Robotica (2018) volume 36, pp. 241–260. © Cambridge University Press 2017
doi:10.1017/S0263574717000339

Robot-to-human feedback and automatic object grasping using an RGB-D camera–projector system

Jinglin Shen† and Nicholas Gans‡,*

†*Apple Inc., Cupertino, CA, USA. E-mail: jinglinshen@gmail.com*
‡*Department of Electrical Engineering, University of Texas at Dallas, Dallas, TX, USA*

(Accepted July 20, 2017. First published online: August 23, 2017)

SUMMARY

This paper presents a novel system for human–robot interaction in object-grasping applications. Consisting of an RGB-D camera, a projector and a robot manipulator, the proposed system provides intuitive information to the human by analyzing the scene, detecting graspable objects and directly projecting numbers or symbols in front of objects. Objects are detected using a visual attention model that incorporates color, shape and depth information. The positions and orientations of the projected numbers are based on the shapes, positions and orientations of the corresponding objects. Users select a grasping target by indicating the corresponding number. Projected arrows are then created on the fly to guide a robotic arm to grasp the selected object using visual servoing and deliver the object to the human user. Experimental results are presented to demonstrate how the system is used in robot grasping tasks.

KEYWORDS: Human–robot interaction; Visual servoing; Object grasping.

1. Introduction

Robots are increasingly used to assist daily activities in home or workplace environments. There are many advantages to using robots in applications such as home service, rehabilitation and assistant living, etc.[1,2,3]. By definition, these assistive applications require human–robot interaction or human-in-the-loop systems, including methods to pass information between the robot and human. Human input can help robots visually locate objects[4], select grasping targets[5], plan motions,[6] etc.

Object grasping and retrieval is one key assistive-robot task, and vision sensors are commonly used for guidance and planning in object grasping. Visual sensors can observe human operators[7] and recognize gestures[8] to accept commands. For example, a robot can obtain information about the grasping target from human input (e.g. object location, grasping directions, etc.). Alternately, vision-based robotic systems can analyze an environment to independently detect, recognize and localize objects. Therefore, less input about the grasping targets is needed from the users, assuming that the scene analysis is reliable. In this work, a vision system is used to detect and locate objects that are possibly desired grasping targets.

Vision-based object detection varies depending on the application. If the object model is known beforehand, object detection is usually done by matching features using key point-based methods such as SIFT[9], or using region-based feature comparison[10] and template matching[11] methods. Alternately, the system looks for any object in the scene that satisfies certain criterion. For example, ref. [12] presents a system that looks for visually attractive object in the environment using a visual attention model based on the visual criteria human observers find stimulating. The saliency model proposed by Itti[13] is one of the most widely used visual attention models.

Accurate depth information is often critical for vision-guided robotic grasping. Algorithms have been developed for 3D reconstruction based on stereo depth maps[14] or structured light[15]. In recent years, low-cost RGB-Depth (RGB-D) cameras (such as the Microsoft Kinect) have become prevalent.

* Corresponding author. E-mail: ngans@utdallas.edu

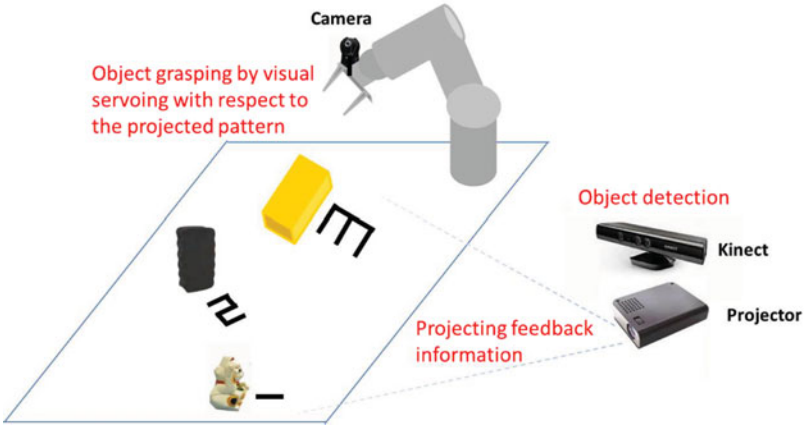

Camera
Object grasping by visual servoing with respect to the projected pattern
Object detection
Kinect
Projector
Projecting feedback information
3
2
1

Fig. 1. Structure of the RGB-D camera–projector system.

Several robotic systems have used RGB-D cameras for interpreting the target scenes, human motions and human intentions[16, 17, 18].

In human–robot interaction, it is critical that the robot can indicate its understanding of the user's commands or convey its knowledge on the operating environments. Despite many interfaces that have been developed, enabling robots to give intuitive feedback to the human remains an open problem. Computer monitors may be not convenient for the users to observe, as they may need to move, or the monitor can block their vision. Moreover, it may not be feasible to have a monitor for some mobile robotic platforms. Therefore, other interfaces may be better. In ref. [19], a robot avatar can provide gesture feedback to inform the user that he/she needs to repeat a command due to classification failure. In ref. [20], a "smart table" is presented that can highlight detected objects from underneath. However, this requires special hardware, and the only information the system provides is the position of the objects.

In this work, we propose a novel system consisting of an RGB-D camera and a Digital Light Processing (DLP) projector to provide interactive robot-to-human feedback information by projecting patterns in front of the detected objects. The structure of the proposed system is shown in Fig. 1. Assumptions are made that the objects are all located on a flat table surface in an indoor environment. The system first detects visually attractive objects using a modified saliency model with depth added as a feature. After the objects are detected, the shape, position and orientation of the objects are estimated from the RGB-D image. Then, the system projects a number onto the table in front of each object, the size and orientation of which is determined according to the size and orientation of the corresponding object. The projected numbers for all the detected objects are displayed simultaneously on the table. The human can then inform the robot of the desired grasping target, such as by pressing a numbered key or speaking to a speech recognition system.

Having selected a desired target for the robot to grasp and retrieve, visual servoing based on a projected pattern is used to control the robot arm to reach its grasping position. The goal pose of the camera is determined by the desired grasping direction and the eye-to-hand transformation. The goal image is generated by predicting what the pattern should look like in the camera view at its goal pose. The novel contribution of this method lies in that the projected pattern for visual servoing is generated according to the desired grasping pose.

Projectors have previously been used to allow users to interact with computers or robots. For example, in ref. [21], users can control a web browser or desktop applications by pointing toward a projected computer screen. A similar system is proposed in ref. [22], where users interact with computers by touching the items projected on the screen. The robotic system equipped with two RGB-D cameras and a projector in ref. [23] can adjust the projection according to human gestures. The system here is unique in that it focuses on allowing the robot to communicate with the human.

We previously developed a stereo camera–projector system that projects "spotlight" patterns on objects detected with a salience model that includes a disparity map[24, 25]. Based on human feedback, a robot will then grasp a desired object. The trifocal tensor constraints are used to match the detected objects in two views and map the feedback patterns to the projector view. The system was limited

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

to projecting spotlights on one object at a time, and the spotlight patterns only indicated the position and estimated size of the objects. The system presented in this work is simpler in design and is capable of projecting unique patterns for each detected object, simultaneously. Furthermore, the system presented here provides more information about the detected objects, such as their positions, sizes and orientations. Last, the projected patterns are used to guide a vision-based controller, which improves the accuracy of the grasping task.

This work uses Simultaneous Image and Position Visual Servoing (SIPVS), which we proposed in ref. [26], to control the robot. Like Position-Based Visual Servoing (PBVS), Image-Based Visual Servoing (IBVS)[27] and combined methods (e.g. ref. [28]), SIPVS requires a goal image and depends on matching and tracking image features. However, the accuracy and efficiency of feature matching and tracking algorithms is poor in cluttered environments. The system proposed here can use the projector to create targets in the environment with known appearance at the goal configuration.

In ref. [29], a projector displays patterns on non-textured objects to aid visual servoing. However, the method can only position the camera at a predefined position to the object, and the features tend to leave the camera field of view when the robot gets too close to the object. In the method proposed in this work, the pattern is projected on the table near the target object, so the features are independent of the object appearance. Since the shapes and structures of the projected patterns are fully controlled, image features can be easily extracted and matched.

The paper is organized as follows. In Section 2, a brief background is provided. In Section 3, we discuss object detection and feedback pattern projection using the proposed RGB-D camera–projector system. Grasping by visual servoing with respect to projected patterns is investigated in Section 4. Experimental results are represented in Section 5. Finally, Section 6 concludes the paper and proposes future works.

2. Background

2.1. Camera and projector models

Define an inertial world Cartesian reference frame $\mathcal{F}_w$ and camera frame $\mathcal{F}_c$. The coordinates of a 3D point are defined as $\mathbf{M} = [X, Y, Z, 1]^T$ using homogeneous coordinates measured in $\mathcal{F}_w$. Its corresponding image plane projection is defined as $\mathbf{m} = [x, y, 1]^T$ as measured in $\mathcal{F}_c$. Under the pinhole camera model, the projection relation between $\mathbf{M}$ and $\mathbf{m}$ is given by

$$\mathbf{m} = \mathbf{PM} = \begin{bmatrix} \frac{1}{Z_c} & 0 & 0 & 0 \\ 0 & \frac{1}{Z_c} & 0 & 0 \\ 0 & 0 & \frac{1}{Z_c} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \mathbf{M},$$

where $\mathbf{R} \in SO(3)$ and $\mathbf{T} \in \mathbb{R}^3$ are rotation and translation of $\mathcal{F}_c$ with respect to $\mathcal{F}_w$, $\mathbf{P} \in \mathbb{R}^{3\times4}$ is the projection or camera matrix, and Z_c is the Z coordinate of the point as measured in $\mathcal{F}_c$. The homogeneous coordinates $\mathbf{m}$ is mapped to pixel coordinates in the image by

$$\mathbf{p} = \mathbf{Km},$$

where $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the calibration matrix of the camera.

A projector can be modeled as a dual of a pinhole camera. For a camera, a point on a 3D surface in space projects to a 2D image point. While for a projector, a 2D image point $\mathbf{m}$ is projected as a line $\mathbf{L}$ in 3D space (i.e. $\mathbf{L}$ is the pre-image of $\mathbf{m}$). The intersection of the line $\mathbf{L}$ with a surface results in a point of light on the surface. To distinguish between the directions of projection, we define projecting from 3D space to 2D image plane to be “projecting in” (i.e. projection of camera), and projecting from 2D image to 3D space to be “projecting out” (i.e. projection of projector).

2.2. Visual attention models and the saliency model

Visual attention models simulate the selective visual attention of human beings. So called *top-down models* consider the effect of cognitive factors on visual attention (e.g. knowledge). *Bottom-up models* simulate only the spontaneous visual cortex response. Visual attention models have been used

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

in robotic applications such as object manipulation[30], object tracking[31] and simultaneous localization and mapping[32].

A popular *bottom-up model* is Itti's saliency model[13]. The model represents the attractiveness of visual stimuli as a gray-scale image called a saliency map. Regions of likely interest to a human will receive a high intensity in the saliency map. A saliency map is typically the combination of different feature maps, depending on the specific applications. In ref. [13], the three features used to generate the saliency map are intensity, color and orientation. Other features have been used in visual attention models such as entropy[33] and depth[34]. A normalization step is often applied to convert all feature maps to the same scale.

An adaptive feature map weighting scheme based on information theory was proposed in ref. [35]. Such a weighting scheme is good for analyzing dynamic scenes. The method defines the probability that an event is observed from a feature map $\mathcal{M}$ as

$$p(\mathcal{M}) = \frac{\sum_{i,j} \mathcal{M}_\tau(i, j)}{\sum_{i,j} \mathcal{M}(i, j)},$$

where $\mathcal{M}(i, j)$ is the value of the i–jth pixel in $\mathcal{M}$ and $\mathcal{M}_\tau(i, j)$ is the pixel value if higher than a threshold τ (else 0). The weight of the feature map is decided by the amount of information obtained from the map

$$W(\mathcal{M}) = -\log(p(\mathcal{M})).$$

The final saliency map $\mathcal{S}$ is given by

$$\mathcal{S} = \sum_{i=1}^{n} W(\mathcal{M}_i)\mathcal{M}_i,$$

where n is the number of features used in the model.

2.3. Visual servoing

Visual servoing is feedback control of a mechanical system using information from visual sensors. The two basic visual servoing methods are PBVS and IBVS[27]. In PBVS, the feedback control is established using the pose error (i.e. the difference between the current position and orientation and the goal position and orientation of a camera). The homography matrix or essential matrix[36] can be used to estimate the pose error between current and goal pose from current and goal images. In IBVS, the feedback control law is derived in the image space. For each feature point, the image error is given by the differences between its coordinates in the current frame and in the goal frame.

PBVS and IBVS each have advantages and drawbacks. In PBVS, the image error is not controlled. Therefore, it is possible for the feature points to move out of the camera field of view, which can cause visual servoing to fail. Conversely, the pose error is not controlled in IBVS so that the robot may reach its joint limit. To overcome the problems of PBVS and IBVS, a number of hybrid and switching control methods were developed, e.g. refs. [28, 37, 38, 39].

We proposed SIPVS in ref. [26]. SIPVS combines the control laws of PBVS and IBVS and stabilizes the pose and image error at the same time. Adaptive depth estimation eliminates the need for prior knowledge of depth or the need to measure it. In this work, SIPVS is used in object grasping operations to guarantee that the desired grasping pose can be achieved.

3. Robot-to-human Feedback Using the RGB-D Camera–Projector System

The proposed system detects visually salient objects using a modified saliency model with depth map as a feature and provides intuitive robot-to-human feedback information by directly projecting patterns in front of detected objects. Numbers are projected in front of each of the detected objects to indicate what objects the system has detected and determined may be of interest to a human user. Positions and directions of the projected numbers specify the detected positions and facing directions

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

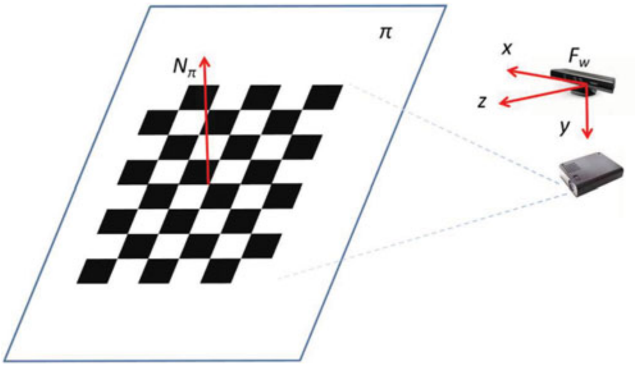


Fig. 2. Calculating the table plane parameters.

of the corresponding objects. The projected patterns are updated repeatedly to adjust to changes in the scene.

3.1. System calibration

A calibration step is first performed. The proposed RGB-D camera–projector system has four "views": the Kinect RGB camera view, the Kinect IR camera view, the Kinect IR projector view and the DLP projector view. The extrinsic parameters must be accurately determined between each pair of views, and intrinsic parameters must be determined for the RGB and IR cameras and the DLP. The extrinsic parameters between the Kinect RGB and IR camera are determined using the calibration library in ref. [40]. RGB camera intrinsic calibration is done using the Matlab Camera Calibration Toolbox[41]. DLP intrinsic calibration is done using the method of Falcao et al.[42], which is also based on ref. [41]. Extrinsic calibration between the DLP and the RGB camera is also similar to ref. [41], and achieved by projecting a checkerboard pattern in view of the camera.

3.2. Recovering the position of the table surface

The proposed system projects feedback patterns on the table in front of the detected objects. The coefficients of the table plane with respect to the system need to be determined. We assume that the RGB-D camera–projector system is located at a fixed position in front of the table, as illustrated in Fig. 2. The world frame $\mathcal{F}_w$ is chosen to be coincident with the Kinect RGB camera frame. A checkerboard image is projected on the table, and n inner corners $\mathbf{m}_i = [x_i, y_i, 1]^T \in \mathbb{R}^3$, $i = \{1 \dots n\}$ are extracted in the camera view. The corresponding 3D points $\mathbf{M}_i$ in $\mathcal{F}_w$, which are the intersections of the pre-images of $\mathbf{m}_i$ and the table plane, are given by

$$\mathbf{M}_i = [x_i z_i, y_i z_i, z_i, 1]^T \in \mathbb{R}^4, i \in \{1, \dots, n\}, \quad (1)$$

where the depth z_i is obtained from the depth map generated by the Kinect sensor. Since all points $\mathbf{M}_i$, $i \in \{1, \dots, n\}$, are located in the same plane, they satisfy the equation

$$\mathbf{M}_i^T \boldsymbol{\pi} = 0,$$

where $\boldsymbol{\pi} = [\pi_1, \pi_2, \pi_3, \pi_4]^T \in \mathbb{R}^4$ is the coefficients of the table surface plane. The normal vector of the table plane is given by the first three elements in $\boldsymbol{\pi}$ as $\mathbf{N}_\pi = [\pi_1, \pi_2, \pi_3]^T \in \mathbb{R}^3$. Define a matrix $\mathbf{P} = [\mathbf{M}_1 \dots \mathbf{M}_n]^T$, then the equation

$$\mathbf{P}\boldsymbol{\pi} = 0$$

holds. The null space of $\mathbf{P}$ estimates the coefficients $\boldsymbol{\pi}$ of the table plane up to a scale factor. A least squares solution can be found via singular value decomposition of $\mathbf{P}$. Note that the same set of feature points $\mathbf{M}_i$, $i \in 1, \dots, n$ for calculating the table plane parameters can also be used for extrinsic calibration between the Kinect RGB camera and the projector, so the two steps can be combined.

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

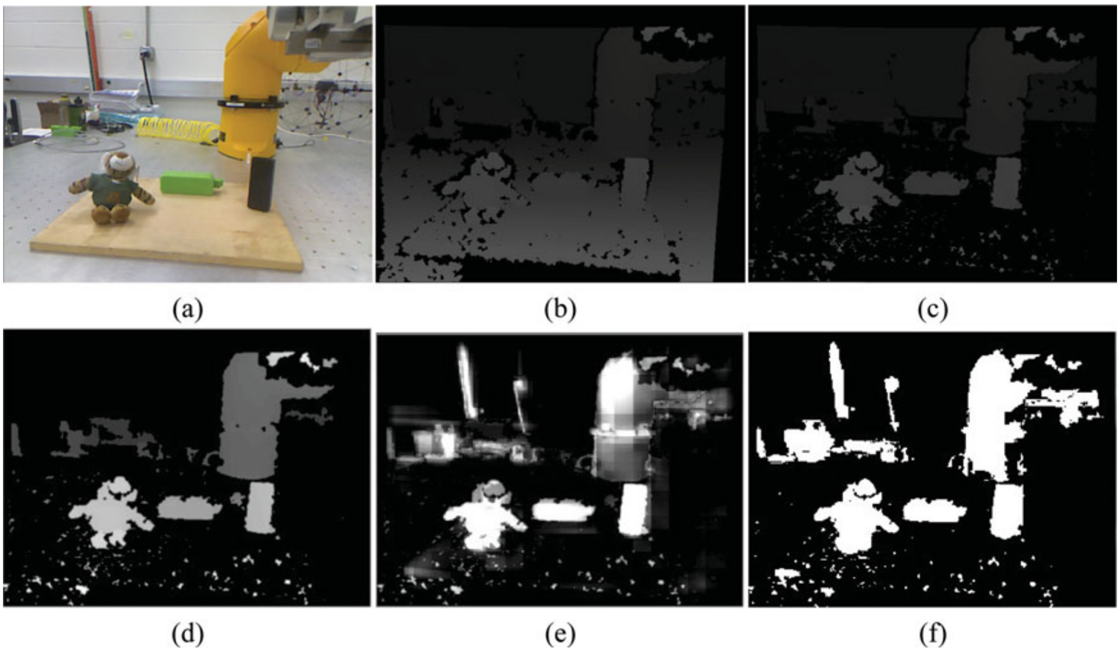


Fig. 3. Pre-processing on the disparity map and saliency map generation.

3.3. Saliency-based object detection

The system provides feedback information for robot–human interaction. The system first detects objects that are likely to attract the attention of human users based on a modified saliency model. In addition to the commonly used intensity, color and orientation features, a disparity map from the RGB-D camera is added as a feature to generate the saliency map. The disparity map helps isolate objects that are out of the plane of the table and can be grasped by the robot. The information theory-based weighting scheme described in Section 2.2 is used to determine the weight of each feature map, and the object locations are estimated by the conspicuous regions in the thresholded saliency map.

Since the feedback patterns are projected on the table surface in front of the detected objects, both the Kinect and the projector need to be tilted toward the table. However, this will cause the table surface to have relatively large values in the disparity feature map. The example given in Fig. 3 shows the effect of the table surface on the disparity map and the saliency map. A scene with three objects and the corresponding disparity map is given in Fig. 3(a) and (b), respectively. It can be seen from Fig. 3(b) that the objects on the table have similar disparity values to the table surface.

To extract useful information from the disparity map, a pre-processing step is applied to remove the table surface and the background before adding the disparity map to the saliency model. Table surface removal is achieved by setting equal to zero each point $\mathbf{m}_d$ in the disparity map with a corresponding 3D point $\mathbf{M}_i$ satisfying

$$\mathbf{M}_i^T \boldsymbol{\pi} <= 0,$$

where $\boldsymbol{\pi}$ is the coefficients of the table plane representation found in Section 3.2. After removing the table surface, the background wall is also removed by finding the maximum value in the histogram of the disparity map. Alternately, the maximum range of the projector and/or the reachable space of the robot can be used to define a disparity value threshold.

Again consider the example given in Fig. 3. Figure 3(c) and (d) show the disparity map after the table surface and the background wall is removed, respectively. The disparity map in Fig. 3(d) better indicates the salient objects in the scene, compared to the original disparity map. The saliency map built with the disparity map in Fig. 3(d) is given in Fig. 3(e). Figure 3(f) shows the thresholded saliency map. After segmentation, the segments in the disparity map with areas larger than a threshold represents the estimated object locations.

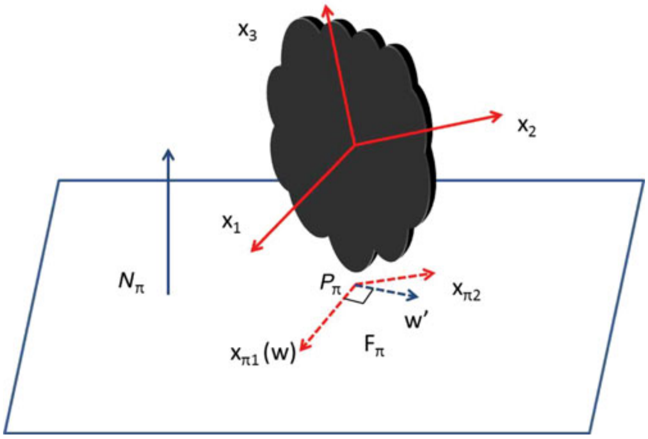


Fig. 4. Determining the orientation of the projected pattern for an object.

3.4. Projecting feedback patterns on the table

There are two purposes for projecting patterns in front of the detected objects: (1) to provide intuitive information to the human users about the robot's knowledge of the detected objects; (2) to provide visible features to match and track during the grasping operations. The position, size and direction of the patterns are decided by analyzing the 3D shape information of the detected objects.

A point cloud of the scene is first obtained from the Kinect. The objects are segmented from the point cloud using the saliency map segmentation result as a mask. Define a set of 3D points in homogenous coordinates

$$O = \{\mathbf{M}_i | \mathbf{M}_i \in \mathbb{R}^4 \text{ belongs to a detected object}\}$$

and a set of 2D points in the saliency map

$$o = \{\mathbf{m}_i | \mathbf{m}_i \in \mathbb{R}^3 \text{ is in the region representing the object}\}.$$

For a 3D point $\mathbf{M}_j$, we have $\mathbf{M}_j \in O$ if the corresponding 2D point $\mathbf{m}_j \in o$. Since the saliency model is a coarse estimation of the object region, the point cloud segmentation typically contains outliers. A refined estimate $\bar{O}$ of the object is given by removing the outliers in O using the Mahalanobis distance. The estimate $\bar{O}$ is defined as a set of points

$$\bar{O} = \{\mathbf{M}_i | \mathbf{M}_i \in O, D_m(\mathbf{M}_i) < T\},$$

where $D_m(\mathbf{M}_i)$ is the Mahalanobis distance of $\mathbf{M}_i$ to the set of points O, and T is a tunable threshold. Note that $\bar{O}$ does not reflect the complete shape of the object, since the point cloud is only available for the side of the object that is visible to the Kinect.

We assign a Cartesian frame $\mathcal{F}_o$ to the object, with the origin at the mean of the points in $\bar{O}$. The axes of $\mathcal{F}_o$, denoted as unit vectors $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, are the major axes of object and are determined as the eigenvectors of the covariance matrix of the points in $\bar{O}$ (i.e., we perform principal component analysis). The axis $\mathbf{x}_3$ is assigned as the axis vector with the smallest inner product with the table surface normal. Note that this process works without modification for symmetric point objects such as cylinders and spheres.

To locate the position of the projected feedback pattern with respect to the object, a reference point $\mathbf{p}_\pi$ where the object contacts the table is determined by projecting the origin of $\mathcal{F}_o$ into the table plane $\boldsymbol{\pi}$. The orientation of the projected pattern is determined by projecting $\mathcal{F}_o$ axes $\mathbf{x}_1$ and $\mathbf{x}_2$ into the table plane to give vectors $\mathbf{x}_{\pi 1}$ and $\mathbf{x}_{\pi 2}$ by

$$\mathbf{x}_{\pi i} = \mathbf{N}_\pi \times (\mathbf{x}_i \times \mathbf{N}_\pi), i = 1, 2$$

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

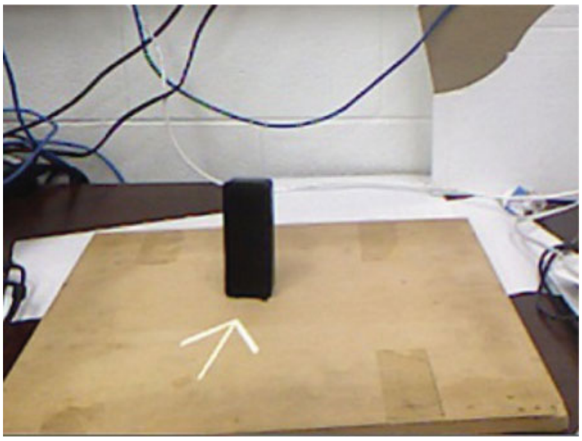


Fig. 5. An arrow projected in front of a graspable object.

as illustrated in Fig. 4. The facing direction $\mathbf{w}$ of the object is chosen as either $\mathbf{x}_{\pi 1}$ and $\mathbf{x}_{\pi 2}$ according to

$$\mathbf{w} = \begin{cases} \mathbf{x}_{\pi 1} & \text{if} \quad \mathbf{p}_{\pi}^{T}(\frac{\mathbf{x}_{\pi 1}}{||\mathbf{x}_{\pi 1}||}) >= \mathbf{p}_{\pi}^{T}(\frac{\mathbf{x}_{\pi 2}}{||\mathbf{x}_{\pi 2}||}) \\ \mathbf{x}_{\pi 2} & \text{if} \quad \mathbf{p}_{\pi 1}^{T}(\frac{\mathbf{x}_{\pi 1}}{||\mathbf{x}_{\pi 1}||}) < \mathbf{p}_{\pi}^{T}(\frac{\mathbf{x}_{\pi 2}}{||\mathbf{x}_{\pi 2}||}). \end{cases} \tag{2}$$

The coordinates of the reference point $\mathbf{p}_{\pi}$ gives a vector from $\mathcal{F}_{w}$ to $\mathbf{p}_{\pi}$. Therefore, the condition in Eq. (2) means that the vector $\mathbf{w}$ is chosen to be either $\mathbf{x}_{\pi 1}$ or $\mathbf{x}_{\pi 2}$, whichever points closer to the direction of the RGB camera. Assume $\mathbf{w} = \mathbf{x}_{\pi 1}$ in the case of Fig. 4. The vectors $\mathbf{x}_{\pi 1}(i.e., \mathbf{w})$ and $\mathbf{x}_{\pi 2}$ are generally not orthogonal to each other. A vector $\mathbf{w}'$ which is orthogonal to $\mathbf{w}$ can be found by the cross product of $\mathbf{N}_{\pi}$ and $\mathbf{w}$. Define a Cartesian coordinate frame $\mathcal{F}_{\pi}$ with origin $\mathbf{p}_{\pi}$ and two axes $\mathbf{w}$ and $\mathbf{w}'$ in the plane of the table. The third axis is given by $\mathbf{w} \times \mathbf{w}'$.

When there is a single detected object, an arrow pattern is then projected on the table. The arrow consists of three line segments. The tip of the arrow head is set to be a distance d away from the reference point $\mathbf{p}_{\pi}$ along the direction $\mathbf{w}$. The end point of the shaft line segment is set to be of the length of the longest principle axis of the object (i.e. the largest eigenvalue of the covariance matrix of $\bar{O}$) from the tip vertex along the direction $\mathbf{w}$. We set d to one-half the arrow length. The end points of the line segments of the arrow head are chosen to be symmetric about the arrow shaft, e.g. at a length of one half the shaft length and making an angle $\pm 30°$ with the shaft line segment.

Having designed the desired positions of the endpoints in the coordinate frame $\mathcal{F}_{\pi}$, their 3D positions $\mathbf{M}_{ci} \in \mathbb{R}^4$ expressed in the camera frame can be calculated, since the 3D coordinates of the reference point $\mathbf{p}_{\pi}$ is given by Eq. (1), and the two vectors $\mathbf{w}$ and $\mathbf{w}'$ in the camera frame are known. Let the translation and rotation of the projector with respect to the Kinect RGB camera be $\mathbf{T}_p \in \mathbb{R}^3$ and $\mathbf{R}_p \in SO(3)$, which is found in the extrinsic calibration step in Section 3.1. The vertices are mapped to the corresponding points m_{pi} in the projector view by

$$\mathbf{M}_{pi} = \begin{bmatrix} \mathbf{R}_p & \mathbf{T}_p \\ 0 & 1 \end{bmatrix} \mathbf{M}_{ci},$$

$$\mathbf{m}_{pi} = \begin{bmatrix} \frac{1}{Z_{pi}} & 0 & 0 & 0 \\ 0 & \frac{1}{Z_{pi}} & 0 & 0 \\ 0 & 0 & \frac{1}{Z_{pi}} & 0 \end{bmatrix} \mathbf{M}_{pi},$$

where Z_{pi} is the third element in $\mathbf{M}_{pi} = [X_{pi}, Y_{pi}, Z_{pi}, 1] \in \mathbb{R}^4$. A bitmap is generated with a black background and white line segments with the desired end points. Displaying this image from the projector results in the desired appearance of the arrow projected on the table. An example is shown in Fig. 5, with an arrow oriented to point at an eraser.

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

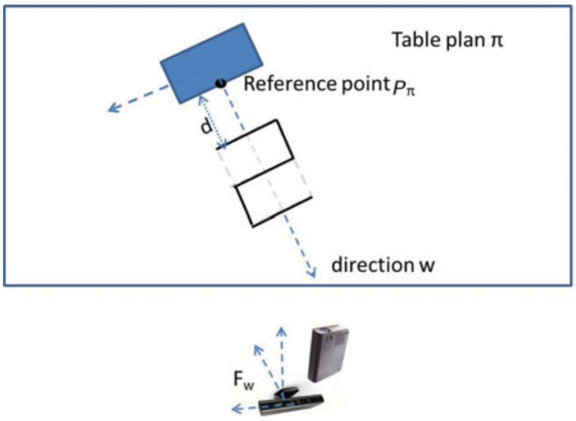
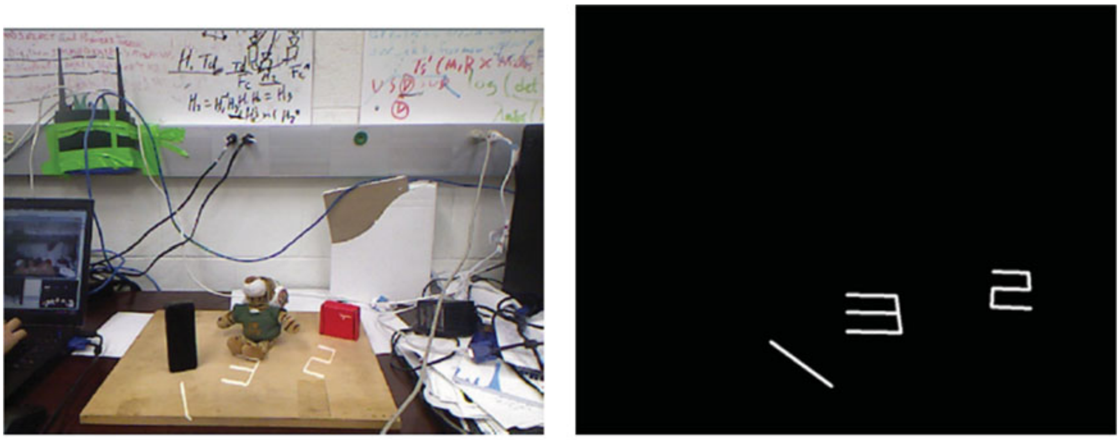


Fig. 6. Projecting a number in front of a detected object.



(a) (b)

Fig. 7. Projecting numbers for multiple detected objects simultaneously.

When there are multiple objects detected by the attention model, we project numeric symbols. Each numeric symbol is rectangular and consists of several line segments. The end points of each segment can be calculated according to the coordinates of $\mathbf{p}_\pi$, the direction $\mathbf{w}$, distance d and the size of the pattern, similar to the way the arrow line segments were determined. This is illustrated in in Fig. 6. Figure 7(a) shows a scene with four objects of different shapes, sizes and orientations. Different numbers are projected on the table, whose appearances are adaptive to the corresponding objects using the method described in this section. Figure 7(b) shows the bitmap that is displayed by the projector to give the desired placement on the table.

To define a grasping target, the system only needs to receive the corresponding number as input from the human user. In our implementation, the user strikes the desired number on a computer keyboard, but it would be straightforward to use a specialized keypad, hand-held device or speech recognition. Then object grasping can be carried using a visual servoing algorithm with respect to the projected pattern.

4. Object Grasping by SIPVS Control

Once a grasping target is specified by the human user, object grasping operations are conducted using SIPVS control. In SIPVS, the image error and the pose error converge simultaneously. By using the eye-in-hand configuration for visual servoing, SIPVS ensures that the image features remain in the

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

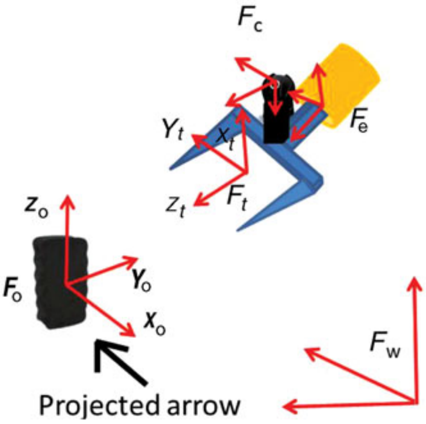


Fig. 8. Visual servoing with respect to a projected pattern.

field of view of the camera and the robot arm does not reach its joint limits when approaching the desired goal pose.

The projected pattern corresponding to a selected grasping target is used to provide image features for visual servoing. Since the position of the projected pattern on the table can be determined by the system, the patterns can be generated automatically to ensure that the features are in the camera view as the robot arm approaches the goal pose. The problem addressed in this section is how to generate the proper goal image that will guide the robot to its grasping position. Note that the goal image and goal end effector pose are defined without the need for knowledge of the robot base frame in the world frame, which is an important contribution of this approach.

4.1. *Definitions and notations*

As illustrated in Fig. 8, a camera and a gripper are attached rigidly to the robot end-effector. This camera on the robot wrist will be used to guide the tool to the desired position with respect to the object. The world frame $\mathcal{F}_w$ is defined as the Kinect RGB camera frame. A tool frame $\mathcal{F}_t$ is assigned to the gripper, with its origin located between the two fingers. The camera frame and the end-effector frame are defined as $\mathcal{F}_c$ and $\mathcal{F}_e$, respectively. The target object frame $\mathcal{F}_o$ is determined as described in Section 3.4. In this section, the notation ${}^{y}\mathbf{H}_x$ is used for any homogenous transformation $\mathbf{H}$ from the reference frame at pose y to the frame at pose x.

Like PBVS and IBVS, the pose error and image error measurements in SIPVS rely on the accuracy of feature matching and tracking in the camera view. In the proposed the system, features are obtained from the projected pattern on the table, which eliminates the dependency of feature extraction on the appearance of the target object. With the camera-gripper configuration shown in Fig. 8, the tool frame $\mathcal{F}_t$ is chosen to be aligned with the object frame $\mathcal{F}_o$ with an additional rotation of 180°about the x-axis, given by the rotation matrix

$$ {}^{o}\mathbf{R}_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}. $$

This results in the gripper being aligned with the object axes and approaching the object along the axis of $\mathcal{F}_o$ that is closest to vertical. Furthermore, this keeps the camera positioned and oriented to view the pattern in front of the object. As discussed in Section 3.4, the position, size and shape of the projected pattern with respect to the object is fully controlled. In an object grasping task, the projected pattern should be generated such that it can be seen by the wrist camera at all times during visual servoing until the robot end-effector reach its goal pose.

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

4.2. Automatic generation of goal pose and goal image

Recall that the object frame $\mathcal{F}_o$ had axes $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ given by the major axes of the object. The pose of an object in $\mathcal{F}_w$ is described by a homogenous rigid body motion ${}^w\mathbf{H}_o$ as

$${}^w\mathbf{H}_o = \begin{bmatrix} {}^w\mathbf{R}_o & {}^w\mathbf{X}_o \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times4},$$

where ${}^w\mathbf{R}_o = [x_1, x_2, x_3] \in SO(3)$, and ${}^w\mathbf{X}_o \in \mathbb{R}^3$ is the origin of $\mathcal{F}_o$ in $\mathcal{F}_w$. Then the desired goal pose of the tool frame ${}^w\mathbf{H}_t \in \mathbb{R}^{4\times4}$ is obtained by

$${}^w\mathbf{H}_t ={}^w \mathbf{H}_o{}^o\mathbf{H}_t = \begin{bmatrix} {}^w\mathbf{R}_o{}^o\mathbf{R}_t & {}^w\mathbf{X}_o \\ 0 & 1 \end{bmatrix}. \tag{3}$$

Knowing the desired goal pose of the tool frame ${}^w\mathbf{H}_t$, the next step is to generate the goal pose of the camera frame ${}^w\mathbf{H}_c$ and a goal image such that the desired ${}^w\mathbf{H}_t$ is achieved when the visual servoing converges. The homogenous transformation ${}^e\mathbf{H}_t$ between $\mathcal{F}_e$ and $\mathcal{F}_t$ is determined via a CAD model of the gripper. Eye-to-hand calibration is done using Tsai's method[43], which produces the homogenous transformation ${}^e\mathbf{H}_c$ between $\mathcal{F}_e$ and $\mathcal{F}_c$. Therefore, the transformation ${}^t\mathbf{H}_c$ between $\mathcal{F}_t$ and $\mathcal{F}_c$ is given by

$${}^t\mathbf{H}_c ={}^t \mathbf{H}_e{}^e\mathbf{H}_c = ({}^e\mathbf{H}_t)^{-1}{}^e\mathbf{H}_c. \tag{4}$$

The goal pose of the camera frame with respect to the world frame ${}^w\mathbf{H}_c$ is given by

$${}^w\mathbf{H}_c ={}^w \mathbf{H}_t{}^t\mathbf{H}_c. \tag{5}$$

Substituting Eqs. (3) and (4) into Eq. (5), the goal pose of the camera ${}^w\mathbf{H}_c$ can be solved.

Let the 3D homogenous coordinates of the ith feature point on the projected pattern be ${}^w\mathbf{X}_i \in \mathbb{R}^4$. When the camera is at the goal pose, the feature point can be represented in $\mathcal{F}_c$ as

$${}^c\mathbf{X}_i ={}^c \mathbf{H}_w{}^w\mathbf{X}_i = ({}^w\mathbf{H}_c)^{-1}{}^w\mathbf{X}_i \in \mathbb{R}^4.$$

Finally, ${}^c\mathbf{X}_i$ is projected into the image plane at the camera goal pose as

$${}^c\mathbf{x}_i = \mathbf{K}_c \begin{bmatrix} \frac{1}{{}^cz_i} & 0 & 0 & 0 \\ 0 & \frac{1}{{}^cz_i} & 0 & 0 \\ 0 & 0 & \frac{1}{{}^cz_i} & 0 \end{bmatrix} {}^c\mathbf{X}_i \in \mathbb{R}^3,$$

where $\mathbf{K}_c$ is the calibration matrix of the camera, and ${}^cz_i$ is the depth element in ${}^c\mathbf{X}_i$ when the camera is at the goal pose. The set of feature points ${}^c\mathbf{x}_i$ defines the goal image that guides the tool frame $\mathcal{F}_t$ to the desired grasping position.

4.3. Feature extraction and matching

One advantage of using a projected pattern to guide visual servoing is that the structure of the pattern is fully controlled and the feature extraction is independent from the appearance of the target object. Therefore, the pattern can be designed in a way such that the features can be extracted and matched easily. An arrow shape is used in this section, which is shown in Fig. 5 and illustrated in Fig. 9. The goal arrow image when the camera is at the desired goal pose $\mathcal{F}_c^*$ is automatically generated using the method given in Section 4.2. A sample goal image is illustrated in Fig. 9(a). The four endpoints of the arrow are extracted as feature points.

After a grasping target is specified, a corresponding arrow pattern is generated and projected on the table. An image of the scene is captured by the camera just before the pattern is projected. The arrow is then easily recovered as the difference between the two images.

In the initial and goal images, the line segments $\mathbf{ab}$, $\mathbf{ac}$, $\mathbf{ad}$, $\mathbf{a}'\mathbf{b}'$, $\mathbf{a}'\mathbf{c}'$, $\mathbf{a}'\mathbf{d}'$ are detected using the Hough transform[44]. As seen in Fig. 9(a) and (b), the three line segments intersect at the top vertex, so

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

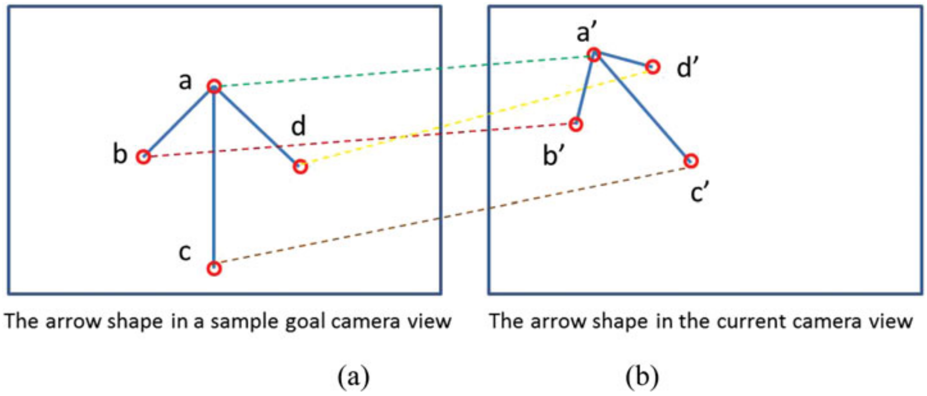

a
b
c
d
a′
b′
c′
d′
The arrow shape in a sample goal camera view
The arrow shape in the current camera view
(a)
(b)

Fig. 9. Matching features on a projected arrow.

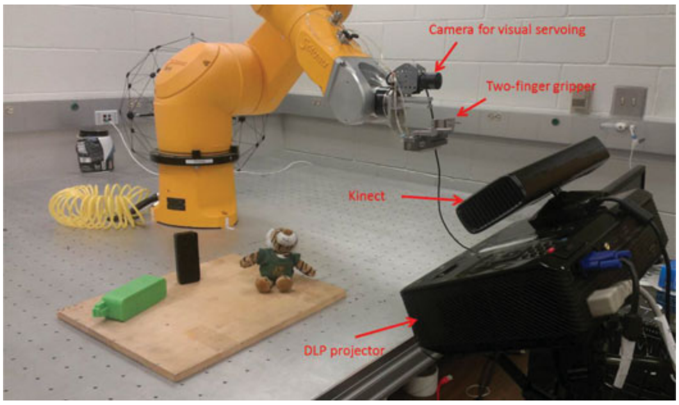


Fig. 10. Experimental environment and a typical system setup.

the feature points **a** and **a**′ can be easily matched. Next, the angle between the each pair of the three line segments is calculated. In the goal image, we have

$$\begin{cases} \angle \text{bac} < \angle \text{bad} \\ \angle \text{cad} < \angle \text{bad}. \end{cases} \tag{6}$$

Note that the inequalities in Eq. (6) are preserved in the image taken at any camera pose $\mathcal{F}_c(0)$. Therefore, the middle line segments **ac** and **a**′**c**′ can be located, and points **c** and **c**′ can be matched. Points **b** and **b**′ and **d** and **d**′ can also be matched based on the extracted line segments and positive or negative angle with respect to **ac** and **a**′**c**′.

During visual servoing, the feature points are tracked in the camera view $\mathcal{F}_c(t)$ as the camera moves toward the goal pose. Using the matched features in the goal image at $\mathcal{F}_c^*$ and the current camera image at $\mathcal{F}_c(t)$, the image error and pose error for SIPVS can be calculated.

5. Experimental Results

The experiments are conducted in a typical indoor environment. Several objects are placed on a flat table surface. A six-degree-of-freedom Staubli TX90 robot arm is used, with a two-finger gripper and a digital camera attached rigidly to the end-effector. The camera used in the experiments is a Matrix Vision mvBlueFOX camera with 1024 × 768 resolution. The resolution of the Kinect RGB camera view is 640 × 480. The objects are all located in the workspace of the robot arm. The experiment environment and a typical setup of the RGB-D camera–projector system is shown in Fig. 10. It can be seen that the Kinect is attached to the top of the projector. The system tilt toward the table surface to observe the objects in the scene and project patterns on the table. Note that the Kinect does not have to be attached to the projector, as long as their relative pose is fixed.

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

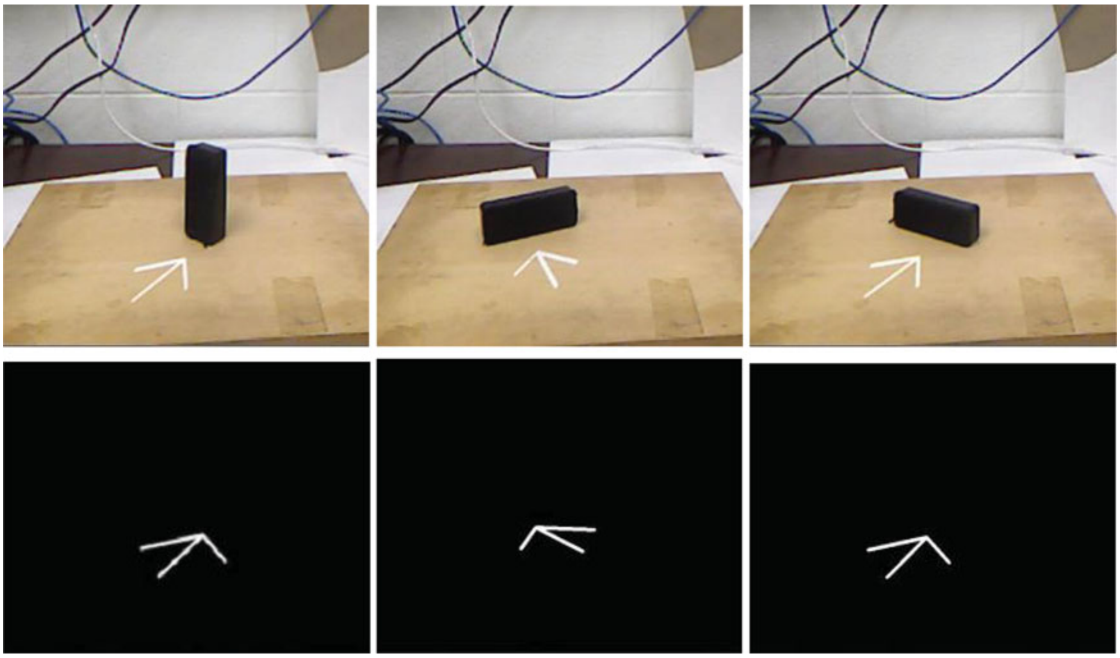


Fig. 11. The top rows shows examples of the direction and orientation of the projected arrow changed according to the different poses of the object. The bottom row shows the corresponding images being displayed by the projector, such that the arrow projects to the tabletop at the desired position and orientation.

5.1. Feedback pattern projection

Figure 11 illustrates the projection of an arrow pattern for an object placed at different locations. The first row shows the scenes in which a black eraser was placed with different poses and the corresponding projected arrows. It can be seen that the arrow was always displayed at the desired position and oriented along the facing direction of the object as described in Section 3.4. The generated arrow patterns in the projector view are illustrated in the second row. The arrow shapes were warped in the projector view so that the projected patterns on the table appear to be the desired shapes and sizes.

Projections of number patterns are shown in Fig. 12. The first row shows the scenes with the objects at different locations and the projected numbers. The black eraser was placed at three different locations with different poses. The number 1 was always projected in front of the eraser and oriented according to the facing direction of the eraser. The stuffed toy was at the static position in the three scenes, therefore, the number 2 was projected at the same position. The second row shows the number patterns generated according to the object poses in the projector view. This is the image that the projector draws. The numbers were warped so that they appeared to be the desired shapes and sizes on the table.

5.2. Quantitative analysis of object detection

We evaluate the performance of the object detection method by testing whether it generates the feedback patterns in the desired way as described in Section 3.4. The object detection is considered to be successful if the generated pattern is aligned with the center of the object and orientated along the side close to the direction of the multi-view system. If no pattern is projected or a badly positioned projected pattern means that the object detection fails. The proposed method was evaluated for both scenes with a single object and scenes with multiple objects.

Experiments were performed for single-object scenes using 20 different objects with different shapes, colors and sizes. Each object was tested at five different poses with respect to the system. The experiment was repeated three times for each pose of the object. Figure 13 shows all of the objects used.

Some successful detection results are given in Fig. 14. The first row shows five scenes and the generated projected patterns as the results of object detection. The second row shows the

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

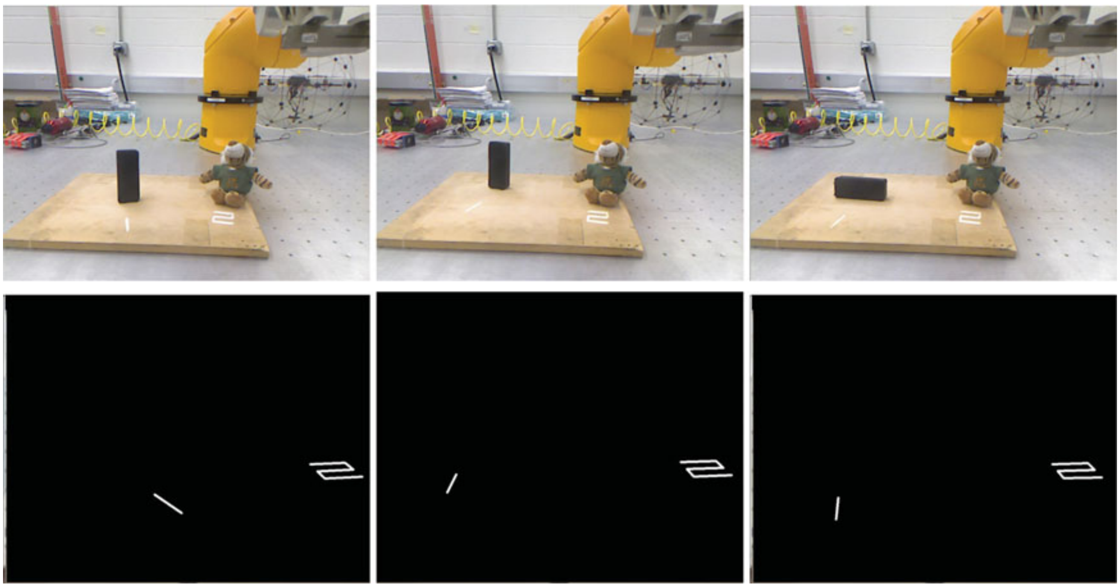


Fig. 12. The top rows shows examples of the directions and orientations of the projected numbers changed according to the different poses of the objects. The bottom row shows the corresponding images being displayed by the projector, such that the numbers project to the tabletop at the desired position and orientation.

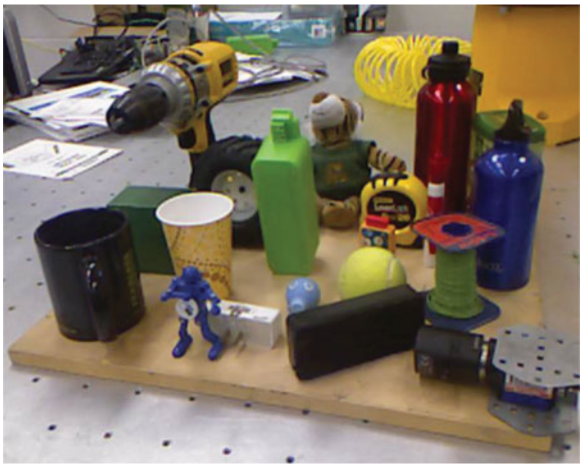


Fig. 13. Objects used in the object detection experiments.

corresponding thresholded saliency maps. It is can be seen that in every example, the feedback pattern is aligned with the center of the object and oriented along the side facing the system. The results show that the saliency-based object detection method is robust against the variation in color, size, shape and lighting conditions.

For all the 300 single-object experiments, the feedback patterns were successfully projected in front of the corresponding objects for 93.6% of the trials. Detection failures were caused mainly by two reasons. One reason is a bad detection result in the saliency map. This is usually caused by one or more poorly performed features in the saliency model such as non-uniform color of the object or inaccurate disparity. The other possible reason is due to the shape of the object. Since the mean of the point cloud and the major axes are used to determine the position and the orientation of the projected pattern, an irregular shape or spatial visibility of the object in the camera view could cause the pattern not projected at the desired location and orientation. Two examples are shown in Fig. 15. The shape of the camera in column (a) was not completely reflected in the saliency map, which caused a poorly positioned projected pattern. In column (b), the pose of the box with three sides visible to the system led to a projected pattern aligned with one of its edges. In both cases in

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

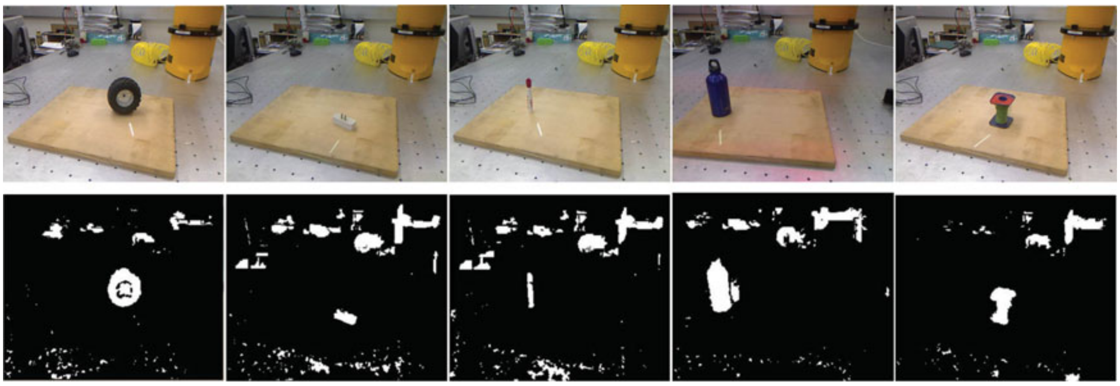


Fig. 14. Successful object detection causes a pattern with the desired appearance projected in front of the object.

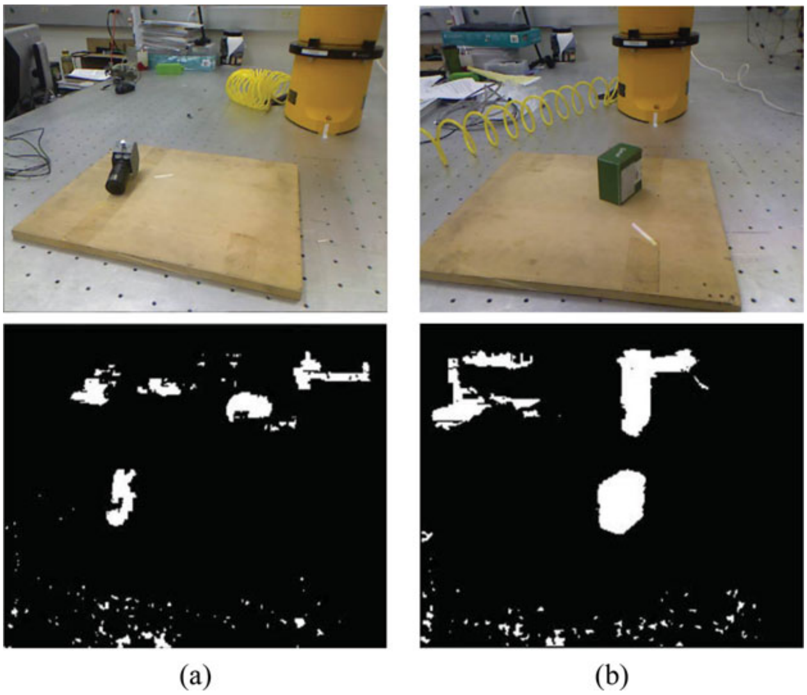


Fig. 15. (a) Detection failure caused by incomplete shape in the saliency map; (b) Detection failure caused by partial visibility in the camera view.

Fig. 15 (a) and (b), the position and direction of the projected patterns cannot be used to determine a successful object-grasping operation.

Next, we evaluate the object detection method for scenes with multiple objects. A total of 20 scenes are used, including 10 scenes with occlusions between objects and 10 scenes without occlusion. There are three tofive objects in each scene. The experiment was repeated three times for each scene. The detection success rates for scenes without occlusion, scenes with occlusion and all scenes are given in Table 5.2. When there is no occlusion between objects, the detection rate is similar with the detection rate for single-object scenes. However, in 3.1%, the projected patterns for different objects overlap, which is not handled by the current pattern generation method. Such overlap can be detected and avoided by adjusting the position, size and orientation of patterns, since the 3D structure of the scene can be recovered by the multi-view system and the coordinates of the patterns on the table is known. The overlap detection will be considered in the future work.

When occlusions exist in the scene, the detection success rate is lower, because occlusion detection is not included in the current object detection method. However, since depth information is included in the saliency-based object detection, in some cases, the system is able to separate the occluded objects when they are at different depth. Some detection examples for scenes with multiple objects

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

Table I. Success rate of object detection for scenes with multiple objects.

	Object tested	# Successful	Success rate
No occlusion	117	106	90.5%
With occlusion	114	94	82.4%
All cases	231	200	86.5%

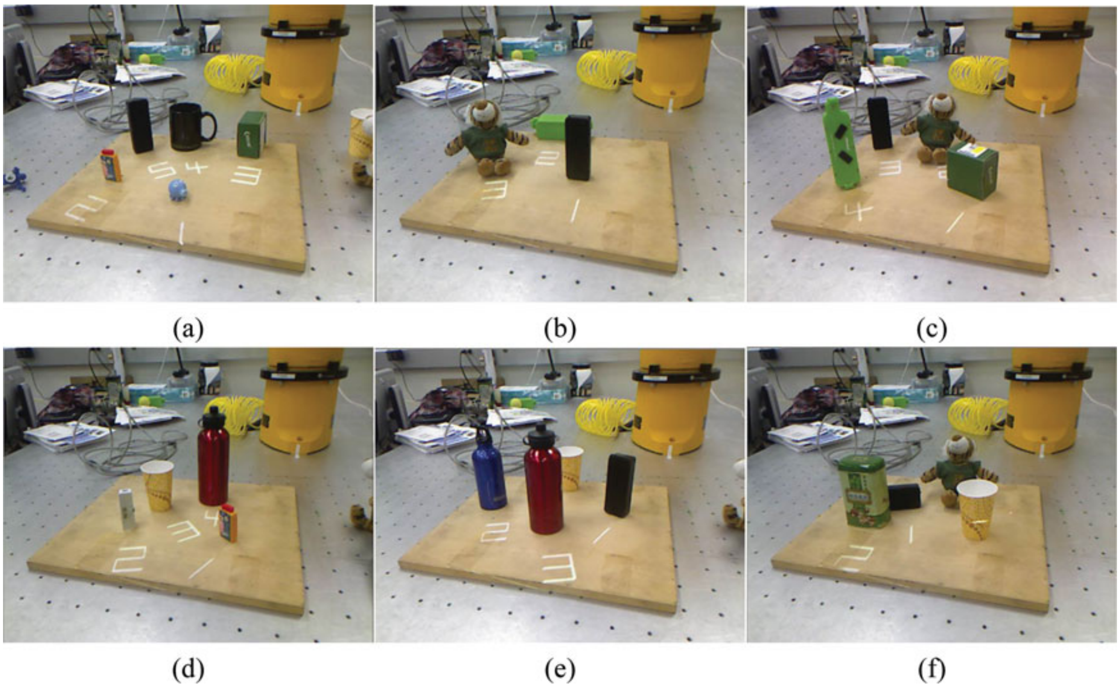


Fig. 16. Examples of object detection for scenes with multiple objects.

are given in Fig. 16. The examples include the scenarios when (a) all objects are detected when no occlusion exists; (b)–(d) occlusion exists and both the occluded objects are detected; (e) occlusion exists and only the object in the front is detected; (f) occlusion exists and neither of the occluded objects is detected.

5.3. *Object grasping by SIPVS control*

We present an object-grasping task guided by the proposed RGB-D camera–projector system. A human user selected the stuffed toy as the grasping target. After the grasping target is selected, an arrow pattern is projected in front of stuffed toy, and the four features points are extracted and matched in the initial camera view as described in Section 4.3. The feature points are tracked using KLT tracker[45] during the visual servoing. To ensure the accuracy of feature tracking, only the four vertices of the arrow shape are projected after they are matched with the feature points in the goal image.

Visual servoing proceeds using the SIPVS control method, which guarantees simultaneous convergence of the pose error and the image error. The object-grasping results are illustrated in Fig. 17. The initial pose of the robot arm can be seen from the Kinect view in Fig. 17(a). The position of the feature points as seen from the initial camera view at the initial pose is given in Fig. 17(c). The green dots indicated the desired position of the feature points in the goal image as calculated in Section 4. From Fig. 17(b) and (d), it can be seen that the gripper is arrived at the desired grasping pose. The four projected feature points appear close to the desired positions in the goal image, which shows that the image error also converged. As discussed in ref. [26], the final pose and image error in SIPVS may not converge to zero due to image noise. However, the experimental results show that the method is accurate enough to guide the gripper to the desired goal position.

Another object grasping experiment is illustrated in Fig. 18. The stuffed tiger was placed at a similar position as in the previous experiment. Figure 18(a) and (b) show the initial pose and grasping

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

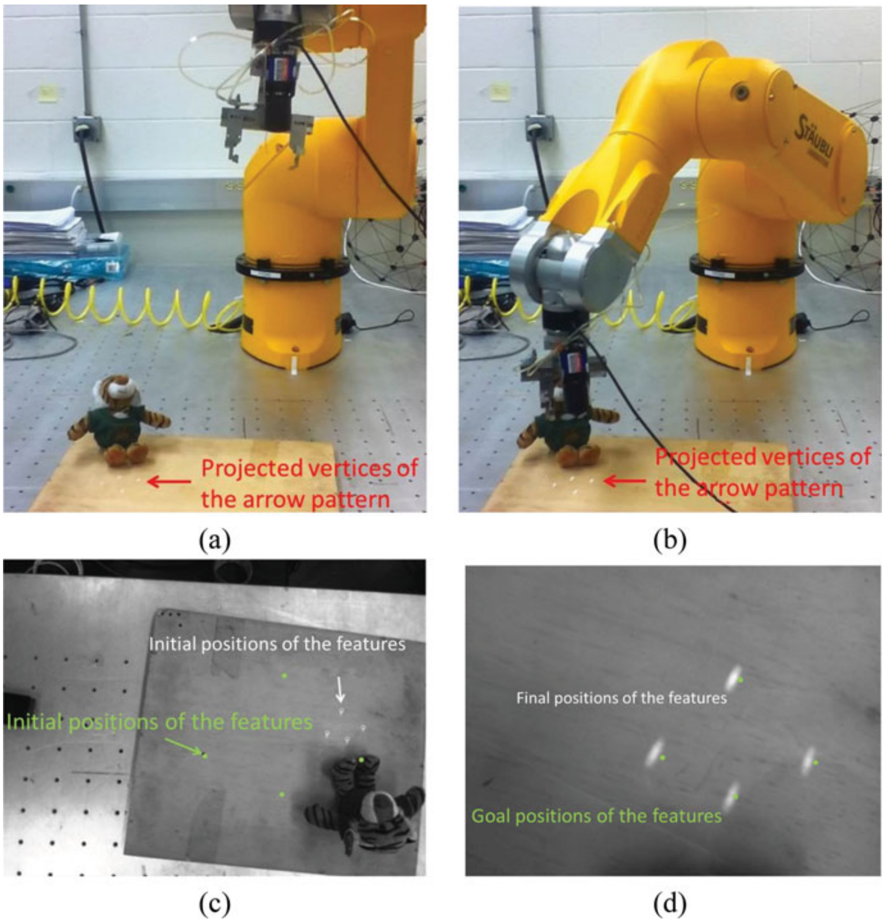

Projected vertices of the arrow pattern
(a)
Projected vertices of the arrow pattern
(b)
Initial positions of the features
Initial positions of the features
(c)
Final positions of the features
Goal positions of the features
(d)

Fig. 17. Object grasping by SIPVS control.

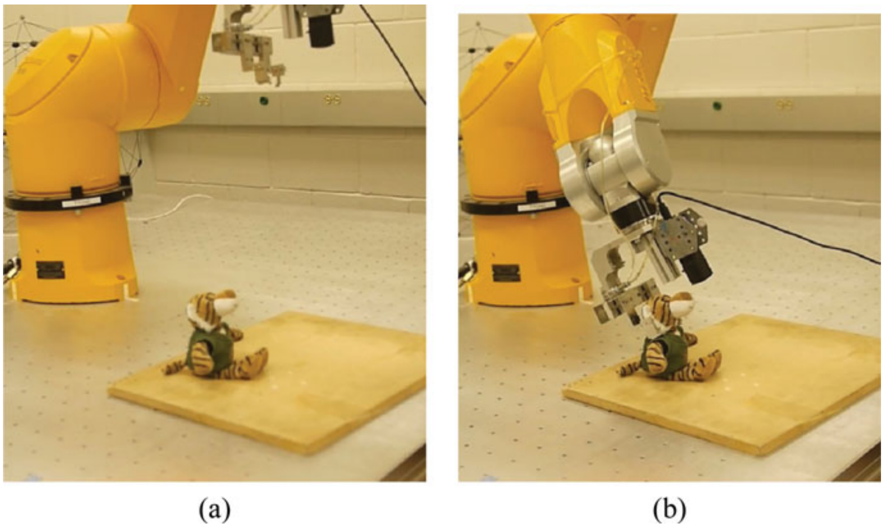


Fig. 18. Grasping direction of the robot is generated automatically according to the pose of the target object.

pose of the robot arm, respectively. It can be seen that, in this experiment, the robot arm grasped the toy from a different direction compared to Fig. 17(b).

The error convergence of the SIPVS during the object-grasping operation is shown in Fig. 19. The convergence of the pose error can be seen in Fig. 19(a). The top figure shows the trajectory of the norm of the translational error, and the bottom figure shows the trajectory of the angular error. It can

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

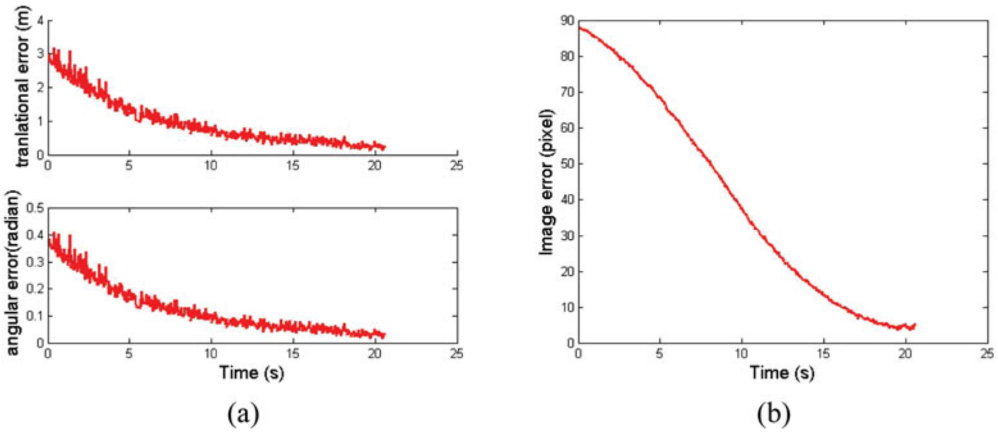


Fig. 19. Object grasping by SIPVS control.

be seen that both the translational error and the angular error approached zero monotonically. The trajectory of the average image error of the four feature points is given in Fig. 19(b). Although the image error decreased monotonically, it converged to a value around 5 pixels instead of 0. This result was possibly due to the errors in the projection of the feature points when generating the goal image, which caused the camera pose that minimized the image error to be slightly different from the goal pose.

6. Conclusion

We explored the use of a RGB-D camera–projector system to provide robot-to-human feedback and to control object-grasping operations. The system detects objects based on a saliency model. By including an RGB-D camera, the system reconstructs the 3D information of the target scene in real time. Feedback patterns are projected on the table in front of the detected objects. The use of numbers and arrow patterns was explored to provide more information about the robot’s understanding of the detected objects. Visual Servoing guides the robot to the grasping pose. An automatic goal pose and goal image generation method was proposed.

The experimental results show that the system can detect salient objects in a typical lab environment and project desired feedback patterns in front of them on the table surface. Automatic object grasping of a selected target was successfully conducted by using SIPVS control with respect to a projected arrow pattern. The error convergence results verified the simultaneous convergence of the pose error and image error.

There are several open avenues for future development of the system. Occlusion detection is needed for the system to work in more general scenarios. As already shown in the robotic vision literature, efficient occlusion detection methods can be developed with the help of depth information obtained from RGB-D cameras. Currently, in the SIPVS-based object grasping, the grasping direction is determined according to a rough estimation of the object shape and pose. Advanced grasping planning is needed to handle object grasping in more general cases. Finally, other projected patterns can be explored to provide more information or enable more complicated interactions between the human and the robots.

Acknowledgments

This work was supported by the Texas Instruments OMAP University Research Program.

References

1. A. Saxena, J. Driemeyer and A. Y. Ng, “Robotic grasping of novel objects using vision,” *Int. J. Robot. Res.*, **27**(2), 2008 (2008).

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

2. T. Jyh-Hwa and L. Su Kuo, "The Development of the Restaurant Service Mobile Robot with a Laser Positioning System," *Proceedings of 27th Chinese Control Conference, CCC 2008*, Kunming, Yunan, China, (Jul. 2008) pp. 662–666.
3. K. Yamazaki, R. Ueda, S. Nozawa, M. Kojima, K. Okada, K. Matsumoto, M. Ishikawa, I. Shimoyama and M. Inaba, "Home-assistant robot for an aging society," *Proc. IEEE* **100**(8), 2429–2441 (2012).
4. M. Johnson-Roberson, J. Bohg, G. Skantze, J. Gustafson, R. Carlson, B. Rasolzadeh and D. Kragic, "Enhanced Visual Scene Understanding Through Human-Robot Dialog," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA (2011) pp. 3342–3348.
5. H. Nguyen, A. Jain, C. Anderson and C. C. Kemp, "A Clickable World: Behavior Selection Through Pointing and Context for Mobile Manipulation," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, Nice, France (2008) pp. 787–793.
6. A. K. Pandey, J.-P. Saut, D. Sidobre and R. Alami, "Towards Planning Human–Robot Interactive Manipulation Tasks: Task Dependent and Human Oriented Autonomous Selection of Grasp and Placement," *Proceedings of the 4th IEEE RAS EMBS International Conference on iomedical Robotics and Biomechatronics (BioRob)*, Rome, Italy (2012) pp. 1371–1376.
7. J. Zhou and J. Hoang, "Real Time Robust Human Detection And Tracking System," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Workshops, 2005. CVPR Workshops*, San Diego, CA, USA (2005) pp. 149–149.
8. N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Trans. Instrum.* **60**(11), 3592–3607 (2011).
9. DG Lowe, "Object Recognition from Local Scale-Invariant Features," *Proceedings of the IEEE International Conference Computer Vision*, volume 2, Kerkyra, Greece (1999) pp. 1150–1157.
10. J. Ma and J. W. Burdick, "A Probabilistic Framework for Stereo-Vision Based 3d Object Search With 6d Pose Estimation," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, USA (2010) pp. 2036–2042.
11. S. Omachi and M. Omachi, "Fast template matching with polynomials," *IEEE Trans. Image Process.* **16**(8), 2139–2149 (2007).
12. B. Rasolzadeh, M. Björkman, K. Huebner and D. Kragic, "An active vision system for detecting, fixating and manipulating objects in the real world," *Int. J. Rob. Res.* **29**(2–3), 133–154 (2010).
13. L. Itti, C. Koch and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(11), 1254–1259 (1998).
14. M. Johnson-Roberson, J. Bohg, M. Bjo andrkman and D. Kragic, "Attention-Based Active 3d Point Cloud Segmentation," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan (2010) pp. 1165–1170.
15. J. Salvi, J. Pages and J. Batlle, "Pattern codification strategies in structured light systems," *Pattern Recognit.* **37**, 827–849 (2004).
16. L. Bo, X. Ren and D. Fox, "Depth Kernel Descriptors For Object Recognition," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA (Sep. 2011) pp. 821–826.
17. A. Gams, J. van den Kieboom, F. Dzeladini, A. Ude and A. Jan Ijspeert, "Real-time full body motion imitation on the coman humanoid robot," *Robotica* **33**, 1049–1061 (2015).
18. H. Takizawa, S. Yamaguchi, M. Aoyagi, N. Ezaki and S. Mizuno, "Kinect Cane: An Assistive System for the Visually Impaired Based on Three-Dimensional Object Recognition," *Proceedings of the IEEE/SICE International Symposium on System Integration (SII)*, Fukuoka, Japan (Dec. 2012) pp. 740–745.
19. Y. Hiroi and A. Ito, "Asahi: Ok for Failure a Robot for Supporting Daily Life, Equipped With a Robot Avatar," *Proceedings of the 8th ACM/IEEE International Conference on Human–Robot Interaction (HRI)*, Tokyo, Japan (2013) pp. 141–142.
20. J. N. Mak, Y. Arbel, J. W. Minett, L. M. McCane, B. Yuksel, D. Ryan, D. Thompson, L. Bianchi and D. Erdogmus, "Optimizing the p300-based brain-computer interface: Current status, limitations and future directions," *J. Neural Eng.* **8**(2), 025003 (2011).
21. P. Beardsley, J. van Baar, R. Raskar and C. Forlines, "Interaction using a handheld projector," *IEEE Comput. Graph. Appl.* **25**(1), 39–43 (2005).
22. R. Kjeldsen, C. Pinhanez, G. Pingali, J. Hartman, T. Levas and M. Podlaseck, "Interacting With Steerable Projected Displays," *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition*, Washington, D.C., USA (2002) pp. 402–407.
23. S.-W. Choi, W.-J. Kim and C. Ho Lee, "Interactive Display Robot: Projector Robot With Natural User Interface," *Proceedings of the 8th ACM/IEEE International Conference on Human–Robot Interaction (HRI)*, Tokyo, Japan (2013) pp. 109–110.
24. J. Shen and N. Gans, "A Trifocal Tensor Based Camera-Projector System for Robot–Human Interaction," *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, Guangzhou, China (2012).
25. J. Shen, J. Jin and N. Gans, "A Multi-View Camera-Projector System for Object Detection and Robot–Human Feedback," *Proceedings of the IEEE International Conference on Robotics and Automation*, Karsruhe, Germany (2013).
26. N. R. Gans, G. Hu, J. Shen, Y. Zhang and W. E. Dixon, "Adaptive visual servo control to simultaneously stabilize image and pose error," *Mechatronics* **22**(4), 410–422 (2012).

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press

27. F. Chaumette and S. Hutchinson, “Visual servo control part I: Basic approaches,” *IEEE Robot. Autom. Mag.* **13**(4), 82–90 (2006).
28. E. Malis and F. Chaumette, “2 1/2D visual servoing with respect to unknown objects through a new estimation scheme camera displacement,” *Int. J. Comput. Vis.* **37**(1), 79–97 (2000).
29. J. Pages, C. Collewet, F. Chaumette and J. Salvi, “A Camera–Projector System for Robot Positioning by Visual Servoing,” *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop, CVPRW ’06*, New York, NY, USA (2006) pp. 2–2.
30. A. Rotenstein, A. Andreopoulos, E. Fazl, D. Jacob, M. Robinson, K. Shubina, Y. Zhu and J. Tsotsos, “Towards the Dream of Intelligent, Visually Guided Wheelchairs,” *Proceedings of the 2^{nd} International Conference on Technology and Aging*, Toronto, Canada (2007).
31. S. Vijayakumar, J. Conradt, T. Shibata and S. Schaal, “Overt Visual Attention for a Humanoid Robot,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, USA (2001).
32. S. Frintrop and P. Jensfelt, “Attentional landmarks and active gaze control for visual slam,” *IEEE Trans. Robot.* **24**(5), 1054–1065 (2008).
33. G. Heidemann, R. Rae, H. Bekel, I. Bax and H. Ritter, “Integrating Context Free and Context-Dependent Attentional Mechanisms for Gestural Object Reference,” *Proceedings of the 3rd International Conference on Computer Vision Systems*, Graz, Austria (2003).
34. Y. Niu, Y. Geng, X. Li and F. Liu, “Leveraging stereopsis for saliency analysis,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA (2012) pp. 454–461.
35. B. Han and B. Zhou, “High Speed Visual Saliency Computation on gpu,” *Proceedings of the IEEE International Conference on Image Processing*, volume 1, San Antonio, TX, USA (2007) pp. I–361–I–364.
36. Y. Ma, S. Soatto, J. Kosecka and S. S. Sastry, *An Invitation to 3-D Vision* (New York, USA, Springer, Nov. 2003).
37. L. Deng, F. Janabi-Sharif and W. J. Wilson, “Hybrid motion control and planning strategies for visual servoing,” *IEEE Trans. Indust. Eng.* **52**(4), 1024–1040 (2005).
38. N. R. Gans and S. A. Hutchinson, “Stable visual servoing through hybrid switched-system control,” *IEEE Trans. Robot.* **23**(3), 530–540 (2007).
39. G. Hu, W. Mackunis, N. Gans, W. E. Dixon, J. Chen, A. Behal and D. M. Dawson, “Homography-based visual servo control with imperfect camera calibration,” *IEEE Trans. on Autom. Control* **54**(6), 1318–1324 (2009).
40. N. Burrus, “Rgbdemo,” Available at: http://labs.manctl.com/rgbdemo/index.php (2012).
41. J.-Y. Bouguet, “Camera calibration toolbox for matlab,” Available at: http://www.vision.caltech.edu/bouguetj/calibdoc/ (2010).
42. G. Falcao, N. Hurtos and J. Massich, Plane-based calibration of a projector-camera system. *VIBOT Master* **9**(1), 1–12 (2008).
43. R. Tsai, *Synopsis Recent Progress on Camera Calibration for 3D Machine Vision* (MIT Press, Cambridge, MA, USA, 1989).
44. J. Illingworth and J. Kittler, “A survey of the hough transform,” *Graphical Models/graphical Models and Image Processing /computer Vision, Graphics, and Image Processing* **44**, 87–116 (1988).
45. C. Tomasi and T. Kanade, “Detection and tracking of point features,” Technical report, Carnegie Mellon University (1991).

https://doi.org/10.1017/S0263574717000339 Published online by Cambridge University Press