Computer-Aided Design 75–76 (2016) 1–12

Contents lists available at ScienceDirect

Computer-Aided Design

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

Variational geometric modeling with black box constraints and DAGs

Gilles Gouaty [a,b,c], Lincong Fang [c,d], Dominique Michelucci [c,*], Marc Daniel [b], Jean-Philippe Pernot [a], Romain Raffin [b], Sandrine Lanquetin [c], Marc Neveu [c]

[a] *Arts et Métiers ParisTech, LSIS Laboratory UMR CNRS 7296, France*

[b] *Aix-Marseille University, LSIS Laboratory UMR CNRS 7296, France*

[c] *LE2I UMR6306, CNRS, Arts et Métiers, Bourgogne Franche-Comté University, Dijon, France*

[d] *School of Information Technology, Zhejiang University of Finance & Economics, Hangzhou, China*

ARTICLE INFO

Article history:
Received 19 July 2015
Accepted 5 February 2016

Keywords:
Black box constraints
Variational geometric modeling
Direct search methods
First order methods
Constraints
DAG

ABSTRACT

CAD modelers enable designers to construct complex 3D shapes with high-level B-Rep operators. This avoids the burden of low level geometric manipulations. However a gap still exists between the shape that the designers have in mind and the way they have to decompose it into a sequence of modeling steps. To bridge this gap, Variational Modeling enables designers to specify constraints the shape must respect. The constraints are converted into an explicit system of mathematical equations (potentially with some inequalities) which the modeler numerically solves. However, most of available programs are 2D sketchers, basically because in higher dimension some constraints may have complex mathematical expressions. This paper introduces a new approach to sketch constrained 3D shapes. The main idea is to replace explicit systems of mathematical equations with (mainly) Computer Graphics routines considered as Black Box Constraints. The obvious difficulty is that the arguments of all routines must have known numerical values. The paper shows how to solve this issue, *i.e.*, how to solve and optimize without equations. The feasibility and promises of this approach are illustrated with the developed DECO (Deformation by Constraints) prototype.

1. Introduction

Industrial CAD software rely on an incremental B-Rep (Boundary Representation) modeling paradigm where volume modeling is performed iteratively through high-level operators [1]. At a lower description level, those modeling operators are based on Euler operators acting directly on the faces, edges and vertices of B-Rep models. In this way, designers do not manipulate low-level geometric entities, but rather manipulate so-called structural and detail features to shape directly the CAD models.

However, even if CAD modelers provide operators (*e.g.*, pad, pocket, shaft, groove, hole, fillet) to get rid of the direct use and manipulation of canonical surfaces and NURBS [2], working with a CAD modeler is almost procedural and requires a lot of intermediate operations to obtain the desired shape of an object. Using such a procedural approach, designers have to make a mental gymnastic to break down the object body into several basic shapes linked to the different operators of the CAD software. Thus, even if a feature-based approach is used [3], modeling a complex shape still requires a lot of operations. This is even truer when dealing with free form objects for which the notion of free form features does not correspond to current industrial practices.

Clearly, an approach closer to the designers' way of thinking is missing and there is still a gap between the shapes designers have in mind and the tools and operators provided to model them. Various approaches have been introduced to bridge this gap: parametric modeling, feature-based modeling and variational modeling approaches.

In parametric modeling, or parametric and feature-based modeling [4], a shape is defined as a function $F(U)$, where F is some function, and U are its parameters. Designers specify with some interactive graphical interface the function F as well as the values of U parameters. Parameters U are geometric variables (e.g. lengths, angles, tolerances, Cartesian coordinates) or material properties (e.g. density, strength, cost). When the values of some parameters in U are modified during the design process, the shape is automatically updated while re-computing $F(U)$. Most of the time, only a part of $F(U)$ is re-computed, using a dependence analysis.

[*] Corresponding author.
E-mail address: dominique.michelucci@u-bourgogne.fr (D. Michelucci).

http://dx.doi.org/10.1016/j.cad.2016.02.002
0010-4485/© 2016 Elsevier Ltd. All rights reserved.

Variational geometric modeling [4] goes a step further. Designers specify constraints the shape $F(U)$ must satisfy as well as the unknown parameters. Then, a numerical solver tries to satisfy the constraints while computing values of unknown parameters of U. When defining a 3D shape, the constraints are often geometric constraints, which relate to different geometric primitives or features. For example, they can be distances or angles between (special points or axes of) geometric primitives or features, incidence or tangency relations between parts of two geometric primitives or features. In this case, their formal expression is simple and can be easily computed. It leads to a system of equations, most of the time algebraic. Numerous combinatorial or numerical methods [5] were proposed first to detect the under-, over-, and well-constrained parts of this kind of systems, and second, for well-constrained systems (which have as many independent equations as unknowns, and have a finite number of solutions) to decompose them into irreducible subsystems and to assemble their solutions.

In practice, numerical methods like Newton iterations, damped Newton or homotopy are used to solve irreducible subsystems and to assemble the partial solutions. The numerical solver typically starts from the previous values of U, read on some interactively provided sketch, or on the previous state of an iteratively edited shape.

Sometimes, some objective functions $G(U)$ must be optimized: for instance a cost, a weight or an energy should be minimized.

- On one hand, if the constraints system is well-constrained, there is a finite set of solutions and the best one, or a good enough one for the sake of computability, must be selected. This discrete problem is combinatorial and can be hard to solve (*e.g.,* Travelling Salesman Problem).
- On the other hand, if the system is under-constrained, it admits a continuum of feasible solutions and, under the usual mild assumptions, a finite number of parameters values U which satisfy both the constraints and some KKT (Karush–Kuhn–Tucker) or FJ (Fritz-John) conditions for local optimality.

The key feature of current variational geometric modeling approaches is that equations are available, and can be represented with tree-like data structures called DAGs (Directed Acyclic Graphs) in numerical analysis, computational geometry and computer algebra, or SLPs (Straight Line Programs) in dynamic geometry. The main advantage of DAGs is to permit to automatically compute the derivatives and Hessians. It is also possible to substitute parameters at the leaves of a DAG with other DAGs, to convert a given DAG into the corresponding polynomial (a list of monomials) or rational function. This is used to numerically evaluate a given DAG with many arithmetics (floating-point, intervals, exact arithmetics) for given numerical values of U. Like this it is possible to study which nodes in a given DAG depend on which parameters, and thus to update efficiently the value of a DAG when some parameters values in U are changed. To summarize this feature, we say that these DAGs are white box DAGs, or white DAGs.

On the contrary, a DAG is called a black box DAG or a black DAG, and a constraint is called a black box constraint or a black constraint, when the corresponding equation, or system of equations, is not available, or is not computable in practice. In this case, it is only possible to evaluate the corresponding DAG for given numerical values of parameters U, and to approximate the gradient with finite differences. There is no guarantee that the underlying function is continuous or smooth everywhere.

To illustrate differences between white and black DAGs, imagine we need a point x inside a given shape s and closest to a given point p. Then x is the solution of the constrained optimization problem $x = \operatorname{argmin}_{x\in s} \|x-p\|^2$, where, with classical Variational Modeling, the condition $x \in s$ must be expressed as a system of mathematical equations. Clearly, if s is a car or a building, it is just infeasible. Actually it is also infeasible for simple shapes, as soon as they involve nested geometric operations (*e.g.,* rounding, blending, Boolean operations, optimizations). On the other hand, computer graphics methods routinely solve this problem $x = \operatorname{argmin}_{x\in s} \|x-p\|^2$ using the routine closestPt(p, s) which does not rely on systems of equations.

In this paper, we propose to use black box DAGs instead of white box DAGs for Variational Geometric Modeling of free form surfaces and subdivision surfaces. We present a prototype, called DECO (Deformation by Constraints), to show the feasibility and promises of this approach. Our research is devoted to free form parametric surfaces as well as to subdivision surfaces due to the gap which currently exists between variational design and free form surface modeling and because subdivision surfaces are largely used in Computer Graphics and animation movies. Moreover subdivision surfaces do not have implicit or parametric equations and are generally manipulated as meshes approximating the limit surfaces. In addition to the specificity of modeling these types of surfaces, the interest of this novel approach is twofold. First, we no longer have to translate, when it is possible, the geometric constraints and the cost function into equations. Second, to express geometric constraints F and cost function G, we can use existing geometric procedures available in Computational Geometry, CAD/CAM and mathematical or numerical software. Assuming interoperability, functions or macros available in a geometric modeler software could be called. Thus this approach permits to easily extend the set of possible constraints. Certainly, with black DAGs, we can no longer use tools of Computer Algebra (for symbolic and exact computations of Jacobians, Hessians, resultants, Gröbner bases) since no equation is available. But we think that the advantages of our approach far outweigh its disadvantages. Additionally, it must be noticed that we aim at obtaining easily a first draft respecting given constraints in a preliminary stage of a design process and not necessarily final objects. The received models can then be exported to any CAD software for further developments.

The proposed approach is modular. It defines a formalism and framework regardless of the resolution method. For example, we use the GNU Scientific Library with the BFGS method. Furthermore, our approach is generic in the sense that one can consider later to treat other types of surfaces. Actually, we simply need to identify variables and define black boxes to make calculations on these surfaces.

The paper is organized as follows. Section 2 studies the related works comparing white and black DAGs. Our new modeler, allowing the specification of a set of constraints as well as an objective function to be minimized is introduced in Section 3. The associated solvers are presented in Section 4. Section 5 is devoted to examples. Finally, Section 6 concludes this paper and exhibits general issues raised by this approach.

2. Related work

Today's industrial CAD modelers are built on top of the well-known B-splines and NURBS paradigms to model free form surfaces [6,2]. Since the expected shapes are generally complex, the designer often has to decompose them into elementary shapes themselves subdivided into several surfaces. Each elementary surface is defined by means of a network of control points, weights and knot sequences. Most of the time, these surfaces must be trimmed to overcome the topological constraints of the mathematical models. Finally, the elementary surfaces are assembled together to produce a manifold solid, *i.e.,* a B-Rep representation expressing the relationships between the vertices, the edges and the faces of the topological model. Nevertheless, interacting at this low level is restricted to experts. Several attempts have been made to try to overcome the limits inherent to