Recognition of Arbitrary 2D Shapes for Pick and Place Solutions using Robot Manipulators

Natasha Govender
Council for Scientific and Industrial Research
Email: ngovender@csir.co.za

Jonathan Claassens
Council for Scientific and Industrial Research
Email: jclaassens@csir.co.za

Abstract—Robotic platforms are consistently playing greater roles in manufacturing processes. In many production lines robots are required to recognize different objects of varying shapes and sizes. They also need to reliably manipulate these objects to achieve some goal. By improving the robustness of these components in uncertain conditions, a number of new applications could be enabled.

We present a complete pick and place system using a robot manipulator which assembles a child’s puzzle. The problem was chosen based on the arbitrary nature of the shapes involved and because of the intricate manipulation necessary for the problem. Our proposed solutions to shape recognition, localization and placement are described in the paper. We chose a Fourier based descriptor for shape recognition and servo the manipulator in vision space for fine manipulation. System performance is presented with promising results.

I. INTRODUCTION

The growth of robotics in industry is at a steady increase. They are finding applications in a wide range of fields from medical science to manufacturing to space exploration. In manufacturing, robots are used to weld, bend, sort or even perform quality inspection. Often, these tasks require grasping a specific object from some bin and placing it at some specified location. Sometimes this is the specific goal of the robot and it is called a ‘pick and place’ system. The problem involves recognising the correct object. Secondly placement relies on precise visual (or some other sensor) information to correctly position the part. Robustness, flexibility and efficiency are all critical to the task.

This paper presents a complete system for picking and placing parts. We chose the toy problem of a child’s shape puzzle because the shapes were relatively arbitrary and a fair amount of dexterity is required to place pieces in their respective positions. Any resulting solution could easily find application in typical industrial problems. The aim is to further broaden the applicability of pick and places systems by improving the robustness of object recognition and manipulation.

The robot we chose to demonstrate with is a Barrett Whole Arm Manipulator (WAM) with 7 degrees of freedom (DoF) and a Salisbury-type Barrett Hand end-effector. The robot is backdrivable and capable of compliant interaction. It is also human safe, however, the extreme capabilities of this robot are not necessary for the problem, any 4 DoF robot will suffice.

An omni-directional camera with a fish-eye lens was mounted on the arm to capture images on the fly. A fish-eye lens has a wider field of view (190°) than an ordinary perspective lens. The scope of this paper is to consider only the 2D shape of the parts or pieces involved. We further assume that the background is easily differentiated from the parts. Textual and feature-based approaches are often not appropriate for these types of applications because parts may contain little or no distinctive features other than boundary shape. Also, the machining process may not have consistent lighting conditions which can adversely affect these approaches. In this paper we use polar coordinates as our shape signature for the Fourier descriptors. We also introduce a novel approach to determine the orientation of the object by using Principal Component Analysis (PCA), which assists when grabbing the object. The technique used for manipulation problem involves servoing using visual information to grasp and position a part.

The structure of the paper is as follows: The next section describes related work. Section III elaborates on the problem and Section IV discusses the process used to capture, calibrate and reproject images onto a 2D plane. A complete description of the processes implemented for shape modeling and for manipulating an object is presented in Sections V and VI. Section VII presents the experimental results. The conclusion and possible future work is discussed in Sections VIII and IX.

II. RELATED WORK

Various shape representation methods, or shape descriptors, exist in the literature. These methods can be classified into two categories: region based versus contour based. In region based techniques, all the pixels within a shape are taken into accounted to obtain the shape representation [12],[11]. Contour based shape representation exploits shape boundary information.

Fourier Descriptors are contour based and capture global shape features in the first few low frequency terms, while higher frequency terms capture finer features of the shape. Wavelet descriptors can also be used to model shape and have an advantage over Fourier descriptors in that they maintain the ability to localize a specific artifact in the frequency and spatial domains [13]. However, wavelet descriptors are impractical for higher dimensional feature matching [15]. The Fourier Descriptor method can also be easily normalized.

Fourier Descriptors are a widely used, all purpose shape description and recognition technique. It has been used in a variety of fields over the years, including commerce, medical,
space exploration, and technical sectors. In the field of computer vision, Fourier Descriptors have been used for human silhouette recognition [4] for surveillance systems, content based image retrieval [17],[6], shape analysis [16],[7], character recognition [8],[9] and shape classification [3]. In these methods, different shape signatures have been exploited to obtain the Fourier Descriptors. These include central distance, complex coordinates, curvature function, and cumulative angles [1]. Most systems use complex coordinates to model the shape boundary [14],[4] but we use polar coordinates because in our experiments this method produces more accurate results.

Pick and place robots are used for a wide variety of material transfer tasks. A good example is a robot picking items off a conveyor belt and placing them in packaging boxes. Some systems are setup with vision guidance to aid the picking process. The typical pick and place application usually involves a high amount of repetitive motion while manipulating of objects of the same shape and size [5]. Our system is robust enough to allow a pick and place application to recognize and localize objects of different shapes and sizes, manipulate these objects and even place them in different end positions as required.

III. THE PROBLEM

A board containing cut out cartoons of different animals was used in the experiments. The shapes were removed from the board and then placed on the table. Figure 1 and Figure 2 display the board and the shapes used in the experiments. The system was required to identify each shape, localize it, grab it and then place it in the correct position on the board.

IV. DATA COLLECTION

An omni-directional camera with a circular fish-eye lens was mounted on the Barrett WAM. A fish-eye lens is a wide-angle lens that captures a broad, panoramic and hemispherical image. The reason for opting for a fish-eye lens is because of its ability to capture an image with a larger field-of-view for localizing a chessboard and shape simultaneously. We use the chessboard as a fixed reference and to identify the plane parameters of the working surface. The lens is also sufficiently small to be mounted on the Barrett hand without obstructing any movement.

A. Calibration

Images taken with fish-eye lenses have severe distortions and the camera needs to be calibrated. The omni-directional camera calibration method developed by Davide Sacramuzza [10] was used. This method requires the camera to observe a planar pattern shown at a few different orientations. No prior knowledge of the motion is required nor a specific model of the omnidirectional sensor. The only assumption is that the image projection function (of the camera optics) can be described by a Taylor series expansion whose coefficients are estimated by solving a two-step least squares minimization problem. For a more detailed explanation the reader is referred to [10].

B. Reprojection

Images captured with a fish-eye lens are spherical in nature. These images are reprojected onto a 2D plane to enable
segmentation and extraction of all the relevant object shape information. There are number of ways to project points that lie on a spherical surface onto a 2D plane. These include stereographic, cylindrical, mercator and equirectangular projection.

Our approach creates a model for mapping the given 3D co-ordinates onto a 2D plane taking into account our camera model. A virtual spherical image is created with the same resolution as the captured images i.e. 1280x960. We do not want to reproject the entire spherical image; just the parts of interest which we assume to occur in the centre of the image. Thus, the virtual image contains this portion. For the z axis, one needs to bear in mind that the projection can occur behind the camera and hence will be have a negative value, as in our case. The z axis controls the distance to the 2D plane.

A model is then created for the mapping of the selected 3D coordinates of the virtual image onto a 2D plane returning a set of (x,y) coordinates. The returned set of 2D coordinates also take into account the affine parameters given by the camera model. This set of (x,y) coordinates is then used as our model and the corresponding values are extracted from each input image. Given the 3D-2D model, it is possible for more than one set of 3D coordinates to project onto the same point on the 2D plane. This can cause some blurring and/or distortion. To counter these effects, a simple box filter is applied to the image.

V. SHAPE MODELING

The captured images are spherical. These are reprojected onto a 2D plane using the method described in Section IV.B. These images are initially converted to grayscale and then into binary images. The method presented in [2] is used to detect and label the various objects boundaries. Each shape is then segmented from the image and stored.

\[ F[r(s)] = R(w) \]  

where 

\[ r(s) = \sqrt{(x(s) - x_c)^2 + (y(s) - y_c)^2} \]

\[ x_c, y_c \] are the shape centroids and \( x(s), y(s) \) are the boundary coordinates

The set of (x,y) boundary coordinates for each shape is converted to polar coordinates. The distance formula described in equation 1 is used to calculate the distance between each (x,y) boundary value and the centre of the shape. These values are converted to polar coordinates. The Fast Fourier Transform (FFT) is then applied to each set of values. Rotation and changes in the starting point only affect the phase of the descriptor. All the phase information can be removed by taking the absolute values of the descriptor elements. It has been shown that the low frequency components of the Fourier Transform are sufficient for shape recognition [4],[17] and thus the entire transform does not need to be used. We found that using the first 15 Fourier co-efficients (excluding the very first component \( F(0) \)) provided sufficient discriminatory information to model a shape. \( F(0) \) is the lowest frequency term and tells us nothing about the shape; only mean position. It is the only component in the Fourier Descriptor that is dependent on the actual location of the shape. By ignoring \( F(0) \), the descriptor becomes translation invariant. The Fourier Descriptor is then normalized to remove any scaling effects.

Each shape boundary is then matched to the shapes extracted from the board using euclidean distance. The energy in the Fourier components decreases sequentially i.e. higher-frequency edge features have less energy than lower-frequency edge features. It is important that contribution of all the Fourier components are similar, therefore we artificially boost the contribution of some of the higher-frequency components. We have found that as the component number increases (which means the energy decreases), increasing the value used to boost the component works best when calculating the euclidean distance. For our experiments components 0-5 were multiplied by a factor of 5, components 6-10 by a factor of 10 and the remaining components by a factor of 15. These values were determined empirically. The shape on the board with the smallest euclidean distance to the shape on the table is considered to be the match.

VI. MANIPULATION OF OBJECTS

The robot manipulation component of the system operates by proceeding through the following states:

1) Move over the board - The robot moves the end-effector mounted camera to a fixed, preprogrammed position over the puzzle board.
2) Identify vacancy - A picture is taken and thresholded to find vacant shape (white) blobs. Blob sizes are thresholded to remove any noise from lighting effects. A random vacant blob is selected and it’s Fourier Descriptor is determined.
3) Move over the parts - The robot moves the camera to a fixed position over the parts.
4) Match vacancy to part - A picture is taken of the parts. By thresholding for intensity, their blobs are determined and processed to determine each respective Fourier Descriptor. The closest match is selected for placement.
5) Determine part 3D location - The height of the table relative to the robot is determined through measurement.
The table plane is also assumed perpendicular to the robot’s vertical axis. The position of the object is found in the omnidirectional camera image, as described in the previous step. The image coordinate can be converted into a direction ray from the center of the camera by using the camera calibration data described above. Where this ray intersects with the table plane is the \((x, y)\) position of the object. That is, the position of the object in the 2D plane perpendicular to the robot’s vertical axis, but offset by the position and rotation of the camera. This offset is known and easily removed.

6) **Execute grasp** - Principal Component Analysis (PCA) involves a mathematical procedure that transforms a number of (possibly) correlated variables in a smaller number of uncorrelated variables called principal components. PCA is mostly used for face recognition, image compression and finding patterns in high dimensional data. We use PCA on the boundary data to calculate the direction of the eigenvector with the biggest change. This direction indicates the direction of the greatest width of the object which we then when grabbing the object. When grasped this direction should be perpendicular to the direction the fingers will apply their force to ensure a better chance of the hand correctly grabbing and holding on to the object.

Once the end-effector is rotated to the orientation, it moves over the object, the fingers are opened and it is dropped until force is measured on the finger tips from the table. Then the hand is raised a specific distance to ensure that the fingers close on to the object sides. During closure of the fingers, digit joint torque is measured to determine that the object was, in fact, grasped and also to limit the force applied to the object.

7) **Place the part** - The arm is returned over the puzzle board. Using a method similar to determining the part position, above, the vacancy is localized in 3D. The end-effector is moved just above the vacancy, orientated to fit and the part is dropped. The fingers are closed and used to push the part into the hole.

The above process is repeated until all the vacancies are filled.

VII. EXPERIMENTS

A. Recognition

The shapes used in the experiments are in the form of animals which include a tortoise, whale, seal, dolphin, fish, crab and so on. There are ten shapes in total. Some shape recognition systems use complex coordinates to model the shape boundary but we opted to use polar coordinates because in our experiments this method produces more accurate results. This could be because of the choice of shapes used in our experiments. Table I shows the results obtained from both methods.

<table>
<thead>
<tr>
<th>Shape</th>
<th>Complex Coordinate Method</th>
<th>Polar Coordinate Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whale</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Seal</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Fish</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Crab</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Dolphin</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Mussel</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Snail</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Octopus</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Star Fish</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Tortoise</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

Fig. 6. The robot arm grasping an object

B. Grasping

The manipulation of the shapes, which includes grasping and moving the object was implemented using the procedure described in Section VI. In the experiments, all the shapes were placed in arbitrary positions on the table. The arm was required to localize, grab and move each shape. The arm was able to locate, grab and move every shape successfully on the first attempt. Figure 6 shows the robot arm grasping the required shape.

VIII. CONCLUSION

This paper presented the various components required to build a robust pick and place system. This included capturing images on the fly using a robotic manipulator, a shape recognition system and the process for grasping an object and moving...
it to the correct end position. The shape recognition system presented uses Fourier Descriptors to model the boundary information with very good results given the similarity of the shapes. The system used to grasp and move the object is also very effective. This system could easily be placed in an assembly line with very efficient results.

IX. FUTURE WORK

We would like to extend the system so that if the incorrect object is recognized, the robotic arm realizes that the object does not fit the required shape. It would then proceed to take the incorrect shape and place it back on the table and pick the next shape with the smallest euclidean distance. This would continue until all the shapes were correctly placed in the board.

REFERENCES