Planning Robot Grasping Motions in the Presence of Uncertainty

Randy C. Brost

CMU-RI-TR-85-12

Computer Science Department and
The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

July 1985

Support for this research was provided by the System Development Foundation, the National Science Foundation, and the Carnegie-Mellon Department of Computer Science.

List of Figures

ABSTRACT

This paper presents a method for automatic planning of robot grasping motions that are guaranteed to succeed, despite bounded variations in the object's location. The method capitalizes on the physics of friction to generate the space of all guaranteed grasp plans that utilize either a squeeze-grasp, offset-grasp, or push-grasp motion. All plans that are found are guaranteed to succeed, even if the worst-case error occurs. From this space of guaranteed plans, a plan can be chosen and executed without sensing or feedback. Executing the motion removes two degrees of uncertainty from the object's position. For simplicity, planar motion of the object during grasping is assumed.

I. Introduction

Uncertainty presents a key problem in manipulation. Two major sources of uncertainty affect manipulator operations: world uncertainty and manipulator uncertainty. World uncertainty is the uncertainty in the position and orientation of objects in the world, while manipulator uncertainty is the uncertainty in the position and velocity of the manipulator.

Most present approaches to robotic manipulation attempt to eliminate the problems associated with uncertainly by eliminating uncertainty itself. World uncertainty is often reduced by confining the robot to a carefully controlled environment, where objects arc constrained to precisely-known locations, or by employing sophisticated sensors that precisely measure the location of objects in the world. Both methods are expensive, and may fail if the precision of the controlled environment becomes corrupted. Manipulator uncertainty can be reduced by building stiffer, more precise mechanisms and control systems for manipulators. However, increasing the precision of a manipulator increases its cost.

In contrast, the underlying philosophy of our research is to develop manipulation operations that work well despite significant uncertainty. We would like to develop algorithms to plan actions that will be guaranteed to succeed, given that the world and manipulator uncertainty are within known bounds. Such actions would operate effectively in a more natural environment. Instead of trying to eliminate uncertainty, we seek methods that will succeed despite uncertainty.

This paper applies this philosophy to the problem of grasp planning. For simplicity, we assume that objects to be grasped axe polygonal prisms lying on a table, constrained to planar motion during the grasping move. In addition, we assume coulomb friction exists at all contacts, and that speeds are slow enough that frictional forces dominate inertial forces. No assumption is made about the distribution of support forces between the object and the supporting plane.

Within this domain, we present an automatic method for producing grasping plans that are guaranteed to succeed even though significant uncertainty is present. No sensing is required, and the plans produced by the method also remove some of the uncertainty in the planar object's orientation and position. The planner described in this paper has been implemented and demonstrated using a Puma 560 manipulator.

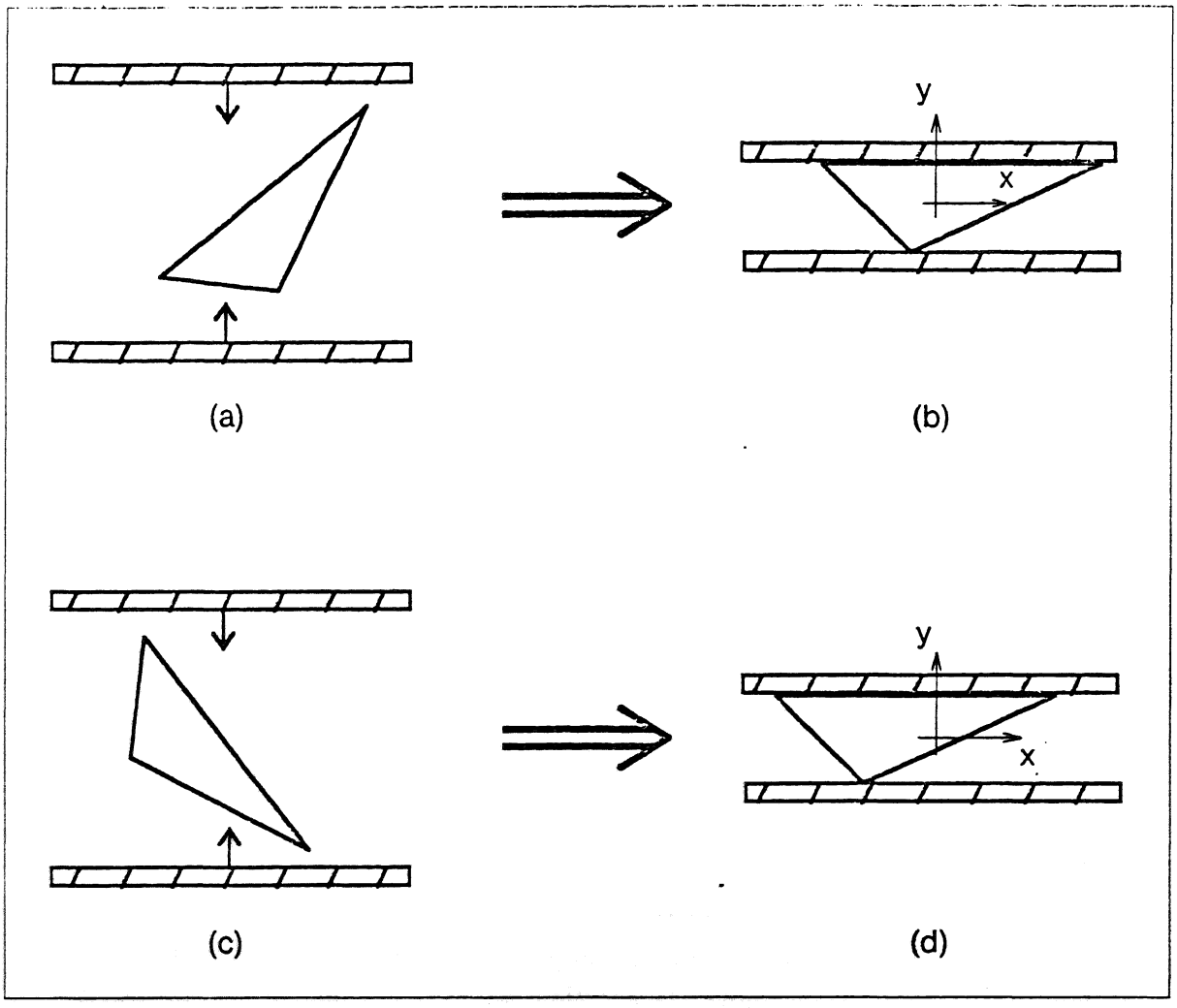


Figure 1. An example grasp plan. Due to position and orientation uncertainty, the triangle might be in either of the configurations illustrated in (a) or (c). Both of these configurations are subjected to the same squeeze-grasp motion, resulting in the final states (b) and (d). In both cases, two degrees of uncertainty (orientation and y-axis position) have been successfully removed; only uncertainty in the x-axis position remains.

Example

An example plan is shown in Figures 1 and 2.

The object to be grasped is a triangular block, lying flat on a tabletop. The exact position and orientation of the triangular block are unknown; however, the block is known to lie within certain limits. There is a continuum of possible block positions within the known limits; two such positions are shown in parts (a) and (c) of Figure 1. The blocks are subjected to identical squeeze-grasp motions, and in both cases the end result is the same: each block has rotated slightly, causing it to be held flat between the gripper fingers.

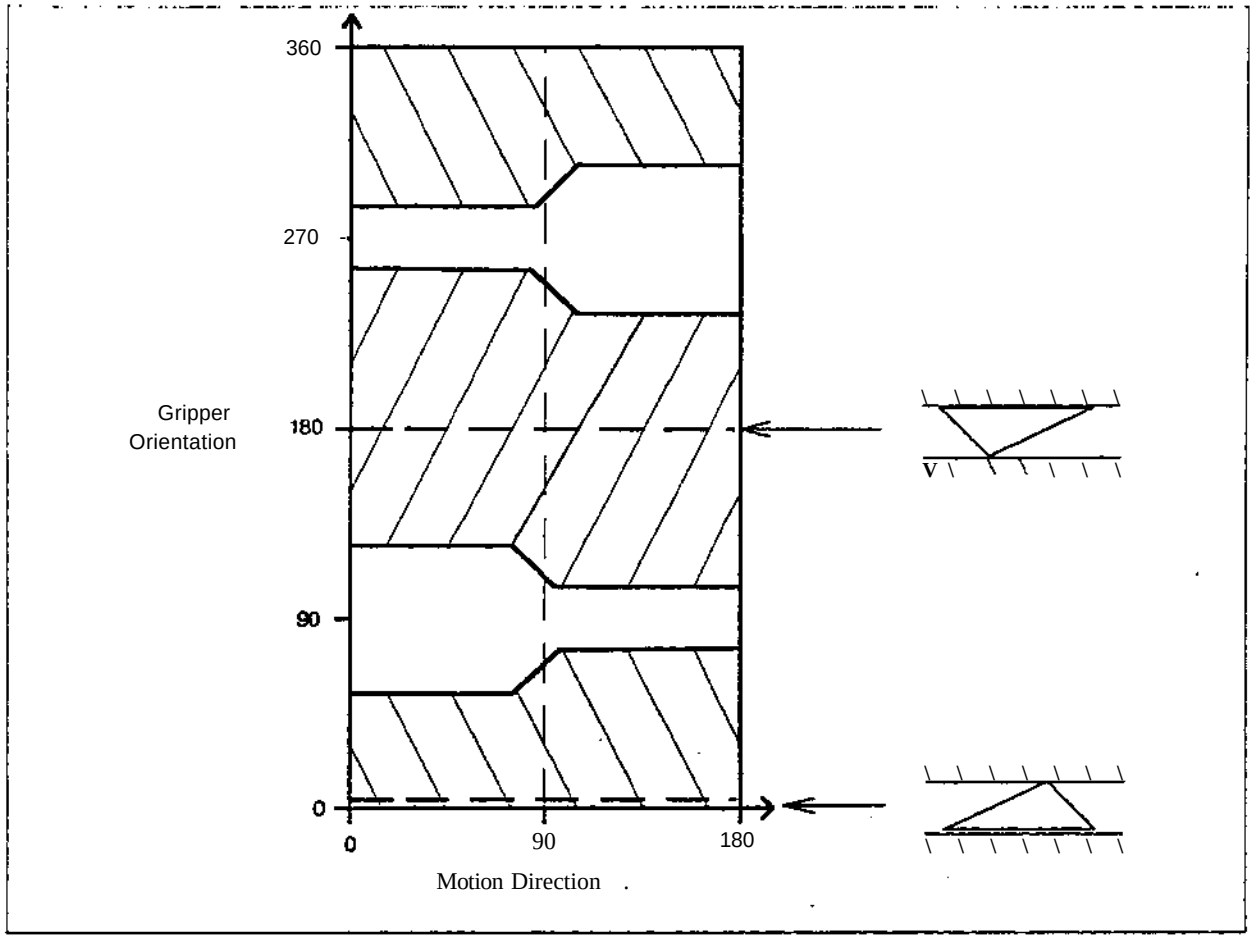


Figure 2. The squeeze-grasp diagram for the triangle. The shaded regions indicate those squeezing moves that are guaranteed to succeed in acliieving the final configurations shown on the right.

A pair of x-y coordinate axes have been added to the gripper to help clarify this explanation. Notice that after the plan is executed, the location of the block is partially determined; while its position in the x-direction is still uncertain, the block's orientation and position in the ^-direction are fully constrained to coincide with the gripper fingers. Thus, after the plan is executed, two degrees of uncertainty in the block's position have been removed; again, note that no feedback or sensing was required.

Our method planned this grasp by using the geometric description of the triangle to produce the *squeeze-grasp* diagram shown in Figure 2. In this diagram, the orientation of the squeezing fingers varies along the vertical axis, and the finger motion direction varies along the horizontal axis. These axes combine to define the space of all possible squeese-grasp motions. Within this diagram* the shaded regions indicate those **squeeze**-grasp moves that are guaranteed to succeed. The heavy dashed lines correspond to the final grasp configuration for each region; these final configurations are illustrated at the right of the diagram.

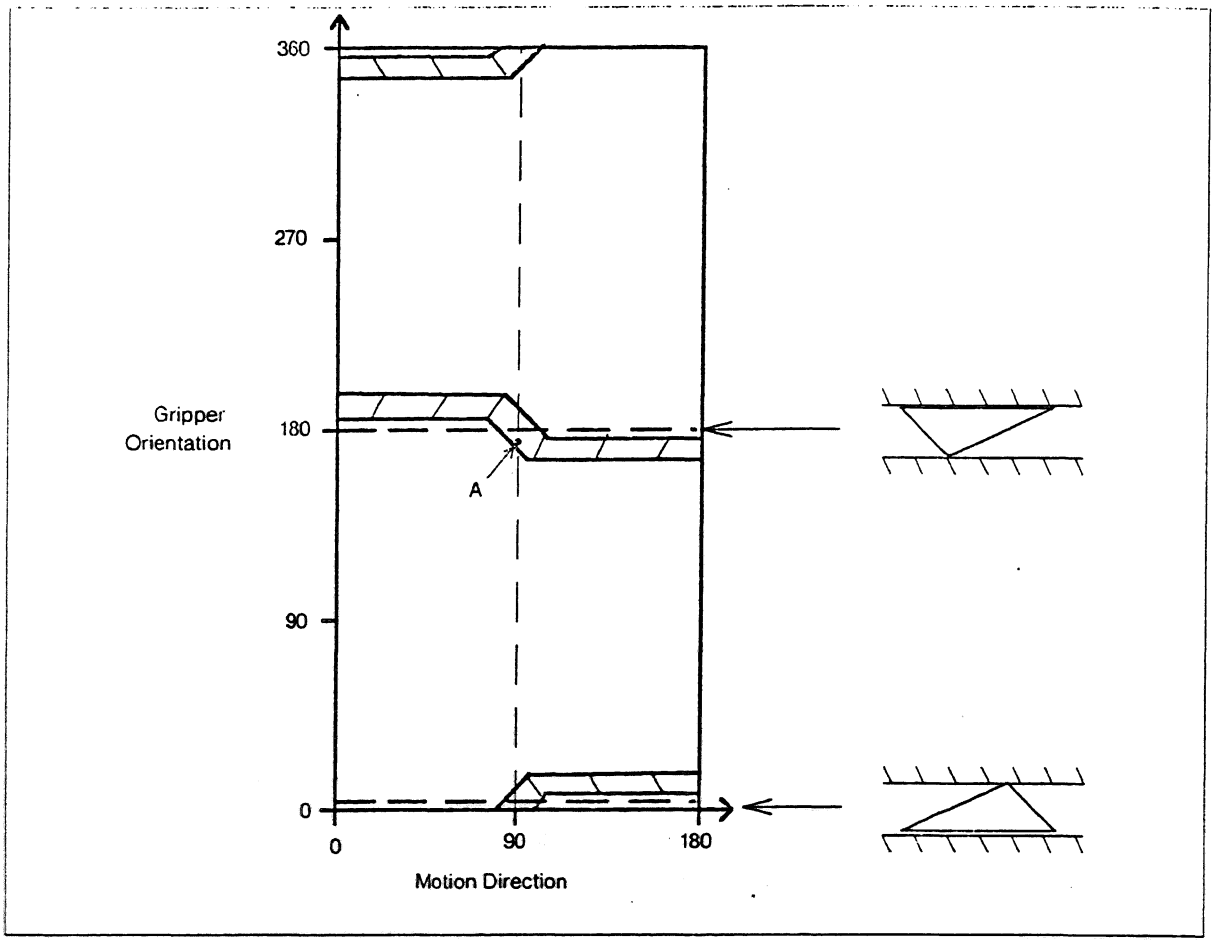


Figure 3. The squeeze-grasp diagram for the triangle, after shrinking for uncertainty in the world and in the manipulator. The shaded regions remaining in this diagram indicate those moves that are guaranteed to succeed, even if the worst-case combination of uncertainties occurs. In this case, the shrinking process assumed that the uncertainty in the polygon's orientation is less than $\pm 50^{\circ}$, the uncertainty in the manipulator's execution of moves is less than $\pm 5^{\circ}$, and the coefficent of friction μ is within the interval $(0.15, 0.35)$. This diagram was used to plan the squeezing move in Figure 1; the move that was chosen is indicated by the point A.

The diagram shown in Figure 2 does not take uncertainty into account. We can compensate for uncertainty by shrinking the regions in the squeeze-grasp diagram by the amount of uncertainty that is present; the remaining regions represent the space of all squeeze-grasp moves that are guaranteed to succeed, even if the worst-case uncertainty is encountered. The result of shrinking the regions in Figure 2 to account for various sources of uncertainty is shown in Figure 3.

The plan shown in Figure 1 represents an example of a squeeze-grasp motion; our method finds guaranteed grasp plans that utilize squeeze-grasp, offset-grasp, and push-grasp motions, which will be explained in detail later. A *guaranteed grasp plan* uses one of the motions mentioned above, assumes only that the initial location of the object is

contained within certain limits, and guarantees that after the plan is executed, the object will be successfully grasped with the uncertainty in the object's location removed.

Within the domain of our assumptions, the method we present finds the space of all guaranteed grasp plans that are possible for any given grasping task. This solution space is computed exactly, without using any simplifying approximations. After the method has computed all possible guaranteed plans, a plan can then be chosen from this space according to other considerations (e.g., obstacle avoidance, untouchable object faces, etc.).

Previous Work

An analysis of the mechanics of friction during grasping operations is fundamental to this paper. Several previous papers have also investigated the effect of friction on the motion of a manipulated object.

Mason's Ph.D. thesis provides the basis for this work, by proving the pushing results described later in Section II. In addition, Mason developed a procedure that found a guaranteed push-grasp operation for a polygon, given the description of the polygon and a range of possible polygon orientations. Mason's procedure returns a specification of a grasping operation that includes a pushing plane orientation, pushing direction, and minimum pushing distance required for the operation to succeed. Mason's procedure returns plans that have guaranteed results, but only addresses a restricted class of polygon grasping problems [Mason 1982; Mason 1984].

While this paper investigates grasping operations using two planar fingers, Fearing investigated grasping operations that used two point fingers. Fearing included the effect of friction in his analysis, and succeeded in predicting the motion of an object under two-point squeezing, and the effect of disturbing forces on an object being held in a two-point grasp [Fearing 1983; Fearing 1984].

Mani and Wilson independently derived a diagram similar to the push-stability diagram of this paper, and used their result to design a device that orients parts being fed to a manufacturing process. Mani and Wilson's method uses the diagram to produce a tree of orienting pushes, and then searches the tree to find a series of pushing moves that will reduce the number of possible polygon orientations to one [Mani and Wilson 1985].

A key feature of the method presented in this paper is that it finds grasp plans that are guaranteed to succeed in the presence of bounded uncertainty. In this respect, this research fits nicely into a series of planning systems previously developed which attempt to plan robot actions despite the presence of uncertainty.

Lozano-Pérez outlined a planning system that created an assembly plan and then used a feedback planner to check the plan for errors arising from the task geometry or

uncertainty. The feedback planner would then modify the original plan to correct the errors [Lozano–Pérez 1976].

Brooks described a planning system that kept track of uncertainty bounds at various stages of a plan's execution, and then checked the resulting plan to determine whether or not the uncertainty at any stage in the plan was too large. If so, then Brooks' system determined whether the addition of sensing or other plan constraints could reduce the uncertainty sufficiently to allow the plan to succeed [Brooks 1982].

Lozano-Pérez, Mason, and Taylor described a method of planning fine motions that are guaranteed to succeed in achieving a final goal. Their Fine-Motion Planner (FMP) achieves a higher-level goal by determining a chain of subgoal regions that can be reached despite the presence of uncertainty in the world and in the robot's execution of motions [Lozano–Pérez, Mason, and Taylor 1984]. Mason showed that this planner is both correct and complete [Mason 1983].

Erdmann showed that Mason's complete planner was not computable in the general case, but demonstrated an implementation of a weaker planner that worked in a similar fashion using backprojections. Erdmann also investigated the mapping of three-dimensional friction cones into configuration-space, and the effect of multiple frictional point contacts on the forces applied to an object [Erdmann 1984].

This paper determines all of the stable grasp configurations for a polygon being held by a parallel-plane gripper. Several other papers have also addressed the problem of finding stable grasp configurations.

In his thesis, Paul planned grasp configurations by applying heuristics that preferred grasping the object by opposing parallel faces along an axis that contained the center of mass [Paul 1972]. Later, Bolles and Paul incorporated simple tactile feedback to sense the location of an object when its location was uncertain [Bolles and Paul 1973].

Brou uses a volume representation comprised of blocks and cylinders to model objects to be grasped, and then develops a method of finding collision-free grasp positions and grasp approach angles for picking up a modelled object [Brou 1981]. Peshkin and Sanderson used a convex-rope algorithm to determine the finger orientations that can be used to approach a given edge or vertex of a non-convex polygon without causing a collision [Peshkin and Sanderson 1985].

Wolter, Volz, and Woo addressed the problem of automatically generating strong grip positions and characterized grip strength in terms of resistance to slipping and twisting under the application of an external force [Wolter, Volz, and Woo 1984]. Holtzmann and McCarthy formulated the static equilibrium equations for an object being grasped by three point fingers. From these equations, they were able to calculate the applied forces required

to maintain stable grasping, and determine whether a given grasp configuration would slip or remain stable due to friction [Holtzmann and McCarthy 1985].

Hanafusa and Asada investigated the conditions required for stable prehension of an object, and the resulting implications for force control of a dextrous robot hand [Hanafusa and Asada 1977]. Salisbury investigated the kinematic and force control requirements for optimum design of a dextrous hand for grasping objects in a variety of configurations [Salisbury 1982; Salisbury 1983].

Cutkosky examined how various finger surfaces and gripping configurations affect grasp stability during the execution of typical manufacturing operations. From this analysis, Cutkosky was able to derive some useful rules for the design of robot hands and wrists, and for choosing optimal grasps [Cutkosky 1985].

Baker, Fortune, and Grosse discovered that a five degree of freedom, three-fingered hand can stably grasp any convex or non-convex polygon by touching the maximally-inscribed circle fit within the polygon [Baker, Fortune, and Grosse 1985].

Asada and By investigated stable prehension of objects in the context of workpart fixturing. By placing fixtures and clamps in an appropriate configuration, they were able to fully constrain the position of a workpart. Their fixtures are analagous to fingers of a multi-finger robot hand; both are used to achieve a stable grasp of an object [Asada and By 1985].

Overview

This paper will proceed by developing the method that is described above. Section II describes our notation and the physics that the method relies upon. Section III explores the motion of a polygon being pushed by a single finger; Section IV characterizes the motion of a polygon when squeezed by two opposing fingers. Section V describes some of the circumstances where a simple squeeze-grasp fails, and defines alternative grasping strategies that succeed in these situations. Section VI discusses how the presence of uncertainty can be incorporated into the planning analysis. Finally, Section VII concludes the analysis, describes a working implementation of the system, and offers some suggestions for further research.

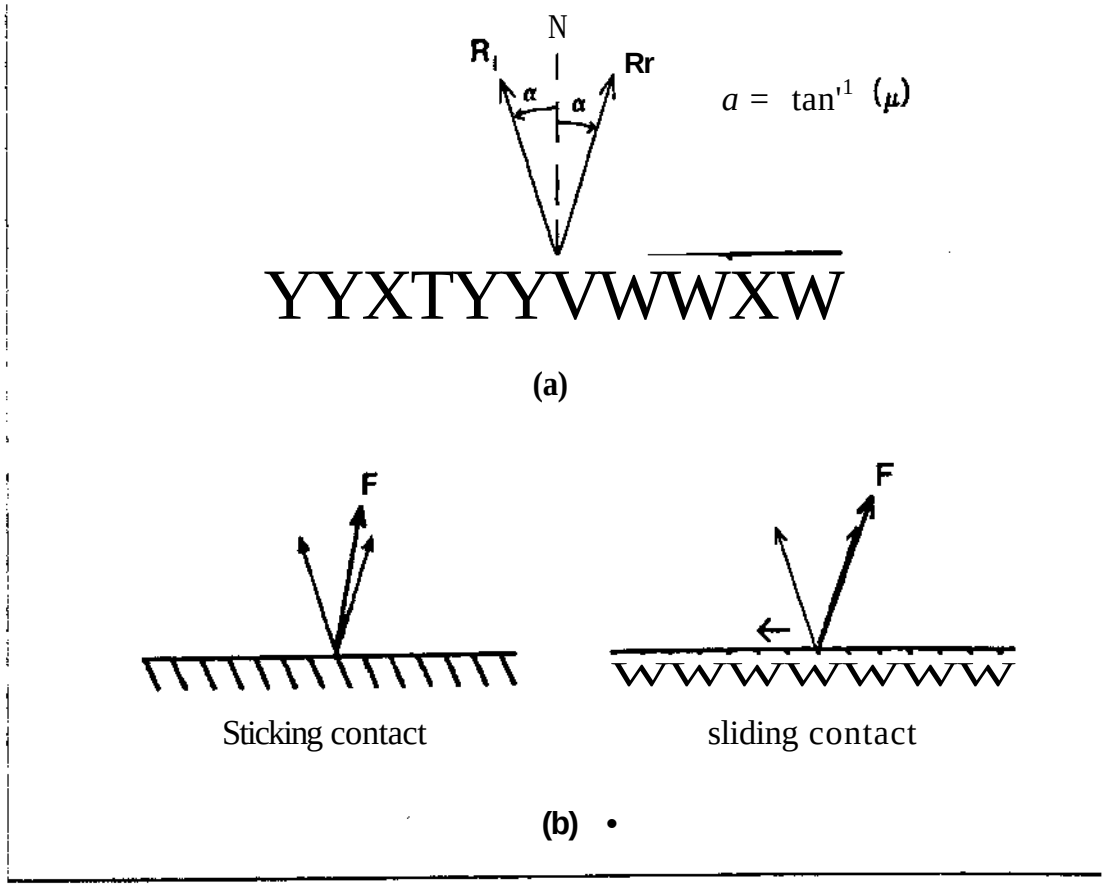


Figure 4* Frirtion *cmmn,* At any point-plane contact, tlio resulting friction cone Ls bounded *nu* rsther *mit* by flic ray* Ri ami R_r, which Be an angle *a* away from tlie normal to the pkr*«\ *m* illft*f srateci k (a). If the contact is sticking to the surface, then the applied force ik* m*f Ifiii thi* *bktkm* COM; if the contact is aiding along the surface, then the applied force *hm* «i liic* «lfe of *the* friction **cone (b).**

IL Taik Mechanics **and Notatkn**

Friciien Cests

Before wt CA» explore tic kleraclioii of a robot and the objects in its surround **workl** /luring grasping, a few physical concepts must be reviewed. First, we will exam the friction cent (Figure 4), Coulomb's law implies that at every instance of point-pl **contact**, a fnciits **coat existJ** tfaat dictates how the frictional contact will react under **application** of ao **exterior force***

T:;r fration cost *m* **defined** by an **angk a** to each side of the surface normal of pi[illegible] o kf tfef **arc**tan of tlie C0elcient of friction *p.* Applied forces that fall **wit)** t[illegible] ftre within tlte friction co»e. Coulomb's law states that if the frictio *ti* [illegible] *Mtkmg* to lie *mtlme* and not sliding, then the applied force lies within !

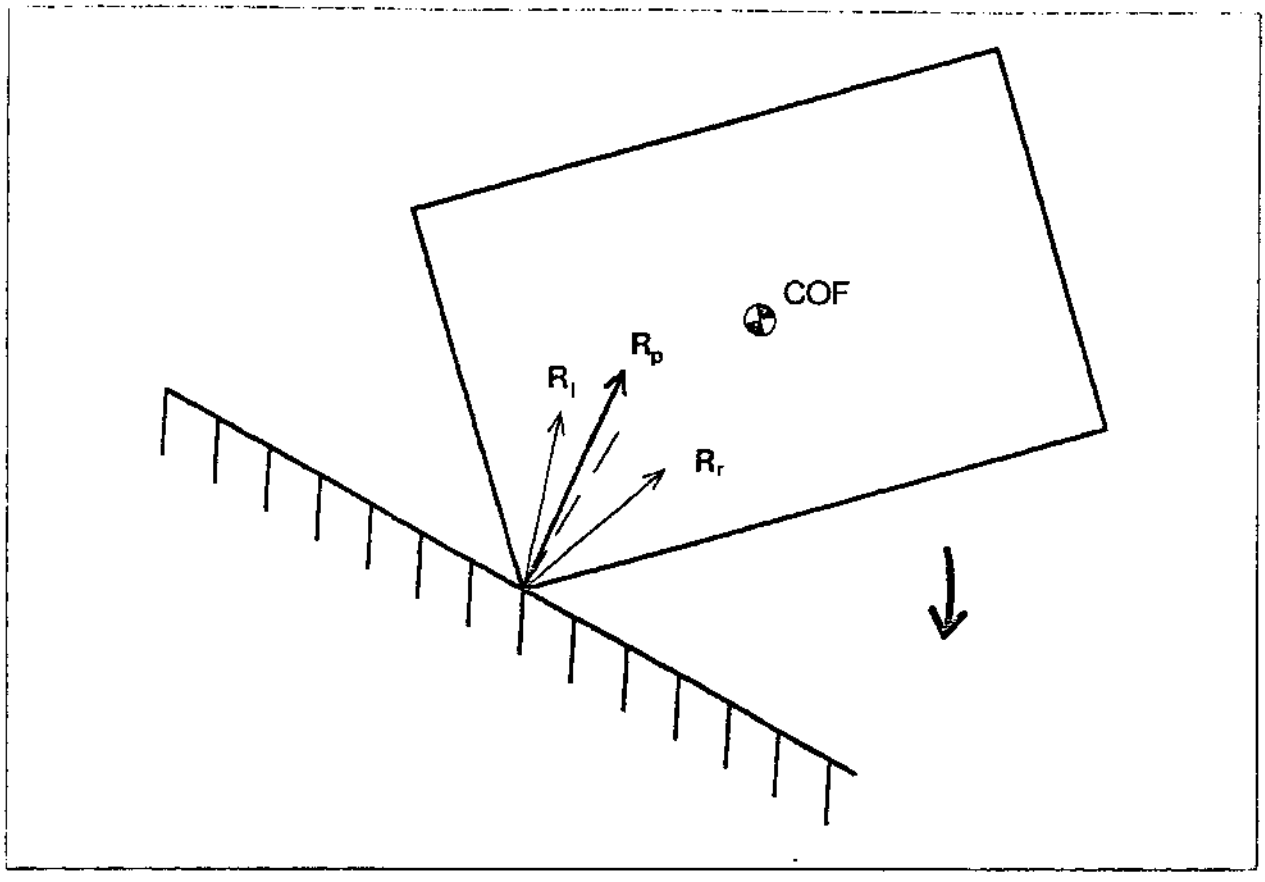


Figure 5. Rotation under pushing. The throe rays Rj, R_p, and R_r vote relative to the center of friction (OOF) to determine the rotation direction. Since R_l <uid R_p both lie to the left of the COF, they win the vote, and the rectangle rotates clockwise. Whether there is sliding or sticking contact between the corner and the pushing plane remains undetermined. Ri and R_r are the left and right rays of the friction cone, respectively, and R_p is the ray of pushing.

friction cone. If the contact is sliding, then the applied force lies on the edge of the friction cone opposite the sliding direction (Figure 4(b)).

Pushing

Next we must consider the motion of an object being pushed. One common problem in grasping operations is: If one object is pushing another along a flat surface, what direction will the pushed object rotate? Under certain assumptions, Mason [Mason 1982; Mason 1984] fonnd a simple answer to this question by isolating three rays (Figure 5):

R_p — ray of pushing (i.e., the direction the pusher is moving)

R_t — left ray of the friction cone

R_r •— right ray of the friction cone

Those three rays determine the rotation direction by voting relative to the center of friction (COF). The center of friction of an object lies at the ccntroid of the object's supporting pressure distribution; for the purposes of this paper, the center of friction can be thought of as the object's center of gravity.

If two or more of the three rays lie to the left of the COF, then the pushed object will rotate clockwise (as shown in Figure 5). If two out of three rays lie to the right of the COF, then the pushed object will rotate counter-clockwise. If one of the three rays points directly at the COF and the other two rays are on opposite sides of the COF, then the pushed object will translate without rotating. This rule applies regardless of the pressure distribution supporting the object; if a flat-bottomed object and *an* object supported by an arbitrary tripod of points were subjected to the same pushing operation, they would both rotate the same way. Note that this rule docs not determine the rate of rotation, but only the direction; the rate of rotation can vary with different pressure distributions. Whether or not the contact is sticking or sliding contact is also indeterminate, but the voting rule holds in either case. Hence, as long as we adhere to Mason's assumptions (outlined below), we can utilize this simple method to determine the rotation direction of a pushed polygon.

Assumptions

In order to assure that the above physical models are valid and simplify the problem of finding reliable grasp plans, we will make a few assumptions:

- Coulomb's law is assumed. This includes the friction between the robot's gripper finger(s) and the object, as well as the friction between the object and its supporting surface. We do not assume that the coefficient of friction /x is the same for these two sources of friction.
- Rrktional forces dominate kertial forces. This assumption is valid for sufficiently slow motions.
- Objects to be grasped are lying on a level planar surface.
- No assumption is made about the distribution of contact forces between the object and the supporting plane.
- # Objects axe essentially ^*Im*^*x*.[19] This implies two restrictions on acceptable objects: First, when the object is pushed, torques that tend to roll the object off of its support face are small compared to the lateral sKding resistance offered. Second, the faces of the object that will be in contact with the pushing plane must be perpendicular to the support face (e«g., a truncated pyramid would not be acceptable because its side faces axe slanted).

Subject to these assumptions, grasping motions reduce to planar problems. We will adopt these assumptions for the rest of the paper; our results will be guaranteed in the real world as long as they are maintained.

For convenience, we will make one further simplifying assumption:

- The gripper fingers of the manipulator will be assumed to be of infinite extent.

This assumption allows the manipulator fingers to be modelled as infinite half-planes in the two-dimensional grasping abstraction, which is helpful in the following ways:

(1) All polygons are essentially convex, since non-convex polygons are equivalent to their convex hulls when pushed or squeezed by an infinite half-plane. Thus concavities in an object do not offer a complication.

(2) Contact with a corner of a finger does not have to be considered.

This infinite half-plane assumption can be met by using the extent of the object to calculate the minimum finger width required to make a finite finger effectively seem like an infinite half-plane. If the manipulator gripper finger meets or exceeds this minimum finger width, then the pushing move may be executed in the real world with guaranteed results.

Once these assumptions are met, the complex physical problem of real-world grasping can be abstracted into the simpler problem of understanding the motion of a polygon sliding along a plane under the influence of a few well-defined rules. This abstracted problem can then be pursued with a purely geometric analysis.

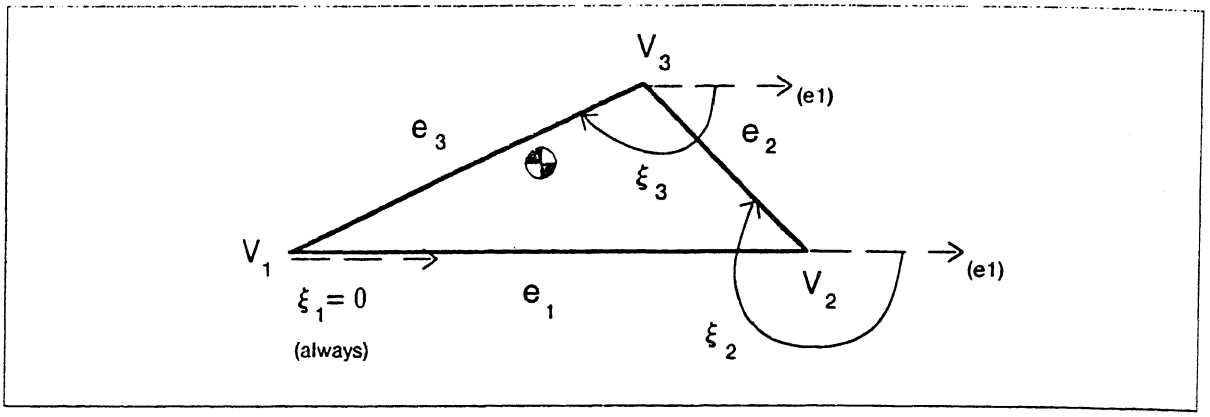


Figure 6. Polygon notation.

Notation

The notation that will be used throughout this paper to describe polygonal objects is illustrated in Figure 6.

The edges of the polygon are numbered $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, ... etc., from some chosen starting edge, increasing counter-clockwise around the polygon. The vertices are numbered in a similar fashion, so that for every edge $\mathbf{e}_i$, vertex $\mathbf{V}_i$ is on the clockwise end of the edge, and vertex $\mathbf{V}_{i+1}$ is on the counter-clockwise end of the edge (see Figure 6). By convention, we will always choose $\mathbf{e}_1$ to be the longest edge of the polygon.

Sometimes it is useful to think of an edge $\mathbf{e}_i$ as a vector $\overrightarrow{\mathbf{V}_i\mathbf{V}_{i+1}}$; when this is the case, we will denote the edge $\vec{\mathbf{e}}_i$. Each edge $\mathbf{e}_i$ has an edge-angle ξ_i, which is the angle measured clockwise from $\vec{\mathbf{e}}_1$ to $\vec{\mathbf{e}}_i$ (See Figure 6).

The triangle illustrated in Figure 6 will be used in subsequent examples throughout this paper. Notice that the COF of the triangle has been intentionally chosen in an unusual position. This unusual choice was made for two reasons: (1) The examples are more interesting with the COF in an unusual position, and (2) this choice demonstrates that the method does not assume that objects are of uniform density.

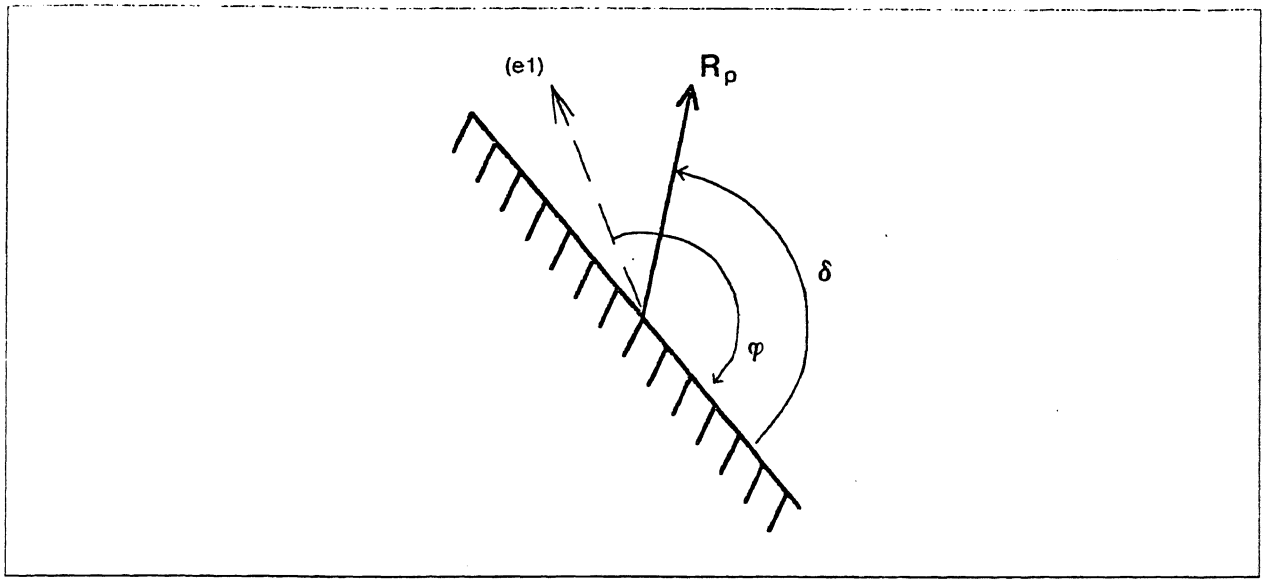


Figure 7. Pushing notation.

III. Pushing with One Plane

In all of the grasping operations considered in this paper, pushing plays an important role. For example, if the part is displaced slightly from its assumed location, then the robot's gripper fingers will not make simultaneous contact with the object. As a result, there will be some period of time when the object is pushed by one finger before the second finger makes contact. Also, one might envision grasping operations where the robot deliberately pushes the object to align it before attempting a grasp. Therefore, it is worthwhile to understand how an object behaves while it is being pushed, and our polygon/infinite half-plane abstraction lends itself well to this problem. The purpose of this section is to develop an understanding of pushing, and to characterize the outcome of various pushing operations on any given polygon.

To investigate polygon pushing, we will divide our analysis into two parts, which are characterized by the following questions:

- Where on the polygon will the pushing plane make first contact?
- How will the polygon rotate while it is being pushed?

To rigorously approach these two questions, it is necessary to formalize our notion of pushing. We will begin by defining a single pushing move as follows: A half-plane finger starts with some orientation at some start point and moves in a constant direction for some pre-defined distance. The orientation of the finger plane and the direction of pushing are held constant throughout the move.

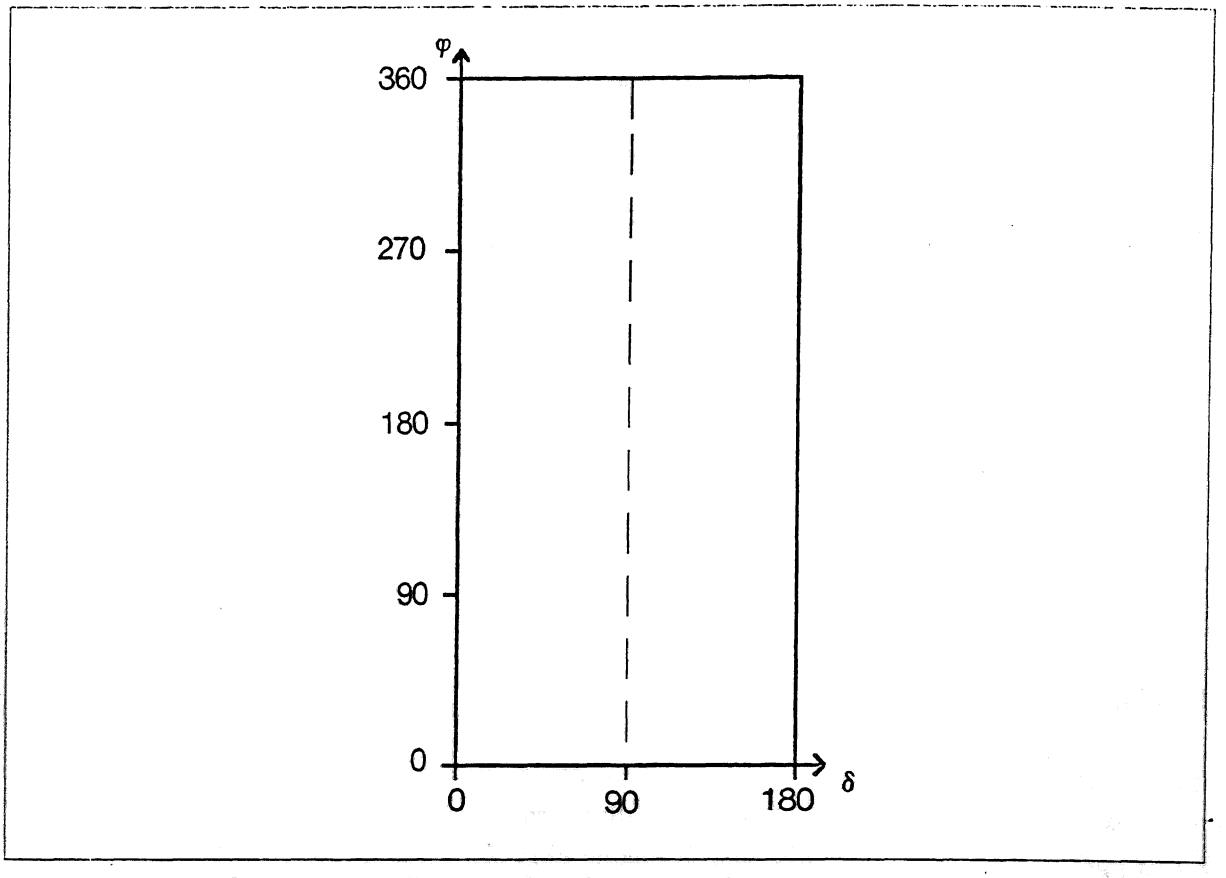


Figure 8. Pushing space. The rectangle defined by the δ-ϕ axes contains all possible pushing moves; the dashed line down the center of the pushing space corresponds to "natural" pushing moves, where $\delta = 90$.

The angle ϕ is the finger orientation, measured relative to edge $\mathbf{e}_1$ of the polygon (See Figure 7). The angle δ is the direction of pushing, measured relative to the finger plane. ϕ is the angle measured clockwise from $\overrightarrow{\mathbf{e}_1}$ to the open face of the pushing plane, while δ is measured counter-clockwise from the pushing plane to $\mathbf{R_p}$, the ray of pushing.

Note that ϕ is measured relative to the *polygon's* coordinate system; this coordinate system is not fixed. Therefore, since the polygon may rotate while being pushed, ϕ can vary during the pushing move even though the orientation of the plane constant throughout the move.

However, since the pushing direction δ is measured relative to the pushing plane instead of the polygon's coordinate system, δ stays constant throughout the pushing move, even if the polygon rotates. Thus angle δ is *invariant* throughout the pushing move. This fact that δ remains invariant during a pushing move will become significant later on.

The set of all possible pushing operations forms a well-defined *pushing space*. This pushing space can be represented as a rectangular coordinate system with δ as the horizontal axis and ϕ as the vertical axis (Figure 8). δ ranges from 0 to 180 degrees, while

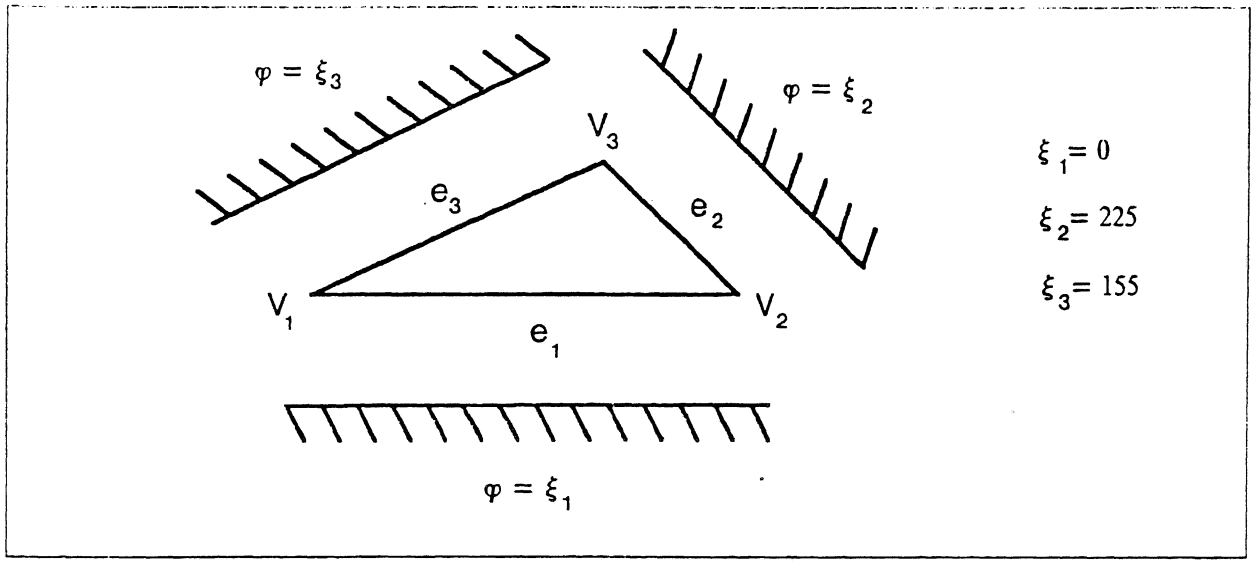


Figure 9. Edge-flat orientations. The pushing planes illustrate the three plane orientations that will make first contact flat against an edge of the triangle. Since each edge-flat plane orientation ϕ is exactly equal to its corresponding edge-angle ξ_i, the ξ_i edge-angles can be directly used to determine the edge-flat lines in the pushing space.

ϕ ranges from 0 to 360 degrees; all pushing operations correspond to a point somewhere within this rectangle (δ values greater than 180 degrees do not correspond to valid pushing operations, since the ray of pushing would point into the pushing plane). The dashed line down the center of the rectangle is the $\delta = 90$ line, which corresponds to those "natural" pushing operations where the ray of pushing is perpendicular to the pushing plane. Note that start position and pushing distance are not included in the pushing space; determining the value for these parameters will be addressed in a later paper.

Where Will the Plane Touch First?

The point where the plane first contacts the polygon depends entirely on the orientation of the plane, and not the direction of pushing. Thus first-contact regions are delimited only by values of ϕ in the pushing space, independent of δ, and we may proceed by only considering the effect of changing the value of ϕ in a pushing operation.

Figure 9 shows a triangle surrounded by three infinite half-planes; each plane is oriented so that it will make flat contact with one of the triangle's edges as the plane approaches the triangle. These special orientations are called the *edge-flat orientations* of the polygon, which correspond to *edge-flat lines* of constant ϕ in the pushing space. Because of the way we chose our notation, the orientation ϕ of each edge-flat plane is exactly the same as the corresponding edge-angle ξ_i; in other words, the polygon's edge ξ_i values determine all of the edge-flat lines in the pushing space.

This gives us the lines in the pushing space where first contact is made with the plane flat against an edge of the polygon. For pushing operations where the value of ϕ is between

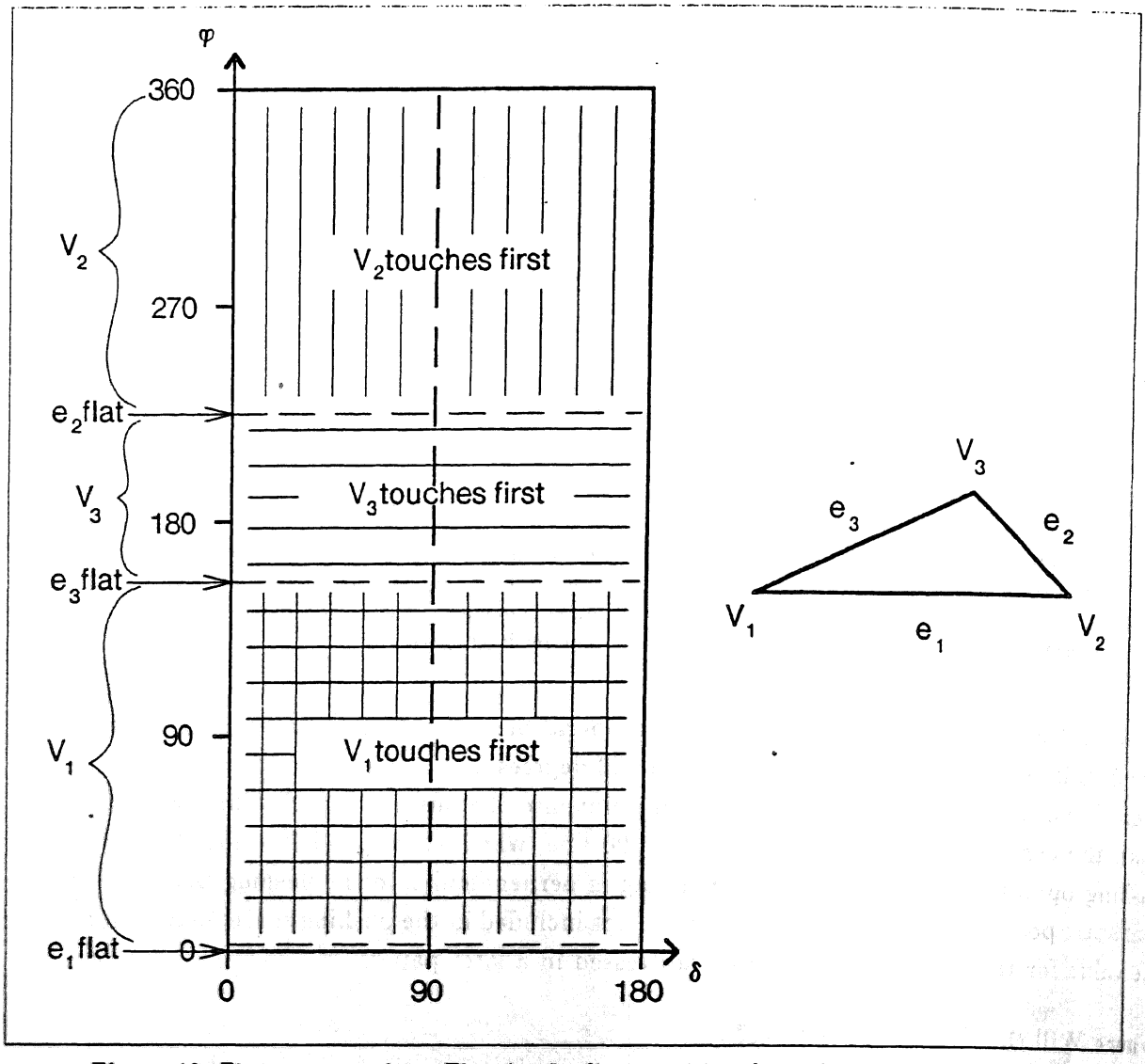


Figure 10. First-contact regions. The edge-flat lines partition the pushing space into bands of vertex first contact.

ξ_{i-1} and ξ_i, first contact is made with the vertex $\mathbf{V}_i$. Thus for all values of ϕ in between the edge-flat lines, contact is made with the polygon vertices, giving us bands of vertex first contact (Figure 10). These bands, combined with the edge-flat lines, completely fill the pushing space; therefore we know the point of first contact for all possible pushing operations.

How Will the Polygon Rotate?

Now that we have isolated regions of first contact in the pushing space, we can analyze the motion of the polygon within any particular contact region. We will first consider the effect of the the friction cone rays $\mathbf{R}_l$ and $\mathbf{R}_r$, and then consider the effect of the ray of pushing $\mathbf{R}_p$.

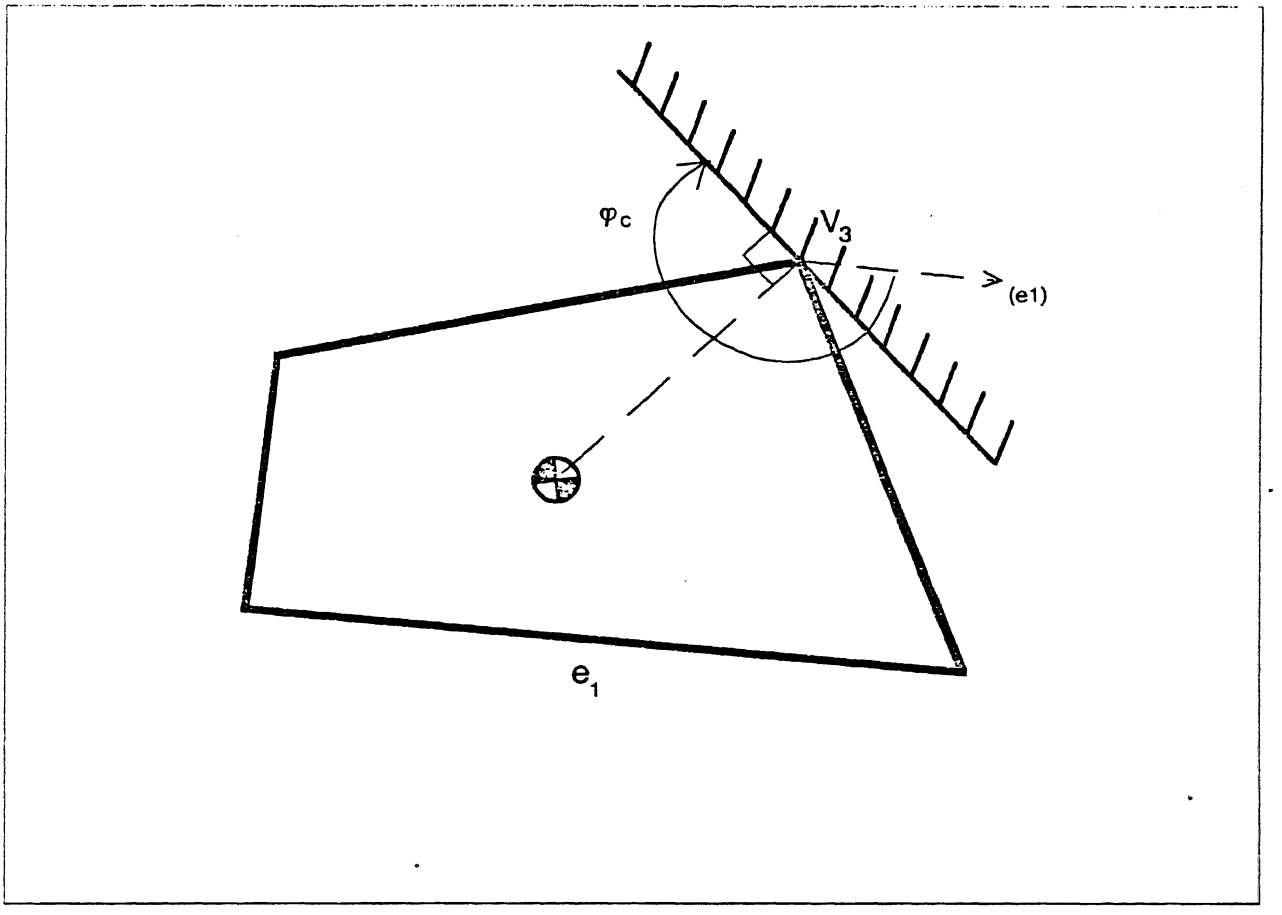


Figure 11. The critical pushing finger orientation ϕ_c. The angle ϕ_c is the plane orientation where the pushing plane normal points directly at the COF. In this case, ϕ_c is for vertex V_3.

For any particular vertex contact region, we will define the critical pushing orientation ϕ_c. ϕ_c is the plane orientation where the plane normal points directly at the COF (Figure 11).

Consider the case of pushing contact illustrated in Figure 12(a); the corresponding first-contact region in the pushing space is shown in Figure 12(b). Recall that the three rays vote to determine the rotation direction. Our strategy is to consider each ray individually and determine how its vote is distributed throughout the contact region; once we have considered all three rays, we can tally the votes to determine the overall rotation direction as it varies within the contact region.

First, consider the left friction cone ray $\mathbf{R}_l$, which lies an angle α away from the pushing plane surface normal. $\mathbf{R}_l$ sweeps back and forth as ϕ varies; notice that when the value of ϕ exceeds $\phi_c + \alpha$, $\mathbf{R}_l$ crosses the dotted line between $\mathbf{V}_1$ and the COF, and $\mathbf{R}_l$'s vote changes from clockwise to counter-clockwise. Thus, $\phi_c + \alpha$ is the *decision value* for $\mathbf{R}_l$. We can use this decision value to split the contact region into two subregions: one where $\mathbf{R}_l$

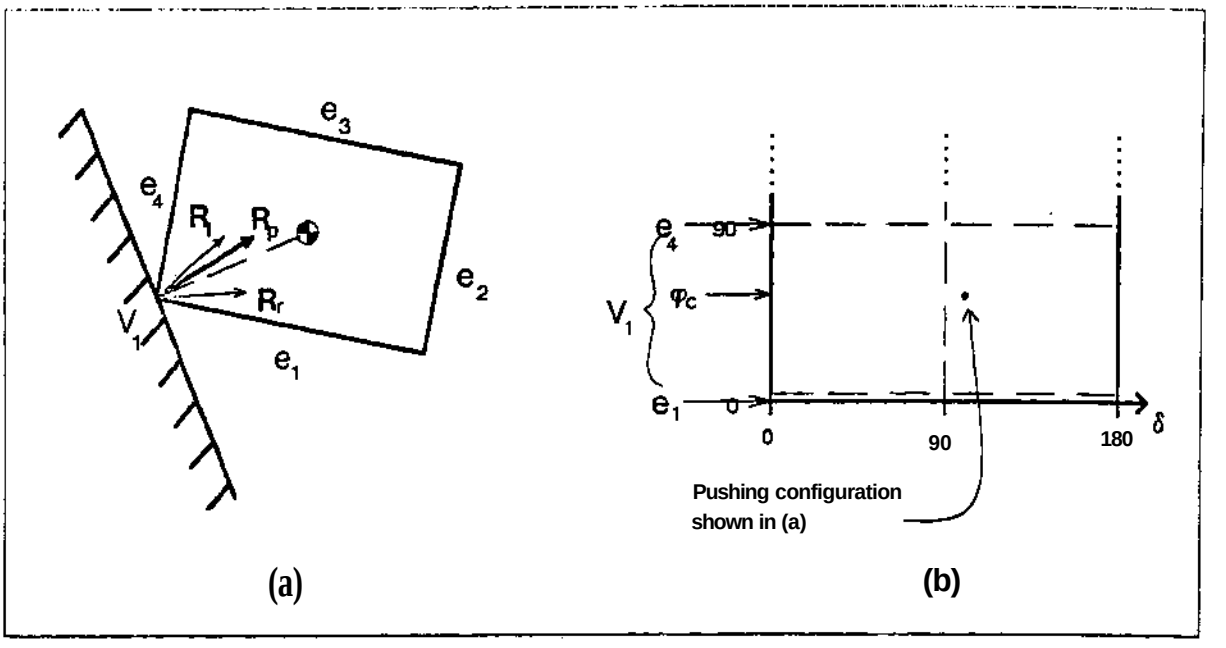


Figure 12. (a) An example case of pushing contact, axul (b) The corresponding first-contact region for (a) in the pushing space.

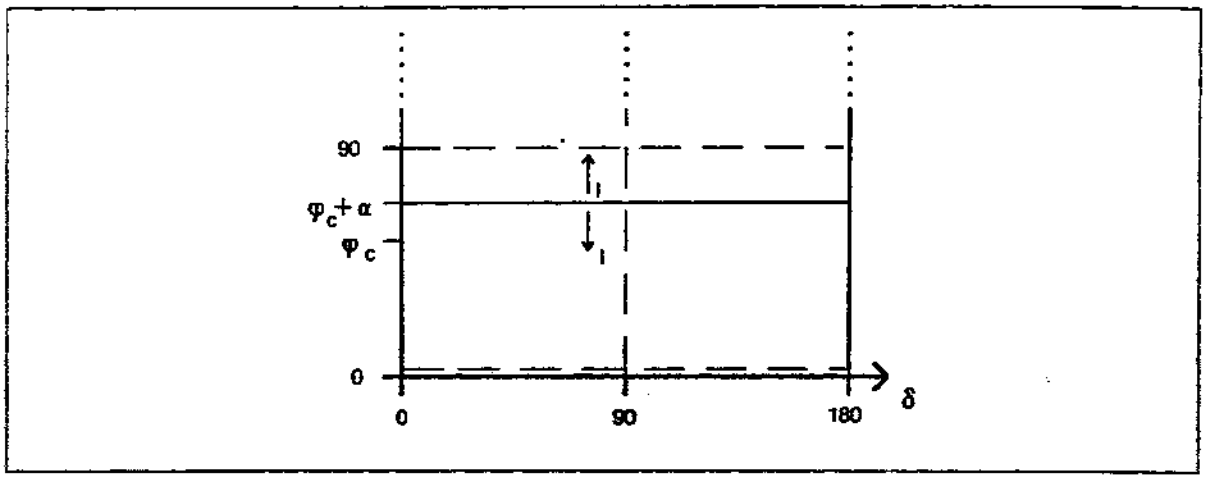


Figure 13. The* decision value for BLVs rotation vote, corresponding to a horizontal line in the pushing spare ckawn at $\$_r$ + a. T/ indicates that Ri votes counter-clockwise above the line, while || indicates that Rf votes clockwise below the decision line.

votes clockwise, and one where E* votes counter-clockwise.^ The decision value for ft* and the resultant splitting of the contact region is shown in Figure 13. Note that a clockwise polygon rotation corresponds to a vertically downward movement in the pushing space, *m* we will denote fi/s clockwise decision by ||. Similarly, a counter-clockwise decision by E§ will be denoted by ff*

Now consider the right friction cone ray E_r. By reasoning similar to the previous

[1] Note that iii some cfwe», the decision value may Be outside the contact region, in which case the mj votes uniformly throughout the ref km.

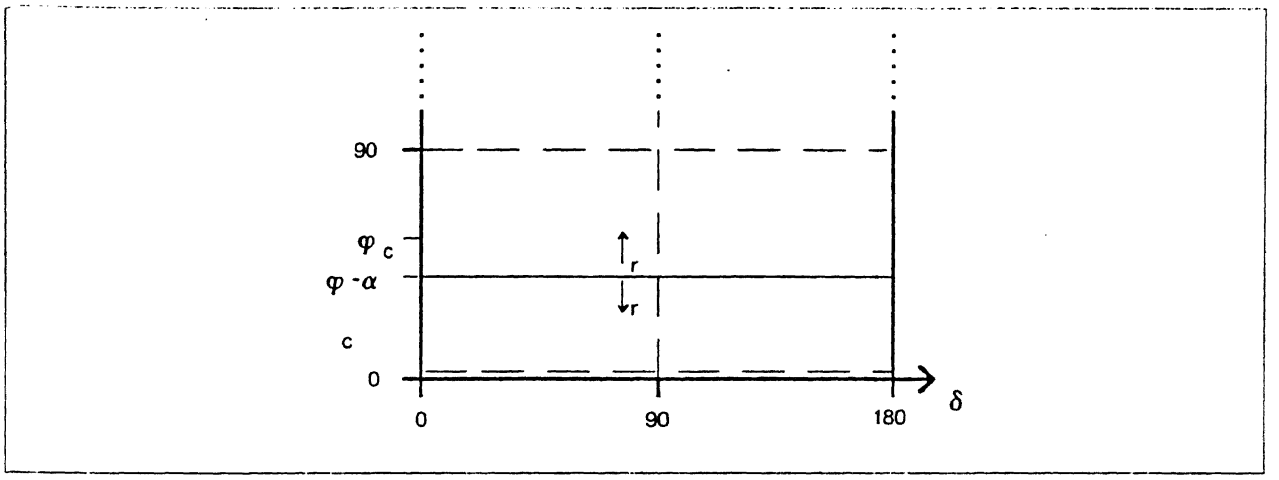


Figure 14. The decision value for $\mathbf{R_r}$, again corresponding to a horizontal line in the pushing space, but this time drawn at $\phi_c - \alpha$. $\mathbf{R_r}$ votes counter-clockwise above the line, and clockwise below it.

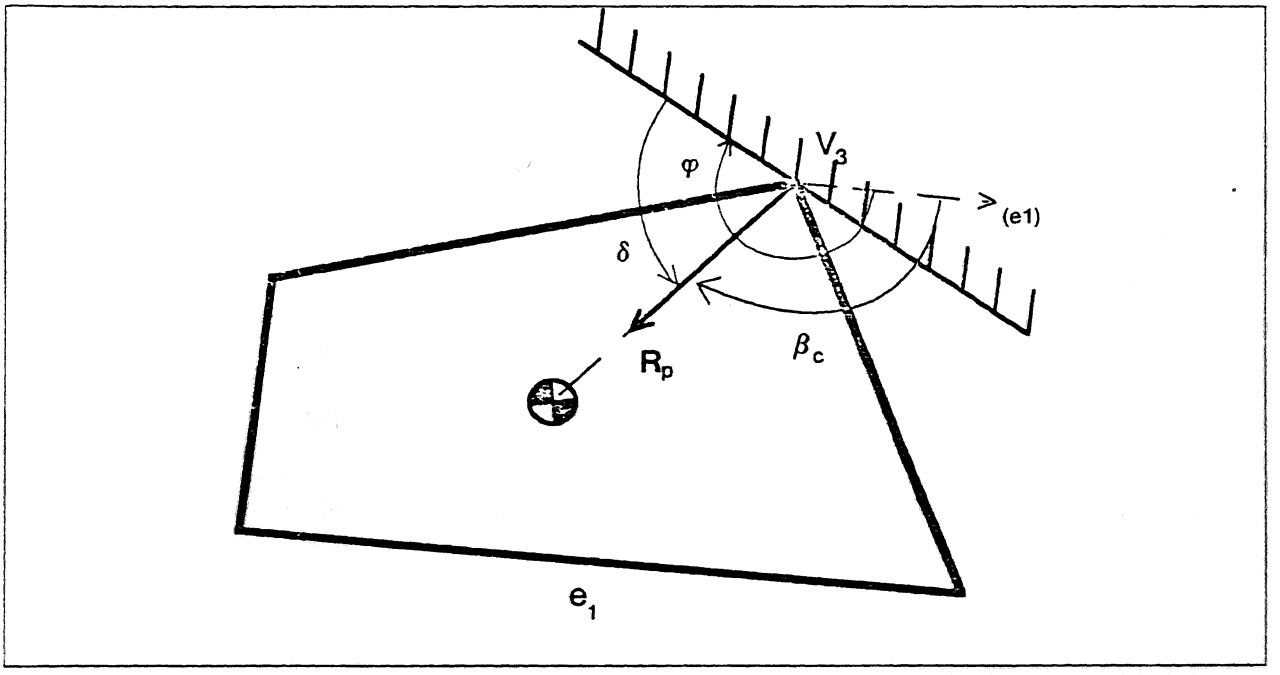


Figure 15. The critical pushing direction β_c. The angle β_c is the angle measured clockwise from $\mathbf{e_1}$ to $\mathbf{R_p}$ when $\mathbf{R_p}$ points directly at the COF. In this case, β_c is for vertex $\mathbf{V_3}$.

analysis of $\mathbf{R_l}$, we can see that as ϕ is reduced below $\phi_c - \alpha$, $\mathbf{R_r}$ crosses the dotted line and changes its vote from counter-clockwise to clockwise. This implies that the decision value for $\mathbf{R_r}$ is $\phi_c - \alpha$. Again, this decision value splits the contact region into two subregions of differing vote, as shown in Figure 14.

Finally, consider the ray of pushing $\mathbf{R_p}$. As with $\mathbf{R_l}$ and $\mathbf{R_r}$, as $\mathbf{R_p}$ crosses the dotted line between $\mathbf{V_1}$ and the COF, its rotation vote changes. However, since $\mathbf{R_p}$ does not vary directly with ϕ, $\mathbf{R_p}$'s decision value does not neatly correspond to a horizontal line of

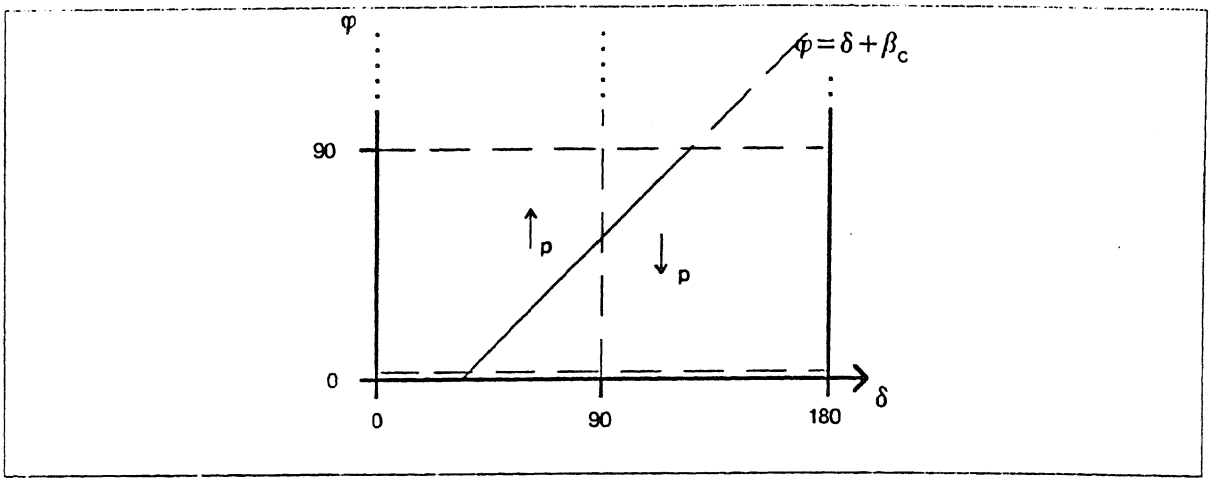


Figure 16. The decision line for $\mathbf{R_p}$'s rotation vote. Since $\mathbf{R_p}$ does not vary directly with ϕ. $\mathbf{R_p}$'s decision line is a line that slopes up at a 45 degree angle in the pushing space. $\mathbf{R_p}$ votes counter-clockwise to the left of the line, and clockwise to the right of it.

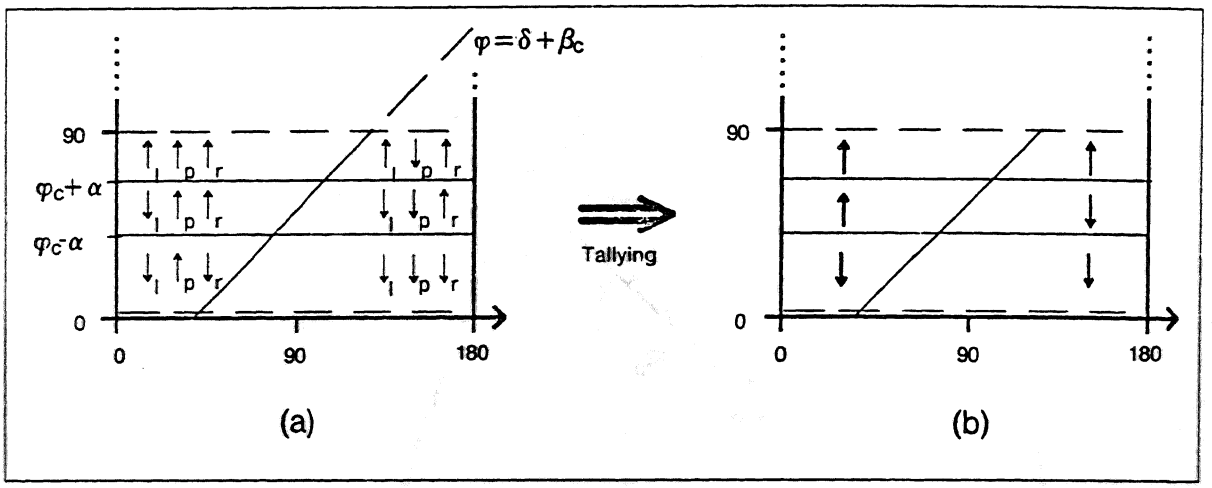


Figure 17. Tallying the votes of each subregion. The three decision lines divide the contact region into six subregions (a). In each subregion, the votes are tallied to yield an overall rotation direction for each subregion (b).

constant ϕ in the pushing space.

Rather, $\mathbf{R_p}$'s direction depends on both ϕ and δ. Let us define β_c to be the angle measured clockwise from $\mathbf{e}_1$ to $\mathbf{R_p}$ when $\mathbf{R_p}$ points directly at the COF (Figure 15).

We can tell by inspection of Figure 15 that $\mathbf{R_p}$ points directly at the COF whenever $\phi - \delta = \beta_c$. As $\mathbf{R_p}$ crosses this critical value, its rotation vote changes.

Thus, the decision line for $\mathbf{R_p}$ is easy to derive: Since $\phi - \delta = \beta_c$ all along the line, it follows that $\phi = \delta + \beta_c$. Since β_c depends only on the geometric configuration of the object, β_c is constant, and the previous equation becomes a line with slope 1 and intercept β_c in the pushing space. This decision line splits the contact region into two subregions, as shown in Figure 16.

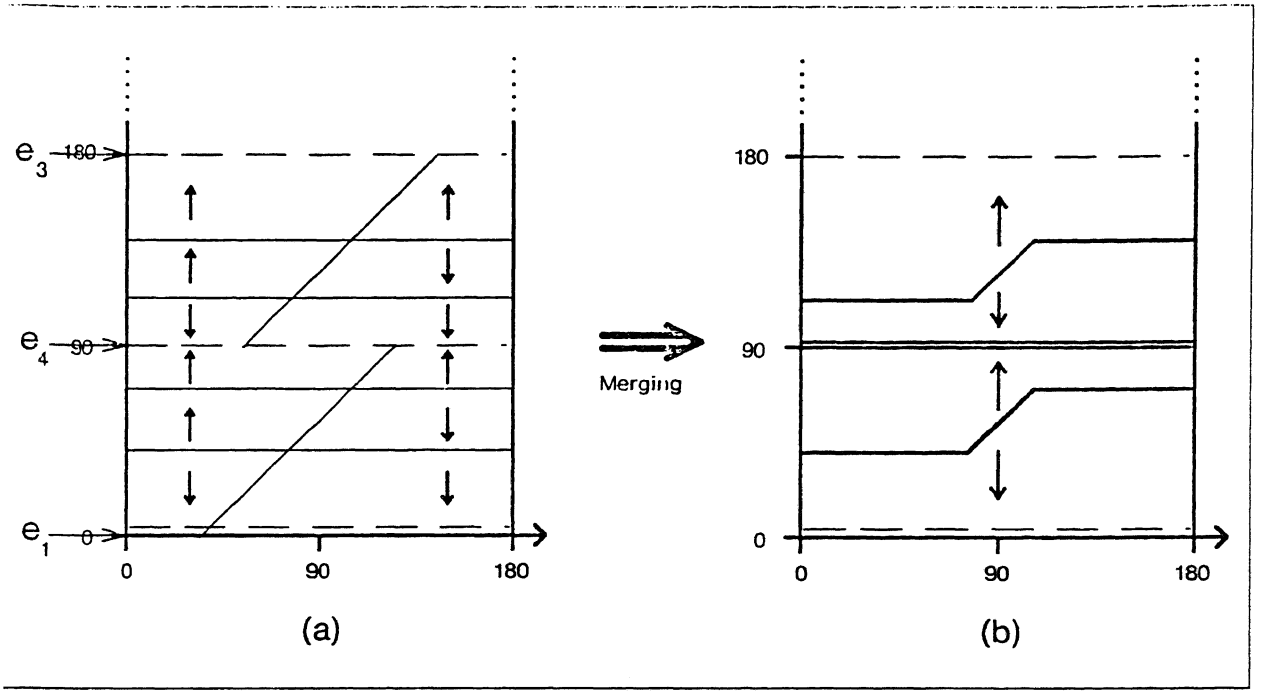


Figure 18. Merging subregions with the same rotation direction. (a) Shows two adjacent first-contact regions, with the results of tallying shown for each subregion. (b) Shows the contact regions after merging those adjacent subregions that have the same overall rotation direction. Remaining are boundaries between regions of opposite rotation direction. Boundaries with rotation directions that move away from the boundary are considered *divergent* and are drawn with a single line: boundaries with rotation directions that move toward the boundary are considered *convergent* and are drawn with a double line.

At this point, each ray has delineated its own subregions of clockwise and counterlockwise rotation; these regions can now be overlaid, and the votes of each subregion an be tallied to yield the overall rotation direction of each subregion. This process is llustrated in Figure 17.

Now that we have the rotation direction of each subregion, we can merge subregions hat have the same overall rotation direction, since the boundary between them is meaningess. This merging process is shown for two adjacent sample contact regions in Figure 18. 'igure 18(a) shows the subregions after their votes have been tallied, and Figure 18(b) hows the boundaries that remain after merging.

After the merging process is complete, two types of boundaries remain: *divergent*, vhere the polygon rotates away from the boundary, and *convergent*, where the polygon otates toward the boundary. We will differentiate between divergent and convergent oundaries by drawing a single bold line for divergent boundaries, and a double bold line or convergent boundaries, as shown in Figure 18.

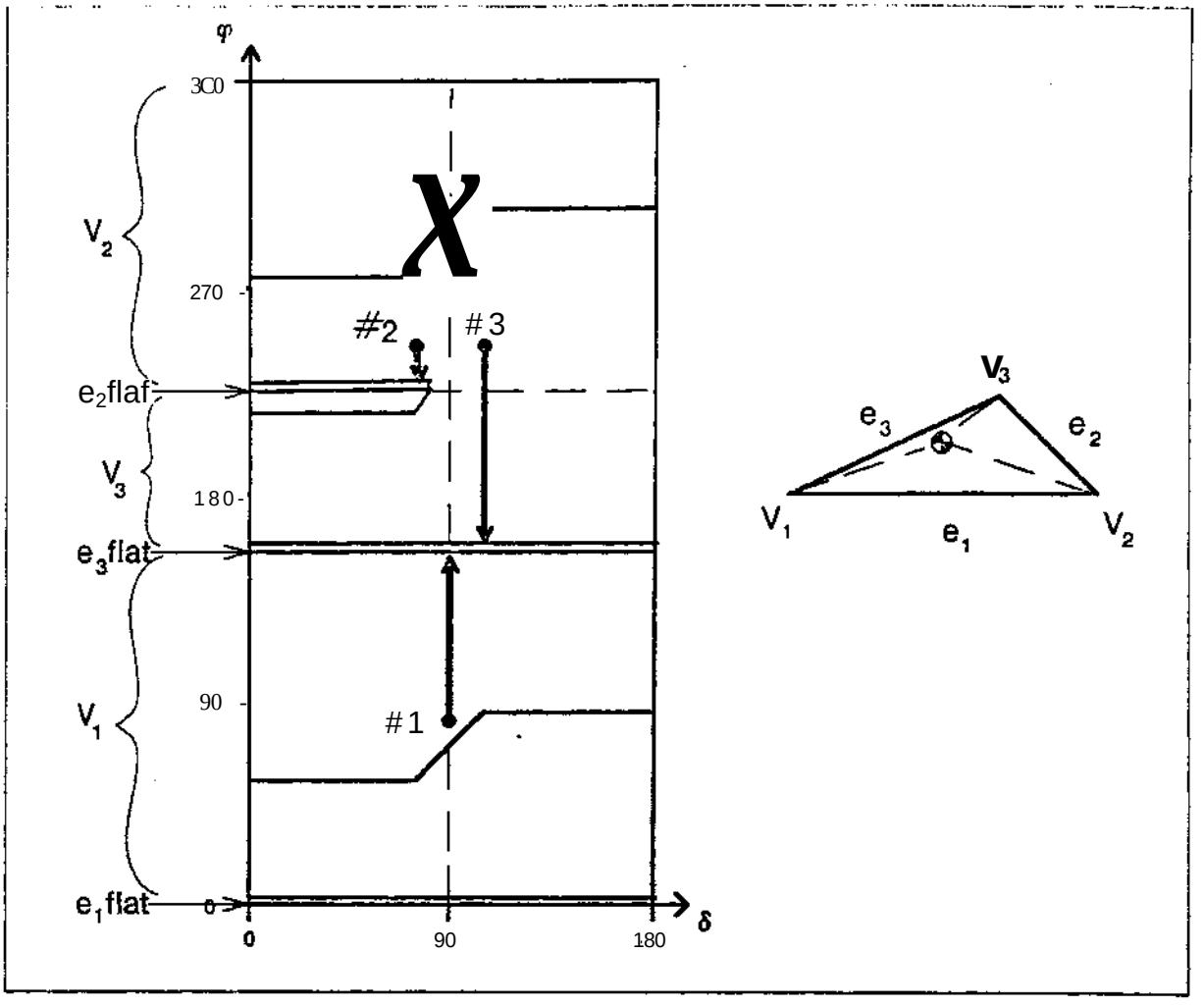


Figure 19. Tliree sample pushing operations. In pushing operation #1, the triangle rotates counter-clockwise onto e^ and remains stable on e_3. In #2, the triangle rotates clockwise onto C2 and remains stable on e_2. Pushing operation #3 starts out with the same orientation $„ but a different pushiu^ directioii *$; n»* a result, the triangle rotates clockwise onto ej, keeps rotating to ej, and then remains stable 011 e^.

The boundaries generated thus fax have implicit stability information embedded within them. The divergent boundaries correspond to unstable pushing operations that might rotate in either direction if perturbed, while the convergent boundaries correspond to stable edge-Hat pushing configurations where the polygon stays flat as the pushing operation continues. The convergent boundaries are stable because any perturbing influences to either side of the boundary are **compensated** by the tendency of the polygon to rotate back to the **edge-flat** configuration.

Thus we can analyse the first-contact region for each polygon vertex and generate rotation/stability **information** for all pushing operations for the polygon.

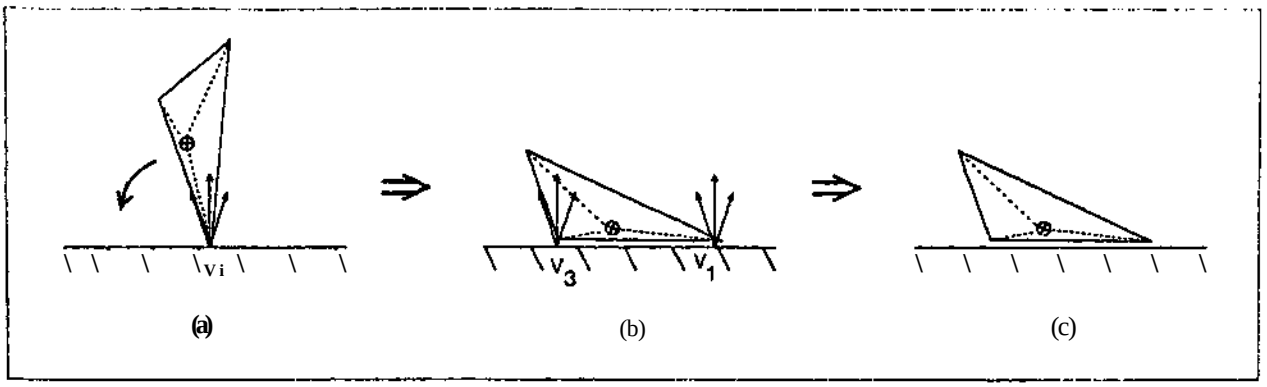


Figure 20. Pushing operation #i. Contact begins in the initial configuration shown in (a), whereupon tin* triangle rotates couuftT-clorkwise until edge C3 lies Hat against the pushing plane (b). Because vertices V3 and Vi tend to rotate the triangle in opposite directions, edge c_3 remains stable against the pushing plane, and the pushing operation terminates* in the configuration shown in (c).

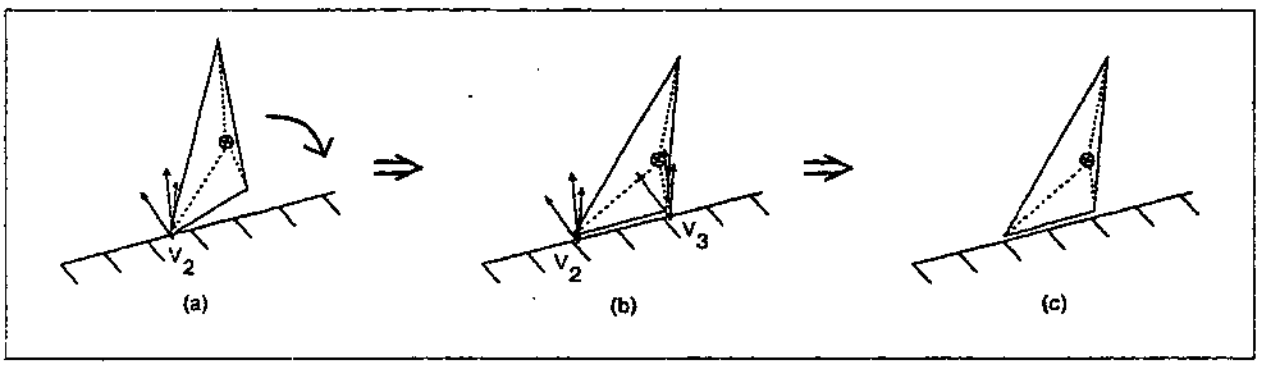


Figure 21. Pushing operation #2. Contact begins in the initial configuration shown in (a), whereupon the triangle rotates clockwise* until edge 02 lies flat against the pushing plane (b). Edge e2 remains stable against the pushing plane because vertices Vo and V3 both oppose rotation away from c^, and the pushing operation terminates in the configuration shown in (c).

The result of carrying out this process on the triangle seen earlier is shown in Figure 19. Notice that the edge-flat line for <to·l$ ontv convergent for some values of «5; this is due to the fact that €2 only remains fiat against the pushing plane when R_p lies to the right of the COF. This is best understood by carefully examining the three sample pushing operations illustrated in Figures 19-22.

As the three example operations demonstrate, pushing operations that begin above the convergent portion of e2*s edge-flat line terminate with $2 fl[a]* against the pushing plane, and pushing operations that begin above the non-convergent portion of e2*s edge-flat line [a]roll past" e2 and terminate with *s%* flat against the pushing plane. To differentiate between these two results, we cam extend a boundary along a vertical line of constant *8* from the end of the convergent portion of ©2[ls] edge-flat line until the next rotation boundary is encountered.

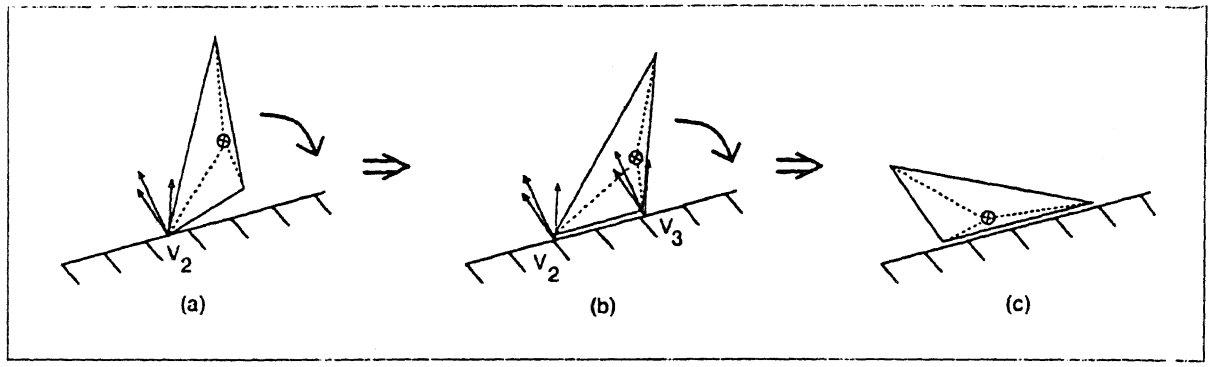


Figure 22. Pushing operation #3. Contact begins in the initial configuration shown in (a). This initial configuration is identical to the initial configuration for pushing operation #2, except that the value of δ is different, implying that the ray of pushing points in a slightly different direction. After initial contact is made, the triangle again rotates clockwise until edge $\mathbf{e}_2$ lies flat against the pushing plane (b). This time, however, vertex $\mathbf{V}_3$ does not oppose rotation away from $\mathbf{e}_2$, and the triangle continues to rotate clockwise onto $\mathbf{e}_3$, leading to the final configuration shown in (c).

Note that this boundary is different from the boundaries drawn previously because it arises from our convention that δ is held constant throughout a pushing move, rather than from a physical difference in the motion of the polygon. One might imagine pushing moves where δ is allowed to vary, in which case these vertical boundaries become meaningless. Since these vertical boundaries are not as "strong" as the boundaries between rotation directions, we will draw them dashed, as indicated in Figure 23.

Once any required vertical boundaries are added, the pushing space is divided into closed *edge-stability regions*, as shown in Figure 23. These regions correspond to all pushing moves that will terminate with a particular edge flat against the pushing plane; each region is labelled with its appropriate edge in Figure 23.

This diagram is the *push-stability diagram* for the triangle; any pushing operation chosen within the edge-stability region for a given edge $\mathbf{e}_i$ is guaranteed to terminate with $\mathbf{e}_i$ flat against the pushing plane.

Note that edge-stability regions don't always fill the entire pushing space. For example, consider a regular octagon with a very high coefficient of friction; if such an octagon is pushed at an angle (i.e., δ not close to 90), it will continue to roll forever, since no edge is stable. This is indicated by the push-stability diagram for the octagon, shown in Figure 24. If a pushing move starts outside the edge-stability region for one edge (δ not close to 90), it can never enter the edge-stability region for any other edge, and so the octagon continues to roll.

This provides the answer to our question "How Will the Polygon Rotate?" We can use the first-contact regions derived in the previous section to generate edge-stability regions. Pushing moves within an edge-stability region for an edge $\mathbf{e}_i$ are guaranteed to terminate with $\mathbf{e}_i$ flat and stable against the pushing plane.

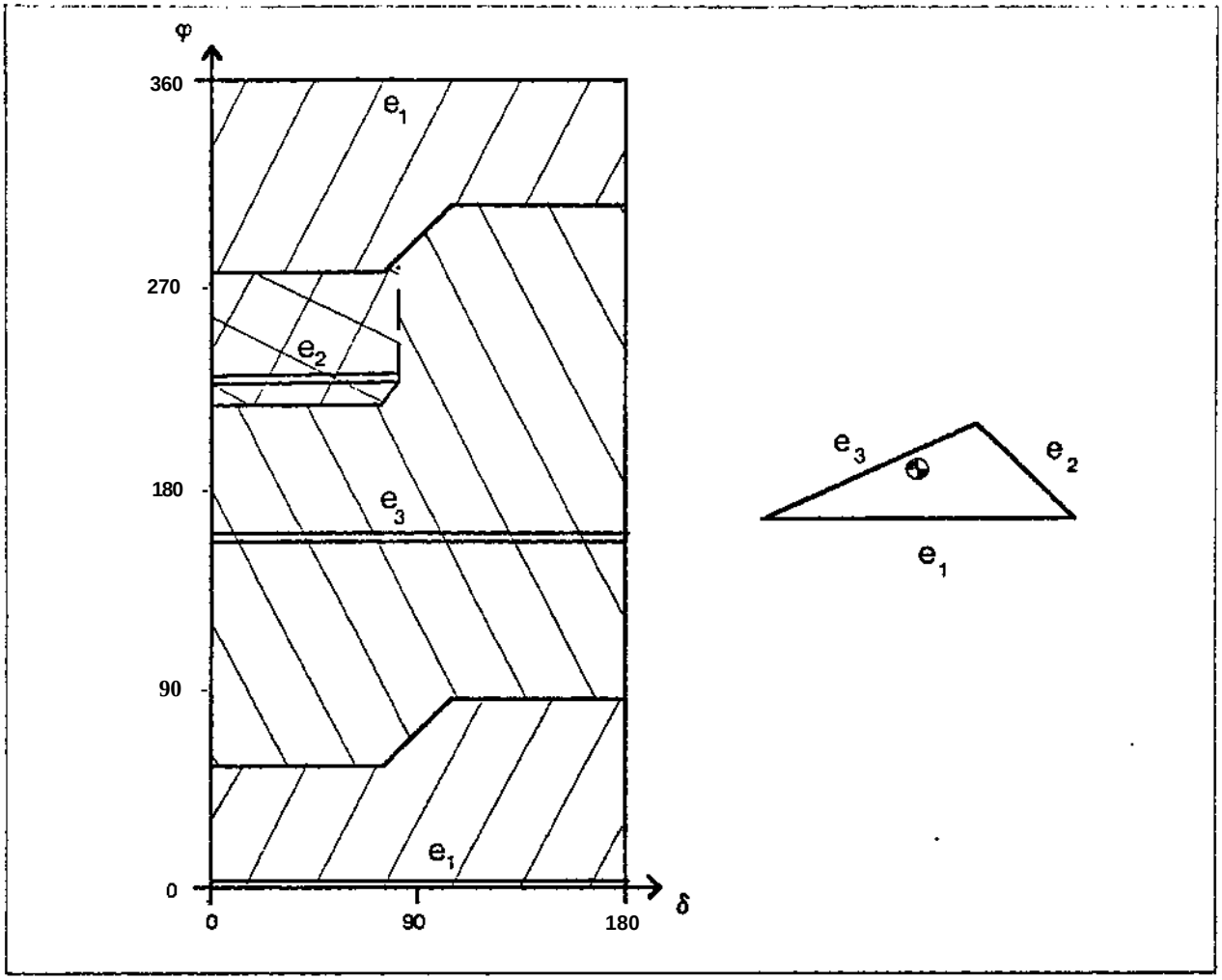


Figure 23. The *pnsli-stahility diagram* for the triangle. Any pushing operation that begins within the edge-stability region for a given edge e^ is guaranteed to terminate with e* flat and stable against the pushing plane.

Summary — One-Plane Pushing

This completes the analysis of one-plane polygon pushing; now we can fully characterize the motion of any convex polygon under pushing by applying the following analysis:

1. To determine where the pushing plane will first contact the polygon, the pushing space is divided into bands of vertex first-contact by edge-lat lines; these edge-flat lines correspond exactly to the polygon's £- edge-angles.

2. To determine how the polygon will rotate during contact, each contact region is split by each ray according to the ray's rotation vote within the contact region. In the resulting subregions, all three votes are tallied to find the overail rotation direction for each subregion.

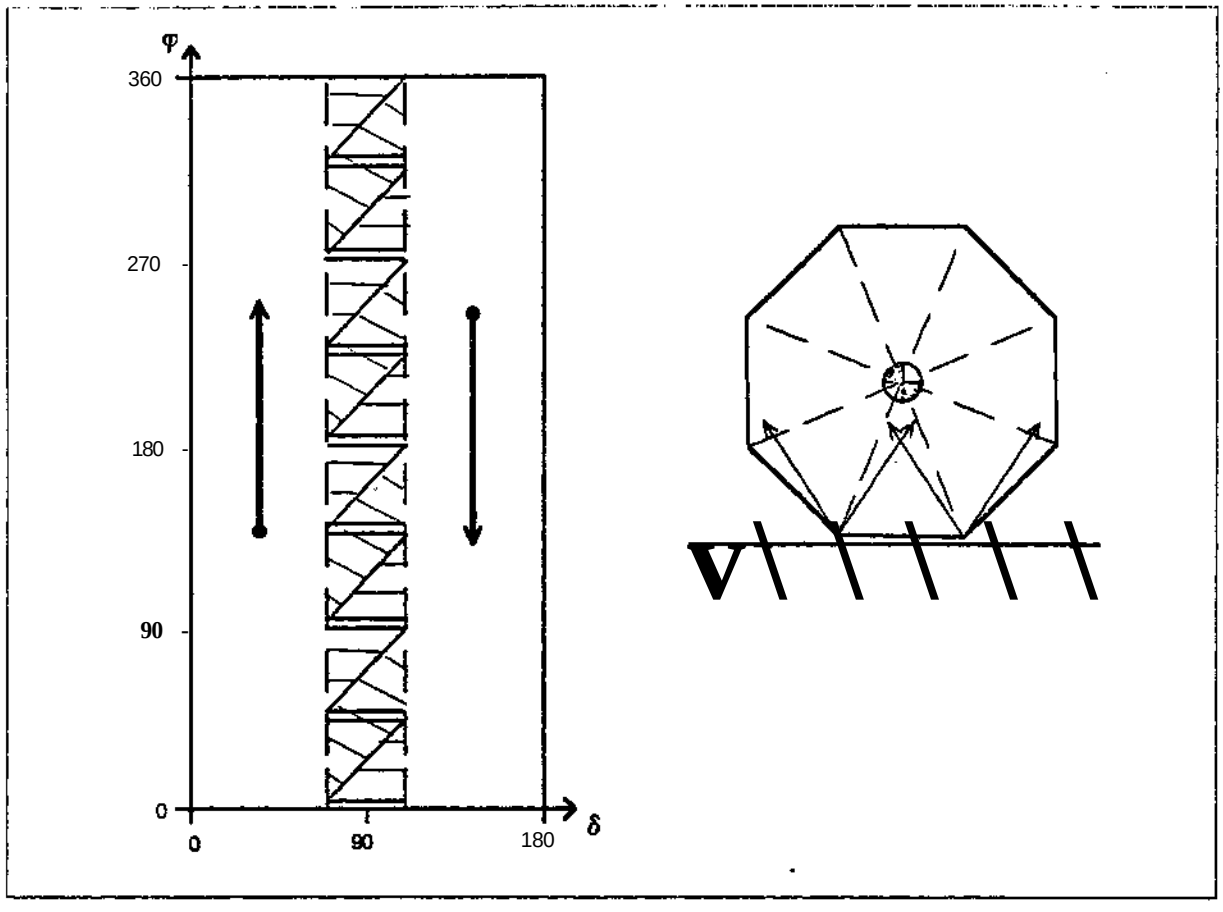


Figure 24. An example of an object tliat lias infinitely unstable ptishing moves. Pushing moves tliat originate outside the shaded area will cause the octagon to continually "roll" along the pusMng plane.

3- Next, all adjacent subreglons with the same rotation direction are merged, defining boundaries of opposite rotation direction. Boundaries where the polygon rotates toward the boundary are taken to be convergent boundaries where the polygon will remain stable during pushing, while boundaries that the polygon rotates away from are taken to be divergent, unstable boundaries,

4. Finally, since *6* is held constant during a pushing move, vertical boundaries are established appropriately for any edge-Eat limes that are partially convergent.

The resulting push-stability diagram shows, for any edge of the polygon, the space of all possible pushing moves that are guaranteed to result in a stable edge-flat configuration.

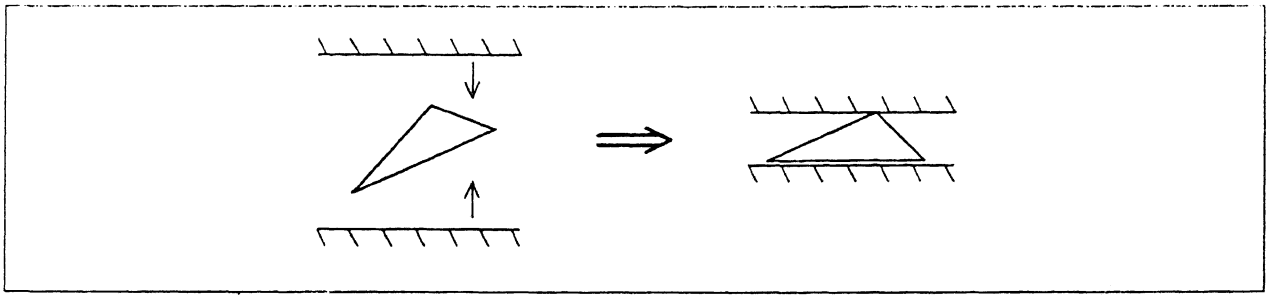


Figure 25. An example of a successful squeezing operation.

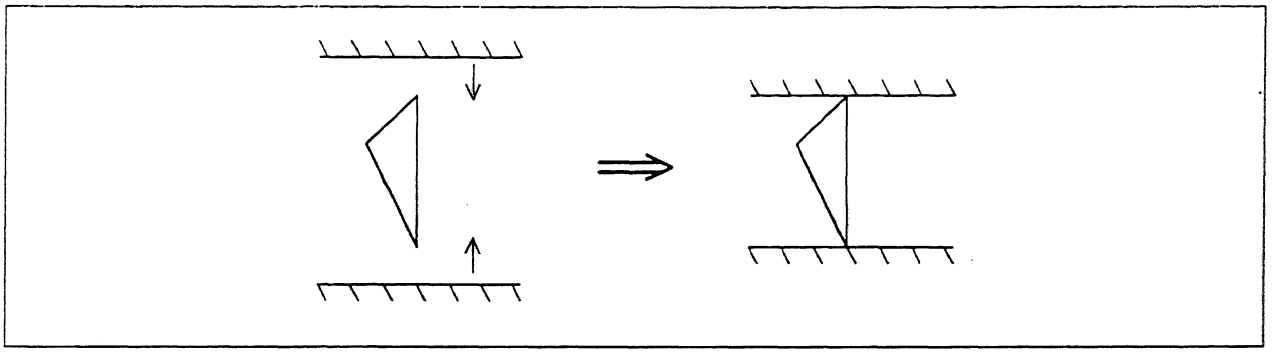


Figure 26. An example of an unsuccessful squeezing operation.

IV. Squeezing with Two Planes

Consider the example squeezing operation shown in Figure 25. The two planes are parallel, and squeeze together on either side of the triangle. As the squeezing planes make contact, the triangle is forced to rotate until edge $\mathbf{e}_1$ is pressed flat against the lower plane. Once this edge is flat against the plane, the fingers can exert a continuous squeezing force to firmly grasp the triangle.

Note that in this configuration, the applied squeezing force can be arbitrarily increased and the triangle will not move. We will refer to this condition as *wedging*. By definition, a polygon is wedged between a pair of squeezing planes if the squeezing force applied by the planes can be increased arbitrarily without causing the polygon to rotate. When a polygon is grasped in such a wedging condition, the grasp can be made arbitrarily strong by appropriately increasing the squeezing force.

The wedging condition illustrated in Figure 25 corresponds to a grasping configuration we would like to achieve; the triangle's orientation is constrained (since edge $\mathbf{e}_1$ is flat against the lower plane), and the triangle is held firmly between the gripper fingers (since the triangle is wedged).

However, not all wedging conditions correspond to desirable grasps. Consider the second example squeezing operation depicted in Figure 26. In this squeezing operation the triangle is also wedged, because the squeezing force can be increased arbitrarily without

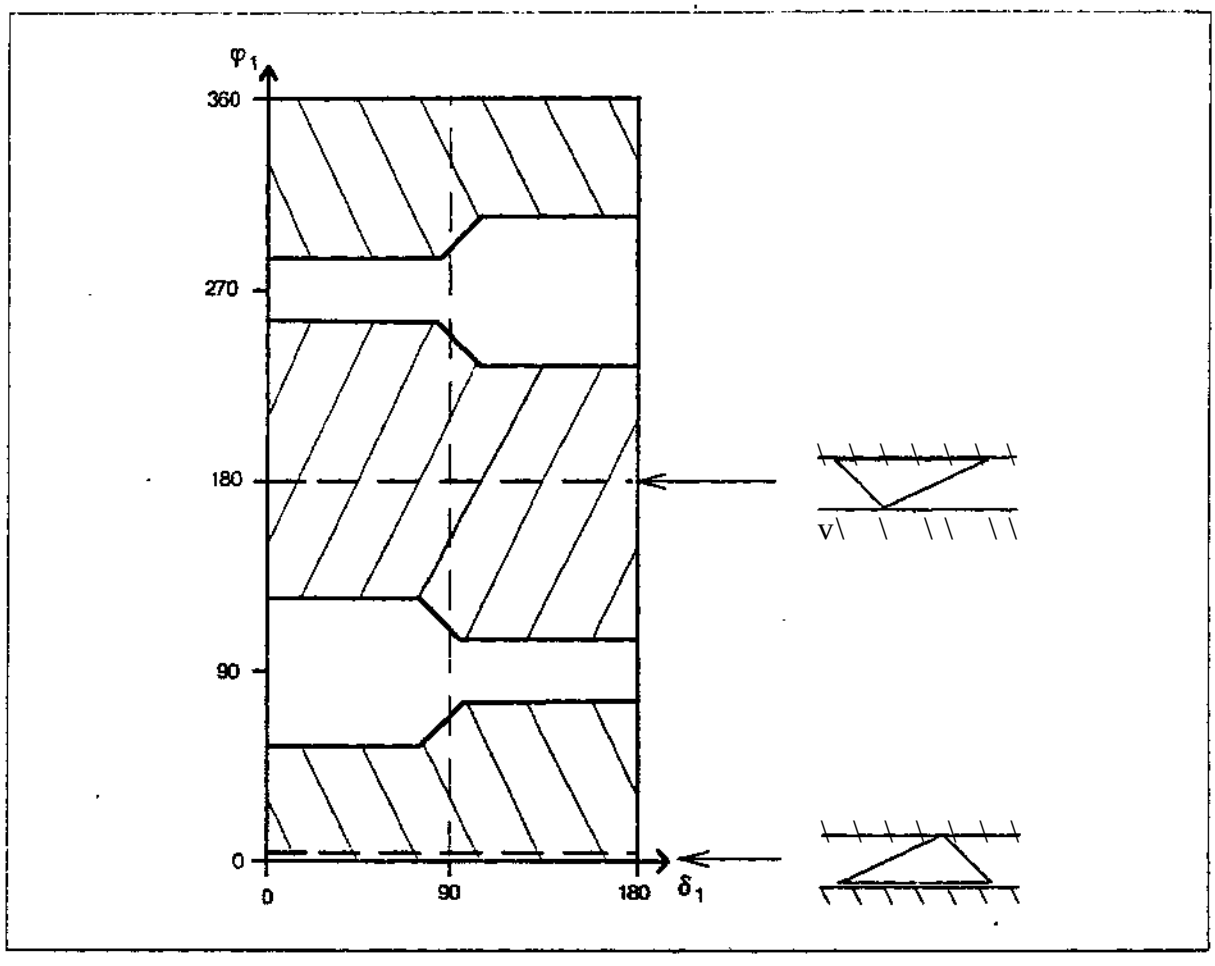


Figure 27. The squeeze-grasp diagram for *the* triangle.

causing a rotation of the triangle. However, in this configuration the triangle's orientation is not fully constrained, because the presence of friction allows this type of wedging to occur over a small range of orientations (this will be explained in more detail later). Therefore, since we have not constrained the triangle's orientation, we have not eliminated the uncertainty in the triangle's position, and this grasping configuration is undesirable. Further, it will become apparent later that this grasp cannot be achieved as reliably as the first.

These two examples prompt us to define two conditions required for a *stable grmp:*

(1) At least one edge of the polygon is fiat against a squeezing plane.

(2) The polygon is wedged between the squeezing planes.

Condition (I) assures that the uncertainty in the polygon's orientation is removed, while condition (2) assures that we have a firm grasp of the object.

With our notion of a stable grasp thus defined, we would like to know what **sqneeiiag**

motions will achieve a stable grasp without getting stuck in one of the undesirable wedging configurations demonstrated in the second example. This query is answered by the *squeeze-grasp diagram*, shown in Figure 27. The squeeze-grasp diagram represents the space of all squeezing moves, and the shaded regions indicate those moves that are guaranteed to succeed in achieving a stable grasp.

The remainder of this section will explain how this squeeze-grasp diagram is generated. We will begin by formalizing our notion of squeezing to allow us to describe squeezing operations in a way similar to the way pushing operations were described previously. Next we will develop a simple algorithm for finding all the pairs of polygon features that can make simultaneous contact with both squeezing planes. After we have found all of these contact pairs, we will then determine which contact pairs provide a stable grasp by meeting our criteria listed above; those pairs that do meet our criteria will be listed as desirable grasp configurations that we seek. In addition, we will look among all of the contact pairs to determine where undesirable wedging configurations occur, and thus delineate those squeezing configurations that should be avoided. Finally, we will describe how pushing phenomena affect a squeezing move, and combine our analysis of squeezing with the pushing analysis of the previous section to produce the complete squeeze-grasp diagram of Figure 27.

Squeezing Notation and the Squeezing Space

At this time we will formalize our notion of squeezing in order to provide a context for the subsequent analysis of squeezing operations. First we will define a single squeezing move. As an initial condition, the parallel squeezing planes start with some orientation, starting position, and separation. The planes then squeeze together along their common normal, keeping their orientation constant. Meanwhile, the two planes can also move laterally in unison. The speeds of plane squeezing and lateral movement are held constant throughout the squeezing move; by choosing different combinations of squeezing and lateral movement speeds, various δ values can be achieved. This motion continues until the planes can't squeeze together any more (i.e., when an opposing force is met that exceeds some pre-set threshold). Note that since guaranteed grasp plans imply that the gripper will have a stable grasp of the object when the planes are finished squeezing, it is not important to assure that the lateral movement stops when the termination condition is met.

The notation used to describe squeezing moves is essentially identical to the pushing notation presented above, but must be modified to account for the presence of two half-planes instead of one. First, the planes are arbitrarily labelled plane #1 and plane #2. The parameters for each plane are ϕ and δ as before, but are now ϕ_1 and δ_1 for plane #1, and ϕ_2 and δ_2 for plane #2 (see Figure 28). Since the two planes are always parallel, these parameters are related as follows:

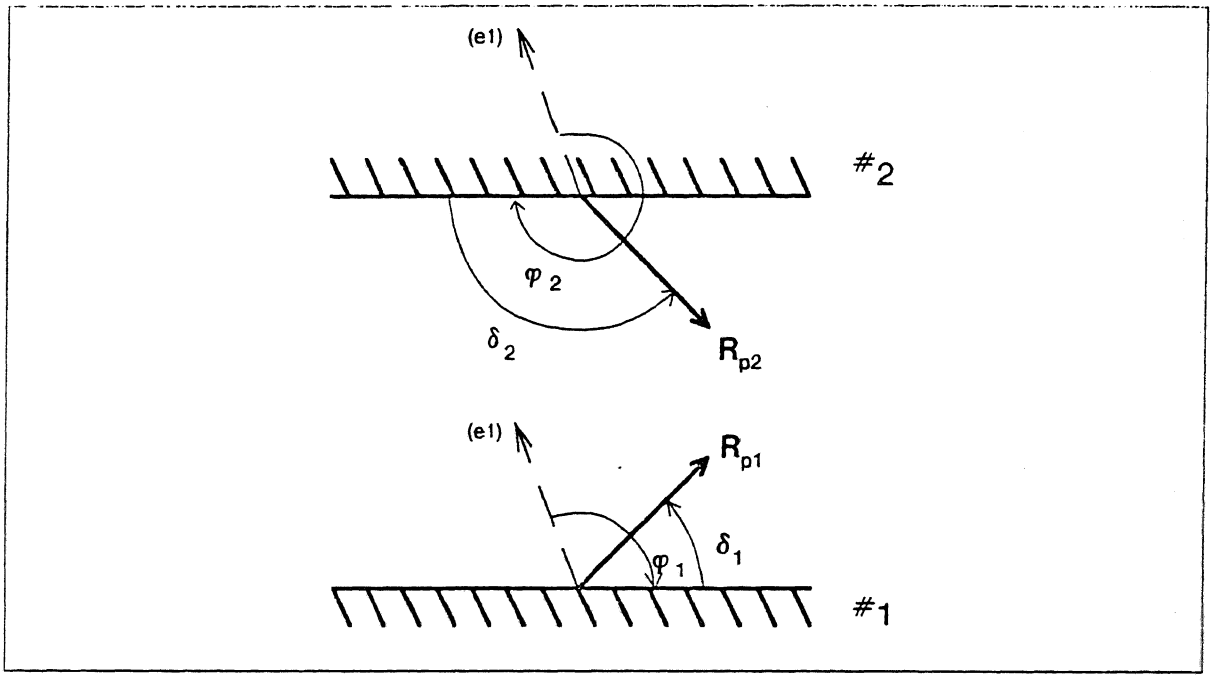


Figure 28. Squeezing notation.

$$\phi_2 = \phi_1 + 180$$

$$\delta_2 = 180 - \delta_1$$

As with pushing operations, the set of all possible squeezing operations forms a well-defined *squeezing space*. This squeezing space can be represented by the same rectangular coordinate system used for the pushing space, where the δ and ϕ axes in the pushing space are replaced with δ_1 and ϕ_1 axes, respectively (Figure 29).

The parameters δ_2 and ϕ_2 can be included in the squeezing space diagram by by drawing the δ_2 axis along the top of the diagram and drawing the ϕ_2 axis along the right side. Note that the δ_2 axis is reversed and the ϕ_2 axis is shifted 180 degrees because of the relationship between the parameters for plane #1 and plane #2.

Where Can the Squeezing Planes Contact the Polygon?

In order to understand the behavior of a polygon under squeezing, we must first be able to predict where the polygon can be touched by the squeezing planes.

To represent the points of contact between the squeezing planes and the polygon during a squeezing operation, we will introduce the concept of a *contact pair*. Two features on a polygon are a contact pair if, for some orientation of the opposed squeezing half-planes, the two features are the first points of contact. We will denote a specific contact pair as an ordered pair of polygon features, where the first feature listed is taken to be in contact with plane #1. For example, (V_1, V_2), (e_2, V_3), etc.

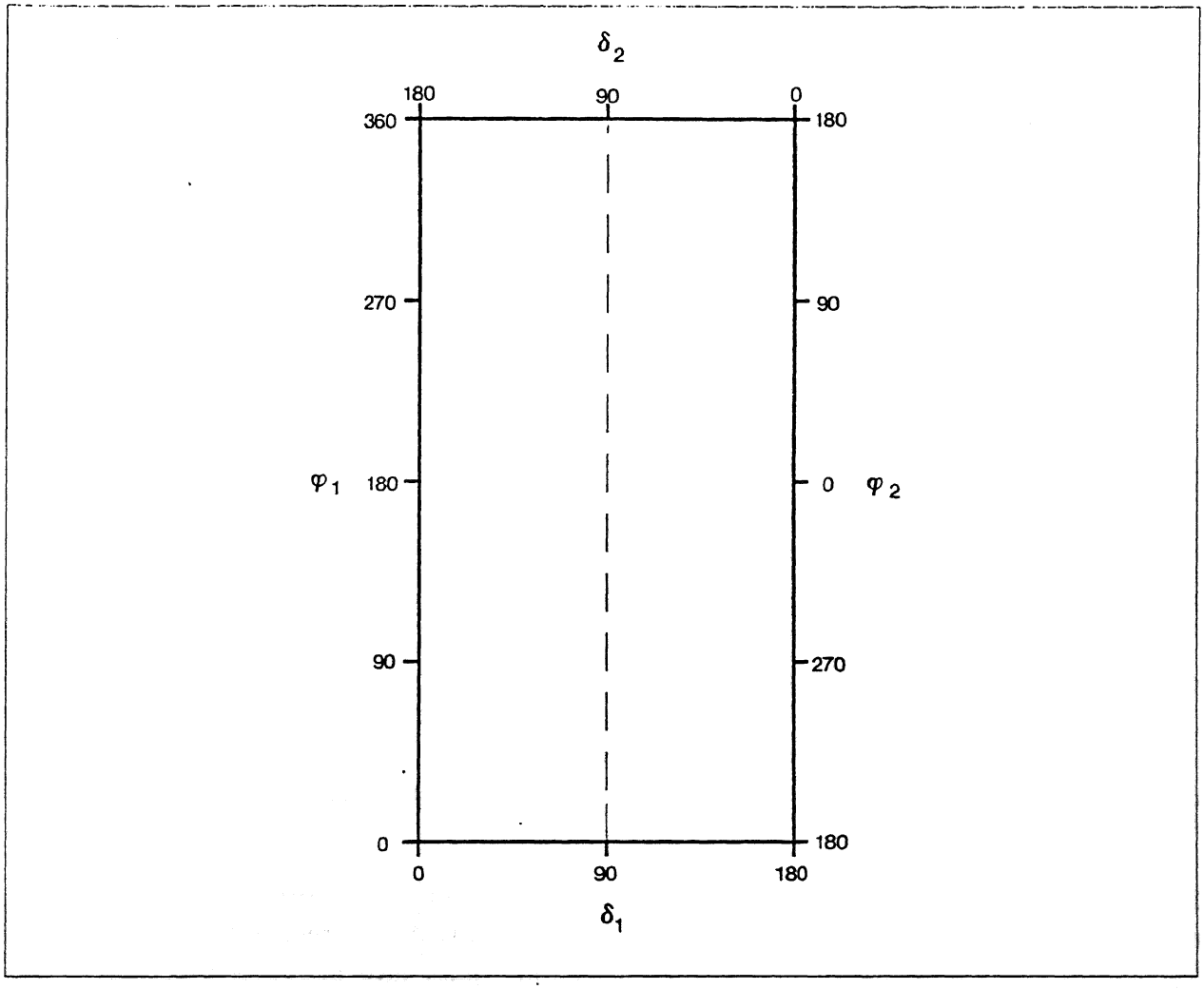


Figure 29. Squeezing space. Squeezing space is similar to pushing space, except for the addition of the new parameters ϕ_2 and δ_2 along the right side and top edges.

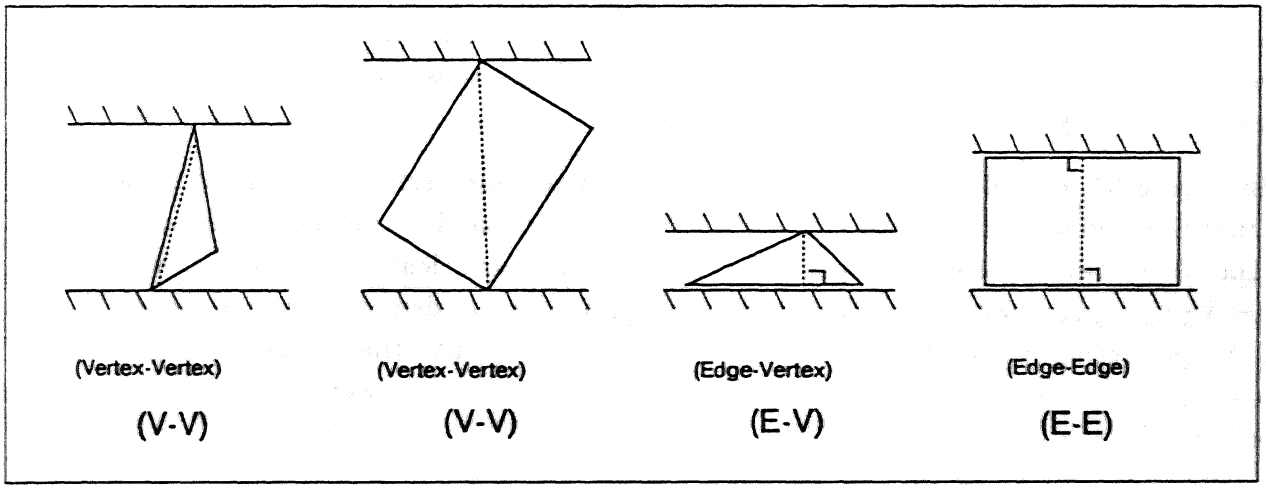


Figure 30. Examples of polygon contact pairs.

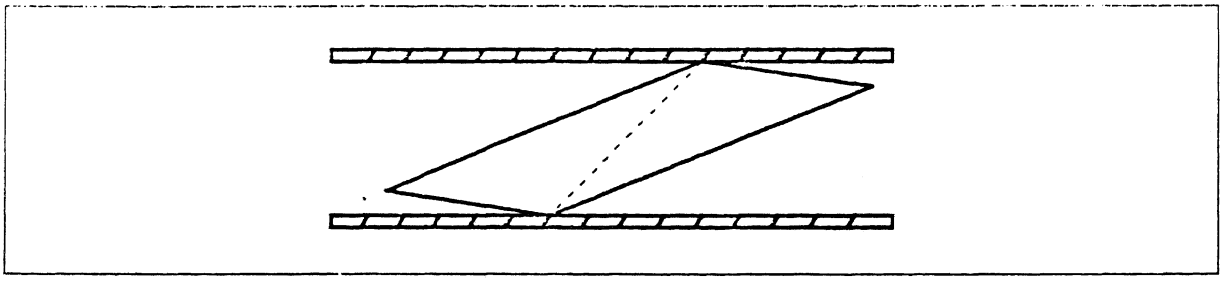


Figure 31. A (V-V) contact pair that cannot wedge.

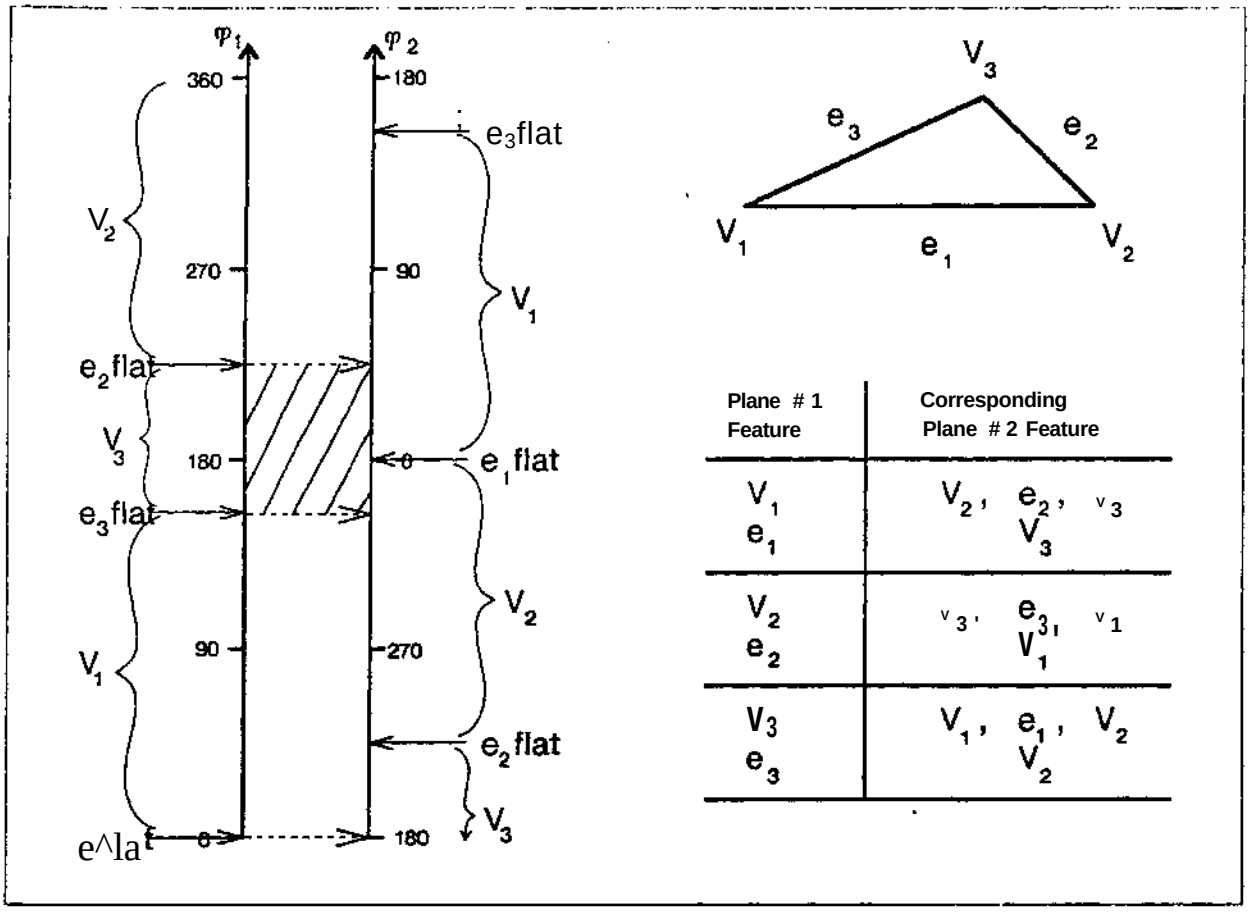
Contact pairs are connected by dotted lines called *contact pair lines*. Some examples of contact pairs are shown in Figure 30. Contact pair lines can occur between two vertices, an edge and a vertex, or two edges; for brevity, these cases will be referred to as (V-V), (E-V), and (E-E) contact pairs, respectively (Figure 30).

Contact pairs are defined to exist for all pairs of contact points, regardless of whether or not the squeezing contact has the potential for wedging. Note that not all contact pairs can wedge, and it may not be possible to orient the squeezing planes normal to the contact pair line (Figure 31). Therefore, it is incorrect to directly associate contact pairs with wedging; squeezing contact with a contact pair is a necessary but not sufficient condition for wedging. We will discuss the conditions required for wedging later; for now, our strategy is to find all of the polygon's contact pairs, and then determine which contact pairs can wedge.

To find all of a polygon's contact pairs, we can capitalize on the results of our previous investigation of pushing. Recall that a polygon's ξ_i values directly correspond to edge-flat lines in the pushing space, which partition the pushing space into bands of vertex first contact. A similar result applies in the squeezing space as well. The bands of vertex first-contact for plane #1 are identical to the first-contact bands derived previously for pushing. Plane #2's first-contact bands are also identical, except that they are shifted 180 degrees in the squeezing space (Figure 32). As we shall see, these two sets of first-contact regions immediately define all of the polygon's contact pairs.

In Figure 32, consider plane #1's first-contact band for vertex V_3 (shaded region). Note that this region overlaps with plane #2's first-contact band for vertex V_1, e_1 edge-flat line, and V_2 vertex first-contact band. This implies that for orientations where plane #1 touches V_3 first, plane #2 can touch either V_1, e_1, or V_2 first, as demonstrated in Figure 33. Thus, by projecting plane #1's first-contact region for V_3 onto the corresponding first-contact regions for plane #2, three of the polygon's contact pairs were found: (V_3, V_1), (V_3, e_1), and (V_3, V_2).

By carrying out this process for all of plane #1's vertex first-contact bands and edge-flat lines and tabulating the results, all of the polygon's contact pairs can be found. The table in Figure 32 shows the results of applying this procedure to our now-familiar triangle;



Plane # 1 Feature	Corresponding Plane # 2 Feature
V_1 e_1	V_2, e_2, V_3 V_3
V_2 e_2	V_3, e_3, V_1 V_1
V_3 e_3	V_1, e_1, V_2 V_2

Figure 32, Detection of contact pairs. The first-contact regions for plane #1 arc projected onto the first-contact regions for plane #2, yielding the table of contact pairs shown at right.

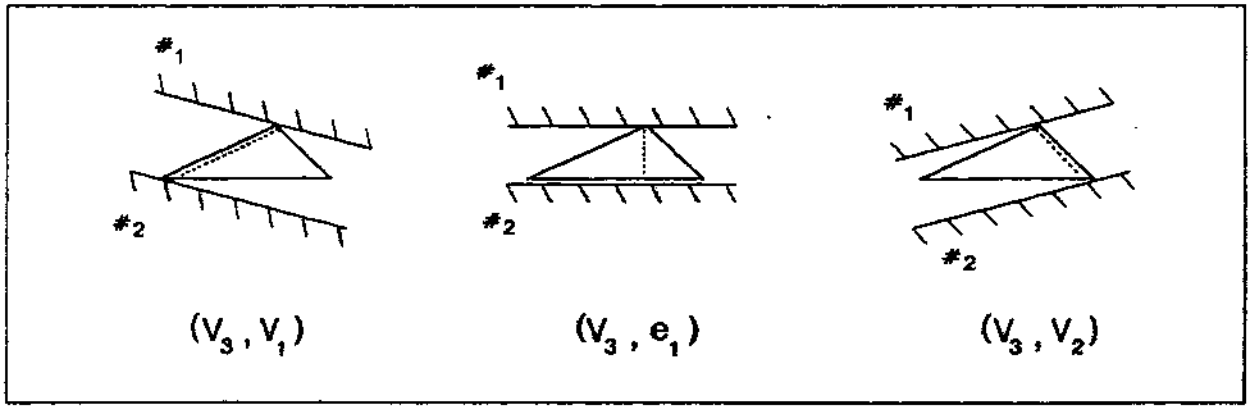


Figure SS. Dlustration of the three contact pairs founid from the shaded region in Figure '32.

note that all of the polygon's contact pairs (ia both directions) were returned by the procedure.

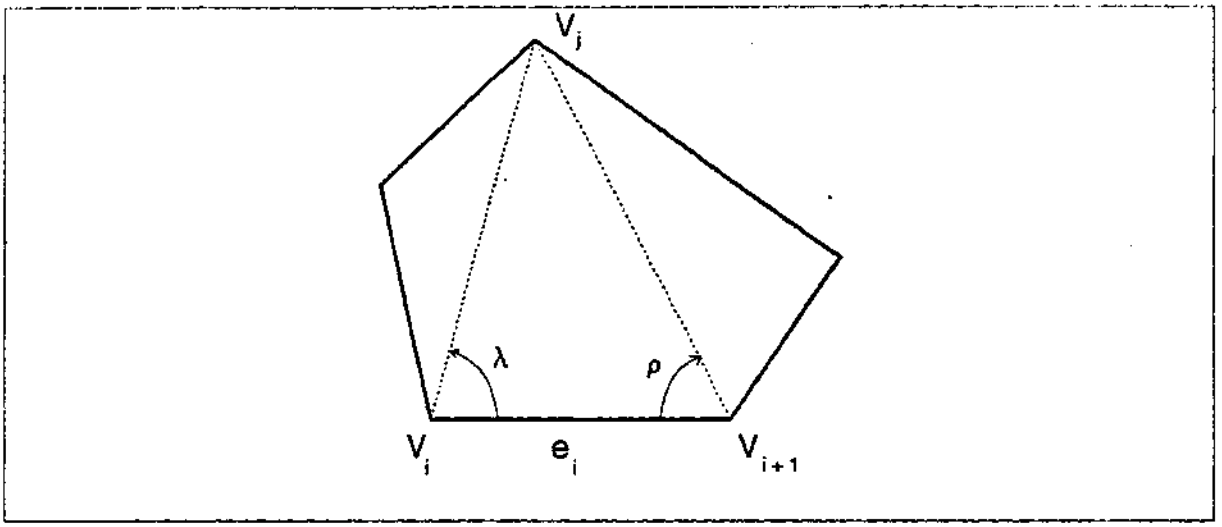


Figure 34. The «uij(lcs A ·find *p* for an edge-vertex (E-V) contact pair.

Whîch Contact Pairs Provide a Stable Grasp?

Now that we have a list of all of the polygon's contact pairs, we can determine which contact pairs provide a stable grasp of the polygon. Recall the requirements for a stable grasp that were put forth earlier:

(1) At least one edge of the polygon is flat against a squeezing plane.

(2) The polygon is wedged between the squeezing planes.

When the above conditions are satisfied, the object is guaranteed to be held in a stable grasp, with two degrees of uncertainty removed from the object's position.

Determining what squeezing orientations achieve these conditions is not difficult. Condition (1) is met in all cases of {E-V) or (E-E) contact pairs; all instances of these contact pairs have already been detected by the algorithm described in the previous section. **Therefore,** all stable grasps can be found by sioiply determining which instances of (E-V) and (E-E) contact pairs wedge, thus satisfying condition (2).

First we will consider (E-V) contact pairs. We can determine whether or not any given (e_i> Vy) contact pair will wedge by examining the angles A and />, shown in Figure 34- These angles axe the interior angles formed between the ends of the edge e_t- and the opposite **vertex Vy;** A is the angle fcmed at the left **vertex,** while *p* is the aagle formed a the right vertex.

These angles **indicate** whether the contact pair **(e,-, Vy) will** wedge when **squeezed,** *km* [E-V) contact pair wedges when both *p* and A are less than (90 + a). To see **this, con*id*r** the three cases of (E-V) sqtieeiing contact shown in Figure 35:

- In case (a), *p* is less than (30 + a), but *X h* not* Notice the contact pair line **labelled** *. **This** line lies **outside** the contact friction cooes — this implies that there is *mot*

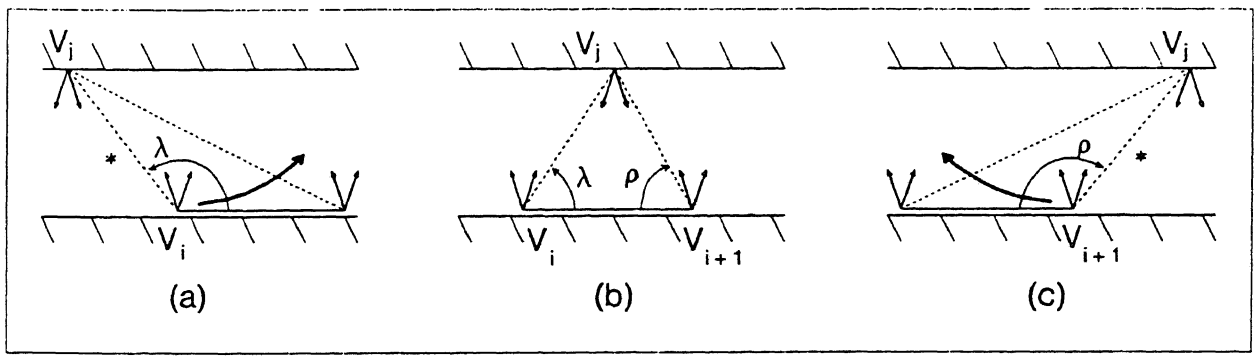


Figure 35. Three (E-V) squeezing-contact cases. In case (a), the polygon rotates counter-clockwise, since $\lambda > (90 + \alpha)$. In case (b), the polygon is wedged and does not rotate, since both λ and ρ are less than $(90 + \alpha)$. In case (c), the polygon rotates clockwise, since $\rho > (90 + \alpha)$.

enough friction to make vertices $\mathbf{V}_i$ and $\mathbf{V}_j$ stick, and the polygon slips, rotating counter-clockwise.

- In case (b), λ and ρ are both less than $(90 + \alpha)$, and the polygon is wedged between the squeezing planes. This is true despite the fact that both (V-V) contact pair lines lie outside their friction cones. The contact between $\mathbf{V}_i$ and $\mathbf{V}_j$ cannot slip, because it would result in a clockwise rotation that would cause $\mathbf{e}_i$ to enter plane #1. Likewise, the contact between $\mathbf{V}_{i+1}$ and $\mathbf{V}_j$ cannot slip, because the counter-clockwise rotation that would result is also obstructed by $\mathbf{e}_i$. Note that if either or both of the dotted lines were within the friction cone, the polygon would still be wedged, since no slipping could occur.

- In case (c), λ is less than $(90 + \alpha)$, but ρ is not. By reasoning similar to case (a) above, the polygon slips as it is squeezed, this time rotating clockwise.

This analysis provides us with a simple way to determine whether or not any given (E-V) contact pair will wedge. If λ and ρ are both less than $(90 + \alpha)$, then the contact pair will wedge; otherwise, the polygon will rotate when squeezed.

After developing the wedging conditions for (E-V) contact pairs, the wedging analysis for (E-E) contact pairs is trivial. Basically, the same conditions apply: (E-E) wedging occurs only when both λ and ρ are less than $(90 + \alpha)$. The only change to the previous (E-V) analysis is that the definitions of ρ and λ are different for an (E-E) contact pair; these definitions are illustrated in Figure 36.

Thus, each (E-V) or (E-E) contact pair can be quickly checked for wedging; if both λ and ρ are less than $(90 + \alpha)$, then the contact pair will wedge, and is considered a stable grasp that we seek. Those contact pairs that provide a stable grasp correspond to *seek lines* in the squeezing space. The seek lines for the triangle are shown in Figure 37; these seek lines show all of the stable grasp configurations that are possible for the triangle.

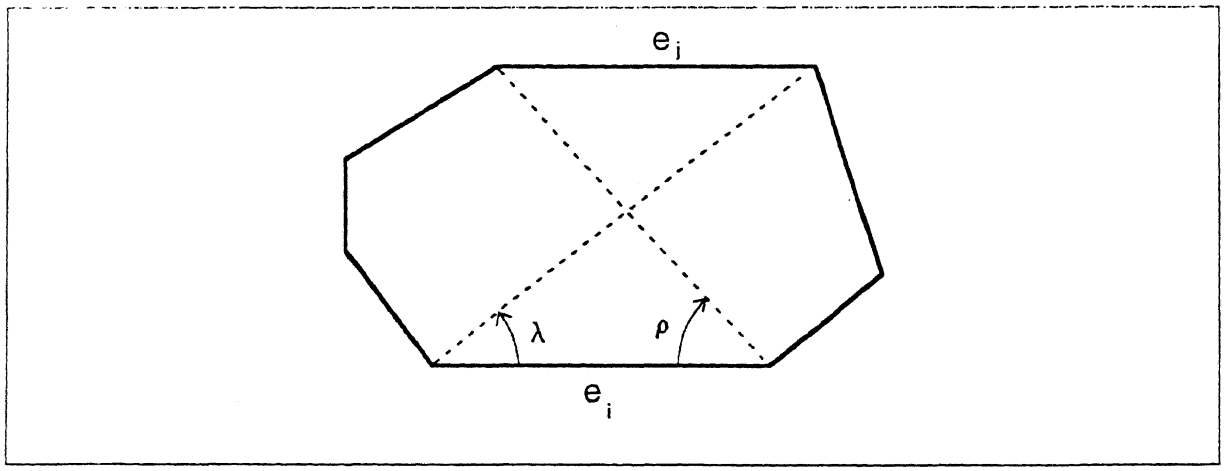


Figure 36. The angles λ and ρ for an edge-edge (E-E) contact pair.

So, to determine which contact pairs provide a stable grasp, we can take all of the (E-V) and (E-E) contact pairs found by the procedure described in the previous section, and calculate their λ and ρ angles. If both λ and ρ are less than $(90+\alpha)$, then the contact pair provides a stable grasp, and the contact pair's seek line is added to the list of seek lines for the polygon.

What Polygon Orientations Wedge in an Undesirable Configuration?

As shown in the second example presented in the beginning of this section, there are some (V-V) squeezing configurations that wedge, but are undesirable because the uncertainty in the polygon's orientation is not removed (Figure 38). It is precisely these (V-V) wedging configurations that we wish to avoid; therefore, we will proceed by examining each (V-V) contact pair in turn, determining what squeezing orientations (if any) result in wedging. We will then avoid these undesirable orientations.

Wedging occurs for a (V-V) contact pair when the contact pair line lies within the friction cone at each vertex (Figure 38). If the contact pair line lies outside the friction cones, then the vertices slip, causing the polygon to rotate.

To determine what squeezing plane orientations cause a given contact pair to wedge, consider the special-case squeezing operation illustrated in Figure 39. In this configuration, the squeezing planes are perpendicular to the contact pair line. We will denote the orientation of plane #1 in this special case as ϕ_s.

Since both squeezing planes are perpendicular to the contact pair line, the contact pair line lies within both friction cones, and the polygon is wedged. Now imagine rotating the parallel squeezing planes slightly in either direction; as long as the planes are rotated less than α away from ϕ_s, the contact pair line remains within the friction cones, and wedging continues. However, as soon as the planes are rotated further than α away from ϕ_s, the contact pair line lies outside both friction cones, and the polygon slips and rotates as the

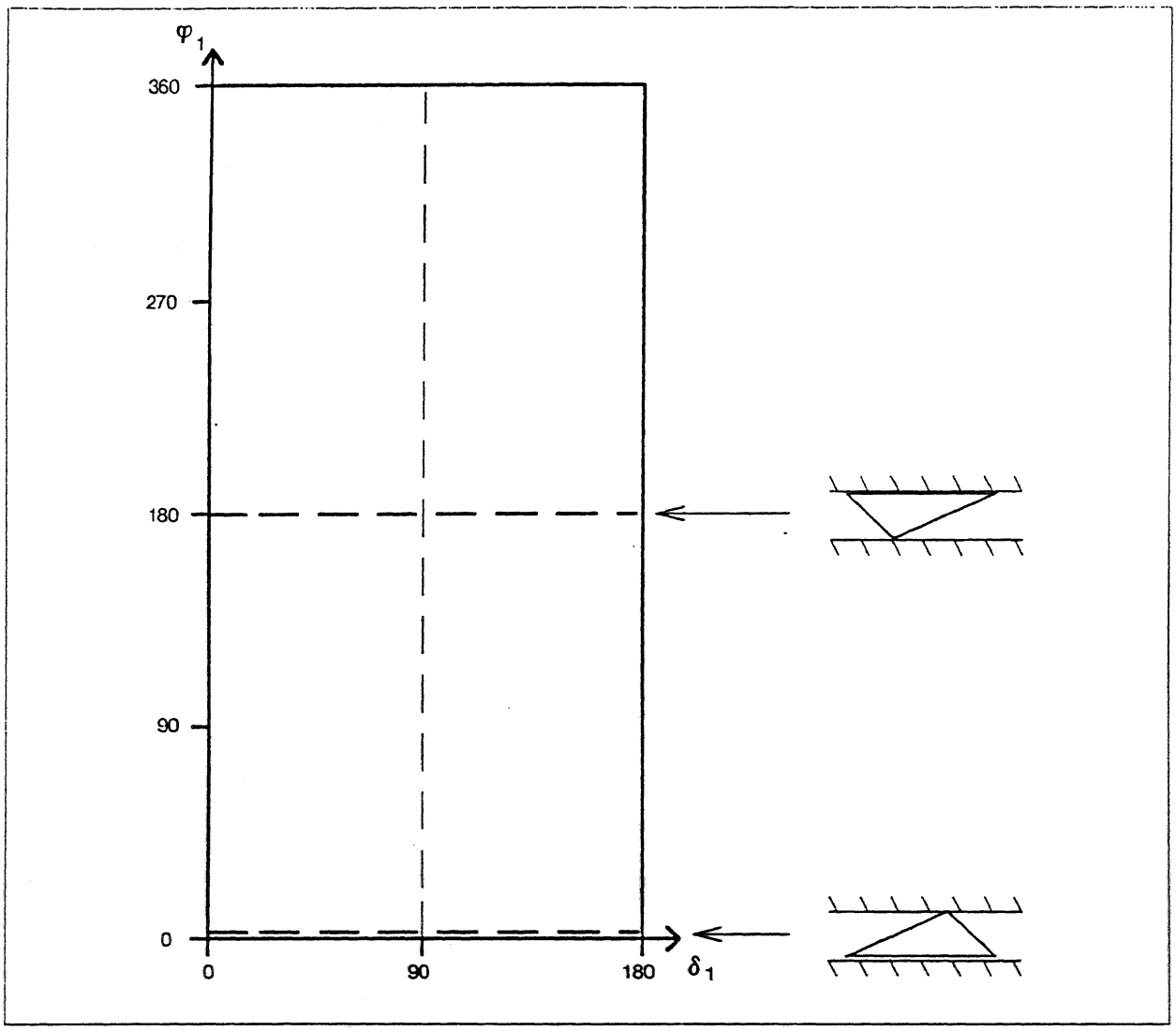


Figure 37. Seek lines in the squeezing space. These seek lines indicate all of the triangle's stable grasp configurations. The grasp configurations corresponding to each seek line are shown on the right.

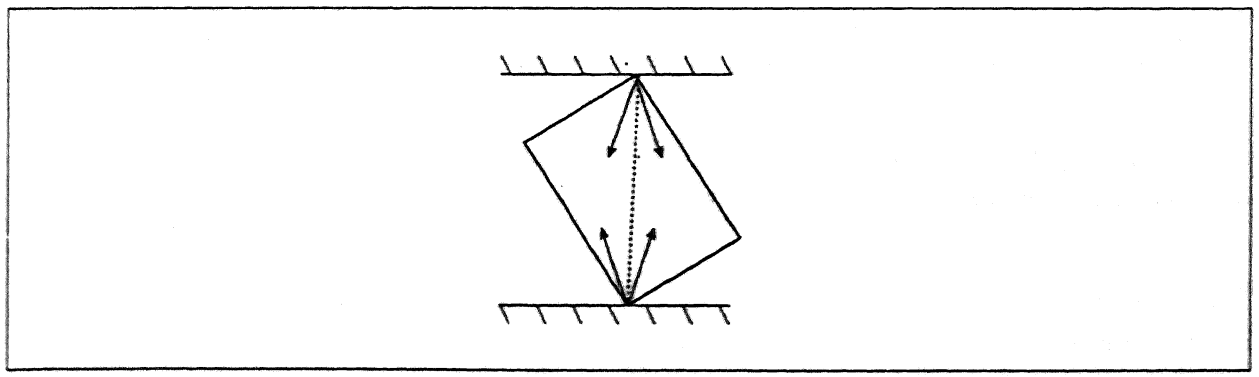


Figure 38. A (V-V) contact pair that is wedged. No matter how hard the polygon is squeezed, it will not rotate.

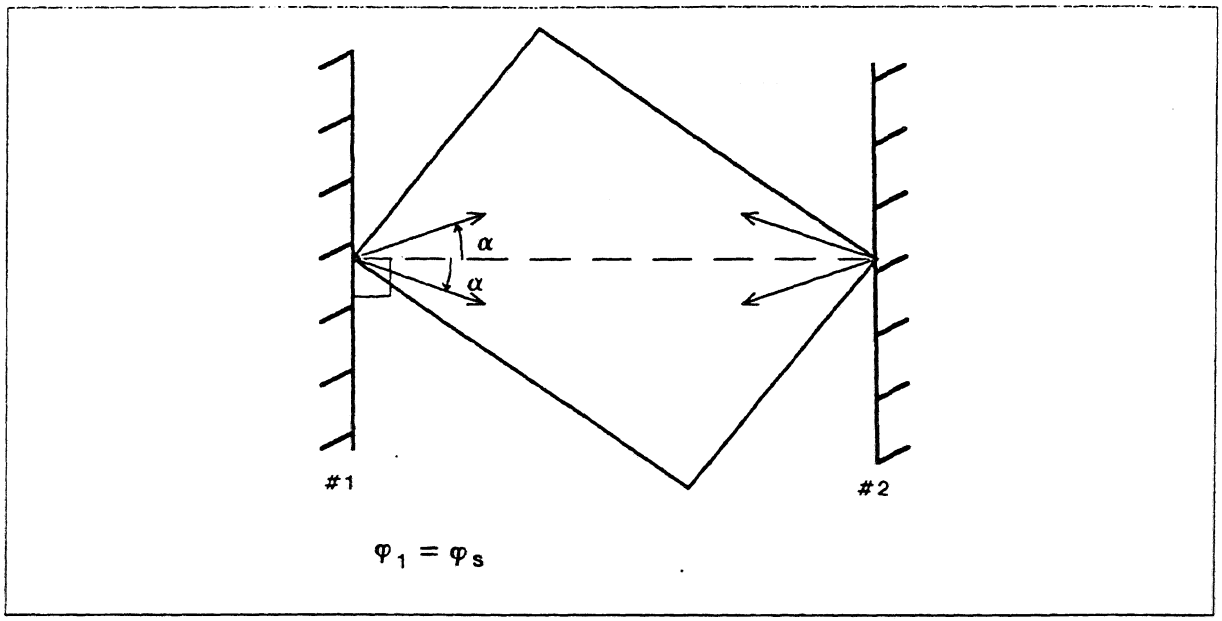


Figure 39. Special squeezing case. The contact pair line is perpendicular to both squeezing planes.

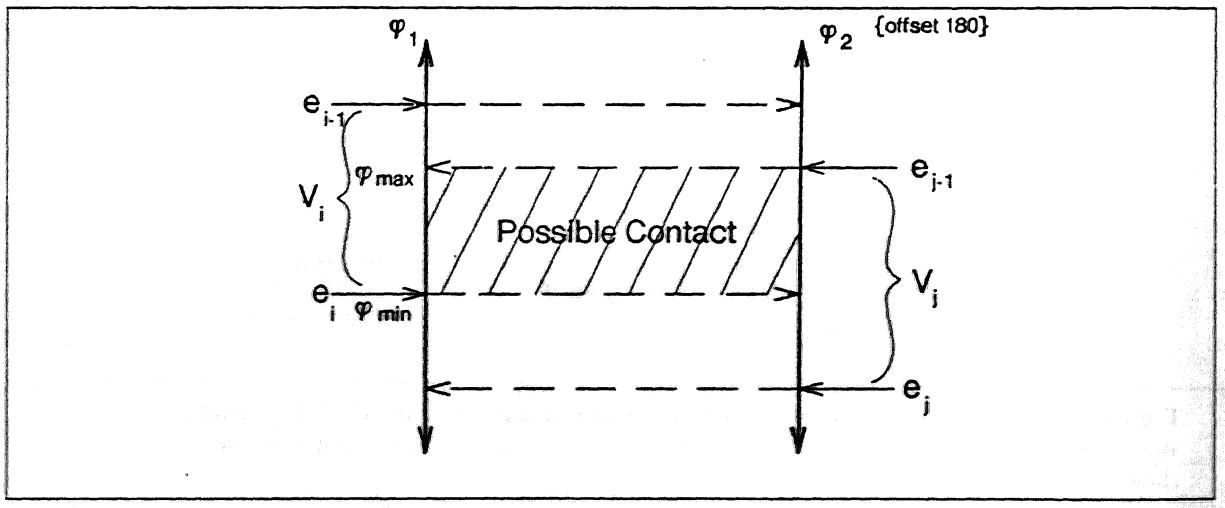


Figure 40. The region of possible contact with a contact pair. For the contact pair $(\mathbf{V}_i, \mathbf{V}_j)$, simultaneous contact with both squeezing planes can occur in the interval (ϕ_{min}, ϕ_{max}).

planes squeeze together. Thus, undesirable (V-V) wedging occurs for plane orientations with ϕ_1 values in the range $(\phi_s - \alpha, \phi_s + \alpha)$.

However, not all values of ϕ_1 within $(\phi_s - \alpha, \phi_s + \alpha)$ are realizable for some contact pairs, since the edges adjacent to the contact vertices may prevent the squeezing planes from reaching the wedging orientations (recall Figure 31, where wedging was impossible). Therefore, it is necessary to determine what ϕ_1 interval corresponds to possible contact configurations with a given contact pair, and then compare this interval with the ϕ_1 interval that causes wedging.

Consider the representation of a contact pair (V_2-, Vy) in squeezing space (Figure 40). As before, the region of V_2- first contact with plane -//-I is shown on the left, while the region of Vy first contact with plane //2 is shown on the right. Recall that the contact pair-finding procedure discovered this contact pair by detecting the overlap between the V_t Qrst-contact region and the Vy first-contact region (shaded area); this overlap also conveniently corresponds to the realizable plane orientations for the contact pair. Thus, we can infer from the diagram that the interval ($<f>_{min}$ ^*mnx)* °f realizable plane orientations for any given (V;, Vy) contact pair can be easily calculated as follows (recall that £,- is the value of e.;'s edge-flat line):

$$cj>_{min} = max\{\& \{tj - 180)]$$

$$(j>rnax = "\dot{W}n[\&-1, (f/-l \sim 180)]$$

Note that $<f>_{min}$ is always less than $<f>_{max}$ °r ^c this contact pair wouldn't have been detected in the first place.

At this point in the analysis, we know that:

- Contact occurs for values of $<f>i$ in the non-empty interval $(<f>_{ri}i_n, <f>mnx)>$
- Wedging occurs for values of $4>i$ in the non-empty interval $(<f>_3 — a, <\$ + a)$.

The intersection of these two intervals corresponds to the range of plane orientations that wedge. Thus we can intersect these intervals to yield a, *wedging avoidance region A in* squeezing space:

$$^A = (<£mtm\ 4>max) \text{ H } [<f>_s - OL, <f>_9 + a)$$

The interval of possible contact and interval of wedging axe completely unrelated; they may coincide or partially overlap, one can be contained within another, or they may be completely disjoint. Thus, nil avoidance regions axe possible.

Computing the wedging avoidance region *A* is simple:

$$A_min = max\{4>_{min}, (<f>_{\&} - a)]$$

$$A_{max} = min[4>_{max}, (\$8 + a)]$$

(If *Amin > Amax, then A = nil)*

This calculation can be carried out for all of the polygon's (V-V) contact pairs, thus finding all of the wedging avoidante **regions** for the polygon (Figure 41}. If we **choose** a squeezing operation that stays out of these regions, then we are guaranteed that **the** polygon will not get stuck in an undesirable (V-V) wedging configuration.

So to determine what polygon orientations wedge in undesirable configurations, we

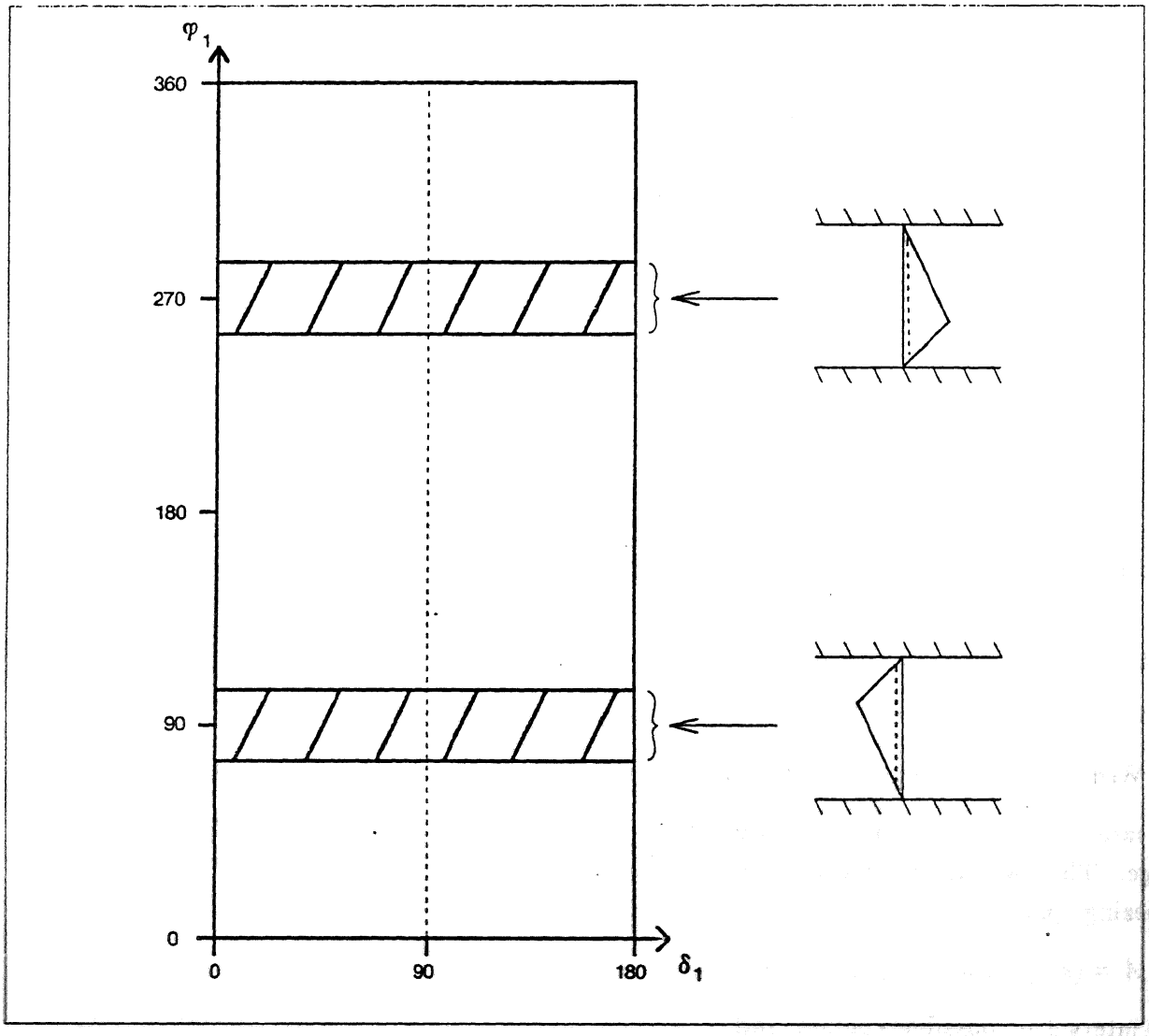


Figure 41. Wedging avoidance regions. Squeezing operations that occur within the shaded regions encounter (V-V) wedging.

can calculate the wedging avoidance region A for each (V-V) contact pair that was found by the contact pair-finding procedure. All squeezing operations that stay out of the resulting avoidance regions are guaranteed not to get stuck in any undesirable (V-V) wedging configurations.

Combining Seek Lines and Avoidance Regions

After we have found all of the seek lines and avoidance regions for a polygon, we can combine these seek lines and avoidance regions to form a *seek/avoidance diagram*, which fully describes the behavior of the polygon under squeezing (Figure 42).

At this point, it is worthwhile to investigate the motion of the polygon when subjected to squeezing operations that lie in the "white space" between avoidance regions and seek

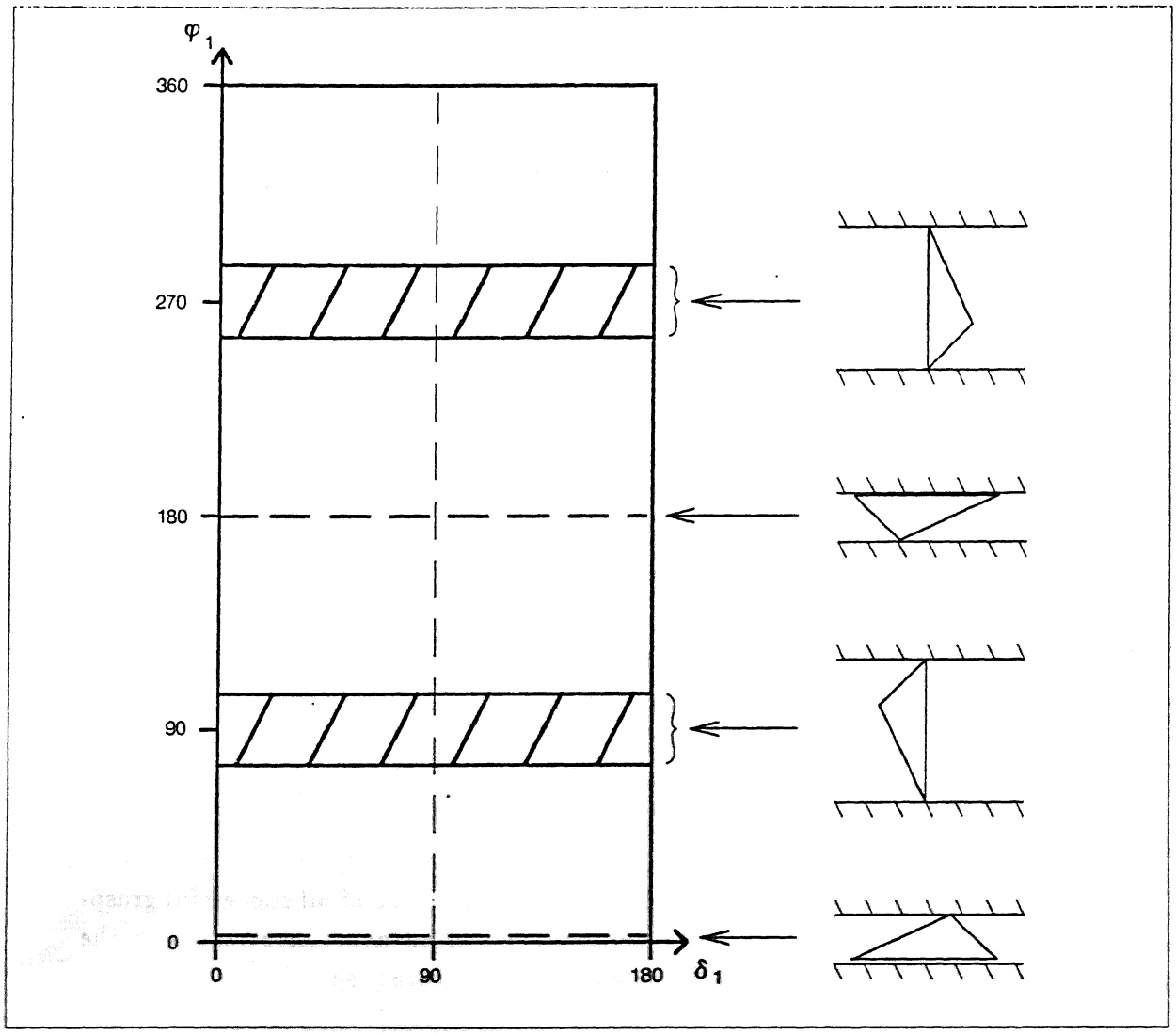


Figure 42. Wedging seek/avoidance diagram. Squeezing operations that take place within the shaded avoidance regions result in undesirable (V-V) wedging. The dashed lines are seek lines that correspond to (E-V) or (E-E) contact pairs that wedge when squeezed, thus providing a stable grasp.

lines.

Consider the example shown in Figure 43(a). The rectangle is being squeezed along its (V_1, V_3) contact pair and is not wedged; as the planes squeeze together, the rectangle rotates clockwise until $\mathbf{e}_1$ lies flat against plane #1, and stable grasping is achieved. The corresponding segment of the rectangle's seek/avoidance diagram is shown in Figure 43(b); the rectangle starts at the heavy dot near the avoidance region, and moves vertically down to the $\mathbf{e}_1$ seek line (as with pushing, we require that δ is constant).

Notice that the rectangle's rotation direction is constrained by the squeezing planes; the rectangle can only rotate clockwise, since counter-clockwise rotation would cause the

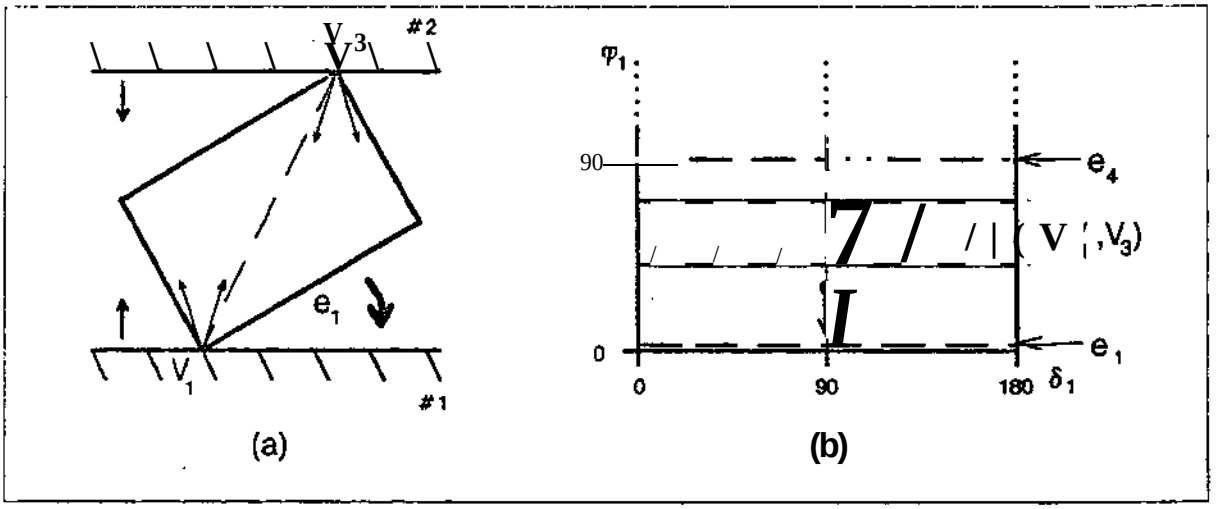


Figure 43. The rotation of a. rortangle UJK!(T squeezing, (a) As tlio planes squeeze together, the rectangle rotates clockwise, (b) The stqueoxing move of (a), illustrated in the squeezing space. The rectangle starts slightly below the (V^V*) wedging avoidance region, and travels moves down to the ei stable grasp seek line,

vertices Vj and V3 to enter the squeezing planes. Hence an analysis of rotation direction can be avoided completely; polygons under squeezing always rotate away from avoidance regions toward seek lines. Therefore, we know that any squeezing operation that begins between two avoidance regions will always rotate directly to the seek line between the avoidance regions, without wedging along the way.[2]

Thus, the resulting seek/avoidance diagram indicates the space of all successful grasping motions; any squeezing operation that begins outside of an avoidance region in the diagram is guaranteed to terminate in a stable grasp at the nearest seek line.

How Can We Include Pushing Information?

At this point, we have developed the seek/avoidance diagram, which characterizes the outcome of all possible squeezing operations. However, this result is not directly applicable to real-world squeezing operations because, in the real world, two-plane contact does not exist throughout all phases of a squeezing move. Because the location of the object is not precisely known, it is unlikely that both squeezing planes will touch the object simultaneously; rather, the squeezing move will consist of a "pushing phase* when only one plane touches the polygon* followed by a ^squeezing phase*[1]s when both planes are in contact with the polygon* Further^ it is not known a *priori* which plane will contact the polygon first. Thus, we need to devise a method of integrating the previous pushing

[2]It turns out that there »' a single *m*t*k* fine brtween every pair of avoidance regions. However»it is *wot* true that tlierc ›» *wbite PfMtce* to cither ttidc of every peck line. Some neck lixic* are adjacent to, or buried within, avoidance regions. Titus, not every *mwk* line that provides stable prehension *"m* reachable through itqiseeaüug. Aa rxwmpfc of th'» wifl be prevented in a later aection, ^Wlten Simple Squeezing Paila."*

analysis with our new knowledge about squeezing.

Our strategy for integrating the pushing and squeezing analyses will be as follows:

A. We will first assume that plane #1 touches first; this implies that there is a pushing phase of unknown duration where plane #1 pushes the polygon, followed by a squeezing phase where both planes squeeze the polygon. Using this assumption, we will combine the push-stability diagram for plane #1 and the squeezing seek/avoidance diagram to produce the *offset-grasp diagram*, which is the space of squeezing moves that are guaranteed to succeed, given that plane #1 touches the polygon first.

B. We will pursue the same sort of strategy for plane #2, to produce the space of squeezing moves that are guaranteed to succeed, given that plane #2 touches first.

C. Finally, we will intersect these two diagrams to produce the space of squeezing moves that are guaranteed to succeed, regardless of which plane touches first. This diagram is the *squeeze-grasp diagram*, and is the primary result of this paper.

Objective A

Let us proceed with point A above. Assume that plane #1 touches the polygon first, and that plane #1 pushes the polygon for some time of unknown duration before plane #2 comes in contact with the polygon. Between the time that plane #1 touches the polygon and the time that plane #2 touches, the motion of the polygon is described by its push-stability diagram (recall Section III). However, plane #2 may come in contact with the polygon at any time, immediately switching the motion to that described by the squeezing seek/avoidance diagram.

Since we cannot predict when this contact will occur, we must choose operations that can have pushing phases of any duration, without ever placing the polygon in any of the avoidance regions of the seek/avoidance diagram. We can find the space of all such moves by superimposing the push-stability and seek/avoidance diagrams and rejecting those moves which might get caught in an avoidance region.

This rejection process is illustrated in Figure 44. Figure 44(a) illustrates the push-stability diagram for the triangle, superimposed with the triangle's seek/avoidance diagram. We wish to reject all pushing operations that can lead to wedging if plane #2 touches the triangle at an inconvenient time; what regions of the diagram should be deleted?

Obviously, pushing moves that begin in one of the avoidance regions are rejected because an immediate contact with plane #2 would cause the squeeze-grasp to fail. Further, all pushing moves that start outside an avoidance region but pass through an avoidance region at some time during the move should be rejected, since plane #2 might contact the triangle while it is within the avoidance region. Point A in Figure 44(a) indicates a move of this type.

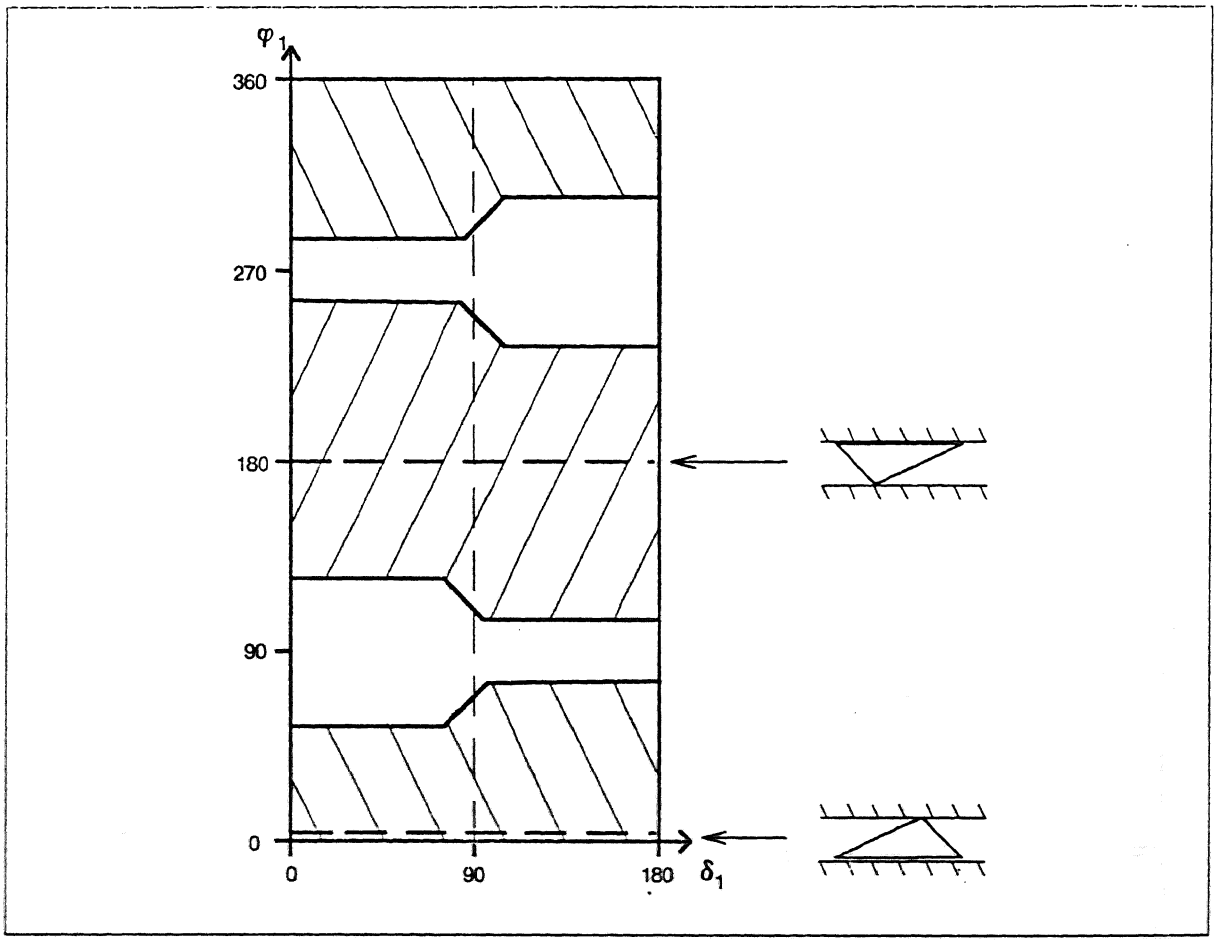


Figure 46. The squeeze-grasp diagram for the triangle. The squeeze-grasp diagram shows the space of all squeezing moves that are guaranteed to succeed, regardless of which plane touches first. This diagram is the result of intersecting the diagrams of Figure 44(b) and Figure 45(b).

This can be accomplished by applying a transform to plane #2's offset-grasp diagram. A short bit of algebra reveals that the transform is quite simple: To generate the offset-grasp diagram for plane #2 in terms of plane #1's coordinates, the normal offset-grasp diagram for plane #2 is vertically phase-shifted 180 degrees, and reflected about the $\delta = 90$ line. The result of applying this transform to the triangle's offset-grasp diagram is shown in Figure 45(b). This transform arises from the algebraic relationships between the parameters for plane #1 and plane #2 (see squeezing introduction above).

The resulting diagram displays the space of push-squeezing moves that are guaranteed to succeed, assuming plane #2 touches first, expressed in plane #1's coordinates (Figure 45(b)). This completes objective B of the above strategy.

Objective C

Objective C wraps-up the squeezing analysis. The diagrams produced in objectives A and B delineate the space of guaranteed push-squeezing moves, assuming respectively

that plane #1 or plane #2 touches the polygon first. The areas that are common to both diagrams represent those squeezing moves that are guaranteed to succeed if plane #1 touches first, or if plane #2 touches first. So, by intersecting the diagrams, we will generate the space of all squeezing moves that are guaranteed to succeed, regardless of which plane touches first. This final result is the *squeeze-grasp diagram* for the polygon.

The squeeze-grasp diagram for the example triangle is shown in Figure 46; this diagram is the result of intersecting the shaded regions of the diagrams in Figures 44(b) and 45(b). Any squeezing move that begins within a shaded area of the squeeze-grasp diagram is guaranteed to succeed and provide a stable grasp of the triangle in the configuration shown.

So to include pushing information into our squeezing analysis, we simply execute the following steps:

1. Superimpose the squeezing seek/avoidance diagram over the push-stability diagram to yield the offset-grasp, diagram, which is the space of squeezing moves that are guaranteed to succeed, assuming plane #1 touches first.

2. Transform the offset-grasp diagram for plane #2 into plane #1's coordinates by vertically phase-shifting the diagram 180 degrees, and reflecting about the $\delta = 90$ line. The result is the space of all squeezing moves that are guaranteed to succeed, assuming that plane #2 touches first.

3. Intersect the results of Steps 1 and 2 to yield the squeeze-grasp diagram, which delineates the space of squeezing moves that are guaranteed to succeed, regardless of which plane touches first.

Summary — Two-Plane Squeezing

This completes our investigation of two-plane squeezing; a short summary of the entire squeezing analysis follows.

The pairs of polygon features that can make simultaneous contact with both squeezing planes can be found by projecting the first-contact regions for plane #1 onto the first-contact regions for plane #2, yielding a table of all of the polygon's contact pairs.

The polygon orientations that provide a stable grasp can be found by calculating the angles λ and ρ for each (E-V) or (E-E) contact pair. If both angles are less than $(90 + \alpha)$, then the contact pair wedges when squeezed, thus providing a stable grasp. Each contact pair that provides a stable grasp corresponds to a seek line in the squeezing space; the resulting seek lines show all of the stable grasp configurations that are possible for the polygon.

The polygon orientations that should be avoided because they cause undesirable (V-V) wedging are found by calculating the wedging avoidance region A for each (V-V) contact pair. The avoidance region A is found by intersecting the intervals $(\phi_s - \alpha, \phi_s + \alpha)$ and (ϕ_{min}, ϕ_{max}), where $(\phi_s - \alpha, \phi_s + \alpha)$ is the range of plane orientations where the particular (V-V) contact pair wedges, and (ϕ_{min}, ϕ_{max}) is the range of realizable plane orientations for the contact pair.

The resulting seek lines and wedging avoidance regions are then combined to form a single squeezing seek/avoidance diagram, which fully describes the behavior of the polygon under squeezing.

Finally, this squeezing information can be integrated with previously-derived pushing information by superimposing the squeezing seek/avoidance diagram on the push-stability diagrams for plane #1 and plane #2, and then intersecting the resulting spaces. This process produces the *squeeze-grasp diagram* for the polygon, which shows the space of all squeezing moves that are guaranteed to succeed in grasping the object, while simultaneously removing two degrees of uncertainty from the object's position.

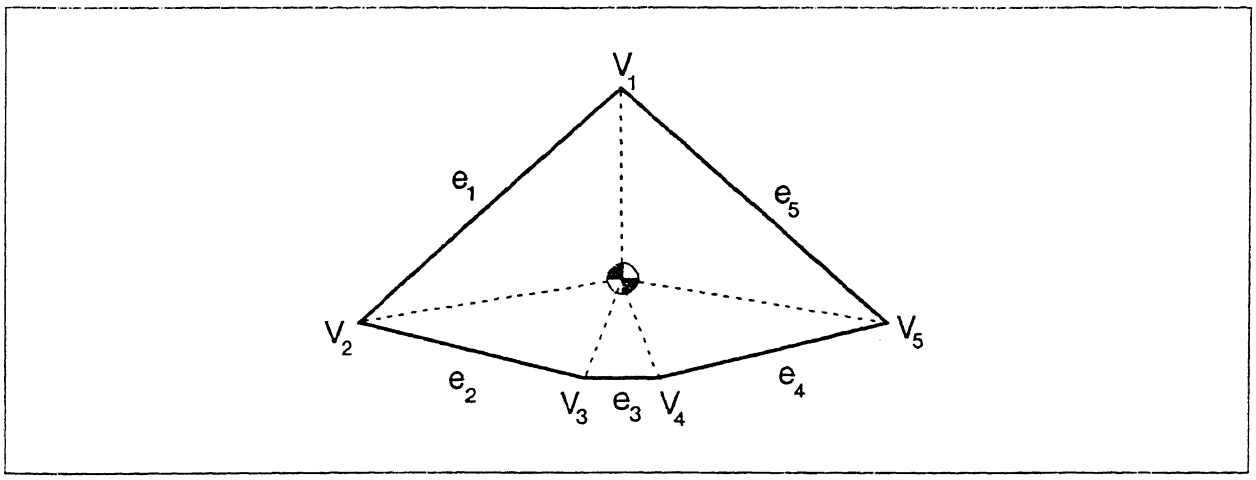


Figure 47. The "roly-pointy" object.

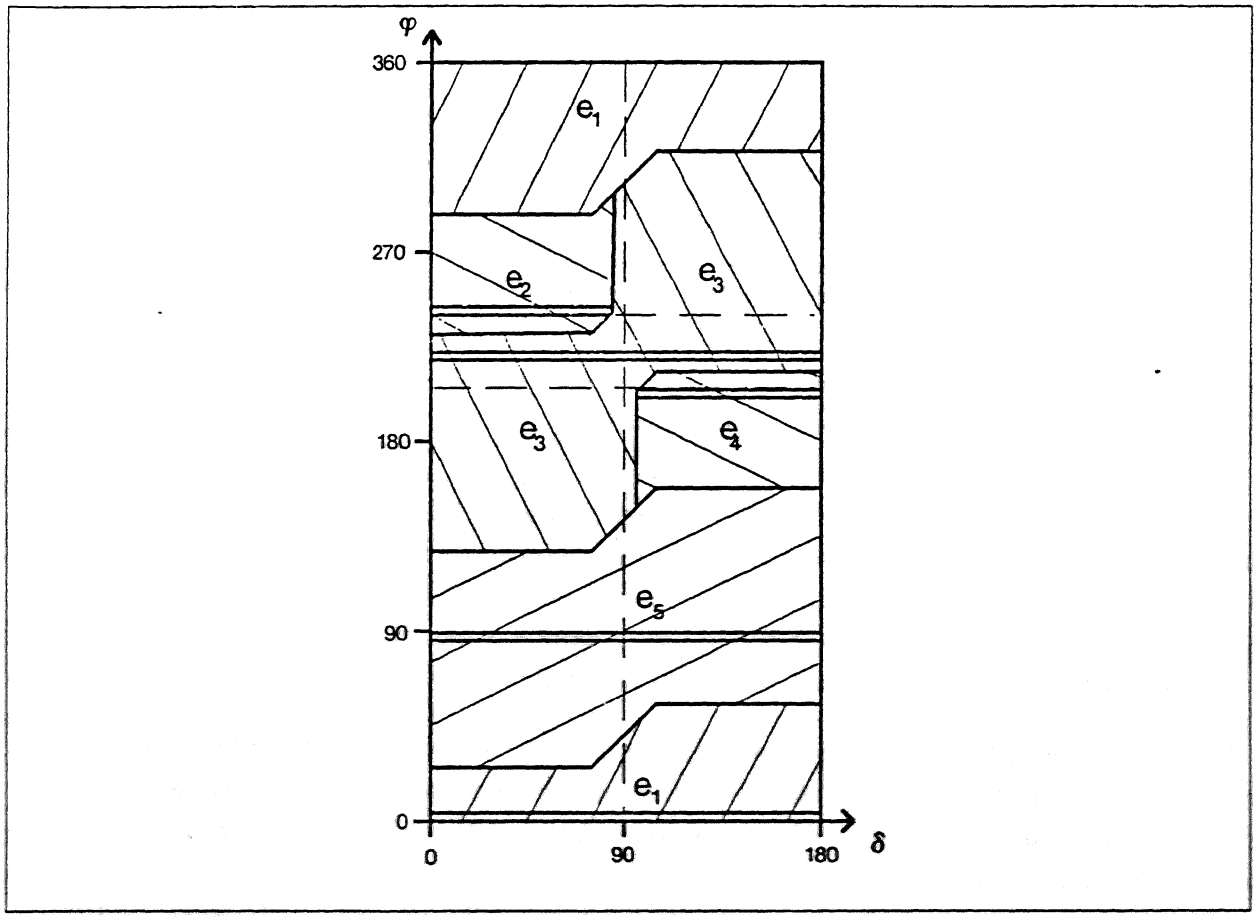


Figure 48. Push-stability diagram for the roly-pointy.

V. When Simple Squeezing Fails

Consider the object shown in Figure 47, which we will refer to as a "roly-pointy."

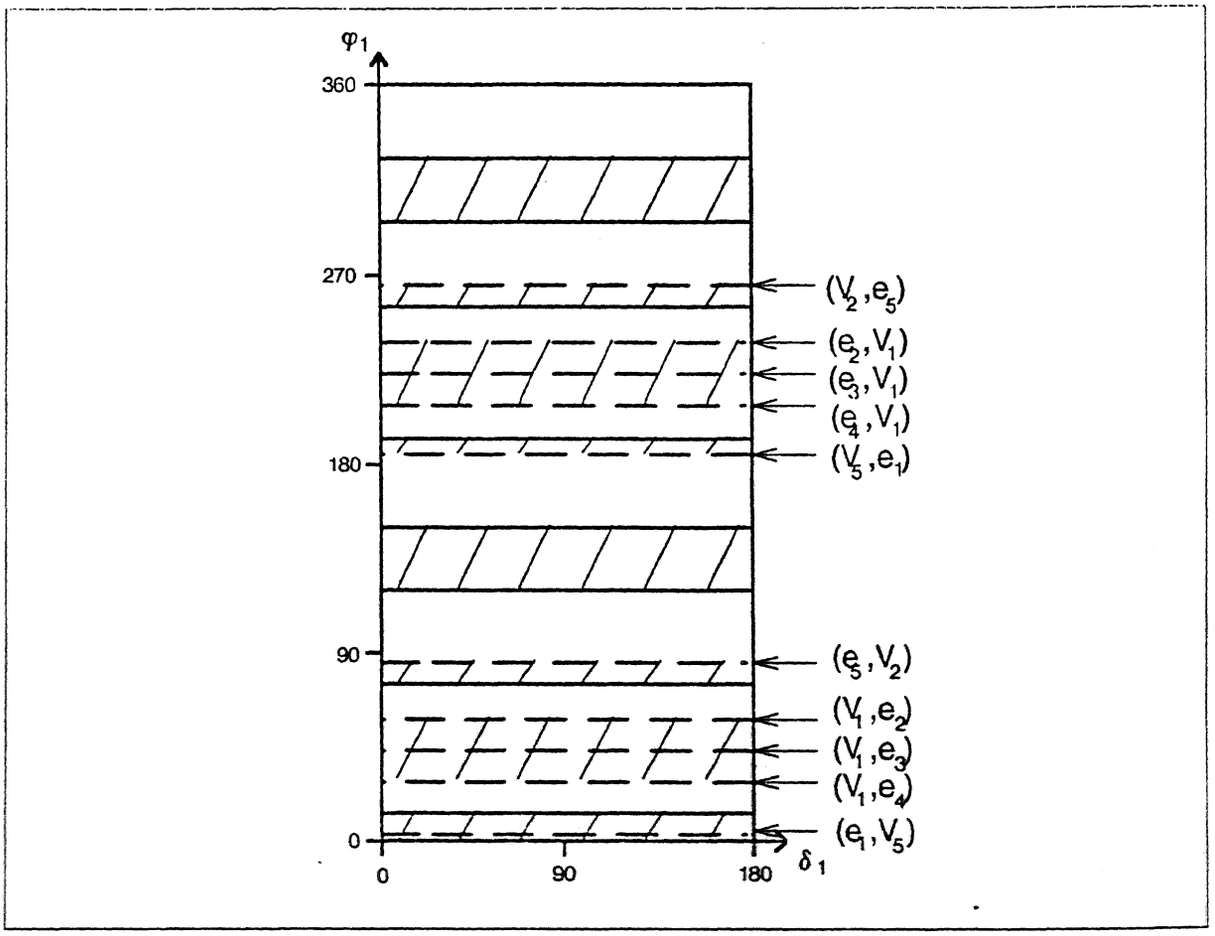


Figure 49. Roly-pointy seek/avoidance diagram.

This object is interesting because of its squeeze-grasp diagram, which is developed in Figures 48–57. Figure 48 shows the push-stability diagram for the roly-pointy, produced by the method presented in Section II. The wedging seek-avoidance diagram for the roly-pointy is shown in Figure 49; notice the many avoidance regions, and that several seek lines are adjacent to or contained within avoidance regions.

With this object, the move rejection process is very severe. After the seek-avoidance diagram is superimposed over the push-stability diagram and the moves that can wedge are removed, only a few small regions remain. Figure 50(a) shows the space of push-squeeze moves that are guaranteed to succeed, assuming that plane #1 touches first, while Figure 50(b) shows the same regions, assuming that plane #2 touches first. However, notice that the intersection of these diagrams is nil! Therefore, there is no possible squeeze-grasp that is guaranteed to succeed, even in the presence of very small uncertainty.

How can this be so? No squeezing move is possible for this object because there is no (E-V) contact pair which can be reached by a squeezing move where either plane may touch first.

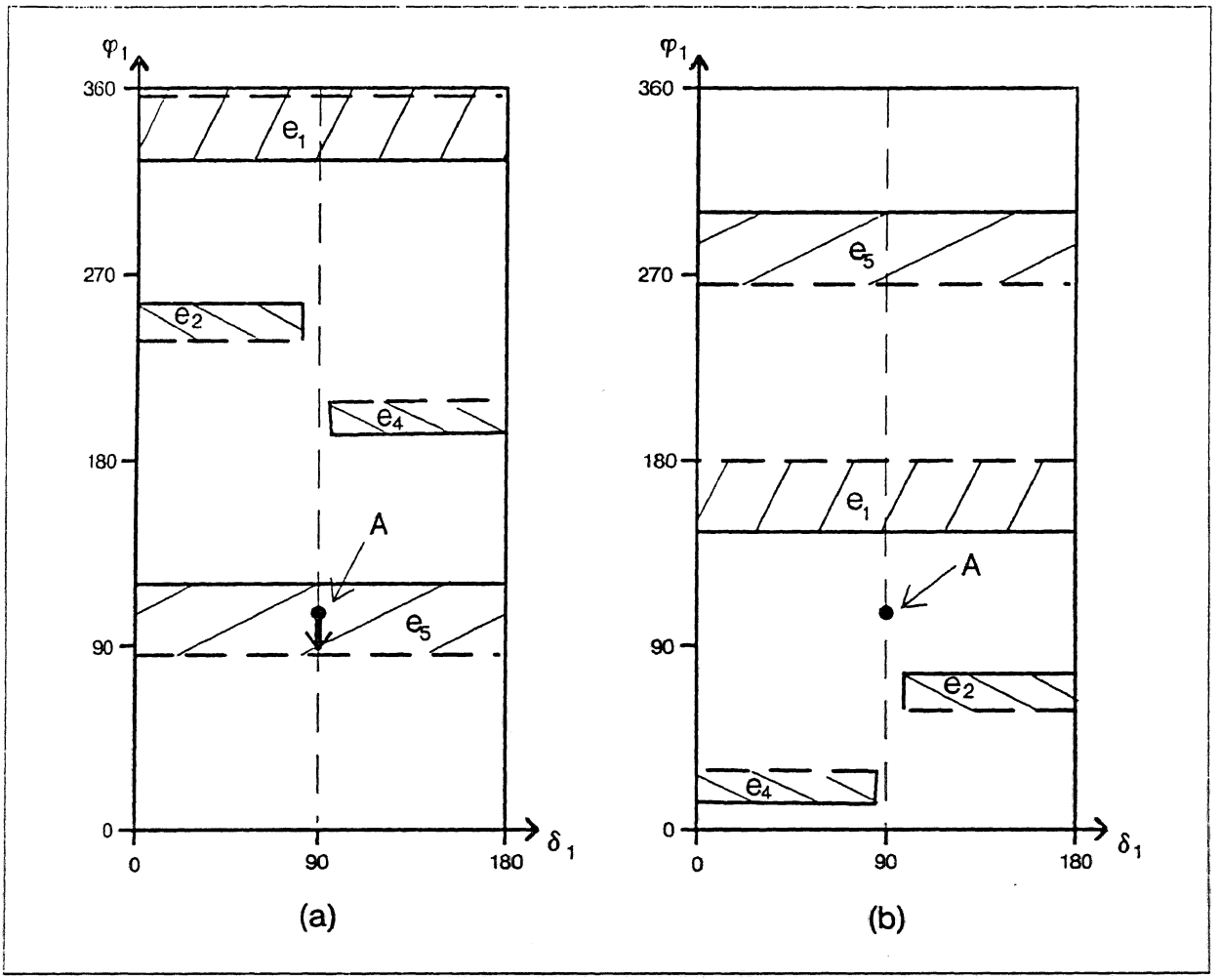


Figure 50. Roly-pointy diagrams after rejection of moves that may (V-V) wedge. (a) Assuming plane #1 touches first. (b) Assuming plane #2 touches first. Notice that the intersection of (a) and (b) is nil.

For example, consider the squeezing move labelled A in Figure 50. Since this move is in a shaded region of Figure 50(a), A represents a squeezing move that is guaranteed to succeed if plane #1 touches first.

To see why this is the case, we will examine the result of applying the squeezing move A to the roly-pointy, first assuming that plane #1 touches first, and then assuming that plane #2 touches first.

Figures 51 and 52 show move A's result, assuming that plane #1 touches first. Here we assume that plane #1 makes contact before plane #2, and pushes the roly-pointy for some time of unknown duration before plane #2 makes contact. If plane #2 touches right away, then the roly-pointy slips as it is squeezed until edge e_5 is flat against the plane (Figure 51). If plane #2 doesn't make contact for a long time, then the roly-pointy rotates clockwise under pushing until e_5 is flat against the pushing plane, and then e_5 remains flat against

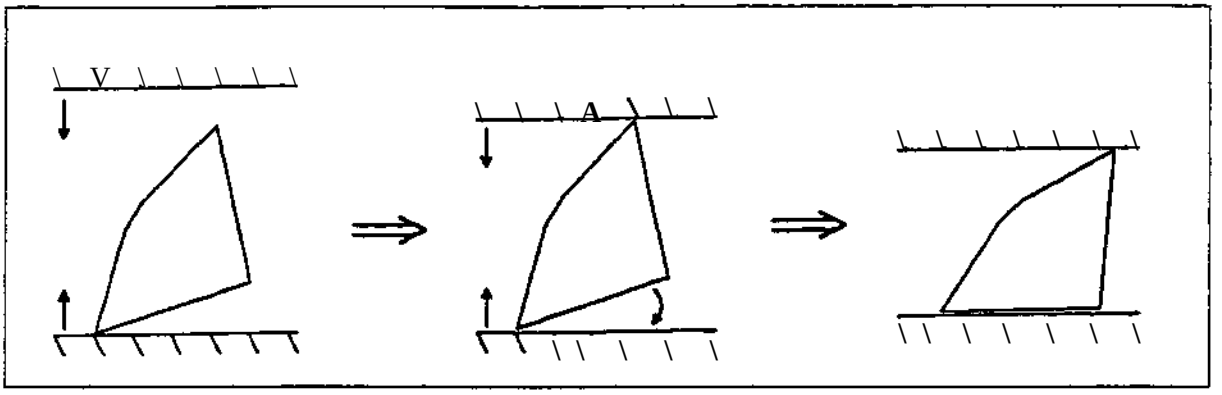


Figure 51. Tlie result of applying the squeezing move A of Figure 50. assuming that plane #1 touches first, and plane #2 makes contact immediately. When the move is finished, edge C5 of the roly-pointy is flat against plane #1.

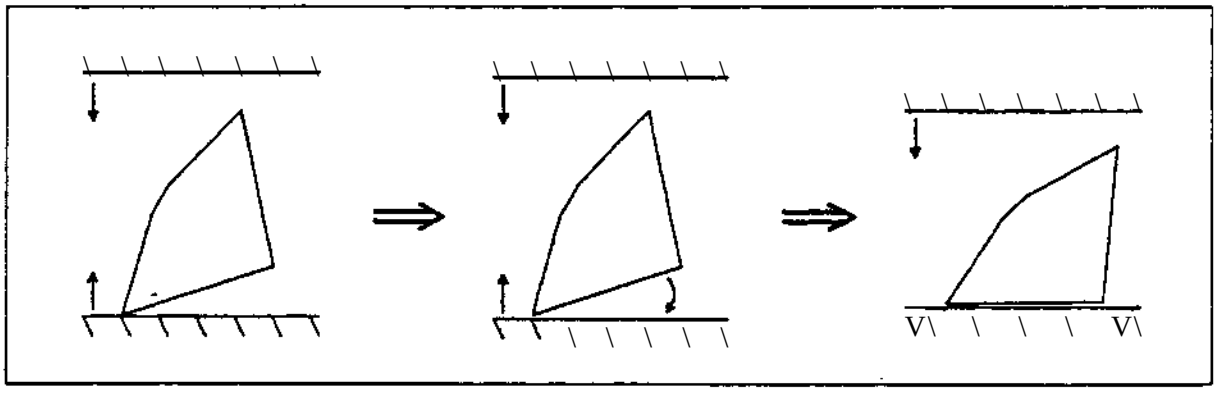


Figure 52. The result of applying the squeezing move A of Figure 50, assuming that plane #1 touches first, and there is a long delay before plane #2 makes contact. As in Figure 51, edge e$ is Bat against plane #1 when the more is finished.

the pushing plane until plane #2 eventually makes contact, completing the squeezing move (Figure 52). Notice that the roly-pointy ends up in the same final configuration in either case. This is why move A is in a shaded region of Figure 50(a); it is guaranteed to succeed as long as plane #1 touches first.

However, our results are not as good if we apply move A to the roly-pointy and plane #2 touches first (Figure 53). If plane #1 touches right away* then the polygon slips as it is *squeezed* until €5 is fiat against plane #1. and the grasp succeeds as before (Figure 53). However, if plane #1 doesn't make contact for a long time, then the roly* pointy rotates clockwise under pushing by plane #2, and will continue rotating until ®$ is fiat and stable against against plane #2* *Thus,* if a long delay occurs before plane #1 makes contact, then the resulting final configuration *k* entirety different than if plane #1 made immediate contact (Figure 54). Therefore we can make no guarantee about **the** result of move A if plane #2 touches first; this explains why A is not in a shaded region of Figure 5G(b).

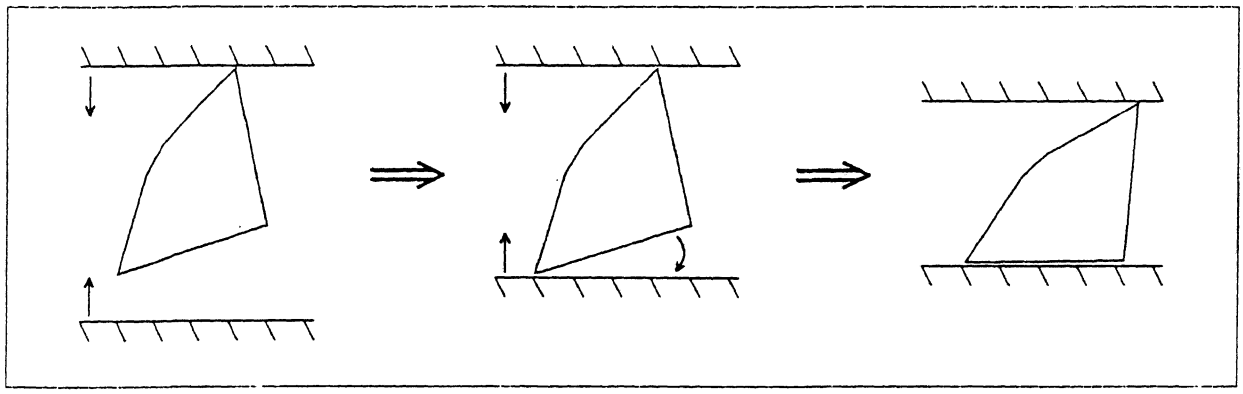


Figure 53. The result of applying the squeezing move A of Figure 50, assuming that plane #2 touches first, and plane #1 makes contact immediately. When the move is finished, edge $\mathbf{e}_5$ flat against plane #1, as in Figures 51 and 52.

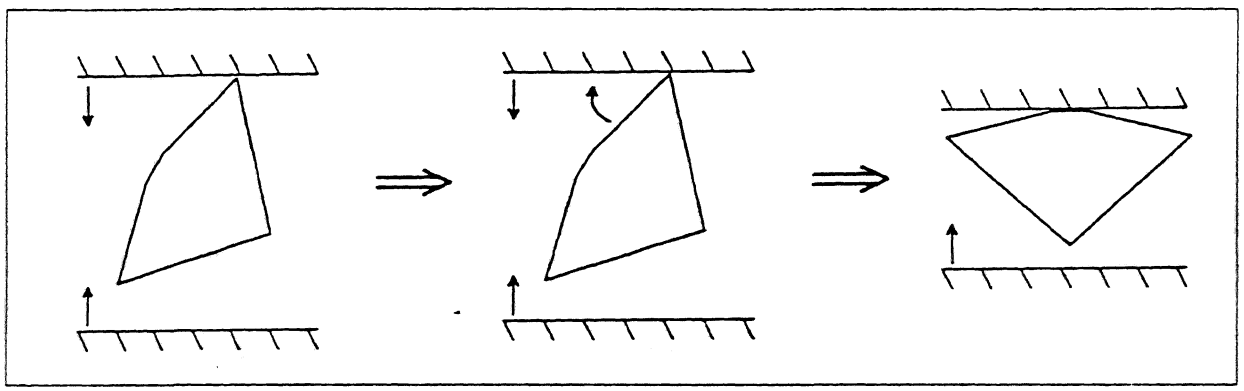


Figure 54. The result of applying the squeezing move A of Figure 50, assuming that plane #2 touches first, and there is a long delay before plane #1 makes contact. In this case, the roly-pointy rotates clockwise until $\mathbf{e}_3$ is flat against plane #2, winding up in a final configuration that is completely different than in Figures 51-53.

And so it is unhappily true: our diagrams are not lying to us, and there is no squeeze-grasp that can pick up the roly-pointy with guaranteed results. However, there is a way to successfully pick up the roly-pointy. Recall that move A was guaranteed to succeed in plane #1 touched first; if we can insure that plane #1 touches the roly-pointy before plane #2, then we can use move A with guaranteed results.

The Offset-Grasp

To meet this end, we will define the *offset-grasp* (Figure 55). The offset-grasp is executed by offsetting the gripper's initial position to one side so that it is always true that plane #1 touches before plane #2. If we employ this grasping motion, then we can plan offset-grasps by simply using the offset-grasp diagram generated in the previous section. Figure 44(b) shows the offset-grasp diagram for the triangle; Figure 50(a) shows the offset-grasp diagram for the roly-pointy. This offset-grasp diagram is the space of all squeezing moves that are guaranteed to succeed, as long as we offset the gripper sufficiently to assure that plane #1 touches before plane #2.

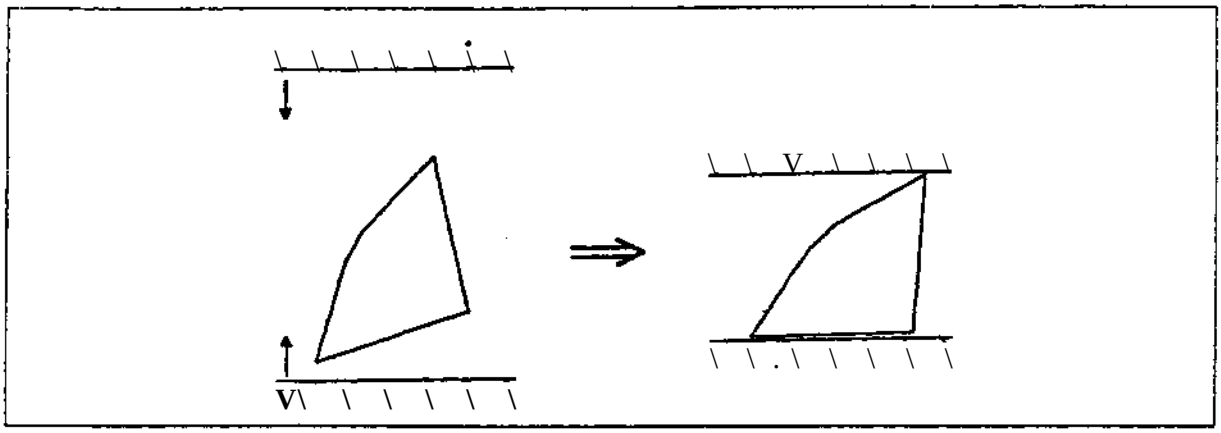


Figure 55. The offset-grasp. The grippcr fingers are offset to one side to insure that plane #i touches before plane #2.

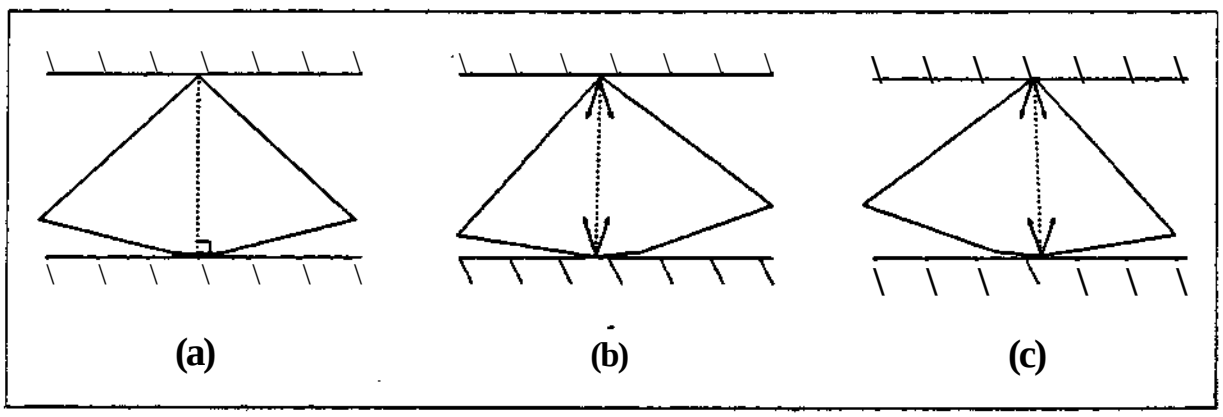


Figure f>6. Demonstration of why the roly-pointy cannot be squeeze-grasped along its (^{e}3iV|) contact pair, (a) The desired stable grasp configuration. This configuration cannot be achieved by a squeeze-grasp because orientations (b) and (c), which are very close to the desired grasp, are caught in undesirable (V-V) wedging conditions and do not slip to flat when squeezed.

The additioB of the offset-grasp considerably increases the number of grasping motions available for picking up the rofy-pointy. However, there are still some grasp configurations that we might Hie to achieve, but remain unavailable to as.

For example, suppose that for some reason we wanted to pick up the roly-poinfy fey its (e3$_s$Vi) contact pair; i.e., whore eg is lat against plane #1$_?$, and plane #2 *sqatmm* Vj, At present, this grasp is tmawlable to ms$_?$ since the squeeie-grasp^ 'diagram for tbe roly-pointy is- nil$_f$ .and there is no shaded region for «3 in the offset-grasp diagram far the roiy-pomty. Figure 56 shows why this ,gra^p is impossible to achieve through squeezing* Since the vertical an either side ef «j both wedge, orientations that are rotated only slightly clockwise or counter^clO'C^Wise from the edge-flat configuration lew! to (V-V) wedging, and do not slip to flat when sqneesed. Thus no scpeeiing motion can guarantee that #3 ends up flat against plane **#1.**

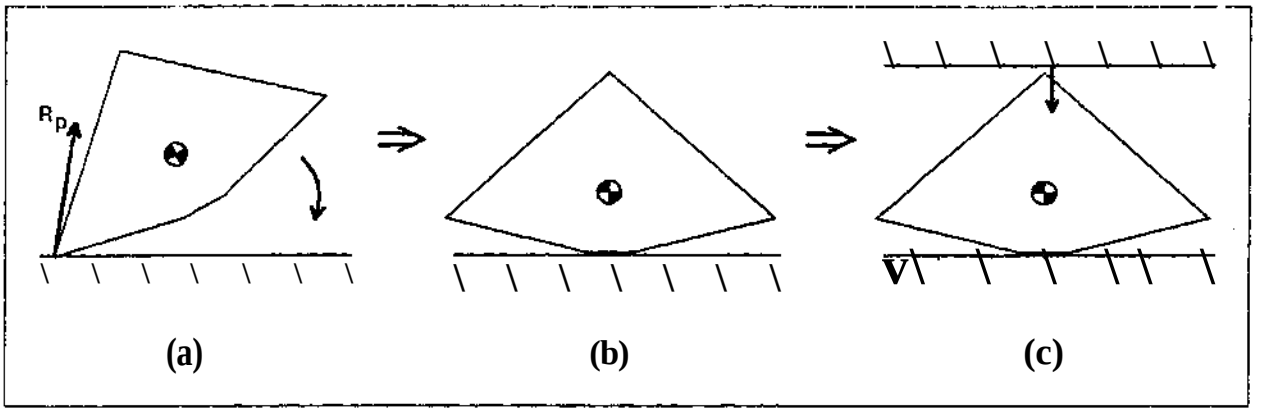


Figure 57. Tho push-grasp, (a) First the object is pushed with our plane until (b) the dosiml edge lies flat against the plane. ;uici then (o) the second phuio squeeze in to pick up the object.

The Push-Grasp

Again, there is a solution to this problem which will allow us to achieve the desired configuration: the *push-grasp* (Figure 57). A push-grasp motion consists of a pushing phase, where the object is pushed until it is in alignment with plane #1, followed by a squeezing phase, where plane #2 squeezes in to firmly grasp the object.

A push-grasp will succeed for any edge that is stable when pushed, regardless of whether or not the edge remains stable when squeezed. To see this, consider the polygon's configuration at the end of the pushing phase of the push grasp; in the squeezing space, this configuration falls either on a seek line, within an avoidance region, or in the "white space[51] between avoidance regions. We can see by the following that in all three of these cases, the grasp is guaranteed to succeed:

(1) If the configuration is on a stable-grasp seek line, then it is obvious that the grasp will succeed when the second plane squeezes the polygon.

(2) If the configuration is within an avoidance region, then we know that the polygon will wedge when squeezed, simply because the configuration is within a wedging avoidance region. However, we also know that some polygon edge is flat against the pushing plane, because we just finished our pushing operation. Therefore, since an edge is fiat against the plane and the polygon will wedge when squeezed, we know polygon is actually on an (E-V) or (E-E) seek Hue, and the grasp will succeed by the analysis in (1), above.

{3} If the configuration is in the white space between avoidance regions, then we know that when the polygon is squeezed, it will immediately rotate to the adjacent (E-V) or (E-E) seek line without encountering any avoidance regions along the way, thus providing a successful grasp.

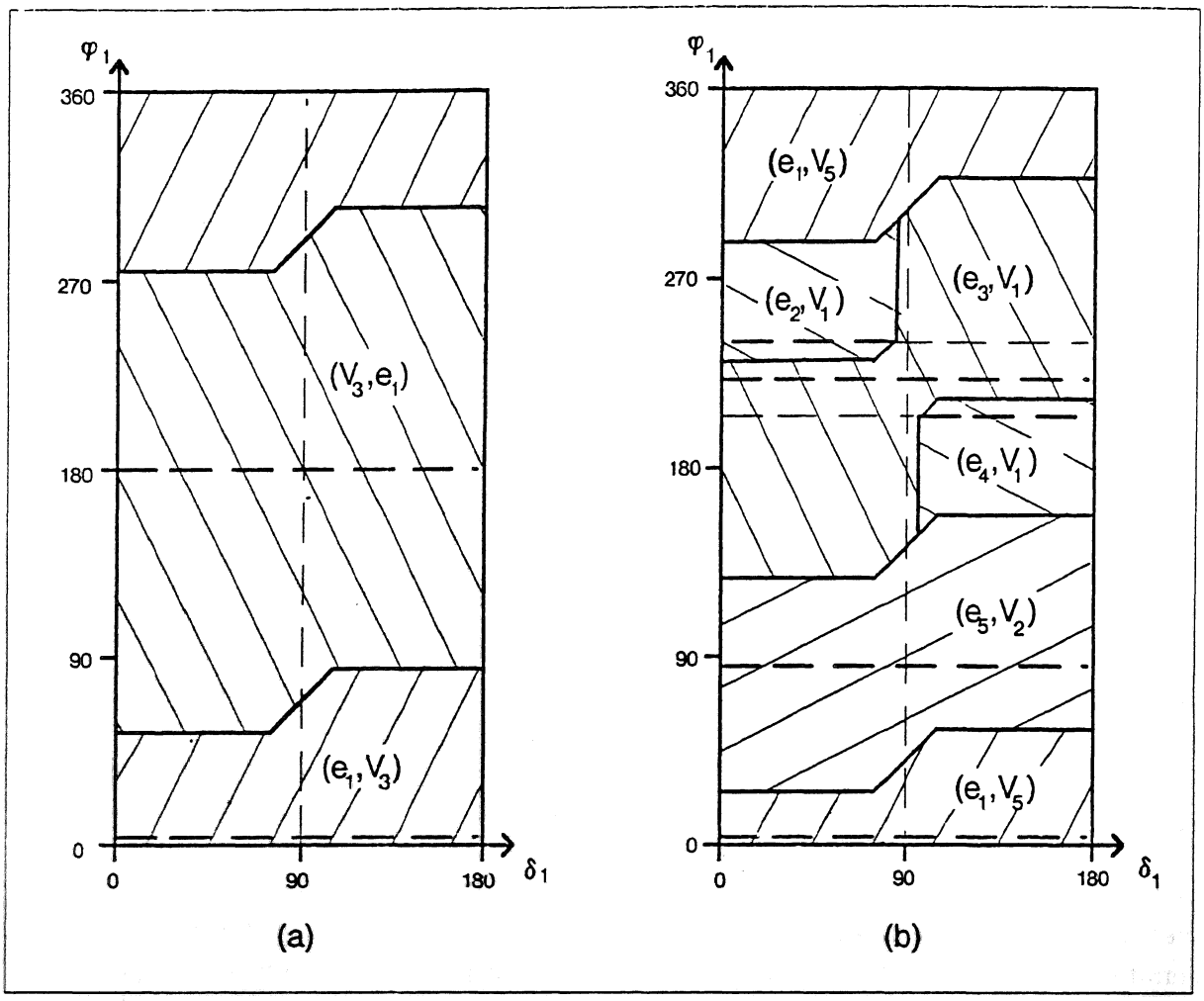


Figure 58. Push-grasp diagrams. (a) The push-grasp diagram for the triangle. (b) The push-grasp diagram for the roly-pointy.

Therefore, all points within edge-stability regions in a polygon's push-stability diagram correspond to successful push-grasps. Combining the pushing and squeezing analysis to produce a *push-grasp diagram* is then done as follows:

Each edge-stability region is examined in turn; if the corresponding edge is stable under squeezing, then the region is converted to a push-grasp region. If the corresponding edge is not stable under squeezing, then the region is converted to a push-grasp region for the seek line that the polygon will rotate to when squeezed (this can be looked up in the seek/avoidance diagram). Finally, any adjacent regions that correspond to the same final squeezing configuration are merged, yielding the complete push-grasp diagram for the polygon.

Notice that this diagram does not indicate how far the polygon has to be pushed in order to assure that the desired edge is flat against the plane; this topic, while important, will be deferred to a later paper.

The resulting push-grasp diagram represents the space of all push-grasp moves that are guaranteed to succeed in grasping the object in the final squeezing configuration shown. The push-grasp diagrams for the triangle and roly-pointy are shown in Figure 58.

In this section, we have seen an example situation where an ordinary squeeze-grasp motion will fail. In addition, we have defined two additional grasping motions: the offset-grasp and the push-grasp. These additional grasping motions extend the capabilities of our grasp-planning method in two important ways:

(a) Objects that were otherwise impossible to grasp through squeeze-grasp motions can now be successfully grasped, and

(b) More uncertainty in the object's location can be tolerated, since previous moves that were rejected because they passed through avoidance regions are now acceptable.

We now have three grasping motions: the squeeze grasp, offset-grasp, and push-grasp. The next section will discuss a method for incorporating the presence of uncertainty into our analysis.

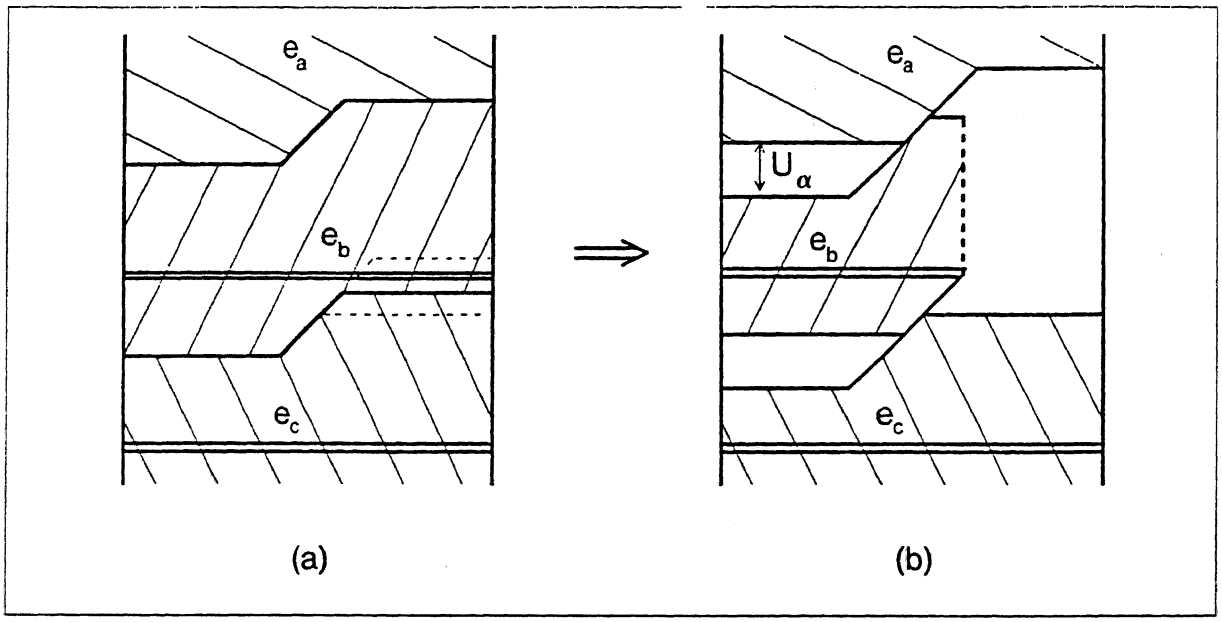


Figure 63. The effect of U_μ on stable edges. If shrinking regions to compensate for U_μ causes a divergent rotation boundary to cross a convergent boundary, then it is no longer clear whether or not that edge is stable for those values of δ where the boundaries crossed. Thus that portion of the convergent boundary must be assumed to have indeterminate stability, and all operations that finished on that portion of the convergent boundary must be avoided.

its uncertainty interval U_α, then the polygon will "rollover" to $\mathbf{e}_c$; if α is near the low end of U_α, then $\mathbf{e}_b$ will remain stable as before. Therefore we can't be sure whether or not $\mathbf{e}_b$ will remain stable or rollover to $\mathbf{e}_c$, and so the uncertainty in μ has caused the stability of $\mathbf{e}_b$ to become *indeterminate* for those values of δ that are included in the crossover of the rotation boundaries. As a result of this indeterminacy, all pushing moves that rotate into this indeterminate region must be avoided.

Thus, shrinking the stability regions of Figure 63(a) to account for U_μ is slightly more difficult than simply expanding the horizontal segments of the divergent rotation boundaries; the stability region for $\mathbf{e}_b$ must also have all pushing moves with indeterminate results removed. This can be accomplished by establishing a new vertical boundary from the intersection of the divergent and convergent boundaries, as shown in Figure 63(b); this new boundary removes exactly those moves that lead to indeterminate results.

For squeezing diagrams, the compensations for U_μ are simpler. Uncertainty in μ widens existing (V-V) avoidance regions to the worst-case α value, with the wider friction cone perhaps creating some new avoidance regions as well. This can be handled by assuming the worst-case (widest) friction cone when generating the seek/avoidance diagram.

The (E-V) and (E-E) seek lines are also affected. Since worst-case α values may yield conflicting decisions on whether or not a given (E-V) or (E-E) contact pair will wedge when squeezed, some edge-flat contact pairs may have indeterminate squeeze-stability once U_μ is

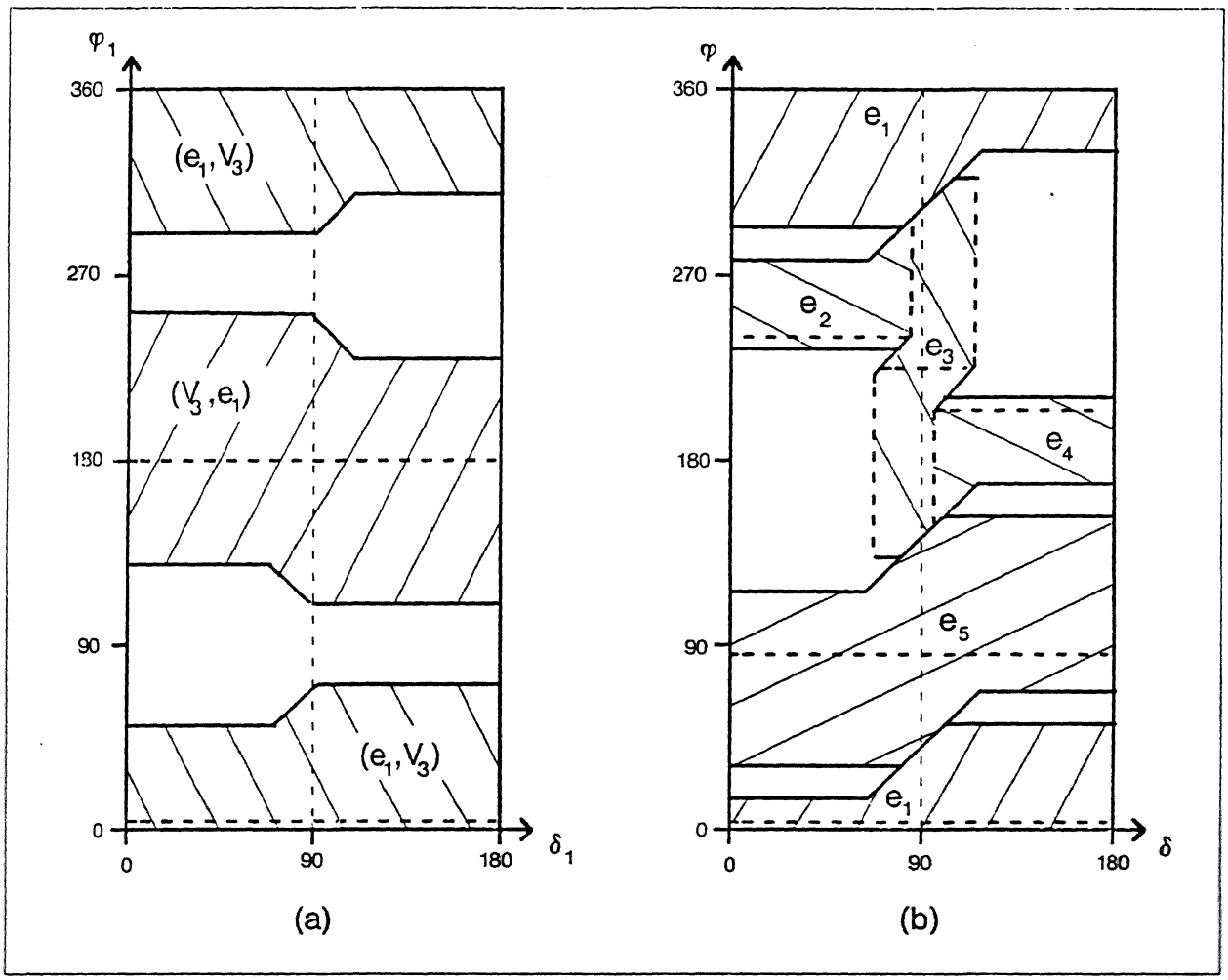


Figure 64. Squeeze-grasp and push-grasp diagrams, after shrinking regions for uncertainty in the coefficient of friction U_μ. (a) Remaining regions in the triangle's squeeze-grasp diagram, where $\mu \in (0.15, 0.35)$, implying that $\alpha \in (8°, 19°)$. (b) Remaining regions in the roly-pointy's push-grasp diagram, where $\mu \in (0.2, 0.5)$, implying that $\alpha \in (11°, 27°)$.

considered. Thus some contact pairs previously thought to slip and rotate when squeezed might actually wedge, and some seek lines expected to wedge might actually slip and rotate; these indeterminate cases should be avoided. This effect of U_μ on edge-flat contact pairs can be handled by checking each (E-V) and (E-E) contact pair individually to see if the high and low values in U_α yield a conflicting squeeze-stability conclusions; if so, then that contact pair should be avoided by adding the appropriate avoidance region to the seek/avoidance diagram.

Since uncertainty in μ can affect the stability information embedded in push-grasp and squeeze-grasp diagrams, region-shrinking for U_μ is best accomplished before shrinking for U_ϕ, U_{ori}, or U_δ. Figure 64 shows the result of shrinking regions to account for U_μ, given the triangle squeeze-grasp diagram of Figure 46 and the roly-pointy push-grasp diagram of Figure 48. In the diagram for the roly-pointy, note that the uncertainty in μ caused e_3

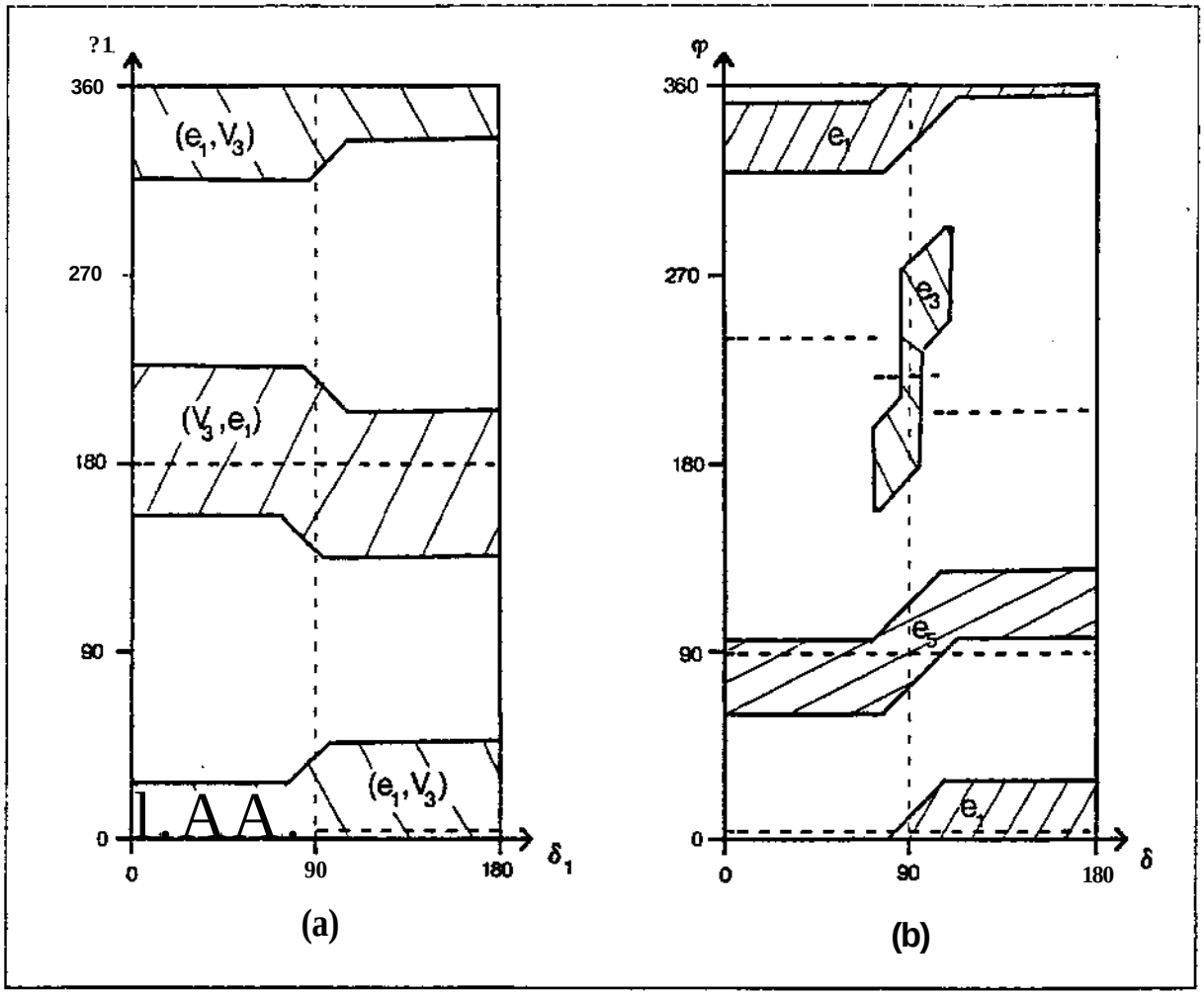


Figure 65. Result of shrinking regions* including aH types of uncertainty, (a) Remaining regions for the triangle's sqnee&c*-grasp diagram, (b} Remaining repions for the roly-pointy's push-grasp diagram. Uncertainty Tallies for these digrams were as follows: Uncertainty hi *fi: p* € (0.15,0.35) for (a), *ft* € (0.2,0.5) for (b); uncertainty 'm $ = ±5°; uncertainty in polygon orientation = ±20[u]; uncertainty i& £ = ±5°.

to have indeterminate stability for values of *6* not close to 00 degrees.

After the diagram has been modified to account for the uncertainty in /*, we can shrink the regions to account for the other sources of uncertainty, as described earlier. Figure 65 shows the result of shrinking the regions of Figure 64 to account for $U\$_t$ U_{ori}, and $U\$_{\%}$ in addition to *Up.* Notice that the regions for the roly-pointy edges *2 ^^ [e]4 have disappeared; this implies that there is too much uncertainty for these grasps to succeed.

Compensation for uncertainty can be accomplished after the pmsh-grasp and sqneeie~ grasp diagrams have beea generated, and we have explained uncertainty handling in thii way* However, it is computationally easier to compensate for uncertainty as the diagrams are being generated- This should be possible is most cate$ because *U^ Us,* and *Up* will all be known for a particular rob#t Dace the polygon has been defined, *U^* might also

be available, if it depends on the angular resolution of a sensor that is determining the polygon's position. Generating these diagrams while simultaneously compensating for uncertainty is not difficult, and will not be explained in detail here.

After all uncertainties have been accounted for (whether they were built-in at the time of diagram generation or added later by shrinking regions), the regions that remain constitute the space of all *guaranteed grasp plans.* This is the space of all grasping moves that are guaranteed to succeed, even in the face of worst-case uncertainty. This guarantee holds as long as we have established an upper bound on the uncertainties present in the problem, and the previously-stated assumptions are met.

VII. Conclusion

Making the Final Decision

In this paper, we have defined a class of three types of grasping operations: squeeze-grasp, offset-grasp, and push-grasp. For each of these operations, we have described a method for computing the space of all grasping moves that are guaranteed to succeed, even if the worst-case error occurs. Choosing an operation from among these guaranteed grasps can then be accomplished by taking into account other, higher-level considerations:

- optimal pushing distance
- avoiding nearby obstacles
- avoiding untouchable surfaces (e.g., wet glue on an edge that will be pressed against another object)
- optimal grasp balance
- maximum grasp firmness

The priorities associated with these factors will vary from situation to situation; the final grasping decision will be deferred to a higher level of processing which takes into account these and other inputs.

Implementation and Experimental Results

The grasp-planning method described in this paper has been implemented in LISP on a Symbolics 3600 Lisp Machine and physically tested using a Puma 560 manipulator. The program takes as input the shape of a polygon and the location of its COF, as well as the lower and upper bounds in \x and the uncertainty bounds *U$*, U_{or}*£,* and *US.* Given these inputs, the program can generate any of the diagrams described in this paper.

The program generates push-stability and offset-grasp diagrams very quickly, even though no serious effort was made to optimize the code. Push-grasp and offset-grasp diagrams are calculated almost instantaneously, while squeeze-grasp diagrams take one or two seconds to compute. The squeeze-grasp diagram takes extra time because the code that intersects the plane #1 and plane #2 diagrams runs in $O(n^2)$ time, while the push-grasp and offset-grasp diagrams are generated in $O(n)$ time (where n is the number *o{* polygon edges).

The implementation described above was used to informally test the validity of the planning method. A variety of planar objects were measured and their shapes were input into the grasp-planning system. Pushing and squeezing operations were tested by generating the push-stability or squcese-grasp diagram for an object^ selecting a particular

operation from the diagram, and then performing the selected operation with the Puma to verify that the expected result was actually achieved. Moves were selected from throughout the diagram, both well within regions and near region boundaries. Applying various pushing and squeezing operations consistently resulted in achieving the correct final configuration, even for extreme operations whose predicted results seemed counter-intuitive.

Future Research

While this research adequately solves a restricted class of grasping problems, it is limited in several ways. Extending this grasp-planner to overcome these limitations provides several areas for future research. Some likely extensions to the current grasp-planning method are:

- Remove the infinite half-plane assumption to directly account for fingers of finite length. This would allow the planner to capitalize on concavities in the polygon, and pick up large objects that extend past the ends of the gripper fingers.
- Presently the method assumes perfect knowledge of the shape of the object and the location of its COF. In many situations, however, this information will only be known approximately. Therefore, incorporating the uncertainties $U_s\wedge_{apc}$ and $UcoF^{m}$ *he analysis would be useful.
- While many grasping tasks meet our two-dimensional planar motion abstraction> a full three-dimensional extension of this analysis would make the planning method outlined in this paper applicable to a broader range of tasks.

Research investigating these areas is currently in progress at the CMU Manipulation Lab.

Acknowledgements

I would like to thank my advisor, Matt Mason, first and foremost for spending generous amounts of his time teaching me to think like a computer scientist. I would also like to thank Matt for introducing me to the problem of object stability under pushing, and his many ideas and insights along the way to its solution.

I would also like to thank Ken Goldberg, Michael Peshkin, Harry Printz, and Rick Szeliski for reading drafts of this paper and lending many helpful comments and suggestions. Yu Wang's help in building gripper fingers and other lab apparatus is greatly appreciated.

Finally, I would like to thank my wife, MaryAnn, and all the members of my family for their support and love that has helped me in countless ways.

This research was conducted at the Computer Science Department of Carnegie-Mellon University and was supported by a grant from the System Development Foundation. The author is supported by a fellowship from the National Science Foundation.

References

Asada, H., and By, A. B., "Kinematics of Workpart Fixturing," Proceedings, IEEE Conference on Robotics and Automation, pp. 337-345, St. Louis, MO, March 1985.

Baker, B. S., S. Fortune, and E. Grosse, "Stable Prehension with a Multi-Fingered Hand," Proceedings, IEEE Conference on Robotics and Automation, pp. 570-575, St. Louis, MO, March 1985.

Bolles, R., and Paul, R., "The Use of Sensory Feedback in a Programmable Assembly System," Stanford Artificial Intelligence Laboratory Memo AIM-220, October 1973.

Brooks, R. A., "Symbolic Error Analysis and Robot Planning," *International Journal of Robotics Research,* vol. 1, no. 4, Winter 1982.

Brou, P., "Implementation of High-Level Commands for Robots," M.S. Thesis, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, February 1981.

Cutkosky, M. R., "Grasping and Fine Manipulation for Automated Manufacturing," Ph.D. Thesis, The Robotics Institute, Carnegie-Mellon University, January 1985.

Erdmann, M. A., "On Motion Planning with Uncertainty,"- M.S. Thesis, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, AI-TR-810, August 1984.

Erdmann, M, A., "Using Backprojections for Fine Motion Planning with Uncertainty," Proceedings, IEEE Conference on Robotics and Automation, pp. 549-554, St. Louis, MO, March 1985. *

Fearing, R. S., "Simplified Grasping with Dextrous Robot Hands," Proceedings, American Control Conference, San Diego, CA, June 1984.

Fearing, R. S., "Touch Processing for Determining a Stable Grasp," M.S. Thesis, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, September 1983.

Grossman, D. D., and Blasgen, M. W. 1975. "Orienting Mechanical Parts by Computer-Controlled Manipulator." *IEEE Transactions on Systems, Man, and Cybernetics* SMC-5:5.

Hanafusa, EL, and Asada, H.3 "A Robot Hand with Elastic Fingers and its Application to Assembly Process," IFAC Symposium on Information and Control Problems in Manufacturing Technology, Tokyo, 1977,

Holzxnann, W., and McCarthy, J. M., "Computing the Friction Forces Associated With a Three-Fingered Grasp,[9] Proceedings, IEEE Conference on Robotics and Automation, pp. 594-600, St. Louis, MO, March 1985.

Inoue, H., "Force Feedback in Precise Assembly Tasks," Artificial Intelligence Laboratory, Massachusetts Institute of Technology, AIM-308, August, 1974 (Reprinted in Winston, P. H. and R. H. Brown **(eds),** *Artificial Intelligence: An MIT Perspective,* MIT Press, 1979).

Lozano-Pérez, T., "Automatic Planning of Manipulator Transfer Movements," *IEEE Transactions on Systems, Man, Cybernetics,* vol. SMC-11, no. 10, 1981 (Reprinted in Brady, M. et. al. (eds), *Robot Motion*, MIT Press, 1983).

Lozano-Pérez, T., "The Design of a Mechanical Assembly System," M.S. Thesis, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, AI-TR-397, December 1976.

Lozano-Pérez, T., M. Mason, and R. Taylor, "Automatic Synthesis of Fine-Motion Strategies for Robots," *International Journal of Robotics Research,* vol. 3, no. 1, Spring 1984.

Lozano-Pérez, T., and Wesley, M. A., "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," *Communications of the ACM,* vol. 22, no. 10, October 1979.

Mani, M., and Wilson, R. D. W., "A Programmable Orienting System for Flat Parts," Proceedings, NAMRI XIII, Berkeley, CA, May 1985.

Mason, M. T., "Automatic Planning of Fine Motions: Correctness and Completeness," Proceedings, IEEE Computer Society International Conference on Robotics, Atlanta, GA, March, 1984 (also Carnegie-Mellon University Robotics Institute Technical Report CMU-RI-TR-83-18, December 1983).

Mason, M. T., "Mechanics of Pushing," Proceedings, Second International Symposium of Robotics Research, Kyoto, Japan, August 1984.

Mason, M. T., "Manipulator Grasping and Pushing Operations," Ph.D. Thesis, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, AI-TR-690, June 1982.

Paul, R., "Modelling, Trajectory Calculation and Servoing of a Computer Controlled Arm," Ph.D. Thesis, Stanford Artificial Intelligence Laboratory Memo AIM-177, November 1972.

Peshkin, M. A., and Sanderson, A. C., "Reachable Grasps on a Polygon: The Convex Rope Algorithm", The Robotics Institute, Carnegie-Mellon University, unpublished paper, January 1985.

Salisbury, J. K., "Articulated Hands: Force Control and Kinematic Issues," *International Journal of Robotics Research*, vol. 1, no. 1, Spring 1982.

Salisbury, J. K., and Roth, B., " Kinematic and Force Analysis of Articulated Mechanical Hands," *Journal of Mechanisms, Transmission, and Automation in Design,* vol. 105:35-41, March 1983.

Wolter, J. D., R. A. Voltz, and A. C. Woo, "Automatic Generation of Gripping Positions", Center for Robotics and Integrated Manufacturing, University of Michigan, February 1984.