Experimental VGR for 3D Object Tracking and Contour Following

Chee Kit Wong and Patrick P.K. Lim
Industrial Research Ltd
24 Balfour Road, Parnell, Auckland, New Zealand
{k.wong, p.lim}@irl.cri.nz

Abstract

This paper describes an approach that analyses point cloud information from a stereo camera to enable a vision guided robotic platform to perform 3D object tracking and contour following. The process uses a combination of 2D image processing, as well depth analysis to analyse and identify a target object as its position and orientation, or contour changed along the x, y and z axes. PID controllers were used to react to these variations and actuate the manipulator robot, effectively following the target object or surface contour. For object tracking, the VGR system was able to track a specified target amongst identical objects, even when it was partially occluded. Similarly, contour following experiments demonstrated the robot's ability to keep a set distance, whilst remaining orthogonal to the surface of the object. Hence, the experiments demonstrated the robustness of this approach. However, it is important to note that the algorithm does rely on consistent and accurate information from the stereo camera.

1 Introduction

Vision Guided Robots (VGR) have been widely implemented in various industries. Mostly in the manufacturing environment where conditions can be controlled and object position is deterministic. When the object is stationary such as fruit in a bin, its pose can be easily determined and robotic picking can be readily executed [Wong et al., 2009].

However, when the object is moving such as lamb carcasses on a processing chain, it presents several challenges [Lim et al., 2010]. A critical aspect when tracking an object whether for processing or picking is the object pose. In most cases, the object is constrained within the workspace allowing a robotic system to determine the object position reliably at any given time. Constraint is achieved with a conveying system that restricts the movement to a known trajectory and tracking can be performed with an encoder delivering precise position of the object. Such is the case with automated packing of food products like biscuits. With more complex objects like meat portions, the orientation is also an important consideration and generally the product is pre-orientated for easier robotic gripping.

When an object is delivered unconstrained, it poses greater challenges. In applications such as quality assessment and harvesting of fruit products, the objects are largely unconstrained. In general, tracking of these objects would require both position and orientation information captured at a relatively high frequency. Applications are diverse such as inserting an endoscope into a bore hole [Biegelbauer and Vincze, 2006], pig tracking [Frost et al., 2004] and fruit picking [Baeten et al., 2007].

In this study, the primary aim is to develop a concept to perform non-contact tracking of an object within a workspace for the purpose of picking the object while in motion. As with natural products, the object is assumed to be non-symmetrical and largely unconstrained. Panin and Knoll [2006] presented an automatic real-time system for 3D object pose tracking based on point feature matching in image sequences. Collewet and Marchand [2008] focused on target tracking by visual servoing which does not rely on geometric features but rather relies on luminance of all pixels which does not require any tracking or matching process. Silveira and Malis [2007] performed real-time visual tracking by developing a robust method based on a proposed model of illumination changes together within appropriate geometric model of image motion.

The secondary aim which is closely related to the above is to follow the contour of an object in 3D space. Robot contour following is necessary in applications where the end-effector tool must conform to the contour of an object during processing. An example is seam welding. For known contours, the path can be easily programmed. For unknown surfaces, it is beneficial for a VGR to have the capability to do it automatically. Lim et.al [2002] demonstrated a contact based telerobotic system that followed the contour of an object. Prabuwono et.al [2008, 2010] introduced autonomous tracking using the gradient, staircase and sweeping radius method with an industrial robot.

In this study, we develop a VGR system with a stereo camera mounted on a robot end-effector to derive the object pose in real time such that it is able to track an object or follow the contour of an object in 3D space. We present vision strategies and robotic control for implementation on an experimental system. Section 2 describes an overview of the VGR system which supported the methods for tracking and following tasks presented in section 3. Following that section 4 documents the experiments conducted and results. The paper concludes in section 5 and presents a discussion of further work.

2 Overview of VGR System

The vision guided robot system consists of a six-axis manipulator robot, a stereo camera, lighting, a personal computer (PC) for point cloud analysis and another PC that performs as the PID controller. The stereo camera is connected to the point cloud analysis PC through a firewire connection whilst the PCs and IRB2400 robot are all connected using the TCP/IP protocol through a network switch.

The complete system is shown in Figure 1 and individual components are described in the following subsections.

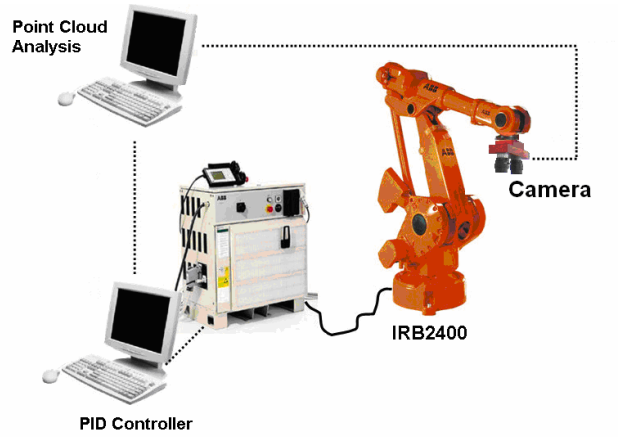

Figure 1: Vision Guided Robot System

2.1 Manipulator Robot

The manipulator robot is a 6-axis IRB 2400 robot from ABB. The S4Cplus controller runs ABB's Robot Application Protocol (RAP) which executes commands from the remote PC.

2.2 Stereo Camera

Three-dimensional information is required to perform object tracking in six degrees of freedom. For this, a STH-MDCS stereo camera from Videre Design was used. It has twin imagers fitted with 6mm lenses, configured with a fixed baseline of 9cm. Images are stored in 8-bit grayscale format and are transferred 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 software development kit to produce point cloud data. Multiple parameters can be adjusted to achieve different levels of sensitivity and depth range, to suit the target application.

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 mapped to robot cartesian space.

The stereo camera is attached to a mounting plate, which is fitted to the robot mounting flange. The mounting plate allows for additional tools to be fitted, when required. Since there is no end-effector tool at this stage, the left lens of the stereo camera is calibrated to be the tool centre point of the manipulator robot. Consequently, the objective of the object tracking and following is to keep the centre of the image of the left lens, aligned to the centre of the tracked object. The stereo camera and its configuration on the robot are shown in Figure 2.

It is worth noting that this system is not exclusive to stereo camera and will work with other sensing devices that can generate point cloud data.

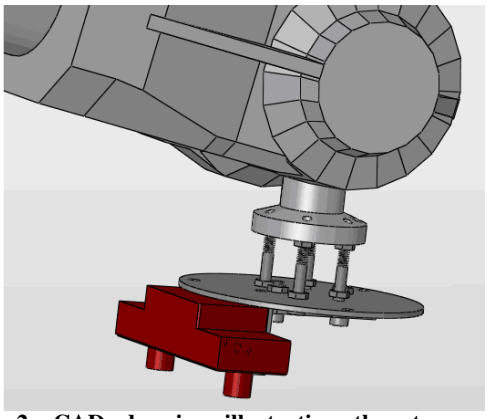

Figure 2: CAD drawing illustrating the stereo camera mounted on the manipulator robot.

2.3 Lighting

As with any vision based systems, it is important to operate with stable lighting conditions. Various types of lighting was tested, including fluorescent, incandescent and LED. The desire is to obtain even lighting conditions throughout the work area, so that the stereo imaging algorithm is able to operate at the optimum performance, and unaffected as much as possible by shadowing and external lighting. More importantly, the lighting must not interfere with the image sensors. Hence, a direct current or high frequency light source is more suitable to suppress flickering effects.

LED lighting seemed to be suitable for this application but initial tests with LED spotlights and panels demonstrated the difficulty in achieving constant light source throughout the work area. Some areas were too bright, causing saturation, whilst others were not bright enough.

To solve this issue, neon-flex lighting was used and mounted as part of the end effector. This product essentially consists of a series of LEDs set in bendable plastic tube. This flexibility allows it to be easily shaped and mounted. More importantly, it also acts as a light diffuser.

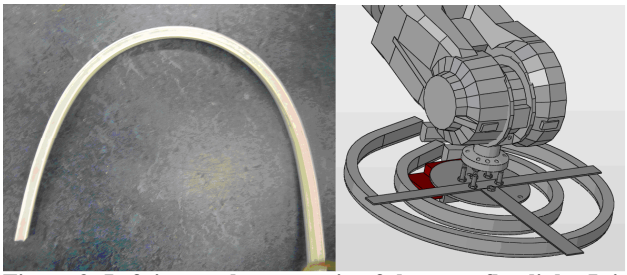

Figure 3: Left image shows a strip of the neon-flex light. It is flexible and can be cut to required length. Right image is a CAD illustration of the light mounted on the manipulator robot

The neon-flex light is commonly used for sign or feature lighting for buildings and landmarks and runs on a DC power supply. A couple of concentric circles were formed using the neon flex and were mounted onto the mounting plate, surrounding the stereo camera. Hence, the light would point in the same direction as the stereo camera. This reduces the effects of shadowing, and since the light has a soft plastic shelf, it cannot be easily damaged.

The neon-flex light and its mounting configuration are shown in Figure 3.

3 Tracking and Following with 6 DOF

In essence, there are five parts to this process:

1. Analyse target
2. Identify target
3. Compute changes
4. Actuate robot
5. Update target data

Before object tracking can commence, it is necessary to analyse and extract sufficient information on the object. This information can then be used at a later time to identify the same target so analysis can be made to determine if its position and orientation has changed and if so, the robot can be actuated accordingly. The high level sequence of events is summarised in the flowchart in Figure 4.

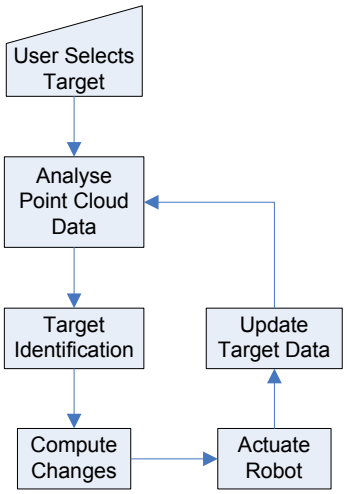

Figure 4: Flowchart summarising the high level events involved in object tracking and following

Section 3.1 describes the extraction of key pieces of information of the target from point cloud data, whilst Section 3.2 describes the PID controller that was used for target following.

3.1 Processing of Point Cloud Data

For this implementation, three key pieces of information are required of the target:

1. location in 3D space
2. size
3. orientation

Flowchart in Figure 5 summarises the algorithm used to extract these information. The remainder of this section describes these processes in detail.

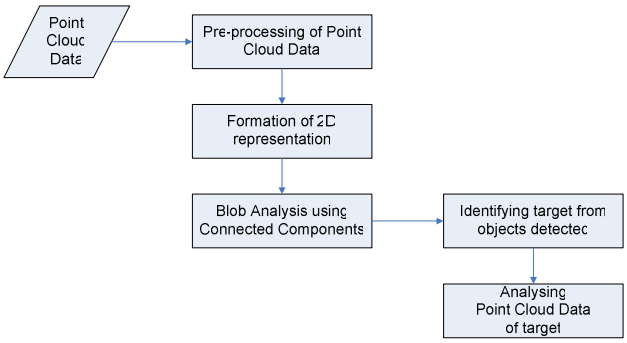

Figure 5: Summary of the processes involved in extracting information of the target from point cloud information from the stereo camera.

Pre-processing

Pre-processing prepares the point cloud data so that accurate analysis can be performed. 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 maximum conceivable distance between the stereo camera and the target in the real world. This has the effect of eliminating errors created by false correspondences, as well as reducing the amount of data for processing. Such values were likewise replaced with NANs.

Forming a 2D representation using 3D point cloud

For this work, the object is selected by the user through a graphical user interface (GUI). The average height of the object is sampled at the selection point using a 5x5 pixel window.

A binary representation is formed by applying a band pass filter to the point cloud data, where pixels which has z-values within this band is assigned a value of '1' and pixels with z-values outside the band, is assigned a value of '0'. Of course, this approach would only work if the contour of the object surface is within the band pass filter. So, it is assumed that some apriori knowledge of the object exists.

Once the binary image is achieved, connected component algorithm can then be applied to the image for blob analysis, described in more detail in the following section.

Blob analysis using Connected Component

At this point of the processing, 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.

Two-dimensional blob analysis is used to locate the centroid and orientation of the axes for each object detected. In addition, other information such as area and bounding box dimensions are computed as well. In addition, the algorithm also filters out any object that is smaller than a given size. The object with the centroid closest to the selection point is classified as the target object. It is important to note that blob analysis was performed on the 'flattened' point cloud data, which was

based on pixels that are within a certain height of interest.

Due to the imperfect nature of stereo imaging, it is unlikely to generate a perfect point cloud data. Hence, there will be 'holes' at various locations, where stereo matching fails. To assist the tracking algorithm, small holes within objects are filled before blob analysis begins.

The blob analysis algorithm was adapted from an image processing package, NI Vision. The following steps summarise the blob analysis procedure:

1. Small objects of area less than n-pixels are removed.
2. The size of the image is reduced to fit the internal object limits.
3. Small holes within objects are filled.
4. 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.
5. Objects are measured. The centroids are located and the slopes of the axes are calculated.

Analysing target position and orientation

Once information on the target object has been extracted and stored, tracking can proceed. As before, known erroneous data is removed and a band pass filter is applied to the point cloud data around the last known height of the tracked object. Blobs are formed and analysed.

After that, a second band pass filter is applied, based on the size of object tracked to eliminate objects that are not close to the tracked object in size.

Finally, the distance between each object and the last known location of the tracked object is calculated. It is assumed that the object that returns the smallest distance is the tracked object. In principle, this approach works well especially if the system runs at a high refresh rate. However, it assumes a fairly accurate result from the stereo imaging algorithm. Else, the target object will be incorrectly discarded. Once the tracked object has been identified, object following can proceed.

Target position and orientation

To physically follow the robot with 6DOF, the system has to determine if the tracked object has changed position and orientation in the x, y and z planes, so the manipulator can physically adjust itself to satisfy the aforementioned conditions.

Positional changes

Whilst it is possible to simply use the x, y and z values of the pixel that corresponds to the centroid of the object, in practice it is not a good approach as it may not contain a value due to failure in stereo matching. The approach used here is to calculate the average values within a 5x5 pixel window, centred on the computed centroid. This increases the probability that at least one pixel has been matched. In the event where no matches are found, the window size is extended continuously by a pixel until a valid value is found.

Tracking the pitch and roll

The pitch and roll of the object is not straightforward as it requires samples of two z-values. Using these z-values, the slant in both directions can be calculated using basic trigonometry. Ideally, the sample locations should be as far apart as possible, to reduce the affects of noise. However, if the sample area is too close to the edge, then there is a higher likelihood of being affected by an occluding object.

The approach taken is to calculate the average of the z-values within 5x5 pixel sampling windows adjacent to the 5x5 centroid window. For example, Figure 6 shows an object that is 21x25 cells in size, where each cell represents a pixel of information from the point cloud data with the black pixel representing the centroid of the object. To calculate the pitch, the average z-values are calculated for the two windows shaded in blue. Similarly, the average z-values are calculated for the two windows shaded in red for the roll calculations.

Two assumptions were made with this approach:

1. The object is not symmetrical around an axis.
2. The object has a relatively flat surface

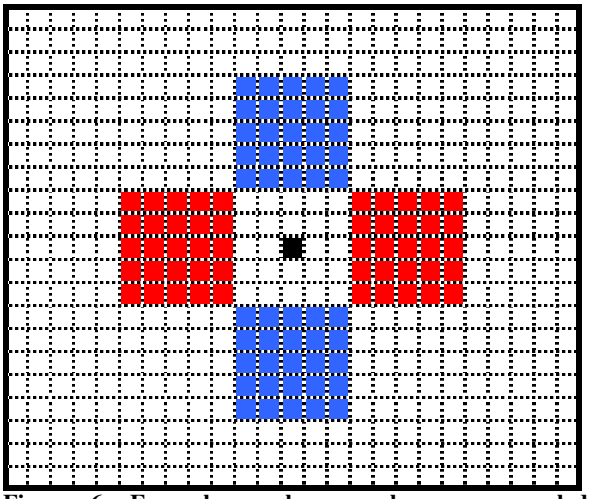

Figure 6: Example on how z-values are sampled to determine the pitch and roll of a target. The average z-value is calculated for the two 5x5 windows shaded in blue and red are then used to calculate the pitch and roll respectively using basic trigonometry. The cell in black represents the center pixel.

Tracking the yaw

The yaw of an object is essentially the orientation around the z-axis, or in this case, the object as viewed by the camera from above. The orientation of the object is calculated during blob analysis, as described in previous sebsection on tracking the pitch and roll.

3.2 PID Controller

The PID (proportional-integral-derivative) controller is a control loop feedback mechanism that is commonly used for process control applications. Since this is a classical control theory and no modifications were made for this project, the PID control will not be described in detail in this paper. For more information, please refer to [Åström and Hägglund, 1995]. Suffice to say, the PID controller calculates the error (difference between measured variable and a desired set point) and adjusts the process inputs to minimise the error as quickly as possible. Figure 7 below shows the block diagram of a PID control loop.

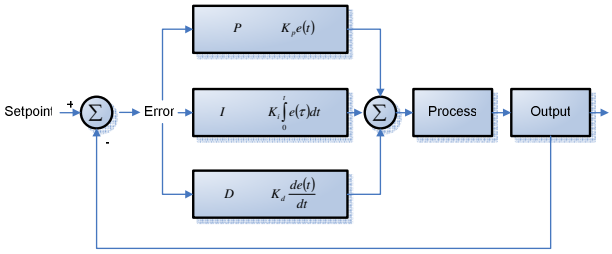

Figure 7: Block diagram of a PID controller.

As shown, the PID controller has three main terms. The proportional term can be used to reduce the rise time; the derivative term can be used to damp out any transient oscillations, and integral term can be used to remove any steady state error between input and output.

For this project, six PID controllers were implemented for three position axes and three orientation axes. The parameters were empirically determined by generating a step input into the controller and recording the response. For this application, the response was configured for a moderately quick rise time but low over-shoot. For the setup of this project, values of 0.2, 0.001 and 0.11 were respectively used for the PID parameters. A sample of the step response is shown in Figure 8.

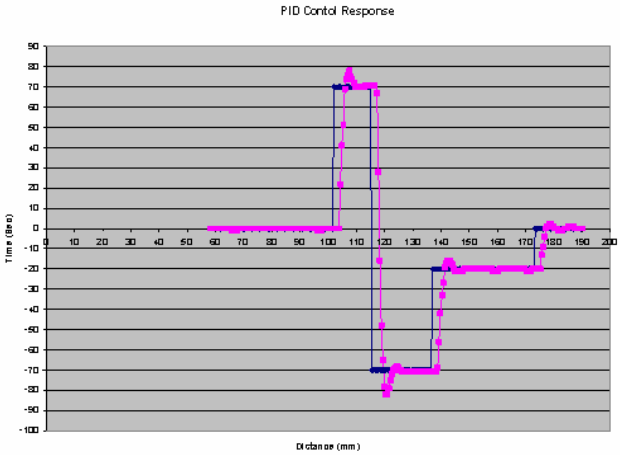

Figure 8: Recorded response of the PID controller to multiple step inputs

The blue line represents the input to the PID system and shows set point. The first change at time t = 102 seconds from zero to 70mm, the second at t = 114 seconds from 70mm to -70mm; and the third and fourth changes at t = 135 seconds and t = 174 seconds respectively. All these changes were manually set through a GUI.

The PID response (pink line) shows an overshoot of approximately 18%, which is acceptable and a rise time of about 3 seconds, which is quick enough for our requirement. However, it would be ideal to have a quicker rise time and lower overshoot.

4 Experimental Results

4.1 Object tracking and following in 6DOF

The algorithms described in this paper were tested to perform target tracking and following in the presence of both static and dynamic identical targets. The objective of this exercise was to keep the tool centre point of the manipulator robot (the left lens of the stereo camera) to be:

1. centred to the object's centroid
2. kept at a set distance away
3. orientated to the major axis of the object
4. perpendicular to the surface of the object around the centroid

For the experiment shown in Figure 9, nine objects were placed on a table in front of the manipulator. The objects were essentially printed pattern on sheets of plain paper. Some were rectangular and others were square in shape, with widths and lengths varying between 100mm to 120mm. An exampe of the pattern used is shown in Figure 10. Please note that this is not to scale.

Patterns were used to improve the performance of the stereo matching algorithm. However, it is important to note that the objects can be exactly identical in shape and size. The system is robust enough to cope with this situation. For this experiment, the objects consist of:

- Three static objects placed on the surface of the table with varying orientations.
- Two static objects elevated from the table surface at different heights.
- Two objects mounted on a turn table, rotating at around one revolution every four seconds at an elevation of 150mm. The angular velocity was set arbitrarily and does not aid or affect the performance of the system.
- One object mounted on a linear actuator that completes one cycle of extend and retract every 3 seconds. The actuator is not parallel to the table and hence, the height of the object was constantly changing. Again, linear velocity was set arbitrarily and does not aid or affect the performance of the system.
- One target object mounted on thin Perspex, manually moved to test the tracking and following system.

Figure 9: Setup of the object tracking and following experiment. The manipulator robot tracks a user specified target amongst identical objects, that are both static and dynamic in nature.

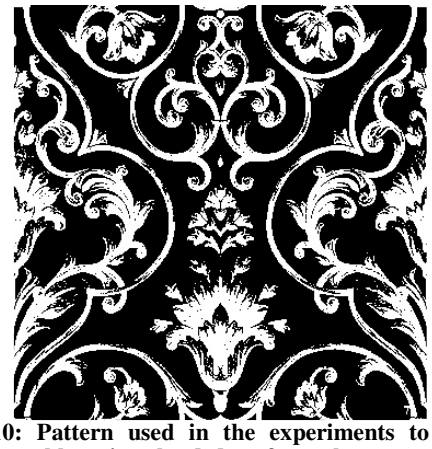

Figure 10: Pattern used in the experiments to assist in acquiring stable point cloud data from the stereo matching algorithm.

At the start of each experiment, the tracked target is selected via a mouse click on the object through the graphical user interface. Once selected, tracking is activated and instantly, the system calculates the positional and alignment difference of the object as described in Section 3. These differences were then sent to the PID controllers, which generate the appropriate commands to move the robot to minimise the errors.

Experiments validate the effectiveness of this approach. The tracking algorithm was able to identify the target and the manipulator able to follow it without problems, even when the target was partially occluded by an identical object, as shown in Figure 11.

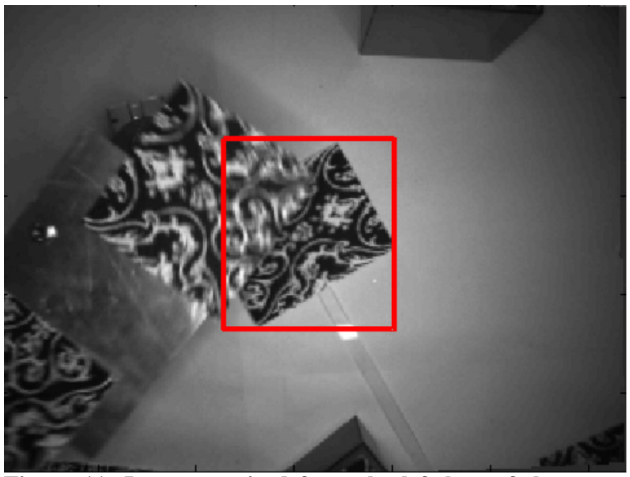

Figure 11: Image acquired from the left lens of the stereo camera. In this example, the target object is temporarily occluded underneath an identical object that is rotating on a turn table. Red border illustrates the identification of the target object. As can be seen, the partial occlusion has not severely affected the tracking algorithm.

At the time of the experiment, the system is limited to the following factors:

1. whilst the system continues to function when the object is partially occluded, orientation tracking would be affected as the shape of the tracked object has changed
2. if the target object has a low profile, then the table has to be untextured. Else, the target cannot be differentiated from the background. However, if the object has a certain height, then the table does not need to be untextured.
3. the top surface of the target object has to be a flat
4. the algorithms have not been streamlined and hence, the movement of the target is limited to a maximum speed of 10cm/s.

It is also important to reiterate that this system relies upon consistent data from the stereo matching module. Inconsistent data will result in an unstable robot that never reaches steady state.

4.2 Contour Following

The second experiment is a slight variant to the track-and-follow exercise. Instead of following a target object as it moves through space, this experiment attempts to keep the tool centre point at a fixed distance, whilst remaining orthogonal to the surface of a static object, as the robot moves between two user-specified points at constant velocity.

The physical setup of the system is similar to the object tracking exercise. Here, an object is placed on a table, underneath the extended arm of the manipulator robot and in the view of the stereo camera. The object used for this exercise was a model car. As before, printed pattern is used to assist the stereo matching computation, which was stuck onto the surface of the model.

The procedure begins with the user selecting the starting and ending locations on the object from the GUI through mouse clicks. Once selected and activated, the robot moves itself so that a) the centre pixel of the camera is directly above the start point and b) the camera is aligned to the contour following path i.e. the path connecting the starting and ending points.

Contour following then begins with the algorithm constantly following the height of the centre pixel, the slant along the x-axis and the slant along the y-axis using the same algorithms are described for target tracking and is decribed in Section 3.1. As with the target tracking experiment, changes in these parameters are compensated by the PID controller. Figure 12 is a screenshot of the experiment.

The system was able to follow the contour well and as with target tracking, the experiment shows is highly dependant on the data from the stereo camera.

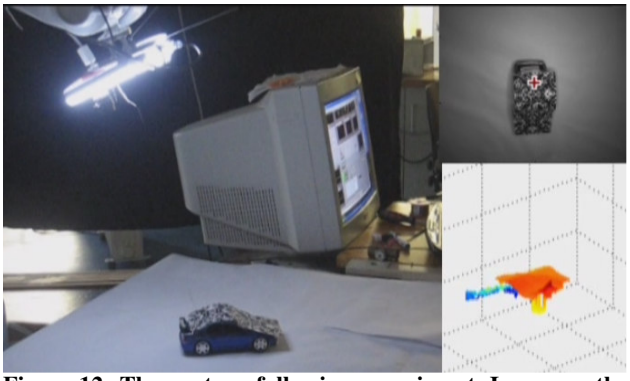

Figure 12: The contour following experiment. Image on the left depicts the experimental setup. Top right shows the view of the left lens of the stereo camera, with the red crosshair representing the center of the image. This is also the location where the robot has to remain at a set distance and be orthogonal to the surface of the object. Bottom right represents the point cloud information.

5 Conclusions and Future Work

We have presented an approach that utilises 3D data for object tracking and contour following. Using point cloud information from a stereo camera, a combination of 2D image processing and depth analysis were used to detect objects in the viewable space. Information about the location and orientation of the object is extracted so that object tracking and contour following can be performed using a manipulator robot and PID controllers. Both target tracking and contour following experiments successfully demonstrated this implementation. However, it also highlighted its dependancy on consistent data from the stereo camera.

A future work is to look at how projected texture can be used to replace the printed patterns used in this project. This involves inestigation into the technology for projection, as well as patterns that assist in stereo imaging. We have briefly experimented into the former by using a mini projector mounted onto the mounting plate, adjacent to the stereo camera. However, we faced issues with exposure, focus and throw distance; all factors that limit the feasibility of this approach. Whilst we have not performed extensive reseach into pattern optimization, perhaps lesson can be learnt from a piece of work that was published recently by Konolige [2010].

References

[Åström and Hägglund, 1995] Karl J. Åström , Tore Hägglund. *PID Controllers: Theory, Design, and Tuning*. ISA, USA, 1995.

[Baeten et al., 2007] Johan Baeten, Kevin Donne, Sven Boedrij, Wim Beckers and Eric Claesen. Autonomous fruit picking machine: A robotic apple harvester, 6th International Conference on Field and Service Robotics, pages 531-539, Chamonix, France, 2007.

[Biegelbauer and Vincze, 2006] Georg Biegelbauer and Markus Vincze. 3D Vision-guided bore inspection system, 4th IEEE International Conference on Computer Vision Systems, New York, 2006.

[Collewet and Marchand, 2008] Christophe Collewet and Eric Marchand. Modelling complex luminance variations for target tracking. IEEE Conference on Computer Vision and Pattern Recognition, pages 1 – 7, Anchorage, AK, June 2008.

[Frost et al., 2004] A.R. Frost, A.P. French, R.D. Tillett, T.P., Pridmore, S.K. Welch. A vision guided robot for tracking a live, loosely constrained pig, Computer and Electronics in Agriculture, 44(2):93 – 106, August 2004.

[Konolige, 2010] K. Konolige. Projected texture stereo, 2010 IEEE International Conference on Robotics and Automation (ICRA), pages 148 – 155, Anchorage, AK, May 2010.

[Lim et al., 2002] Patrick Lim, Jeffery Yang, Nicholas Hildreth and Werner Friedrich. Application of Shared Telerobotic Control for Contour Following Processes, Australasian Conference of Robotics and Automation 2002.

[Lim et al., 2010] Patrick Lim, Jason van Beurden, Phil Stucki, Christopher Lennox and Geoff Bates. Automatic Gas injection for reduced strain depelting of ovine carcass, 25th CAD/CAM, Robotics and Factories of the future (CARsFOF), South Africa, 2010.

[Panin and Knoll, 2006] Georgio Panin and Alois Knoll. Fully automatic real-time 3D object tracking using active contour and appearance models, Journal of multimedia, 1(7):62-70, November 2006.

[Prabuwono et al., 2008] Anton S. Prabuwono, M.A. Burhanuddin and Samsi Md. Said. Autonomous contour tracking using staircase method for industrial robot. 10th International Conference on Control, Automation, Robotics and Vision, pages 2272 – 2276, Hanoi, Vietnam, December 2008.

[Prabuwono et al., 2010] Anton S. Prabuwono, Samsi Md. Said, M.A. Burhanuddin and Riza Sulaiman. Performance evaluation of autonomous contour following algorithms for industrial robot, Robot Manipulators Trends and Development, pages 377 – 398, March 2010.

[Silveira and Malis, 2007] Geraldo Silveira and Ezio Malis. Real-time visual tracking under arbitrary illumination changes. IEEE Conference on Computer Vision and Pattern Recognition, pages 1 – 6, Minneapolis, USA, June 2007.

[Wong et al., 2009] Chee Kit Wong, Patrick Lim, Roger Clist and Raymond Liu. VGR strategies for robotic bin picking of natural products, Australasian Conference of Robotics and Automation, Sydney, Australia, December 2009.