ARTICLE IN PRESS
Computer-Aided Design ▮ (▮▮▮▮) ▮▮▮–▮▮▮
Contents lists available at ScienceDirect
Computer-Aided Design
journal homepage: www.elsevier.com/locate/cad

Over-constraints detection and resolution in geometric equation systems[☆]

Hao Hu, Mathias Kleiner, Jean-Philippe Pernot [*]

Arts et Métiers ParisTech, LSIS Laboratory, UMR CNRS 7296, France

ARTICLE INFO

Keywords:
Free-form surface deformation
Redundant and conflicting constraints
Structural decomposition
Numerical analysis
Decision support
Design intent

ABSTRACT

This paper proposes an original decision-support approach to address over-constrained geometric configurations in Computer-Aided Design. It focuses particularly on the detection and resolution of redundant and conflicting constraints when deforming free-form surfaces made of NURBS patches. Based on a series of structural decompositions coupled with numerical analyses, the proposed approach handles both linear and non-linear constraints. The structural decompositions are particularly efficient because of the local support property of NURBS. Since the result of this detection process is not unique, several criteria are introduced to drive the designer in identifying which constraints should be removed to minimize the impact on his/her original design intent. Thus, even if the kernel of the algorithm works on equations and variables, the decision is taken by considering the user-specified geometric constraints. The method is illustrated on academic and industrial examples realized with our prototype software.

1. Introduction

Nowadays, designers rely on 3D CAD software to model sophisticated shapes based on free-form curves and surfaces. In industrial design, this geometric modeling step is often encapsulated in a larger Product Development Process (PDP) which may incorporate preliminary design, reverse engineering, simulation as well as manufacturing steps wherein several actors interact [1]. Actually, the final shape of a product often results from a long and tedious optimization process which tries to satisfy the requirements associated to the different steps and actors of the PDP. Requirements can be seen as constraints. They are generally expressed either with equations, a function to be minimized, and/or using procedures [2]. The latter refers to the notion of black box constraints, not addressed in this paper, which focuses only on geometric constraints that can be expressed by linear or non-linear equations.

To satisfy the requirements, designers can act on variables associated to the different steps of the PDP. More specifically, in this paper, variables are supposed to be the parameters of the NURBS surfaces involved in the shape optimization process. To shape a free-form object defined by such surfaces, designers then have to specify the geometric constraints the object has to satisfy. For example, a patch has to go through a set of 3D points and satisfy to position constraints, the distance between two points located on a patch is fixed, two patches have to meet tangency constraints or higher-order continuity conditions, etc. Those geometric constraints give rise to a set of linear and non-linear equations linking the variables whose values have to be found. Due to the local support property of NURBS [3], the equations do not involve all the variables and some decompositions can be foreseen. Additionally, designers may express involuntarily several times the same requirements using different constraints thus leading to redundant equations. But, the designers may also involuntarily generate conflicting equations and may have to face over-constrained and unsatisfiable configurations.

Sometimes, over-constrained configurations can be solved by inserting extra degrees of freedom (DoFs) with the Boehm's knot insertion algorithm. As a consequence, many control points are added in areas where not so many DoFs are necessary [4]. This uncontrolled increase of the DoFs impacts the overall quality of the final surfaces which become more difficult to manipulate than the initial ones. Furthermore, some structural over-constraints cannot disappear following this strategy and dedicated decision-support approaches have to be developed to identify and manage over-constrained configurations.

Unlike advanced 2D sketchers available in most commercial CAD software, and which can interactively identify the over-constraints during the sketching process, it is not yet completely possible to pre-analyze the status of 3D NURBS-based equation systems before submitting them to a solver. Thus, there is a need for developing a new approach for the detection and resolution of redundant and conflicting constraints in NURBS-based equation systems. This corresponds to the identification

☆ This paper has been recommended for acceptance by C.L. Wang and Y. Chen.

* Corresponding author.
E-mail address: jean-philippe.pernot@ensam.eu (J.-P. Pernot).

http://dx.doi.org/10.1016/j.cad.2017.05.019

and treatment of over-constrained, well-constrained and under-constrained parts. In this paper, the treatment corresponds to the removal of constraints before solving. Once the constraints removed, the equation system often becomes under-constrained and the designer also has to add a requirement by mean of a function to be minimized so as to solve and find the values of the unknowns. This aspect is not part of the proposed approach but it will be discussed when introducing the results in which a particular functional is minimized.

Removing user-specified constraints is a perious step as the result do not fully satisfies what the designers have specified. Thus, it is not only important to develop an approach that is able to remove over-constraints, but also desirable to develop decision-support mechanisms which can help the designers identifying and removing the right constraints, i.e. the ones which preserve as much as possible the initial design intent.

This work contribution is to address these two difficult issues by proposing an original decision-support approach to manage over-constrained geometric configurations when deforming free-form surfaces. The algorithm handles linear as well as non-linear equations and exploits the local support property of NURBS. Based on a series of structural decompositions coupled with numerical analyses, the method detects and treats redundant as well as conflicting constraints. Since the result of this detection process is not unique, several criteria are introduced to drive the designer in identifying which constraints should be removed to minimize the impact on his/her original design intent. Thus, even if the kernel of the algorithm works on equations and variables, the decision is taken by considering the geometric constraints specified by the user at a high level.

The paper is organized as follows. Section 2 introduces the background and reviews the related works. Section 3 introduces the framework of our algorithm, details the principles and characteristics of its different steps and proposes criteria for evaluating its results. The proposed approach is then validated on both academic and industrial examples which are described in Section 4. Finally, Section 5 concludes this paper by discussing the main contributions as well as future work.

2. Background and related work

This section introduces how designers can specify their requirements within an optimization problem. It also analyses the existing methods used to detect structural or numerical over-constraints.

2.1. Modeling multiple requirements in an optimization problem

During the last decades, many deformation techniques have been proposed and it is not the purpose of this paper to detail all of them. Most of the time, when speaking of deformation techniques working on NURBS curves and surfaces, the goal is to find the position X of some control points so as to satisfy user-specified constraints which can be translated in a set of linear and/or non-linear equations $F(X) = 0$. Since the problem is often globally under-constrained, i.e. there are less equations than unknown variables, an objective function $G(X)$ also has to be minimized. As a consequence, the deformation of free-form shapes often results from the resolution of an optimization problem:

$$\begin{cases} F(X) = 0 \\ \min G(X) \end{cases} \tag{1}$$

For some particular applications, the optimization problem can also consider that the degrees, the knot sequences or the weights of the NURBS are unknown. However, in this paper, only the position of the control points are considered unknown. Depending on the approach, different objective functions can be adopted but they often look like an energy function which may rely on mechanical or physical models. The constraints toolbox can also contain more or less sophisticated constraints with more or less intuitive mechanisms to specify them.

Thinking to the PDP as well as to the needs for generating shapes which satisfy multiple requirements, one can notice that designers have access to three main parameters to specify their requirements and associated design intent within an optimization problem. They can effectively act on the unknowns X to decide which control points are fixed and which ones can move. In this way, they specify the parts of the initial shape which should not be affected by the deformation. Of course, designers can make use of the constraints toolbox to specify the equations $F(X) = 0$ to be satisfied. Finally, designers can also specify some of their requirements through the function $G(X)$ to be minimized. For example, they can decide to preserve or not the original shape while minimizing an energy function characterizing the shape deformation.

However, most of the existing free-form shape deformation techniques do consider that the problem resulting from the set of equations $F(X) = 0$ is under-constrained [5,6] and few attention has been paid to the analysis and processing of possible over-constraints. This paper proposes an approach to detect conflicting and redundant equations, and to help the designer in solving those issues by simply removing some constraints. However, Sections 3.4 and 4 discuss the possibility to fix more or less control points and thus modify the unknown vector X, as well as the possibility to modify the overall deformation behavior through the customization of the objective function $G(X)$ to be minimized.

2.2. Geometric over-constraints

Geometric over-constraints are classified into structural and numerical over-constraints [7]. Structural over-constraints can be detected from an analysis of the DoFs, at the level of either the geometry or the equations. Numerical over-constraints are usually determined from an analysis of the solvability of the equations system. Since our approach is based on equations, both aspects are to be defined.

2.2.1. Structural over-constraints

Jermann et al. give a general definition of structurally over-constrained, well-constrained and under-constrained equation systems at a rather macro level and considering the dimension of the space [8]. This definition has been here adapted to system of equations where the system is expected to be fixed with respect to a global coordinate system.

Definition 1. The degree of freedom $DoF(v)$ of a geometric entity v is the number of independent parameters that must be set to determine its position and orientation. For example, in 2D space, it is equal to 2 for points and lines. For a geometric constraints system G with a set V of geometries, the degree of freedom of all the geometries is $DoFs = \sum_{v \in V} DoF(v)$.

Definition 2. The degree of freedom $DoC(e)$ of a geometric constraint e is the number of independent equations needed to represent it. For instance, distance constraints have one DoC in 2D and 3D. For a geometric constraints system G with a set E of constraints, the degree of freedom of all the constraints is $DoCs = \sum_{e \in E} DoC(e)$.

Definition 3. A geometric constraints system G is *structurally well-constrained* if G satisfies $DoCs = DoFs$ and if all the subsystems after decomposition satisfy $DoCs \leq DoFs$.

Please cite this article in press as: Hu H., et al. Over-constraints detection and resolution in geometric equation systems. Computer-Aided Design (2017), http://dx.doi.org/10.1016/j.cad.2017.05.019.