Computer-Aided Design 48 (2014) 42–52

Contents lists available at ScienceDirect

Computer-Aided Design

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

Defining tools to address over-constrained geometric problems in Computer Aided Design

Mireille Moinet [a,1], Guillaume Mandil [b,*], Philippe Serre [a]

[a] Laboratoire d'Ingénierie des Systèmes Mécaniques et des MAtériaux EA-2336, SUPMECA, 3 rue Fernand Hainaut, 93407, Saint-Ouen Cedex, France

[b] Laboratoire des Sciences pour la Conception, L'optimisation et la Production, Grenoble-INP / UJF-Grenoble 1 / CNRS, G-SCOP UMR5272 Grenoble, F-38031, France

HIGHLIGHTS

- Support for analyzing over-constraint sketches in CAD.
- Use of non-cartesian geometric modeling.
- Tool for decision making in constraint choice.
- Methodology to solve over-constraint problems.
- Possible application in collaborative environment.

ARTICLE INFO

Article history:
Received 19 November 2012
Accepted 16 November 2013

Keywords:
Geometric modeling
Non-cartesian
Over-constrained
Geometric constraints
GCSP
Specification

ABSTRACT

This paper proposes a new tool for decision support to address geometric over-constrained problems in Computer Aided Design (CAD). It concerns the declarative modeling of geometrical problems. The core of the coordinate free solver used to solve the Geometric Constraint Satisfaction Problem (GCSP) was developed previously by the authors. This research proposes a methodology based on Michelucci's witness method to determine whether the structure of the problem is over-constrained. In this case, the authors propose a tool for assisting the designer in solving the over-constrained problem by ensuring the consistency of the specifications. An application of the methodology and tool is presented in an academic example.

1. Introduction

In Computer Aided Design (CAD), the model is the computer representation of the object being designed. This geometric model is often named the Digital Mock-Up (DMU) and is now the core of CAD systems. In this study, the geometric model is the reference model.

There are essentially two strategies for building a digital mock-up in CAD systems: the procedural approach and the declarative approach [1]. This paper focuses on declarative approaches because they are often used in the 2D sketcher and 3D assembly workbenches of CAD systems [2] and because the authors have already worked on them in [3–6]. The declarative approach assumes that the designer first specifies a list of generic geometric objects and, second, a list of constraints between the objects defined previously. Then, a software application is used to solve all these constraints and build the virtual object. In order to obtain a valid object, it is necessary to ensure that all the specifications (generic objects and constraints) given by the user are consistent.

Moreover, the DMU is used in many simulations that cause the geometry to evolve due to changes made to the specifications required by the design team. It is therefore very important to maintain consistency in the statement of the problem. In general, problems that present inconsistencies are of two types: under-constrained problems and over-constrained problems. In this paper, we focus on over-constrained problems.

At present, when a designer comes across an over-constrained problem, no plans are available (at best a message is displayed on the screen). They must unravel it alone. Maintaining the consistency of the digital mock-up is even harder when several designers are involved in the design process. In this paper the authors propose to generate relations between the parameters of specifications in view to guiding users to ensure the consistency of the sets of parameters used. Therefore this paper will describe

* Corresponding author. Tel.: +33 4 76 57 47 76; fax: +33 4 76 57 46 95.
E-mail addresses: mireille.moinet@supmeca.fr (M. Moinet), guillaume.mandil@g-scop.eu (G. Mandil), philippe.serre@supmeca.fr (P. Serre).
1 Tel.: +33 1 49 45 25 47; fax: +33 1 49 45 29 29.

0010-4485/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cad.2013.11.002

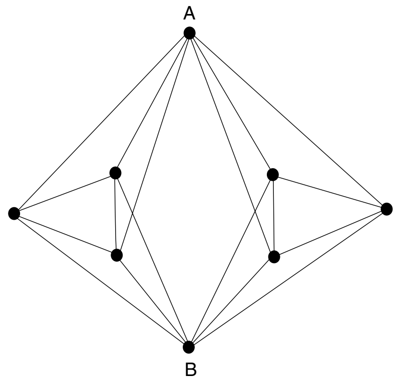

Fig. 1. Geometry of the double banana.

how to provide users with the elements necessary for maintaining the consistency of the geometry created. How is it possible to help designers to clearly specify their geometry? In the framework of collaboration, if a geometric problem is over-constrained, who arbitrates between the values of different specifications? This paper presents a solution for managing over-constrained geometric problems by giving users a tool for generating consistent sets of specifications.

The authors have already developed a conceptual model based on vectorizing the geometry for generic geometric objects. The associated solving strategy uses a coordinate-free representation. The major advantage of this approach is that it is unnecessary to take the cartesian reference frame into account for solving purposes. In 3D space, a geometric object is characterized by a Gram matrix that is positive semidefinite and has rank 3. Some elements of this matrix must have specific values imposed by the user, the others are unknowns. The geometric problem is solved by obtaining a Gram matrix **H** that meets the above conditions. This entails a matrix completion problem [7,8]. Indeed, in order to complete a partial matrix it is necessary to make specific choices of values for the unspecified entries.

The solution proposed by us is to find a transformation **T** that changes an initial object into a final object. The **G** Gram matrix characterizes the initial object. **H** Gram matrix characterizes the final object and is defined by the fundamental relationship (1), as explained in greater detail in [9]:

$$\mathbf{H} = \mathbf{T}\mathbf{G}\mathbf{T}^t. \tag{1}$$

More specifically, in our study, $\mathbf{T} = (\mathbf{I} + \mathbf{C}\mathbf{X})$, where **C** is the topological connection matrix, **X** the vertex perturbation matrix and **I** the Identity.

The authors make use of their previous work to address the problem of consistency. The proposed methodology and tool are applied to a case study shown in Fig. 1. It is a 3D bar structure called "double banana". The lengths of the 18 bars are specified, it can be seen that this system is not rigid because each banana can rotate about the axis defined by the two end points A and B connecting them.

We begin in Section 2 by giving a series of tools used to describe the geometric problem. Section 3 describes the coordinate free formulation applied to this GCSP[2] and presents the solving method. In Section 4, a method for analyzing whether the geometric problem is overconstrained or not, is described. Following this, if the problem is declared over-constrained, we propose a method for assisting a design team to seek a consistent set of constraints. Finally, Section 5 uses the example of the double banana to illustrate the application of this methodology.

2 Geometric Constraint Satisfaction Problem.

Remark. The Einstein notation or Einstein summation convention is employed in this paper. This notation implies the summation over a set of indexed terms in a formula, thus achieving notational brevity.

2. A coordinate free-model for representing the geometry

This section presents a non-cartesian model that characterizes geometric objects. We recall here the basics of the method as described in [5]. The characteristic of this approach is that it is unnecessary to perform cartesian reference frame, which is a real benefit for the sketching tasks of CAD designers. The principle is that any geometry can be represented as points and vectors. These are the central elements of our modeling. In the following, we describe three models that fully describe the design geometry.

2.1. The topological model

The geometry is reduced to a skeleton composed of points and line segments. An incidence matrix **C**, establishes the relation between each point of an object and its edges. It is an $n \times m$ matrix, where n and m are the number of edges and vertices respectively, such that $C_i^j = -1$ if the edge e_i leaves vertex p_j, 1 if it enters vertex p_j and 0 otherwise. Fig. 2 gives an example of an incidence matrix. Edges are oriented arbitrarily.

2.2. The geometrical model

The geometrical object is closely related to the topological model. Indeed, a vector is associated with each edge. Therefore each vector is an oriented bipoint. Thus the geometrical model is represented by a list of vectors.

Given $\mathcal{V}$ a set of n non-normed vectors as $\mathcal{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)$. The Gram matrix is the mathematical tool used to represent this vectorial model. Thus $\mathbf{G}(\mathcal{V})$ is defined by: $G_{ij}(\mathcal{V}) = \langle \mathbf{v}_i, \mathbf{v}_j \rangle$ where $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ is the scalar product between vectors $\mathbf{v}_i$ and $\mathbf{v}_j$. This matrix is also noted **G** in the following.

The Gram matrix of a set of vectors is constructed for each representation of a sketch. This Gram matrix fully defines the metrics of the object. It should be noted that this modeling is independent of the Cartesian coordinate system since all the vectors are defined in relation to one another: the vectors are not represented by their Cartesian coordinates but by their relative scalar products. An advantage of this approach is the possibility of ensuring specification consistency by verifying the mathematical properties of the Gram matrix (symmetrical, positive-semidefinite, rank, etc.). For example, by calculating specific determinants, it is possible to know whether or not there is a solution to the problem (see [10]).

2.3. The specification model

For our purpose, it is assumed that a geometric problem is defined by a skeleton, totally defined by the topological and geometrical models, and by a list of geometric constraints. In this study, we only focus on the length of the vectors, called L, or the angle between two vectors called α. All the specifications are stocked in **S**. It is a partially filled Gram matrix.

Element S_{ii} is known if the user chooses a specific length for $\mathbf{v}_i$, as presented in Eq. (2).

$$L_i = \sqrt{S_{ii}}. \tag{2}$$

Element S_{ij} is defined if the user specifies the angle between $\mathbf{v}_i$ and $\mathbf{v}_j$. Thus,

$$\cos(\alpha_{ij}) = \frac{S_{ij}}{\sqrt{S_{ii}}\sqrt{S_{jj}}}. \tag{3}$$

These three models fully characterize the GCSP.