Computer-Aided Design 50 (2014) 51–60

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Coordinate-free geometry and decomposition in geometrical constraint solving

P. Mathis *, P. Schreck

ICube, UMR CNRS 7357, University of Strasbourg, France

HIGHLIGHTS

- We solve 2D/3D geometric constraints by using coordinate-free formulation.
- Statements are translated into formalism that reduces the number of equations.
- A decomposition algorithm is proposed to produce small subsystems.
- Subsystems are solved by homotopy.

ARTICLE INFO

Article history:
Received 6 March 2013
Accepted 28 January 2014

Keywords:
Geometrical constraint solving
Cayley–Menger determinants
System decomposition
Equational system optimization
Homotopy

ABSTRACT

In CAD, a designer usually specifies mechanisms or objects by the means of sketches supporting dimension requirements like distances between points, angles between lines, and so on. This kind of geometric constraint satisfaction problems presents two aspects which solvers have to deal with: first, the sketches can contain hundreds of constraints, and, second, the problems are invariant by rigid body motions. Concerning the first issue, several decomposition methods have been designed taking invariance into account by fixing/relaxing coordinate systems. On the other hand, some researchers have proposed to use distance geometry in order to exploit invariance by rigid body motions. This paper describes a method that allows us to use distance geometry and decomposition in the same framework.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Problems of geometric constraint satisfaction where constraints express geometric requirements related to distances, angles, tangencies and incidences occur in many fields such as CAD, geometric modeling, robotic or molecular modeling. The problem consists in finding the position and the orientation of some given objects in a way that some given constraints are satisfied. More particularly, in CAD, problems come from the specification of pieces or mechanisms given in the form of a sketch drawn by a designer using basic geometric primitives such as points, lines, planes, circles, spheres, etc., and metric requirements such as distances between points, angles between lines or planes. In that framework, the aim is to get coordinates for the points, planes, circles, etc., so that the incidence relations given by the sketch, and the added metric constraints are fulfilled. By essence, these problems are invariant by direct isometries (or rigid body motions for engineers), that is, applying a rigid body motion on a solution yields another solution. It is also usual to assume that the objects are well defined that is there is a finite number of solutions up to rigid body motions. In that situation, the problem is said to be *well-constrained*.

Several methods designed to solve such problems are described in the literature (see [1] for a recent state of the art). Some methods consider the coordinates of primitives and try to solve the equation system either numerically, for instance with Newton–Raphson [2] or homotopy [3] methods, or symbolically for instance by using Gröbner basis or Ritt–Wu principle [4]. However, these methods suffer drawbacks that we want to overcome: symbolic methods have an exponential complexity, and numerical methods are unable to produce all the solutions within a reasonable time. Other methods are more sticked to geometry: they use knowledge-based systems (KBS) [5,6] or graph operations [7] to solve small problems and are designed to decompose larger problems into smaller subproblems when it is possible. These methods for decomposing geometric problems always take into account the rigid body motion invariance. The way they proceed amounts to first fix some coordinates and then relax them by letting act the group of rigid body motions on the solutions for the subproblems.

* Corresponding author. Tel.: +33 368854561.
E-mail addresses: mathis@unistra.fr (P. Mathis), schreck@unistra.fr (P. Schreck).

http://dx.doi.org/10.1016/j.cad.2014.01.012
0010-4485/© 2014 Elsevier Ltd. All rights reserved.

When successful, these techniques may allow to yield all solutions in a reasonable time.

Unfortunately, 3D problems, even the ones involving few objects, are hard or impossible to solve by using only constructive methods. And numerical solvers have to face systems with dozens of equations and whose global degree is too high to enable the use of homotopy to follow all the possible paths. For some years, another approach has been studied which ignores coordinates in the first time and which is more able to take rigid body motion invariance into account [8–10]. This approach is based on coordinate-free geometry. The idea consists of replacing coordinates by all the distances between points. The main ingredient to write down equations is the Cayley–Menger determinant between some sets of constrained points. This method had been successful on some problems like the Stewart platform problem [8] or the Malfatti problem [9] where the equation system to solve is very small compared to the classical approach based on Euclidean coordinates. However, such systems were built by hand and, in the coordinate-free geometry framework, there are no known methods for decomposing big systems.

We propose here a method to automatically produce equation systems using the Cayley–Menger determinant and to structurally decompose the constraint systems whenever possible. The resulting systems can then be solved by classical homotopy in order to yield either all the solutions, or, at least, the highest number of possible solutions.

This paper is organized as follows. Section 2 introduces distance geometry and Cayley–Menger determinants. Section 3 gives algorithms to provide Cayley–Menger systems from constraints systems. Section 4 presents our algorithm of decomposition of Cayley–Menger systems. The numerical approach and some results are given in Section 5. Section 6 concludes and mentions some limits and future works.

2. Coordinate-free geometry

2.1. Geometrical constraint system

A geometric constraint system, in short GCS, is defined by a triple $C[X, A]$ where X is a set of unknowns, A is a set of parameters and C is a set of constraints on X and A. The unknowns correspond to some geometric *entities* (also called primitives or objects) such as points, lines, circles, planes, *etc.*, the constraints are relationships involving distances, angles, tangencies, incidences, *etc.*, and the parameters are the values imposed as dimensions like distance and angle values.

More specifically, we consider here rigid bars problems as encountered for instance in CAD, robotics and molecular modeling. So, entities are points and hyperplanes, that is, lines in 2D and planes in 3D; constraints are about distances between two entities and angles between hyperplanes. The model that we use does not include 3D lines. However, it is often possible to replace them by pair of incident points. For instance, when angles between 3D lines are considered, they often concern adjacent segments. So the angle between segment $p1p2$ and segment $p1p3$ can be transformed into a distance that is expressed as a function of lengths of both segments and the angle. Unless otherwise stated, we assume that statements are in 3D. It is also assumed that constraint systems are well-constrained, meaning first that there is a finite number of solutions up to rigid body motions, and second, that in some open neighborhood of the parameter values, each solution is a continuous function of the parameter values. Thus parameters are assigned to non-critical values (or regular values). For instance, for a triangle specified by its three lengths, no length equals the sum of the two other lengths. Note that the triangle can still have some special property, like a right angle, or being equilateral: we do not impose the parameter values to be generic. Indeed, the zero distance is used later on to specify point–plane incidence.

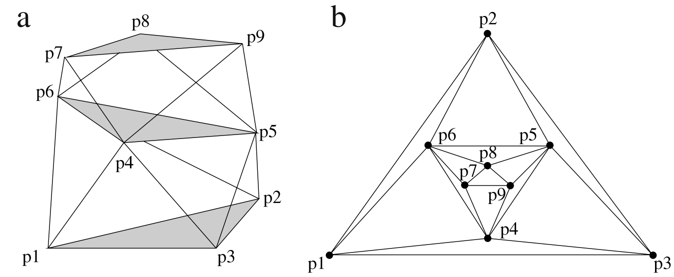

Fig. 1. Double stewart platform.

A geometric constraint system is usually associated with its corresponding *graphs of constraints*. Indeed, as constraints always involved two objects in our case, a constraint system can be represented by graph $G = \langle V, E\rangle$ where V is the set of vertices for entities and E is the set of pairwise edges for constraints. An edge between:

- two points mean the point–point distance constraint;
- two hyperplanes represent the angle constraint;
- a hyperplane h and a point p mean distance between p and its projection onto h (it is 0 if p lies on h).

In the rest of the paper, for the sake of simplicity, when there is no ambiguity, we confound both notions of GCS and its associated graph of constraints.

As an example, we consider a GCS coming from a mechanism constituted by two stacked Stewart platforms. Recall that a Stewart platform is an articulated system that can be represented as a triangle basis connected by six bars to another triangle whose position depends on the lengths of the bars. Fig. 1(a) shows a double Stewart platform and Fig. 1(b) is the corresponding graph. The GCS includes 9 points and 21 distance constraints.

2.2. Cayley–Menger determinants

Geometrically, the notion of determinant is related to volume. By classical operations on determinants, Cayley expressed the volume of a simplex in terms of distances. If coordinates of points are rows of matrices, multiplications of such matrices perform dot products that eliminate coordinates and make distances appear. Later, Menger studied its relevance to solve geometrical problems. Since then, the Cayley–Menger determinant is used to express or solve the geometric problem in a coordinate-free framework. Lu Yang [11] extended Cayley–Menger determinants to hyperplanes and hyperspheres.

Given a set of n points $\{p_1, \ldots, p_n\}$ in the Euclidean space of dimension d, the Cayley–Menger (CM for short) determinant of these points is defined by:

$$D(p_1, \ldots, p_{n-1}, p_n) = \begin{vmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & r_{1,2} & \cdots & r_{1,n} \\ 1 & r_{2,1} & 0 & \cdots & r_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & r_{n,1} & r_{n,2} & \cdots & 0 \end{vmatrix}$$

where $r_{i,j}$ is the squared distance between p_i and p_j. Then $D(p_1, \ldots, p_n)$ is the determinant of a symmetric matrix.

In dimension d, a set of $n \geq d + 2$ points specified by a Cayley–Menger determinant is embeddable in $\mathbb{R}^d$ if $D(p_1, \ldots, p_n) = 0$. In particular, in 3D, for 5 and 6 distinct points it comes:

$$D(p_1, p_2, p_3, p_4, p_5) = 0$$

Download English Version:

https://daneshyari.com/en/article/440088

Download Persian Version:

https://daneshyari.com/article/440088

Daneshyari.com