Vision Strategies for Robotic Manipulation of Natural Objects

Chee Kit Wong, Patrick P.K. Lim, Roger Clist and Raymond ShuHe Liu
Industrial Research Ltd
24 Balfour Road, Parnell, Auckland, New Zealand
{k.wong, p.lim, r.clist, r.liu}@irl.cri.nz

Abstract

This paper describes a vision guided robotic platform that will be used for the handling of natural products in a multi-layer stack that is typical in a bin situation. It is designed to automatically recognise non-rigid objects, to estimate their pose, and to select suitable picking points using depth information from a 3D camera. We adopted a novel approach by using a topographical method to segment the objects and then with elevation contours to build volumetric data for each object. When the periphery for each object is established, we introduced strategies by which decisions will be made for task planning. These strategies identify the appropriate object to pick and the optimum picking position. Results of the experimental trials demonstrate the robustness of this approach for specific types of natural products.

1 Introduction

The automatic grasping of parts in a bin with the use of a robotic arm is commonly known as robotic bin picking. It is a complex task that is usually carried out by a vision guided robot (VGR). Recent developments in VGR have been in the vision technology. Now with low-cost 3D vision, complex issues can be overcome with novel approaches in image processing and analysis.

1.1 Literature review

The challenge of robotic bin picking of rigid parts has been around for a long time (see [Lowe, 1987] and for recent examples, see [Dupuis et al., 2008; Perreault and Olivier, 2007; Bloss, 2006]). Rigid parts are generally a known quantity where shape and features are invariant. Although a main issue had been with image occlusion but with modern 3D stereoscopic imaging, to a certain extent this has been solved. Partial occlusion could be addressed through pattern matching. Boehnke [2007] used a 2.5D image to generate the outline of objects in order to perform the common pattern matching. Some examples currently employed are time of flight sensors [Fuchs and May, 2008; May et al., 2006] and single stereo vision [Seal et al., 2005; Watanabe et al., 2007].

Robotic bin picking with non-rigid objects such as fruit introduces a number of issues. Fruit is largely variable in size, shape, firmness and sometimes texture. One example where this can be employed is in the kiwifruit packing industry. In the repacking process which goes right through the year, fruit from the coolstore is loaded in bins. The fruit currently go through a screening process to detect blemishes or bruises before undergoing quality inspection and then sorting to be packed into cartons.

Using traditional techniques employed for rigid objects would simply not work and alternative methods must be developed. Foresti and Pellegrino [2004] developed a vision based system to recognise deformable objects by segmenting colour images based on texture. With 3D data, Kirkegaard and Moeslund [2006] applied harmonic shape context (HSC) features and a graph-based scheme. Other examples of 3D vision was applied to manipulation of deformable objects [Firai et.al, 2001; Wögerer et al., 2005] and visual servoing [Kumar and Jawahar, 2006].

1.2 Overview of VGR platform

Robot system

Our system comprises a 6-axis IRB 2400 robot, custom gripper, stereo camera and remote PC as shown in Figure 1. The S4Cplus controller runs ABB's Robot Application Protocol (RAP) which executes commands from the remote PC via a TCP/IP ethernet connection.

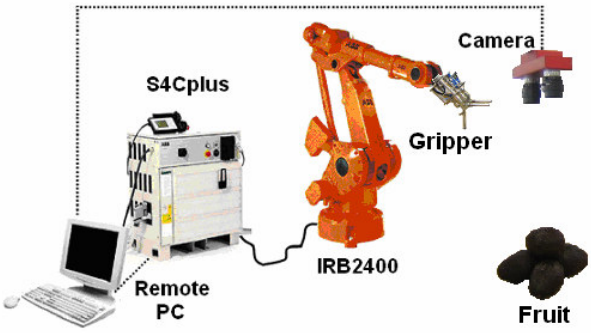

Figure 1: Vision guided robot system components

Gripper tool

A custom 3-fingered end-effector was designed and fitted to the robot mounting flange. It is a pneumatically actuated gripper comprising a moveable thumb and two parallel and opposing fingers controlled by digital signals from the robot controller. When the remote PC sends the coordinates and motion commands to the robot controller, the gripper approaches the target object at the appropriate pose to pick it up.

Stereo camera

The target product to be handled by the robot is multilayer, and therefore the vision system must report product height information. We decided to use a stereo camera for this purpose, and selected an STH-MDCS camera from Videre Design. The camera has twin imagers fitted with 6mm lenses, with a baseline of 9cm. Images are stored in 8-bit grayscale format and are transfered to the host PC through IEEE-1394 firewire ports. The stereo images are then processed using functions from SRI International's Small Vision System (SVS) library and source code development kit to produce point cloud data.

Calibration

The stereo camera was calibrated for internal (lens distortion and decentering) and external (camera spatial offset) parameters using the SVS *smallvcal* application with an A3 checkerboard of 54mm squares analyzed as recommended by the camera manufacturer. Additional calibration factors were determined using test objects at two diverse spatial locations so that the stereo cartesian coordinates were precisely mapped to robot cartesian space.

1.3 Overview of paper

In this paper we apply a topographical segmentation method to the point cloud of a 3D static image overlooking a multi-layer stack of fruits. The method builds volumetric data and establishes a periphery for each identified object. Fused with apriori knowledge, this method is a robust way to discriminate between objects. Next we present strategies to select which object to pick based on the available picking positions termed as free space. This approach was implemented and tested on our VGR platform comprising a 3D camera, robot and a multi-layer stack of avocados. Results of the experiment indicate the robustness of this approach.

2 Processing point cloud data

Point cloud data provided by the stereo camera image acquisition and processing library describes a set of vertices in three dimensions, represented by cartesian values. These data are processed in three steps: pre-processing, object identification, and blob analysis. The objective is to identify objects, along with their respective features, such as centroid, etc. Sections 2.1 to 2.3 describe these processes in more detail.

2.1 Pre-processing

Pre-processing prepares the point cloud data so that accurate analysis can be performed during later stages of processing. The following procedures are performed during pre-processing:

- Removal of known errorneous data. When the stereo camera system is not able to compute distance values, it stores the data as (0,0,0). Such values were replaced by NAN (not a number) to prevent them being part of calculations.
- Removal of data that has a depth (z-value) that is greater than the distance between the stereo camera and the table in the real world. Since the table is the base of the experiment, any data beyond this distance is incorrect, having been created by false correspondences. Such values were likewise replaced with NANs.

Pre-processed point cloud data is then passed to the next module for object identification.

2.2 Object Identification using contour analysis

To be able to successfully grasp an object, an accurate identification algorithm is crucial to successfully manipulate an object. This is especially true when dealing with natural objects.

Firstly, natural objects have irregular shapes and sizes. It is not often two objects are identical and hence, every fruit has different location for optimal pickup. If the shape of an object is computed incorrectly, then the manipulator will most likely not achieve a stable grasp of an object.

Secondly, natural objects are easily damaged. Incorrect identification of a grasping point may result in damaged objects. The problem is further compounded when dealing with objects that are located at various heights. If the depth reported is not deep enough, the robot will fail to grasp the object. Conversely, if the depth is too deep, the gripper will damage the object that is located directly below.

Hence, the process of object identification is not straightforward, especially when objects are touching each other.

Essentially, we identify objects by accumulating contours grown from seed peaks in the point cloud data. That is, the highest peak is initially identified from the point cloud. The contour for this height is computed and all the points are categorised based on proximity. Below a set threshold, points are considered to be belonging to the same object. Else, a new object is created. We termed this process *slice analysis*.

Slice analysis is performed on the point cloud data at different heights, starting from the highest point down to a pre-determined depth. Intermediate heights are set empirically. Smaller distances between each slice generates better precision but at the same time, increases processing time.

Slice analysis is especially useful for segmenting objects that are touching. Touching objects can be easily confused to be a single item as neighbouring objects share points along the boundary region. Slice analysis is able to handle this problem without needing further processing. However, this approach does assume that objects have a single peak.

As noted above, analysis of the point cloud data is performed only down to a certain depth from the topmost detected peak. The reason is simple: it is naturally easier to manipulate objects located on the top layer. Processing information beyond the fruits on the top layer is unnecessary as the additional information does not assist in the pickup process. In fact, managing the information this way provides for a more accurate object identification, as well as decreasing the time it takes to process a set of data.

A summary of the object formation algorithm is as follows:

1. Detect highest point from point cloud data
2. Compute contour points for this height
3. Group points based on their proximity to each other
4. Create a new object with points that do not belong to existing objects
5. Increase depth by a pre-determined constant

6. Repeat steps 2-5 until processing reaches the required depth from the starting height
7. For each object, apply convex hull computation to determine the object's boundaries
8. Plot all objects, each filled with different colours
9. Perform blob analysis on the 2D plot thus formed, as described in Section 2.3.

2.3 Blob analysis using 2D image processing

Processing can now move into the 2D domain, since we are more interested in object perimeters (for grasping) rather than their 3D volumes.

In two dimensions, each object is basically a silhouette which has a centroid, a major axis and a minor axis. The major and minor axes pass through the centroid and define the lines about which the turning moments are minimal and maximal, respectively. For manipulation purposes, the point where the minor axis and the perimeter of the object intersects is very interesting, as gripping across this axis will permit the object to be lifted with a degree of stability.

Two-dimensional blob analysis is used to locate the centroid and orientation of the axes for a particular object. In addition, other information such as area and bounding box dimensions are computed as well.

The locations of the points of intersection of the axes with the perimeter (or *pivot points*) are not generated directly in a traditional blob analysis. Therefore we developed a heuristic method which builds on a traditional blob analysis by tracing the axes radially outwards from the centroids until the perimeter is first reached, as shown in Figure 2.

The blob analysis algorithm was adapted from an open-source image processing package, ImLab[1]. The program extracts information such as blob size, centroid position, major and minor axis slopes and lengths, from which the x, y locations of the four pivot points of each object are derived.

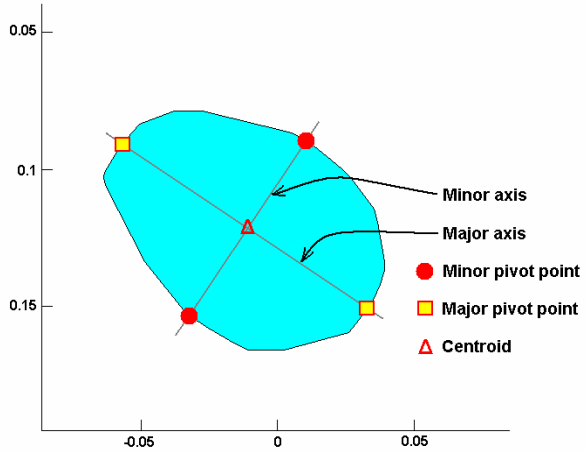

Figure 2: Axes, pivot points and centroid computed during blob analysis

For the purposes of this project, it is assumed that each axis intersects the object's perimeter at only two points. In addition, the algorithm also filters out any object that is smaller than a given size.

[1] IM is a toolkit for digital imaging developed at Tecgraf/PUC-Rio, Brazil.

The following steps summarise the blob analysis procedure:

1. The input 2D plot is converted to a grayscale 8-bit image. Individual colours are assigned to unique gray values (widely separated within the 1-255 range).
2. The grayscale image is thresholded at 0, creating a binary image with the background rendered black and the objects rendered white.
3. Small objects of area less than 1000 pixels are removed.
4. The size of the image is reduced to fit the internal object limits.
5. Object regions are found by performing a 4-connected blob analysis on the above image. Objects are labeled with values 1, 2, 3, etc. according to the order in which they are found, from bottom left to top right.
6. Objects are measured. The centroids are located and the slopes of the axes are calculated.
7. The pivot points are found by tracing the axes in the image radially outwards from the centroids until the object boundaries are reached.

Since ImLab operates using pixel coordinates and cell size, all x,y point locations are converted to real-world coordinates when exported to the next process.

3 Strategies for object selection

Once objects have been identified, the system needs to determine 1) the best object to pick and 2) the best pickup location for the particular object.

The manipulation strategy is based around a 3-fingered end-effector built for grasping avocados. The strategy requires the 'thumb' of the gripper to move into any available space around the fruit. Two additional fingers hover above the object and when the gripper is in place, all fingers are actuated to grasp the object. Figure 3 shows the gripper being positioned to grasp an object.

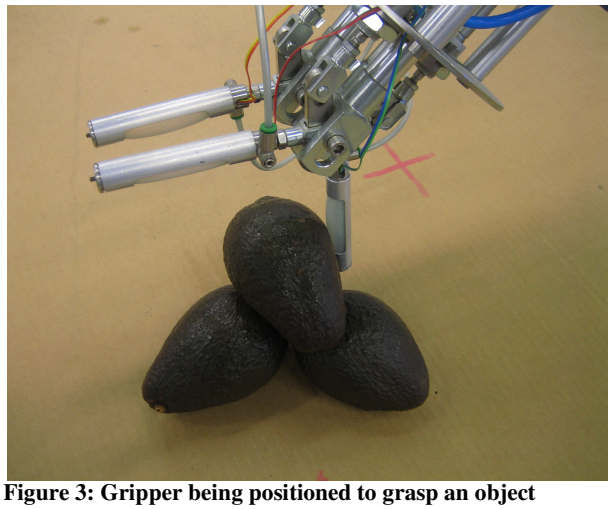

Figure 3: Gripper being positioned to grasp an object

Given this strategy, it is essential to firstly determine what free space is available around all objects, and secondly, determine which of those would provide for the most stable manipulation. This in turn determines the object to be picked up. The rest of this section describes the selection process in more detail.

3.1 Computing free space

Free space is computed based on the proximity between adjacent objects. For this, test points are placed at set intervals along the periphery of each object. For each test point, a circle is drawn. If any points from neighbouring objects are within this circle, then the test point would be considered unsuitable for pickup. The radius of the circle is set to be slightly larger than the diameter of the thumb.

3.2 Selection criteria

Two criteria were implemented for ranking the suitability of a pickup point:

- Proximity to major and minor pivot points
- Proximity to the nearest object.

For each of these criteria, each pickup point is assigned a score between 0 and 1, with 0 being most unsuitable and 1 being most suitable.

Whilst we have implemented only two criteria, this approach provided the flexibility to add further criteria as needed. The following subsections describe the criteria in more detail.

Criteria 1: Proximity to major and minor pivot points

From experimentations, it was found that the best approach to grasp an object is to position the thumb near the intersection between the minor axis and the perimeter of the object, which we termed a minor pivot point (see Figure 2). Once positioned, the gripper has to align the top fingers across the object by pointing towards the object's centroid. It was also found that the stability decreases as the thumb moves away from the minor axis and towards the major axis, with the critical point approximately midway. Past this point, the gripper has significant issues in grasping the object.

From this observation, pickup points that are closer to the minor axis are much more desirable than those that are further away. Conversely, a pickup point is deemed less desirable if it is closer to the major axis or its pivot points.

Instead of having two measures: one for the distance to the minor axis and another for the distance to the major axis, the ratio $\frac{d_{\min or}}{d_{major}}$ is used instead. Smaller values indicate greater suitability of a pickup point. An advantage of this *distance ratio* is its impartiality to the size of an object.

Nonetheless, the distance ratio in its pure form is not a good scoring system. We needed a scoring system which equates to 1 (the highest value) when the ratio is 0 i.e. the pickup point is on a minor axis pivot point. As the pickup point moves away from the minor axis (and towards the major axis), the score decreases.

The following equation is used to generate the score:

$$\exp\left(-\sigma\times\left(\frac{d_{\min or}}{d_{major}}\right)\right)$$

σ determines the spread of the score for different distance ratios, and was selected based on the desired score when the distance ratio is 1. As mentioned before, a distance ratio of 1 is a critical value as it defines the point where the grasping stability of an oval-shaped object changes dramatically.

Criteria 2: Proximity to the nearest object

The second criteria is based on the proximity of a pick up point to the nearest neighbouring object. The algorithm rewards pickup points that have large clearance for the gripper's finger to be inserted and penalizes pickup points that have smaller spaces.

The following equation is used to compute the score based on the size of gap available:

$$\frac{1}{1+\alpha\exp((\rho-x)\beta)}, \text{ where}$$

x - distance to the closest object in millimetres
ρ - is the minimum allowable distance in millimetres
α - sets the score at the critical point when $x = \rho$
β - controls the slope of the transition region between score of 0 and 1

If the distance from a pickup point to the nearest object is below the minimum allowable distance ρ, the score will be low. As this distance increases above ρ, the score increases. However, this increase in score stops after a certain distance as the extra distance does not make any impact on the ease of pickup.

3.3 Object selection

After the scores for each criteria are determined, an overall score for each pickup point is computed using the equation:

$$score_{overall} = (score_{criteria\,1} * weight_{criteria\,1} + score_{criteria\,2} * weight_{criteria\,2}) / 2$$

This equation allows for weightings to be assigned to the respective criteria, based on their importance. Once the overall scores are computed, the pickup points are ranked and the one with the highest score is selected. If there are several objects with the same highest score, then a random object will be chosen.

3.4 Gripper orientation

The robot tool coordinate system is configured such that the robot's Tool Centre Point (TCP) is calibrated to the base of the gripper thumb. With the z-axis protruding from the thumb and the x-axis towards the other fingers, it is a convenient way for efficient robotic motion planning. It therefore remains for the gripper to be rotated about the z-axis to align with the optimum orientation to grasp the selected fruit.

At this stage we know the cartesian coordinates of the pick up point and the relative x,y coordinates of the centroid, which may be assumed to be close to the fruit's centre of mass. These coordinates permit the orientation of the gripper to be determined, as shown in Figure 4 below.

4 Experimental Results

4.1 Experimental setup

Our VGR environment comprises a stereo camera, robot and PC as described in Section 1.2 previously. The basic requirement is to process stereo images to produce point cloud data.

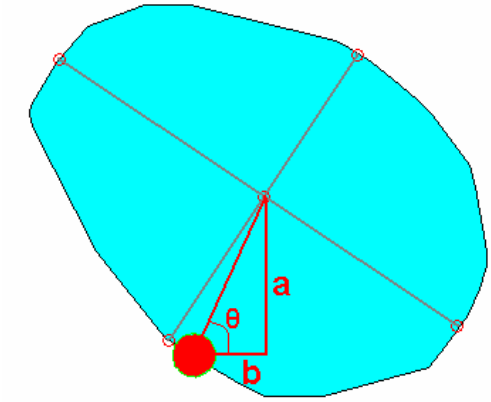

Figure 4: Diagram showing computation of the orientation of the gripper based on the centroid of the object

However, the methods described here are not restricted to the type of sensor used, as long as a set of point cloud information is available. In fact, a more accurate sensor that does not rely on texture information would increase the performance of the system.

4.2 Image Texture

Stereo processing algorithms calculate range by recognizing features in a scene when viewed from two viewpoints. If the physical positions of the two viewpoints are known, then the spatial disparity of feature points in the two images permits distances to be calculated trigonometrically.

It therefore follows that the scene must contain recognizable features, or texture, in order for the stereo algorithms to operate. If the scene has few features, then it may be possible to create texture by projecting non-uniform illumination, or patterns, on to the surfaces.

Images of fruit can be hard to process in stereo if their surfaces lack striations. Many fruit have a fine stipple texture but its uniformity renders them difficult for stereo processing.

Initially we wrapped such fruit with bands of white masking tape on which dark irregular wavy lines were drawn. This proved to be excellent for the experiments, as the wavy lines adequately covered the top and side surfaces of each fruit, and the irregular form reduced the opportunity for false stereo correspondences.

In another approach, we projected a structured light pattern using a data projector mounted horizontally (to meet its operational requirements) and its beam deflected vertically downwards using a mirror mounted at 45 degrees to the vertical axis. A variety of patterns created in a drawing program could then be projected on to the fruit surfaces, either as black on white or white on black.

We experimented with two projected structures: a parallel set of wavy lines, and a matrix of isolated, but varied, icons. Both forms of structure operated well on topmost surfaces, but the wavy lines were superior because they dropped down the sides of the fruit, and more clearly identified the fruit boundaries. Even so, the perceived edges of the fruit were not faithfully representative of the true fruit perimeters, especially if the fruit were close or touching.

For this reason, the bulk of the experiments were carried out using fruits wrapped with tape and marked irregularly.

4.3 Experimental results

When objects are placed randomly into a bin, they can have one of many different arrangements. These arrangements can be broadly classified into three categories:

1. Objects situated far apart
2. Objects in close proximity
3. Objects stacked to form multiple layers

The sections which follow report experimental results for these situations.

Objects situated far apart

Objects located far apart is the easiest situation to handle. When objects are singulated, they are easily identified, processed and picked up.

An example of this situation shown in Figure 5. This image was captured through the camera's left lens. The objects have random orientations, but are spaced far enough for the gripper to have full access to any object and without difficulties.

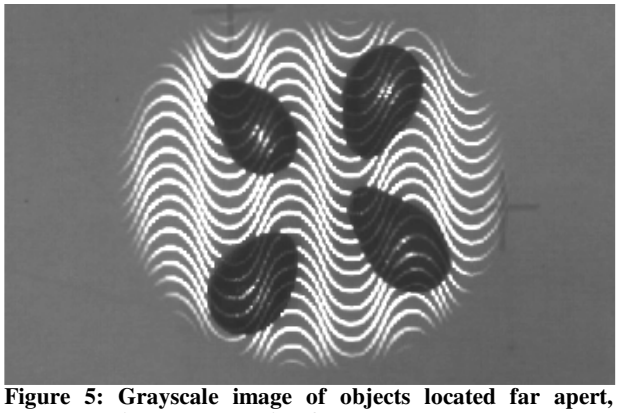

Figure 5: Grayscale image of objects located far apert, captured using the camera's left lens

A mesh plot of the computed point cloud data can be seen in Figure 6. The plot shows each object is well defined. Nonetheless, some information is still missing, as represented by the white coloured holes in the objects. In the plot, the temperature of the colour represents a point's proximity to the camera. That is, hotter colours signify closer points and cooler colours signify points that are located further away.

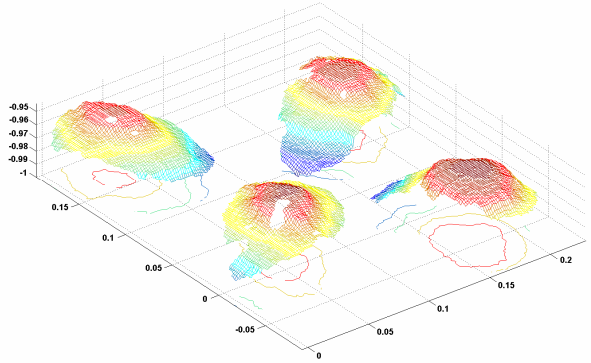

Figure 6: Mesh plot of the point cloud data with objects positioned at a distance from each other

Figure 7 is a two-dimensional plot of the point cloud data that was extracted from the slice analysis. One can clearly see each of the objects available for picking.

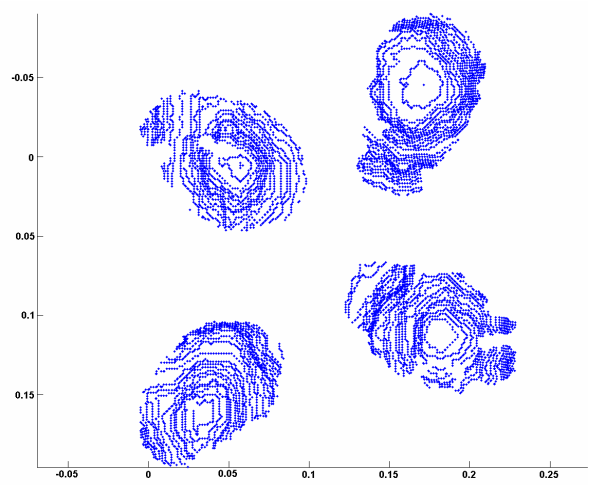

Figure 7: Plot of data extracted from all iterations of slice analysis

The result of processing is shown in Figure 8. Four objects were detected and are represented by blobs, each with a different colour. The grey lines represent the major and minor axes that were computed for each object.

Locations that would allow for the gripper to access for pickup are represented by green circles. It is obvious that for this situation, it has been identified that each object can be picked up from any part of the object. The green circles that have been filled with black are locations that have been calculated to be suitable for pickup. That is, the overall scores for these locations exceed the minimum value allowable for a point to be classified to be suitable. Finally, the red solid circle represents the location that was chosen from the computation. For this situation, the final point selected is more or less a random event as there are a number of suitable candidates.

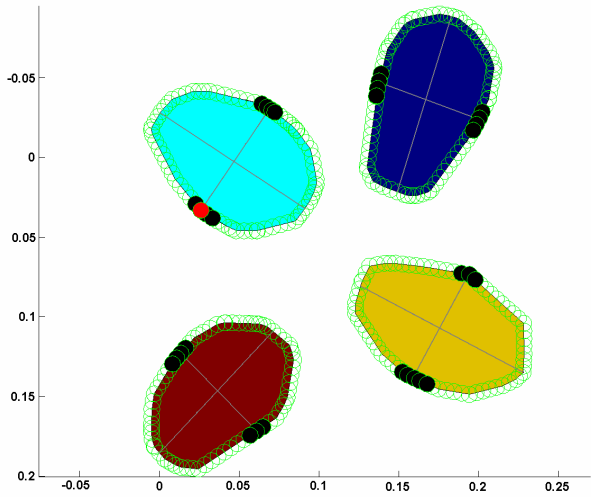

Figure 8: Plot showing the objects identified, shown in different colours. The lines through the objects are the major and minor axes and the green circles signify locations that the gripper is able to access. Green circles that have been filled with black are locations that have been computed as suitable picking points whilst the selected point is represented by the red solid circle.

Objects in close proximity

The second scenario presented here consists of objects that were placed in very close proximity. At least one side of the fruit is obstructed by another, and hence pickup cannot be performed at any location. This scenario can be seen in Figure 9 with its corresponding mesh plot shown in Figure 10.

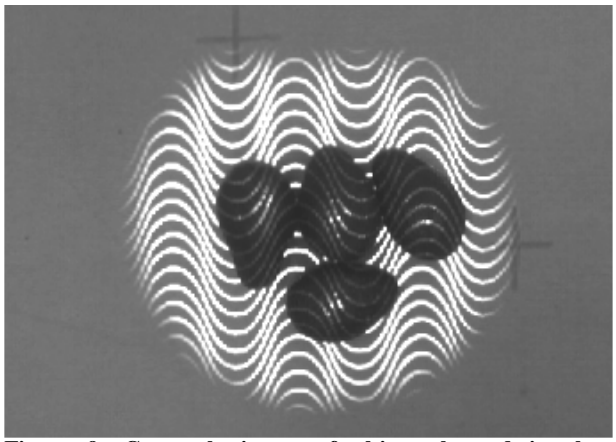

Figure 9: Grayscale image of objects located in close proximity

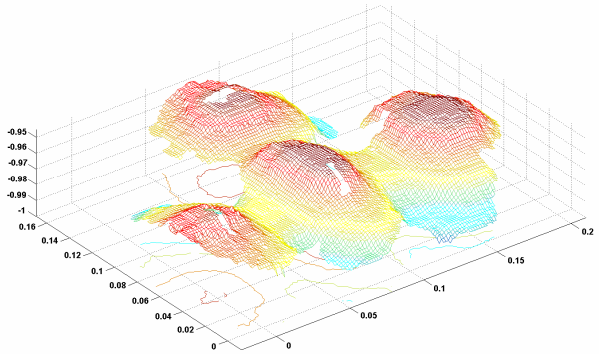

Figure 10: Mesh plot of the touching objects

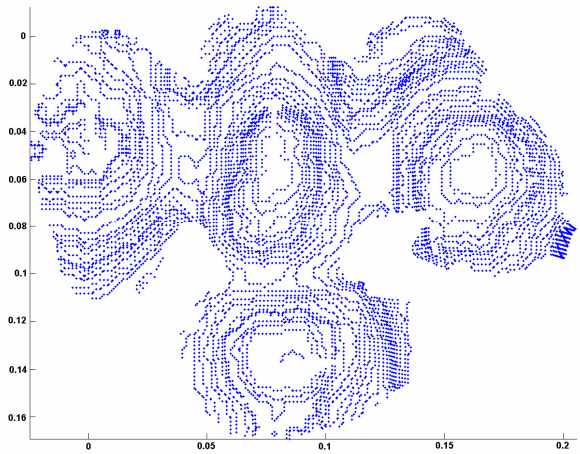

Figure 11: Plot of data extracted from all iterations of slice analysis essentially shows one large connected object

A 2D plot of the point cloud data can be seen in Figure 11. It is evident in the diagram that it is difficult to recognise the individual objects especially around the

regions where the objects touch each other. Therefore, recognising the objects here is not a straight forward task, especially when the shape of the objects plays an important role on how the gripper would attempt to pickup an object.

The identified objects are shown in Figure 12 below. Like before, four objects were detected. However, due to the proximity of the objects, the gripper is not able to access all locations around the objects. The locations that are not accessible are represented by red unfilled circles. For the same reason, there are a lot less suitable locations for the system to choose from.

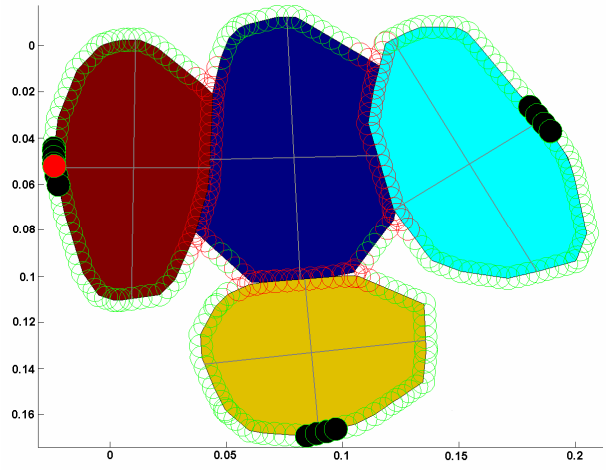

Figure 12: Plot showing identified objects in various colours. Lines through the objects depict the major and minor axes of each object. As before, the green unfilled circles signify locations that the gripper is able to access. In addition, red unfilled circles are now visible and these represent locations that are not suitable for pickup. The black-filled green circles are locations that have been computed as suitable picking points whilst the selected point is represented by the red solid circle.

Objects stacked to form multiple layers

The final arrangement presented here involves arranging the objects in a stacked configuration, as shown in Figure 13. Figure 14 shows the mesh plot of the objects. As seen in the mesh plot, the four objects look like one large object. Hence, it is difficult to identify the individual objects if all point cloud data were processed.

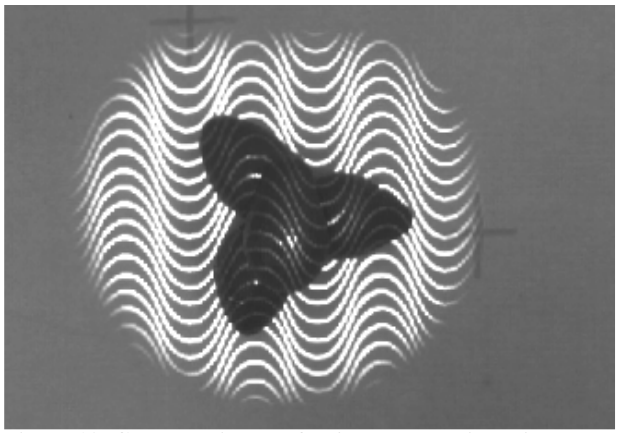

Figure 13: Grayscale image of objects stacked in a pile

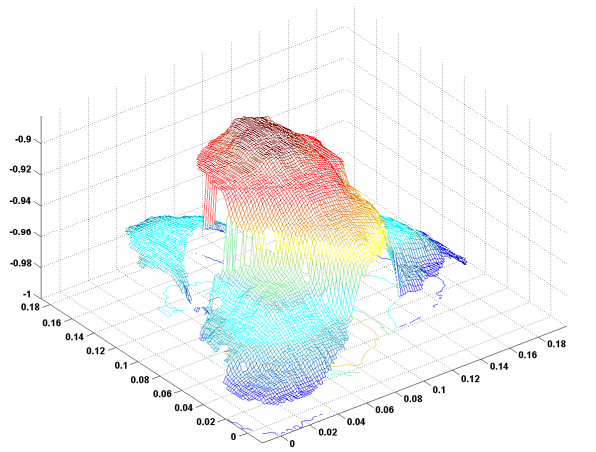

Figure 14: Mesh plot showing one large object which shows the difficulty in identifying each object from the point cloud data

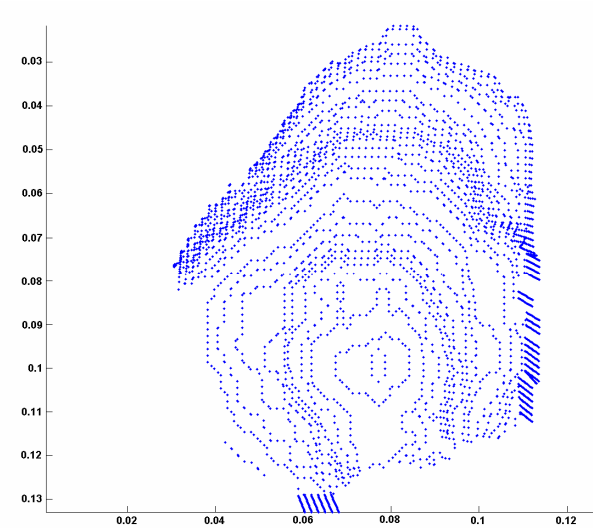

Figure 15: Plot of data extracted from all iterations of slice analysis essentially shows only the top object is used

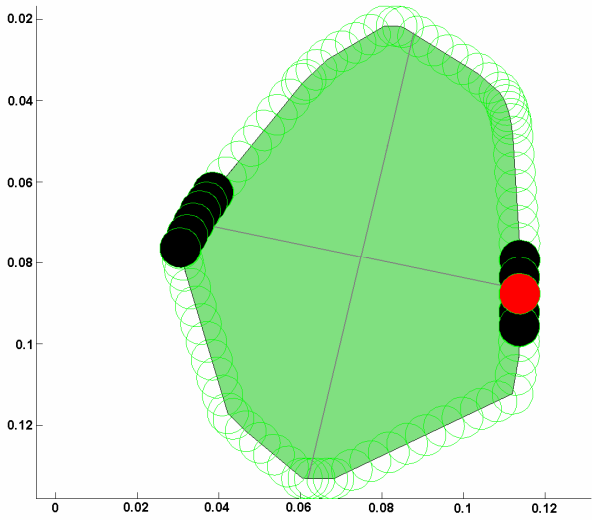

Figure 16: Plot showing a single object formed. Even though other objects exist underneath, this object is unobstructed

and hence, the gripper is able to access the fruit at all locations (green unfilled circles). The black solid circles are locations that have been computed as suitable picking points whilst the selected point is represented by the red solid circle.

It is also worth noting that more than one object can exist on the top layer. Furthermore, the system proposed works with more than two layers as well. As mentioned earlier, processing only occurs up to a set depth. Hence, lower layers do not affect the performance of the system.

Figure 15 shows that only point cloud data from the top object has been extracted for processing, with the blob formed shown in Figure 16.

5 Conclusions

We have presented a novel topographical slicing method and have applied it to 3D point cloud data generated by a stereo camera. This approach was demonstrated through the recognition of natural objects arranged in a stacked pile. Experiments conducted in various scenarios showed the robustness of this approach to discriminate between objects. Also by applying certain selection criteria, the optimum picking pose for a custom made gripper was determined. It is envisaged other grippers could be applied with the appropriate set of criteria imposed. It is modular and forms one of the building blocks for a vision guided robotics system.

References

[Bloss, 2006] Richard Bloss, "Smart robots that picks parts from bins", Assembly Automation, Vol. 26, No. 4, 2006, pp 279 – 282.

[Boehnke, 2007] K. Boehnke, "Object localization in range data for robotic bin picking", Proceedings of the 3rd Annual IEEE Conf. on Automation Science and Engineering Scottsdale, AZ, USA, Sept 22-25, 2007.

[Dupuis et al., 2008] Donna Dupuis, Simon Leonard, Matthew Baumann, Elizabeth Croft, Jim Little, "Two-Fingered Grasp Planning for Randomized Bin-Picking", Proceedings for the Robotics: Science & Systems 2008 Manipulation Workshop - Intelligence in Human Environments, 2008.

[Firai et al., 2001] Shinichi Firai, Tatsuhiko Tsuboi and Takahiro Wada, "Robust grasping manipulation of deformable objects", Proceedings of the 4th IEEE Int. Symp on Assembly and Task Planning Soft Research Park, Fukuoka, Japan, May 28-29, 2001.

[Foresti and Pellegrino, 2004] Gian Luca Foresti, Felice Andrea Pellegrino, "Automatic visual recognition of deformable objects for grasping and manipulation", IEEE Trans on systems, Man and Cybernetics – Part C: Applications and Reviews, Vol. 34, No.3, August 2004.

[Fuchs and May, 2008] Stefan Fuchs and Stefan Man, "Calibration and registration for precise surface reconstruction with Time-Of-Flight cameras", International Journal of Intelligent Systems Technologies and Applications , Volume 5 Issue ¾, Nov 2008.

[Kirkegaard and Moeslund, 2006] J. Kirkegaard and T.B. Moeslund, "Bin-picking based on harmonic shape contexts and graph-based matching", Proceedings of the 18th International Conference on Pattern Recognition, ICPR2006, pp. 581-584.

[Kumar and Jawahar, 2006] D Santosh Kumar and C. V Jawahar, "Visual Servoing in Presence of Non-Rigid Motion", Proc. Of the 18th Int. Conf. on Pattern Recognition – Vol. 4, ISBN ~ ISSN:1051-4651, 2006, pp 655 – 658.

[Lowe, 1987] David G. Lowe, "Three-dimensional object recognition from single two-dimensional images", Artificial Intelligence, Volume 31, Issue 3, pp 355-395.

[May et al., 2006] Stefan May, Bjorn Werner, Hartmutt Surmann and Kai Pervolz, "3D time-of-flight cameras for mobile robotics", IEEE/RSJ Int. Conf on Intelligent Robots and Systems, Beijing, 2006.

[Perreault and Olivier, 2007]Louis Perreault and Pierre Olivier, "Bin-picking system for randomly positioned objects", US Patent 7313464 Issued on December 25, 2007.

[Seal et al., 2005] Jonathan R. Seal, Donald G. Bailey, Gourab Sen Gupta, "Depth Perception with a Single Camera", 1st International Conference on Sensing Technology, November 21-23, 2005 Palmerston North, New Zealand.

[Watanabe et al., 2007] Takahashi Watanabe, Akira Kusano, Takayuki Fujiwara and Hiroyasu Koshimizu, "3D Precise inspection of electronic devices by single stereo vision", MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, Japan.

[Wögerer et al., 2005] Christian Wögerer, Gerald Nittmann, Petra Tatzer, "Intelligent Manipulation of Non-Rigid Parts in industry applications", Proc. of the 2005 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, California, USA, 24-28 July, 2005.