Stability of Assemblies

Raju Mattikalli
Pradeep Khosla
Bruno Repetto
David Baraff

Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

High level assembly plans prescribe the sequence in which parts come together along with their motions. At each stage of assembly the stability of subassemblies is an important concern. In this paper we address the problem of gravitational stability of assemblies of frictionless rigid bodies. Solution methods to the problems of determining if an assembly is stable, and finding a stable orientation for a given assembly are proposed. The solution methods for the two problems compute constrained motions of individual parts that decrease gravitational potential energy, in order to determine stability and search for stable orientations. Linear programming is used to obtain numerical solutions. The problem of finding a stable orientation is formulated as a maximin problem. The solution to this problem is the first general method for automatically determining stable orientations. The program to solve the two problems takes as input a geometric model of the assembly and information related to grounding constraints. Preliminary results are presented.

1 Introduction

In recent years, there has been much interest in automated assembly planning. High-level assembly plans prescribe the order in which parts come together as well as the motions that bring them together. Automated *disassembly* of computer models of assemblies is a good method for obtaining assembly plans [7][9][12]. At each stage of assembly, the stability of subassemblies is an important concern. An assembly of parts is said to be stable if the parts are in static equilibrium under the influence of external and internal forces. External forces could be in the form of field forces, for example gravitational forces, or in the form of contact forces, for example gripper forces. Internal forces between parts arise from their mutual contact. Magnetic forces between parts may also produce internal forces in an assembly.

Consider the assembly of parts shown in Figure 1. Components C_2 and C_3 are identical and are placed within the right angled component C_1. For the moment, let us ignore frictional forces. C_1 is firmly attached to a gripper, as indicated in the figure by the ground symbol. The vector ***g*** indicates the direction of a uniform gravity field. In the orientation shown in Figure 1(a), the assembly is unstable under the influence of gravity. Gravitational forces produce a downward translational motion on component C_2 which causes component C_3 to translate towards the right. Is there an orientation of the assembly in which it is gravitationally stable? The orientation shown in Figure 1(b) is a stable one. The gravitational force on component C_2 is balanced by the internal contact forces exerted by the other two components. The same is true for component C_3.

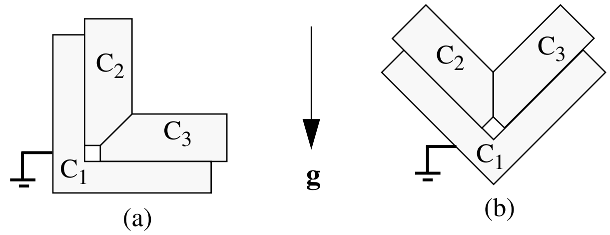

Figure 1: (a) An assembly that is gravitationally unstable. (b) The assembly in an orientation that makes it gravitationally stable.

In this paper, we address the problem of the stability of a collection of contacting objects, acted upon by a uniform external gravity field. One (or more) of the objects is assumed to be fixed in a gripper. All the objects are assumed to be rigid and frictionless. Under these assumptions, we solve the following problems:

Problem 1: (Stability) Given an assembly in a given orientation, is the assembly stable?

Problem 2: (Stable Orientation) Given an assembly, determine an orientation (if one exists) that makes the assembly stable under the gravity field.

In the first problem, if we find that the assembly is unstable, we would also like to know which parts in the assembly will have motion and in what direction. This can be used either to redesign parts or provide fixtures during assembly.

This research was supported in part by the Engineering Design Research Center, an NSF Engineering Research Center at Carnegie Mellon University

In the second problem, if no stable orientation exists, we would like to find an orientation that is closest to being stable along with the motion directions of the unstable parts for that orientation. A measure of 'closeness' is defined in section 2.

1.1 Previous Work

Palmer[8] obtained computational complexity results for stability problems in which assemblies consist of rigid planar polygons. If a set of interaction forces exists that results in static equilibrium for an assembly, then the assembly is said to be *potentially stable.* The complement of potential stability is *guaranteed instability*, in which an assembly is unstable for all valid interaction forces. If there exist no legal infinitesimal motions for which the assembly is unstable, then it is said to be *infinitesimally stable*. This type of stability is a conservative approximation to the stability problem and is a stronger form of stability than potential stability. Palmer showed that the potential stability problem is in *P*, while the infinitesimal stability problem is *NP*-hard. The complexity of the infinitesimal stability problem is solely a consequence of ill-defined surface normals at contact points, for example, vertex-to-vertex contact. Following Baraff[1], our method disallows degenerate contact geometry by imagining a local extension of the contact surface in order to generate a well-defined surface normal.

One approach to determine stability is to model the interaction between parts in an assembly in terms of unknown forces. Potential stability for systems with friction can be determined by attempting to find a set of forces that produces equilibrium. Blum, Griffith, and Neumann[3] implemented a potential stability test based on linear programming that searches for such a solution. The analysis begins by assuming that the assembly in question is stable. For an assembly in equilibrium, one can generate a set of force balance equations in terms of unknown interaction forces between parts and known gravitational forces. If forces that simultaneously satisfy all the force balance equations cannot be found, then the assembly is declared unstable. However, no information is obtained about the impending motion of unstable parts.

Boneschanscher *et al*.[11] tried to determine which parts are unstable (but not their impending motion) by solving for the stability of each part individually. In their work, the assembly is assumed to be sitting on a table with external forces being applied to it. Conditions on the forces between individual parts and the table are written as systems of inequalities. By transforming the contact graph, a larger system of inequalities is formed. The emerging system is solved using linear programming. If any subassembly during this process is found to be unstable, the entire assembly is declared unstable. However, as Boneschanscher *et al*. point out, their technique will not work if there are loops in the contact graph. This is a significant shortcoming since loops occur quite frequently in assemblies. In section 2.2 we shall compare our formulation with that of Boneschanscher *et al.*'s and argue that their inability to handle loops is a fundamental problem of the method.

Baraff[1] presents a method to analytically calculate forces between systems of frictionless rigid bodies in resting contact. At each point of contact, an unknown repulsive normal force exists. The normal forces must be strong enough to prevent interpenetration while at the same time being workless. This implies that the normal force must be zero wherever contact is broken. The normal forces can be found by solving a quadratic program. Once the normal forces have been computed, the ensuing acceleration is given by Newton's first law. Thus, stability can be determined by checking if any parts have nonzero acceleration. Additionally, the impending motion is indicated by the acceleration. This method differs from Boneschanscher *et al.*'s in that it can be used to solve the infinitesimal instability problem for frictionless systems with loops.

1.2 Approach Outline

Our method to solve the stability problem follows similar lines to Baraff[1]. However, instead of taking a vectorial approach by solving for forces, we adopt the Lagrangian approach and solve for part motions. We formulate an expression for the change in potential energy of a system of bodies subject to a given virtual displacement. Given a set of all legal virtual displacements (that is, those which do not violate non-interpenetration constraints), the displacement which results in the largest decrease in potential energy is selected. If no legal virtual displacement decreases the potential energy of the system, then the configuration is stable. Although this approach offers no advance over Baraff's in determining stability, it is extensible to the problem of finding the most stable orientation.

Consider a system of frictionless bodies in contact with each other such as that shown in Figure 1(a). The system is at rest and is placed within a gravitational field. One (or more) of the bodies are fixed. A valid configuration is one in which no two objects intersect each other. Assume that the initial configuration of the system is valid, and let the system be displaced (in a virtual manner) from this initial configuration. Given the contact geometry, constraints can be placed on the displacement to prevent the system from moving into an invalid configuration. The system is conservative, since only gravitational forces perform work for any displacement consistent with the contact constraints.

If the system is initially at rest, the kinetic energy T of the system is zero. If the system undergoes a motion, then T must increase, which means that the potential energy U of the system must *decrease*. Thus, if there is no legal displacement which decreases U, the system cannot undergo any motion. Conversely, if there exists a virtual displacement for which U does decrease, then the system is unstable and is guaranteed to begin moving.

Let $\delta \boldsymbol{r}_i$ denote a virtual translation of the ith part and let

δU denote the virtual change in U caused by this displacement. If the ith part has mass m_i, then

$$\delta U = -\sum_{i=1}^{N} m_i \boldsymbol{g} \cdot \delta \boldsymbol{r}_i . \tag{1}$$

If all legal virtual displacements produce nonnegative values of δU, then it can be concluded that the system is guaranteed to be stable. Our approach will be to minimize δU over the set of legal virtual displacements. If the minimum value of δU for a given orientation is nonnegative, then it can be concluded that the system is stable in that orientation; otherwise the system is unstable. Section 2 describes the basic formulations employed. Numerical solution methods for determining stability and finding a stable orientation (if one exists) are presented in Section 3.

2 Formulation

The mathematical formulation of the stability problem is as follows. Let $\delta \boldsymbol{p}_i = (\delta \boldsymbol{r}_i, \delta\theta_i)$ represent a virtual displacement of the ith body in the system, where $\delta \boldsymbol{r}_i, \delta\theta_i \in \mathbf{R}^3$. The vector $\delta \boldsymbol{r}_i$ denotes a translational displacement of the ith part, while $\delta\theta_i$ denotes a rotation of magnitude $\|\delta\theta_i\|$ of the body around its center of mass. The axis of the rotation is along the $\delta\theta_i$ direction. For an assembly of N bodies, given $\boldsymbol{g}$, we want to determine N vectors $\delta \boldsymbol{p}_i$ such that (1) the vectors correspond to a motion that minimizes the change δU in potential energy of the system, and (2) the bodies do not penetrate each other. Since only the direction of the displacement (and not its magnitude) matters, we will set an upper bound on the magnitude of the vectors $\delta \boldsymbol{p}_i$.

2.1 Non-Interpenetration Constraints

Consider two bodies m and n within an assembly making contact over the segment ab, as shown in Figure 2. If body m undergoes a displacement $\delta \boldsymbol{p}_m$, then point a, as attached

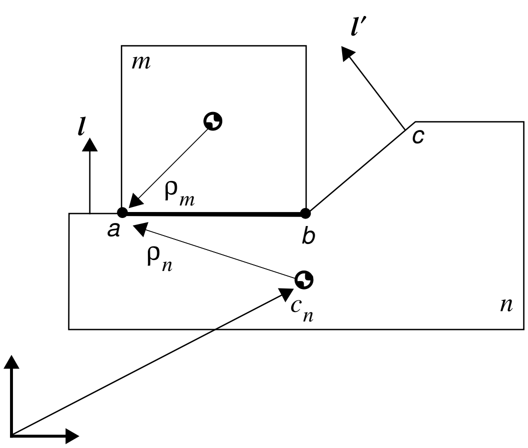

Figure 2: Two bodies m and n within an assembly. The displacements of point a and b on bodies m and n are used in formulating the nonpenetration constraints

to body m undergoes a particular displacement $\delta \boldsymbol{a}_m$. Similarly, a displacement $\delta \boldsymbol{p}_n$ of body n causes a separate displacement $\delta \boldsymbol{a}_n$ of point a, as attached to body n. To prevent interpenetration from occurring, the relative displacement $\delta \boldsymbol{a}_m - \delta \boldsymbol{a}_n$ cannot have any component opposite the direction $\boldsymbol{l}$. Similarly, we must require that the relative displacement at b also have no component opposite $\boldsymbol{l}$. However, to prevent interpenetration from occurring along the segment bc, we must additionally constrain the relative displacement at b to have no component opposite the direction $\boldsymbol{l}'$. This gives rise to a total of three constraints.

Let c_m denote the position of the center of mass of body m in a global frame of reference, and let ρ_m be the displacement from c_m to point a. Let c_n and ρ_n be defined similarly. If body m undergoes a displacement $\delta \boldsymbol{p}_m = (\delta \boldsymbol{r}_m, \delta\theta_m)$, then the displacement $\delta \boldsymbol{a}_m$ of a on body m is

$$\delta \boldsymbol{a}_m = \delta \boldsymbol{r}_m + (\delta\theta_m \times \rho_m) . \tag{2}$$

Similarly, the displacement of a on body n is

$$\delta \boldsymbol{a}_n = \delta \boldsymbol{r}_n + (\delta\theta_n \times \rho_n) . \tag{3}$$

To prevent interpenetration, we require

$$(\delta \boldsymbol{a}_m - \delta \boldsymbol{a}_n) \cdot \boldsymbol{l} \geq 0 . \tag{4}$$

Substituting equations (2) and (3) into (4) we obtain

$$((\delta \boldsymbol{r}_m + \delta\theta_m \times \rho_m) - (\delta \boldsymbol{r}_n + \delta\theta_n \times \rho_n)) \cdot \boldsymbol{l} \geq 0 . \tag{5}$$

To prevent interpenetration at b in the $\boldsymbol{l}$ direction, we would add the constraint

$$(\delta \boldsymbol{b}_m - \delta \boldsymbol{b}_n) \cdot \boldsymbol{l} \geq 0 \tag{6}$$

where $\delta \boldsymbol{b}$ was the displacement of point b (although equation (6) would be expressed in terms of m and n's displacements, as in equation (5)). Similarly, to prevent interpenetration in the $\boldsymbol{l}'$ direction, we would add the constraint

$$(\delta \boldsymbol{b}_m - \delta \boldsymbol{b}_n) \cdot \boldsymbol{l}' \geq 0 \tag{7}$$

(again, in the form of equation (5)).

The constraint equation at a contact point is trivially modified if one of the bodies is grounded. (If both bodies are grounded, contact between them can clearly be ignored.) For example, if body n was grounded, equation (5) would be

$$\delta \boldsymbol{r}_m + (\delta\theta_m \times \rho_m) \cdot \boldsymbol{l} \geq 0 \tag{8}$$

with similar modifications for equations (6) and (7).

Note that it is not necessary to generate a constraint for *every* contact point between two bodies. For example, if we prevent points a and b of body m from interpenetrating body n, we prevent interpenetration at any point along the segment ab (for configurations near the present one). Thus constraints at the end points are sufficient to prevent interpenetration along the entire line segment. Similarly, the

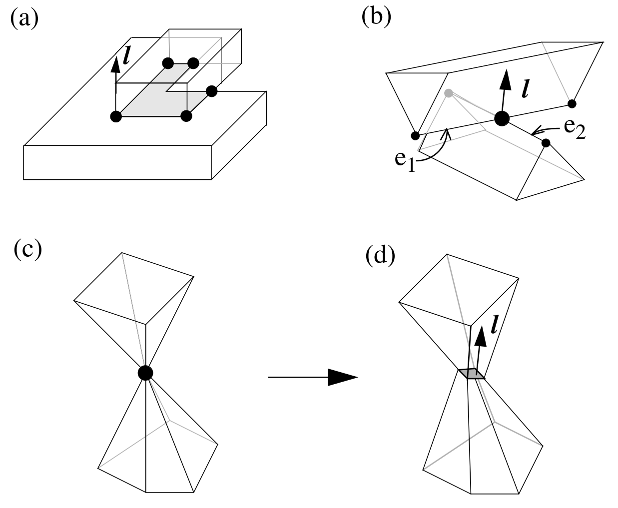

Figure 3: Contact between bodies. (a) A polygonal surface of contact. (b) Edge contact. (c) Degenerate vertex-vertex contact. (d) Degenerate contact resolved by planar approximation.

non-interpenetration constraint for the polygonal contact of Figure 3 is written in terms of 5 constraint inequalities: one for each vertex of the convex hull of the polygon. The constraint inequality for each vertex is a function of the unknown displacements of the two bodies, the position of the vertex, and the normal vector $\boldsymbol{l}$. The position of the vertex and the vector $\boldsymbol{l}$ are obtained from the geometric model of the assembly. Figure 3 shows the normal vectors for the different kinds of contact possible. Figure 3(b) shows two edges making contact at a vertex. In this case, the normal vector $\boldsymbol{l}$ is well-defined and is perpendicular to the two edges as shown. Figure 3(c) shows a degenerate vertex-vertex contact. As stated in Section 1.1, this contact situation makes the stability problem *NP*-hard. Being an uncommon situation, we shall assume that a surface of contact with nonvanishing area exists. The orientation of this surface is selected so that the normal direction $\boldsymbol{l}$ lies within the solid, as shown in Figure 3(d).

To obtain constraints for more complex contact geometry, Mattikalli and Khosla[10] propose a geometric representation of motion constraints. This method can be used to generate constraints for curved line segments of contact.

2.2 Determining Stability

As stated in section 1.2, a system at rest in a gravitational field will move so as to decrease its potential energy as much as possible, subject to the system's motion constraints. In Boneschanscher *et al.*[11], the stability of an assembly was determined one part at a time. It was assumed that the motion constraints on each part could be separated and satisfied independently. However, this assumption breaks down whenever a closed loop is present in an assembly. When loops are present, the equations of motion of parts are coupled, and cannot be readily separated. Instead, motions must be solved for simultaneously. Our method deals with loops by using linear programming to search for a motion that decreases potential energy while simultaneously satisfying *all* the motion-constraint equations formulated in section 2.1. Another advantage of our method is in terms of computational effort. For an assembly of m parts, with n vertices of contact, Boneschanscher *et al.*'s algorithm requires an average of $O(mn^3)$ steps to determine the stability of an assembly. Using linear programming, our method takes an average of $O(mn^2)$ steps. In section 4 some results obtained by applying our method to assemblies with loops are presented.

To determine stability, we need know if their exists a set of displacement vectors $\delta \boldsymbol{p}_i$ that yields $\delta U < 0$ and is consistent with the motion constraints. Equivalently, we wish to minimize δU over the set of legal displacements. However, if $\delta \boldsymbol{p}_i$ is a vector of legal displacements, then $\alpha(\delta \boldsymbol{p}_i)$ is also a vector of legal displacements, for any positive scalar α. It is therefore necessary to bound the magnitudes of the displacements $\delta \boldsymbol{p}_i$ considered by the linear program.

The formulation to determine if an assembly in a given orientation is stable is the following. Given an assembly of N bodies and a gravity vector $\boldsymbol{g}$, we determine stability by solving the linear program

$$\begin{aligned} &\text{minimize} \quad z = -\sum_{i=1}^{N} m_i \boldsymbol{g} \cdot \delta \boldsymbol{r}_i \\ &\text{subject to} \quad \|\delta \boldsymbol{r}_i\|_\infty \le 1,\ \|\delta\theta_i\|_\infty \le 1 \qquad 1 \le i \le N, \\ &\qquad \delta \boldsymbol{r}_k = 0 \text{ and } \delta\theta_k = 0 \qquad \forall k \in G, \\ &\qquad (\delta \boldsymbol{r}_i - \delta \boldsymbol{r}_j) + (\delta\theta_i \times \rho_i^p - \delta\theta_j \times \rho_j^p) \cdot \boldsymbol{l} \ge 0, \\ &\qquad\qquad \forall (p, i, j, \boldsymbol{l}) \in P \end{aligned} \tag{9}$$

The infinity norm $\|v\|_\infty$ of a vector v is the maximum absolute value over all the components of v. Thus, the condition $\|\delta \boldsymbol{r}_i\|_\infty \le 1$ requires that all components of $\delta \boldsymbol{r}_i$ have magnitude less than one, and similarly for $\delta\theta_i$. The index set G indicates all grounded bodies.The set P contains triples $(p, i, j, \boldsymbol{l})$, denoting a contact between bodies i and j at point p with surface normal $\boldsymbol{l}$. The vector ρ_i^p is the displacement from the center of mass of body i to p, and similarly for ρ_j^p. Let $\bar{z}$ be the minimum z satisfying the conditions of equation (9). Note that $\bar{z}$ can never be greater than zero since $\delta \boldsymbol{r}_i = \delta\theta_i = 0$ for each part is a legal motion and yields $z = 0$. If $\bar{z} = 0$ then U cannot be decreased and the system is stable. Otherwise, $\bar{z} < 0$ so U can be decreased and the system is unstable. (Note that the displacements which minimize z only approximately indicate the impending motion. Computation of the exact impending motion requires a method such as that used in Baraff[1].) Solutions to some test cases using the above linear program are shown in Figure 8 and Figure 9.

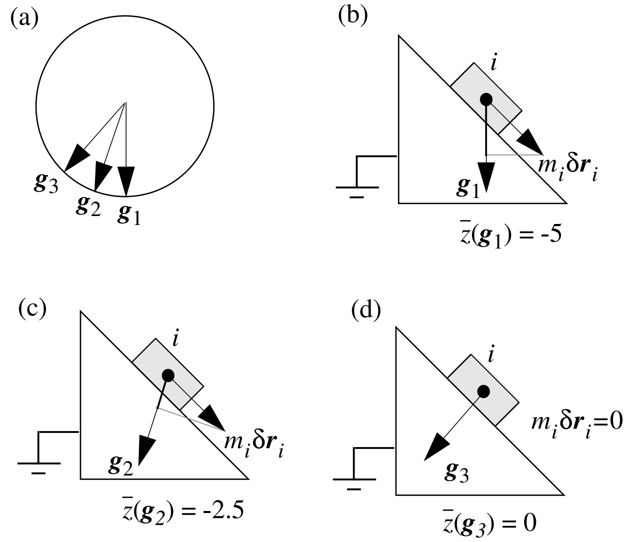

Figure 4: An assembly in three different orientations with respect to gravity. The projection $m_i\delta \boldsymbol{r}_i \cdot \boldsymbol{g}$ indicates the closeness of the orientation to a stable one. (d) shows a stable orientation.

2.3 Finding a Stable Orientation

This section describes an extension to the minimization of the previous section that enables us to find a stable orientation for an assembly or show that no such orientation exists. Rather than reorienting all the parts of the assembly, we will imagine changing the direction of the gravity vector $\boldsymbol{g}$ in an attempt to make an assembly stable. Henceforth then, $\boldsymbol{g}$ is regarded as a variable, and not as a constant. Consider the assembly shown in Figure 4. For a given choice of $\boldsymbol{g}$, let $\bar{z}(\boldsymbol{g})$ be the optimal value of z in equation (9) for that particular choice of $\boldsymbol{g}$. Figure 4 shows how $\bar{z}(\boldsymbol{g})$ varies as we change the gravity direction $\boldsymbol{g}$. When the gravity direction is changed from $\boldsymbol{g}_1$ to $\boldsymbol{g}_2$, $\bar{z}(\boldsymbol{g})$ increases from a value of -5 to -2.5. When $\boldsymbol{g}$ is changed to $\boldsymbol{g}_3$, $\bar{z}(\boldsymbol{g}_3)$ is zero, which means the assembly is stable with the choice $\boldsymbol{g} = \boldsymbol{g}_3$.

In general, since $\bar{z}(\boldsymbol{g})$ is bounded above by zero for any assembly, we can search for a stable orientation of the assembly by trying to find a value for $\boldsymbol{g}$ that maximizes $\bar{z}(\boldsymbol{g})$. If we can find $\boldsymbol{g}$ such that $\bar{z}(\boldsymbol{g}) = 0$, then we have found a gravity direction which makes the assembly stable (or, equivalently, we have found a way to reorient all the parts to induce stability in a vertical gravity field). However, if $\bar{z}(\boldsymbol{g}) < 0$ for all $\boldsymbol{g}$ then the assembly cannot be made stable. If we regard $\bar{z}(\boldsymbol{g})$ as a metric for instability, then the direction $\boldsymbol{g}$ which maximizes $\bar{z}(\boldsymbol{g})$ is the "most" stable orientation with respect to this metric.

Let us phrase the above discussion as an optimization problem. Given an assembly, we are searching for a value of g that maximizes the function $\bar{z}(\boldsymbol{g})$, where $\bar{z}(\boldsymbol{g})$ is itself defined as the minimum value of z in equation (9) for a particular choice of $\boldsymbol{g}$. Thus, maximizing $\bar{z}(\boldsymbol{g})$ results in a *maximin* problem.

Minimax problems with constraints are in general very difficult to solve, and have an extensive literature[5]. In the next section, we present an iterative solution method for maximizing $\bar{z}(\boldsymbol{g})$ by solving a series a linear programs.[1]

3 Numerical Solution Method

Let us rephrase the maximin problem more formally. Let $\delta\boldsymbol{p}$ denote a vector of N displacements of the form $\delta\boldsymbol{p}_i = (\delta\boldsymbol{r}_i, \delta\theta_i)$ for $1 \le i \le N$. The constraints in equation (9) on the displacements limit allowable displacements $\delta\boldsymbol{p}$ to some feasible region, which we shall denote as the set A. Note that A is a convex domain since it is the conjunction of some number of linear inequalities. In considering values for $\boldsymbol{g}$, we are concerned only with direction, and not with magnitude. If we constrained $\boldsymbol{g}$ to be a unit vector, we would have to perform a maximization over a nonconvex and nonlinear domain. Instead of maximizing over the unit sphere of directions, we will maximize over vectors that lie on the unit "diamond." The vectors we consider will be of the form $\boldsymbol{g} = (g_x, g_y, g_z)$ such that $|g_x| + |g_y| + |g_z| = 1$. This is still a nonconvex set but we can partition the "diamond" into eight facets by restricting the signs of the components of $\boldsymbol{g}$. Thus, let S_1 be the set of vectors $\boldsymbol{g}$ such that g_x, g_y, and g_z are all greater than zero and $g_x + g_y + g_z = 1$. Let S_2 be the same as S_1 except that g_z is less than zero and $g_x + g_y - g_z = 1$. The sets S_3 through S_8 are defined to cover all the remaining sign combinations on g_x, g_y, and g_z.(For planar configurations, we would need only four partitions.) We will let S denote the union of the eight partitions. When we consider vectors $\boldsymbol{g}$ in S, we first consider vectors in S_1, then S_2 and so on, to avoid dealing with nonconvex search domains. Using this notation, we can phrase our problem as

$$\max_{\boldsymbol{g} \in S_i} \bar{z}(\boldsymbol{g}) = \max_{\boldsymbol{g} \in S_i} \left[\min_{\delta p \in A} z(\boldsymbol{g}, \delta p) \right] \tag{10}$$

where $1 \le i \le 8$ and z is regarded as a function of $\boldsymbol{g}$ and $\delta\boldsymbol{p}$.

Let us consider an iterative solution method to solve this maximin problem. A first approach to solving (10) might be to guess an initial value of $\boldsymbol{g}$ in S_i, solve the minimization with $\boldsymbol{g}$ fixed and obtain $\delta\boldsymbol{p}$. Using this $\delta\boldsymbol{p}$ as a constant and $\boldsymbol{g}$ as a variable, we could then solve the maximization problem. However, this method is flawed: if $\delta\boldsymbol{p}$ is held constant and the maximization problem with $\boldsymbol{g}$ as a variable, we might obtain a value for $\boldsymbol{g}$ that does not produce the same $\delta\boldsymbol{p}$ if we minimize with respect to that $\boldsymbol{g}$. We can correct this as follows. For a given value of $\boldsymbol{g}$, there exists a unique value of $\delta\boldsymbol{p}$. In Figure 5, the domain S (which we temporarily imagine as the unit sphere for simplicity) is parti-

1. Following the acceptance of this paper, we discovered a transformation of our maximin formulation that admits a simpler numerical solution method. There was insufficient time to include those results in this paper, but details can be found in Baraff *et al.*[2].

tioned into sets, with members in a given set producing the same displacement $\delta \boldsymbol{p}$ in the inner minimization.

To remedy our solution method, the outer maximization must be performed to find a direction of $\boldsymbol{g}$ *within a given partition*. This can be done by imposing constraints on the outer maximization.

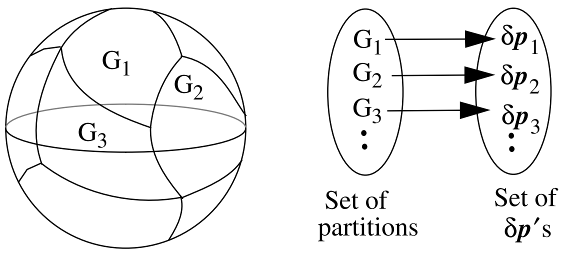

Figure 5: The domain of $\boldsymbol{g}$ can be partitioned into sets such that all elements in the set produce the same $\delta \boldsymbol{p}$ for the maximization in equation (10).

3.1 Restricting $\boldsymbol{g}$ to within a Partition

Consider the $6N$-dimensional configuration space (Figure 6) corresponding to the unknown displacements in formulation (8). The shaded region represents the feasible set of displacements A. The objective function $z(\boldsymbol{g}_1, \delta \boldsymbol{p})$ is a linear function of the unknown $\delta \boldsymbol{p}$ where $\boldsymbol{g}$ is fixed at $\boldsymbol{g}_1$, and defines a family of parallel hyperplanes. The direction of the normal of the hyperplanes is determined by the value of $\boldsymbol{g}_1$. In Figure 6, the objective function $z(\boldsymbol{g}_1, \delta \boldsymbol{p})$ is shown for different values of $\delta \boldsymbol{p}$: $z_1 = z(\boldsymbol{g}_1, \delta \boldsymbol{p}')$, $z_2 = z(\boldsymbol{g}_1, \delta \boldsymbol{p}'')$, and $z_3 = z(\boldsymbol{g}_1, \delta \boldsymbol{p}''')$. The function z achieves its maximum value z_3 in the shaded region when $\delta \boldsymbol{p} = v_1$. Let line z_3 be pivoted at vertex v_1 and its slope changed. It can be seen that the slope of line z_3 can be varied to lie between the slopes of lines l_1 and l_2 without changing the optimality of v_1. This suggests that the value of $\boldsymbol{g}$ (which determines the slope of line z_3) can be varied so as to maintain the same optimal point. The range of $\boldsymbol{g}$ for which the optimality is maintained defines the partition G_1 (assuming $\boldsymbol{g}_1$ lies within G_1). Given the optimal point $\delta \boldsymbol{p} = v_1$, we can write down conditions that tell us how much $\boldsymbol{g}$ can be varied from $\boldsymbol{g}_1$ without changing the fact that $\delta \boldsymbol{p} = v_1$ is optimal.

The simplex method for linear programming finds what are known as *basic* optimal solutions. One of the key features of a basic optimal solution is that it indicates an invertible matrix B, a matrix N, and a row vectors c^B and c^N. (The elements of B and N come from the coefficients of the motion-constraint inequalities of the problem, while the elements of c^B and c^N depend upon part masses and the gravity direction.) These matrices and vectors satisfy the condition

$$c^B B^{-1} N - c^N \leq 0. \tag{11}$$

Furthermore, any change in the gravity direction which maintains this condition results in the same basic optimal solution $\delta \boldsymbol{p} = v_1$[4]. Thus, equation (11) defines a convex region in the space of the $\boldsymbol{g}$ variables (that contains $\boldsymbol{g}_1$) for which the value of g may be changed without changing the optimality of $\delta \boldsymbol{p} = v_1$. When equation (11) is added to the list of constraints for the maximization in (10), $\boldsymbol{g}$ is restricted to lie within a partition.

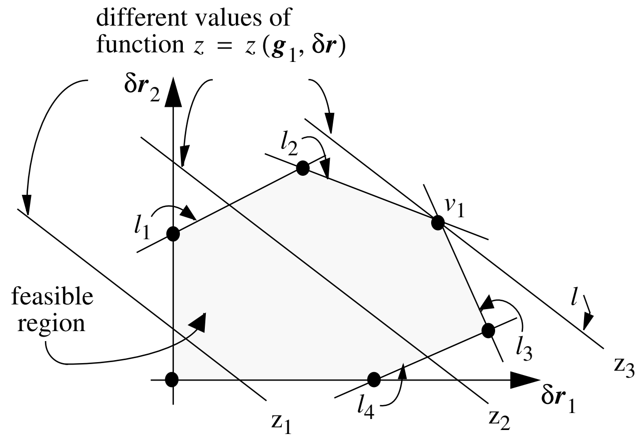

Figure 6: The feasible region and the objective function of an LP problem.

The numerical solution method is thus as follows. We start at an arbitrary point $\boldsymbol{g}_a$ (Figure 7) in the domain of $\boldsymbol{g}$ (again, with the domain of $\boldsymbol{g}$ drawn as a unit sphere rather than as the linearized domain S.) We then attempt to move to a point $\boldsymbol{g}_b$ that is "more" stable within the given partition. If this point $\boldsymbol{g}_b$ is on the boundary of the partition, we must continue the search for a minimum by stepping into the adjacent partition. The property that each partition is convex is used to step into the adjacent partition. The convexity of each partition can be inferred from equation (11) which defines the partition as a union of half spaces (which is always convex). As shown in Figure 7, the directed arc from $\boldsymbol{g}_a$ to $\boldsymbol{g}_b$ can be used to take an epsilon step into the adjacent partition. The new value $\boldsymbol{g}_c$ is now used as an initial guess, a new optimal value $\delta \boldsymbol{p}$ is obtained, constraints for the new partition G_2 are formulated, and the method repeats. If at any stage the optimal solution $\delta \boldsymbol{p}$ for a given $\boldsymbol{g}$ lies in the interior of the partition, the process is terminated. Also, if a point on the boundary of a partition is revisited, then the process is terminated. In either case, we have found a stationery point for the maximin problem. As we show in the next section, the objective function $\bar{z}(\boldsymbol{g})$ is concave, so that the stationary point found is in fact an optimal solution.

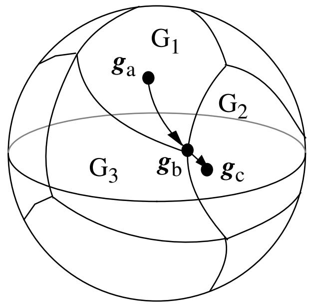

Figure 7: Searching for a $\boldsymbol{g}$ that maximizes $\bar{z}$. From the initial point $\boldsymbol{g}_a$, maximization over the partition G_1 produces $\boldsymbol{g}_b$. The directed arc from $\boldsymbol{g}_a$ to $\boldsymbol{g}_b$ is used to take an epsilon step into the adjacent partition to $\boldsymbol{g}_c$.

3.2 Convexity of the Maximin Problem

The domain A for legal displacements is convex, because it is a conjunction of linear inequalities. The domain S for gravity directions is nonconvex, but is the union of convex partitions S_1 through S_8. As stated earlier, we will run our iterative procedure on each of these partitions. Thus, our solution domains for $\boldsymbol{g}$ and $\delta\boldsymbol{p}$ are convex. To show that the solution found by our procedure does in fact maximize $\bar{z}(\boldsymbol{g})$, we need merely show that the function $\bar{z}$ is itself concave.

Since we are maximizing $\bar{z}$, we want to show the $\bar{z}$ is concave (equivalently, $-\bar{z}$ is convex), that is

$$\bar{z}(\alpha\boldsymbol{g}_1 + (1-\alpha)\boldsymbol{g}_2) \geq \alpha\bar{z}(\boldsymbol{g}_1) + (1-\alpha)\bar{z}(\boldsymbol{g}_1) \qquad (12)$$

for any $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ in the same partition and $0 \leq \alpha \leq 1$. Since

$$\bar{z}(\boldsymbol{g}) = \min\left(-\sum_{i=1}^{N} m_i \boldsymbol{g} \cdot \delta\boldsymbol{r}_i\right) \qquad (13)$$

we have

$$\begin{aligned} &\bar{z}(\alpha\boldsymbol{g}_1 + (1-\alpha)\boldsymbol{g}_2) = \\ &\min\left(-\sum_{i=1}^{N} (\alpha\boldsymbol{g}_1 + (1-\alpha)\boldsymbol{g}_2) \cdot m_i \delta\boldsymbol{r}_i\right) = \\ &\min\left(\left(-\alpha\boldsymbol{g}_1 \cdot \sum_{i=1}^{N} m_i\delta\boldsymbol{r}_i\right) + \left(-(1-\alpha)\boldsymbol{g}_2 \cdot \sum_{i=1}^{N} m_i\delta\boldsymbol{r}_i\right)\right) \end{aligned} \qquad (14)$$

Since $\min(f_1 + f_2) \geq \min f_1 + \min f_2$ for any functions f_1 and f_2, we have

$$\begin{aligned} &\min\left(\left(-\alpha\boldsymbol{g}_1 \cdot \sum_{i=1}^{N} m_i\delta\boldsymbol{r}_i\right) + \left(-(1-\alpha)\boldsymbol{g}_2 \cdot \sum_{i=1}^{N} m_i\delta\boldsymbol{r}_i\right)\right) \geq \\ &\min\left(-\alpha g_1 \cdot \sum_{i=1}^{N} m_i\delta\boldsymbol{r}_i\right) + \min\left(-(1-\alpha)\boldsymbol{g}_2 \cdot \sum_{i=1}^{N} m_i\delta\boldsymbol{r}_i\right) \\ &= \alpha\bar{z}(\boldsymbol{g}_1) + (1-\alpha)\bar{z}(\boldsymbol{g}_2) \end{aligned} \qquad (15)$$

Thus

$$\bar{z}(\alpha\boldsymbol{g}_1 + (1-\alpha)\boldsymbol{g}_2) \geq \alpha\bar{z}(\boldsymbol{g}_1) + (1-\alpha)\bar{z}(\boldsymbol{g}_2) \qquad (16)$$

and we have that $\bar{z}(\boldsymbol{g})$ is concave.

The result that $\bar{z}(\boldsymbol{g})$ is concave is significant since this means that the stationary points found for each partition of S by the iterative procedure of the previous section are a global maximum for that partition of S. By computing the maximum over each of the eight facets of the “diamond” and taking their largest value, we can find the global maximum of $\bar{z}(\boldsymbol{g})$. If the global maximum is zero, then we have found a value for $\boldsymbol{g}$ that yields a stable orientation. Otherwise, the system is unstable, and we have a value of $\boldsymbol{g}$ that minimizes this unstability, as measured by the function $\bar{z}(\boldsymbol{g})$.

4 Results

The solution methods for the two problems described in section 3 have been implemented and tested. In this section we present some results. Geometric models of assemblies are created using the nonmanifold geometric modeler Noodles[6]. The geometry of contact is derived directly from the model and is used to construct the linear programs to be solved. Figure 8 and Figure 9 are examples of the stability problem and Figure 10, Figure 11 and Figure 12 are examples of the stable orientation problem.

Figure 8 shows an assembly of 3 parts where part C_1 is grounded. C_2 and C_3 are chamfered at angles of $60°$ and $30°$ and are arranged as shown. When the assembly is tested for stability, it is declared unstable. The displacements $\delta\boldsymbol{p}$ that minimize z produce a downward translation for C_2 of magnitude 1.0 and a sideward translation of 0.577 $(1/\tan 60°)$ for C_3.

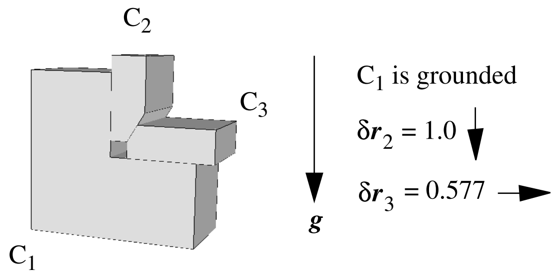

Figure 8: The assembly is unstable. The displacements found are indicated alongside the assembly.

The assembly in Figure 9 consists of four identical blocks chamfered at two edges at an angle of 45°. They are placed within a square frame as shown. This assembly is a good example of the interdependence of part motions because of the presence of loops within the contact graph.This assembly is stable because although the parts can move in a circular fashion, this motion does not result in a net decrease in potential energy.

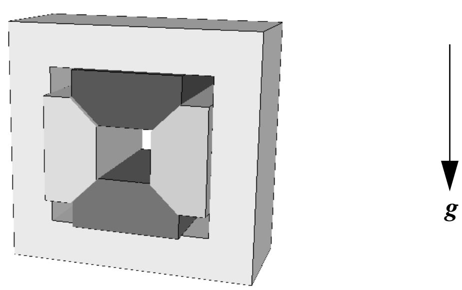

Figure 9: An assembly with 4 identical blocks within a frame. This is a good test assembly since its graph of connections has loops. This assembly is stable.

Figure 10(a), Figure 11(a) and Figure 12(a) show assemblies in unstable orientations; in all cases the L-shaped object is grounded. The grid shows the XY plane with gravity along the $-Z$ direction. Stable orientations are computed and are shown alongside each assembly in (b).

In Figure 10, the assembly is stable over a range of orientations. The method that we have proposed does not distinguish between these orientations, and may choose any one of them as a stable orientation. We plan to extend our method to produce the entire range of stable orientations.

5 Summary and Conclusions

Geometric models of parts and assemblies are increasingly being used to plan manufacturing processes, including assembly. Analysis of the physical behavior of objects and assemblies is being automatically derived for process planning. In this paper, the problem of the gravitational stability of assemblies is addressed. The assumptions made are as follows. (a) The objects are rigid and their contacts are frictionless and (b) one (or more) of the objects is grounded. The second assumption, in general, is a statement of constraint on the absolute motion of some objects. Solutions to two problems have been proposed. The first addresses the problem of determining whether a given assembly is gravitationally stable. Using a Lagrangian formulation of virtual work, motions that decrease potential energy are formulated and computed using linear programming. The second problem that is addressed is that of finding a gravitationally stable orientation for an assembly. This can be formulated as a maximin problem in terms of the orientation vector and the motions of the bodies. Algorithms to solve the two problems have been implemented and preliminary results have been presented. The solution to the second problem is the first general method that automatically determines stable orientations of assemblies. This is a useful tool in automatic assembly planning.

In future work, the stability analysis needs to be extended to include frictional forces. The stability result that is predicted by a frictionless model is too conservative. By considering frictional effects, the range of stable orientations found can be extended. Additionally, we would like to extend our method so that it can report the range of stable orientations, rather than picking a single stable orientation arbitrarily. This is useful during assembly planning when an assembly needs to be reoriented.

6 Acknowledgments

The authors would like to thank Matt Mason for very useful input during early discussions. Also the help of Levent Gursoz and Jim Hemmerle for their insight into the mechanics of bodies, and Jagadisan Viswanathan's contribution during our discussions in solving LP problems is greatly appreciated.

References

[1] David Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. *Computer Graphics.* 24(4):223-232, August, 1989.

[2] David Baraff, Raju Mattikalli, Bruno Repetto and Pradeep Khosla. Technical Report CMU-RI-93-13, Carnegie Mellon University, 1993.

[3] M Blum, A Griffith and Bernard Neumann. *A stability test for configurations of blocks.* Technical Report AI Memo No.188, Massachusetts Institute of Technology, February, 1970.

[4] Vasek Chvatal. *Mathematical Sciences Linear Programming*. W H Freeman and Co., New York, 1983.

[5] Vladimir Demianov and Vasilii Nikolaevich. *Introduction to Minimax.* Wiley, New York, 1974.

[6] Levent Gursoz, Young Choi and Fritz Prinz. Vertex-Based representation of Non-Manifold Boundaries. *Geometric Modeling for Product Engineering*. North - Holland, New York, 1988, pages 107 - 130.

[7] Luiz S Homem de Mello. *Task Sequence Planning for Robotic Assembly.* PhD thesis, Carnegie Mellon University, May, 1989.

[8] Richard S Palmer. *Computational complexity of motion and stability of polygons.* PhD thesis, Dept. of Computer Science, Cornell University, May, 1989.

[9] R. S. Mattikalli, P.K. Khosla and Y. Xu. Subassembly Identification and Motion Generation for Assembly: A Geometrical Approach. *Proceedings of IEEE Conference on System Engineering*, pages 399-403. 1990.

[10] R. S. Mattikalli and P. K. Khosla. Motion Constraints from Contact geometry: Representation and Analysis. *Proc. of the IEEE Int. Conf. on Robotics and Automation, 1992*. IEEE, May, 1992.

[11] Nico Boneschanscher, Hans van der Drift, Stephen J Buckley and Russell H Taylor. Subassembly Stability. *In Proc. of the National Conference on Artificial Intelligence*, pages 780-785. Aug., 1988.

[12] Randall H Wilson. *On Geometric Assembly Planning*. PhDThesis STAN-CS-92-1416, Stanford University, Stanford, California 94305, March, 1992.

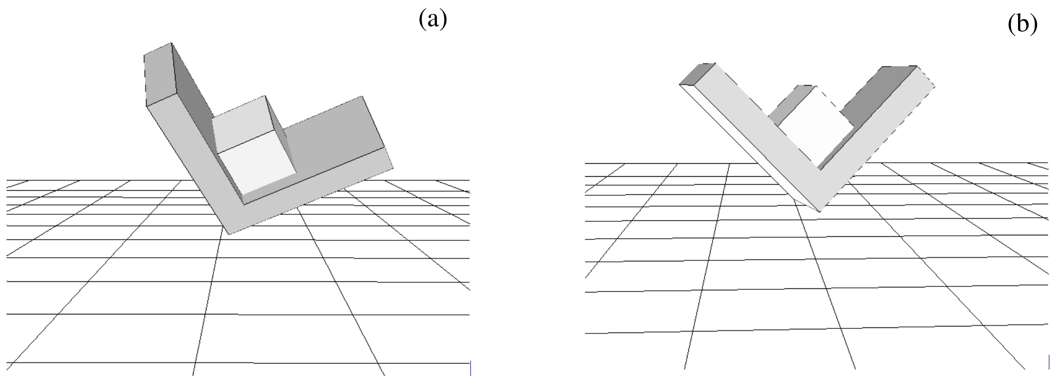

Figure 10: Finding a stable orientation. The L-shaped object is grounded.

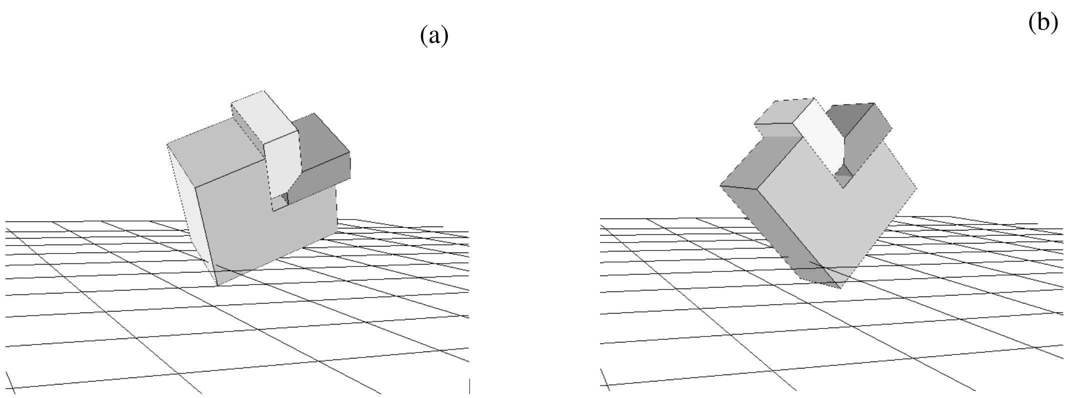

Figure 11: Finding a stable orientation. (a)The L-shaped part is grounded and the two blocks are identical with a chamfer of 45 degrees. (b) The stable orientation shown on the right has the L-shaped part at an angle of $45°$.

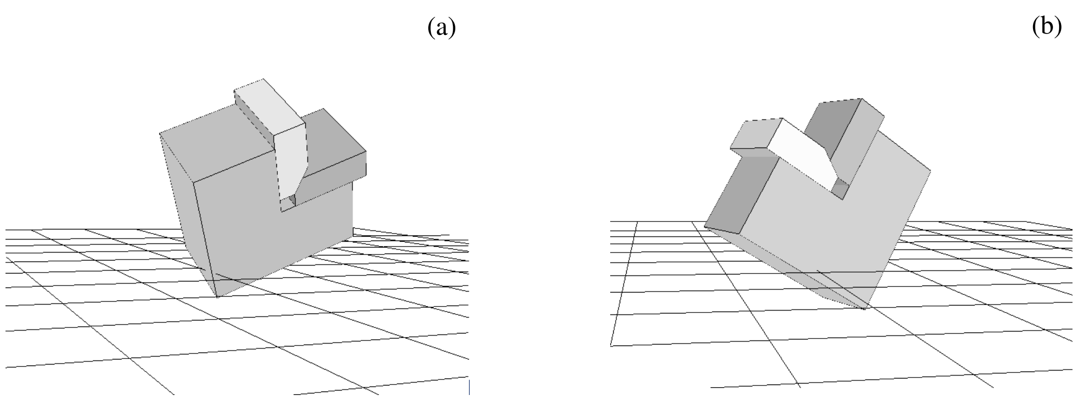

Figure 12: Assembly with the two blocks chamfered at $60°$. (b) In the stable orientation, the grounded L-shaped object leans at an angle of $60°$.