Proceedings of the 1999 IEEE/RSJ
International Conference on
Intelligent Robots and Systems

A Fast Two Dimensional Image Based Grasp Planner

Kevin Stanley* *Student Member, IEEE*
Jonathan Wu *Member, IEEE*
Ali Jerbi *Member, IEEE*
Vision and Scanning Group
National Research Council of Canada
Vancouver, BC Canada

William A. Gruver *Fellow, IEEE*
School of Engineering Science
Simon Fraser University
Burnaby, BC Canada

Abstract

This research concerns a grasp-planning algorithm that is fast and capable of determining grasp points for planar non-degenerate objects. We use a novel representation of the target and a quadtree based sampling scheme to generate a set of candidate grasps which are evaluated using a cost function. This function returns the first acceptable grasp point it finds. The resulting system has an execution time of seconds and is suitable for a large number of planar grasp planning problems.

I. Introduction

Over the past ten years, research in grasp planning and computing power have advanced to the point where online grasp planning is feasible. Both theoretical and practical aspects of the problem have been examined to determine the computational bounds, mechanical constraints, and application issues. Research into grasp planning can be categorized into model-based, dexterous and sensor based grasping algorithms. Model based grasp planning algorithms are the most theoretically complete. Dexterous grasp planning and manipulation involves the planning of the grasp, and motion of the target through motion of the fingers. Sensor based grasp planning generally involves planning grasps directly from sensor input without the aid of a model.

Nguyen[1] laid out the basic framework for grasp planning and stability in his 1989 paper on grasping polygonal shapes. His work is the foundation for model based grasp planning. Ponce, *et al.* [2] expanded Nguyen's work to curved two-dimensional shapes, and Montana [3] gave definitions of contact stability. Work in model based grasp planning has primarily moved to dexterous manipulation of parts using multi-degree of freedom hands (see Shimoga [4] for an overview). Sensor based grasp planning, which until recently has not received much attention, dynamically determines the grasp of an unknown object based on sensor input. Sensor based grasp planning is generally used in conjunction with visual servoing for vision guided part manipulation[5]. Bard, *et al.* [6] proposed a grasp planning system using a voxel representation of the target. Their implementation placed a heavy computational load on the processor because of the large number of voxels required for accurate representation of the target. They used an octtree data representation to reduce the complexity of the system. Janabi-Sharifi and Wilson [7] presented a feature based grasp planner and manipulation system that relies on global features to determine a grasp. Sanz, del Pobil and Inesta [8][9] have described a planner employing symmetry information to determine suitable grasp points.

We propose a sensor based planar grasp planner, similar to that of Sanz *et al.* in structure, but different in execution. Our grasp planner is a local image based grasp planner which employs a cost function to determine grasp quality as in Sanz *et al.* However, there are several key differences between our planner and previous work. First, we use an innovative representation of the target to make the cost function evaluation possible with little more computational effort than a set of lookups. Second, we use a progressive sampling technique similar to a quadtree expansion to generate potential grasp points. Third, we use different feature calculations and definitions to generate our cost function. The result is a general, fast, two-dimensional grasp planner. Our planner achieves planning speeds from fractions of a second to less than eight seconds depending on the complexity of the target.

To reduce the number of calculations, we assume that most objects in industrial assembly tasks will have several suitable grasp points. Symmetrical objects are common in assembly and machining tasks. The proposed algorithm evaluates points on the boundary of the object, and progressively refines the search until a grasp is found or the pixel level is reached. To accomplish the sampling of the perimeter of the object, the image is divided into many increasingly finer resolution windows. If the window contains a piece of the boundary, a single point on the boundary is sampled and the system determines the quality of the grasp. The fitness of the grasp point is determined by a cost function. The cost function is a weighted linear combination of the parameters describing the geometry and stability of the grasp. The algorithm

*National Research Council of Canada
Integrated Manufacturing Technologies Institute
3250 East Mall Vancouver, BC, Canada, V3J 1H1

iterates until a grasp with a cost less than a user-specified maximum is found. The grasp points are returned and a setpoint for the visual servoing control system is generated. The grasp planner finds grasp points that are satisfactory, that is, they are collision free, they are stable, and they pass near the center of area. The algorithm evaluates all grasp points at a given level of resolution, and returns a grasp point that satisfies the requirements, and is rated the best by the cost function. We make no claims on the global optimality of the grasp point based on any current used criterion, only that the point is graspable, and relatively better than the others examined.

To simplify the problem several assumptions were made.

1. *There is good intensity contrast between the target and the background.*
2. *Parallel grippers only grasp normal to the object surface.*
3. *Target objects are planar or prismatic.*
4. *Target objects are graspable.*
5. *The target is completely visible (no occlusion or clipping).*
6. *The grasp planner should operate on the order of seconds.*

II. Image Processing

The image-processing segment of the grasp planner must alter the image to create an efficient boundary representation of the target. We perform the image processing operations on a DSP for speed, freeing up the host CPU for control and communication tasks. The final representation of the image used for grasp planning should have the following properties:

1. *Representation of interior and exterior of the object.* Besides knowing where the object's boundary is located, we also must know what is inside and outside the target.
2. *Representation of the boundary.* The boundary representation should make feature calculation efficient.

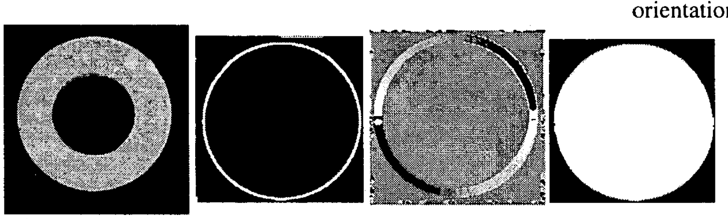

Figure 1: Image Processing Results

The exterior of the object is represented as black pixels (value 0), the interior is represented by white pixels (value 255) and the boundary is represented by the gradient orientation, which is analogous to the surface normal. The gradient orientation is represented as an angle from 0 to π, discretised on the grayscale interval 1 to 254. The result is a black and white image surrounded by a grayscale shell with a period of two as shown in the "Final Image" portion of Fig. 1. This representation allows the grasp-planning algorithm to identify easily the edge, interior and exterior and also to find the surface normal using only a table lookup.

We use a standard hole elimination procedure to ensure that the grasp planner only generates grasps spanning the exterior boundary of the target. Our planner does not consider interior grasps valid. The negative of the original image is taken, and the background inverted, so the only remaining elements are holes. The original is then added to the new image to generate a solid target.

The representation in Fig. 1 is a combination of the area of the target as determined by a grayscale blob analysis and the boundary of the target as determined by the gradient direction. The gradient direction is analogous to the surface normal, an important measure for the determination of the stability of the grasp. The gradient and gradient direction are determined by nine element Sobel operators where the gradient magnitude is given by[10]:

$$g = \sqrt{\left(g_x^2 + g_y^2\right)} \quad (1)$$

The gradient orientation is given by:

$$\theta = \tan^{-1}\left(\frac{g_y}{g_x}\right) \quad (2)$$

The gradient measures are based on the derivative of the grayscale; therefore the gradient magnitude and orientation are susceptible to noise. In order to reduce the noise, multiple iterations of a smoothing filter are applied to the holeless image before the gradient orientation is generated, giving rise to the wide ring apparent in the "Gradient Orientation" image in Fig. 1.

The use of the gradient orientation to represent the boundary is critical for the speed performance of the system. Most of the features, used to determine the grasp quality in our system, are directly based on the surface normal and hence the gradient magnitude.

III. Grasp Planning

The grasp-planning algorithm is based on a quadtree resolution expansion of the image. At each level of the tree the pixels corresponding to the x and y midpoints of the current window are sampled. If no suitable grasp points are found, the cells of the quadtree are divided, and the procedure repeated. An example expansion is shown in Fig. 2. Because a breadth first search is used, the algorithm has more in common with sampling than searching. If the boundary were stretched along a line, the final result would be a discrete sampling of the frequency quadrupled for every level of expansion. A fitness function is used to determine the suitability of a grasp point. The fitness function is a linear combination of features that represent the quality of the current grasp. The behavior of the system is tied closely to the fitness function.

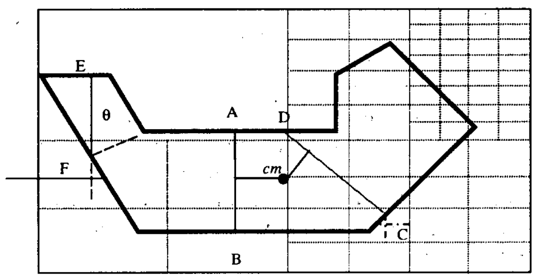

Figure 2: Example Object Expansion and Features

Fig 2. shows the quadtree expansion and the different features used in calculating the cost function for the grasp analysis. The dotted lines show the quadtree decomposition of the image. Cells not containing edge elements are discarded. The dashed lines above and to the left of C show midpoint sampling. The lettered pairs illustrate potential grasp points for analysis.

The node evaluation function is the main algorithm of the grasp planning system, which is performed, on every node in the lowest level of the tree. The feature extraction functions are called, then the opposite grasp point is found, and then the cost function is evaluated. The node evaluation algorithm also records the best grasp point at a given level of resolution. The feature evaluation and cost function routines are called twice for each node, once for the boundary point corresponding to the x midpoint, and again for the boundary point corresponding to the y midpoint.

While not crucial to the overall method, the opposite point algorithm is often executed. The search for the opposite grasp point is performed directly on the final image to reduce the amount of bookkeeping and memory. Because we know the second grasp point must be along the line normal to the surface at the first grasp point, we can reduce the two dimensional search to a single dimensional search. Because the system only has four values, interior, exterior and both edges, a steepest decent search is impossible. Examining the data set closely reveals that in fact are only two values, inside (max) and outside (min). The algorithm terminates when it reaches an edge value. We used binary search which terminates in *log(N/n)* time, where N is the total number of pixels in the line and n is the width in pixels of the opposing boundary.

A. Objective Function

The objective function dictates the behavior of the system by determining the relative quality of the examined grasps. The current fitness or cost function is a linear weighted sum of three features: the curvature of the boundary in the window, the distance of the grasp line from the center of area, and the angular difference in surface normal between the two grasp points. The objective function is described by the following:

$$c_{ab} = a\sigma_\theta + b(\theta_1 - \theta_2) + cd_{cm} \tag{3}$$

where σ_θ is the standard deviation of the grayscale histogram, θ_1 - θ_2 is the difference between surface normals of the two grasp points and d_{cm} is the distance from the grasp axis to the center of mass. Examples of these parameters are shown in Fig 2.

1. *Curvature:* We use the standard deviation of the surface normal along the boundary segment as a measure of curvature. This measure of curvature is independent of the size of the current window, it is consistent for similar curves, it is independent of the curve's position, and it is very easy to calculate given the grayscale histogram of the boundary orientation. Because the grayscale histogram is generated to determine if the window contains a boundary, it can be reused to find the curvature. The curvature can be calculated in a time of *2(N-2)* where N is the number of grayscale levels.
2. *Parallelism:* The value of the parallelism feature determines the grasp stability. Parallelism is measured as the difference between the surface normals of the opposing grasp points. Because the first grasp point is assumed to be along the normal of the surface, it is guaranteed to be within the friction cones. However, the second grasp point may not lie within the opposing friction cone. Therefore, the stability of the grasp is determined by the difference in orientation between the two surface normals. In addition, because we have assumed parallel jaw grippers, the most stable condition occurs when the opposing grasp points are on parallel faces of the

target. This measure selects grasp point pairs that lie on faces closer to parallel.

3. *Distance to Center of Mass.* The feature is calculated as the perpendicular distance from the center of area of the object to the line connecting the two grasp points. This is a measure of the stability of the grasp with respect to rolling about the grasp axis.

B. Stability

The key measure of the quality of a grasp is its stability. In his seminal 1989 paper Nguyen [1] determined that if a two fingered grasp with friction is to be stable, the friction cone from each grasp point must overlap, and the line connecting the grasp points must lie within this overlapping region. While we do not explicitly calculate this, a combination of feature weighting and our assumptions ensures that grasps are always stable.

Because we assume that all grasps must be normal to the current axis, we also know that all the initial points are stable. The applied force is always normal, and, therefore in the center of the friction cone. (See Fig. 3). Consequently, the stability of the grasp is completely determined by the relationship of the surface normal of this first grasp point to the second. According to our algorithm, the second grasp point is always along the normal of the initial grasp point we know that this line is within the first friction cone. If the line is also within the second friction cone then the system should be stable. The parallelism feature determines the stability. If the difference in angle is greater than the threshold

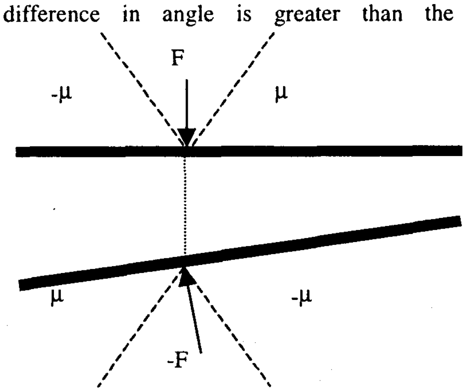

Figure 3: Stability Conditions

determined by the friction cone, then the coefficient of parallelism b must be sufficiently large to return a high cost. We have tuned the parallelism coefficient to return a value greater than unity for unstable configurations.

IV. Results

The algorithm was tested on several classes of shapes to determine its ability to correctly and efficiently determine acceptable grasp points for all types of objects. Objects were based into three broad classes:

1. Straight convex objects
2. Straight concave objects
3. Curved concave objects

A. Experimental Configuration

We have tested our system using an end-effector mounted Toshiba IK-M41A mini-camera with a resolution of 640 by 480 pixels. After the object was located with a thresholding technique, we reduced the window to the bounding box of the target. Image processing was performed on a Matrox Genesis DSP board. Grasp

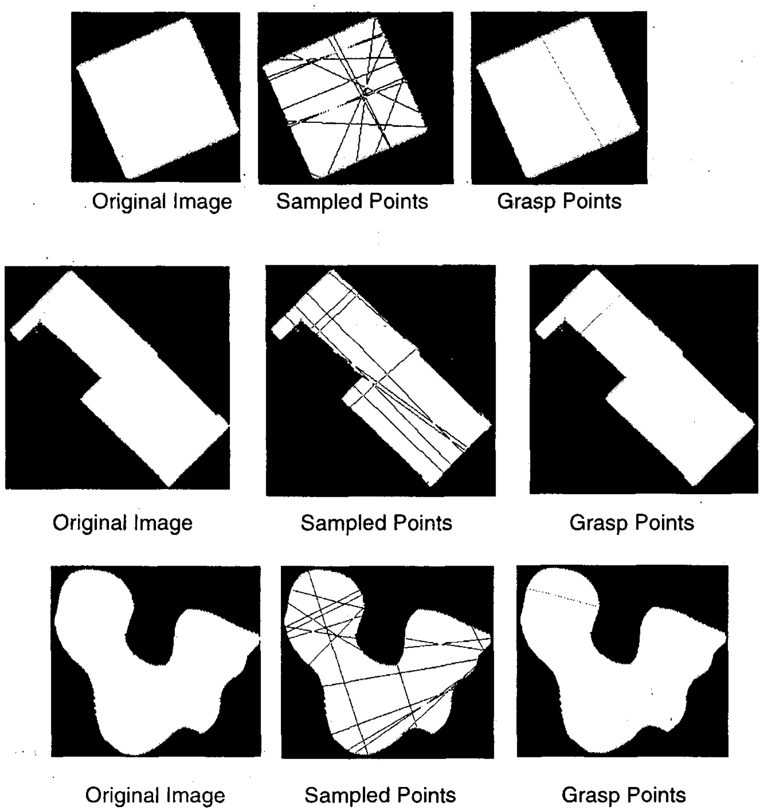

Figure 4: Planning Results

planning and feature calculations were performed on a dual 200MHz Pentium Pro computer running with under the QNX operating system.

B. Straight Convex Objects

This set of objects includes the common convex polygons. The grasp planner must not only find acceptable grasp points for simple shapes, it should also find good grasp points. In this case, the distance metric must distinguish between points on the periphery of the target and grasp points whose contact line passes though the center of mass of the object. To demonstrate the behavior of the system for this case we have used a square block as shown in Fig. 4. The grasp selected for the square is near the center of mass along the edges, as we desired.

C. Straight Concave Objects

The class of straight edged concave objects includes all non-convex polygons. Non-convex polygons are commonly found in flanges, braces and other assembly pieces. The requirements for grasp planning for non-convex polygons are similar to the requirements for convex polygons; the grasp point must be stable and preferably near the center of mass. Collision detection must be added to ensure that the robot does not collide with the target. We use a simple linear approximation to model the parallel grippers when performing collision detection. To illustrate the behavior of the system with respect to concave polygons, we have run the grasp planner with a bracket, as shown in Fig 4. The grasp planner generates a grasp across the narrower region of the target in a physically valid location.

D. Curved Concave Objects

The class of curved concave objects includes all curved non-convex objects and all objects with mixed straight and curved sides, making it the most general and complex set. We only assume that the boundary of the target is piecewise continuous. For this class of shapes, the focus of the grasp planner is finding an acceptable grasp point quickly. The d_{cm} measure and its coefficient must be small enough not to disqualify points far away from the center of mass if they have good curvature and parallelism characteristics. To demonstrate the behavior of the system with respect to curved concave objects we used the arbitrary shape shown in Fig. 4. For this example the planner determined a grasp point relatively early. There are few grasp points on that object, and the number of iterations required to determine the location is sensitive to the initial configuration of the target.

E. Time Results

The primary benefit of our algorithm is reduced evaluation time due to efficiency of representation and early stopping for suitable solutions. Theoretically, the algorithm should be exponential with the level of expansion; however, the data shows a significantly lower complexity, bounded by a polynomial of order 2. It appears that the early stopping and the pruning of non-boundary nodes reduce the problem to less than exponential complexity. From Table 1, the time required to plan a grasp varies from less than a second to less than 8 seconds, depending on the complexity of the object. Table 1 shows the average time for each level of expansion as well as the number of points examined in that level. The time durations listed in Table 1 are for the grasp-planning portion of the algorithm only. The time required to perform image processing is independent of target complexity and therefore not included.

The algorithm finds grasp points quickly. For most simple applications, a grasp point can be found within two levels of expansion. More complex objects require more samples but can be analyzed by the same algorithm. In all cases grasp planning is completed in a matter of seconds.

Table 1 : Average Execution Times

Expansion Level	Number of Points	Average Time (s)
1	32	0.093233
2	128	0.453123
3	512	1.20644
4	2048	2.4762
5	8192	7.79045

V. Future Work

The grasp-planning algorithm is an excellent example of local image based processing; however ,there are some additional issues that need to be addressed for the system to function robustly. First, the image noise needs to be reduced to eliminate unnecessarily rejected points. Second, the system needs to be integrated with the current visual servoing system. Third, additional search and cost functions should be investigated to determine their effects on the system. Finally, a sensor fusion framework should be set up to integrate multiple cameras into the workcell so the system can plan grasps for non-planar targets.

VI. Conclusion

The major contributions of this paper are an efficient image representation and a fast sampling algorithm. We have developed a fast, accurate image based algorithm capable of planning grasps for a wide variety of planar targets. A secondary advantage of this approach is its ease of application to sensor fusion systems. Building such a system is the major focus of our future work.

The flexible grasp planning system uses an approximate decomposition method to determine useable grasp points for objects of various shapes. The system is designed to find grasp points that are acceptable, but not optimal. The system uses a quadtree decomposition of the object boundary to determine likely sites for grasping the object. The source for the quadtree analysis is a grayscale image of the object, composed of an explicit representation of the interior exterior and surface normals around the object. The quality evaluation of the grasp is performed using a cost function composed of a weighted sum of the difference in normals, curvature and distance from the center of area. Potential grasps are checked for physical validity using a collision checker and a model of the maximum grip length of the end-effector. We have demonstrated successful grasp planning for a variety of different target shape classes. We have shown that grasps are planned in the order of seconds. Future work will include integration with a visual servoing system and a sensor fusion system for using multiple cameras to perform three dimensional grasp planning and execution.

VII. References

1. Van-Duc. Nguyen, "Constructing Stable Grasps," *Int. J. Robotics Research,* Vol. 8, No. 1, February 1989.
2. Jean Ponce, Darrell Stam, and Bernard Faverjon, "On Computing Two-Finger Force-Closure Grasps of Curved 2D Objects," *Int. J. Robotics Research, Vol. 12, No. 3,*. June 1993.
3. David. J. Montana, "Contact Stability for Two-Fingered Grasps," *IEEE Trans. on Robotics and Automation,* Vol. 8, No. 4, August 1992.
4. K. B. Shimoga, "Robot Grasp Synthesis Algorithms: A Survey," *Int. J. Robotics Research*, Vol. 15, No. 3, June 1996.
5. Seth Hutchinson, Gregory D. Hager,. and Peter I. Corke, "A Tutorial on Visual Servo Control," *IEEE Trans. on Robotics and Automation*, Vol. 12 No. 5, October 1996.
6. Christian Bard, Christian Laughier, Christine Milesi-Bellier, Bill Triggs, and Gianni Vercelli, "Achieving Dextrous Grasping by Integrating Planning and Vision-Based Sensing," *Int. J. Robotics Research,* Vol. 14, No. 5, October 1995.
7. Farrokh Janabi-Sharifi, and William J. Wilson, "Automatic Grasp Planning for Visual Servo Controlled Robotic Manipulators," *IEEE Trans. on Systems Man and Cybernetics*, Vol 28, NO. 5, October 1998.
8. P.J Sanz, A. P. del Pobil, and J. M. Inesta, "Curvature-Symmetry Fusion in Planar Grasping Characterization from 2D Images," *Proc. of Industrial Engineering Applications of Artificial Intelligence and Expert Systems, Amsterdam, The Netherlands,* 1997.
9. P.J Sanz, A. P. del Pobil, and J. M. Inesta, and G. Recatala, "Vision Guided Grasping of Unknown Objects for Service Robots," *Proceedings of the 1998 IEEE International Conference on Robotics and Automation,* Leuven, Belgium, May 1998.
10. E. R. Davies, *Machine Vision,* Academic Press, New York, 1997.