Generating Grasping Positions Using Models Based on Duality

by

Michael Dannhardt

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Masters of Electrical Engineering

in

Electrical Engineering

APPROVED:

Douglas Lindner, Chairman

John Roach

Richard Conners

October, 1988

Blacksburg, Virginia

Generating Grasping Positions Using Models Based on Duality

by

Michael Dannhardt

Douglas Lindner, Chairman

Electrical Engineering

(ABSTRACT)

An object model based on the principle of duality where a plane has a point as a dual and a point has a plane as a dual is presented. Using this representation we present methods for performing intersection detection and face overlap checks to determine if a parallel jaw gripper is in a position to execute a valid grasp. Valid grasps are ones physically realizable and can lift an object when it is assumed that the gripping forces and the friction coefficients are large enough to keep the object from slipping. The objects considered are polyhedra and valid grip checks for both convex and concave objects are developed. A generate and test method for synthesizing valid grasp positions is also presented.

Acknowledgements

I would like to thank Dr. John Roach for presenting me with an opportunity to work with him on this part of his research. He has presented me with many intriguing ideas that will influence my thinking for years to come. I would also like to thank Dr. Doug Lindner and Praveen Paripati. Dr. Lindner offered advise on writing a technical paper and it was his welcomed criticisms that helped me with the structure and clarity of this thesis. Praveen was always willing to sit down and discuss some new concept or offer advise on a new direction to investigate. Without his help, progress would not have been as rapid or as enjoyable.

Finally, I want to thank my parents for boundless moral support and encouragement. They continue to help me more than they realize.

Table of Contents

INTRODUCTION **1**

1.1 Force-closure and Stability 3

1.2 Generating / Synthesizing Grasps 5

1.3 Dual Space Representation 10

THE DUAL REPRESENTATION **14**

THE OBJECT MODEL **22**

3.1 Recorded Information 22

3.1.1 Storing object face planes 23

3.1.2 Storing object vertices 25

3.2 Origin Location and Axis Orientation 26

DEFINITION OF A VALID GRASP **28**

4.1 The Gripper Model 28

4.2 Valid Grip Criteria 30

GRASPING CONVEX OBJECTS 34
5.1 Convexity of the Dual Model 35
5.2 Suitable Face Pairs 37
5.3 Non Zero Overlap of Parallel Faces 39
5.3.1 Perpendicular vertex projection 39
5.3.2 Point inclusion on a face 40
5.4 Pad/Face Overlap 44
5.5 Interference Detection 44
5.5.1 Interference between gripper and object 44
5.5.2 Interference between gripper and table top. 47
5.6 Convex Hulls of Concave Objects 49
5.6.1 Non simple polyhedron 49
5.6.2 Finding convex hulls in dual space 51
5.7 Summary 52

GRASPS ON CONCAVE OBJECTS 55
6.1 Suitable Face Pairs 55
6.2 Face Intersection Detection 58
6.3 Jordan Curve Theorem 60
6.4 Face Rotation 61
6.5 Face Projection 63
6.6 Point Rotation 65
6.7 Dual Edge Boundaries 66
6.8 Application of Jordan Curve Corollary in Dual Space 69
6.9 Checking for Gripper/Object Intersections 72
6.10 Summary 74

GENERATING GRASP POSITIONS 75

7.1 Finding Candidate Grasping Areas 76
7.1.1 Generating Candidate Plane Pairs 76
7.1.2 Prioritizing Candidate Pairs 77
7.1.3 Checking Face Overlap 79
7.2 Finding Valid Positions and Orientations 80
7.3 Summary 82

CONCLUSION **84**

Bibliography **87**

Vita **90**

List of Illustrations

Figure 1. An object and its dual representation. 24

Figure 2. The parallel jaw gripper and the two models used to represent it. 29

Figure 3. Face overlap. 32

Figure 4. Convex polygons have convex polygonal duals. 36

Figure 5. Parallel faces and their representation. 38

Figure 6. Perpendicular point projection in 41

Figure 7. Determining if a dual point is on a dual face. 43

Figure 8. Checking for intersections with the gripper's palm. 46

Figure 9. Checking for intersection with the table top. 48

Figure 10. A concave polygon and its non simple dual. 50

Figure 11. Finding the convex hull for a polygon from its dual representation. 53

Figure 12. A non convex object with its signed dual representation. 57

Figure 13. Checking vertex pairs of face A against plane B. 59

Figure 14. Parameters for locating the dual plane A. 62

Figure 15. Geometry of the dual face projection. 64

Figure 16. Angles defining the boundaries of a dual edge. 68

Figure 17. Examples of applying the Jordan Curve Corollary in dual space. 71

Chapter I

INTRODUCTION

In a manufacturing environment we often find machines such as turning machines, gauges, material handling systems and robots, interacting to produce some part or assemble. Usually these machines form what is typically called a cell and fall under the control of a "cell controller". The interface between a cell controller and a robot is a "black box" that translates what the cell controller request into instructions for the robot controller. As the manufacturing environment becomes more unstructured, such as the surface of a plant or the ocean floor, it is desirable that the controller specify operations in terms of their effects on objects in the environment. For example, place the part on the conveyor rather than a sequence of move points and open and close points for the robot to execute. This type of robot programming is known as task level programming. For a robot to execute these tasks it must first have the high level instructions translated into a sequence of specific actions by a task planner. A very important part of any task planner for a robotic system is grasp determination: how to choose a good grip on a target object.

The grip determination process is dependent on the type of hand doing the grasping. Robot hands are becoming more sophisticated so that the next generation of robots can perform a greater variety of tasks. These hands, such as the Stanford-JPL [1] hand or the Utah-MIT [2] hand are designed to grasp arbitrary objects and apply arbitrary forces and moments to them. To understand these new hands, much of the current work concerning gripping is in the area of analyzing multi-finger and dexterous finger grips.

Researchers have been looking at how the fingers of such hands interact with the objects they are holding by developing equations describing the forces and torques exerted by the finger tips on the objects. Finger contacts are described as hard or soft and as with friction or without friction. The independent degrees of force that can be generated by a contact depends on the type of finger contact. Forces and torques are generally dealt with by using a six element vector called a wrench, three elements for the independent torques possible and three for the three independent translational forces possible [1]. With these models, researchers can study what types of forces particular grasps can exert and develop relationships that must be satisfied for stability.

Another issue involved in the grasping problem is the development of an algorithm for generating grasp positions. This area has not received as much attention, must of the work in grasp planning being done on analyzing grasps rather than synthesizing them. This seem natural since the problems of stability and the related problem of path planning must first be well understood so that a task planner can decide what to look for as a reasonable grasping goal. We will now look at some of the research being done in the areas of grasp stability analysis and generating grasp configurations.

1.1 Force-closure and Stability

Kerr and Roth [3] developed kinematic relationships between the fingers of a generalized dexterous hand and the object it is holding. This allows them to determine where the singularities of the hand's global Jacobian matrix are and hence tell if the hand has certain configurations that can not exert the appropriate forces to make a grasp force closure. It also allows them to determine optimal directions of finger joint movement in order to avoid high joint velocities, although they do not consider the question of path planning.

The concept of force closure has received much attention with respect to multi-point grasp configurations. A force closure grip is defined as a grip that can exert an arbitrary force and moment to the object being grasped. Nguyen [4] gives a mathematical definition of a force-closure grasp in 2D. He models finger contacts as frictionless point contacts and proves that all force-closure grips can be made stable by solving for the spring stiffness coefficients representing the compliance of the particular fingers. He shows how his method can be used not only to stabilize the grasp but to place the center of compliance within a compliance polygon.

Nguyen [5] is one of the few to address the problem of actually synthesizing planner grasps that have force closure. He presented a way of representing finger contacts as combinations of a few primitive contacts, each primitive capable of exerting a singularly directed force. Given a set of edges, an algorithm returns locations for four frictionless point contacts and he then solves for the set of wrenches (forces and torques) such that any arbitrary wrench can be applied to the object. This is the dual to solving the forces and torques required to give a total twist convex of zero. A twist convex can be thought of as the compliment to the wrench convex, i.e., it describes the independent degrees of freedom an object has.

When Nguyen [6] extended the idea of planner force closure to 3D he uses the idea of contact primitives again and formulates the problem as a solution of linear inequalities such that the contact wrench matrix can exert arbitrary forces and torques on the object. He uses bounding circular disks to approximate convex faces and non convex faces by sets of overlapping circular disks and suggest methods such as Voronoi Diagrams be used to find local convex regions. He shows that planner grasps require four frictionless contacts and 3D grasps require seven frictionless contacts at a minimum to be force closure.

Nguyen [7] goes on to show that all 3D force-closure grips can be made stable by choosing the correct spring constants. He uses the same potential function to describe a grasp as in his earlier paper describing the stabilization of planner polygons. This potential function is the sum of the potential energy functions from all linear and angular springs in the hand. He defines grip stability as having the system in equilibrium, i.e. the gradient of the potential function equal to zero. As in the planner case the center of compliance can be chosen by adjusting the stiffness matrix. This is a great advantage over a mechanical remote center of compliance since it could be relocated depending on the operation to be performed with the part. For example it could be placed at the end of a peg for a peg in the hole operation.

The definition of what is a stable grasp is at times viewed differently among different researchers. Nguyen [7] by using a minimum in the sum of the potential energies of the springs in the fingers, assures a grasps will return to its equilibrium position after a small perturbation which does not cause the object to slip with respect to the finger. Cutkosky [8] uses the same criteria and calls it infinitesimal stability. The perturbation to the system is considered a disturbance to the displacements of the springs, i.e., compliant motion. Others consider the case of the object slipping relative to the gripper and the question will it return and/or remain in equilibrium. Fearing [9] uses exactly this definition for stability when he presents a method of finding constraints on the angles between polygonal edges to be grasped such that when the planner object is "grabbed" by two fingers with friction, it will rotate or slide to an equilibrium point with respect to the gripper.

When dealing with grippers that depend on distributed contact for secure grasps, such as the parallel jaw gripper, often a grasp is defined as stable if the object is fixed with respect to the hand. Jameson and Leifer [10] define a quasi-static analysis in which they assume an object is held stably, as just defined, with an external load slowly increasing until the object slips. The magnitude of the external load, as the object is sliding, gives them a margin of stability in that direction. In order to evaluate a grasp to choose an optimal one from a set, Wolter, Volz and Woo [11] develop what amount to two measures of stability. A rotational torque to cause a rotational slippage is calculated and used as a stability measure in the same manner as Jameson and Leifer [10] but they also calculate how far the grippers jaws would have to separate in order for the part to twist while gripped. This second measure is more in line with the stability criteria of perturbation rejection. Cutkosky [8] also points out that calculating how much force or torque the grip can resist before slipping is a useful measure of how "good" a grasp is although he doesn't define it as a measure of stability.

1.2 Generating / Synthesizing Grasps

Much more work has been done in the area of analyzing grasps then in to synthesizing them. As mentioned earlier, Nguyen [4] has generated regions to place fingers to make a grasp stable if given which edges to search over. Baker, Fortune, and Grosse [12] work with a hand having three fingers oriented in a ring and aimed at the center of the ring. The angle between fingers is adjustable as are the forces each finger can exert. They construct a maximal inscribed circle within the polygon to be grasped. Such a circle will touch the polygon either at two points or three. The two or three points where the circle tangantly intersects the polygon are where the fingers are to touch. He then calculates the forces the fingers must exert such that there is no translation of the polygon. Unwanted torques do not need to be considered since the line of action for all the finger forces are through a common point, the center of the circle.

Mason [13] investigates grasping objects whose position and orientation are only partially known. He does this by developing mathematical representations for the mechanics of planar objects being pushed and squeezed. Ignoring the objects inertia, he assumes all forces and torques are caused by the gripper and planner distributed friction. Through pushing and squeezing on certain features, an object can be rotated to a stable position. A stable position here means the object does not rotate with respect to the pushing gripper. After a sequence of push and squeeze operations, the object has a higher degree of certainty as to its orientation and location.

Brost [14] also considers the problem of planner grasping under bounded uncertainties in the position of the target. He develops a push-stability diagram which is object dependent. From the diagram he can tell if the squeezed object will rotate or wedge into a stable grasp position and can tell which direction the planner object will rotate when pushed. He considers the gripping surfaces to be infinite parallel lines and as such does not have to worry about the location of the gripper, only its orientation. He does not extend his method to non-convex object.

There are other considerations beside force closure and stability to the grasp determination problem. In order that a grasp configuration be safe, the gripper and the object most not collide and the configuration must be reachable. In a more realistic situation, the environment maybe cluttered and so we must also watch for collisions with the manipulator links as they move and worry not only about colliding with the target object but with the obstacles in its vicinity as well. Hence, grasp planning is related to obstacle avoidance yet there are significant differences. The goal of grasp planning is to identify a single configuration, not a path, and this configuration wants to contact an obstacle, the target object, in a specific way.

How a search is performed to find regions where the gripper is free to move is very dependent on the representation used to model the objects and gripper. Hayward [15] gives a review of possible schemes used for representing solid objects. He points out that representations such as constructive solid geometry, cell decomposition, and octree, which have primitives that are volumes, are easier to use for interference detection verses representations based on features, the latter being called boundary representations. Sweep representations are presented as having the greatest generality and being the most structured of the schemes, but he notes it is not well understood computationally. Hayward goes on to describe an efficient method for storing and sorting for an octree representation and an algorithm to do collision detection. He points out that transformations on octrees such as rotations and translations are computationally intensive because they involve re-arranging the nodes of the tree, and so the methods presented do not work well for moving robots. Herman [16] describes a collision free motion planning scheme based on an octree representation also.

As a sub problem of trying to move a gripper to a configuration while avoiding collisions is the problem of moving a point object through a cluttered field. Volpe and Khosla [17] make use of a topographical potential energy function for guidance. Regions the end effector should move to are modeled as attractive potentials (energy valleys) and areas to avoid are modeled as repulsive potentials (energy poles). Thus for each point in the real work space, there is a modeled value of potential energy and an associated gradient or force telling the end effector or point which way to move. The major problem with this concept is the presence of local minimum where the gradient gives no net force and the manipulator stops. He goes on to describe a type of elliptical potential function to help alleviate this problem.

Lozano-Perez [18] attempts to calculate free areas among obstacles where arbitrary objects can move by defining configuration space. An object in 3D has a position and a orientation and so is represented as a 6-dimensional configuration vector. The 6-dimensional space of configurations for an object is called its configuration space. Finding where objects can be moved collision free is a matter of solving unions and differences of sets and this presents significant problems in 6-dimensions. Lozano-Perez presents a method of computing what he calls slice projections. These are projections from 6-dimensional space to 3-dimensional space and the number of slices he takes depends on the tolerances he is trying to achieve.

Ghallab [19] makes the point that "inference making systems involving pattern-directed non-deterministic searches are of such high combinatorial complexity as to impede their effectiveness in real time". He suggest that the use of application and domain specific knowledge be used for pruning the searches. This is particularly true in the search of grasping positions.

In order to reduce the complexity of the grasp search, Rao, Medioni, and Bekey [20] decompose visual data and range data into a connected set of volumetric shape primitives. These primitives are generalized cones and are generated by rotating a bounded patch of the object about an inferred axis of rotation. He points out that these are difficult to compute and he presently breaks the image into primitives by hand. Grasp modes are found for each primitive via a look up table and heuristics are applied to generate a sorted list. Examples of heuristics used are to use the grasp which is on the primitive with the most volume or the grasp which is on the primitive closest to the center of mass for the object.

Feddema and Ahmad [21] present an algorithm for determining static grasps with respect to the gripper for an "n" fingered robot hand. They prune their search using a priori knowledge of the object to be grasped. He defines sets that contain object related information like "forbidden grasping regions", "faces in initial contact with another surface" and sets containing grasp related information such as "object surfaces contacted by grasp G_i". Performing set operations he calculates sets of all surfaces contacted by each possible grip. Finally each face is discretized into possible grasping points for the fingers. For each of the possibilities (number of points given number of fingers) the forces needed to hold the object in equilibrium are calculated and the minimum energy solution taken.

Laugier [22] describe a program that generates grasp positions for a parallel jaw gripper. He uses a hybrid geometrical description of the object. It includes the object's topology linking geometrical entities together with descriptions such as which side of a face matter is on, the object's geometry for metric property calculations (vertex coordinates, radius of a circular edge), and the objects physical attributes such as weight or location of the center of mass. Laugier's approach is to generate a list of all pairs of geometrical entities, faces, edges, and vertices, and then uses the list as candidate grasping features. These are passed through consecutive filters which pass or reject each pair. Two heuristics guide the "reasoning". One attempts to estimate the quality of each grasping pair not yet rejected and the other evaluates the pair's change of passing the remaining filters. The success of the program is in the use of extensive object models.

Wolter, Volz, and Woo [11] have also done work on generating gripping positions for a parallel jaw gripper. They first make the point about using a priori knowledge about the target object and the obstacle to perform as many computations off line as possible. Assuming the worst case of no a priori knowledge, they selects pairs of parallel faces facing away from each other and for each pair generate a finite set of approach vectors using an edge driven algorithm which spaces the approach vectors out uniformly and perpendicularly along each side of the face and bisect the face's vertices. Form this set, each approach vector is tested to see which ones can be achieved without having the gripper intersect the part. Approach vectors are assumed to be straight and parallel with the fingers of the jaw. They propose a projection method to reduce the problem to a two dimensional intersection problem for each pair of grippable surfaces. From the grips remaining, one is selected based on a weighted average of its resistant to rotation under an increasing torque and its resilience to twisting within the jaws of the gripper.

1.3 Dual Space Representation

As pointed out earlier, the representation used for object modeling influences the methods used in the searching for grasp configurations. The central focus of this thesis is analyzing grasps for interference detection using an object model based on the spherical dual representation.

The spherical dual representation was first presented by Roach, Paripati, and Wright [23] as a method for storing models of objects for use in a CAD system. They developed this representation by starting with the Gaussian sphere as a basis but point out that by simply storing the Gaussian image of an object a great deal of information is lost. They recognized the need to encode additional information with the Gaussian image and mention previously tried schemes such as storing the area of the face or the curvature for smooth objects as "weights" associated with the points of the Gaussian sphere. They choose as their "weight", a measure based on the inverse perpendicular distance of the face from the origin and call it the measured Gaussian image. As with other extensions to the Gaussian sphere representation, they encountered trouble in reconstructing non-convex objects. To rectify the problem, they encode more information with the model by connecting points on the Gaussian sphere by cords if the faces they represent share a concave edge. It is shown that this model exhibits many of the properties of "duality" as presented by Huffman [24]. Constructive solid geometry and primitive solids are employed so users of the system can build up complex object models.

Huffman [25] was one of the first to realize that the idea of geometric duality might be useful in interpreting object drawings. Huffman [24] showed that the information lost in a picture of a polyhedron, the orientation of the faces, is preserved in the picture of the dual polyhedron. Pictures are the projections of the three dimensional polyhedra. He presented as an application for the properties of dual space a method of determining which faces of a polyhedron would be illuminated from a light source and which would be in a shadow.

Mackworth [26] developed a program called POLY that can help interpreting the meaning of pictures of polyhedral scenes. The conceptual framework for his program is based in a large part on Huffman's idea of duality. In particular, Mackworth uses a picture where a picture is again a projection of the scene, either orthogonal or perspective, and its corresponding dual picture in conjunction, to determine how to interpret lines in the picture. Lines could represent concave or convex edges, or a crack. He then extends the program to help interpret the pictures of vertices by comparing them with their dual pictures. His program gives users addition information to help interpret a picture of a scene if it has both the scene picture and the dual scene picture as input.

Draper [27] builds on Mackworth's program by labeling all edges as being connected or non-connected. Then for each proposed labelling, it tries to construct a corresponding dual scene picture. This is done by initially drawing duals of a couple of edges and then trying to complete the dual picture without violating any of the known correspondence between dual pictures and real pictures. The initial choices of the first two edges influence the placement of the remaining dual edges. If a dual picture can be found then the proposed labelling is a valid labelling and may represent what is in the actual scene. The diagram then forms the bases for determining which edges are concave and which are convex.

Researchers in the field of vision hope that these ideas of face and edge labelling may help them interpret their visual data. Shafer [28] uses the idea of a dual representation and dual picture, also called gradient space, to interpret a line drawing from an image. His line drawings include the shadow regions cast by the faces of the object and the bulk of his work is the task of geometric analysis of these shadows. This is what Shafer calls shadow geometry. He begins his work by offering a collected set of results on gradient space properties and definitions for generalized cylinders, presenting results for both orthogonal projections and perspective projections. He then develops a theory describing the relationships among surface orientations in the drawings with shadows based on the concept of illumination surfaces. Illumination surfaces are planner and represent a boundary between an illuminated region and a dark region. Even by including the shadow information, Shafer points out that additional information such as the position of the light source is still needed if exact surface orientations are to be found.

In this thesis, definitions and theorems describing some of the main properties of duality are forwarded and we present a description of how they are used to describe an object. It will be shown that the object model used does not store the concave edge connections explicitly as does Roach's model [23] but rather stores the planes which define real object vertices. We will be working with the parallel jaw gripper and will not be concerned with multifingered dextrous grippers. We will define the real world model for what we call the parallel jaw gripper and its dual representation and present our definition of a valid grasp. Object features for grasping exclude vertices and edges, e.g., grasping of an object by a vertex and a face is not allowed. We consider only parallel face pairs. We assume that the coefficient of friction is high enough and the normal forces exerted large enough to be able to pick up an object with out it slipping. Hence we are not concerned with stability and we will not be considering the dynamics involved with grasp configurations.

Having defined the valid grasp criteria for this thesis, we look at the problem of analyzing a grasp configuration. A grasp configuration is the gripper model and the object model together, with the gripper in a particular position with respect to the object and is used to determine if it is valid or not. Convex objects are considered separately from concave objects because grasp configurations are easier to analyze on convex objects. This leads us to develop methods of finding the convex hulls of non convex objects to take advantage of this fact. The analyzing for validity and the methods for finding the convex hull are performed in the dual domain so to minimize object reconstruction.

We will present methods and heuristics for searching out safe gripper locations in the dual domain. As in other algorithms that use boundary representations for models, our method is a generate and test using discrete points on the faces for grasping points. Since we assume no prior knowledge about the object, these points are generated by the algorithm.

The thesis is organized as follows: Chapter 2 contains definitions and theorems concerning dual space. Chapter 3 presents the method in which we use the duality concept to model an object. In Chapter 4 we describe the parallel jaw gripper model and the definition of a valid grasp. Chapter 5 is concerned with analyzing grasps on convex objects to see if the grasp meets the valid grasp criteria presented in Chapter 4. A method of constructing the convex hull of a non convex object directly in dual space is also described. Chapter 6 is concerned with the same grasp validity checking but on non convex objects. In Chapter 7 a method of generating grasping configurations is presented which uses the work of Chapters 5 and 6 for testing whether the generated grasp is valid. Finally, the conclusion summarizes the must important results.

Chapter II

THE DUAL REPRESENTATION

In this chapter we introduce the concept of duality. We will present some basic definitions and theorems as well as some important terminology to be used throughout the remainder of this thesis. While most of the work presented here involves a three dimensional space we will often find it convenient to present ideas in two dimensions and so we will present a few of the properties of duality for the degenerate case of two dimensions.

The principle of duality in geometry arises from the interchangeability of points and lines in axioms of plane geometry. According to this principle, every theorem has a dual theorem and every figure a dual figure attainable by interchanging the words point and line in two dimensions or point and plane in three dimensions [29]. In the three dimensional real space R^3 with coordinate axises $\hat{x}$,$\hat{y}$ and $\hat{z}$ forming a right handed system, the equation of a plane can be described by:

$$\mathrm{A}x + \mathrm{B}y + \mathrm{C}z = \mathrm{D}. \tag{2.1}$$

There are three degrees of freedom so (2.1) can be "normalized". Huffman[24] chooses to normalize (2.1) with respect to the $\hat{z}$ axis as:

$$z = \mathrm{a}x + \mathrm{b}y + \mathrm{c} \tag{2.2}$$

This normalization does not allow representation of planes parallel to the $\hat{z}$ axis, a fact not very desirable in a CAD system. Alternately, we normalize (2.1) as:

$$\mathrm{a}x + \mathrm{b}y + \mathrm{c}z = 1. \tag{2.3}$$

This normalization allows planes parallel to the $\hat{z}$ axis to be represented but disallows representation of planes passing through the origin. This restriction is a less severe one then using Huffman's normalization since the origin can always be relocated.

Consider R^3 as the usual Euclidean space. Let (P,R^3) be the set of planes in R^3. Let D^3 be a copy of R^3 with orthogonal coordinate system having $\hat{u}$, $\hat{v}$, and $\hat{w}$ as unit direction vectors.

Definition (2.3.1): Let the mapping $f\colon (\mathrm{P},R^3) \rightarrow D^3$ be given by $f\colon (\, \mathrm{a}x + \mathrm{b}y + \mathrm{c}z = 1\,) \rightarrow (\mathrm{a,b,c})$. The image of f is called here three dimensional *dual space* . In the degenerate two dimensional case, the mapping $g : (\mathrm{P},R^2) \rightarrow D^2$ is given by $g : (\, \mathrm{a}x + \mathrm{b}y = 1\,) \rightarrow (\mathrm{a,b})$ and the image of g is two dimensional *dual space* .

Remark (2.3.2): The dual to a plane P: $\mathrm{a}x + \mathrm{b}y + \mathrm{c}z = 1$ is a point $\mathrm{P}' = (\mathrm{a,b,c})$ and is located in dual space by a position vector $\vec{\mathrm{P}}' = \mathrm{a}\hat{u} + \mathrm{b}\hat{v} + \mathrm{c}\hat{w}$. In two dimensions, a line L: $\mathrm{a}x + \mathrm{b}y = 1$ has a dual which is a point $\mathrm{L}' = (\mathrm{a,b})$ located by the position vector $\vec{\mathrm{L}}' = \mathrm{a}\hat{u} + \mathrm{b}\hat{v}$.

Remark (2.3.3): The dual to a line in three dimensions is also a line. A line in three dimensional real space is defined by the intersection of two planes. The dual to that line is defined in dual space as the line passing through the two dual planes.

Remark (2.3.4): The dual to a point in three dimensional real space is a plane in dual space. A point being the intersect of three planes has a dual which is a plane defined by the three dual planes. In two dimensions, a point is the intersection of two lines. The two dimensional dual is a line passing through the two dual points.

With definition (2.3.1) and remarks (2.3.2) through (2.3.4) we see that polygons in R^2 map to polygons in D^2 . Likewise, polyhedra in R^3 map to polyhedra in D^3. In the following as throughout this thesis, the duals to geometric features such as points, lines, and planes as well as bounded planes, i.e., polygons and faces will be indicated by a prime.

Note, when we speak of the "dual plane", we are referring to a point in dual space D^3 that represents the plane.

These definitions and remarks show the complete symmetry between dual space and real space: planes have duals that are points, lines have lines as duals and points have planes as duals. Note that the two dimensional cases are obtained by simply setting the z and w values to zero.

In this thesis we will often consider whether a position vector to a dual plane "intersects" a plane. We do this by considering a position vector as a line segment $\vec{P}'$ from the origin to the point it locates, P'. $\vec{P}'$ intersects a plane if P' is on one side of the plane and the origin is on the other. This can be tested as follows. If the plane is given by: $Xu + Yv + Zw - 1 = 0$, and the dual plane by: (A,B,C), then $XA + YB + ZC - 1 = c$, where $c \leq 0$ if there is no intersection and $c > 0$ if the position vector does intersect the plane.

Since dual space is defined by an orthogonal coordinate system similar to real space, all the vector properties that hold in R^3 hold in D^3. Often we will find it convenient to "overlay" the two coordinate systems. By "overlay" we mean place the dual and real origins at a common point and align the x and u , y and v and z and w axises. We now present some useful properties relating dual space D^3 and real space R^3.

Theorem (2.3.5):[30] The angle between two planes in real space R^3, is equal to the angle between the position vectors locating the dual planes in dual space D^3 .

Proof: Let two planes be given by:

$$P_1:\ A_1x + B_1y + C_1z = D_1 \tag{2.4}$$

and

$$P_2:\ A_2x + B_2y + C_2z = D_2 \tag{2.5}$$

with normals

$$\vec{n}_1 = < A_1, B_1, C_1 > \tag{2.6}$$

and

$$\vec{n}_2 = < A_2, B_2, C_2 > . \tag{2.7}$$

Their duals are:

$$\vec{P'}_1 = < \frac{A_1}{D_1}, \frac{B_1}{D_1}, \frac{C_1}{D_1} > \tag{2.8}$$

and

$$\vec{P}'_2 = < \frac{A_2}{D_2}, \frac{B_2}{D_2}, \frac{C_2}{D_2} > \tag{2.9}$$

respectively. The angle between the planes is found from the dot product of their normals,

$$\vec{n}_1 \bullet \vec{n}_2 = |\vec{n}_1|\,|\vec{n}_2| \cos\theta \tag{2.10}$$

hence,

$$\theta = \cos^{-1}\left\{ \frac{\vec{n}_1 \bullet \vec{n}_2}{|\vec{n}_1|\,|\vec{n}_2|} \right\} \tag{2.11}$$

$$= \cos^{-1}\left\{ \frac{A_1A_2 + B_1B_2 + C_1C_2}{\sqrt{A_1^2 + B_1^2 + C_1^2}\,\sqrt{A_2^2 + B_2^2 + C_2^2}} \right\}$$

The angle between $\vec{P}'_1$ and $\vec{P}'_2$ in R^3 can be derived from:

$$\vec{P}'_1 \bullet \vec{P}'_2 = |\vec{P}'_1|\,|\vec{P}'_2| \cos\beta \tag{2.12}$$

so,

$$\beta = \cos^{-1}\left\{ \frac{\vec{P}'_1 \bullet \vec{P}'_2}{|\vec{P}'_1|\,|\vec{P}'_2|} \right\} \tag{2.13}$$

$$= \cos^{-1}\left\{ \frac{\frac{1}{D_1D_2}(A_1A_2 + B_1B_2 + C_1C_2)}{\frac{1}{D_1}\sqrt{A_1^2 + B_1^2 + C_1^2}\;\frac{1}{D_2}\sqrt{A_2^2 + B_2^2 + C_2^2}} \right\}$$

$$= \theta$$

Corollary (2.3.6): Parallel planes in real space are represented in dual space by points on a common line through the dual origin.

Theorem (2.3.7):[30] The perpendicular distance from the origin to a plane in real space is:

$$\frac{D}{\sqrt{A^2 + B^2 + C^2}} \tag{2.14}$$

Proof: Some scalar s times the unit normal to the plane, $\hat{n}$, such that the point q described by $s \bullet \hat{n}$ is a point on the plane will be this distance. The unit normal to the plane of (2.1) is:

$$\hat{n} = \frac{A}{\sqrt{A^2 + B^2 + C^2}}\hat{i} + \frac{B}{\sqrt{A^2 + B^2 + C^2}}\hat{j} + \frac{C}{\sqrt{A^2 + B^2 + C^2}}\hat{k}. \tag{2.15}$$

Substituting the point located by $s \bullet \hat{n}$ into (2.1) and solving for s:

$$A\frac{sA}{\sqrt{A^2 + B^2 + C^2}} + B\frac{sB}{\sqrt{A^2 + B^2 + C^2}} + C\frac{sC}{\sqrt{A^2 + B^2 + C^2}} = D \tag{2.16}$$

$$\frac{s}{\sqrt{A^2 + B^2 + C^2}}(A^2 + B^2 + C^2) = D \tag{2.17}$$

$$s = \frac{D}{\sqrt{A^2 + B^2 + C^2}}. \tag{2.18}$$

Corollary (2.3.8): The distance to a dual plane from the dual origin is equal to the inverse perpendicular distance of the plane it represents to the origin in real space.

Theorem (2.3.9):[30] In three dimensions a line in real space is perpendicular to its dual line when the real and dual coordinate systems are overlayed.

Proof: Let the intersecting planes be P_1 and P_2 given by: $a_1x + b_1y + c_1z = 1$ and $a_2x + b_2y + c_2z = 1$ respectively. The intersection of the planes gives a line in the direction:

$$\vec{n_l} = < a_1, b_1, c_1 > \cdot < a_2, b_2, c_2 > \tag{2.19}$$

$$= < b_1c_2 - c_1b_2, c_1a_2 - a_1c_2, a_1b_2 - a_2b_1 > .$$

The dual planes are located by: $\vec{P_1}' = < a_1, b_1, c_1 >$ and $\vec{P_2}' = < a_2, b_2, c_2 >$ and the line between these two points has the direction: $\vec{n_l}' = < a_1 - a_2, b_1 - b_2, c_1 - c_2 >$. The angle between the direction vector of the line and the direction vector of the dual line can be found from their dot product.

$$n_l \cdot n_l' \tag{2.20}$$

$$= < b_1c_2 - c_1b_2, c_1a_2 - a_1c_2, a_1b_2 - b_1a_2 > \bullet < a_1 - a_2, b_1 - b_2, c_1 - c_2 >$$

$$= a_1b_1c_1 - a_1b_2c_1 - a_2b_1c_2 + a_2b_2c_1 + a_2b_1c_1 - a_1b_1c_2 -$$

$$a_2b_2c_1 + a_1b_2c_2 + a_1b_2c_2 - a_2b_1c_1 - a_1b_2c_2 + a_2b_1c_2$$

$$= 0$$

Since the dot product is zero the lines are perpendicular.

Chapter III

THE OBJECT MODEL

In this chapter we introduce our geometric model for objects based on the dual principles presented in Chapter 2. Since these properties deal with planes and intersections of planes, the only objects that can be represented are polyhedra. Objects with non-planar surfaces must first be approximated with facets, a common technique in graphics and CAD systems. It is assumed that these models are given to us and do not need to be generated.

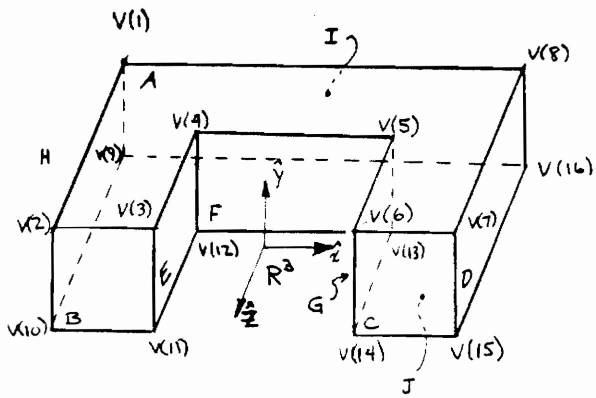
3.1 Recorded Information

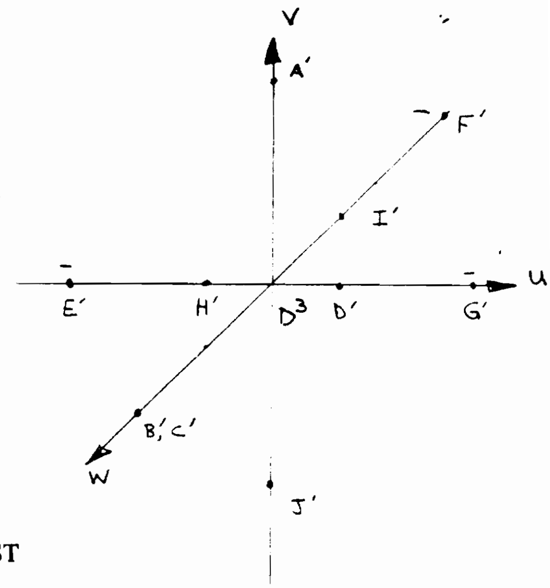
The object model we will use to check the validity of a grasp configuration is comprised of two tables. The first table is a list of the dual planes that make up the object and the second table is a list of the vertices of the object. The table of the dual planes is simply a list of points in dual space. The vertex table has an entry for each vertex of the object. Each entry contains a list of all the face planes comprising that vertex.

3.1.1 Storing object face planes

Storing only the dual representation of the face plane provides information about the face's orientation and position relative to the origin but does not give an indication as to which side of the plane is the outward side of the face. This information is needed because two parallel faces facing in the same direction cannot be grasped. To solve this ambiguity the dual planes have a positive or negative sign attached to them. Positive if the outward normal of the face points away from the origin and negative if it points toward the origin. Faces E, F, and G in Figure 1 have negative values attached to their duals while the remaining faces have positive values attached to the duals of their planes. Example, face E would be stored as: $E' = (e_x , e_y , e_z , -)$ where e_x, e_y, and e_z are the x, y, and z coefficients of the plane containing E and normalized by our convention.

Actually, storing the coefficients of the normalized plane and the sign could more compactly be stored as a unit vector in spherical coordinates indicating the normal direction of the face plane with a weight assigned equal to $\frac{1}{d}$ where d is the perpendicular distance to the plane. The weight is given a positive or negative sign by the convention described above. In Figure 1, E would be stored as $E' = (\alpha , \beta , -\frac{1}{d_E})$ where α and β are the point's spherical coordinates and the d_E is the perpendicular distance of the plane containing face E from the origin. This is what is known as the spherical dual representation since the dual planes are represented as points on a unit sphere





FACE LIST

A	$(0, a_y, 0, +)$
B	$(0, 0, b_z, +)$
C	$(0, 0, c_z, +)$
D	$(d_x, 0, 0, +)$
E	$(e_x, 0, 0, -)$
F	$(0, 0, f_z, -)$
G	$(g_x, 0, 0, +)$
H	$(h_x, 0, 0, +)$
I	$(0, 0, i_z, +)$

VERTEX LIST

$v(1)$	A,I,H
$v(2)$	A,H,B
$v(3)$	A,B,E
$v(4)$	A,E,F
$v(5)$	A,F,G
$v(6)$	A,G,C
$v(7)$	A,C,D
$v(8)$	A,D,I
$v(9)$	J,I,H
$v(10)$	J,H,B
$v(11)$	J,B,E
$v(12)$	J,E,F
$v(13)$	J,F,G
$v(14)$	J,G,C
$v(15)$	J,C,D
$v(16)$	J,D,I

Note: B and C are in the same plane so $b_z = c_z$ but B and C are stored as seperate dual faces.

Figure 1. An object and its dual representation.

[23]. We will not use this representation in this thesis perferring instead to use the signed rectangular coordinate method of the previous paragraph.

3.1.2 Storing object vertices

The table of vertices has an entry for each vertex of an object. Each entry in this table stores which planes intersect to form that vertex. Knowning these planes we can calculate the actual co-ordinates of each vertex in R^3 from which we get the coefficeints for each dual vertex.

Dualism as described here is symmetrical; a plane maps to a point, a line to a line, and a point to a plane. So a point being the common intersection of three or more planes is represented as a plane in the dual domain. The plane being defined by the three or more points representing the planes. With this in mind, it may seem at first glance that storing the vertex information is redundant since coplanar points in the dual domain define a plane that in real space represents the point of common intersection for the planes, i.e., a vertex, represented by the coplanar points. In fact the coplanar points in dual space represent planes that would, but do not *necessarily* intersect at a common point. An example is a pyramid with its top cut off. Figure 1 gives the complete vertex listing for the object shown.

From the vertex information all the equations of the lines in which the object's edges lie can be calculated. If any vertex contains at least two planes in common with another vertex, then those planes must intersect to form an edge of the object. In the example of Figure 1, vertices v(1) and v(2) share faces A and H, hence we conclude there is an edge from v(1) to v(2), between H and A. The equation of an edge line is found from solving for the intersections of the common planes.

Points in the dual domain model are allowed to be equal. This is necessary because a distinction must be made between faces that lie in a common plane. B and C in Figure 1 represent an example of two distinct faces in a common plane. This underlines the fact that although the information stored in the models are duals of planes, the addition of positive or negative signs and separate storage of faces in the same plane implies we are storing the duals of faces, that is, bounded planes.

From this model of the object we often will draw a *dual object* to help visualize certian properties of the model. A dual object in D^3 is drawn by drawing the dual planes on the coordinate system and including all dual edges between the appropriate dual faces. These dual edges then form faces on a dual object which represent the vertices of the original object. In two dimensions a dual polygon is drawn by connecting the daul edges if they represent adjacent edges of the original polygon. These line segments represent the vertices of the original polygon.

3.2 Origin Location and Axis Orientation

The normalization we have chosen prohibits placing the origin on a plane containing a face. Barring this restriction, the same information will be in the model no matter from where the dualization is performed, so the choice of origin becomes a matter of convenience. As will be shown there are advantages to having the origin inside the object in that the duals are better behaved. Convex objects will have convex duals and concave objects will have duals which contain a distinct convex portion representing the convex hull of the object. Mostly this helps with the visualization of the problem of grasping in the dual domain. To this end the duals used here are taken from the centroid of the objects. This guarantees that the origin will be within at least the convex hull of the object. It also has the advantage that it locates the center of mass in the model and would be useful in calculating dynamic properties of a grasp in order to give a qualitative measure of the tendency of the object to slip or rotate. We are not concerned with calculating these dynamic properties in this thesis.

Once the origin has been chosen, an orientation for the real space coordinate axis system needs to be specified. We only require that the y-axis be perpendicular to the face on which the object rests. The positive $\hat{y}$ direction points away from the table top. This orientation allows us to detect where the table top is from the dual model of the object and to detect gripper positions that collide with it.

Chapter IV

DEFINITION OF A VALID GRASP

Having presented the representation used for modelling objects, we now describe what we are calling the parallel jaw gripper and the method by which we model it. Next we consider the gripper model in union with the object model and we present the conditions necessary for this union to represent a valid grasp.

4.1 The Gripper Model

Figure 2 shows the model of the parallel jaw gripper we will be considering here. It is characterized by a maximum finger opening, palm depth, and exterior dimensions. The maximum finger opening is a parameter of the gripper which will be considered often. It is called the

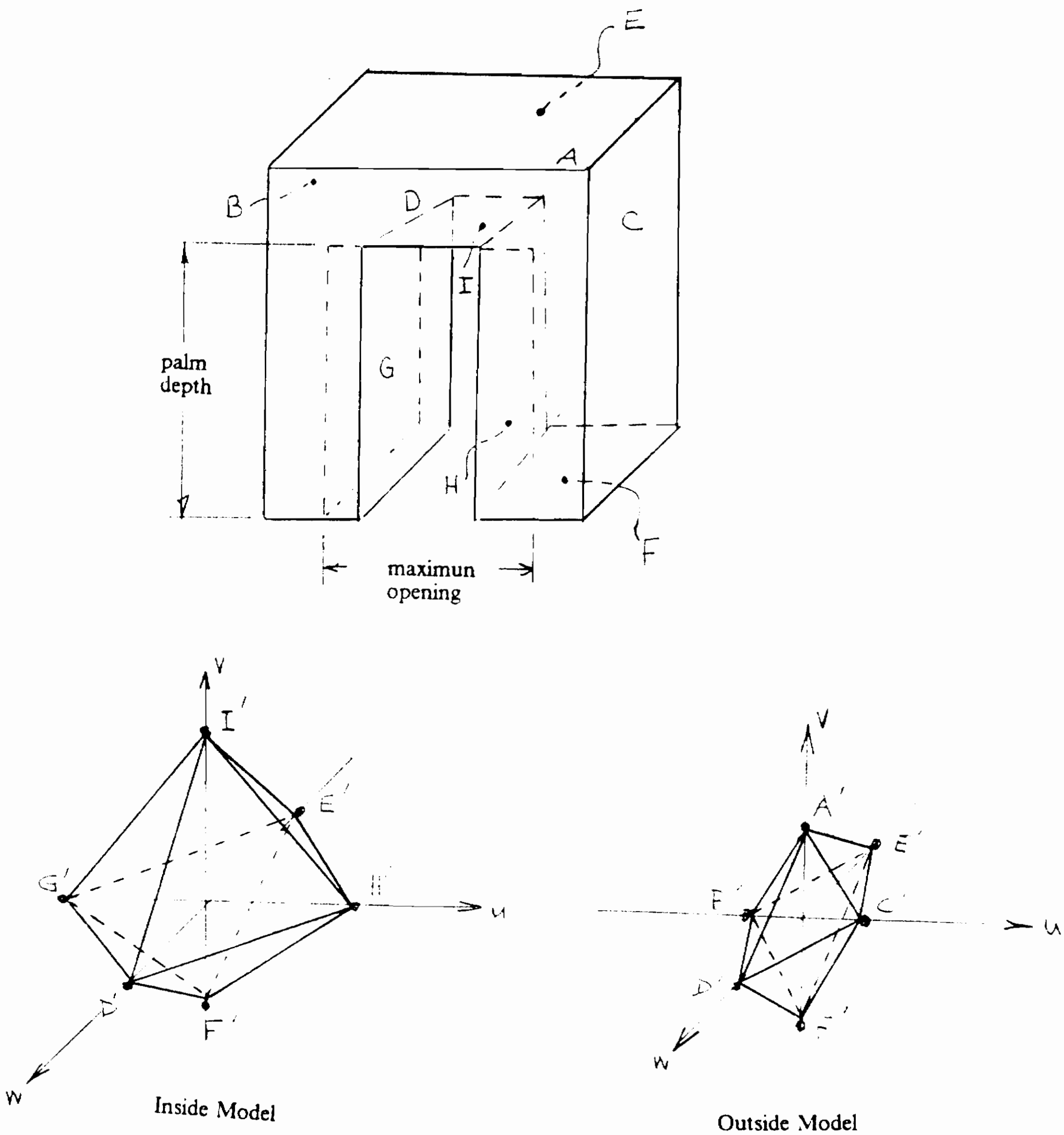


Figure 2. The parallel jaw gripper and the two models used to represent it.

grip width tolerance . The fingers are idealized in that they close parallel to each other and do not trace an arc as they close. In applying the valid grip criteria describe below, we will consider the swept volume of the gripper as the fingers move from fully open to their grip position.

We will use two dual models to represent the gripper. One is a dualization of the "inside" features of the gripper, i.e., the gripping pads and the gripper's palm. This model is used for checking pad/object contact and for clearance between the object and the palm. The second dual is of the "outside" features of the gripper and is used to check for unwanted intersections (collisions) with the object. The reason for using separate models is that the outer surfaces of the gripper are involved only in interference checks and only on concave objects or simple obstacle collision detection. Additionally, the inside model changes in terms of metric properties depending on the finger separation while the outside model remains constant. It is in this manner that we automatically take into account the swept volume of the fingers mentioned above. It will be clear in the discussions to follow which model is used for the test under consideration.

4.2 *Valid Grip Criteria*

The following three conditions define what is considered a good grip and are concerned with object or face properties and relationships between the object and the gripper in the grip position. In addition to these three conditions we consider a simple obstacle avoidance case as a forth condition for a good grip.

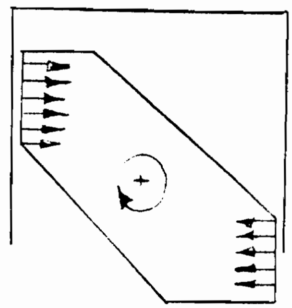
1. Object faces suitable for gripping are considered to be pairs of parallel planes, facing away from each other, and separated by less then the maximum opening of the hand. Hence, only exterior grips are studied as opposed to interior grasps such as grasping the inside of a hole by expanding the gripper's fingers.

2. Parallel faces must also have nonzero overlap. Nonzero overlap means that when the outline of one face is projected onto the outline of another parallel face, there is a nonzero intersection of area. This is required so as the gripper closes, moments will not be generated that would cause the object to rotate. Figure 3 demonstrates this idea.

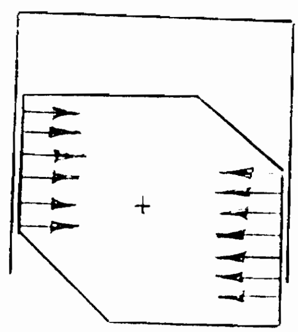
3. There must be an intersection of the gripper's pads with the intended faces of the object but there can be no other intersection of the gripper with the object. For a good grip, the only contact between the gripper and the object allowed is between the gripper's gripping pads or fingers and the intended faces. This is a non interference criteria which keeps the gripper from colliding with the object.

4. As well as identifying unwanted intersections of the gripper with the object there is the concern of intersecting other objects in the vicinity of the target object. The table top the object part rests on is an obstacle and will be the only one considered in this thesis. The methods to be described for identifying unwanted intersections between the target object and the gripper could easily be applied to avoid other obstacles if each obstacle is considered individually with the gripper.

We intend to identify whether a grip position is valid in terms of the above four criteria directly from the dual model with the dual of the gripper superimposed in the position to be



Zero Overlap



Non Zero Overlap

Figure 3. Face overlap.

checked. Any reconstruction of the object from the model should be keep to a minimum. We want to perform all computations in the dual domain directly.

Chapter V

GRASPING CONVEX OBJECTS

In the next two chapters we analyze a grasp configuration in terms of the criteria presented in Chapter 4 to determine if it is valid. Convex objects are considered in this chapter separately from concave objects. This is because convex objects are more easily checked for criteria 3 which is concerned with the non interference between the gripper and the object. We begin by proving the convexity of the dual to a convex object and use this property as we consider each criteria of Chapter 4 in turn. In the last section of this chapter we present a method of calculating the convex hull of a non convex object. This method allows us to use the results developed in this chapter to check certain types of grasp configurations on non convex objects.

5.1 *Convexity of the Dual Model*

Convex polyhedra have the property that any plane containing a face will divide space into empty space and the space containing the polyhedron. This guarantees that no object edges or vertices, other than the ones belonging to the face in that plane, will intersect the plane. This fact greatly simplifies the grasping of this class of objects. In the dual domain the following fact is also of great utility.

THEOREM (5.1.1): A convex polygon in R^2, enclosing the origin has a dual that is itself a convex polygon.

PROOF: Refer to Figure 4. Assume the dual polygon is non convex and the concavity is at the e_i vertex. A line through e'_{i-1} and e'_{i+1} of the dual polygon represents the intersection of edges e_{i-1} and e_{i+1} belonging to the real polygon. Since the line from e'_{i-1} to e'_{i+1} does not intersect the position vector $\vec{e}'_i$, it represents an intersection of those two edges closer to the origin then the edge e_i and so the original polygon is a non simple and hence non convex.

Hence all convex polyhedra, having convex polygons for faces, have convex duals. These convex duals divide dual space into two regions. Since the origin of dual space corresponds to infinity it is known that the interior of the dual polyhedron represents all free space while the exterior of the dual polyhedron represents the volume of the real object. Hence, for convex objects containing the origin, the finite volume of the object maps to an infinite volume in the dual domain and the infinite free space maps to a finite volume. This defines the dual volume in which the gripper can be located.

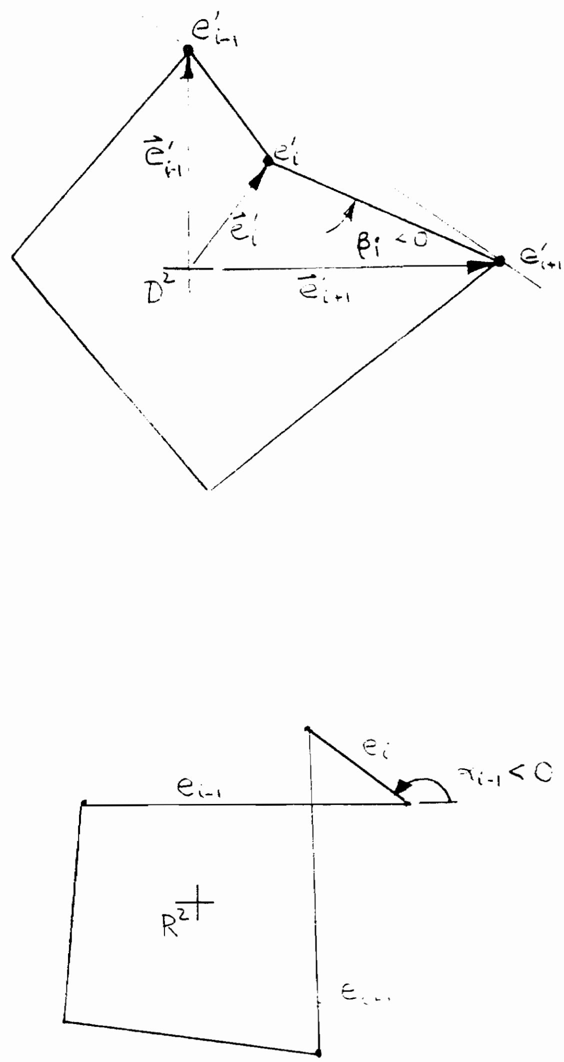


Figure 4. Convex polygons have convex polygonal duals.

To search for a valid grasp the dual object is analyzed to find suitable faces for grasping, then the gripper is dualized in the frame to be checked and this gripper dual is compared with the object dual. In the dual domain the face pair of the object to be gripped is checked for nonzero overlap and checks are made to detect unwanted collisions between the object and the gripper and between the gripper and the table top.

5.2 *Suitable Face Pairs*

It is required that faces suitable for grasping lie in planes parallel to each other and close enough together to fit within the gripping pads of the gripper when it is fully open. Parallel faces are identified on the object model as points on a common line through the origin. Since the dual model is itself convex any points found will lie on either side of the origin and represent faces with opposite pointing normals. The absolute inverse distances to the two points are checked to insure they are faces spaced closely enough together to be grasped. Hence two face planes represented by $P'_1 = (A_1, B_1, C_1)$ and the other by $P'_2 = (A_2, B_2, C_2)$ located by $\vec{P}'_1$ and $\vec{P}'_2$, will be suitable if:

$$s\vec{P}'_1 = \vec{P}'_2 \tag{5.2.1}$$

for some scalar $s < 0$ and,

$$\frac{1}{|\vec{P}'_1|} + \frac{1}{|\vec{P}'_2|} < \text{grip width tolerance.} \tag{5.2.2}$$

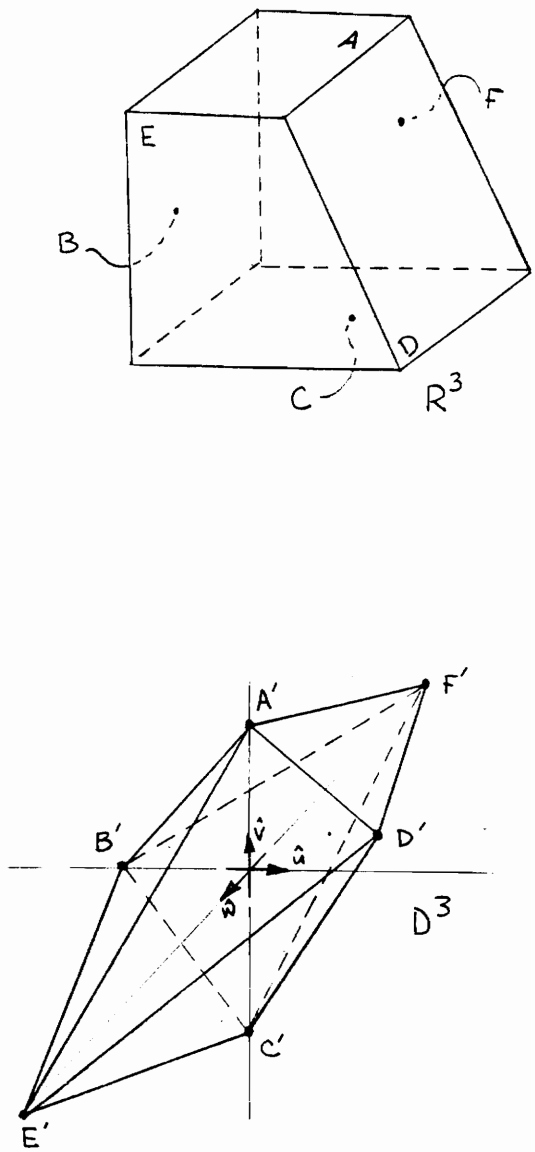


Figure 5. Parallel faces and their representation.

In Figure 5, dual face pairs A′C′ and F′E′ lie on common lines through the origin and so represent parallel pairs of faces. Pair F′E′ represent a pair of faces closer to each other than face pair A′C′.

5.3 *Non Zero Overlap of Parallel Faces*

As well as the above requirement, suitable face pairs must have non-zero overlap for reasons mentioned earlier. To check this condition, the vertices of one face are perpendicularly projected onto the parallel face plane. These projected vertices are tested to see if they lie on this parallel *face* and then the other face's vertices are projected in the same manner onto the first face plane for the same test. If such a vertex can be found, then the faces have non zero overlap. However, this is a sufficient but not necessary condition for overlap. If no such vertex is found here, then a method to be presented in Chapter 7 will have to be used to check for overlap.

5.3.1 Perpendicular vertex projection

To find the location of the projected vertex, $\vec{V}_p$, let the vertex to be tested be located by $\vec{V}$, and the face plane be P: $ax + by + cz = 1$, with normal $\vec{n}_p = < a,b,c >$. Referring to Figure 6, two conditions must be satisfied:

$$\vec{n}_p \cdot \vec{V}_p = 1 \tag{5.3.1}$$

guarantees that V_p is on P and,

$$\vec{V}_p = \vec{V} + s\,\vec{n}_p. \tag{5.3.2}$$

guarantees V_p is an orthogonal projection of V onto P. Substituting,

$$\vec{n}_p \bullet < \vec{V} + s\,\vec{n}_p > = 1 \tag{5.3.3}$$

$$\vec{n}_p \bullet \vec{V} = 1 - s\,\vec{n}_p \bullet \vec{n}_p \tag{5.3.4}$$

$$s = \frac{1 - \vec{n}_p \bullet \vec{V}}{|\vec{n}_p|^2} \tag{5.3.5}$$

and so $\vec{V}_p$ can be calculated.

The normal $\vec{n}_p$ is obtained directly from the dual model as equivalent to $\vec{P}'$ and $\vec{V}$ is obtained from the equation of its dual, V′. The method for determining if a point is on a face is presented in the following paragraphs.

5.3.2 Point inclusion on a face

Checking to see if a point lies on the faces of a convex object in dual space is accomplished quite easily. The following theorem will be used.

Theorem (5.3.1): Let Q be a point in R^3 and P_k be the k^{th} face plane of a convex object containing the origin. If a dual point Q′ is incident on a dual plane P'_k of an N faced convex object's model and Q′ does not intersect the dual object model, i.e.,

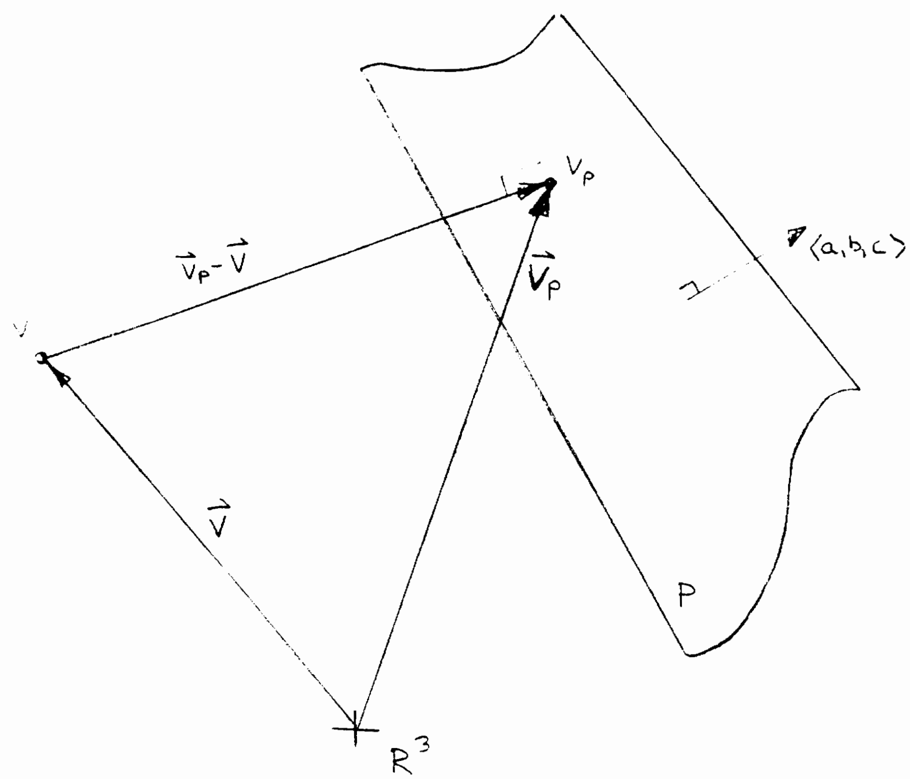


Figure 6. Perpendicular point projection in R .

$$\{ \vec{P'}_i \mid i = 1 \text{ to } N \} \cap Q = P'_i \quad i = k \tag{5.3.6}$$

and

$$\{ \vec{P'}_i \mid i = 1 \text{ to } N \} \cap Q = 0 \quad i \neq k$$

then the point Q is on the face k which lies in plane P_k.

Proof: If a dual point passes through a dual plane then the point is on the plane as follows: let the point be $Q = (q_x, q_y, q_z)$ and the plane be P: $Ax + By + Cz = 1$ so the duals are Q′: $q_x u + q_y v + q_z w = 1$ and $P' = (A,B,C)$ respectively. If Q′ passes through P′ then $q_x A + q_y B + q_z C = 1$ or $Aq_x + Bq_y + Cq_z = 1$ which means Q is on P, a necessary condition for Q to be on the face contained on P. Referring to Figure 7, as Q moves around on P the plane Q′ changes orientations in the dual domain. If Q moved to a face edge it would be on the face plane P and a bounding edge plane, P_e of the object. Q′ would lie on the corresponding dual edge intersecting both P′ and P'_e. If the Q moved over the edge, off the face to a point R, it would lie on a hypothetical plane P‘, ε away from and parallel to P_e. Q′ would now intersect $\vec{P'}_e$ at a distance $\dfrac{1}{(d+\varepsilon)}$ where $d = \dfrac{1}{|\vec{P'}_e|}$. Since P'_e is a vertex in the dual domain, Q′, would be intersecting the dual object.

So, to check if a point is on a convex object's face, we simply check that the dual point passes through the face's dual plane and none of the position vectors to the other dual planes of the object intersect the dual point. To check if a position vector intersects a dual point, we use

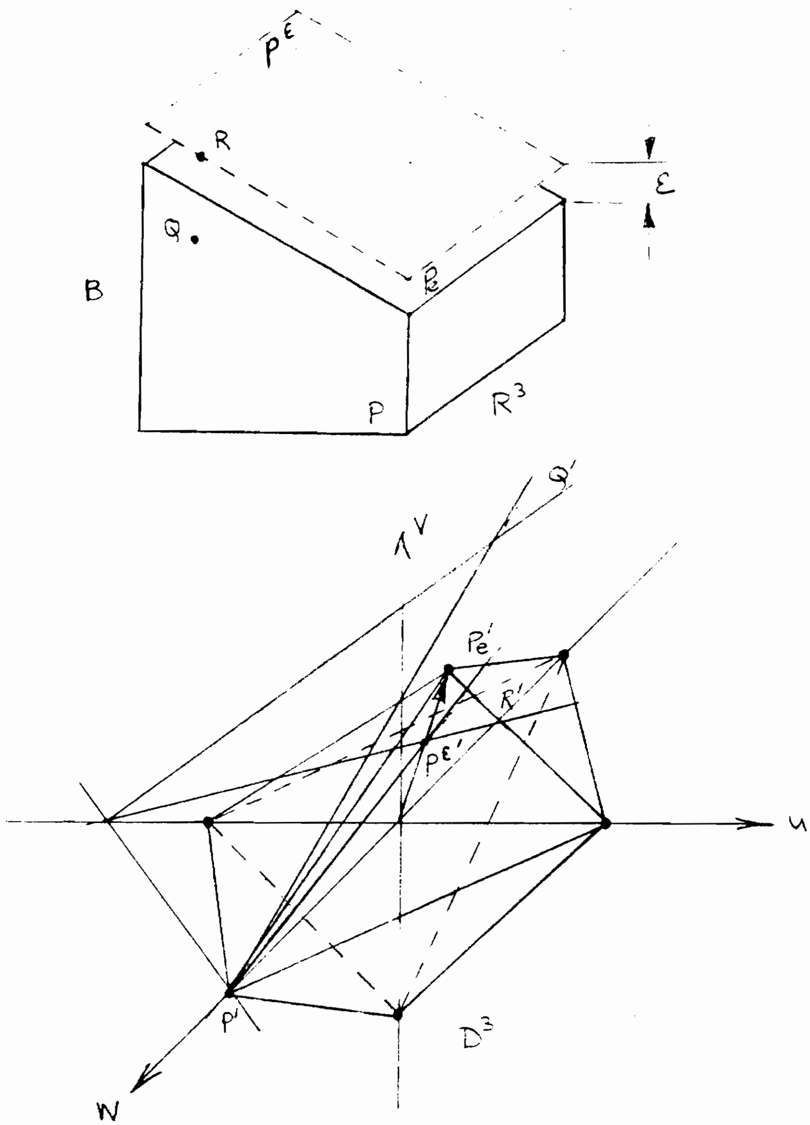


Figure 7. Determining if a dual point is on a dual face.

the method described in Chapter 2. With the ability to check if a point (vertex) is included on a *face,* we can perform the check to guarantee non-zero overlap and to solve the following problem.

5.4 *Pad/Face Overlap*

When the gripper is dualized in a frame such that it's gripping pads lie in the same plane as the intended faces to be grasped, the duals of the pads' planes and the faces' planes will be located by equal position vectors. This does not guarantee that the pads will lie on the intended faces. Nonzero overlap between the pads and the faces must be checked to insure proper intersection. For a pad face to intersect an object face we require at least one vertex of the pad to lie on the object face and/or at least one face vertex will lie on the pad face. Now it can easily be checked whether either of the bottom vertices of a gripper pad lie on a face or if one or more face vertices lie on the gripper pad by using the method presented in section 5.3.2. Both pads of the gripper must be checked to insure they both properly overlap on the intended faces.

5.5 *Interference Detection*

5.5.1 Interference between gripper and object

So far we have checked that the gripper's pads are in a legal position for grasping the object, that is the gripper's pads lie on a pair of parallel, non-zero overlapping faces. We now must check for unwanted intersections between the gripper and the object. It should be noted that only the inside model of the gripper, i.e. the model of the gripping surfaces and palm, were used for the pad/face overlap test. This is also the case for the following discussion on interference detection. A special case of obstacle avoidance is also presented at the end of this section which will make use of the outside model of the gripper.

There is only one unwanted intersection between a convex object and the gripper that can occur when the gripper's pads are on a pair of the object's faces. It is an intersection between the palm of the gripper and the object. A necessary condition for the object to intersect the palm is to have at least one of its vertices on the outward side (wrist side) of the palm plane. If an object vertex were on the outward side of the palm plane then a hypothetical plane through the vertex and parallel to the palm plane would be further from the origin than the palm plane. This condition can be checked in the dual domain. If the position vector to the dual plane containing the palm does not intersect any of the dual vertices of a convex object then this necessary condition for an object/palm intersection is not met and no further check is required to know that the palm is clear of the object. An example of this case is shown in Figure 8a.

While a dual vertex intersected by the position vector of the dual palm plane as in Figure 8b, is a necessary condition, it is not a sufficient condition to guarantee a palm object intersection. If a vertex is on the outward side of the palm plane then any connected vertices to that vertex are checked to see if they lie on the origin side (inward side) of the palm plane. A vertex on the origin side of the palm plane is indicated by a dual that is not intersected by the position vector to the dual palm plane. If such a vertex is found then the connecting edge intersects the palm plane but may or may not intersect the palm face. To check to see if the edge intersects the palm, the

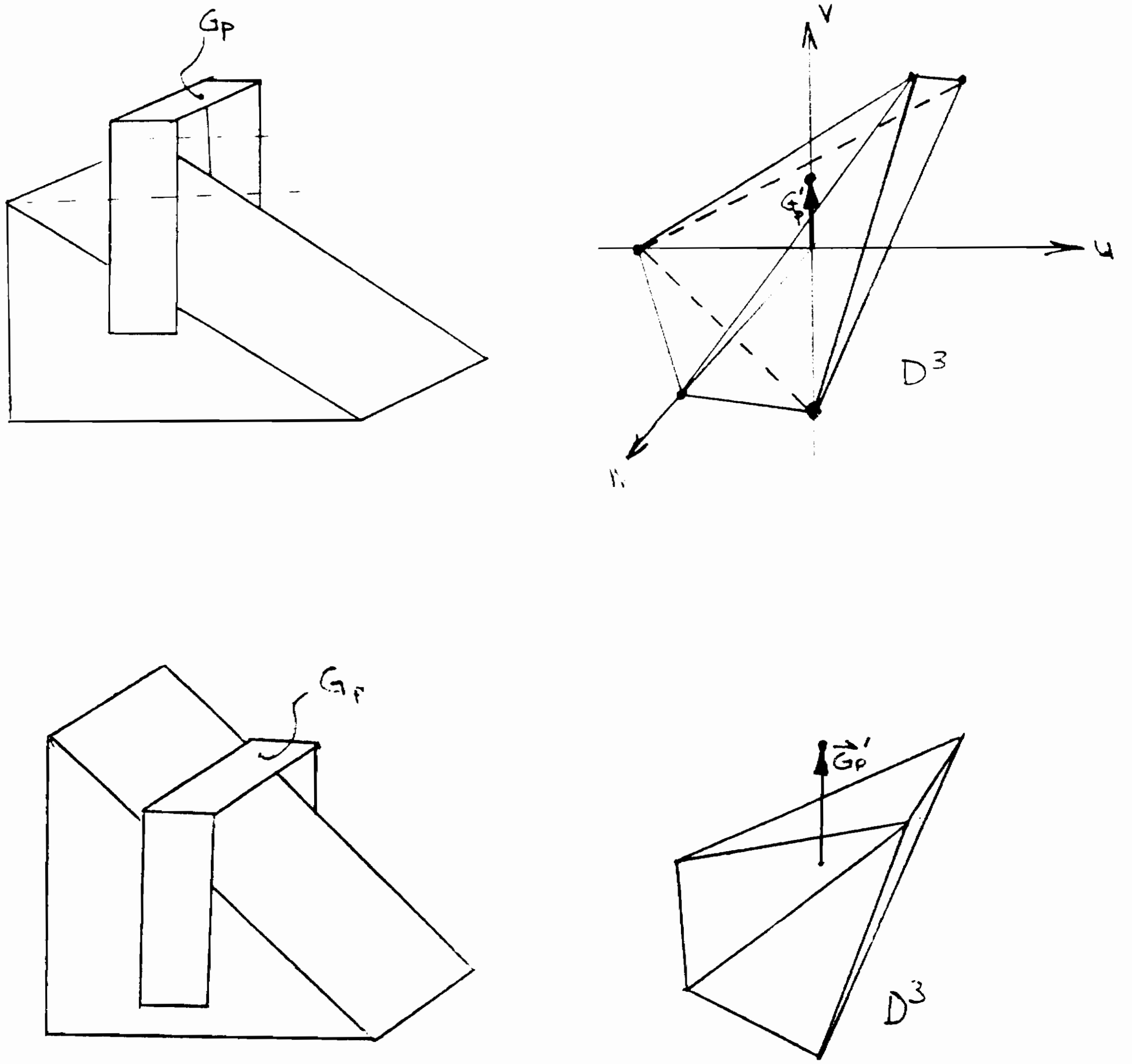


Figure 8. Checking for intersections with the gripper's palm.

intersection of the connecting edge and the palm plane is calculated and that point is checked in the dual domain to see if it lies on the palm's face.

If the gripper is enclosing the origin, this is a very easy check by the last theorem since the gripper's model looks like a convex object, i.e., it has a convex dual. If it does not enclose the origin then a method to be described in the chapter on concave objects must be used to determine if the point is on the gripper's palm. The above procedure is carried out for all dual vertices intersected by the position vector to the palm's dual plane.

5.5.2 Interference between gripper and table top.

As mentioned earlier, another unwanted collision is between the gripper and the table top that the object is resting on. The detection of this unwanted condition is simplified since the objects are dualized from a real world frame that has the y-axis perpendicular to the face plane the object is resting on and therefore the table top. There can only be one face in this plane for convex objects. This face is immediately found from the model representation as the dual point on the $-\hat{v}$ axis.

If there is a collision with the table top, it will be with one of the outside vertices of the gripper. A collision would occur if one of these vertices were placed below the table plane and would be indicated in the model if the position vector to the dual table top plane intersecting any of the dual vertices of the gripper. A very simple check. Figure 9a depicts a case where the gripper is intersecting the table top and Figure 9b a depicts a case where the gripper is clear of the table top.

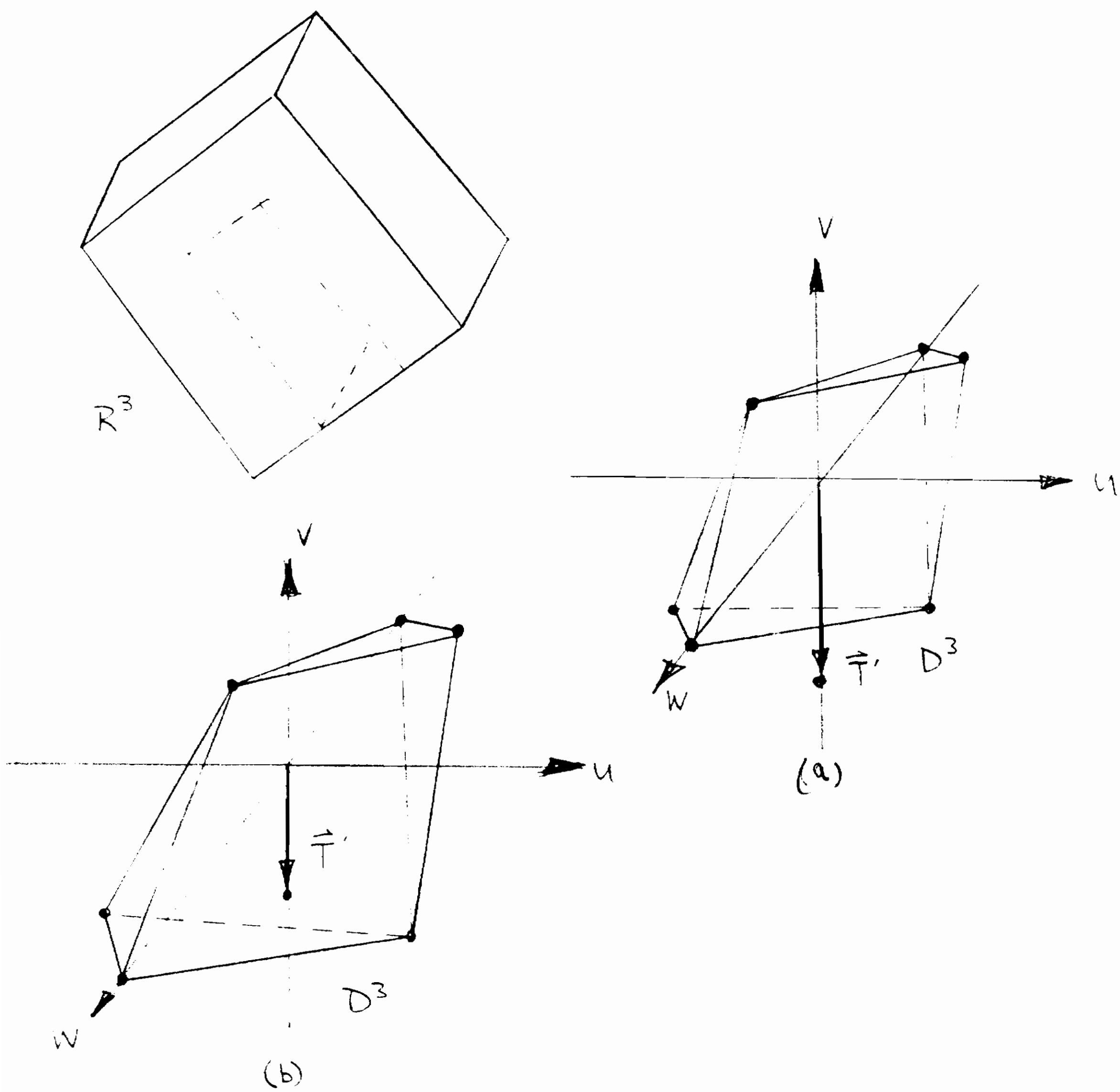


Figure 9. Checking for intersection with the table top.

5.6 *Convex Hulls of Concave Objects*

5.6.1 Non simple polyhedron

Determining whether or not a grasp on a concave object is valid is not as straightforward as with convex objects. The problem arises because a face plane of a non convex object does not divide space into a half space containing the whole object and a half space containing no object. This leads to duals that are not simple polyhedra. A simple polygon is defined as having edges that intersect only at the vertices of the polygon and divide a plane into exactly two regions. A simple polyhedron is made up of simple polygon faces. Figure 10 is an example of a face with a concavity and its non-simple polygon dual. Since the polyhedra duals are not simple, they do not divide dual space into two volumes, one representing the object and one representing free space as with convex duals. This makes determining whether or not a point is on a face a much more difficult task. In the next chapter a version of the Jordan Curve theorem will be presented to deal with this problem. In this section a method is presented that can determine whether a grasp position is good but cannot conclusively determine whether it is bad.

The method to be described in this section will find the convex hull of an object from its dual space model. A check of the grasp position against the dual model of the convex hull as described in the previous sections can then be performed. If there is no interference detected between the convex hull of the object and the gripper then there is no interference with the actual object. This is computationally simpler than the method presented in the next chapter which can determine if the gripper is intersecting the object at a grasp position within the convex hull of an object.

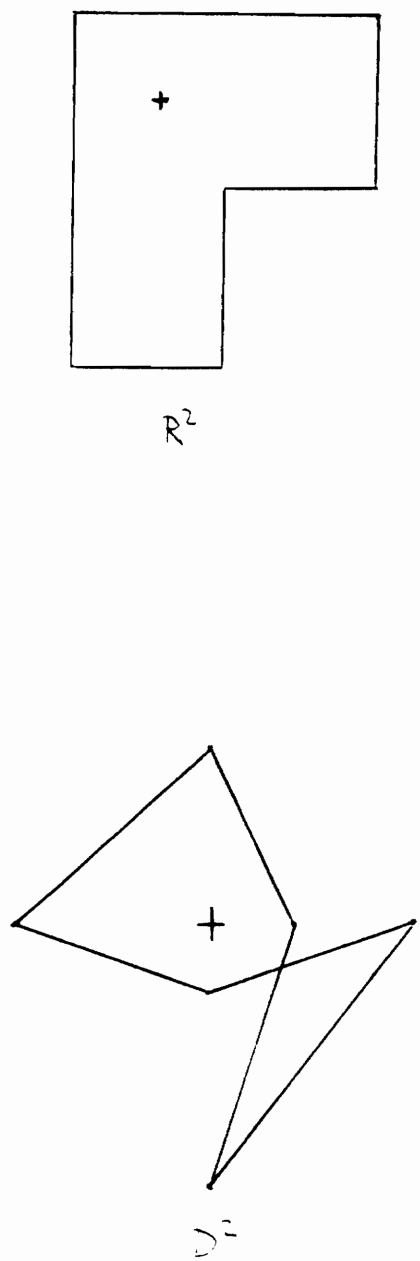


Figure 10. A concave polygon and its non simple dual.

5.6.2 Finding convex hulls in dual space

Since the models presented here are dualized from an origin located at the center of mass of the object, the origin will always be located within the convex hull of the object. Consequently, when dual space is filled from the dual origin, which is the mapping of the infinite plane from real space, a bounding set of planes will be found that form a closed polyhedron. These bounding planes represent the vertices belonging to the convex hull of the object. The intersections of this set of bounding dual vertices represent real planes and edges as well as the constructed ones belonging to the convex hull. The principle can be demonstrated in 2 dimensions with the following theorem.

Definition (5.6.1): A constructed edge of a convex hull is an edge belonging to the convex hull of a non convex object but not an edge of the object itself.

Theorem (5.6.2): The vertices of a concave polygon through which the edges and constructed edges pass to form the convex hull will have duals that are lines and intersect to define the vertices of a simple dual polygon. These vertices of this simple dual polygon will be the duals of the edges and constructed edges belonging to the convex hull of the original polygon.

Proof: A constructed edge, e, of the convex hull of a concavity will be a certain perpendicular distance, d_e , away from the origin. The N vertices within the concavity (concave or convex) can be thought to have lines through them parallel to the constructed edge e, $e \parallel l_i$, $i = 1...N$, where l_i are these imaginary lines. See Figure 11. The perpendicular distances, d_i , to each line l_i will be less than d_e, $d_i < d_e$, $i = 1...N$. In the dual domain each dual imaginary line, l'_i , and the dual of the constructed edge e′ will lie on a common line through the origin since they are parallel. Since the perpendicular distance to a line is equal to the inverse of the distance to its dual, $\left|\frac{1}{d_e}\right| < \left|\frac{1}{d_i}\right|$, $i = 1...N$, and so $|\vec{e}'| < |\vec{l}'_i|$, $i = 1...N$. This says no dual vertices intersect the position vector to the dual constructed edge. So any duals of constructed edges of a convex hull will form real vertices of a simple polygon in the dual domain.

In Figure 11, the convex hull of the polygon in R^2 is made up of edges A,E,F, and G, and the constructed edge e. In two dimensional dual space this convex hull is found from the portion of the dual polygon surrounding the origin, which is itself convex. In the example of Figure 11, this portion of the dual polygon has vertices A′, E′, F′, G′ and e′ , corresponding directly to the edges of the convex hull of the original polygon.

The above proof leads to a natural method of finding the convex hull of a polygon from its dual representation. Find all the intersections of the dual vertices. Any intersection point located by a position vector that does not intersect any dual vertices, represents an edge in the convex hull of the original polygon. This argument is easily extended to 3 dimensions by taking all combinations of three dual vertices and finding their point of intersection. If a position vector to one of these points does not intersect any dual vertices, then that point represents a face plane of the convex hull of the object.

5.7 Summary

With what we have developed in this chapter we can determine if a grasp on a polyhedron object is valid given the dual representations of the object and the gripper in the positions to be checked. By directly checking the duals candidate grasping planes we can tell if they are parallel and

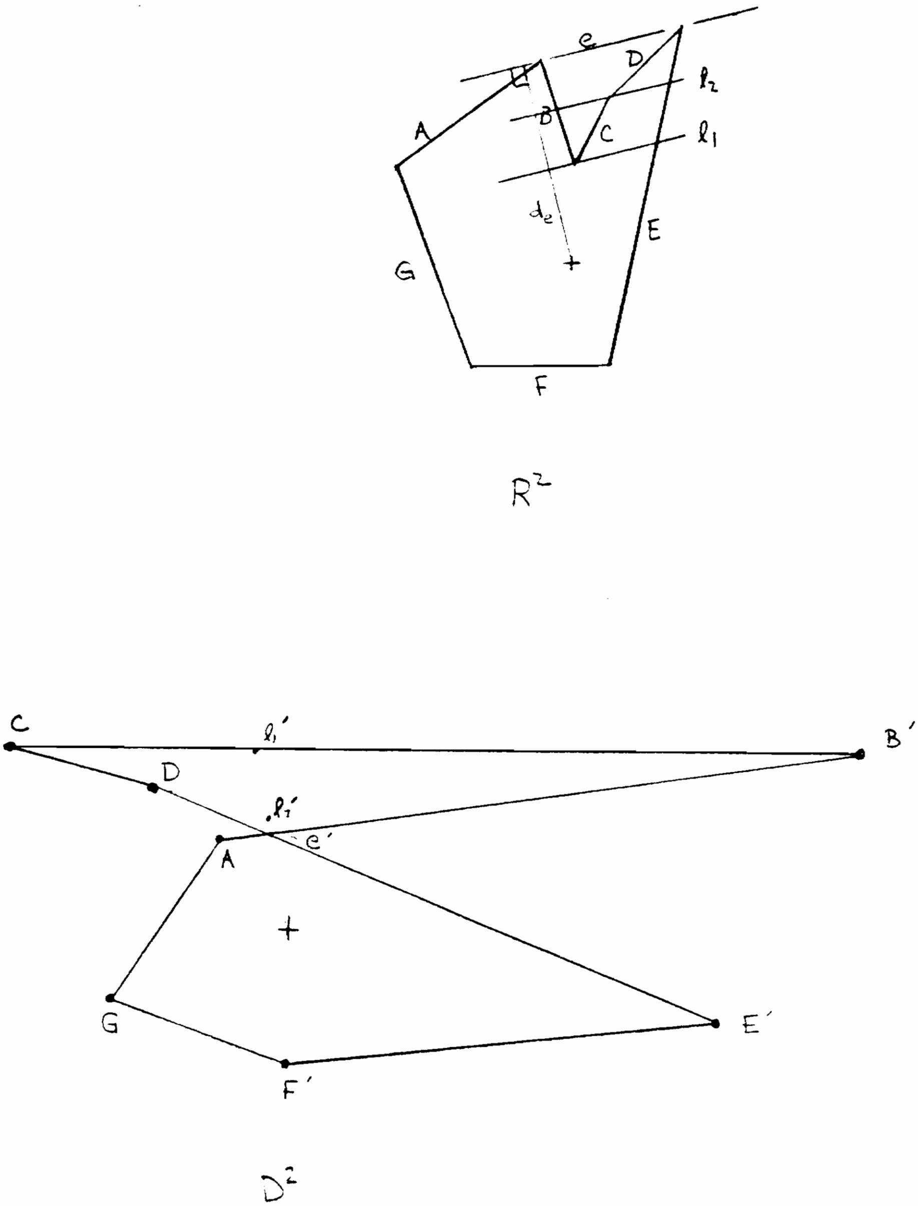


Figure 11. Finding the convex hull for a polygon from its dual representation.

close enough together. By doing simple perpendicular projections of the vertices of one face on to its parallel face and using the test for point inclusion on a convex face, we can check for face overlap.

Using the same point inclusion test also allows us to check for gripper pad/face overlap and palm/object intersections. For non convex we showed how to find the convex hull directly in the dual domain. Knowing the convex hull of the object and knowing which planes of the convex hull are real ones, the methods developed earlier in the chapter can be applied to the grip determination problem.

Chapter VI

GRASPS ON CONCAVE OBJECTS

If no suitable grasp is found on the convex hull of a concave object, it may be necessary to check a grip position inside the convex hull of the object. This is a much more complicated procedure because as seen earlier concave objects do not have simple polyhedra as their duals. This causes the test to determine whether a point is on a face to become the union of several conditions. A version of the Jordan Curve Theorem will be presented to help simplify this problem.

6.1 Suitable Face Pairs

As earlier, the first test for a valid grip is to determine whether the gripper pads are in contact with parallel, opposite faces. This still has the necessary condition that the dual face planes lie on a common line through the dual origin, but with concave objects this is not sufficient. Parallel faces on concave objects may occur with the same outward direction or maybe facing each other. This is where the sign information attached to the dual face planes becomes important. Faces parallel to each other and on opposite sides of the origin will have the same sign attached if they face each other or if they face away from each other; negative if the faces face each other and positive if they do not. So for exterior grips, we look for positive pairs on either side of the origin. In Figure 12, A′F′ or A′D′ are examples of acceptable pairs. Faces parallel but on the same side of the origin will have different signs attached if they face each other or face away from each other. A pair of dual planes on a common line with the negative point closer to the origin indicates faces facing each other. If the closer dual plane is positive then the faces point away from each other and are valid face planes for grasping. In Figure 12, A′B′ and F′E′ are examples of valid face plane pairs while B′C′ and E′D′ are not.

After insuring the face normals are correct, the spacing between the pair is checked by looking at the inverse distances to the dual planes. The spacing is equal to the sum of the inverse distance to the dual planes if they are on different sides of the origin. If they are on the same side of the origin, the spacing is equal to the absolute value of the difference of the inverse distances. This can be compactly stated as:

$$\left| \frac{\mathrm{sgn}[P'_1]}{|\vec{P'}_1|} + \frac{\mathrm{sgn}[P'_2]}{|\vec{P'}_2|} \right| < \text{grip width tolerance} \tag{6.1.1}$$

where $\vec{P'}_1$ and $\vec{P'}_2$ are the position vectors to the dual face planes and $\mathrm{sgn}[P'_i]$ returns the sign given point P'_i.

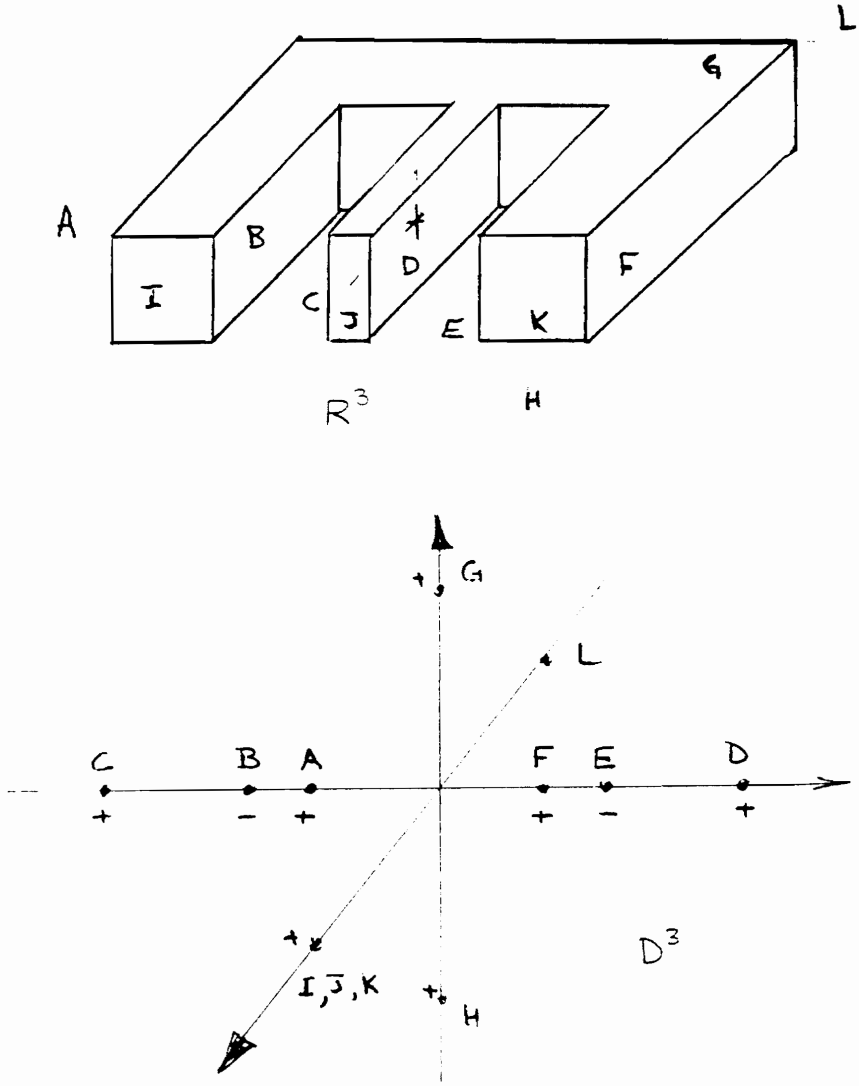


Figure 12. A non convex object with its signed dual representation.

6.2 *Face Intersection Detection*

When the gripper is in a position to be checked for meeting the valid grip criteria we must have a method for determining whether the pads are actually on faces and determining whether there are any unwanted intersections of the gripper with the object. One way to determine if there are unwanted intersections is to check that no faces of the gripper intersect with any of the faces of the object.

The following method can be used in real space to determine if two faces, A and B intersect. First we define a "vertex pair". A vertex pair is any two vertices that are connected by an edge. To determine if two vertices define a pair, we look in the vertex table. If the two share at least two common planes then they form a pair.

Check all pairs of connected vertices of face A against the plane of the other face B. For ever pair found that has one vertex on either side of the face plane B, there is a possibility that A and B intersect along these edges. For each pair found satisfying this condition, calculate the point of intersection between the edge connecting the vertex pair and the plane. Now check if this point is on the face belonging to the plane. We can perform this face intersection check completely in the dual domain. The face with the fewer vertices should be checked against the plane of the face with the greater number of vertices to reduce the number of checks to be performed.

The check to determine if a vertex pair has a vertex on either side of a plane is simple. Figure 13 will be referenced to demonstrate the procedure. Plane A is checked against plane B. If the position vector to B′ intersects one and only one of a connected pair of dual vertices, then in real space, those vertices lie on either side of face plane B. In Figure 13, vertex pair (v′(1),v′(2)) and (v′(2),v′(3)) are the only pair satisfying this condition, $\vec{B}'$ intersecting v′(2) and not v′(1) or

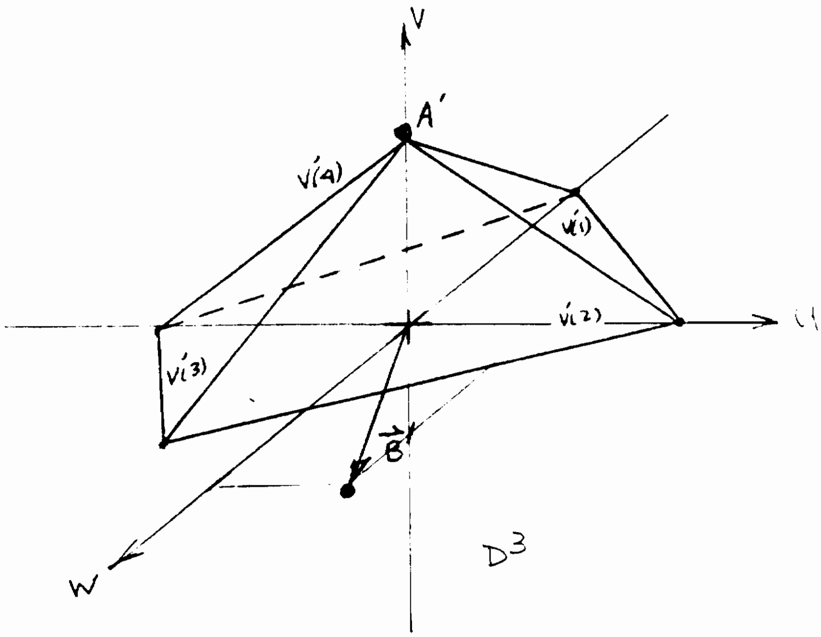


Figure 13. Checking vertex pairs of face A against plane B.

$v'(3)$. If the equation for the dual vertex is: $v_x u + v_y v + v_z w - 1 = c$, where $c = 0$, then entering the coordinates for a dual plane, (A,B,C), will give a $c < 0$ or $c > 0$ depending on which side of the dual vertex it is on. So if a different sign of c is obtained for each vertex of the pair the intersection point of the edge connecting the vertices and the face plane B is calculated. We must now determine if this dual point is on face B′ .

6.3 *Jordan Curve Theorem*

A method for determining if a point is on a non convex face is needed.

Jordan Curve Theorem (6.3.1):[31] If S is any simple polygon in a plane P, the points of P which are not on S can be divided into two sets A and B in such a way that any two points in the same set can be joined by a polygonal path not intersecting S , while no two points, one of which is in A and the other in B , can be so joined.

Corollary (6.3.2):[31] A half line from a point in A in a direction not parallel to any side of polygon S , will intersect S an odd number of times while such a line drawn from a point in B will intersect an even number of times.

To apply the Jordan Curve corollary in R^3 the half lines are normally taken parallel with an axis in order to simplify calculations. In two dimensions the half line is a locus of points with a constant x value and a variable y value, for example. In three dimensions, two degrees of freedom must be varied in order that the half line remain on the face. For example, x constant, y variable, and $z = f(y)$, unless the face is perpendicular to a coordinate axis. This is why it is preferred to use the Jordan Curve corollary (6.3.2) in two dimensions. The same applies to its application in dual space.

To reduce the problem to two dimensions, a method of projecting the dual face, its dual edge planes, and the dual point to be checked to a single planner dual polygon and dual point will be presented next along with a procedure to apply the above corollary directly in dual space.

6.4 *Face Rotation*

In reducing the problem to a two dimensional one, the coordinate axes are rotated such that if $\vec{A}'$ is the position vector to the dual face plane of interest, it will be aligned with the $\hat{w}$ axis. After the rotation the dual model is reduced to the u-v plane by a simple projection. Let $\vec{A}' = < a_u, a_v, a_w >$ and make an angle α between the projection of $\vec{A}'$ onto the u-v plane and the positive $\hat{u}$ axis and an angle β between $\vec{A}'$ and the positive $\hat{w}$ axis. Refer to Figure 14.

$$\alpha = \tan^{-1}\left[\frac{a_v}{a_u} \right] \tag{6.4.1}$$

and

$$\beta = \tan^{-1}\left[\frac{\sqrt{a_u^2 + a_v^2}}{a_w} \right]. \tag{6.4.2}$$

Bringing the coordinate axis $\hat{w}$ into alignment with $\vec{A}'$, can be thought of as first rotating $\vec{A}'$ about $\hat{w}$ by $-\alpha$ to give an $\vec{A}^{1\prime}$ and then rotating the new $\vec{A}^{1\prime}$ about $\hat{v}$ by $-\beta$ to get an $\vec{A}^{2\prime}$.

$$\vec{A}^{2\prime} = \mathrm{Rot}[v, -\beta]\vec{A}^{1\prime} \tag{6.4.3}$$

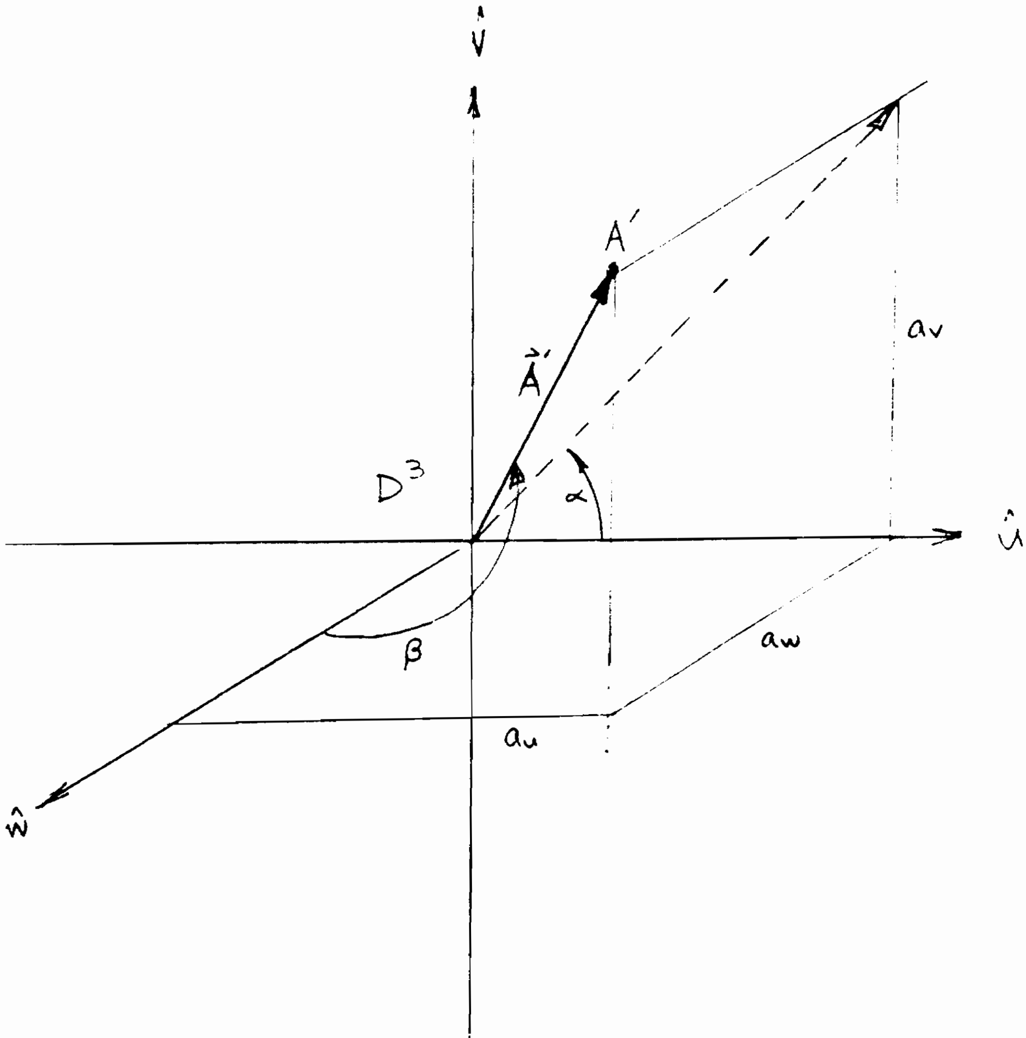


Figure 14. Parameters for locating the dual plane A.

$$= \mathrm{Rot}[v, -\beta]\ \mathrm{Rot}[w, -\alpha]\vec{\mathrm{A}}'$$

$$= \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\beta\cos\alpha & \cos\beta\sin\alpha & -\sin\beta \\ -\sin\alpha & \cos\alpha & 0 \\ \sin\beta\cos\alpha & \sin\beta\sin\alpha & \cos\beta \end{bmatrix}$$

$$= [\mathrm{ROT}]\,\vec{\mathrm{A}}'.$$

6.5 *Face Projection*

After the rotation the projection of the dual face to a planar polygon is performed. The projection is carried in the dual domain. For each edge plane of the face find the intersection of the dual edge with the u-v plane. The point found defines the plane perpendicular to the face plane and intersecting the face along the same edge as the original edge plane. Refer to Figure 15. Let $\vec{\mathrm{A}}'$ be the position vector to the dual face plane and $\vec{\mathrm{B}}'$ be the position vector to a dual edge plane defining the face. Let $\vec{\mathrm{L}}$ be the vector from A′ to B′ and $\vec{\mathrm{C}}$ be the position vector (two dimensional) to the projected point. From Figure 15,

$$\vec{\mathrm{L}} = \vec{\mathrm{B}}' - \vec{\mathrm{A}}' \tag{6.5.1}$$

$$t\,\vec{\mathrm{L}} = \vec{\mathrm{C}} - \vec{\mathrm{A}}' \tag{6.5.2}$$

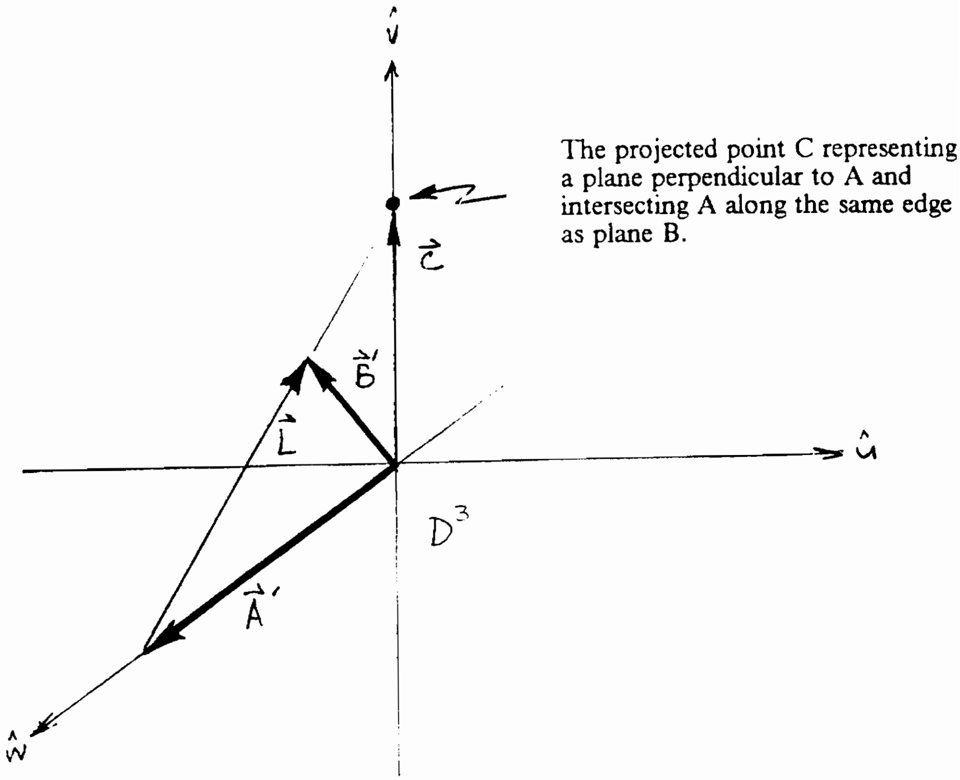


Figure 15. Geometry of the dual face projection.

In rectangular coordinates,

$$< t\,l_u, t\,l_v, t\,l_w > \;=\; < c_u, c_v, 0 > \;-\; < 0, 0, a_w > \tag{6.5.3}$$

Solving for t,

$$t = -\frac{a_w}{l_w} \tag{6.5.4}$$

The projected point is then,

$$c_u = -\,a_w\left(\frac{l_u}{l_w}\right) \tag{6.5.5}$$

$$c_v = -\,a_w\left(\frac{l_v}{l_w}\right). \tag{6.5.6}$$

$$c_w = 0 \tag{6.5.7}$$

A list defining how the projected vertices are connected is obtained by searching the original vertex list to find elements that have both the face plane and an edge plane in common. This procedure completely defines a planar face equivalent to the original.

6.6 Point Rotation

Since the dual domain coordinate system has been rotated, the dual point that is being tested must be found in the new system. The simplest method is to find the new axis intercepts of the plane representing the point. If the point being tested is,

$$R = (r_x, r_y, r_z) \tag{6.6.1}$$

then

$$': r_x u + r_y v + r_z w = 1. \qquad (6.6.2)R$$

The new coordinate axes are given by:

$$\vec{u}' = \cos\beta \cos\alpha\vec{u} + \cos\beta \sin\alpha\vec{v} - \sin\beta\vec{w} \tag{6.6.3}$$

$$\vec{v}' = -\sin\alpha\vec{u} + \cos\alpha\vec{v} \tag{6.6.4}$$

$$\vec{w}' = \sin\beta \cos\alpha\vec{u} + \sin\beta \sin\alpha\vec{v} + \cos\beta\vec{w} \tag{6.6.5}$$

so the new axis intercepts are:

$$\left[\frac{1}{I_u}, \frac{1}{I_v}, \frac{1}{I_w}\right]^T = [\text{ ROT }]\left[\frac{1}{r_x}, \frac{1}{r_y}, \frac{1}{r_z}\right]^T. \tag{6.6.6}$$

The projection of the dual point in the rotated system is the line passing through $\frac{1}{I_u}\hat{u}$ and $\frac{1}{I_v}\hat{v}$. The problem now is strictly a two dimensional one and the following dual version of the Jordan Curve corollary (6.3.2) can be applied.

6.7 *Dual Edge Boundaries*

To apply corollary (6.3.2), we first must be able to determine whether a point is on a line segment, i.e. an edge of a polygonal face. Recalling that we are now working in two dimensional dual space D^2, the line segment is represented by three points which are the duals of the edge line and the two lines intersecting the edge line to form the ends of the edge. Let the dual edge line be B′ and the bounding dual lines be A′ and C′. See Figure 16. The three points form two angles in dual space with the dual edge line B′ as the vertex. All points on the face will be represented as lines within one of the angles A′B′C′ and intersecting B′. The correct angle is the angle that does not bound the origin, for if it did, one of the points on the edge would be at infinity.

There are two cases to consider. In the first, the dual lines A′ and C′ are on the same side of the line through the origin and B , $\overline{OB'}$ as in Figure 16a. Let the line for edge B be given by $b_x x + b_y y = 1$ so its dual is $B' = (b_x, b_y)$. In D^2 , $\overline{OB'}$ is given by

$$b_y u - b_x v = 0. \tag{6.7.1}$$

The duals to the two boundary lines are $A' = (a_x, a_y)$ and $C' = (c_x, c_y)$. By substituting these points into (6.7.1) we get the following condition.

$$[\, b_y a_x - b_x a_y \,][\, b_y c_x - b_x c_y \,] > 0. \tag{6.7.2}$$

If (6.7.2) is satisfied, then for a point P to lie on *edge* B, the following conditions must be met:

1. $\vec{B}' \cap P' = B'$ (6.7.3)

and

2. $[(\vec{A}' \cap P' \neq 0)$ and $(\vec{C}' \cap P' = 0)]$ or $[(\vec{A}' \cap P' = 0)$ and $(\vec{C}' \cap P' \neq 0)]$.

In the second case, A′ and C′ are on opposite sides of $\overline{OB'}$ as in Figure 16b, and

$$[\, b_y a_x - b_x a_y \,][\, b_y c_x - b_x c_y \,] < 0. \tag{6.7.4}$$

For P to be on *edge* B, the conditions are now:

1. $\vec{B}' \cap P' = B'$ (6.7.5)

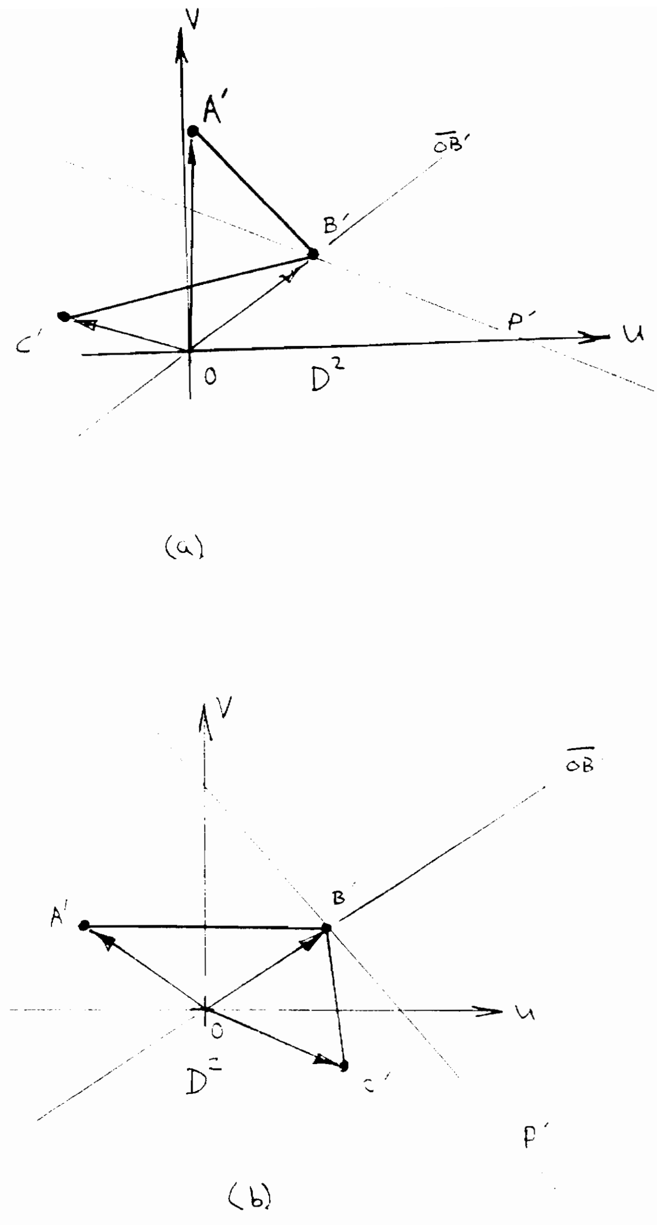


Figure 16. Angles defining the boundaries of a dual edge.

and

2. $[\vec{A}' \cap P' = 0]$ and $[\vec{C}' \cap P' = 0]$.

So to tell if a point is on an edge we check to see if (6.7.3) or (6.7.5) is satisfied depending on which of (6.7.2) or (6.7.4) respectively is met. In both Figure 16a and Figure 16b the line P′ represents a point P on edge B.

6.8 *Application of Jordan Curve Corollary in Dual Space*

Since we can now rotate and project any face, we can look at applying the corollary to the Jordan Curve Theorem in dual space to tell if a point P lies on a face F.

The half line from the point under test, P, to infinity can be in any direction, so for convenience we let it be parallel to the y axis. In D this half line is an infinite set of lines between P′ and the u axis with the same u intercept as P′ where $P' : p_x u + p_y v = 1$. Any dual edges of F′ that are in the region between P′ and the u axis represent edges that the half line may intersect.

If the N dual edges of F′ are $e'_i \;\; i = 1 \ldots N$, where each $e'_i = (e_{ix}, e_{iy})$, then the ones that satisfy

$$p_x e_{ix} + p_y e_{iy} - 1 \leq 0 \tag{6.8.1}$$

and

$$[\, e_{iy} > 0 \text{ if } p_y > 0 \,] \text{ or } [\, e_{iy} < 0 \text{ if } p_y < 0 \,]$$

define a set PI, of possible edges intersected.

The last two conditions of (6.8.1) correspond to looking at the half line from P′ to positive infinity or to negative infinity respectively and depends on which side of the x axis P is on. For each $e'_i \in PI$ satisfying (6.8.1), the line from $(p_x, 0)$ to (e_{ix}, e_{iy}) is computed and checked by the methods of section 6.7 to see if it represents a point of intersection between the half line and the edge e_i. If the number of intersections indicated after checking all e_i's of PI have been so checked is odd, then the point P is on F, if the number of indicated intersection is even then P is not on F.

Figure 17 shows an example. In D^2 as the line P'_1 is rotated about its u axis intercept such that its v intercept moves toward the origin, it represents the point P_1 moving parallel to the y axis toward negative infinity. As P'_1 rotates it intersects E′ then F pr before reaching the origin. By the checks of section 6.7 when P'_1 is incident on each of these points, it represents a point on an edge, therefore P'_1 has a count of two and so it does not represent a point on the face.

Likewise, as P'_2 is rotated about its u intercept it becomes incident on C′ and then A′. By the checks of section 6.7 when it is incident on C′ it does not represent a point on an edge, when it is incident on A′ it does represent a point on an edge and so P'_2 has a count of one and represents a point on the face.

Using the methods of the last three sections, it can be determined if the gripper pads are in contact with the intended faces and not just lying in the same planes. This is done by checking each vertex of the intended face against the intended gripping pad. If one vertex of the face can be found to lie on the pad or one vertex of the pad on the face, then there is overlap. Each of the two gripper pads are checked in this manner.

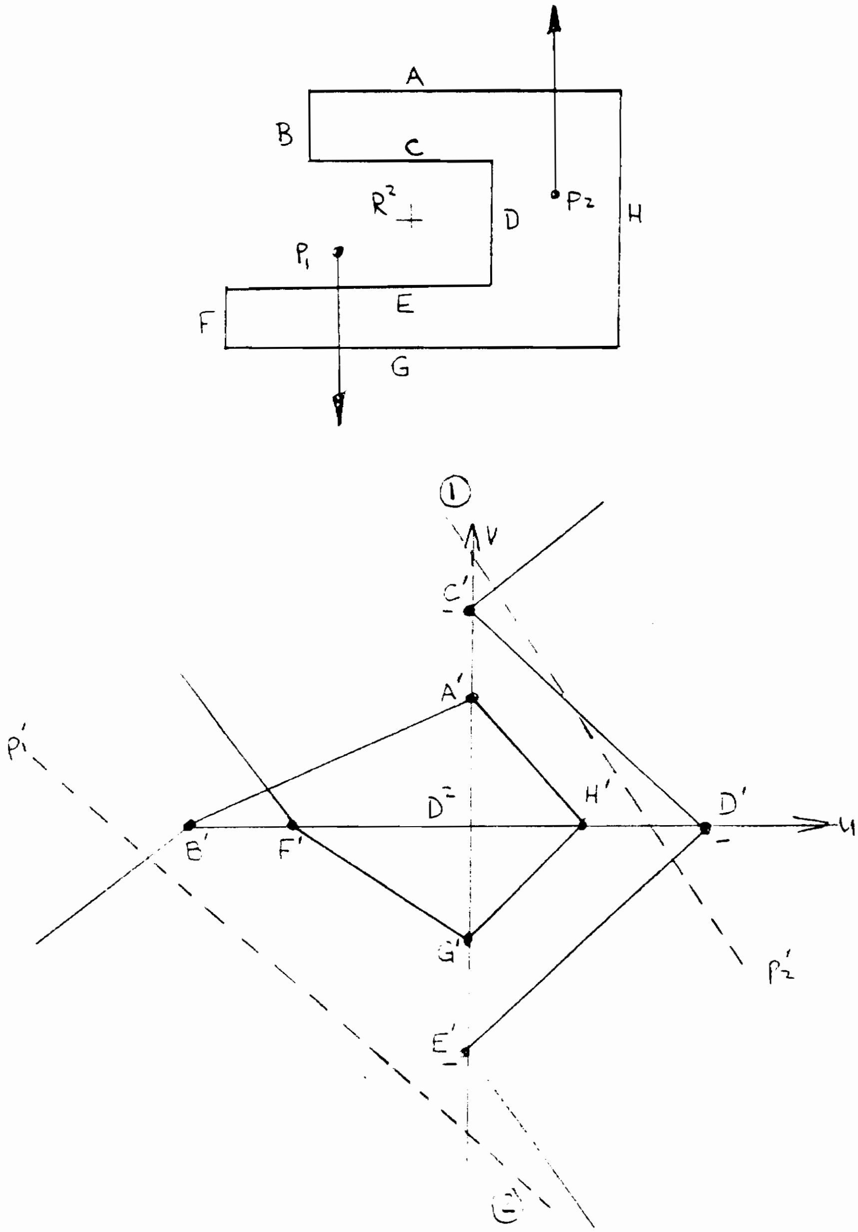


Figure 17. Examples of applying the Jordan Curve Corollary in dual space.

6.9 Checking for Gripper/Object Intersections

After it is confirmed that the pads are in correct contact with the intended faces we must check for unwanted gripper/object intersections. This can be a costly operation if each face of the gripper is arbitrarily checked in turn to see if it is intersecting any of the object's faces. This was the method used for the convex objects but there we could restrict the checks to a single face of the gripper, the palm. While we do not have any prior knowledge about the object, we do have it for the gripper and we use this knowledge to facilitate the gripper/object intersection check.

By recognizing that the gripper, G, is composed of a convex polyhedron, G_O, with a convex section G_I, removed, we can exploit some of the nice properties of convex objects to write explicitly the conditions for a point to lie outside of the gripper body. G_O corresponds to our outside model and G_I corresponds to the inside model of the gripper.

For a point to be *in* a convex polyhedron, it must be on the origin side of planes containing faces with normals pointing away from the origin and not on the origin side of planes that contain faces with normals that point toward the origin. For a point P and a polyhedron with N faces, f_i, this can be written in the dual domain as:

$$\begin{gathered}\text{If } \operatorname{sgn}[f'_i] = \text{positive then } \vec{f'_i} \cap P' = 0 \quad \text{for all } i, \; i = 1 \text{ to N} \\ \text{and} \\ \text{If } \operatorname{sgn}[f'_i] = \text{negative then } \vec{f'_i} \cap P' \neq 0 \quad \text{for all } i, \; i = 1 \text{ to N}\end{gathered} \tag{6.9.1}$$

For a point to be outside of this convex polyhedron, the compliment of (6.9.1) is taken to yield:

$$\begin{gathered}\text{If } \operatorname{sgn}[f'_i] = \text{positive then } \vec{f'_i} \cap P' \neq 0 \quad \text{for some } i, \; i = 1 \text{ to N} \\ \text{or} \\ \text{If } \operatorname{sgn}[f'_i] = \text{negative then } \vec{f'_i} \cap P' = 0 \quad \text{for some } i, \; i = 1 \text{ to N}\end{gathered} \tag{6.9.2}$$

Let $\vec{f'}_i$ $i = 1$ to 6 represent the faces of the convex hull of the gripper, what we have called the outside model of the gripper and $\vec{f'}_i$ $i = 7$ to 12 represent the faces that compose the convex piece that is removed, what we have called the inside model of the gripper, V'_j be the j^{th} vertex of the object and sgn[f'_i] returns the sign of point f'_i.

So for the parallel jaw gripper considered a convex polyhedron with a convex section removed, the conditions for a vertex to lie outside the gripper body are that it lie outside G_O or inside G_I.

$$
\begin{array}{ll}
\text{If } \operatorname{sgn}[f'_i] = \text{positive then } \vec{f'}_i \cap V'_j \neq 0 \quad \text{for some } i, \quad i = 1 \text{ to } 6 \\
\qquad\qquad\qquad\qquad \text{or} & (6.9.3) \\
\text{If } \operatorname{sgn}[f'_i] = \text{negative then } \vec{f'}_i \cap V'_j = 0 \quad \text{for some } i, \quad i = 1 \text{ to } N
\end{array}
$$

or

$$
\begin{array}{l}
\text{If } \operatorname{sgn}[f'_i] = \text{positive then } \vec{f'}_i \cap V'_j = 0 \quad \text{for all } i, \quad i = 7 \text{ to } 12 \\
\qquad\qquad\qquad\qquad \text{and} \\
\text{If } \operatorname{sgn}[f'_i] = \text{negative then } \vec{f'}_i \cap V'_j \neq 0 \quad \text{for all } i, \quad i = 1 \text{ to } 12
\end{array}
$$

For no object/gripper intersection to occur, (6.9.3) must be satisfied for all vertices of the object. Clearly intersection checks on the convex hulls are much simpler procedures since we do not have to check all vertices but only those not on the origin side of the palm plane.

The checks for the gripper/table top intersection is performed in the same manner as described in the chapter on convex objects since the check is independent of the target object.

6.10 Summary

In this chapter we have extended the results from the chapter on convex objects. We can now check for parallel faces when the faces lie on the same side of the origin and we can differentiate between parallel faces facing toward each other and faces facing away from each other as required for an external grasp. We developed a method of applying the Jordan Curve corollary (6.3.2.) directly in the dual domain so we can now check for non zero face overlap and gripper pad/face contact. We also showed how modeling the gripper as two convex pieces, as in constructive solid geometry, allowed us to develop a simplified check for determining if there was an unwanted intersection of the gripper and the object.

Chapter VII

GENERATING GRASP POSITIONS

All the work in this thesis up to this point has been concerned with procedures for checking whether a particular grasp configuration is valid. No simple or guaranteed method has yet been found which will generate valid gripping positions. As mentioned in the introduction, most present algorithms attempting to do this task start with a set of candidate gripping positions either defined apriori or generated by a few simple heuristics, then consecutively filter the set to obtain a set of feasible grip locations. These are usually evaluated and sorted using greatly varied criteria based on what the algorithm's author considers an "optimal grip", for example stability, reachability, or simply findability. These are essentially generate and test algorithms and differ mainly in the "reasoning" used in the generate stage and/or the methods used in the test stage which are often dependent on the model being used as stated in the introduction.

Since the model we are working with is a boundary representation, methods for computing free space volumetric solutions are much more difficult to calculate then a cell decomposition or octree representation. Previous work on grip determination using boundary representations, although not working in the dual domain, never the less give many useful heuristics which we can apply to a generate and test algorithm for grip determination using the dual representation presented in this thesis.

The algorithm forwarded here is a two stage generate and test. The first stage of generation and test determines a set of suitable face pair combinations. This set includes all surface pairs with properties desirable for gripping with the parallel jaw gripper. The second stage of generation and test breaks candidate grasping areas for each pair of faces found, into a finite number of discrete points. For each point a set of gripper orientations are tested to find valid grasp configurations. In an attempt to reduce the number of tests performed some intelligence is built into the algorithm in the form of heuristics. These will help in determining which set of generated face pairs stand the best chance of containing findable, valid grasping points. Heuristics are also used to some extent in the face discretization in an effort to generate gripping positions on the object that will pass through the remaining filters.

7.1 Finding Candidate Grasping Areas

7.1.1 Generating Candidate Plane Pairs

A set FC is generated which contains all possible pair combinations of all the dual faces of an object. If F is the set of all dual object faces then,

$$FC = \{\ (f'_i, f'_j) : f'_i, f'_j \in F'\ , i \neq j\ \} \tag{7.1.1}$$

From this set we apply the criteria for suitable face pairs presented earlier,

$$\left| \frac{\text{sgn}[f'_i]}{|\vec{f'_i}|} + \frac{\text{sgn}[f'_j]}{|\vec{f'_j}|} \right| < \text{grip width tolerance,} \tag{7.1.2}$$

$$\vec{f'_i} = s\, \vec{f'_j} \text{ for some scalar } s \tag{7.1.3}$$

where

$$\begin{aligned} &\text{if} \quad s < 0 \quad \text{then} \quad \text{sgn}[f'_i] = \text{sgn}[f'_j] = \text{positive} \\ &\qquad\qquad\qquad \text{and} \\ &\text{if} \quad s > 0 \quad \text{then} \quad \text{sgn}[f'_i]\,|\vec{f'_j}| + \text{sgn}[f'_j]\,|\vec{f'_i}| > 0 \end{aligned} \tag{7.1.4}$$

Condition 7.1.4 guarantees the face normals are correct for a pair of faces to be gripped.

Also eliminated are face pairs that are know not to be grippable by some a priori information and pairs that have a face in contact with the table top, recognized by either f'_i or f'_j being the point on the $-v$ axis closest to the origin. The duals of face pairs known to be ungrippable a priori are placed in a set and subtracted from *FC*. In this paper we assume no prior knowledge about the part.

Equations (7.1.2) through (7.1.4) filter *FC* to give a set *FPO* containing the duals of parallel opposing faces with proper spacing.

7.1.2 Prioritizing Candidate Pairs

The set *FPO* is now divided into three sub sets with different degrees of priority for further processing. $FPO_{ch} \in FPO$ represents the set of all pairs with planes located on the convex hull of the object. These are identified on the dual model as pairs (f'_i, f'_j) satisfying:

$$\vec{f'_i} \cap O' = f'_i \text{ and } \vec{f'_j} \cap O' = f'_j \tag{7.1.6}$$

where O' is the dual object. The set FPO_{ch} is searched first because if valid grasping faces can be found in the set, the interference checking will be greatly simplified.

The set $FPO_d \in FPO$ is the subset representing pairs with planes located on either side of the origin. These pairs have s in (7.1.3) and (7.1.4) less then zero. This set is attractive because it will tend to minimize torques exerted on the gripper and the chances of the object twisting while grasped. Not all of FPO_d will need to be searched by recognizing $FPO_{ch} \in FPO_d$. Hence, FPO_{ch} is attractive for this reason as well as the simplified interference checking.

The third set contains the duals of remaining faces. $FPO_s \in FPO$ represent the pairs containing planes that are on the same side of the origin where

$$FPO_s = FPO - FPO_d \tag{7.1.7}$$

and FPO_s can be recognized by s in (7.1.3) and (7.1.4) being greater then zero.

All three sets are sorted by the criteria that the pairs with planes which are more centered around the origin have a higher priority. This will keep the center of the gripper as close as possible to the centroid of the object. For pairs represented by set FPO_s we want the average perpendicular distance to the planes to be as small as possible. This means ranking by the minimum of:

$$\frac{1}{2}\,|d_i + d_j| \tag{7.1.8}$$

where d_i and d_j are the perpendicular distances to planes of faces f_i and f_j. If the pair has planes on different sides of the origin, we want to rank by the minimum of:

$$|d_i - d_j|\,. \tag{7.1.9}$$

These two ranking criteria can be accomplished for all pairs in FPO by a single ranking of the dual faces by minimizing the values of:

$$\left| \frac{\mathrm{sgn}[f'_i]}{|\vec{f'}_i|} - \frac{\mathrm{sgn}[f'_j]}{|\vec{f'}_j|} \right| . \tag{7.1.10}$$

From this point the search continues by first only looking at FPO_{ch}, then only at $FPO_d - FPO_{ch}$, and finally at FPO_s only after later filters eliminate all elements of a set do we go back and begin working with the next set.

7.1.3 Checking Face Overlap

In checking for non zero overlap we first calculate the set of all intersections between the edges of the two projected faces. This can performed in the dual domain. After doing the rotation and projection of the faces under consideration by methods of section 6.4 and 6.5, calculate all lines formed by connecting the dual edges of one face with the dual edges of the other face. Each of these lines is checked to see if it is *on* both dual edges as explained in section 6.7. Each line found to be on both dual edges represents a vertex belonging to the set of vertices defining the area of overlap of the two projected faces. This area is called the *overlap polygon* , OP_{ij} , of face pair (f_i,f_j). The vertices found above form a set which is either a subset or equal to the full set of vertices belonging to OP_{ij} called OPV_{ij} . The guaranteed full set of vertices belonging to the overlap polygon is found from

$$OPV_{ij} = \{ v' = e'_i \cap e'_j : e'_i \in f'_i ,\ e'_j \in f'_j \} \cup \{ v'_i \subset f'_j \} \cup \{ v'_j \subset f'_i \} \tag{7.1.11}$$

where e'_i and e'_j are the dual edges of f_i and f_j respectively and the sets $\{ v'_i \subset f'_j \}$ and $\{ v'_j \subset f'_i \}$ are the vertices of one face which lie on the polygonal face of the other.

Note when set FPO_{ch} is tested for this overlap, the overlap polygon will be convex since the two faces and their projections are convex. This will help out in the second stage of the algorithm where we pick discrete grasping points in the overlap polygon for each face pair. Summarizing, what we have found so far are sets of vertices defining regions on face pairs that are feasible gripping locations. These regions now need to be checked to see if the gripper can grasp them in some manner without collisions.

7.2 *Finding Valid Positions and Orientations*

At the second stage we generate a discrete set of points for each face pair's overlap polygon. The locations of the planes of the face pair have already restricted the gripper to three degrees of freedom. By picking discrete points to test at, two more degrees of freedom are removed leaving the gripper with only one rotational degree of freedom.

One method to find a point on the overlap polygon would be to define a pseudo-center for the polygon equal to the median value of its vertices. Note that since the origin is located within the object there is a chance that such a point may lie near the perpendicular projection from the origin making the solution in dual space numerically ill conditioned. This arises because of the inability of our model to represent planes or lines through the origin. An example of an extreme case is if the projected origin intersects a projected face at $(0,y)$ or $(x,0)$, then the dual point could not be calculated.

The pseudo-center can be found as follows. The dual vertices can be written as:

$$[\, v_{ij1}, v_{ij2}, \ldots, v_{ijN} \,]^T \begin{bmatrix} u \\ v \end{bmatrix} = 1 \qquad (7.2.1)$$

or

$$[\,V'\,]\begin{bmatrix} u \\ v \end{bmatrix} = 1 \tag{7.2.2}$$

where the v'_{ijk}'s are row vectors containing the coefficients of the k^{th} dual vertex of OP_{ij} and so V' is N × 2. The dual pseudo-center, C'_{ij}, is $C'_{ijx}u + C'_{ijy}v = 1$ with

$$C'_{ijx} = \frac{1}{N}\sum_{k=0}^{N} V_{k1} \tag{7.2.3}$$

and,

$$C'_{ijy} = \frac{1}{N}\sum_{k=0}^{N} V_{k2} \tag{7.2.4}$$

As mentioned at the end of the last section the overlap polygon is convex if both faces are convex. In this case we are guaranteed that the point found above is on the face. But if one or both faces are concave we can not say this. Also, we would like more then one test point per face pair to improve the odds of finding a successful grip location.

To generate more points we take three of the overlap polygon's vertices at a time and use (7.2.1) to find a pseudo-center for each. Three vertices taken at a time will always define a convex region and the center as defined, will always be contained in this region. A region could be totally on the overlap polygon, partially on it, or not on it at all, but there will always be some of these triangular regions totally contained on OP_{ij} and so we are sure to find some points in the desired overlap polygon. So for each pair of faces remaining take all three at a time combinations of their overlap polygon vertices, find their pseudo-centers and check to make sure they are on the actual overlap polygon. Let the set of points found for each OP_{ij} be called OPP_{ij}. These are used as the discrete points for further testing.

Each of these points defines an axis for the gripper when it is in a position to grasp. If the gripper is going to grasp a desired point it is restricted to a single rotational freedom about that axis. So discretizing the 360° the gripper can rotate in gives a final set of configurations that must pass the non interference tests as presented in the previous chapters. Points and orientations passing these final checks are considered valid grasp positions.

To pick a final grasp position we can take the first few produced and know that they are the "best" in so far as the order in which they were processed reflects what we considered as more desirable features in a pair, i.e., keeping the center of mass centered between the fingers and grabbing on the convex hull. Alternately, we can take all pairs from each of the sets FPO_{ch} , FPO_s , and $FPO_{ch} - FPO_d$ that make it through these filters, retaining their order to preserve their "quality" ranking, and apply application specific criteria or other rating measures to them.

7.3 *Summary*

In this chapter we have presented a means of generating grasp positions by first generating a set of all possible face pair combinations of the target object and then applying the criteria for valid grasping faces developed in previous chapters to filter the set. Next we presented a method for prioritizing the set of feasible face pairs for further processing and described a method for generating grasping points on each face pair. To the face pairs remaining the interference checks were applied as a last test for checking the validity of the generated configuration.

As a final note to this chapter, it seems realistic that grasps could be found on the convex hulls of objects if the relative size of the gripper to the parts is large. Typically we would not expect a small gripper to be used to pick up a much larger object. Additionally, as the number of facets used for approximating curved surfaces are increased to improve accuracy, the more parallel faces are available for gripping.

Chapter VIII

CONCLUSION

This thesis has addressed the problems involved with determining if a grip position for the parallel jaw gripper is a valid one when the objects are represented by a boundary model based on duality principles. The solution to the gripper/object interference problem was found for two classes of objects, convex and non convex and a third case involving the convex hulls of non convex objects. The interference problem was solved completely in the dual domain, minimizing reconstructing the object from the model.

We showed that convex objects have convex duals and this lead to simplified procedures for checking if a point was on a face and hence made checking grasp configurations easier. This property of having a convex dual warranted studying convex objects as a separate class of objects and for developing a method for calculating the convex hull of non convex objects from their dual representations.

Difficulties with determining if a point is on a face of an object with concavities required more complex procedures for checking grasp configurations. We presented a method of applying a corollary of the Jordan Curve Theorem directly in the dual domain to determine if a point was included on a non convex face. The procedure presented included a method for projecting an object's dual edges to obtain a two dimensional representation of the face and a method of determining if a point was on a bounded edge. The ability to check for point inclusion on a face gave us a means for checking that the criteria for face pair overlap and gripper pad/face contact in the concave cases were met.

A method of checking if two faces intersect which depended on being able to determine if a point is on a face was presented, but to use it for determining gripper/object intersections required checking every face of the gripper against every face of the object. While we assumed no prior knowledge about the objects being grasped, we did show that using knowledge about the gripper allowed us to model it as a difference of two convex pieces. This lead to the use of simpler methods derived for convex objects to simplify this checking for gripper intersections with concave objects.

Computational error analysis must now be studied to determine to what extend the developed algorithms will be affected by inaccuracies in such things as finger positioning and object location. Also of concern are the numerical ill conditions that occur when we try to calculate the duals of faces which lie in planes passing near the origin. This will have to taken into account when tolerances are set for determining which planes are parallel from the dual model representation.

While the bulk of the thesis dealt with analyzing grasp configurations to determine if they were valid, we also presented a means of synthesizing valid grasp configurations. As in much of the work in the field of path planning using other forms of boundary representations for object modeling, our method is a generate and test algorithm. The rules or heuristics we presented for generating points and final grip configurations were intended to give "intelligence" to the algorithm and there by reduce the amount of testing to be performed. Procedures were developed to prioritize possible points and configurations for further processing which adds to the intelligence of the algorithm by insuring that the first configurations to be tested "valid" were in some sense "better" then ones check later. The testing of the generated configurations was based on the procedures developed in the previous chapters.

The idea of what is a "better" grasp configuration is a topic of further research. In this thesis we only considered a pass/no pass type of criteria for evaluating a grasp with the exception of Chapter 7 where generated candidate grasps were sorted by how close they were to the center of mass of the object is to the center of the grip. This was done for reasons of stability. Much more work could be done on calculating the other dynamic properties of a grasp configuration in order to analyze an object's stability with respect to the gripper under various accelerations and torques as the object is worked on and moved about in the work space.

Another open area of research, not only for models based on duality but for any boundary representation, is the topic of path planning. It is of great interest for the gripping problem for two reasons. Generate and test algorithms are clumsy brut force methods that are computationally expensive. Methods for generating boundaries of free space for the gripper to move in would allow solving for all positions the gripper could use to grasp an object. Secondly, although our algorithm gives valid grasping configurations, it does not guarantee that the positions can be reached. A path must be found from the gripper's present position to the final grip position such that the gripper and the manipulator to which the gripper is attached, do not collide with the object or with other obstacles in the work space. Once methods to solve this problem are understood, what is considered an optimal grasp configuration may have to redefined as we consider what should be an optimal path/grasp trajectory combination.

Bibliography

1. Salisbury, J. K. (1982) "Kinematic and Force Analysis of Articulated Hands", PhD. Thesis, Stanford University Department of Mechanical Engineering, May.

2. Jacobson,Iversen,Knutti,Johnson,Biggers (1986) "Design of the UTAH/M.I.T Dextrous Hand" 1986 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, San Francisco, April 1986, pp. 1520-1532.

3. Kerr, J, and Bernard Roth (1986) "Special Grasping Configurations with Dexterous Hands", 1986 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, San Francisco, April 1986, pp. 1361-1366.

4. Nguyen, Van-Duc (1986) "The Synthesis of Stable Grasps in the Plane", 1986 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, San Francisco, April 1986, pp. 884-889.

5. Nguyen, Van-Duc (1986) "Constructing Force-Closure Grasps", 1986 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, San Francisco, April 1986, pp. 1368-1373.

6. Nguyen, Van-Duc (1987) "Constructing Force-Closure Grasps in 3-D", 1987 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, Raleigh, pp. 240-245.

7. Nguyen, Van-Duc (1987) "Constructing Stable Grasps in 3D", 1987 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, Raleigh, pp. 234-239.

8. Cutkosky, M.R. (1984) "Mechanical Properties for the Grasp of a Robotic Hand", The Robotics Institute Report CMU-RI-TR-84-24, Carnegie-Melon University, September 1984.

9. Fearing, R. (1984) "Simplified Grasping and Manipulation with Dexterous Robot Hands", M.I.T. A.I. Laboratory Memo 809, November 1984.

10. Jameson, J. W. and L. J. Leifer (1986) "Quasi-Static Analysis: A Method for Predicting Grasp Stability", 1986 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, San Francisco, April 1986, pp. 876-883.

11. Wolter, J.D., R.A. Volz, and A.C. Woo (1985) "Automatic Generation of Gripping Positions", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, Vol. 15 March/April pp. 204-213.

12. Baker, B.S., S. Fortune, and A.C. Woo (1985) "Stable Prehension with a Multi-fingered Hand", 1985 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, St. Louis, March 1985, pp. 570-575.

13. Mason, M. T. (1982) "Manipulator Grasping and Pushing Operations", M.I.T. A.I. Laboratory, AI TR 690 1982.

14. Brost, R.C. (1986) "Automatic Grasp Planning in the Presence of Uncertainty", PROCEEDINGS OF THE 1986 INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, pp. 1575-1581.

15. Hayward, V. (1986) "Fast Collision Detection Scheme by Recursive Decomposition of a Manipulator Workspace", 1986 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, San Francisco, April 1986, pp. 1044-1048.

16. Herman, M. (1986) "Fast Three-Dimensional, Collision-Free Motion Detector", 1986 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, San Francisco, April 1986, pp. 1056-1063.

17. Volpe, R. and Khosla, P. (1987) "Artificial Potentials with Elliptical Isopotential Contours for Obstacle Avoidance", 1987 IEEE DECISION AND CONTROL, Vol. 1, pp. 180-191.

18. Lozano-Perez, Thomas (1982) "Automatic Planning of Manipulator Transfer Movements" in ROBOT MOTION, pp.499-535.

19. Ghallab, Malik (1986) "Coping with Complexity in Inference and Planning Systems" in ROBOTICS RESEARCH, ed. Faugeras and Giralt 1986.

20. Rao, K., G. Medioni, H. Liu, G.A. Bekey (1988) "Robot Hand-Eye Coordination : Shape Description and Grasping", 1988 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, Philadelphia, pp. 407-411.

21. Feddema, J.T., Ahmad, and Shaheen (1986) "Determining a Static Robot Grasp for Automatic Assembly", 1986 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, San Francisco, April 1986, pp. 918-924.

22. Laugier, C. (1986) "A Program for Automatic Grasping of Objects with a Robot Arm", in ROBOT GRIPPERS, ed. D.T. Pham and W.B. Heginbotham, I.F.S. Publications Ltd.

23. Roach, John W., Praveen K. Paripati, Jeff S. Wright (1987) "A CAD System Based on Spherical Dual Representations", COMPUTER, Vol. 20, No. 8, August, pp. 37-44.

24. Huffman, D. A. (1977) "A Duality Concept for the Analysis of Polyhedral Scenes" in MACHINE INTELLIGENCE 8, Ellis Horwood Limited, Chichester, pp. 475-492.

25. Huffman, D. A. (1971) "Impossible Objects as Nonsense Sentences" in MACHINE INTELLIGENCE 6, Ellis Horwood Limited, Chichester, pp. 475-492.

26. Mackworth, A. K. (1973) "Interpreting Pictures of Polyhedral Scenes", ARTIFICIAL INTELLIGENCE, Vol 4, North-Holland Publishing Company, pp. 121-137.

27. Draper, S.W. (1981) "Dual Space and Line-Drawing Interpretation", ARTIFICIAL INTELLIGENCE, Vol 17, 1981.

28. Shafer, S. A. (1983) "Shadow Geometry and Occluding Contours of Generalized Cylinders", Carnegie-Mellon University Computer Science Department Report CMU-CS-83-131 a PhD dissertation, May.

29. Hilbert, D. and S. Cohn-Vossen (1952) GEOMETRY AND THE IMAGINATION, Chelsea Publishing, New York.

30. Wright, Jeff S. (1985) ANALZING SOLID OBJECTS FOR GRASPING, Honors Thesis for BSEE from VPI & SU, Blacksburg, VA.

31. Arnold, B. H. (1962) INTUITIVE CONCEPTS IN ELEMENTARY TOPOLOGY, Prentice-Hall, Inc. Englewood Cliffs, N.J., pp. 91-92.

Vita

Mr. Dannhardt was born on Long Island, NY in 1961. He grew up in Bristol, VA graduating from high school in June 1980.

Mr. Dannhardt attended Virginia Polytechnic Institute and State University (Virginia Tech) for his undergraduate work and received his Bachelor of Science in Electrical Engineering in June 1984.

Mr. Dannhardt worked for General Motors' Saginaw Division in Saginaw, Michigan from July 1984 to June 1986. At Saginaw Division he was responsible for machine automation and factory monitoring systems. He returned to Virginia Tech in September 1986 to pursue his Master of Science in Electrical Engineering. He now works for USDATA in Reston, VA. on the development of man machine interfaces and their applications.

Michael Dannhardt