Object Retrieval Strategy In Unstructured Environments Using Active Vision and A Medium Complexity Gripper

MS-CIS-90-96
GRASP LAB 246

Marcos Salganicoff
Michael Chan
Sanjay Agrawal
Ruzena Bajcsy

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

December 1990

Object Retrieval Strategy in Unstructured Environments using Active Vision and a Medium Complexity Gripper

Marcos Salganicoff, Michael Chan, Sanjay Agrawal and Ruzena Bajcsy
GRASP Laboratory
Department of Computer and Information Science
University of Pennsylvania

December 21,1990

Abstract

Depth information from a mobile laser stripe scanner mounted on a PUMA560 robot is used for simple thresholding by z-height and region growing. Superquadric surfaces are then fit to the regions segmented. This data reduction to three axis parameters, three Euler angles and two squareness parameters allows grasp planning using the PENN hand medium complexity end effector. Additionally, in order to take into account the spatial relationships between objects, they are grouped according to a nearest neighbor measure by distance between centroids in the x-y plane and also by height. The convex hull of the groups is then computed using Graham's method. The convex hull object list permits objects with the best clearance for grasp to be identified, thus reducing the possibility of unwanted collisions during during the enclosure phase. The geometric properties of the object are then used to determine whether an approach parallel or normal to the plane of support is necessary. This list of candidate grasps for the object is checked for intersections with the bounding boxes of neighboring objects and the finger trajectories. The most stable collision free grasp preshape which passes the intersection testing is chosen. If no grasp is collision free the next best object in terms of topology is chosen. Height clustering information is used to determine a baseline height for transporting objects in a collision free fashion. By combining these simple strategies of favoring objects at the exterior of groups and tall objects for initial grasping and removal, the chances for successful task completion are increased with minimal computational burden. [1].

[1]Acknowledgements:This work was in part supported by: Air Force grant AFOSR F49620-85-K-0018, Army/DAAG-29-84-K-0061, NSF-CER/DCR82-19196 Ao2, NASA NAG5-1045, ONR SB-35923-0, NSF INT85-14199, NSF DMC85-17315, ARPA N0014-88-K-0632, NATO grant No. 0224/85, DEC Corp., IBM Corp.,

1 Introduction and Motivation

Situations may often arise in hazardous materials removal or industrial packing situations where the class of objects to be manipulated is fairly well known, but the location and orientation of the objects and with respect to its neighbors is not known in advance and must be determined by active vision and tactile sensing. Examples of these situations might be the removal of broken fuel rods at the bottom of a reactor containment vessel, unstable ordnance or propellants which must be disposed of, or in undersea archaeological situations where the class (and thus the shape) of objects may be known in advance.

In these situations many issues must be resolved. First there is recognition of the target object is some reliable fashion which may involving active sensor movement in order to better determine the identity of an object. Then there is the determination of graspability and associated risk. An important observation is that graspability is determined not only by the object geometry and substance properties and the hand properties alone, but also by the spatial configuration of the object relative to it's surface of support and relative to other obstacles. The finger trajectories of a given grasp must be collision free with respect to neighboring objects and the approach pose of the wrist must also satisfy the collision constraints of neighboring objects. Once grasped in a stable manner, the object must then be transported safely and collisions must be avoided during transport. All of the above must be done in a timely way, and must take into account unanticipated changes in the environment during the task. Finally, the computational penalty for these unanticipated changes should be minimal. We attempt to address these issues in this work.

1.1 Previous Work

There is a large body of work in classical task planning. Previous work has generally assumed perfect sensory information and generated complete task sequences, exemplars of such systems include STRIPS [6] and the work of Lozano-Perez [5] and many others too numerous to mention. While impressive in the abstract, and having many interesting provable properties, these systems tend to suffer from overcommitment, namely any anomaly during task execution forces the system to completely replan the entire task execution. This entails a significant computational burden and associated time delays in reaction. Our approach is more in the spirit of Georgeff [2] and Schoppers [11] and reactive planning. We attempt to design a simple enough algorithm for generating the next action that it can be computed on the fly. Any anomaly detected by the system sensors requires only sensory action for reassessing system state and a small amount of computation to plan the next action. In general, actions are selected that bring the system closer to the goal state, although if the system errs, it quickly replans and recovers. Actions are also selected to enforce task constraints such as stable grasping and avoidance of collisions during grasping and transport. Previous work using this approach include the work of Tsikos [13] who used finite state automata formalism to determine next actions in parcel stack handling and segmentation and Campos [3].

2 Range Sensing

Active perception is accomplished via a laser ranging system mounted on the wrist of a PUMA-560 robot so that it may flexibly explore a large work area conisting of a large portion of entire reachable workspace of the robot. Grasps are then planned for the removal of objects in that workspace using a hand/arm subsystem. The arbitrarily sized and oriented region of interest is decomposed into subregions which are merged and compensated as needed to form a complete description of the entire region. Objects may be rescanned as necessary from different directions to mitigate the effect of illumination and line of sight occlusions which are inherent in laser-stripe type scanner.

2.1 The Image Coordinator

To make the system as general as possible, the input to the Image coordinator consists of $\{x, y, z, \phi, \theta, \psi\}$ coordinates and n by m subscans scans which are merged to form a unified range image, see fig. 9. The image coordinator is responsible for controlling the range scanning of this arbitrarily oriented rectangular patch of the workspace. Since the mobile scanner can only scan a fixed width swath of workspace, the image coordinator commands the robot to move in a trajectory which completes each subscan. The subscans consist of arm trajectories at fixed velocities.

2.1.1 The Mobile Laser Range Imaging System

Mobile laser ranging system present several advantages of over fixed scanners. The most important of these is that laser source and camera line of sight effects may be minimized by scanning from different directions. Maver [9] has investigated strategies to yield maximum information using a minimum number of scans and then Merging them. Sakane [10] has also tackled this problem in the HEAVEN system which permits efficient locating of cameras and lighting sources given the placement of objects in the scene, but this is done a-priori, not during the task. In our case, we use a simple strategy of scanning from multiple directions and Merging the different views.

Our Laser Range Imaging System consists of two components: The *LOOKER* and the *GUS* processing unit. The *LOOKER* is composed of a laser stripe generator and SONY XC-39 camera which generates video signal of the images obtained under the illumination of the laser stripe, and the *GUS* unit [13] processes the continuous sequence of laser images and generates a range image of the scene in real time.

2.1.2 Operation of the System

The *LOOKER* is called by its name because it can easily be mounted on the tip of a Puma 560 robot and can be made to "look" from different direction of a scene, see fig. 1. The entire system is implemented using the HEAP robot sensory driven robotics environment [1].

In operation, it moves linearly at a known constant velocity under robot control, thereby scanning the scene we are interested in. By geometry, it can be shown that the position of the laser stripe as observed by the camera is a measure of height of the nearest object intercepted by it. This video signal is sampled at a rate of $60Hz$ by the *GUS* processing unit and the range image is produced in real time.

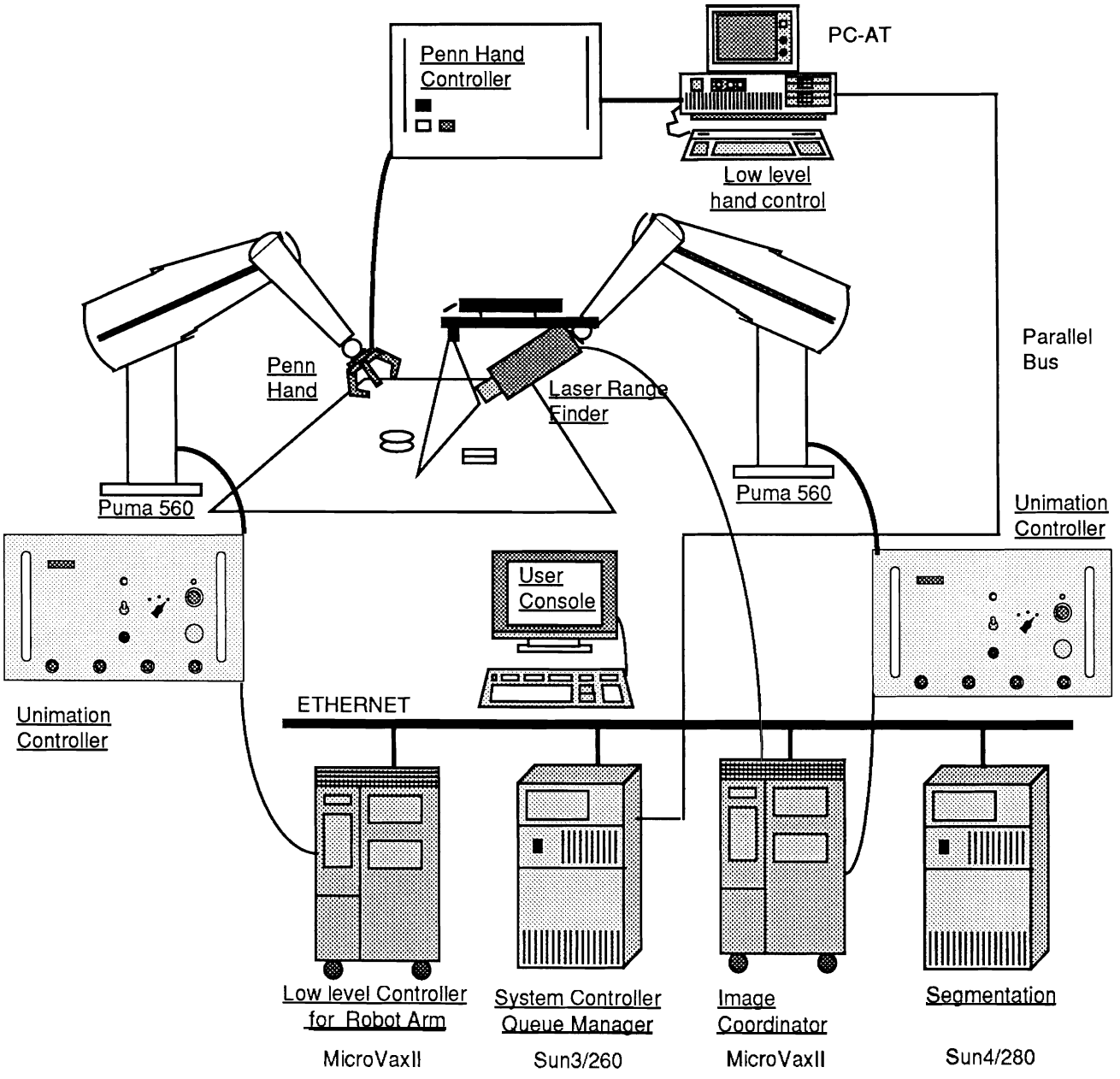

Figure 1: System Hardware

Synchronization between scanning motion and image generation is ensured by the ability to send a triggering command along a serial line connecting the host computer controlling the robot and the *GUS* processing unit.

The imaging volume of a single scan and the resolution of the range image are summarized as follows (for a motion rate of 4cm/sec):

Axis	X (width)	Y (length)	Z (height)
Imaging volume (mm)	135	164	172
Resolution (mm /pixel)	.23	.485	.672

Since the size of an image is limited by the imaging volume of *LOOKER* for a given resolution, multiple number of scans are needed in order to cover whole workspace we are interested in. Having the scanner under manipulator control allows us the flexibility of variable resolution in the Y direction. Noting that the resolution in the Y direction (the scanning direction) is a function of velocity of the scanning motion, it is often useful to obtain a coarse large area scan (scanning at a higher velocity) in order to locate approximately where the object(s) is/are.

This is useful in employing the robot for initial quick cursory scans of large amounts of the workspace. Gross forms be picked off during this phase and subsequently scanned at higher resolution finer detail is needed to characterize the object, regions with little interest may be subsequently ignored.

In surface regions where the laser stripe cannot illuminate or the camera cannot "see", pixel values of zero are assigned. Multiple number of scans of the *same* scene from *different* direction are needed to recover the occluded part of the scene as much as possible.

Another limitation of the imaging system is that orthographic projection is assumed in the generation of the range image . Software compensation is employed to counteract errors of this kind, especially for tall objects [15].

2.2 Subscan Merging, Region Growing and Culling based on Height, Connectedness and Area

Once all of the subscans have been performed an erosion operator is applied on each of the scans to reduce spurious measurements due to the sensing method. The subimages are then merged into a unified depth image encompassing the entire scene of interest. A height threshold of 5mm is applied to the height information. All points which pass this thresholding operator are then passed to an 8-connected region growing process. The region growing algorithm is $O(n^2)$ where n is the dimension of the image in pixels. When this algorithm terminates, it yields a list of regions, the extremal x and y values for each subregion (to form subwindows) and an associated area in pixels. Regions with areas below a minimum size (500 pixels) are discarded since we have a minimal size which may be grasped reliably by the manipulator.

2.3 Subregion Labelling, Extraction and Surface Fitting

Our domain consists of objects with arbitrary height, and partially constrained orientation, in that two of the major axes of the object must be parallel to the plane of support. The objects are not

currently stacked due to a the significant increase in vision computations to reliably accomplish this. Otherwise, the height and orientation in the plane is not constrained. The next phase of processing consists of generating the associated subimage for each bounding box containing the associated region's z-values. These subimages are passed to a superquadric surface fitting procedure [12] which generates a set of parameters for a parametric superellipsoid which best fits the range data of the sub-image. This results in a significant data reduction from a complicated range image to a set of 11 parameters which characterize the object and its position in the scene. These eleven parameters are $\{x, y, z, \phi, \theta, \psi, a_1, a_2, a_3, e_1, e_2\}$: where x, y, z describe the location of the centroid relative to the scanner frame; the ϕ, θ, ψ are Euler angles describing the rotational orientation of the principle axes of the shape; and e_1 and e_2 describe the squareness of the superquadric. The description is approximate, indicating the gross shape and pose of the object. However, it is sufficient in our case, since we are dealing with a medium complexity end-effector [14] and so are attempting rough enclosing grasps rather the fine manipulation.

In some instances, the region growing routine may fail in the sense that two objects may be adjacent in that there may be face-face, face-vertex or face-point contacts between objects. These contacts translate into 8-connectedness in the region image and so the region growing process will lump the data for two objects into one region and generate a single subimage for two objects. This case is handled by checking the goodness of fit functions for the resulting superquadric. If the goodness of fit is poor, this implies that the computed surface poorly explains the range data for that subimage. At that point more powerful segmentation techniques may be applied which take into account edge information or other higher level descriptions of the objects (see, for example [8, 12, 4]). Currently we simply do not process these cases. Even so, it is important to recognize that an unidentified object (obstacle) is present so that it may be avoided by use of its bounding box, or rescanned or segmented as needed by use of the active placement of the scanner.

Finally, our system may be fooled by stacked objects which appear to the scanner as a single object and would have a reasonable goodness of fit, but are actually non rigid, being composed of multiple objects. To handle such cases would require segmentation using edge information and also exploratory procedures to characterize the mechanical degrees of freedom between the constituent objects [3], but this is beyond the scope of this work.

3 Grasp Candidate Pruning

In order to proceed with the object retrieval in a safe manner we must make four fundamental decisions. These are which object to process next, which approach vector to use, the final grasp and the offset height to use when transporting the object. These are all interelated and so we proceed in the following fashion. First we select the best candidate to process first by looking a scene configuration information such as where the object is relative to its neighbors, and the object's relative stability (indicated by its height). We process objects with the greatest height and least neighbors first based on the fact that higher objects present the greatest chance of accidental collisions and they are in the least stable state in general, although this generally depends on the ratio of the major axes. First we cluster objects by height (see fig. 2), and form stratas of objects which are processed from tallest to shortest order. Then we cluster objects into groups in the plane

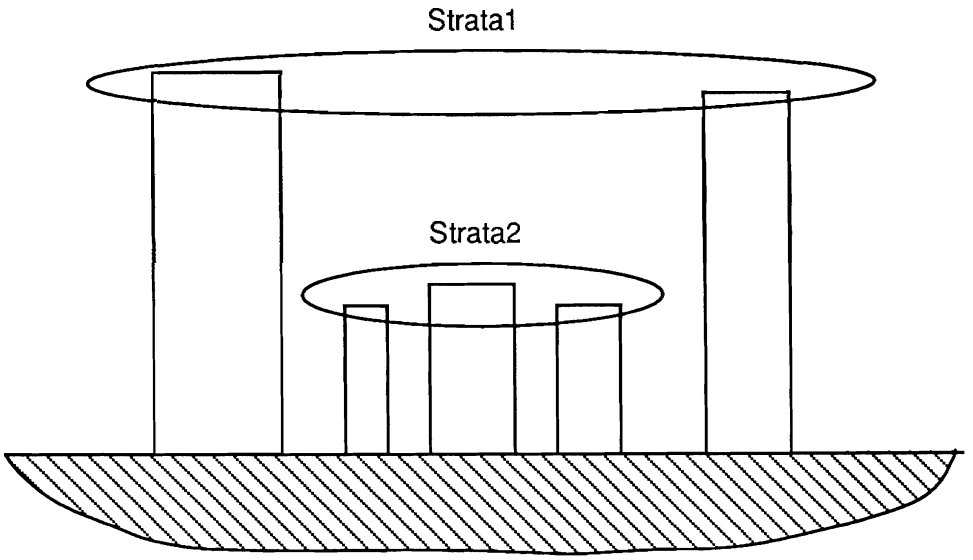

Figure 2: Strata formed by clustering by height.

and extract objects that form extremal point sets (convex hulls) for a given group (see fig. 6), we can in general, very quickly identify (in $O(nlogn)$ time) objects which are good candidates based on these criteria.

Once we have identified the object we would like to process first, we form a set of candidate grasps based on the geometry of the target. These grasps are ranked in terms of decreasing stability. We then check intersections of these candidate grasps with extent boxes of neighboring obstacles to find the most stable grasps which present no collision threat to neighboring obstacles.

3.1 Spatial and Height Clustering

Rather than computing a detailed and complete plan of the entire object retrieval sequence, we adapt a sensory driven approach in which a given action is locally computable based on the given state of the environment which inferred from the tactile and vision sensors. This is in line with our strategy of least commitment, which permits a flexible response without complete replanning in the event of unanticipated changes in the world. In this case we must replan only for the next object.

The spatial planning for the sequencing of the retrieval of objects in the scene is based on two simple rules of thumb. Objects which are grouped together in close proximity (using a nearest neighbor measure) should be grasped starting from the perimeter of the group and dealt with in layers, similar to the layers of skin in an onion. Also, given that the convex hull perimeter set of objects is available, each object in the set should be evaluated for its most obstacle free approach path. If possible an approach from the top should be preferred as the kinematic constraints for the PUMA are less and the possibility of collision with non-target objects by non-sensorized parts of

the manipulator increases drastically. In some cases, taller objects must be grasped from lateral approaches due to the instability of enclosing the object from a top approach.

The general outline of the grasp selection and sequencing process is illustrated in the flowchart in figure 3.

First we begin by clustering objects based on height in the z direction. This is done by first determining which axis of the superquadric is parallel to the z world axis using the following technique for determining a_x, a_y and a_z. The superquadric return a description of the object in the form $\{x, y, z, \phi, \theta, \psi, a_1, a_2, a_3, e_1, e_2\}$. These may be converted to a homogeneous transform matrix of the form using the Euler angles ϕ, θ, ψ to generate the rotation submatrix and the offset vector $\vec{p} = [p_x, p_y, p_z]^t$ to generate the translation.

$$ {}^{W}_{sq}T = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \vec{r_1} & \vec{r_2} & \vec{r_3} & \vec{p} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{21} & r_{22} & r_{23} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

The superquadric fitting routine returns with a_1 as along the x axis a_2 along the y axis and a_3 in the z axis in the ${}^{W}_{sq}T$ frame, where W is the world frame, with the z axis pointing up. Since our assumption is that the object is supported by the x, y plane, one of the three vectors $\vec{r_1}, \vec{r_2}, \vec{r_3}$ must have an inner product with [001] (the z axis) of value $\pm(1 \pm \epsilon)$ The ϵ permits some leeway of errors in the orientation of the superquadric surface fit due to numerical and imaging errors. This vector $\vec{r_n}$ with the appropriate inner product indexes the x, y or z axis in ${}^{W}_{sq}T$; based on this we pick $2a_1, 2a_2$ or $2a_3$ one of which is the height of the object above the plane of support.

Since we begin with tallest objects, we first decide whether they meet the criteria for mandating a lateral approach, namely that the ratio of a_z to a_x and a_y is above a certain threshold, known as ρ_{min} and that a_z is greater than a given height threshold. If so we attempt to plan for a lateral approach and grasp (see section 3.5). Otherwise we plan for an approach from the top.

This approach allows objects with the least amount of obstruction and stability and which present the greatest subsequent collision hazard to be removed first. This in turn will decrease constraints for subsequent objects since fewer objects will be present to cause path interference for approaches. In essence, by removing an object we are hill climbing since each object adds to the constraint set and the strategy is to gradually decrease the spatial constraint set in some low risk manner.

3.2 Convex Hull Extraction

The input to the algorithm consists of the x, y position of the centroids of all objects in the scene and the orientation of the principle axes, including target objects and obstacles, see fig. 4. This is generated by the segmenting and surface fitting processes as previously discussed. Each objects is then labelled and a pairwise distance is computed for all objects, requiring $\frac{n(n+1)}{2}$ storage locations in a table. A graph is then built according to the following rule: Let there be n objects and the set of objects be $O = \{O_1, O_2, ..., O_n\}$.The first phase consists of determining all pairwise distances which are less than a minimum distance threshold, d_t. A set of edges, $E = \{e_{ij} \mid d(O_i, O_j) < d_t\}$ is

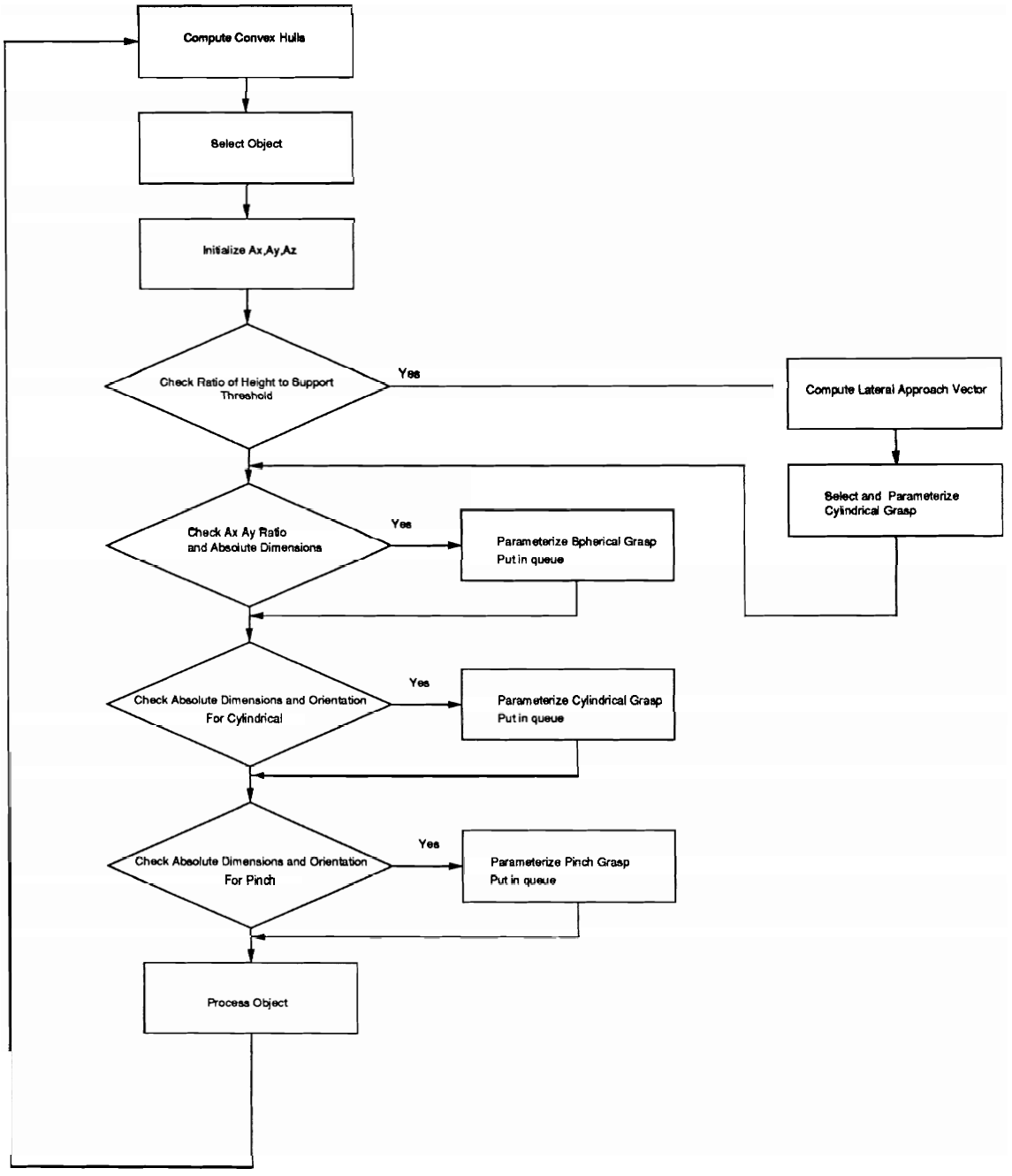

Figure 3: Control Flow for Candidate Grasps

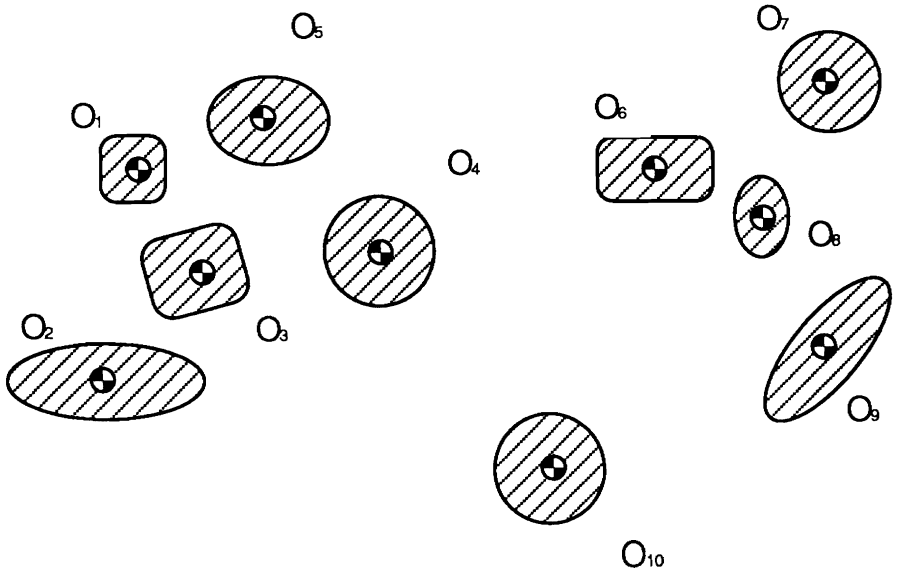

Figure 4: Initial Data Provided To Planner (Top View).

created for all objects that satisfy this minimum distance relation between each other, see fig. 5. Here $d(O_i, O_j)$ is the Euclidean distance metric between the centroids of objects O_i and O_j.

After this, the graph $G = \{O, E\}$ is searched and all connected components extracted. This forms a set of clusters $\{G_1, ..., G_m\}$ of nearby objects, see fig. 6.

Once the groups have been identified, a list of objects which form the convex hull of the cluster is formed for each group using Graham's method [7]. The leftmost bottom-most member of the grouping is identified as it must be a member of the convex hull. This point is labelled as the starting point. This method consists of identifying an interior point in the group and then performing a lexicographic sort based on polar angle and distance from the interior point. The result of this sort is then traversed in counterclockwise order and each triple examined as to whether it is a left or right handed turn. If a triplet results in a right handed turn, the midpoint of the turn is deleted from the current list of convex hull objects and a triplet consisting of the remaining two points and the previous point is tested. If it is a left turn then the triplet is advanced by one. When the last item in the triplet is the start point then what remains is the convex hull of the grouping. Essentially this process serves to remove all items which are in the interior of the grouping. This is advantageous since this objects will in general have more constraints as to which grasps and lateral approach vector are acceptable. The result of such a computation is shown in fig. 6. Also note that the angles of the resulting convex hull triples serve as an excellent indicator of the approachability of the object from a direction parallel to the plane, the more negative the magnitude the inner product of the two consecutive unit vector in the direction of the line segments, the more acute the perimeter angle at that point and the better approach clearance afforded parallel to the plane.

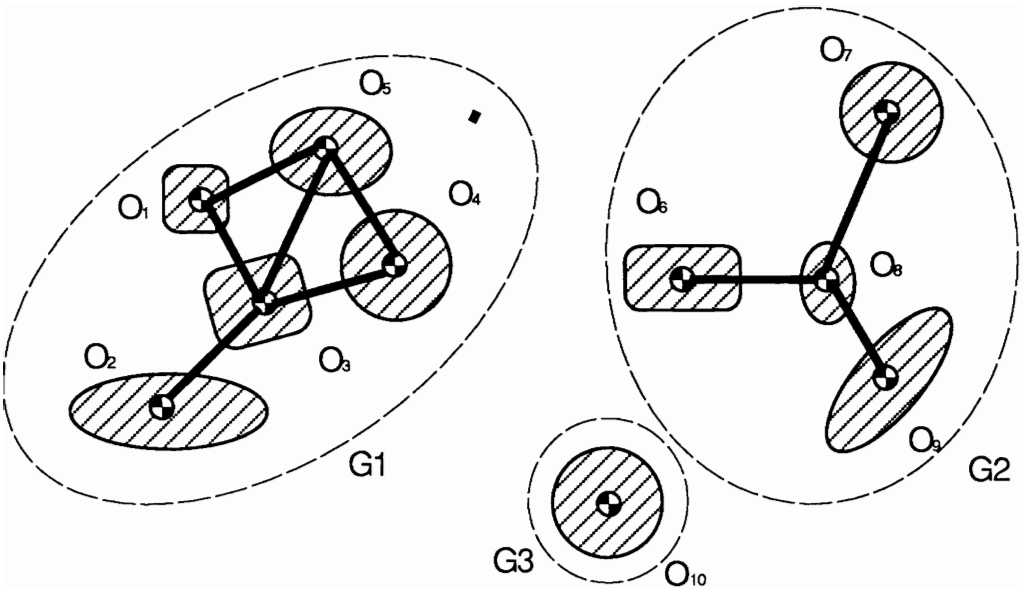

Figure 5: Connected Component Extraction.

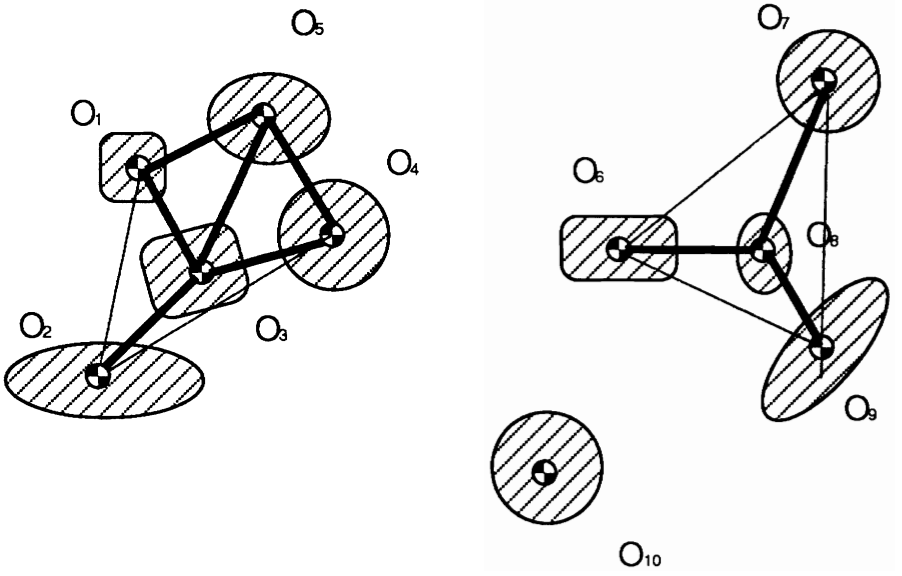

Figure 6: Nearest-neighbor clustering and convex hulls.

3.3 Approach Normal to the Plane of Support

Once the target object has been selected and the lateral approach ruled out for a given target object, grasps using the approach normal to the plane of support are then generated and tested.

Based on the kinematics of the PUMA 560 robot and that of the PENN Hand, together with this superquadric representation of the object concerned, the grasp planner picks a z axis approach vector, the desired grasping orientation, and which grasping configuration is most appropriate for the object. If a successful grasp is found, an approach vector is determined based on the additional constraint that one cannot easily change the wrist orientation of a PUMA560, since that would imply going through a singularity since we move in Cartesian space. If no feasible grasp is obtainable, then the next highest object is then evaluated and this iterates until an acceptable object is found.

3.4 Grasp Planning Based on Superquadrics Surface Descriptions for single objects

Our grasp planning is based solely on geometric object information for the sake of simplicity. In our grasp planning, we have adopted the following three grasping primitives:

- *Spherical grasp*
- *Cylindrical grasp*
- *Pinch grasp*

With the superquadric representation of the objects in the scene, we know the size of an object along its three major axes as well as its position and orientation, which can be characterized by a single homogeneous transform (the object frame) with the smallest a_i value is defined to be in the x-axis direction and the largest in the z axis direction. However, the 11-parameter superquadric representation of an object is not always unique [12]. For instance, two different row-pitch-yaw combinations can represent the same object, but with the positive z-axis pointing in opposite directions.

First, a homogenous matrix equivalent to the Euler angles is computed. We first attempt to compute a grasp which approaches along the z axis. The column vector of the rotation submatrix represent the coordinates of the unit vectors of the superquadric frame relative to the world frame. Since our objects are assumed to be oriented in the plane of support, then one of the column vectors will have a unit z component with positive or negative sign. Once this axis is identified, the two other axes in the x-y plane of the world frame can be identified.

Once the axis which is oriented in the z direction has been identified, the object frame is rotated so that this axis points in the positive z direction of the world frame. After this, the appropriate tool transformation may be determined from the orientation of the two axes in the plane. In general, we grasp so that the major superquadric axis is aligned with the x axis of the hand frame $\{hand\}$, this insures that the grasp width of the hand will not be exceeded if at all possible. At the same time, for each grasp (cylindrical, spherical and pinch) we have two candidate hand orientations,

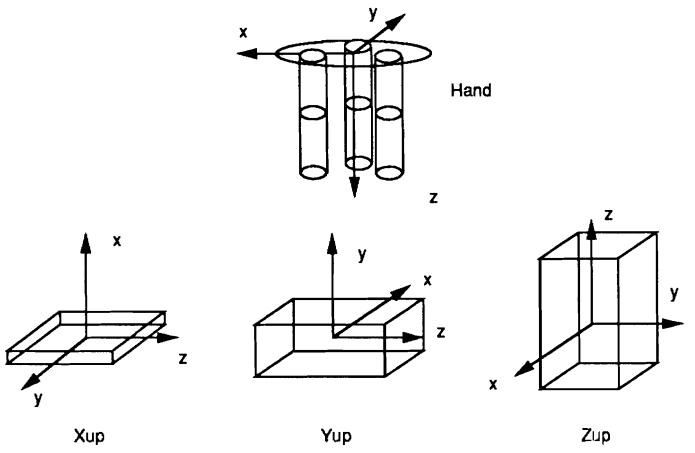

Figure 7: Object and Hand Frames

each rotated by π of each other. We pick the one which results in the least amount of rotation in the robot wrist relative to the current robot configuration.

Using analytical geometry we use the superquadric parameters described below to select which of the three primitive grasp types is to be used:

- a_1, a_2 and a_3 are the measures of the size of the object along its three principal axes. If the magnitude of these values and their relative sizes are within some predetermined range based on the geometry of the hand, a *spherical grasp* which will enclose the object is considered first.
- If a *spherical grasp* is not possible, a *cylindrical grasp* around the major axis of an object and with the fingers closing on the shorter of the minor axis is preferred.
- If both of the above are not possible because the object is too small, a *pinch grasp* will be generated.
- If none of the above is possible then the object is deemed ungraspable.

The position which the hand should approach from, the desired and final position and orientation of it are calculated and are formulated as homogeneous transforms respectively.

For the purposes of grasping we then define a_z as largest a_n, a_y as the intermediate value and a_x as the smallest value. We then examine which grasp are possible according to the following scheme. If a_z does not lie parallel the the world z axis then we can rule out a lateral approach and proceed with an approach from the top.

Top grasp wrist orientations are computed using the following set of matrices

$$R_{x_{up}} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \tag{2}$$

$$R_{y_{up}} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \tag{3}$$

$$R_{z_{up}} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \tag{4}$$

We then premultiply the rotational submatrix of ${}^{W}_{sq}T$ by the appropriate matrix from above depending on its configuration to yield the desired ${}^{W}_{grasp}T$. This brings the superquadric rotation component into the desired wrist orientation which defines the tool orientation for the approach, see fig. 7. Approach is determined by backing off along the $-z$ axis of the tool frame by a distance determined by the extent of the object along the z axis of the tool frame and the distance of the palm of the hand to the center of the robot wrist.

These positions are obviously not unique, for instance, one can approach from one side of a rectangular block as supposed to the opposite side if an approach parallel to the plane of support is being considered. However, the one which minimizes the arm movement in Cartesian space is always preferred.

The generated grasps are represented by bounding boxes which are then clipped against the obstacle bounding boxes in the plane to check for possible collisions. The hand bounding box consists of conservative bounds for the finger trajectories during the enclosure phase of the grasp.

3.5 Approach Parallel to the Plane of Support

If the object height and ratios are such the a lateral approach is necessitated the a cylindrical grasp is chosen and the approach vector in the plane chosen as follows. As mentioned before, he convex hull algorithm used (Graham's Method) returns objects on the convex hull in sorted counter-clockwise order. By definition all ordered consecutive triplets are "left-hand" turns. This implies that the angle formed $\angle O_i, O_j, O_k > \pi$, see fig. 8. The approach vector will lie somewhere on this angular interval. A new set of k nearest neighbors is found for each hull object using another distance threshold, d_a. In general, $d_a > d_t$, because the approach path for the robot wrist requires more room than the finger clearance, and d_a is chosen to provide ample room for the robot wrist. This angular range is then searched for the largest unobstructed angular approach interval cone, α_{max}, which is larger then α_{min} the minimum sized approach cone, and has length greater then l_{min} so the hand wrist combination may clear obstacles. If no approach cone with $\alpha > \alpha_{min}$ and $l > l_{min}$ can be found, then this candidate object is left alone and processed in a subsequent pass when fewer obstacles will be present.

This approach vector points in the direction formed by bisecting this interval angle.

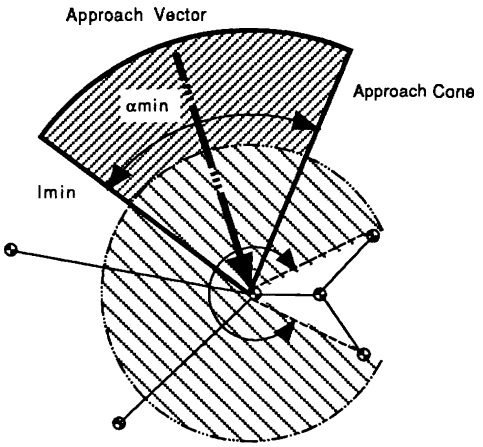

Figure 8: Determination of the Approach Vector

3.6 Task Execution

The planner puts the grasp type and approach direction on the motion queue for the coordinator to process [1]. The coordinator expects grasp type, parameters and approach direction and attempts to execute them. It monitors the outer finger tactile sensors for unexpected contacts and proceeds through if sensor values are as expected. If an anomaly is detected, such as unexpected contact or loss of contact with the target, then the hand/arm system backs off into a safe position and a rescan segmentation is scheduled to reasses the environment. Collisions between the two robots are avoided by having safe positions for each one and observing mutually exclusion. The transport height of the object is determined by taking the maximum height of the current strata, computing the height of the current object in its grasped orientation and adding the palm to wrist offset so that it will clear all other objects.

4 Results

An example of the system's operation is shown beginning in fig. 9 which is the initial range data for the scene acquired by the scanner and merged and eroded. Fig. 10 shows the result of surface fitting, height clustering and grouping. On the tallest object strata (shown by the thicker lines) is currently being processed for possible grasping. The selected grasp (cylindrical) is depicted by the thin lines comprising the three finger trajectories in the plane. The next figure 11 shows the final strata being processed after the first strata is complete. The outer perimeter line demarcates the convex hull of the grouping.

5 Discussion

The approach we have taken here is to attempt to decompose an inherently three dimensional situation in a two dimensional one. This is done by decomposing the scene into stratas of objects with similar height. Collision checking is also done in two dimensions because of this approach, which decreases the complexity of intersection checking computations significantly at the expense of some precision since an approximation is used. A drawback of this approach is that conservative bounding boxes for objects are used that might discard grasps that would be collision free, but at least here we are erring on the side of caution.

A philosophy of minimum commitment is observed at various levels. Vision processing is done as simply as possible initially by use of a coarse thresholding and sampling. Finer resolution scans of "interesting" regions may be accomplished on demand by varying the scan speed of the scanner. The simplest segmentation is first attempted, namely that of connectedness and region growing, followed by surface fitting. If the quality measure of the segmentation is inadequate, then the subimage may be passed to a more capable but computationally more expensive segmentation process. The culling out of areas below a given threshold area prevent the system form engaging in the surface fitting and grasp planning of objects the are inherently to small to be grasped and prevent noise in the form of spurious measurements from being problematic. Minimum commitment planning is used by only planning for the next action and basing this planning only on what is observable via the sensors. A drawback of this philosophy is that the future behavior is not known at the onset of the task and risk optimality of the strategy is not easily provable. However in an unstructured environment is not generally not predictable, so it makes little sense to attempt to predict the success of complete task plans in mischievious environments such as the real world. The revised planning process based on sense data is in generally to costly and the system ends up thrashing constantly replanning. This stems not only from unknown agents in the environment, but also from the intrinsic uncertainty in predictive physical models, especially as we try and predict state further and further in the future, as anyone who has tried to plan their wardrobe based on a five day forecast is acutely aware. We address this by reducing the complexity (and optimality) of the planning process, but gain flexibility by rapidly detecting and responding to errors and unforseen events sensed by sensors. From another perspective, this approach also treats each object retrieval as an information gathering process allows us to better characterize the environment as we continue the task.

6 Conclusion

We have demonstrated a methodology for object grasping and retrieval using various conservative simplifying assumptions which in general will be low risk and require little computation. The system is extensible and robust to task execution failures. Future extensions will include adaptive components to the action generation based on past performance in grasping and collisions and the addition of more powerful manipulatory routines.

References

[1] S. Agrawal, M. Salganicoff, M. Chan, and R. Bajcsy. *HEAP: A sensory driven distributed manipulation system.* Technical Report MS-CIS-90-90, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, Pa., 1990.

[2] M.P. Georgeff annd A.L. Lansky. Reactive reasoning and planning. In *AAAI-87*, pages 677–682, 1987.

[3] M. Campos and R.K. Bajcsy. *A robotic haptic system architecture.* Technical Report MS-CIS-90-51, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, Pa., 1990.

[4] R. Bajcsy et. al. *Final report on advanced research in range image interpretation.* Technical Report MS-CIS-90-14, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, Pa., 1990.

[5] T. Lozano-Perez et. al. *Robotics Research 4*, chapter Handey: A Task-Level Robot System, pages 29–36. MIT Press, Cambridge, Ma., 1988.

[6] R.E. Fikes and N.J. Nilsson. Strips: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[7] R.L. Graham. An efficient algorithm for determining the convex hull of a planar set. *Information Processing Letters*, 1(4):132–133, 1972.

[8] A. Gupta. *Part description and segmentation using contour, surface and volumetric primitives.* Technical Report MS-CIS-89-33, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, Pa., 1989.

[9] J. Maver and R.K. Bajcsy. *How to decide from the first view where to look next.* Technical Report MS-CIS-90-39, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, Pa., 1990.

[10] S. Sakane, T.Sato, and M. Kakikura. Model-based planning of visual sensor using a hand-eye action simulator system :heaven. In *Proceedings of the 3rd International Conference on Advance Robotics*, Versailles, France, 1987.

[11] M.J. Schoppers. Universal plans for reactive robots in unpredictable environments. In *IJCAI-87*, pages 1039–1046, 1987.

[12] F. Solina. *Shape Recovery and Segmentation with Deformable Part Models.* Technical Report MS-CIS-87-111, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, Pa., 1987.

[13] **C.J. Tsikos and R.K. Bajcsy.** *Redundant Multi-Modal Integration of Machine Vision and Programmable Mechanical Manipulation for Scene Segmentation.* Technical Report MS-CIS-88-41, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, Pa., 1988.

[14] N.T. Ulrich, R. Paul, and R.K. Bajcsy. A medium complexity end-effector. In *Proceedings, 1988 IEEE International Conference on Robotics and Automation*, pages 434–437, 1987.

[15] H.J. Wang and A. Gupta. *Projection Correction.* Technical Report, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, Pa., 1989.

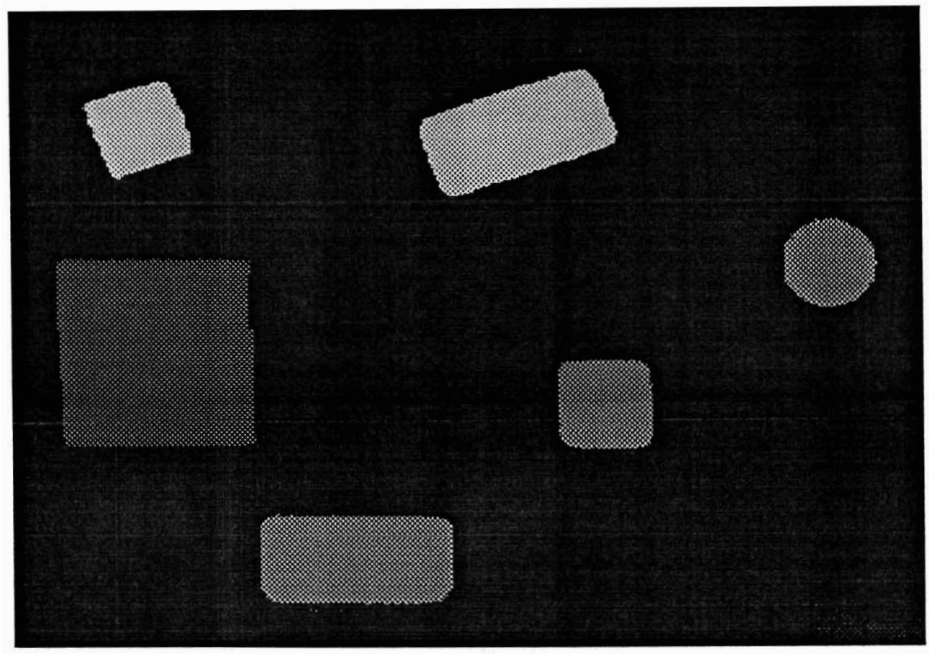

Figure 9: The Merged Range Image

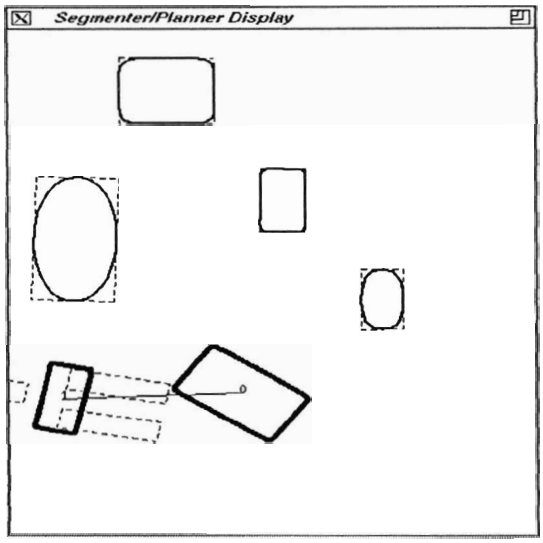

Figure 10: The Reduced Superquadric Representation of the Objects in the First Strata

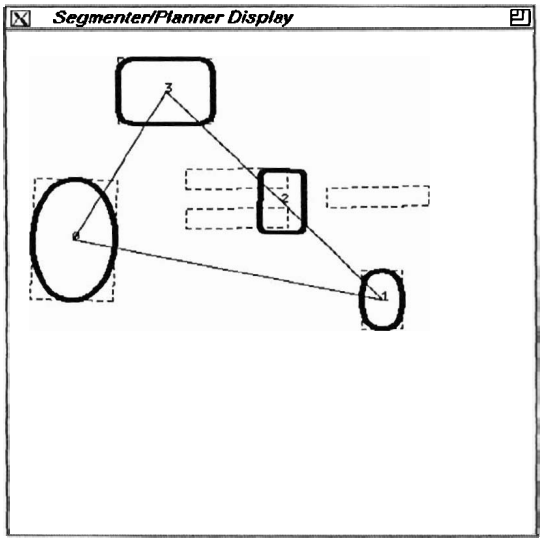

Figure 11: Convex Hull Processing and Grasp Selection for Second Strata