Visual Servoing for Object Manipulation: A Case Study in Slaughterhouse

Wu, Haiyan; Andersen, Thomas Timm; Andersen, Nils Axel; Ravn, Ole

Published in:
Proceedings of the International Conference on Robotics and Automation

Link to article, DOI:
10.1109/ICARCV.2016.7838841

Publication date:
2016

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Wu, H., Andersen, T. T., Andersen, N. A., & Ravn, O. (2016). Visual Servoing for Object Manipulation: A Case Study in Slaughterhouse. In *Proceedings of the International Conference on Robotics and Automation* IEEE. https://doi.org/10.1109/ICARCV.2016.7838841

Visual Servoing for Object Manipulation: A Case Study in Slaughterhouse

Haiyan Wu, Thomas Timm Andersen, Nils Axel Andersen, Ole Ravn
Automation and Control
Technical University of Denmark
Kgs. Lyngby, 2800, Denmark
{*hwua, ttan, naa, or*}*@elektro.dtu.dk*

***Abstract*—Automation for slaughterhouse challenges the design of the control system due to the variety of the objects. Realtime sensing provides instantaneous information about each piece of work and thus, is useful for robotic system developed for slaughterhouse. In this work, a pick and place task which is a common task among tasks in slaughterhouse is selected as the scenario for the system demonstration. A vision system is utilized to grab the current information of the object, including position and orientation. The information about the object is then transferred to the robot side for path planning. An online and offline combined path planning algorithm is proposed to generate the desired path for the robot control. An industrial robot arm is applied to execute the path. The system is implemented for a lab-scale experiment, and the results show a high success rate of object manipulation in the pick and place task. The approach is implemented in ROS which allows utilization of the developed algorithm on different platforms with little extra effort.**

I. Introduction

With increasingly enhanced sensing capability, advanced control solutions and powerful hardware platforms, robotic systems start stepping into various areas, such as navigation, exploration, entertainment, industry, human welfare and so on [1]–[6]. In recent years robotic system is more and more widely involved in processing and production of industry, either working along side human-being or cooperating with human/other robots to complete task together. In some cases the object involved in the task has constant physical parameters such as size, shape, color and so on. However, with robots involved in different applications for example the robotics system in food industry, the variety of the objects has to be considered during system design. For the tasks in food industry, for example the tasks in slaughterhouse, the objects usually appear in different size although they share similarity in shape, see Fig. 1 as an example. Fig. 1(a) shows example of chickens that are processed in poultry slaughterhouse. The chickens are close in shape and color, but they differ in size and weight. Fig. 1(b) gives another example with pigs being the target object. The rota stick inserted in the throat of the pig has to be removed. In this case, the position and motion of the rota stick depend on the size and weight of the pig. These differences have to be dealt with if a robotic system is considered for completing tasks. Therefore, a realtime sensing system is required to provide instantaneous information about each piece of work for the control system. This work focuses on providing a general realtime sensor-based control system to applications where dynamic adjustment to varying objects is a must.

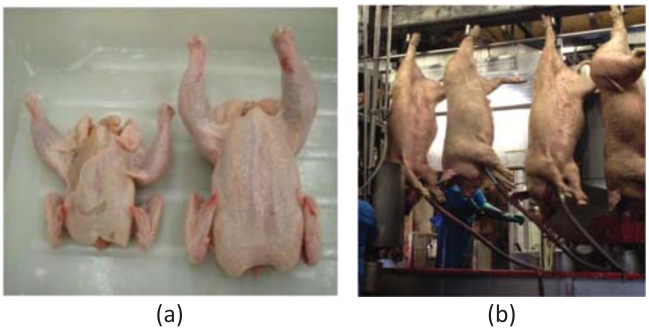


Fig. 1. Chickens shown in (a) and pigs shown in (b) as target objects in slaughterhouse have similar shape but different size.

Visual information obtained from camera is utilized for closed-loop robot control, which is referred to as visual servoing system [7], [8]. An overview about the properties and challenges of visual servo systems can be found in [9]–[11]. A position based visual servoing (PBVS) is applied in this paper, where the object information is retrieved from the image and converted to 3D pose (including position and orientation) information for robot control. With the PBVS the control tasks are planned in the 3D Cartesian space, and the camera model is required for mapping the data from 2D to 3D space. In order to build up a visual servoing system, it needs knowledge from different areas including robot modelling such as kinematics and dynamics, control theory, computer vision including image processing and camera calibration, sensor system integration and so on [12]–[14].

This paper focuses on a case study of visual servoing in slaughterhouse. A pick and place task, which is a common task in slaughterhouse, is selected for system demonstration, as shown in Fig. 2. The object for manipulating in this task is loin which is transferred by a conveyor belt. The task here is to grab the loin from the conveyor belt and hang it onto a Christmas tree. In order to complete the task, it requires a vision system which detects the loin in realtime. Then, the robot arm has to track the motion of the loin based on

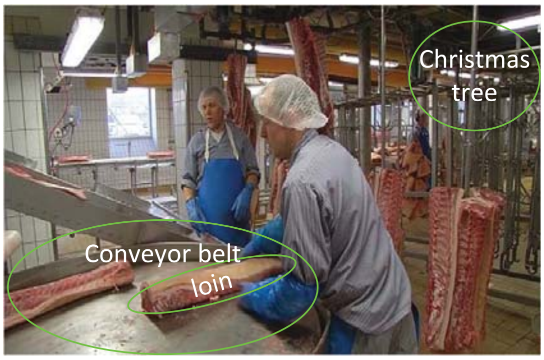


Fig. 2. A pick and place task in slaughterhouse: the target object loin has to be grabbed from the conveyor belt and transferred to the hook on a metal Christmas tree.

the online visual information. The loin is grasped from the conveyor belt at a certain position and transferred by the robot to a pre-defined goal position. The remainder of this paper is organized as follows: the overall system platform including robot arm, camera and gripper is described in section II. The image processing algorithm, the coordinate transformation, the path planning algorithm and the robot arm control are presented in section III. In section IV, the experimental setup and validation of the system are discussed.

II. Hardware Platform

Pick and place task is a common task type for a robotic system in industry and is therefore selected as the test scenario for this work. Fig. 3 (a) shows a general platform for pick and place task. Objects with different size and shape are transferred by a conveyor belt, while sensors are utilized to provide instantaneous information about the objects. The robot is used to pick up the object from the conveyor belt and move it to a desired position. It has to be mentioned that each hardware component in the system has its local coordinate system, e.g. the camera, robot and gripper have their own frames as denoted by C_c, C_r and C_g in Fig. 3 (a). The transformation matrix among these coordinates, such as transformation matrix T_{c2g} from the camera frame to the gripper frame and that T_{g2r} from the gripper frame to the robot base frame, have to be determined before passing the visual information to the robot control.

The selected hardware for this work within the platform is shown in Fig. 8 (b). It consists mainly of four parts: the robot arm, the gripper, the visual sensor, and the computer. The details about these components are described in the following.

A. Robot Arm

In this work, an industrial robot arm Motoman MH5L [15] is mounted for completing the manipulation task. The MH5L is a compact 6-axis robot with a weight of 29 kg. It has an extended 895 mm reach and a maximum payload 5 kg. The motion range and maximum speed for each axis are listed in Tab. I.

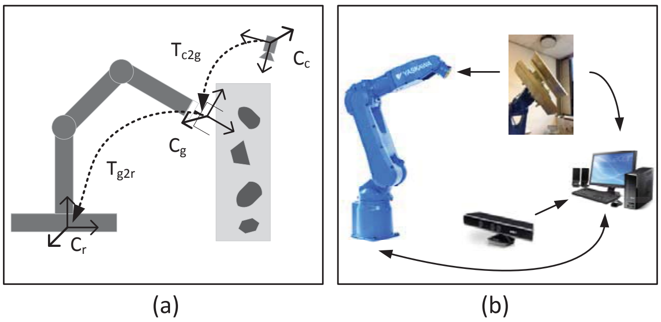


Fig. 3. General hardware platform for pick and place task shown in (a) and selected hardware for loin task in slaughterhouse shown in (b).

TABLE I
Specifications of MH5L.

Axes	Motion range [°]	Maximum speed [°/sec.]
S	± 170	270
L	$+150/-65$	280
U	$+255/-138$	300
R	± 190	450
B	± 125	450
T	± 360	720

The open source software *ROS Industrial* [16]–[18] provides tools and drivers for industrial hardware. It is used for communicating with the robot arm trough the Motoman industrial robot controller *FS100* [19].

B. Camera

In order to capture the instantaneous information of the loin, including its position and orientation on the conveyor belt, a camera has to be included in the system. In this work, the Microsoft X-Box Kinect sensor [20] is selected as the optical sensor for object detection. The Kinect sensor provides both color image and depth image from an RGB camera and an infrared camera respectively. The Kinect sensor has been adopted in many indoor robotic applications, e.g. the Kinect sensor is utilized for 3D reconstruction and interaction based on GPU pipeline in [21], for tracking human hand articulation in [22] and for mobile robots navigation in [23]. A study about using Kinect for robotics applications is given in work [24].

In this work, the depth image from Kinect sensor is utilized for object localization, and the RGB image is used to calculate the 3D coordinates. The parameters of the Kinect sensor relevant for this project are listed below (from work [25]):

- depth sensor range: 0.8 m - 4.0 m
- nominal special range: 320 × 240 pixels, 16-bit depth
- framerate: approx. 30 frames/sec.
- nominal depth resolution at 2 m distance: 1 cm

The Kinect open source software *freenect* provided by OpenKinect [26] in ROS is applied for image streaming and automatic calibration of the Kinect sensor.

C. Gripper

The gripper used in this work for grasping loin is designed by Danish Technological Institute DMRI [27]. The gripper is shown in Fig. 4. It is a pneumatically actuated gripper with adjustable holding force. It has two jaws and the distance between the jaws can be adjusted. In addition, it has low weight and can be easily mounted on the end effector of the robot.

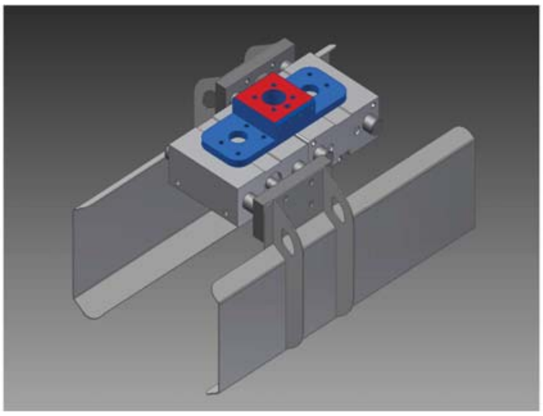


Fig. 4. Gripper for loin task from DMRI.

The robot arm, the camera and the gripper are connected to the same computer running ROS. The calibration among these three hardware components is required, which will be introduced in section III.

III. Algorithm

The overall structure of the algorithm for the pick and place task if given in Fig. 5. For finishing the loin task successfully there are mainly four steps:

- step 1: object detection in the image plane, which finds and locates object in the image. The output of this step is the 2D coordinates of the object $[u, v]$;
- step 2: coordinates transformation from 2D image coordinates to 3D Cartesian space including position $[x, y, z]$ and orientation information $[\alpha, \beta, \gamma]$ (Euler angles [28]). It is based on offline calibration between the camera and the robot system. The output of this step is the relative pose $X = [x, y, z, \alpha, \beta, \gamma] \in \Re(6)$ between the object and the robot.
- step 3: combined online and offline path planning based on the visual feedback, which generates reference path for robot arm control. The output of this step is a serial of poses along time axis $P(t)$;
- step 4: control of robot arm in joint space by mapping the poses $P(t)$ from Cartesian space to joint space $(q_1(t), ..., q_6(t))$ through robot Jacobian. The output of this step is the command signal sent to the robot arm.

The details of these four steps are illustrated in the following.

A. Object Detection

The visual sensor Kinect is chosen here to detect object in realtime. The images are transferred to the computer for image processing. Features such as color, shape and size of the object can be utilized for object detection. In this work, the object loin is transferred by a conveyor belt whose height is known. The Kinect sensor is mounted above the conveyor belt (facing the conveyor belt) and placed horizontally to it. Therefore, the depth information is utilized for locating the object in the depth image.

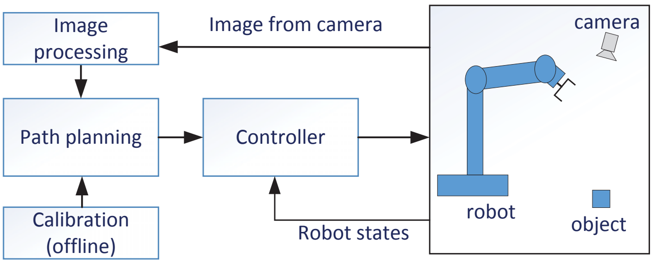


Fig. 5. The overall structure of the algorithm for realtiem sensor-based robot system.

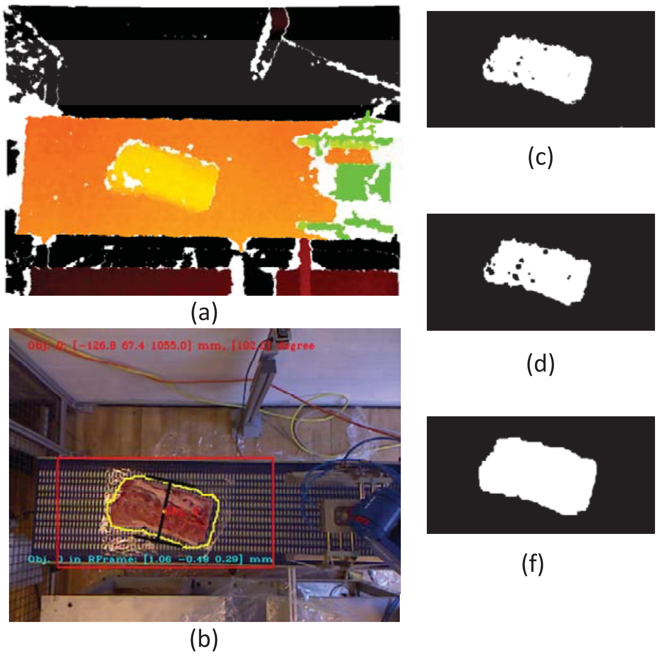
The input image contains the depth information of the object, as show in Fig. 6 (a). After receiving the image, the thresholding method is applied to distinguish between the background and the object. A binary image is resulted after the thresholding algorithm, see Fig. 6 (c). Then, two morphology operators dilation and erosion are applied to remove noise. The results are shown in Fig. 6 (d) and (f). The contour of the object is retrieved from the previous step using the algorithm proposed in work [29]. The contour detection algorithm provides object information in the image plane, including the center location, the orientation and the area, see Fig. 6 (b). In this work, the area of the object is also used for the purpose of illuminating the disturbance in the image. Only the object with an area within a certain range is considered to be the candidate of the expected object for the task. It has to be mentioned, that in this work only a certain area in the image is selected for searching objects. A rectangular as shown in Fig. 6 (b) is used to highlight the field of interest in the image plane. The object detection algorithm is only applied to the area inside the rectangular. It helps in speeding up the image processing algorithm and suppressing the objects/noises in the background with similar distance to the Kinect sensor as well.

Once the object is determined in the depth image, its position and orientation are mapped to the RGB image for both visualization (see Fig. 6 (b)) and retrieving the 3D coordinates in the camera frame. Then, the results are used for generating 3D position and orientation of the object in the robot base frame.

B. Coordinate Transformation

Locating object in the image plane gives 2D position and orientation of the object, which need to be transferred to 3D pose for the robot control. The coordinate transformation from the image plane to the robot base frame is illustrated here.

Assume that the object is located in the image with the position coordinate $[u, v]$ (referring to the object's center) and the orientation γ. In order to transfer this 2D image


Obj. 0: [-126.8 67.4 1055.0] mm, [102.3] degree
Obj. 0 in RFrame: [1.06 -0.48 0.29] mm

(a)

(b)

(c)

(d)

(f)

Fig. 6. Image processing results: (a) depth image; (b) detected object after contour detection algorithm; (c) result of thresholding; (d) result obtained after erosion; (f) result obtained after dilation.

coordinate $[u, v]$ to 3D coordinates $[X_c, Y_c, Z_c]$ in the camera frame, an offline calibration is required to determine the intrinsic parameters of the camera. Then, the 3D coordinates of the object in the camera frame can be obtained by Eq.(1) considering a pinhole camera model.

$$X_c = \frac{u - p_x}{f_x} Z_c, \qquad Y_c = \frac{v - p_y}{f_y} Z_c, \tag{1}$$

where $[p_x,\ p_y]$ denotes the principle coordinates, and f_x, f_y denote the focal length along the x, y directions. These four intrinsic parameters about the camera can be obtained through the offline camera calibration. As the Kinect sensor is mounted horizontally parallel to the conveyor belt, the distance Z_c from the camera to the object along the camera optical axis is known. Therefore, the position of the object can be calculated through Eq.(1) with $[u, v]$ determined online and $[p_x, p_y, f_x, f_y]$ determined offline. For mapping the 2D orientation to 3D orientation, only the rotation around the camera optical axis γ has to be determined online, since the loin lays on the conveyor belt and can only rotate around the optical axis of the camera. The four parameters $[X_c, Y_c, Z_c, \gamma]$ are then transferred to the robot base frame by Eq. (2).

$$\begin{aligned} P_{obj} &= T_{b2c} * \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & X_c \\ \sin(\gamma) & \cos(\gamma) & 0 & Y_c \\ 0 & 0 & 1 & Z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ P_{obj} &= \begin{bmatrix} r_{11} & r_{12} & r_{13} & X_r \\ r_{21} & r_{22} & r_{33} & Y_r \\ r_{31} & r_{32} & r_{33} & Z_r \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{2}$$

where r_{ii}, $i = 1, 2, 3$ denotes the element of the resulted rotation matrix, and $[X_r, Y_r, Z_r]$ is the object position coordinates in the robot base frame. The position and orientation of the object in the robot based frame are passed to the next step for generating the desired path for the robot control.

C. Path Planning

The path planning is divided into two parts: the online part and the offline part. In order to grasp the object from the conveyor belt, the gripper mounted on the robot arm has to track the motion of the object and grasp it from the conveyor belt at the correct time. The three position parameters x, y, z and three orientation parameters α, β, γ (Euler angles calculated from the rotation matrix) need to be determined during path planning. The robot arm is placed with its z axis parallel to the optical axis of the Kinect sensor. With the height of the conveyor belt is known, the height z of the gripper for tacking and grasping is defined offline. The position of the object in the $x-y$ plane as well as the rotation γ around the z-axis needs to be configured online according to each piece of work. The other two rotational angles α, β are defined offline. The online and offline path planning are summarized in Tab. II, where the pick and place task is divided into four subtasks: tracking, grasping, lifting and hanging. The underline parameters require online visual feedback, while the rest parameters are achieved by offline path planning. The parameters that are not appear in the table remain constant.

TABLE II
OFFLINE AND ONLINE PATH PLANNING.

	degrees of freedom
tracking	$\underline{x}, \underline{y}, z, \underline{\gamma}$
grasping	$\underline{x}, \underline{y}, z$
lifting	z, γ
hanging	$x, y, z, \alpha, \beta, \gamma$

D. Robot Control

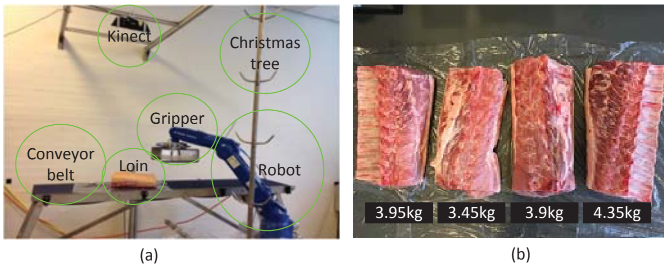
Once the path for the robot arm control is determined by the previous step, the control commands including the three positions and three orientations are communicated to the controller FS100 for moving Motoman MH5L robot arm.

IV. EXPERIMENT

In this part, the algorithm proposed in section III is implemented for the pick and place task.

A. Experimental Setup

The overall experimental setup is shown in Fig. 7(a). The Kinect sensor is mounted above the conveyor belt with a height of 1.15 m. The conveyor belt transfers the loin from the left side to the right side in the figure with a velocity of about 0.4 m/s. The metal Christmas tree with 16 hooks is standing on the left side of the robot arm. Only two hooks within the workspace of the robot arm are chosen as goal positions for hanging. Fig. 7(b) shows four pieces of loin used in the experiments. The loins have different weight (between 3.45 kg



(a)

(b)

Fig. 7. Experimental setup: (a) the experimental platform; (b) four pieces of loin as target objects used in the experiment.

and 4.35 kg) and size (length between 29 cm and 35 cm, width between 15 cm and 17 cm, height between 7 cm and 11 cm).

The camera intrinsic parameters resulted from offline calibration is given in Tab. III. For the calibration between the Kinect sensor and the robot arm, a world frame within the field view of the sensor is assigned to the conveyor belt. A chess board is used to obtain the extrinsic parameters between the camera and the world frame. The transformation matrix from the world frame to the robot base frame is resulted from manual measurement. From these two steps the transformation matrix required in Eq. (2) for converting coordinates in camera frame to robot frame is calculated by Eq. (3).

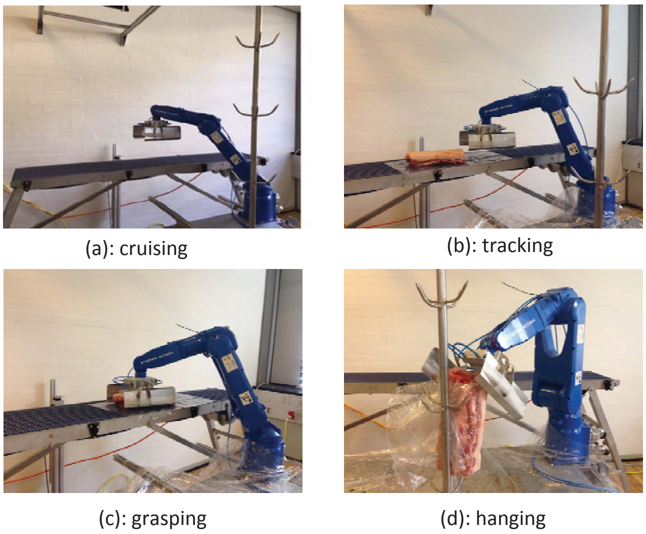
TABLE III
INTRINSIC PARAMETERS OF KINECT RGB CAMERA.

p_x	p_y	f_x	f_y
317.98	216.75	544.06	544.23

$$
\begin{aligned}
T_{w2c} &= \begin{bmatrix} -0.009 & 0.995 & -0.106 & 0.193 \\ 1.010 & -0.007 & 0.007 & 0.081 \\ 0.006 & -0.106 & -0.995 & 1.258 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
T_{b2w} &= \begin{bmatrix} 0 & -1.0 & 0 & 0.49 \\ 1.0 & 0 & 0 & 0.955 \\ 0 & 0 & 1.0 & -0.083 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
T_{b2c} &= T_{b2w} * T_{w2c} \\
&= \begin{bmatrix} -1.01 & 0.007 & -0.007 & 0.409 \\ -0.009 & 0.995 & -0.1060 & 1.148 \\ 0.006 & -0.106 & -0.995 & 1.175 \\ 0 & 0 & 0 & 1 \end{bmatrix},
\end{aligned}
\tag{3}
$$

where T_{w2c} denote the coordinates transformation between the camera and the world frame, and T_{b2w} denotes the coordinates transformation between the world and the robot frame.

B. Experimental results



(a): cruising

(b): tracking

(c): grasping

(d): hanging

Fig. 8. Snapshots obtained during the experiment for pick and place task: (a) cruising and waiting for object; (b) tracking with visual feedback; (c) grasping object from the conveyor belt; (d) hanging the object to the desired goal position.

Snapshots of the experiment are shown in Fig. 8. Fig. 8(a) shows the system in the cruising status. At this stage, the robot is at the initial pose, and the vision system is ready for capturing the object while it enters the field of view. Fig. 8(b) shows a snapshot when the robot arm is tracking the motion of the loin on the conveyor belt. Fig. 8(c) gives a glimpse of grasping object from the conveyor belt, while Fig. 8(d) shows the final step of hanging the object onto the desired hook on the Christmas tree.

The six joint angles during the pick and place task is shown in Fig. 9. The time it takes to finish the tracking, grasping, lifting, moving towards goal and hanging is about 5 s. After it finished the placing task, the robot arm moves back to its home

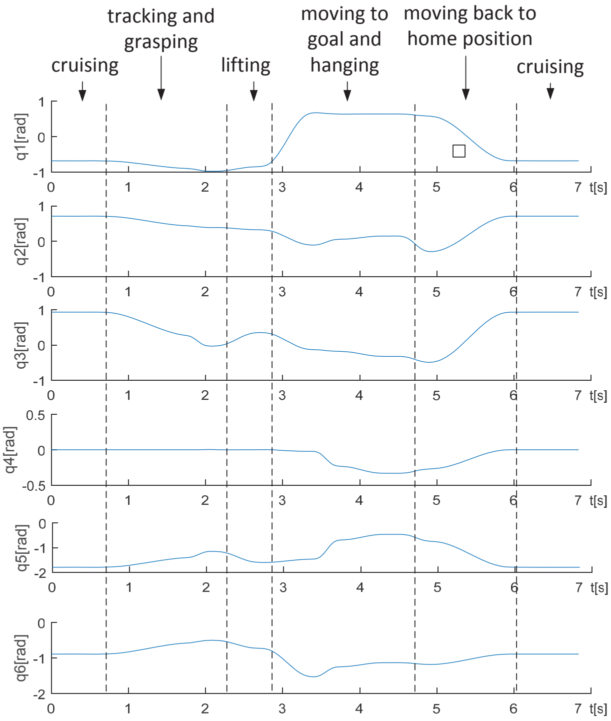


Fig. 9. Robot joint angles during pick and place task.

position and is ready for picking up the next loin. The system has tested with the four pieces of loin shown in Fig. 7(b). It ran in total 20 trials with a success rate 85%. The failure is caused mainly by the low friction between the gripper and the loin.

It has to be mentioned that the speed of the whole system can be improved by reducing the delay from the image processing, optimizing the path planning algorithm and the platform setup (positioning of robot, conveyor belt and the Christmas tree).

V. Conclusion

In this work the realtime visual information is utilized as feedback for robot control to deal with the object variety. As a case study the pick and place task of loin in slaughterhouse is selected for system test. The Kinect sensor is applied in this task to capture the information of the current loin appearing on the conveyor belt. An path planning algorithm is proposed combining the offline and online information of the system. A lab-scale experiment is designed to evaluate the system. The experimental results demonstrate a relatively high success rate 85% after testing different objects.

The developed system provides a generic solution for pick and place task. As the image processing, the path planning and robot control are integrated in ROS, the results of this work can be utilized with little effort for similar applications on different hardware platform. The future work is concerned with improving the system performance through fault diagnosis and extending the system with force/torque sensor for object hanging considering different object types.

Acknowledgment

This work is supported by the Danish Innovation Project *RealRobot*. The authors would like to thank the partners from the Department of the Computer Science DIKU `http://www.diku.dk` who assisted the image processing part, and the Danish Technological Institute DMRI `http://www.dti.dk/dmri` for providing the pneumatic gripper.

References

[1] H. Li, H. Wu, L. Lou, K. Kühnlenz, and O. Ravn, "Ping-pong robotics with high-speed vision system," *2012 12th International Conference on Control, Automation, Robotics and Vision, ICARCV 2012*, vol. 2012, no. December, pp. 106–111, 2012.

[2] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of intelligent and robotic systems*, vol. 53, no. 3, pp. 263–296, 2008.

[3] H. H. Lund, "Modular robotics for playful physiotherapy," in *Rehabilitation Robotics, 2009. ICORR 2009. IEEE International Conference on*, pp. 571–575, IEEE, 2009.

[4] B. Gates, "A robot in every home," *Scientific American*, vol. 296, no. 1, pp. 58–65, 2007.

[5] T. Brogårdh, "Present and future robot control developmentan industrial perspective," *Annual Reviews in Control*, vol. 31, no. 1, pp. 69–79, 2007.

[6] H. Wu, L. Lou, C.-C. Chen, S. Hirche, and K. Kühnlenz, "A framework of networked visual servo control system with distributed computation," in *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, pp. 1466–1471, IEEE, 2010.

[7] J. Hill and W. Park, "Real time control of a robot with a mobile camera," in *the 9th international Symposium on Industrial Robots*, pp. 233–246, 1979.

[8] K. Hashimoto, *Visual Serving: Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*, vol. 7. World scientific, 1993.

[9] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006.

[10] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 5, pp. 651–670, 1996.

[11] F. Chaumette and S. Hutchinson, "Visual servo control. ii. advanced approaches [tutorial]," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 1, pp. 109–118, 2007.

[12] D. Kragic, H. I. Christensen, *et al.*, "Survey on visual servoing for manipulation," *Computational Vision and Active Perception Laboratory, Fiskartorpsv*, vol. 15, 2002.

[13] H. Wu, W. Tizzano, T. Andersen, N. Andersen, and O. Ravn, *Hand-Eye Calibration and Inverse Kinematics of Robot Arm using Neural Network*, pp. 581–591. Springer, 2013.

[14] H. Wu, L. Lu, C.-C. Chen, S. Hirche, and K. Khnlenz, "Cloud-based networked visual servo control," *I E E E Transactions on Industrial Electronics*, vol. 60, no. 2, pp. 554 – 566, 2013.

[15] Motoman Product Overview, *http://www.motoman.co.uk*.

[16] M. Quigley, J. Faust, T. Foote, and J. Leibs, "Ros: an open-source robot operating system,"

[17] Robot Operating System, *http://wiki.ros.org/*.

[18] Ros Industrial Program, *http://wiki.ros.org/Industrial*.

[19] FS100 controller datasheets, *http://www.motoman.com/datasheets/fs100 controller.pdf*.

[20] Kinect sensor, *https://msdn.microsoft.com/en-us/library/hh438998.aspx*.

[21] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568, ACM, 2011.

[22] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect.," in *BmVC*, vol. 1, p. 3, 2011.

[23] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling," in *Advanced Robotics (ICAR), 2015 International Conference on*, pp. 388–394, IEEE, 2015.

[24] R. A. El-laithy, J. Huang, and M. Yeh, "Study on the use of microsoft kinect for robotics applications," in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pp. 1280–1288, IEEE, 2012.

[25] M. R. Andersen, T. Jensen, P. Lisouski, A. K. Mortensen, M. K. Hansen, T. Gregersen, and P. Ahrendt, "Kinect depth sensor evaluation for computer vision applications," *Technical Report Electronics and Computer Engineering*, vol. 1, no. 6, 2015.

[26] OpenKinect, *https://openkinect.org*.

[27] Danish Technological Institute DMRI, *http://www.dti.dk/dmri*.

[28] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[29] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.