either on the robot or on the obstacle, which covers the nearest distance between the robot and the obstacle. The forward kinematic of the robot is used to estimate the collision points. Each link of the robot can be seen as a segment in 3-D space. The coordinates of the endpoints of the segments, i.e. the coordinates of the joints, in Cartesian space can be obtained by forward kinematics.

The problem of estimating the collision points can be seen as the problem of searching the nearest point between the obstacle segment  $\mathbf{q}_{l1} - \mathbf{q}_{l2}$  and the segments that stand for the robot links. Firstly, the distance between segment  $\mathbf{q}_{l1} - \mathbf{q}_{l2}$  and the  $i^{th}$  segment of the robot links  $[x_i, y_i, z_i] - [x_{i+1}, y_{i+1}, z_{i+1}]$  can be calculated by 3-D geometry, which is denoted by  $d_i$ ; the nearest points on the robot link and the obstacle are denoted by  $\mathbf{p}_{cri}$  and  $\mathbf{p}_{coi}$ , respectively. Then we have the collision point  $\mathbf{p}_{cr} = \mathbf{p}_{cri\min}$  and  $\mathbf{p}_{co} = \mathbf{p}_{coi\min}$  and the distance  $d = d_{i\min}$  where  $i_{\min} = \arg \min_{i=0,1,\dots,n} d_i$ .

### 5 Obstacle Avoidance and Restoring Control

Once the target object's co-ordinates has been detected, end effector desired trajectory for the visual tracking task can be defined as the moving trajectory of the target. While for visual guided manipulation task, the end effector desired trajectory can be generated according to the location of the target object. Denote the joint velocities as  $\dot{\theta}$  and end-effector velocity as  $\dot{x}$ , then they must satisfy  $\dot{x} = \mathbf{J}\dot{\theta}$  where  $\mathbf{J}$  is the Jacobian matrix of each arm. If the dimension of  $\dot{\theta}$  is larger than the dimension of  $\dot{x}$ , the manipulator will become kinematically redundant, which can be used to achieve some secondary goals, e.g. obstacle avoidance, in addition to the end-effector trajectory following task. For the inverse kinematic problem of the kinematically redundant manipulator, infinite number of solutions can be find. The general solution is given by (6).

$$\dot{\theta} = \mathbf{J}^\dagger \dot{x} + (I - \mathbf{J}^\dagger \mathbf{J})\mathbf{z}, \quad (6)$$

where  $\mathbf{J}^\dagger = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$  is the pseudo-inverse of  $\mathbf{J}$  and  $\mathbf{z}$  is an arbitrary vector, which can be used to achieve the obstacle avoidance [10].

### 5.1 Collision avoidance strategy

When the collision points  $\mathbf{p}_{cr}$  and  $\mathbf{p}_{co}$  are found, the manipulator needs to move away from the obstacle. The desired velocity  $\dot{\mathbf{x}}_o$  moving away from the obstacle is designed as (7).

$$\dot{\mathbf{x}}_o = \begin{cases} \mathbf{0}, d \geq d_o \\ \frac{d_o - d}{d_o - d_c} v_{max} \frac{\mathbf{p}_{cr} - \mathbf{p}_{co}}{d}, d_c < d < d_o, \\ v_{max} \frac{\mathbf{p}_{cr} - \mathbf{p}_{co}}{d}, d \leq d_c \end{cases} \quad (7)$$

where  $d = \|\mathbf{p}_{cr} - \mathbf{p}_{co}\|$  is the distance between the obstacle and the manipulator;  $v_{max}$  is the maximum obstacle avoidance velocity;  $d_o$  is the distance threshold that the manipulator starts to avoid the obstacle;  $d_c$  is the minimum acceptable distance and the manipulator will avoid at the maximum speed.

On the other hand, in order to eliminate the influence of the obstructing when the obstacle has been removed, the manipulator is expected to restore its original state. To achieve that, an artificial parallel system of the manipulator is designed in the controller in real time to simulate its pose without the influence of the obstacle, as shown in Fig. 7, where the dashed black line indicates the parallel system. The restoring velocity  $\dot{\mathbf{x}}_r$  is designed as a closed-loop system as

$$\dot{\mathbf{x}}_r = \mathbf{K}_r \mathbf{e}_r, \quad (8)$$

where  $\mathbf{K}_r$  is a symmetric positive definite matrix and  $\mathbf{e}_r = [\mathbf{e}_{r1} \ \mathbf{e}_{r2} \ \mathbf{e}_{r3}]^T$  is the position errors of the joints between the parallel system and the real system.

Based on our previous work [12], the control strategy is given below

$$\dot{\boldsymbol{\theta}}_d = \mathbf{J}_e^\dagger (\dot{\mathbf{x}}_d + \mathbf{K}_e \mathbf{e}_x) + (\mathbf{I} - \mathbf{J}_e^\dagger \mathbf{J}_e) [\alpha \mathbf{z}_o + (1 - \alpha) \mathbf{z}_r] \quad (9)$$

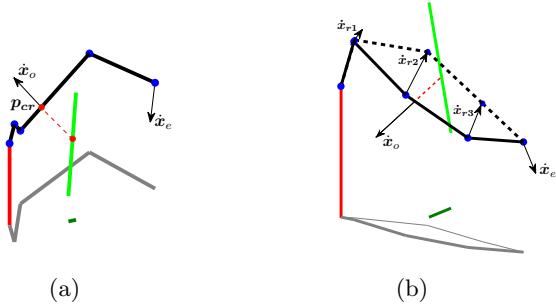
where

$$\mathbf{z}_o = [\dot{\mathbf{x}}_o^T \mathbf{J}_o (\mathbf{I} - \mathbf{J}_e^\dagger \mathbf{J}_e)]^\dagger (\dot{\mathbf{x}}_o^T \dot{\mathbf{x}}_o - \dot{\mathbf{x}}_o^T \mathbf{J}_o \mathbf{J}_e^\dagger \dot{\mathbf{x}}_e) \quad (10)$$

$$\mathbf{z}_r = [\dot{\mathbf{x}}_r^T \mathbf{J}_r (\mathbf{I} - \mathbf{J}_e^\dagger \mathbf{J}_e)]^\dagger (\dot{\mathbf{x}}_r^T \dot{\mathbf{x}}_r - \dot{\mathbf{x}}_r^T \mathbf{J}_r \mathbf{J}_e^\dagger \dot{\mathbf{x}}_e) \quad (11)$$

### 5.2 Object gripping with hybrid servoing

Stereo camera provides a wider angle of view serving the eye to hand servoing system, while for meticulous sensing of the object, an IR range sensor equipped at the end effector provides an eye-in-hand feedback, and helps to reduces the processing time and provides satisfactory accuracy. The IR range sensor used-provides 16 bit digital data and checks for the location of object along the  $Z$



**Fig. 7.** The artificial parallel system built using Baxter® kinematics model [5]. The solid black line indicates the real manipulator. The dashed black line indicates the manipulator in the artificial parallel system.

and  $Y$  axes. The camera and robot  $X$  axes are perfectly aligned and hence the real-time error is minimal. But the error in other axes needs to be neutralized, partially due to lack of visibility of some portions of the object and objects with ambiguous dimensions. A low pass filter is applied to the range data in order to remove the occasional anomalies.

## 6 Experiment Studies

In order to demonstrate the proposed visual servoing method, two groups of experiments of visual tracking task and visual guided manipulation task are performed, with obstacle and without obstacle for comparison.

### 6.1 Visual tracking task

In the visual tracking experiments, the robot is controlled to follow the object using two manipulators. The experiment video frames are shown in Fig. 8 and Fig. 9 for the experiments with and without obstacle respectively. The end-effector trajectory of this two comparative experiments are shown in Fig. 10.

As can be seen, the robot automatically choose the appropriate manipulator to track the object based on the workspaces. When the obstacle is found near the manipulator, the corresponding manipulator change its pose to avoid collision and, at the same time, keep its end-effector position tracking the object.

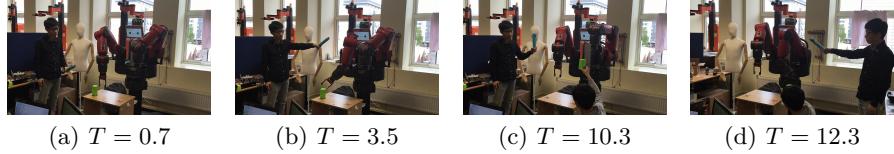
### 6.2 Visual guided manipulation task

In the visual guided manipulation experiments, the robot is controlled to pick up the object and put it into a box at a predefined position. The end-effector trajectory is planed using the Algorithm 2.

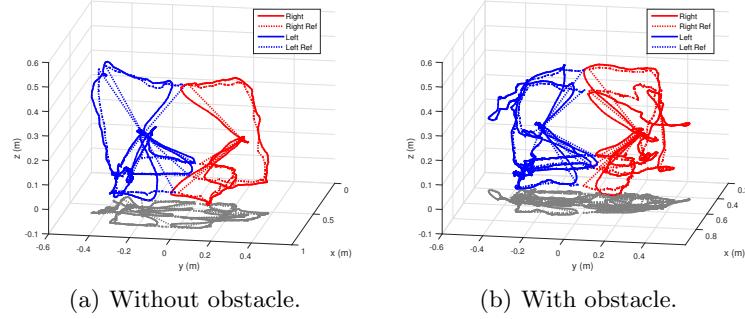
The experiment video frames are shown in Fig. 11 and Fig. 12 for the experiments with and without obstacle respectively. The end-effector trajectory



**Fig. 8.** Video frames of visual tracking without obstacle without obstacle.



**Fig. 9.** Video frames of visual tracking with obstacle with obstacle.



**Fig. 10.** End-effector trajectory of visual tracking. The blue and red dashed lines indicates the reference end-effector trajectory of the left and right manipulators respectively; the blue and red solid lines indicates the actual trajectory of the end-effectors of the left and right manipulators respectively.

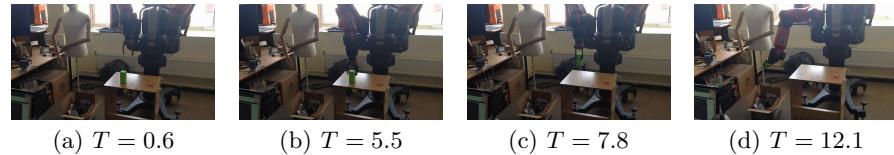
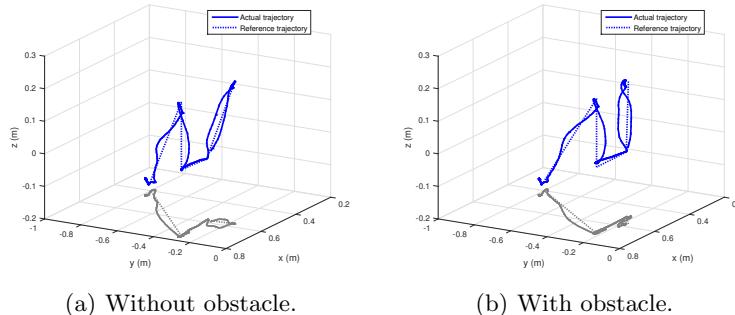
of this two comparative experiments are shown in Fig. 13. As can be seen, the robot can pick up the object precisely even if the position of object has been changed. When the obstacle is found near the manipulator, the corresponding manipulator change its pose to avoid collision but can still move the end-effector right above the box and drop the object into the box.

## 7 Conclusion

This paper developed and implemented a visual servoing control method with automatic obstacle avoidance on the dual arm Baxter robot. Stereo imaging and range sensors are utilized to attain eye-to-hand and eye-in-hand servoing, such that both target object and obstacle are detected and localized. Kinematic redundancy has been exploited for obstacle avoidance. A simulated artificial robot

**Algorithm 2** Visual guided manipulation trajectory planning

- 1: Move to the position 10cm behind the object;
- 2: Move forward step by step with a step length of 0.5cm;
- 3: Check the IR range sensor after each 0.5cm step;
- 4: If the range data is less than a threshold, grip the object and move up by 20cm;
- 5: Move to the predefined position above the box and release the object;

**Fig. 11.** Video frames of visual guided manipulation without obstacle.**Fig. 12.** Video frames of visual guided manipulation with obstacle.**Fig. 13.** End-effector trajectory of the visual guided manipulation. The dashed lines indicates the reference end-effector trajectory; and the solid lines indicates the actual trajectory of the end-effectors.

of same kinematics as Baxter robot is employed as a parallel system to enable the robot restoring back to the natural pose when the obstacle is absent. Two switching mechanisms have been designed, one for task allocation between two arms for visual tracking task, while the other for smooth switching between obstacle avoiding task and restoring task. An object tracking and an object manipulation experiments have been performed to test the develop control method, and

the results have demonstrated the effectiveness and robustness of the developed approach.

## References

1. Association for safe international road travel statistics. <http://www.asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics.aspx>
2. Chang, W.C., Shao, C.K.: Hybrid fuzzy control of an eye-to-hand robotic manipulator for autonomous assembly tasks. In: SICE Annual Conference 2010, Proceedings of, pp. 408–414 (2010)
3. Daigavane, P., Bajaj, P.: Road lane detection with improved canny edges using ant colony optimization. In: Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on, pp. 76–80 (2010)
4. Guizzo, E., Ackerman, E.: How rethink robotics built its new baxter robot worker. IEEE Spectrum (2012)
5. Ju, Z., Yang, C., Ma, H.: Kinematics modeling and experimental verification of baxter robot. In: Control Conference (CCC), 2014 33rd Chinese, pp. 8518–8523 (2014). DOI 10.1109/ChiCC.2014.6896430
6. Kamat, V., Ganesan, S.: A robust hough transform technique for description of multiple line segments in an image. In: Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on, vol. 1, pp. 216–220 vol.1 (1998). DOI 10.1109/ICIP.1998.723460
7. Kofman, J., Wu, X., Luu, T., Verma, S.: Teleoperation of a robot manipulator using a vision-based human-robot interface. Industrial Electronics, IEEE Transactions on **52**(5), 1206–1219 (2005). DOI 10.1109/TIE.2005.855696
8. Li, C., Ma, H., Yang, C., Fu, M.: Teleoperation of a virtual icub robot under framework of parallel system via hand gesture recognition. In: Proceedings of the 2014 IEEE World Congress on Computational Intelligence. WCCI, Beijing (2014)
9. Luo, R., Chou, S.C., Yang, X.Y., Peng, N.: Hybrid eye-to-hand and eye-in-hand visual servo system for parallel robot conveyor object tracking and fetching. In: Industrial Electronics Society, IECON 2014 - 40th Annual Conference of the IEEE, pp. 2558–2563 (2014). DOI 10.1109/IECON.2014.7048866
10. Maciejewski, A.A., Klein, C.A.: Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. The international journal of robotics research **4**(3), 109–117 (1985)
11. Urmson, C., Whittaker, W.: Self-driving cars and the urban challenge. Intelligent Systems, IEEE **23**(2), 66–68 (2008). DOI 10.1109/MIS.2008.34
12. Wang, X., Yang, C., Ma, H.: Automatic obstacle avoidance using redundancy for shared controlled telerobot manipulator. In: Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE 5th Annual International Conference on, p. [Accepted] (2015)
13. Wijesoma, W., Kodagoda, K., Balasuriya, A., Teoh, E.: Road edge and lane boundary detection using laser and vision. In: Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on, vol. 3, pp. 1440–1445 vol.3 (2001). DOI 10.1109/IROS.2001.977183
14. Zlajpah, L., Nemec, B.: Kinematic control algorithms for on-line obstacle avoidance for redundant manipulators. In: Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on, vol. 2, pp. 1898–1903 vol.2 (2002). DOI 10.1109/IRDS.2002.1044033