Control of Prehension for the Transradial Prosthesis: Natural-like Image Recognition System
Djordje Klisić, Miloš Kostić, Strahinja Došen, and Dejan B. Popović

Abstract— We describe the hardware and software for the control of prehension for a dexterous transradial prosthesis. The prehension process comprises hand orientation (three degrees of freedom) and the opening of the hand in a manner that is appropriate for the shape and size of the object. The hardware consists of a standard web camera, accelerometer, ultrasonic distance sensor, laser pointer and an LED illumination system. Software operating in real time estimates the shape and size of the object as well as the relative orientation of the hand with respect to the object. Based on this data, the controller generates signals that are sent to the three-dimensional (3D) wrist rotator, and drives which control fingers and thumb of the transradial prosthesis, thereby preparing the hand for palmar, lateral, or precision (2-digit or 3-digit) grasps. The choice of the grasp follows heuristics captured from healthy humans when grasping and expressed in the form of IF-THEN rules.

Index Terms — Control, Prehension, Transradial Prosthesis, Vision

I. INTRODUCTION

REACHING grasping, and releasing (RGR) functions are types of so-called goal-directed movements. A goal-directed movement can be defined as a planned tuning of the positions of body segments, ultimately leading to the accomplishment of a task [1, 2 and 3]. This type of movement is governed by a highly complex, perception-driven dynamical process [4] that comprises two essentials: 1) planning, and 2) execution. Grasping is planned based on visual input and heuristics acquired through experience and learning.

Visual navigation was first introduced in the field of robotics [5]. Robotics technology has expanded over time, and robots have become part of everyday life. Proper visual navigation (visual servoing), however, is an area of robotics that yet has to be perfected. Several visual servoing solutions have been presented [6, 7 and 8], but all suffer from limitations regarding their speed or the complexity of the scenes, in some cases being designed to deal only with a predefined scene. All of these solutions incorporated at least one camera, and dept perception was handled in different ways (laser, stereovision, and ultrasound). The differences among the methods described in the literature relate to the particular position chosen for the camera: an eye-in-hand configuration [6], a stationary camera [8], or a combination of eye-in-hand, and a redundant system of cameras [7].

The task of this research involves the prehension of a dexterous artificial hand. The important differences between visual control in robotics and the control of an artificial hand are: 1) the dexterous hand is adaptive, which decreases the problems coming from the imperfect alignment of the hand and the object, and 2) the amputee can compensate for the imperfect positioning of the hand by adjusting her/his trunk, shoulder, upper arm, and/or the existing part of the forearm. More precisely, the artificial vision system needs to provide the approximate size and shape of the object as well as the relative orientation of the principal axes of the hand and the object. Based on this information and heuristics captured from healthy humans when grasping, the details of prehension can be planned: the degree to which the hand opens, hand rotation to match the orientations of the object and hand, and the type of the grasp (lateral, palmar, or precision).

We hypothesized that three sensors and two additional accessories would be sufficient hardware for this task (Fig. 1). The sensors used are a camera, a distance sensor and a 3D accelerometer. Two additional accessories employed are a laser diode for pointing at the targeted object and a high power LED, which minimizes shadow effects resulting from normal lighting. We selected the eye-in-hand configuration; the system was mounted on the dorsal side of the forearm slightly elevated, so that the camera had an unobstructed view of the scene in front. Such configuration was selected because the system "sees" the target object frontally, that is, from the perspective of the approaching hand and this is convenient for the estimation of object properties. The order of the operation of the components is the following: 1) the laser diode is used as a pointer to provide precise information to the camera, identifying the target in a scene that may include several objects; 2) the distance sensor assesses the distance between the camera and the target object. This information is important for calculating the size of the object. Furthermore, the distance sensor triggers the operation of the camera and the onset of the analysis of the accelerometer data (hand orientation estimation). The camera begins to record the scene only when the distance to the target is less than 30 cm; and 3) the acquired images are then passed to the image processing software.

The software was developed to fulfill the following functions: 1) reduction of the image to the contour of the object, 2) estimation of the shape of the object (target...
object silhouette), 3) determination of the appropriate type of grasp, 4) rotation of the target object silhouette and estimation of the size of the object, 5) determination of the hand aperture, 6) estimation of the hand orientation with respect to the object, and 7) determination of the necessary pronation/supination and wrist rotations.

II. METHODS AND MATERIAL

Vision sensor. We selected a compact, low-resolution web camera with 300K pixels and resolution up to 640x480. This camera supports three resolution settings (640x480, 320x240 and 160x120) and two color palette types (UYVY and YUY2). The camera was connected directly to the computer via USB port. The “winvideo” adapter within the MATLAB software package proved sufficient to control the operation of the camera selected.

Distance sensor. We used an ultrasound distance sensor that is suitable for distances between 3 cm and 4 m and has a scattering angle of 30\(^\circ\) (SRF05, Devantech, UK). We selected to use the sensor’s digital output, where the pulse duration (width) corresponds to the distance.

Pointing laser diode. We decided to use a red laser diode as the pointer. This pointer determines which object is the target when the scene comprises several objects.

Accelerometer. We used a triaxial accelerometer (MMA7260QT, Freescale, USA). We selected the operation mode in which the accelerometer’s sensitivity is 800 mV/g, where g is the gravitational acceleration. The accelerometer outputs three signals corresponding to orthogonal axes. The low-passed values of these signals are used to determine the orientation of the hand with respect to gravity.

High Power LED. To provide the necessary homogeneous illumination of the scene, we employed a high power LED. The one mounted on our system had 3 W output power and a large scattering angle (120\(^\circ\)). This LED successfully eliminates shadows originating from daylight or room lighting when the distance between the camera and the object is under 30 cm.

A. Data and Image Acquisition.

Both analog and digital signals were acquired by a 16-bit USB DAQ card (NI DAQ 6112, National Instruments, USA). The complete operation of the system was tested with the MATLAB 7.3 and LabView 8.2 software packages on a standard Windows XP-based laptop computer operating at 2 GHz. The flowchart of the program that we developed is presented in Fig. 2.

We configured the camera to acquire images in the YUY2 color palette and with a resolution of 160x120 pixels. The MATLAB Image Acquisition toolbox was used to make snapshots of the scene viewed by the camera. It was necessary to take several frames because the first one was often blurry. The camera was therefore set to capture three frames in a row, and only the third one was sent to the image processing software.

B. Image processing and segmentation.

We selected the YUY2_160x120 mode for operation of the camera. MATLAB Image Acquisition toolbox was used to make snapshots of the scene viewed by the camera. It was necessary to make several frames because the first one was often blurry. Camera was set to take three frames in a row, and since only one frame was used for the analysis, the third one is sent to image processing software.

We used the YUV (YCrCb) color space. The first step in processing was the segmentation of the image. The result of the segmentation is a binary image in which the targeted object is represented with white pixels (1), while everything else is black (0). Such an image is called an image primitive, and most of the processing after segmentation was done with the targeted object’s primitive. The segmentation algorithm that we applied has four stages: 1) edge detection, 2) black and white representation, 3) locating the target object, and 4) binarization of the targeted area.

1) The edge detection. Edges were detected by the Canny method [9]. The Canny method finds edges by looking for local maxima of the gradient of the image. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds to detect strong and weak edges, and it includes the weak edges in the output only if they are connected to the strong edges. This method is therefore less likely than others to be fooled by noise and more likely to detect the true weak edges.
To ensure greater reliability, edges were found in all three components of the YUV image and then superposed (logical OR). Morphological operations "open" and "close" are applied to the resulting image to ensure that the edges are continuous. Now the resulting image contains the contours of all objects and maybe some contours belonging to the background as well.

2) Black and white object representation. In this step, all closed contours and free lines obtained in the previous step were temporary considered as belonging to the objects. Pixels inside the contours were given a value of 1 ("filling" the contours), while everything else was assigned a value of 0. Then, the image was morphologically opened to eliminate the free lines. To separate the objects from the image border, a thin frame of pixels (10-pixel wide) along the image border was set to 0. This could be done since it was known that the target object was always near the center of the image, and therefore unconnected with the image border. The outcome is the "framed" image of the objects.

3) Locating the target object (TO). As mentioned before, the laser beam marks the TO. To find the laser tag in the image, both YUV and RGB spaces were used. The laser tag was considered to be a set of pixels in the V(Cr) and R color planes (logical OR) with the values greater than 95% of the maximum. The TO was identified as the object that contained these pixels, and everything outside this object was set to 0. After this step, it was necessary to perform an acceptance test for the acquired shape. Acceptance test is performed by comparing the solidity of the detected object with heuristically determined threshold. Since bad segmentation occurs when both TO and nearby shadows and/or background "stains" are recognized as the object of interest, the solidity in that case is significantly smaller than in any real object. If the object passed the test, then the segmentation process was finished. Otherwise, if the potential TO failed the test, it meant that the segmented area contains both the TO and the unwanted elements of the background. In this case, the step four was started.

4) Binarization of the segmented area. The algorithm only entered the fourth step when the scene presented adverse characteristics (shadows, bad illumination, background texture, etc.). In this rather unfavorable situation, the segmented area from the step 3 could contain both the TO and the background segments. To find the TO under these conditions, optimal threshold binarization was applied. Binarization is performed in YUV space. If the object was significantly lighter or darker than the background, then the Y component gave the best result; otherwise, the chrominance component yielded a better result (either Cr or Cb, depending of the object color). After applying an optimal threshold on one component matrix, the object containing the center pixels is named TO. Since there was no previous knowledge about the color of the TO, binarization is performed iteratively, starting from Y component, until the resulting TO passes the acceptance test.

C. Estimation of the orientation of the hand vs. the object.

To extract hand orientation, we used the low-passed outputs of the triaxial accelerometer. The first-order filter, with the cut-off frequency set at 0.1 Hz, provided a good estimate of the orientation of the slowly moving hand with respect to gravity.

We grouped the objects in two sets: horizontal objects (pencil, knife, fork, spoon, etc.) and vertical objects (e.g., cup, glass, can, bottle, etc.). This classification was used in conjunction with information about the size and shape of the object to determine the necessary rotation of the hand. Namely, after the type of grasp was established, the required angles of rotation were estimated based on the hand’s orientation with respect to the horizontal or vertical object.

Image size. The apparent size of the object is determined by the 2D projection of the object, that is, a 2D image that the camera sees. The camera system was calibrated at a distance of 25 cm, leading to the following conversion factor: 1 pixel is a square 25 cm away with the sides of 0.117 cm. In normal operation, the system is set to acquire images at distances between 24 cm and 26 cm. First, the estimated orientation of the hand was used to rotate the image of the TO into a default position (vertical, centered in the image). This was done in order to be able to locate the perimeter pixels and estimate the lengths of the object axes in pixels. From this information and calibration data, it was trivial to calculate the object’s size, that is, the lengths of the long and short object axis in centimeters.

Cognition. The cognitive part of the algorithm is implemented as a set of IF-THEN rules. The input for this step is the estimated size of the target object; i.e., the lengths of the short and long object axes. The outputs are the type of grasp and aperture size that are appropriate for gripping the object. Three different grasp types are considered (see Table 1): palmar, lateral, and pinch. These types were selected since they are used for almost 95% of functions during the activities of daily living [3]. Each grasp is available in two predefined aperture sizes (i.e., small, and large).

<table>
<thead>
<tr>
<th>GRASP TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
</table>

TABLE 1: IMPLEMENTED GRASP TYPES ARE DIVIDED TO PINCH, LATERAL, AND PALMAR. PALMAR GRASP TYPE IS SUITABLE FOR LARGE AND THIN OBJECTS.
The basic principle in constructing the rules was to match the size of the object with an appropriate grasp type and size. Namely, large objects are grasped by using lateral grasp if they are thin or palmar grasp otherwise. The pinch grasp is reserved for small objects. The aperture size depends on how thin (wide) the object is. As an example, the rules for selecting a palmar grasp are given in Figure 3.

The cognition is a two step process. In the first step, the IF-THEN rules categorize the target object into size classes (e.g., large, small, thin, wide). This is done by comparing the estimated lengths of the object axes against fixed thresholds, which are specified relative to the size of the artificial hand and the size of its maximal apertures in different grasp types. In the second step, the size attributes assigned to the object are used to select an appropriate grasp type and size.

**STEP 1 – determining object features**

\[ \text{IF } \text{long axis} > T_{\text{SMALL}} \text{ THEN object IS large} \]
\[ \text{IF } \text{short axis} > T_{\text{THIN AND short axis} < T_{\text{VERYWIDE}}} \text{ THEN object IS wide} \]
\[ \text{IF } \text{short axis} > T_{\text{VERYWIDE}} \text{ THEN object IS very wide} \]

**STEP 2 – deciding the grasp type and size**

\[ \text{IF object IS large AND object IS wide THEN palmar small} \]
\[ \text{IF object IS large AND object IS very wide THEN palmar large} \]

The cognitive system is also used to generate control signals that drive the wrist rotator to orient the hand such that the object axes match the axes of the hand, thereby allowing the hand to anticipate the optimal position for grasping. This is implemented by using the common knowledge from the motor control and biomechanics of human grasping of how to form the opposition spaces in different grasp types [3] For example, if the target object was classified as vertical, and the grasp type selected is palmar, then the hand should be oriented so that the palm is parallel to the vertical axis of the object. The angle through which the hand should be rotated is trivial to calculate from the current hand orientation (as estimated by using accelerometer data).

### III. RESULTS

We illustrate the operation of the system with a set of representative results.

The first example is the extraction and recognition of a ceramic glass that is positioned on the desk and surrounded with several other objects (Fig. 4).

This picture demonstrates the steps in the image processing part of the algorithm. The targeted object (i.e., a cup with in the middle with a laser tag on it) was successfully extracted from a rather complex scene including several other objects and a non-trivial background. The initially segmented area (Fig. 4) did not pass the acceptance test, but the optimal binarization successfully separated the object from the unwanted elements.

The second operation that is integrated in the software is picture rotation (Fig. 5). The left panel in Fig. 5 shows the original image, while the right panel shows the rotated image.

![Figure 5: The result of the image rotation operation that is integrated into the software](image)

The rotation angle was generated from the accelerometer data.

**Fig. 6 presents the results of the combined actions of rotation and primitive extraction. The target object primitive is brought into a default position. Now, the perimeter pixels can be determined and the lengths of object axes measured.**

![Figure 6: The original image seen by the camera (left), the extracted primitive (middle), and rotated primitive (right)](image)

We also show an example where the complexity of the scene caused the software and hardware to fail to extract the object correctly (Fig. 7). The target object was red and the algorithm therefore failed to identify the laser tag.

![Figure 7: In this instance, overlapping neighboring objects and incorrect recognition of the laser tag (red dot) created with the laser pointer led to “confusion” in the primitive extraction process. The software was unable](image)
to extract the primitive and instead determined the background to be the object.

Operational testing demonstrated the need for the software to be embedded in the microcomputer for practical uses, especially with complex images where the processing steps require iteration (step 4 of the algorithm).

This system was designed to test ideas and concepts. The original task was to develop a system that integrates vision with cognition borrowed from the heuristics captured in healthy humans when grasping. The outputs from the vision system are information about the position, size, and shape of the object that are likely sufficient for the prehension. The system in its present form is not meant to be used as a commercial accessory within a transradial prosthesis, but it could be used as the basis for developing technologically adequate hardware that can be integrated into an artificial hand.

Another possible use of the system is for the control of rehabilitation systems for the training of individuals with hemiplegia that integrate assistive robots and functional electrical stimulation. In this case, however, the system would need to be positioned on the head, since otherwise it would be very difficult for the user to control the camera.

IV. DISCUSSION

The results presented above show that the system’s image processing is reliable and highly robust. We tested the software with many objects in various lighting conditions, and the overall success rate, compared to human expertise, is 90%. When the HP LED was turned off the change in lighting caused the success rate to drop to about 70%. The scene presented in Fig. 4 was taken during the day with no artificial lighting in the room, and the scenes in Figs. 2 and 3 were taken in the evening with the normal room lights. In both cases, the system extracted TO primitive correctly. The success rate was good in complex scenes, except when either the red spot created by the LED pointer was not recognized (as shown in Fig. 7) or the object images were overlapping. These exceptions call for improvements, such as using more LED pointers in different colors and introducing the requirement that the object needs to be in the center of the image. The use of the LED pointer was simple and effective for large and medium objects, but for small objects, it was much more difficult for the user to aim at the object and this led to errors.

A single accelerometer and low-pass filtering of output signals form it were shown to be sufficiently robust for the recognition of the orientation of the objects that are of interest for practical applications. One possible improvement could be to combine of accelerometer and a gyroscope as the sensors for estimation of the position of the hand. The other envisioned alternative is to use two cameras and implement stereo vision; however, this requires more computing power.

REFERENCES