A Best Next View Selection Algorithm incorporating a Quality Criterion

Nikolaos A. Massios† Robert B. Fisher‡

†Intelligent Autonomous Systems, University of Amsterdam,
1098 SJ, Amsterdam, The Netherlands
massios@wins.uva.nl

‡Dept. Artificial Intelligence, University of Edinburgh,
Edinburgh EH1 2QL Scotland
rbf@dai.ed.ac.uk

Abstract

This paper presents a method for solving the Best Next View problem. This problem arises while gathering range data for the purpose of building 3D models of objects. The novelty of our solution is the introduction of a quality criterion in addition to the visibility criterion used by previous researchers. This quality criterion aims at obtaining views that improve the overall range data quality of the imaged surfaces. Results demonstrate that this method selects views which generate reasonable volumetric models for convex, concave and curved objects.

1 Introduction

When building a complete 3D model of an object, it is necessary to obtain views of the object from several directions, so that data for all surfaces of the object is acquired. Since obtaining a range scanner view is a time-consuming process, a method of automating the data acquisition process would be desirable. The Best Next View problem is that of selecting the next view for the imaging system to take, given some already acquired views of the object. Solving the Best Next View problem is a step towards automating the range data acquisition process.

Various Best Next View algorithms have been proposed in the literature, [1, 5, 6, 7, 10]. According to Tarabanis [8], the approaches used can be divided in the *generate and test* and *synthesis* paradigms. In the generate and test paradigm [1, 5, 6, 10] potential (generated) Best Next Views are evaluated (tested) using some sort of optimality function to decide which view is the best. In the synthesis paradigm [1, 7] constraints on the viewpoint location are used to define volumes of possible view placements. Our approach [4] follows the generate and test paradigm. Its view evaluation function is a combination of two view evaluation criteria. The first criterion, called the visibility criterion, aims at maximising the amount of unseen surface seen by the next view. This is the main criterion used previously. Unfortunately, this can lead to surfaces that are seen at very oblique angles, which results in low quality range measurements. The new quality criterion aims at

BMVC 1998 doi:10.5244/C.12.78

Value	Comment
Surface Normal	Estimated using Least Squares Fitting
Local Quality	Estimate of the quality of the measurement
Region Quality	Estimate of the quality in the region

Table 1: Values associated with Seen voxels.

improving the quality of the data. Use of this criterion is new, and is the main contribution of this research.

In this paper, we begin by describing our approach for selecting the Best Next Views, then we give a brief account of our implementation. Results obtained using our implementation on a variety of objects are presented which suggest that the combination of the visibility and quality criteria provides us with data that covers the objects and is of good quality.

2 Best Next View selection approach

2.1 Volumetric representation

At each cycle of the best next view computation, we use the range data from all previous views to set up a volumetric representation of the scanned object. This volumetric representation is called a *voxelmap*. This is a 3D structure made out of voxels which essentially mark whether their area of space is part of the object or not. Each of these voxels is an identical cube and the collection of these cubes (the voxelmap) is a parallelepiped.

2.1.1 Possible voxel values

The voxel marking scheme is similar, but not identical, to the ones used in [1] and [7]. Each voxel has one of the following values:

Empty. The voxel's area of space is empty.

Seen. The voxel's area of space contains a range measurement (i.e. it corresponds to a seen surface). Each Seen voxel also records several values related to the *quality* of the range measurement used to mark the voxel (see Table 1).

Unseen. The voxel's area of space has not been seen yet.

OcclusionPlane. The voxel's area of space lies on an occlusion plane (see Fig 1). This is essentially an unseen voxel that neighbours with one or more empty voxels.

2.1.2 Voxelmap marking

This voxelmap is constructed following the procedure described by Wren [9]. Some extensions have been made to make this procedure capable of determining the accuracy of each Seen measurement as well as finding the Unseen and OcclusionPlane voxels. Given some already acquired views the algorithm is:

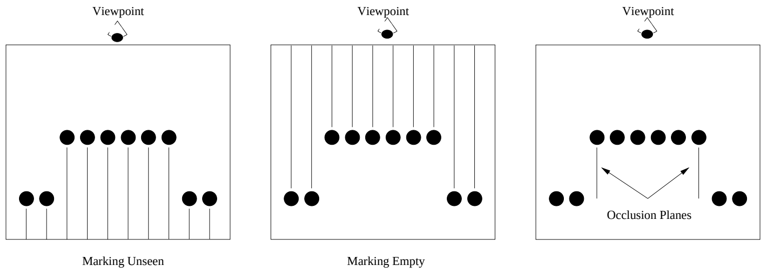

Figure 1: The two different voxel marking ray directions and the occlusion planes.

1. *Determine the size and resolution of the voxelmap.* This looks at the range data, of the already acquired views, to determine its extent and resolution. This information is then used to find how many voxels are needed in each of the 3 dimensions.
2. *Create and Initialise voxelmap.* Allocates the 3-D voxelmap memory structure and also marks all the voxels as Empty.
3. *Mark Unseen voxels.* Shoots a ray from each range point (x_p, y_p, z_p) in the direction pointing away from the viewpoint and marks all the voxels encountered on the way as Unseen (Fig. 1). This step is repeated for each already acquired range image.
4. *Mark Empty voxels.* Shoots a ray from each range point (x_p, y_p, z_p) in the direction pointing towards the viewpoint and marks all the voxels encountered on the way as Empty (Fig. 1)[1] This step is repeated for each already acquired range image.
5. *Mark Seen voxels.* Marks all the voxels that contain range measurements as Seen. Associated with each Seen voxel is a value that relates to how good the measurement used to mark that voxel is. This value can be determined by looking at $\hat{n} \cdot \hat{v}$ where $\hat{n}$ is the local surface normal of the range data point and $\hat{v}$ is the viewing direction (see next section). This step is repeated for each already acquired range image.
6. *Mark OcclusionPlane voxels.* OcclusionPlane voxels are found by checking each Unseen voxel to determine if any of its neighbours are Empty. If such an Unseen voxel is found it is marked OcclusionPlane (Fig. 1). OcclusionPlane voxels would end up being either Empty or Seen if a view that attempts to image them is taken. Therefore, eliminating OcclusionPlane voxels by viewing them also tends to increase the amount of object surfaces imaged.

2.1.3 Quality marking of Seen voxels

The quality of Seen voxels (Table 1) is estimated using the following algorithm.

1. The normals of the current data set are calculated using a least square error fitting of a 3×3 local surface patch (or smaller at edges) in the neighbourhood of each range measurement.

[1] Note that in Fig. 1 the rays are parallel because the camera model of our range scanner is orthographic.

2. For each voxel s to be marked Seen:

   (a) The data set normal is averaged with any existing surface normal estimate.

   (b) The dot product of the normal estimate at voxel s and the viewing direction $\hat{v}$ is calculated. If the dot product is greater than the previous local quality estimate, the dot product becomes the new local quality estimate $quality(s)$.

3. After the local quality estimates are calculated, the region quality estimate of each voxel is found by picking the maximum quality estimate in a $5 \times 5 \times 5$ neighbourhood. If the voxel lies next to the edge of the voxelmap only the neighbours found are used. This step is necessary because, due to inaccuracies, it is possible for a surface to be a few Seen voxels thick. When this is the case the local quality estimates do not reflect the real quality of the surface data.

These steps are repeated for each view taken by the system so far. At the end of this process, the Region Quality measure should reflect the quality of the range measurements collected for that part of the surface.

2.2 Constrained viewpoints and projection

After the voxelmap is constructed, it is projected to points on a constrained tessellated sphere representing potential new viewpoints. The tessellated sphere can be generated using various methods. The method we preferred is that of taking an icosahedron and recursively subdividing it until the required resolution is achieved (see left of Fig. 2). The resolution can be controlled by selecting the number of recursive subdivision steps to be applied. This tessellation has the advantage of generating an even distribution of viewpoints.

After viewpoints on the tessellated sphere are generated those which cannot be achieved due to positioning constraints are removed and are effectively not used in the later stages of our process. For example, the centre of Fig. 2 depicts the viewpoint sphere for the icosahedron tessellation. On the right of the same figure the constrained viewpoint sphere for a constraint on the maximum and minimum elevation angle can be found.

The next step is to project the voxelmap to each point on this constrained tessellated sphere. For projection purposes the sphere is placed at the centre of possible rotation of the object. Each voxel is modeled for projection purposes as a 3-D cube (6-Quads) of different colour (see right of Fig. 5). Black is used for the OcclusionPlane voxels and different shades of grey are used for the Seen voxels. The shades of grey in Seen voxels correspond to the varying Region Quality Estimates. Since the Empty and Unseen voxels are not used in the view evaluation process they are not projected to increase performance. The projected voxelmaps are used in the viewpoint evaluation (see next section).

2.3 Viewpoint evaluation

We propose to evaluate the desirability of viewpoints using the weighted sum of the *visibility* and the *quality* criteria.

The visibility criterion maximises the amount of OcclusionPlane voxels that are visible from the new viewpoint. This can be implemented by defining a function $f_{visibility}(\hat{v})$

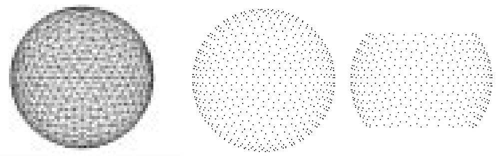

Figure 2: Tessellated sphere, viewpoints and constrained viewpoints.

of the viewing direction $\hat{v}$ such that:

$$f_{visibility}(\hat{v}) = sizeof(OP \cap V(\hat{v})) \tag{1}$$

where OP is the set of OcclusionPlane voxels in the voxelmap and $V(\hat{v})$ is the set of the visible voxels[2] from viewing direction $\hat{v}$.

The quality criterion maximises the amount of low quality voxels that are visible from the new viewpoint. This criterion can be defined using a function $f_{quality}(\hat{v})$ of the viewing direction $\hat{v}$ such that:

$$f_{quality}(\hat{v}) = \frac{\sum_{i=1}^{sizeof(S\cap V(\hat{v}))}(1 - quality(S[i]))(|\hat{v} \cdot normal(S[i])|)}{sizeof(S \cap V(\hat{v}))} \tag{2}$$

where S is the set of Seen voxels in the voxelmap, $V(\hat{v})$ is the set of the visible voxels from viewing direction $\hat{v}$, $quality(S[i])$ is the Region Quality Estimate and $normal(S[i])$ is the surface normal at $S[i]$. The value of $f_{quality}(\hat{v})$ is bound in the region [0,1]. The quality criterion prefers views that will image areas of low quality vertically (i.e. where $|\hat{v} \cdot normal(S[i])| = 1$).

The weighted sum $f_{total}(\hat{v})$ of the two criteria can be then found by:

$$f_{total}(\hat{v}) = w_v f_{visibility}(\hat{v}) + w_q f_{quality}(\hat{v}) \tag{3}$$

Both criteria should be applied to each viewing direction $\hat{v}$ in the constrained tessellated sphere for the partially acquired voxel representation. The direction for which f_{total} is maximal should then be selected to be our Best Next View.

To apply these criteria $\hat{v}$ becomes a point on the tessellated sphere. To determine $sizeof(OP \cap V(\hat{v}))$ the volumetric representation is projected to each $\hat{v}$ of the tessellated sphere and the number of unseen empty voxels visible is counted. The Seen voxels in this projection are also observed to determine $sizeof(S \cap V(\hat{v}))$ as well as $\sum_{i=1}^{sizeof(S\cap V(\hat{v}))}(1 - quality(S[i]))$.

2.4 Summary of our approach

To sum up, our approach uses a volumetric representation for representing the object. The voxels in the volumetric representation are marked as Empty, Seen, Unseen or OcclusionPlane. A tessellated sphere is fitted over the volumetric representation. Then constraints are applied on the tessellated sphere to account for scanner/object positioning limits. All points on the constrained tessellated sphere correspond to possible viewing directions.

[2]visible voxels = voxels that are visible after a projection in direction $\hat{v}$.

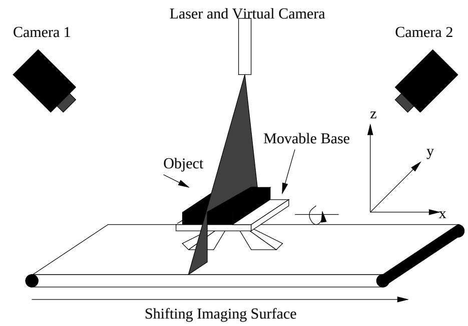

Figure 3: Diagram of the laser range scanner setup with one rotational and one translational degree of freedom, both aligned with the X-axis.

The volumetric representation is projected to every direction in the constrained tessellated sphere and the selection criteria are applied for each view direction.

The criterion application works as follows:

1. The criteria of Equation (3) are applied to determine the desirability of the each view.
2. The direction that has the highest aggregate score is selected.

After the Best Next View is selected, it is used to take the new range image of the object. The old and the new range images are used to rebuild the volumetric representation and the modified volumetric representation is used to select the new Best Next View.

3 Implementation

This approach was implemented in C/C++, using OpenGL to perform the projections. Our implementation used the range scanner available at the Machine Vision Unit to collect the views (Fig. 3). The laser scanner utilises two cameras to collect the images. Using optical lenses, a laser produces a planar beam that intersects the object being modeled thus creating a thin bright line on the surface of the object. The object is placed on a metal base with one rotational degree of freedom which in turn is placed on the imaging surface. This surfaces moves in discrete steps along a very accurate conveyor belt that is driven by a stepper motor. Points along the line of intersection of the laser plane and the object are used to collect range data by means of triangulation. Each camera collects a separate range image of the object. These two range images are then fused [2] into a more accurate range image. The Best Next Views planned correspond to rotations of the object positioning base. So, for the base currently being used, the constrained viewpoints lie on a circle rather than on a sphere.

4 Results

Fourteen parts were used to evaluate our Best Next View planning procedure (Fig. 4). The parts used varied in terms of material, shininess and curvature. The testing process

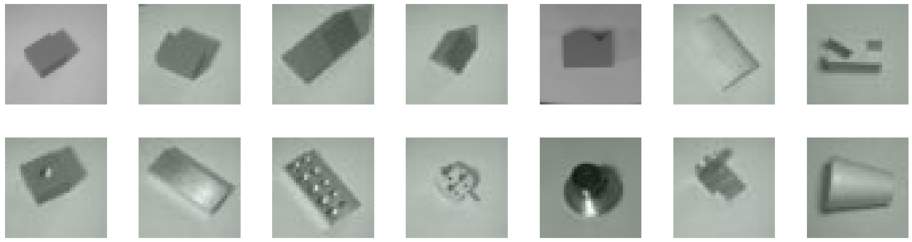

Figure 4: The 14 parts our procedure has been evaluated on.

started from simple parts and progressed to more complicated ones. The size of the parts that could be used was limited by the size of the rotating base and by the range sensor measurement bounds. The largest part's dimensions were $5 \times 10 \times 7cm^3$.

Data was gathered with the part attached to the base. The base rotation angles were measured anti-clockwise looking down the imaging surface motion direction vector (i.e. down the x-axis in Fig. 3). The first view was always taken with the base set at 90^o (i.e. with the base being horisontal).

This first view was used to build a voxelmap. The Best Next View determination process then assessed the current voxelmap every 10^o of possible base rotation to determine the second view direction and then the base was set to the new angle. So effectively only views on a circular arc were used in our experiments. Note however this was due to the type of positioning base available and is not a limitation of our algorithm. The views from the starting and new angle were then merged to build the new voxelmap and the same procedure was repeated until the Best Next View planning program requested the base to be set to an angle that had already been tried. The process then stopped since no new information could be derived from obtaining an already acquired view (except possibly some noise reduction from image averaging).

For most parts 3-4 views were enough and some parts needed up to 5 views. The average number of views used over all the parts was approximately 3.79. Selecting the Best Next View for our base takes 23.5 seconds on a SPARCstation Ultra 1. This means that the total sensor planning cost for most parts is $3.79 \times 23.5 \approx 89$ seconds. Taking one view with our range scanner requires approximately 120 seconds, and this figure accounts the time used to set the base to the correct angle. Therefore, the total view taking cost is $3.79 \times 120 \approx 454$ seconds. So, the the sensor planning cost is one $\frac{1}{5}$ that of the view taking cost, making our approach highly feasible in terms of speed.

Our results demonstrate that a correct voxelmap is constructed for a variety of objects with various properties. More importantly, the combination of the quality and visibility criteria selects views that span convex, concave and curved objects. Due to size limitations here, we will only discuss in greater detail the results obtained for the *double-wedge* part, which is the most suitable for demonstrating the quality criterion.

The *double-wedge* part and its generated voxelmap can be found in Fig. 5. In this voxelmap, the Seen voxels are grey while the OcclusionPlane voxels are black. Some OcclusionPlane voxels remain on the sides of the object. Note also the lighter gray areas (e.g. the one pointed by one of the arrows) correspond to Seen voxels of low Region Quality estimates.

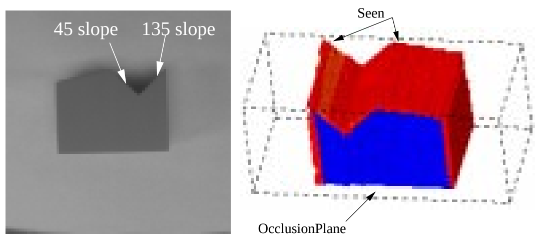

Figure 5: Double-wedge part and resulting voxelmap.

In order for the sensor to see the remaining OcclusionPlane voxels, on the sides of the voxelmap, it is necessary to rotate the object around a different axis, which is not possible with our base. However all other areas have been imaged. This shows that the visibility criterion worked. We can also see that there are very few voxels of low quality remaining. This implies that the quality criterion also performed its task.

In Fig. 6 we give plots of the visibility (a) and quality (b) criteria as well as their weighted sum (c) for the *double-wedge* part. In these plots the vertical axis marks the value of the criterion for different possible best next angles (0-180). The criteria values are obtained after each selected view is acquired (axis pointing out). Since we always start from an angle of 90^o the first evaluation occurs after the 90^o view.

The weights used in equation 3 for the experiments were $w_v = 1$ and $w_q = maximum(f_{visibility})$ for w_q. The justification of this choice is that we wanted to scale the very low ([0,1]) values for the quality criterion and make them comparable with those of the visibility criterion. So for the double-wedge part it was that $w_q \approx 10000$.

The object is first viewed from an angle of 90^o, and after this view the visibility criterion value for the 90^o view is low. This is because there are no OcclusionPlane voxels left to be seen from this view. However, high visibility criterion values are obtained for views that recommend that the part be observed at 0^o and 180^0. After the object is viewed from 0^o (the next view selected), the visibility criterion values become low at 0^o, which means that this side of the part has also been seen. The next view selected is the one that sets the base at 180^o and after this view the visibility criterion values also become low for 180^o. There is now a much smaller peak at the weighted sum at 50^o. This peak is caused by the quality criterion. The quality criterion wants to improve the quality of the data on the slope of the part which is at 45^o to the horisontal. The last view from 130^o is also selected because of the quality criterion now wants to improve the quality of the 135^o slope.

What can be seen from the sum graph is that the quality criterion is driving the Best Next View selection procedure after most OcclusionPlane voxels are seen and the visibility criterion values become very low. This essentially means that we have a 2 step acquisition process that first tries to capture as much data as possible and then tries to improve the already acquired data. This was one of the improvements for Best Next View processes suggested in [3].

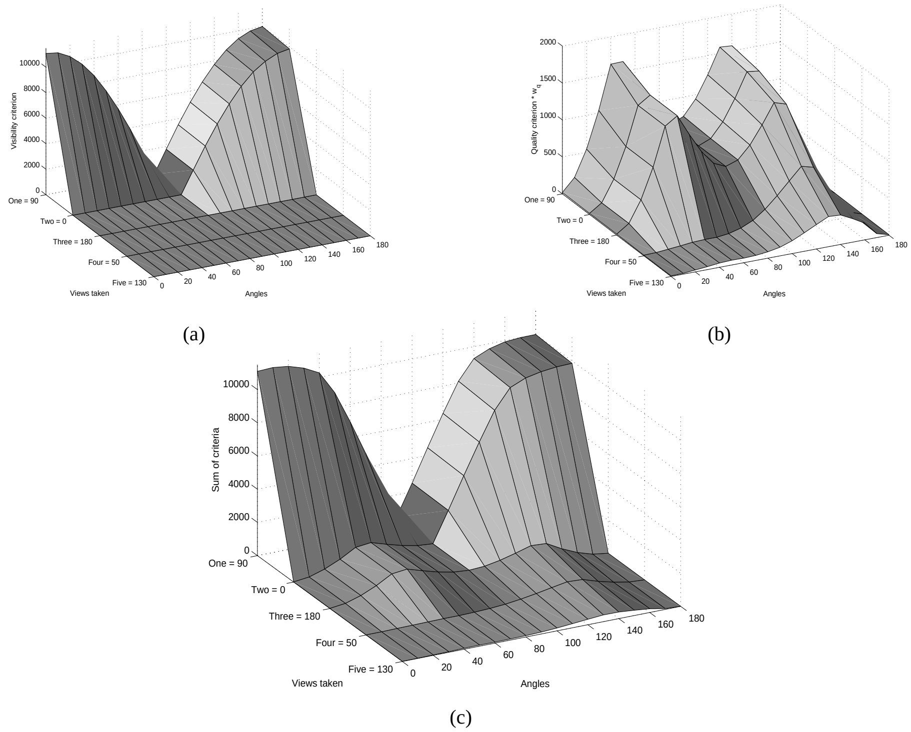

Figure 6: Criteria plots for the double-wedge part. a) The visibility criterion b) the quality criterion and c) the combined criteria. The horizontal axis is the angular viewpoint, the depth axis is the view number (back is first view, front is most recent view) and vertical axis is the criterion score.

5 Conclusions and Further Work

In this paper we used a volumetric representation to select Best Next Views for building 3D models of objects. The main novelty of the paper is to incorporate estimates of the quality of the range measurements into the volumetric representation. This representation is evaluated from viewpoints lying on a constrained tessellated sphere using a visibility and a quality criterion. Our results demonstrate that this approach generates a good acquisition plan for a variety of parts. Our voxelmap marking scheme can use the Region Quality Estimate to find areas of low quality. Projected voxelmap images demonstrate that the quality marking of the voxelmap building procedure also works well. In fact, our results suggest it might be possible to remove some of the erroneous measurements caused by specular reflection, by removing low quality voxels.

To sum up:

- The approach works for our combination of positioning mechanism and sensor.
- The voxelmap constructed is correct and adequately accurate for the task at hand.
- The combination of the quality and visibility criterion is effective.

- The procedure is efficient in terms of time.
- It is the only Best Next View procedure that uses a quality criterion.

Further tests can be done in evaluating views for a part positioning mechanism with more degrees of freedom. In this case it will be necessary to project views to a full sphere rather than a circular strip. Furthermore, in this paper we have analysed the performance of the approach in terms of the quality of the generated voxelmap. However, the voxelmap is not intented to be the final 3D representation of the object and it is only constructed so that the Best Next Views can be planned. More tests can be performed to check whether the 3D CAD models generated using the views selected are correct. Finally, the scheme introduced here improves the quality of previously measured voxels. One could also investigate a scheme that attempts to avoid taking voxels with low quality.

6 Acknowledgments

Many thanks to Anthony Ashbrook for technical help. Nikos Massios worked on this subject as a masters student at the University of Edinburgh.

References

[1] C.I. Connolly. The determination of next best views. In *Proc. IEEE International Conference on Robotics and Automation*, pages 432–435. IEEE Computer Society Press, 1985.

[2] R.B. Fisher, D.K. Naidu, and D. Singhal. Rejection of spurious reflections in structured illumination range finders. In *Proc 2nd Conf. on Optical 3-D Measurement Techniques*, pages 467–474, Zurich, Switzerland, October 1993.

[3] O. Hall-Holt. Scrutinizing real-world geometry: the next best view. Can be found at http://www-graphics.stanford.edu/∼olaf/nbv.html, 1998.

[4] N.A. Massios. Predicting the best next view for a laser range striper system. Master's thesis, Dept. of Artificial Intelligence, University of Edinburgh, Edinburgh EH1 2QL, Scotland, 1997.

[5] J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):417–433, May 1995.

[6] R. Pito. A sensor-based solution to the "next best view" problem. In *ICPR'96*, pages 941–945, 1996.

[7] M.K. Reed, P.K. Allen, and I. Stamos. Automated model acquisition from range images with view planning. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 72–77, Puerto Rico, June 1997. IEEE Computer Society Press.

[8] K.A. Tarabanis, P.K. Allen, and R.Y. Tsai. A survey of sensor planning in computer vision. *IEEE transactions on Robotics and Automation*, 11(1):86–104, February 1995.

[9] D. Wren. *Planning Dextrous Robot Hand Grasps from Range Data using Preshapes and Digit Trajectories*. PhD thesis, Dept. of Artificial Intelligence, University of Edinburgh, Edinburgh EH1 2QL, Scotland, 1997.

[10] H. Zha, K. Morooka, T. Hasegawa, and T. Nagata. Active modeling of 3-d objects: Planning on the next best pose(nbp) for acquiring range images. In *Proc. International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 68–75, Ottawa, Canada, May 1997. IEEE Computer Society Press.