A SPATIAL CONSTRAINT PROBLEM

C. M. HOFFMANN
Purdue University
Department of Computer Sciences
West Lafayette, IN 47907-1398, USA

AND

P. J. VERMEER
Washington and Lee University
Department of Computer Science
Lexington, VA 24450, USA

Abstract. Three-dimensional geometric constraint solving is a rapidly developing field, with applications in areas such as kinematics, molecular modeling, surveying, and geometric theorem proving. While two-dimensional constraint solving has been studied extensively, there remain many open questions in the arena of three-dimensional problems. In this paper, we continue the development of our previous work on configuring a set of points and planes in three-space so that the configuration satisfies a given system of constraints. The constraint system considered consists of six geometric elements and pairwise constraints between triples of the elements. We first review the basic techniques developed in our earlier work germaine to the current problem and explain how the problem we consider in this paper occurs. We then demonstrate how to solve the case of a geometric constraint system with four points and two planes.

1. Introduction

The spatial geometric constraint problem consists of a set of geometric entities, and a prescription of geometric constraints between the elements. The goal is to find all placements of the geometric entities which satisfy the given constraints. Two-dimensional constraint solving has been studied extensively, yet many open questions remain for the spatial problem.

A problem is *well-constrained* if it has a finite number of solutions, while a problem with an infinite number of solutions is *underconstrained.* A problem is *overconstrained* if one constraint can be deleted yet the problem still has a finite number of solutions. An overconstrained problem may have a solution when the additional constraints are consistent with previous constraints, but often overconstrained problems have no solution.

Applications of constraint solving to kinematics include determining whether a mechanism is over- or underconstrained and its degrees of freedom. Further, by arbitrarily fixing underconstrained sketches, instance configurations of mechanisms can be computed. In the case of complex mechanisms, this may lead to reasonable kinematic simulations. Conversely, some techniques used in kinematics can also be used in constraint solving.

In this paper, we continue to develop our previous work on configuring a set of points and planes in three-space so that a given system of constraints is satisfied. The problem considered has six geometric elements and pairwise constrains four triples of the elements. After a brief review of basic techniques and problem context, we demonstrate how to solve the case of a certain geometric constraint system with four points and two planes.

2. General Solution Technique

Given a set of geometric elements and certain constraints between them, there are two basic strategies for solving the constraint problem. An *instance solver* uses the explicit values of the given constraints to determine the possible geometric configurations which satisfy the constraints. A *generic solver* determines whether the given geometric elements can be placed independent of the particular values assigned to the constraints. That is, the constraints have a symbolic rather than numerical value. The geometric elements are placed only after a decision has been made about whether or not the problem is generically well-constrained.

There are a variety of ways to implement each of these two strategies. Numerical constraint solvers first translate the constraints into a system of algebraic equations. This system is then solved using an iterative technique such as the Newton-Raphson method. Examples of numerical solvers include Sketchpad (Sut63) and ThingLab (Bor81). Symbolic constraint solvers also begin by setting up a system of algebraic equations. However, general symbolic computations are applied first to simplify the system, before solving it numerically. Methods such as Gröbner bases (Bos85) or Wu-Ritt (WT86) techniques can be applied to find symbolic expressions for the solutions. For example, Kondo uses the symbolic computation for adding and deleting constraints from a given system (Kon92). Logical inference and term rewriting applies general logical reasoning techniques to the constraint solving problem. This approach has been taken by Aldefeld (Ald88) and Bruderlin (Bru90). For a deeper literature review see (Fud93; DH95).

2.1. GRAPH-BASED CONSTRUCTION SEQUENCES

Our approach to constraint solving is graph-based. Graph-based algorithms for solving geometric constraint problems have a graph analysis phase and a construction phase. First, a graph representation of the problem is constructed, where the nodes of the graph correspond to geometric entities, and an edge corresponds to a constraint between entities. A graph analysis determines whether the problem is well-constrained and a construction sequence. The graph analysis is more discriminating than Gruebler's or Reuleaux's criteria (Bar93; Phi84). It does not, however, account for degeneracies such as three planes intersecting in a line rather than a point due to specific constraint values. If the graph is (generically) well-constrained, this phase also determines a sequence of steps for solving the problem.

The second phase of the graph method takes the construction sequence determined from the first phase and performs the necessary construction steps to actually place the geometric elements. Since the first phase does not depend on the values of the constraints but only on the number and type of constraints between the geometric elements, we have a generic method of constraint solving. The actual values of the constraints only considered in the second phase when the construction steps are carried out.

One way to handle the analysis phase of the graph-based method looks for collections of geometric elements whose members can be placed with respect to one another based on constraints between them. These collections are then placed relative to one another, thus forming new, larger collections of elements, until all constraints have been processed and the locations of all the elements are known. We propose to use this recursive method to analyze the constraint graph for the three dimensional problems. This approach extends the two-dimensional constraint solving method developed in (BFH$^+$94; Fud93). In the following sections we give a very brief overview of the method; further details and examples can be found in (DH95).

2.2. GEOMETRIC ENTITIES AND CONSTRAINTS

In the following, we restrict to considering only points and planes in $\mathcal{R}^3$. A point is represented simply by its coordinate triple (x, y, z). A plane is represented by its unit normal (n_x, n_y, n_z) and the distance from the origin to the plane d. This simplifies matters because each of the geometric entities has three degrees of freedom, thus allowing a uniform handling of the constraints and geometries throughout the solution. The constraints allowed are distance between two points, signed distance between a point and a plane, and angle between two planes. When an angle is given between two planes, it is assumed to be the angle between the sides of each plane positive with respect to the plane normals.

2.3. GRAPH ANALYSIS

The first step of the construction is to form *clusters* of geometric elements which are placed in fixed postions with respect to one another. Because each geometric element has three degrees of freedom, placing a new element requires that it be constrained by three known elements. Thus to begin a cluster, a set of three pairwise constrained nodes is necessary. These three geometric elements are placed into a standard position and the resulting configuration is fixed up to a rigid motion in space. Subsequently, a node is added to the cluster if it is incident to three nodes already in the cluster.

When no further nodes can be added to the cluster, the edges representing constraints between nodes in the cluster are deleted from the original graph, as well as all isolated nodes. Cluster formation is then applied recursively to this subgraph. This cluster forming and subsequent graph pruning is carried out as long as possible. Because a cluster start requires three pairwise constrained elements, there may eventually be unused constraints in the remaining subgraph, yet no new cluster can be started. In this case, any remaining constraint and its two incident nodes forms a *degenerate cluster*.

For cluster formation we need to find a generic placement for the first three elements which are pairwise constrained within a cluster, and we need to place a geometric element from three known elements. Because we use signed distances and angles, well-defined generic configurations exist. For the details see (DH95).

2.4. CLUSTER MERGING

Once initial clusters have been formed, clusters which have geometric elements in common can be placed relative to one another. Each cluster is a rigid body and has in general six degrees of freedom, three rotational and three translational. Exceptions include degenerate clusters such as a plane with an incident point. Here, one degree of freedom is lost by symmetry. To fix a cluster in space, we must determine how to place the shared elements of the cluster with respect to the other clusters sharing them. Three *separate* geometric entities are needed to fix the cluster. Once these elements are known, the rigid-body motion to position the cluster can be determined using the techniques of (RG93). Degenerate clusters share only two geometric elements and can be positioned from them because of cluster symmetry.

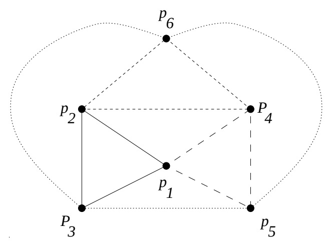

Figure 1. The constraint graph for Case 1 of four points and two planes in four clusters.

Now, if two clusters A and B share two geometric elements, then the clusters are overconstrained, because the relative position of the shared elements is determined independently in both cluster. Therefore, the three shared elements in the cluster must belong to three *separate* other clusters. (For degenerate clusters, the two elements in the cluster are each shared with a different cluster.)

3. Merging Four Full Clusters

We consider merging four clusters, each with three geometric elements shared with the other clusters. Six geometric elements must be placed relative to each other. When the geometric elements are all points, the problem can be solved as a Stewart platform problem(NWM90). In (DH95) the case of five points and one plane is handled. We now show how to solve the constraint system when two of the elements are planes. There are two cases.

3.1. CASE 1

The first case we consider is when the two planes are not in the same cluster. The four clusters for the problem are (p_1, p_2, P_3), (p_1, P_4, p_5), (p_2, P_4, p_6), and (P_3, p_5, p_6). Note that this arrangement satisfies the criterion that each element of a given cluster is in exactly one of the other three clusters. The constraints within each cluster are the distances between the two points in each cluster, and the (signed) distance from each point to the plane in the cluster. A graphical representation of these clusters is shown in Figure 1, with the clusters distinguished by the type of line of the constraint edges. Each edge in the graph represents a distance constraint. A diagram of the geometry of this situation is shown in Figure 2. The open circles in P_3 represent the points in P_3 which satisfy the distance constraints between P_3 and the points p_2, p_5, and p_6, and analogously for the open circles in P_4. The elements of each cluster are connected by the same type of line.

Suppose that the distance between any two geometric elements g_i and g_j, where g is a point or a plane, is given by d_{ij}. If we assume the distances

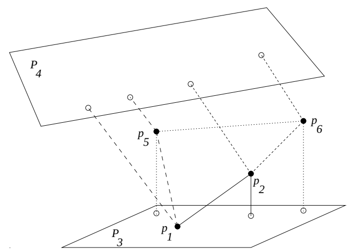

Figure 2. Four points and two planes with distance constraints only.

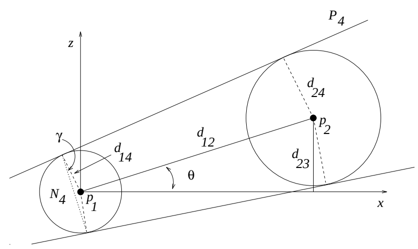

Figure 3. Constructing the plane P_4.

between a point and the plane are signed, then we can modify the problem so that the point p_1 lies in the plane P_3. If we can find a configuration where p_1 lies in P_3 and the respective distances between each of the three points p_2, p_5, and p_6 and P_3 are reduced by the given distance from p_1 to P_3, the actual configuration which satisfies the given constraints can be found by offsetting P_3 in the found configuration by d_{13}.

We place p_1 at the origin, make P_3 the xy-plane, and place p_2 at $(l_1, 0, d_{23})$, where $l_1^2 + d_{23}^2 = d_{12}^2$. The distance constraints between p_1 and P_4 and between p_2 and P_4 force P_4 to be tangent both to the sphere S_{14} centered at p_1 with radius d_{14} and to the sphere S_{24} centered at p_2 with radius d_{24}. If the two distances have the same sign, P_4 must be tangent to the cone containing and tangent to both spheres as shown in Figure 3. If the signs are opposite, the two spheres are contained in and tangent to a different cone, and the spheres are in opposite half-cones of the cone.

Note that the direction of the normal to P_4, $N_4 = (n_x, n_y, n_z)$ must be on a circle within the sphere S_{14}. The projection to the cross-section of this circle is shown in dotted line in Figure 3. To obtain a unit normal, we begin with a circle of radius $\cos\gamma$ in the xz-plane, with center $(-\sin\gamma, 0, 0)$, where γ is the apical angle of the cone. This circle is then rotated by θ about the y-axis, where θ is the angle between the line p_1p_2 and the x-axis. The result of these operations is

$$N_4 : (\cos\gamma \sin\theta \sin u - \sin\gamma \cos\theta, \cos\gamma \cos u, \cos\gamma \cos\theta \sin u + \sin\gamma \sin\theta)$$

This means that the plane P_4 can be written in terms of the variable u as $P_4 : N_4 \cdot (x, y, z) = d_{14}$.

Now, p_5 lies in a plane parallel to P_3 and also on a sphere of radius d_{15} centered at p_1. Thus it must lie on the circle C_5 given by

$$C_5 \,:\, (r_5 \cos v, r_5 \, \sin v, \, d_{35})$$

Similarly p_6 lies in a plane parallel to P_3 and also on a sphere of radius d_{26} centered at p_2. Since the distance from p_1 to the projection of p_2 into P_3 is l_1 (from the earlier positioning of p_2), p_6 must lie on the circle C_6 given by

$$C_6 \,:\, (l_1 + r_6 \, \cos w, r_6 \sin w, \, d_{36})$$

The parameters l_1, $\cos\theta$, $\sin\theta$, $\cos\gamma$, $\sin\gamma$, r_5, and r_6 are all dependent only on the distances given between elements of the clusters. Specifically, we have the following relationships:

$$\begin{array}{lcllcl} l_1 & = & \sqrt{d_{12}^2 - d_{23}^2} & l_2 & = & \sqrt{d_{12}^2 - (d_{24} - d_{14})^2} \\ \cos\theta & = & l_1/d_{12} & \cos\gamma & = & l_2/d_{12} \\ \sin\theta & = & d_{23}/d_{12} & \sin\gamma & = & (d_{24} - d_{14})/d_{12} \\ r_5 & = & \sqrt{d_{15}^2 - d_{35}^2} & r_6 & = & \sqrt{d_{26}^2 - (d_{36} - d_{23})^2} \end{array} \quad (1)$$

We now can use the remaining distance constraints to set up three equations in the unknowns u, v, and w which when solved will give the configurations. The remaining constraints are the distance between p_5 and P_4, the distance between p_6 and P_4, and the distance between p_5 and p_6. These constraints are translated into the following trigonometric equations:

$$N_4 \cdot p_5 + d_{45} = d_{12} \quad (2)$$

$$N_4 \cdot p_6 + d_{46} = d_{12} \quad (3)$$

$$(r_5 \cos v - r_6 \cos w - l_1)^2 + (r_5 \sin v - r_6 \sin w)^2 + (d_{35} - d_{36})^2 = d_{56}^2 \quad (4)$$

Making the standard substitutions $\cos u = \frac{1-q_4^2}{1+q_4^2}$ $\sin u = \frac{2\,q_4}{1+q_4^2}$, $\cos v = \frac{1-q_5^2}{1+q_5^2}$ $\sin v = \frac{2\,q_5}{1+q_5^2}$, $\cos w = \frac{1-q_6^2}{1+q_6^2}$, and $\sin w = \frac{2\,q_6}{1+q_6^2}$, we obtain three equations with the following structure:

$$(A_1\, q_5^2 + A_2\, q_5 + A_3)\, q_4^2 + (A_4\, q_5^2 + A_6)\, q_4 + (A_7\, q_5^2 + A_8\, q_5 + A9) = 0 \quad (5)$$

$$(B_1\, q_6^2 + B_2\, q_6 + B_3)\, q_4^2 + (B_4\, q_6^2 + B_6)\, q_4 + (B_7\, q_6^2 + B_8\, q_6 + B9) = 0 \quad (6)$$

$$(D_1\, q_5^2 + D_3)\, q_6^2 + D_5\, q_5\, q_6 + (D_7\, q_5^2 + D_9) = 0 \quad (7)$$

Here the coefficients A_i, B_j, and D_k are functions only of the constants l_1, $\cos\theta$, $\sin\theta$, $\cos\gamma$, $\sin\gamma$, r_5, d_{35}, d_{36}, d_{45}, d_{46}, and d_{56}. This is exactly the same structure as the system of equations which arose in the case of five points and one plane, and its solution is detailed in (DH95). Note that the system has 16 solutions, in complex projective space.

3.2. CASE 2

The second case occurs when the two planes are both in one of the clusters, and another cluster has only points. Specifically, the clusters now are (p_1, p_2, P_3), (p_1, p_4, p_5), (p_2, p_4, P_6), and (P_3, p_5, P_6). The constraints are distances between any point-point or point-plane pairs within each cluster, and an angle constraint α_{36} between the two planes in the fourth cluster.

As before, we assume that p_1 lies in P_3, and we begin by positioning p_1 as the origin, P_3 as the xy-plane, and p_2 as the point $(l_1, 0, d_{23})$, where $l_1^2 + d_{23}^2 = d_{12}^2$. Based on the distance constraints between the point p_4 and the points p_1 and p_2, we can determine the circle C_4 on which p_4 must lie, which is the intersection of the sphere S_{14} centered at p_1 with radius d_{14} and the sphere S_{24} centered at p_2 with radius d_{24}. Assume that the center of this circle is l_4 units from p_1 along the line p_1p_2, and that the radius of the intersection circle is r_4. Then the intersection circle can be found by rotating the circle of radius r_4 centered at $(l_4, 0, 0)$ about the y-axis by θ degrees, where θ is the angle between the line p_1p_2 and the x-axis. This results in the following equation for C_4:

$$C_4 \ : \ (l_4 \cos\theta + r_4 \sin\theta \sin u, \ r_4 \cos u, \ -l_4 \sin\theta + r_4 \cos\theta \sin u)$$

The values of $\cos\theta$ and $\sin\theta$ are computed as in Equations 1, and the values of l_4 and r_4 are

$$l_4 \quad = \quad \frac{d_{24}^2 - d_{14}^2 - d_{12}^2}{2d_{12}} \qquad\qquad r_4 \quad = \quad \sqrt{d_{14}^2 - l_4^2}$$

The circle of possible points for p_5 is identical to the previous case:

$$C_5 \ : \ (r_5 \ \cos v, r_5 \ \sin v, \ d_{35})$$

The plane P_6 is completely determined by its normal and its distance from the origin. Note, however, that d_{16} is not one of the known constraints. Thus we refer to this distance as the variable l_{16}. Now the normal of P_6 can be computed as a function of the angle between P_6 and P_3, and some parameter w as

$$N_6 : (\cos\alpha_{36} \cos w, \cos\alpha_{16} \sin w, \sin\alpha_{16})$$

Since we do know the distance from p_2 to P_6, and p_2 is fixed, we can express l_{16} in terms of this constraint as

$$l_{16} = p_2 \cdot N_6 + d_{26}$$

The three equations which express the remaining constraints are

$$\begin{aligned}(l_4\cos\theta + r_4\sin\theta\sin u - r_5\cos v)^2 + (r_4\cos u - r_5\sin v)^2 + \\ (-l_4\sin\theta + r_4\cos\theta\sin u - d_{35})^2 &= d_{45}^2 \\ C_4\cdot N_6 + d_{46} &= l_{16} \\ C_5\cdot N_6 + d_{56} &= l_{16}\end{aligned}$$

When standard rational substitutions are made for the trigonometric functions, the resulting three equations have form identical to Equations 5, 6, and 7.

3.3. NUMERICAL EXAMPLE

To verify the above process, we consider a numerical example with one predetermined solution The initial configuration is

$$\begin{array}{lllllllll} p_1 &=& (0,0,0) & p_2 &=& (3,0,4) & P_3 &=& xy\text{−plane} \\ P_4 &=& N:(3/5,4/5,0); d:5 & p_5 &=& (2,2,2) & p_6 &=& (6,1,1)\end{array}$$

This means the input to the problem is four clusters with the following distance constraints:

Cluster 1	Cluster 2	Cluster 3	Cluster 4
$d_{13}=0$	$d_{14}=5$	$d_{24}=-16/5$	$d_{56}=3\sqrt{2}$
$d_{23}=4$	$d_{15}=2\sqrt{3}$	$d_{46}=-3/5$	$d_{35}=2$
$d_{12}=5$	$d_{45}=-11/5$	$d_{26}=\sqrt{19}$	$d_{36}=1$

Note the signed distances between P_3 and P_4, and the points with distance constraints with respect to the two planes. From these distances the parameters of the three equations are computed:

$$\begin{array}{llllllllllll} l_1 &=& 3, & l_2 &=& 4/5\sqrt{34}, & \cos\theta &=& 3/5, & \cos\gamma &=& 4\sqrt{34}/25, \\ \sin\theta &=& 4/5, & \sin\gamma &=& 9/25, & r_5 &=& 2\sqrt{2}, & r_6 &=& \sqrt{10}\end{array}$$

Substituting these values into Eqs. (5), (6), and (7) and following the solution procedure detailed in (DH95), we found six solutions for q_5^2, two of which were positive. The corresponding values of q_5, $\cos v$, $\sin v$, $\cos u$, $\sin u$, $\cos w$, and $\sin w$ are shown in Table 3.3, rounded to six digits. Evaluating Eqs. (2), (3), and (4) at these points yielded the following results for the normal N_4 of P_4 and for the points p_5 and p_6 on the circles C_5, and C_6, respectively, rounded again to six digits:

Solution 1

N_4 = (-.937703,.237830, -.253277)
p_5 = (2.67722, 0.9124, 2.0)
p_6 = (3.63658, -3.09754, 1.0)

Solution 2

N_4 = (-.937703, -0.237830, -.253277)
p_5 = (2.67722, -0.9124, 2.0)
p_6 = (3.63658, 3.09754, 1.0)

Solution 3

N_4 = (-0.6,-0.8, 0.0)
p_5 = (2.0, 2.0, 2.0)
p_6 = (6.0, 1.0, 1.0)

Solution 4

N_4 = (-0.6, 0.8, 0.0)
p_5 = (2.0, -2.0, 2.0)
p_6 = (6.0, -1.0, 1.0)

TABLE 1. Real solutions to the numerical example

q_5	$\cos v$	$\sin v$	$\cos u$	$\sin u$	$\cos w$	$\sin w$
0.165721	0.946541	0.322582	.254922	-.966962	.201306	-.979528
-0.165721	0.946541	-0.322582	-.254922	-.966962	.201306	.979528
0.414214	0.707107	0.707107	-.857493	-.514496	.948683	.316228
-0.414214	0.707107	-0.707107	.857493	-.514496	.948683	-.316228

Solution 3 is the predetermined solution, which corroborates the correctness of the solution procedure.

References

B. Aldefeld. Variation of geometries based on a geometric-reasoning method. *CAD*, 20(3):117–126, April 1988.

L. O. Barton. *Mechanism Analysis*. Marcel Dekker Inc., 1993.

W. Bouma, I. Fudos, C. Hoffmann, J. Cai, and R. Paige. A geometric constraint solver. *CAD*, 1994.

A. H. Borning. The programming language aspects of ThingLab, a constraint oriented simulation laboratory. *ACM TOPLAS*, 3(4):353–387, 1981.

N.K. Bose, editor. *Multidimensional Systems Theory*, pages 184–232. D. Reidel Publishing Co., 1985.

Beat Bruderlin. Symbolic computer geometry for computer aided geometric design. In *Advances in Design and Manufacturing Systems*. NSF, 1990.

Ding-Zhu Du and Frank Hwang, editors. *Computing in Euclidean Geometry*, pages 266–298. World Scientific Publishing Co Pte Ltd, 2 edition, 1995.

I. Fudos and C. Hoffmann. Correctness of a geometric constraint solver. Technical Report CSD-TR-93-076, Purdue University, 1993.

I. Fudos. Editable representations for 2d geometric design. Master's thesis, Purdue University, 1993.

K. Kondo. Algebraic method for manipulation of dimensional relationships in geometric models. *CAD*, 24(3):141–147, March 1992.

P. Nanua, K. J. Waldron, and V. Murthy. Direct kinematic solution of a stewart platform. *IEEE Transactions on Robotics and Automation*, 6(4):438–444, Aug. 1990.

J. Phillips. *Freedom in Machinery*, volume 1. Cambridge University Press, 1984.

B. Ravani and Q. J. Ge. Computation of spatial displacements from geometric features. *J. Mech. Design*, 115:95–102, March 1993.

Ivan E. Sutherland. Sketchpad: A man-machine graphical communication system. In *Proceedings–Spring Joint Computer Conference*, pages 329–346, 1963.

Wu Wen-Tsun. Principles of mechanical theorem proving. *J. of Automated Reasoning*, 2:221–252, 1986.