The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Determining an Object's Shape with a Blind Tactile Manipulator

David Devereux
School of Computer Science
The University of Manchester
Manchester, M13 9PL, UK
devereud@cs.man.ac.uk

Paul Nutter
School of Computer Science
The University of Manchester
Manchester, M13 9PL, UK
p.nutter@manchester.ac.uk

Robert Richardson
School of Mechanical Engineering
The University of Leeds
Leeds, LS2 9JT, UK
R.C.Richardson@Leeds.ac.uk

Abstract— **In situations where generating in internal representation of an object's shape can not be carried out via visual methods, such as in low light conditions or due to a hardware fault, an internal representation of the object must be built from tactile information alone. Existing methods for doing so rely on the availability of a complete force sensitive skin. The work described in this paper shows a method that requires no such equipment and in fact requires no external sensors at all. The method demonstrates that an accurate representation of the object can be built with the accuracy depending on the number of attempts at contacting the object. This internal representation of object can then be used for other tasks such as planning grasps.**

I. INTRODUCTION

Whole arm grasping is an extremely useful ability that we as humans use near constantly to interact with our environment. There are multiple stages to the grasping process [1]:

1. Object recognition
2. Grasp planning
3. Initial touch and grab phase
4. Stable grasp phase

The work described in this paper focuses on the first stage, object recognition. Humans and indeed animals use a combination of tactile and visual information to decide the best manner in which to grasp an object. There are various methods of analysing visual data in order to extract an object's shape. Chinellato et al. [2] have a biologically inspired approach, they generate, from an initially supplied rough 3D estimation of the object's shape, a more refined estimation using an octree based reconstruction algorithm. Hoppe et al. [3] can generate a very good estimation of an objects shape from a set of unorganised point data. The point data can be supplied from a variety of sources such as laser range finding or feature extraction. Local groups of the unorganised points are used to generate indications of the tangent and normals at those positions. This is repeated over the entire surface of the object. From this data a manifold of the object's shape is produced. The level of complexity of both these examples is quite large. We shall see that the proposed algorithm in this paper requires much less complexity both in algorithm design and data analysis.

So there exists quite competent methods that can determine an objects shape from visual data but what happens if the robot is put into a situation where it can no longer see? Whether because of a fault (e.g. broken camera) or because of external factors (dense smoke, lack of light). In these situations reliance on a purely tactile input is necessary.

Existing approaches that use tactile information to provide information regarding object shape use the assumption that a force sensitive skin covers the entire surface of the manipulator [4] [5]. This skin provides information such as the location of any contacts and the force between the manipulator and object. This sensing capability would certainly be useful but unfeasible with current technology [6] and adds an extra layer of complexity that can be demonstrated to not be required.

The work shown in this paper demonstrates that an object's shape can be determined for grasping and other purposes from manipulator position information alone. More importantly it will do so without the need for sensors that measure quantities external to the robotic manipulator. This means that the only information available to the controller is the current angles of the manipulator joints and the size of the manipulator itself. It is assumed that the object to identify is within reach of the manipulator, that the manipulator has sufficient length to wrap around the majority of the object and that the object is rigid and incompressible.

This paper will now discuss how to calculate the object's shape given that the positions of the manipulator's links when contact has been made are known. Discussion will then progress on to how to get such link-contact position information. The methods will then be applied to objects and the resulting shapes compared to their originals.

II. EXTRACTING AN OBJECT'S SHAPE

Given that you have a method that can determine the position of the manipulators links when contact has occurred between the object and manipulator you can see that the object's shape is implied from the absence of link positions such as in Fig. 1a. The absence in this case implies the presence of a square. This section describes how to extract the object's shape implied by this absence of link positions.

$$P_i = \begin{bmatrix} x \\ y \end{bmatrix} \tag{1}$$

$$P_i = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2}$$

$$r_i = \sqrt{P_i.P_i} \tag{3}$$

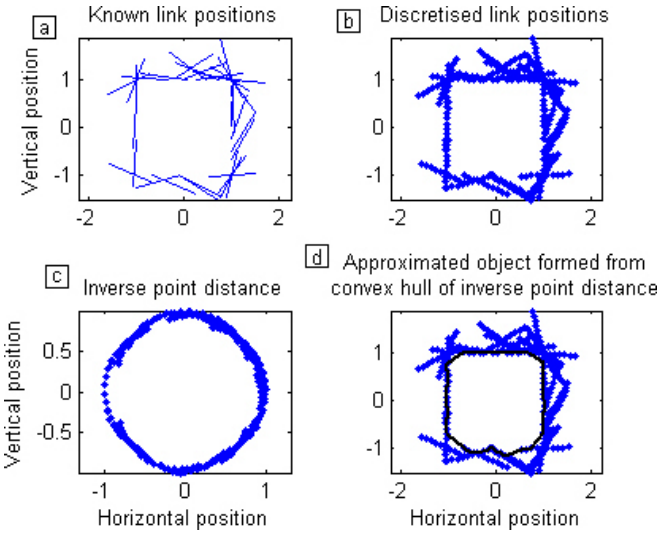

Fig. 1. Steps taken to generate the object's shape

$$d_i(a) = r_i^{-a} \tag{4}$$

$$P_i^{inv} = d_i(a)P_i \tag{5}$$

The procedure to extract the objects shape from this absence of link positions uses a function called the inverse convex hull [7]. The inverse convex hull operates on discrete points so the link positions are first discretised as in Fig. 1b. The operation can operate on both 2D and 3D data, points in 2D space being represented as in (1) and points in 3D space being represented as in (2). A set of points are then calculated from these points that have their distance inverted using (5), where $d_i(a)$ is defined in (4). The 'a' parameter in (4) defines the coarseness of the generated object. r_i is the distance of point i from an origin point inside the object and is calculated using (3), where '.' is the inner or dot product of a vector. Values of 'a' from between 0.05 and 0.5 have been tried but throughout this paper a value of 0.2 has been settled on. Analysis of the performance of the algorithm under varying values of 'a' is still required and is currently left as future work (see section V-B). Fig. 1c shows the set of inverted points P_i^{inv} for the points shown in Fig. 1b, the inversion means that the points that were originally closest to the centre are now the furthest away. This allows for the use of the convex hull algorithm to find the outer most points of P_i^{inv} and in turn those corresponding innermost points in P_i that best represent the object's shape. This is shown for the running example as the solid inner line in Fig. 1d. The points from Fig. 1b have also been plotted to show the points chosen by the inverse convex hull.

The points P_i need to be defined relative to some internal position of the object. This internal position can be found by finding the mean of the discrete link positions. Such a method works if the links are applied evenly over the surface of the object. An alternative method is to find the min-max extremes of the discrete points and select the mid-point of them.

The object is represented as a polygon rather than using some more complicated method. This means that curved surfaces can only be approximated but the level of information that tactile sensors can provide is coarse relative to visual data and as such although a more complicated representation could be used it would not provide any extra accuracy.

An alternative method to the inverse convex hull that may initially be considered but quickly discounted is to discretise the contact links and simply take the points that are closest to the centre. The first problem is that by simply taking the closest points the algorithm will select many points from the same link, it is more usually the case that only a small number of points should be chosen. The second problem is the choice of how many points to take, this value depends on the level of discretisation and the number of links applied. The third problem is concave elements in an object, the algorithm will greedily accept all the concave elements as they will tend to be closer than the rest of the object to the centre and will thus ignore the equally important extremes of the object. The last problem is that the chosen position of the centre point will greatly effect the resulting shape, shifting the centre more to one side will then cause the algorithm to favour points from that side.

A. *Error measure*

An area error measure is used in the test cases to measure the performance of the algorithm. If the area of intersection of the two polygons is calculated as A_{int}, the area of the original polygon is A_{orig} and the area of the calculated polygon is A_{gen}. Then the area error A_{err} is:

$$A_{err} = A_{gen} + A_{orig} - 2A_{int} \tag{6}$$

The intersection area is going to be smaller or the same size as either the original object and the generated object's area. The difference between the actual object's area and the intersection area shows by how much the generated object has entered inside the actual object. Likewise, the difference between the generated object's area and the intersection area shows how much the generated area has overestimated the size of the object.

A percentage error (Err in (7)) is used as a single glance review of performance, 0% being the best possible result.

$$Err = \frac{A_{err}}{A_{orig}} \tag{7}$$

III. ENCIRCLING AN OBJECT

The previous section assumed that the contact positions of the links of a manipulator are known for various attempts at contacting the object. In this section an algorithm is described that can provide such information. The method described here assumes that the object to identify is within reach of the manipulator and that the manipulator has sufficient length to wrap about the majority of the object.

The method for gaining contacts described in this section is based on the movements of an octopus's arm [8]. The octopus's arm has an amazing amount of flexibility but is only used in relatively simplistic movements [9].

To encircle the object in the planar case the manipulator must first be straightened out where it will not contact the object. Then starting with the joint nearest the base of the manipulator the joint moves towards object swinging the more distal joints in with it. Upon contact the link maintains its current position and the next most proximal link moves inwards. This process repeats until the object has been fully encircled. The straightening procedure ensures that the more distal links will not prematurely contact the object before the current link should.

Multiple attempts at this procedure should be accomplished but with the base joint terminating slightly before its previous stop angle each time. By doing this the position of the subsequent links when contacting the object will be shifted in position supplying more information. As well as multiple attempts at contacting the object with a different angle each time, it is a good idea to grasp from different directions as well. This is demonstrated more thoroughly in the next section.

Contacts are detected here through the use of an open loop reference controller although alternative methods could also be used. The open loop reference controller states that there is an expected amount of angular position error for each joint during normal operation. If the error breaches this expected amount then it must be due to a contact.

Each attempt at exploring the object will result in a set of angles corresponding to the joint angles of the arm upon final contact. Through the use of forward kinematics the angles of the joints can be converted into positions in the Cartesian domain. These link positions then need to be shifted by an amount corresponding to the width of the manipulator resulting in two lines for each link of the manipulator. The outside line will rarely be chosen by the inverse convex hull and only when exploring concave objects, therefore they have been left out of any diagrams.

The discussion so far has been about planar manipulators, the method can be applied to 3D situations as well. The manipulator will be applied to the object in the same manner as the 2D case but with a lot more search directions and iterations such that the entire circumference of the object will be systematically contacted. The coarseness of the resulting object will be determined by the amount of directions and iterations performed.

There are a number of matters that should be considered as they further complicate the situation. The first is friction and object mass, this is the friction between the object and the surface it is sitting on rather than the friction between the object and manipulator. The object is not fixed in place, therefore the friction and mass of the object must be sufficient to provide enough reaction force to the manipulator to stop its movement enough to cause contact detections but also not allow the object to move due to that contact force. Ideally the manipulator should have a low weight such that the contact forces are minimised.

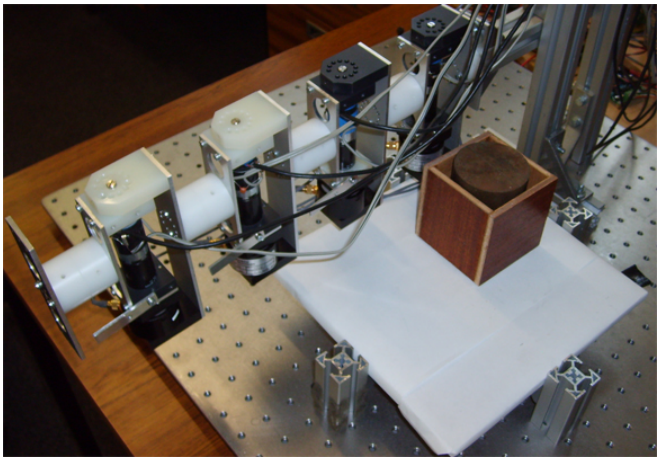

Fig. 2. The experimental setup, ready to encircle

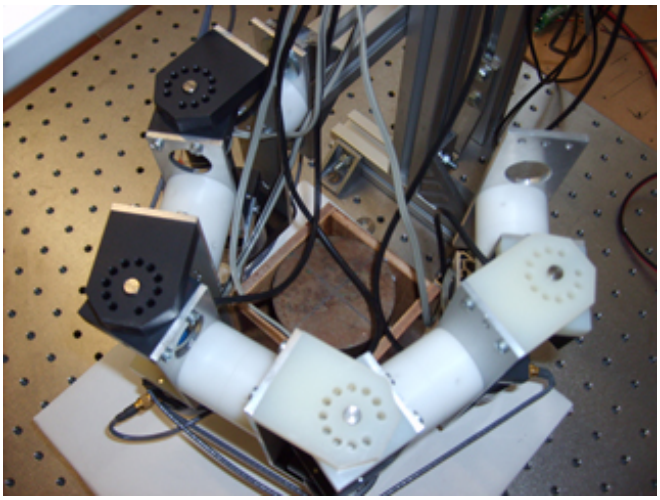

Fig. 3. An example of the manipulator that has encircled an object

The second consideration is concave elements in objects. If the manipulator's width is sufficiently small and the number of directions and attempts chosen are sufficiently large then concave elements should eventually be explored due to the shifting of the end effector caused by the intentional retarding of the initial position of the most proximal link. Of course if the manipulator's width is too large than there is no purpose or possibility of exploring the concave elements as they cannot be reached anyway. Exploration of concave objects has not been further considered here due to the nature of the experimental equipment but is left as future work.

A. *The experimental equipment*

Both of the algorithms described will now be used to calculate the shape of a couple of objects. The experimental hardware can be seen in Fig. 2 in its straight out configuration. An example finished encirclement is shown in Fig. 3. The manipulator consists of four modular links each powered by an electric DC motor coupled to a gearbox.

The actuators are powerful enough to counter gravity for

Fig. 4. The objects being identified

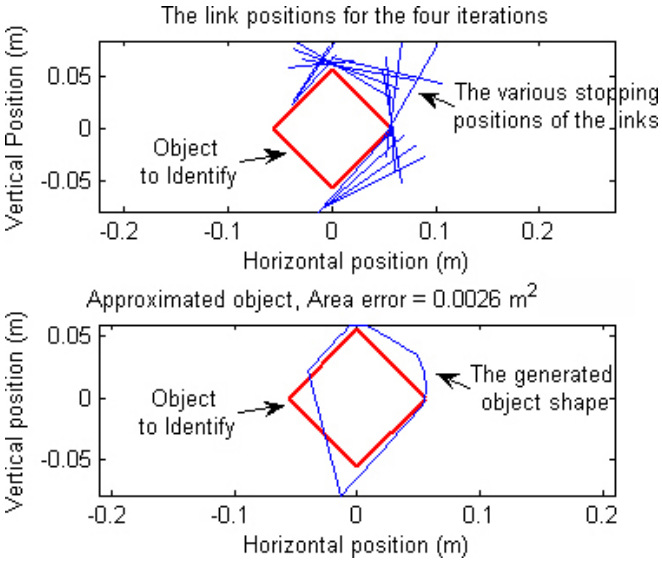

Fig. 5. 1) The link positions of the multiple attempts applied to a cube and 2) the generated object

their own particular module but the combined weight of the modules is too much therefore the robot is restricted to a planar case such that gravity plays no part in the process.

IV. EXAMPLES OF OBJECT GENERATION

Fig. 4 shows the two objects that are to be identified. The box shaped object had to be weighed down as its mass was only 80g which, when compared with the combined mass of the manipulator (4Kgs), is insignificant. Without being weighed down the frictional force would cause a position error that is less significant than the normal operating error.

The length of each module of the motor is 110mm whereas the length of each side of the cube is only 80mm, whilst the radius of the cylinder is 63mm. This shows that the link length is relatively large when compared to the objects and this may cause the generated object to be rather coarse.

Fig. 5 and Fig. 6 show the results of both the algorithms when one direction of motion and position is attempted with four different angles for the proximal link. The cylinder has been approximated by a 100 sided regular polygon during the analysis stage. In all the diagrams the positions of the generated and actual objects are represented in the object's reference frame to allow for an easy visual comparison. Also, bear in mind that the link positions shown in the figures are not from the centre of rotation of one joint to the subsequent one. They are the outer position of that particular link of the robot. The links for this particular robot also extend past the centre of rotation of the next link by a small amount resulting in the small amount of extension viewable in the figures. Both these items account for the links not matching start and end positions.

Ideally in order to demonstrate the benefits of grasping from a different direction the grasp would be repeated but from a different start position and direction. Unfortunately due to the configuration of the robot this is not possible but the same effects can be found by mirroring the results gained from the first example as demonstrated in Fig. 7. This is a valid operation to do in this case due to the symmetry of the object although normally this would not be possible.

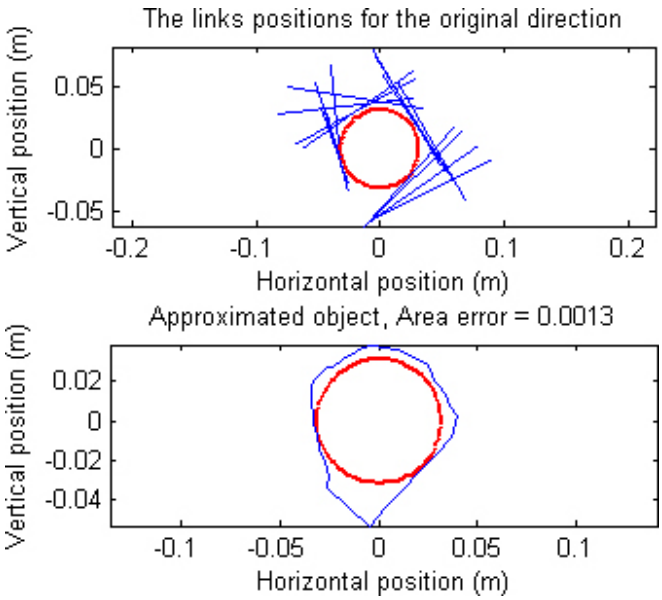

Fig. 6. 1) The link positions of the multiple attempts applied to a cylinder and 2) the generated object

Normally the manipulator should be moved around the object by a known amount.

Fig. 8 and Fig. 9 show the results for the situation when two directions are used. As you can see there is quite an improvement. If a third direction was attempted at a position 90^o around the object then the start position would also be contacted, which is currently the only part that is largely wrong.

To demonstrate how important having lots of small sized links are to these algorithms a simulation was carried out to contrast with the experimental situation. The experimental situation had 4 links of length 0.08m repeated 8 times totaling 32 link positions. The simulation instead has 320

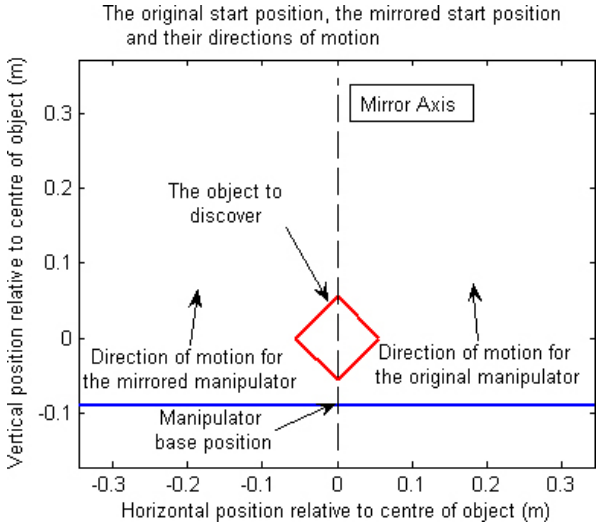

Fig. 7. Mirroring the example motion to simulate grasping from both sides

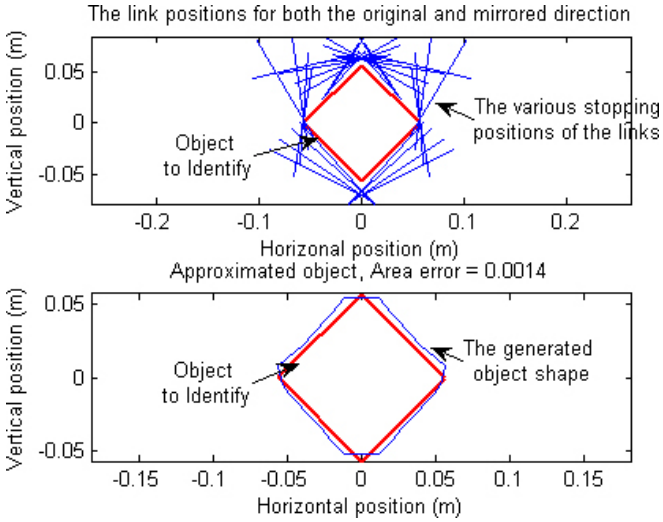

Fig. 8. 1) The link positions of the multiple attempts, mirrored attempts applied to a cube and 2) the generated object

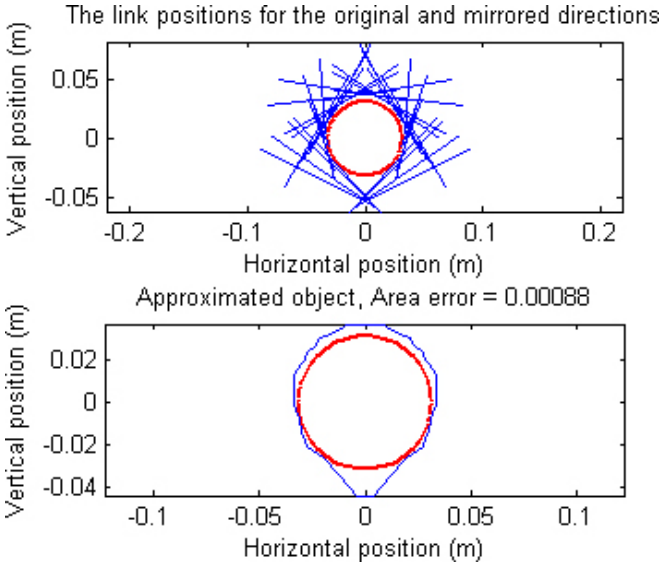

Fig. 9. 1) The link positions of the multiple attempts, mirrored attempts applied to a cylinder and 2) the generated object

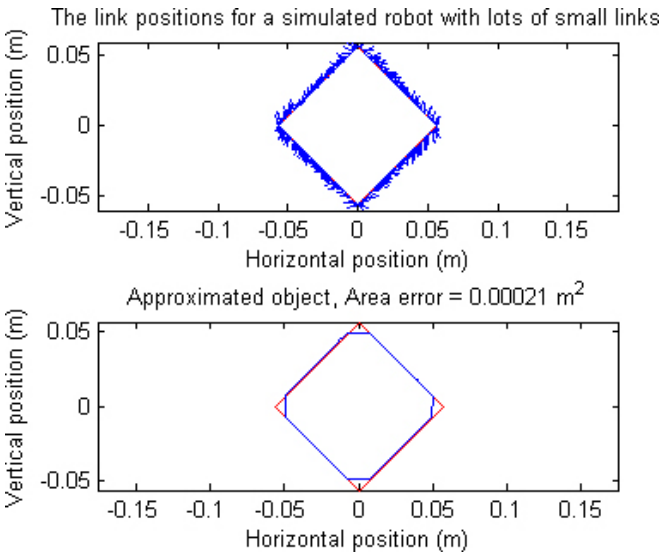

Fig. 10. 1) The link positions of a simulated attempt at encircling and 2) the generated object

link positions of length 0.008m. The results are shown in Fig. 10. The simulation is limited in that the positions of the links are randomised rather than connected to each other but the process does clearly show the benefits of increasing the number of links.

The area of the square is $0.0064m^2$ whilst the area of the circle is $0.0031m^2$. Table I summarises the various area errors for the five different cases. Each additional direction vastly improves upon the accuracy of the object and the simulation with many small links performs very well with only the corners being in error. This is caused by the value of 'a' during the convex hull algorithm. The higher the value, the coarser the resulting object, the lower the value the closer the object resembles the inside shape of the links. In these results a value of 0.2 has been chosen for all cases, a lower value improves the error for the simulated case but increases the error for the experimental results and vica versa.

Each encirclement took roughly 100 seconds, this was due to the high mass and kinematic coupling. In order to compensate for these effects the movement must be slowed down to allow the open loop reference controller to detect more delicate collisions. If the manipulator was improved by decreasing the weight and by supplying a velocity reference to aid in position control then there is no reason why this time can not be vastly reduced

V. CONCLUSIONS AND FUTURE WORK

A. *Conclusions*

It has been demonstrated that by using the two methods described above it is possible to calculate the shape of an object from the position of the manipulator during multiple encirclements alone. It has also been shown that increasing the number of directions and positions of movement during

TABLE I
SUMMARY OF RESULTS

Object and style	Area error (m^2)	Percentage error
Cube one direction	0.0026	40.6%
Cube two directions	0.0014	21.9%
Simulated Cube	0.00021	3.3%
Cylinder one direction	0.0013	41.9%
Cylinder two directions	0.00088	28.4 %

the encirclement algorithm greatly improves the accuracy of the representation of the object. It has also been demonstrated that it is far better to have a large number of small links than a small number of large links.

The algorithms worked well for experimental setup considering the relatively large link lengths of the robot. The control algorithms also had to deal with high friction in the joints and non-linearities such as cogging and backlash in the gearbox and kinematic coupling. If a manipulator was designed for grasping purposes and overcame these short comings then the speed of attaining the contact data and accuracy of the resulting object could be greatly improved.

A source of error in the experimental situations was caused by the relaxation of the joints once contact had occurred for a link. Once contact has occurred the link reverts to maintaining that position, unfortunately this means there is now no position error and thus no torque directed towards the object and the manipulator can move slightly away from the object. If a small force was still applied directed into the object, not enough to move but enough to maintain contact, then that relaxation would not occur and the results should be improved.

As already mention most of the errors in the resulting shape were due to poor manipulator design rather than poor algorithm performance, although even with the few large links the method works well. The only source of errors due to the inverse convex hull procedure are at the corners and this could perhaps be improved upon by selecting a smaller value of 'a' (see next section). Errors in shape generation are not such a big problem for whole arm grasping purposes as the control algorithms could take this into account. When attempting to contact at a planned contact location that in reality is too far out from the shape, then the manipulator would find a lack of contact and could then update the internal representation of object, re-plan the grasp and carry on.

It is recommended to attempt grasps from as many different positions and directions as possible in order that the entire surface be contacted. In the examples shown above with only two attempts the start position is not covered. If a third start position was chosen that was rotated through 90^o then the start position would have been contacted and the resulting shape made more accurate. As such it is recommended to use three different positions and directions for the planar case.

In summary the algorithms described work well even for a robot that has numerous design problems. The methods described have advantages over existing methods as it requires neither sophisticated external sensors such as cameras or force sensitive skin nor complex computational activities to extract the objects shape. The methods do require a certain amount of patience in order to get a detailed picture of the object but it does show that a blind robot can see objects.

B. Future Work

Apart from improvements to the control algorithms and manipulator design there are several extra areas of exploration worth pursuing. It would be useful to explore the parameter space of the inverse convex hull in an attempt to find the optimum values of 'a', and level of discretisation.

Further work that must definitely take place is the exploration of more complicated objects that include concave object. This may require a redesign of the manipulator such that it has more and smaller links.

A possible improvement to the encirclement algorithm is also proposed here. Instead of holding the links of the arm straight out and sweeping them around looking for contact. The arm should instead begin by doing this but as soon as the first contact occurs the distal most links from the point of contact except for a few should curl back on themselves following the path of the proximal links. As contacts are found the arm should unravel itself around the object. Given a smaller manipulator width this method may then be capable of exploring most object types including concave objects.

Lastly, an interesting area for further future work would be to develop a force sensitive skin for use in conjunction with this work and compare the results against the methods that assume the existence of such a sensor.

REFERENCES

[1] C. Bard, J. Troccaz, and G. Vercelli, "Shape analysis and hand preshaping for grasping," in *IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS) '91*, pp. 64–69 vol.1, 1991.

[2] E. Chinellato, G. Recatala, A. P. del Pobil, Y. Mezouar, and P. Martinet, "3d grasp synthesis based on object exploration," in *IEEE International Conference on Robotics and Biomimetics*, vol. 0, pp. 1065–1070, 2006.

[3] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *ACM SIGGRAPH Computer Graphics*, vol. 26, pp. 71–78, 1992.

[4] D. Reznik and V. Lumelsky, "Multi-finger "hugging": a robust approach to sensor-based grasp planning," in *IEEE Int'l Conf. on Robotics and Automation*, pp. 754–759 vol.1, 1994.

[5] F. Asano, Z.-W. Luo, M. Yamakita, and S. Hosoe, "Dynamic modeling and control for whole body manipulation," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 4, pp. 3162–3167 vol.3, 2003.

[6] M. Moll and M. Erdmann, "Reconstructing shape from motion using tactile sensors," in *Intelligent Robots and Systems, 2001. IROS 2001. IEEE/RSJ International Conference on*, vol. 2, pp. 692–700, 2001.

[7] F. Snyder, D. Morris, P. Haley, R. Collins, and A. Okerholm, "Autonomous river navigation," in *Proceedings of SPIE: Mobile Robots XVII* (D. Gage, ed.), vol. 5609, pp. 221–232, 2004.

[8] I. D. Walker, D. M. Dawson, T. Flash, F. W. Grasso, R. T. Hanlon, B. Hochner, W. M. Kier, C. C. Pagano, C. D. Rahn, and Q. M. Zhang, "Continuum robot arms inspired by cephalopods," in *SPIE Unmanned Ground Vehicle Technology VII*, vol. 5804, pp. 303–314, May 2005.

[9] Y. Y. Yoram, S. G. German, F. T. Tamar, and H. B. Binyamin, "How to move with no rigid skeleton? the octopus has the answers.," *Biologist*, vol. 49, no. 6, pp. 250–4, 2002.