Available online at www.sciencedirect.com

ScienceDirect

Computer-Aided Design 39 (2007) 1025–1036

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Constraint-based beautification and dimensioning of 3D polyhedral models reconstructed from 2D sketches

H.L. Zou, Y.T. Lee*

School of Mechanical and Aerospace Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore

Received 18 October 2006; accepted 22 August 2007

Abstract

3D models reconstructed from 2D sketches are inaccurate because of the inherent inaccuracies in the input and the reconstruction method. It is therefore necessary to “beautify” them before use in CAD systems. We present a method that detects geometric constraints, such as parallel and orthogonal faces, present in the reconstructed model and then selects a subset that constrains the object sufficiently and consistently. The subset selection algorithm first prioritizes the constraints depending on their type and then uses a novel method, based on quasi-Newton optimization, to detect and eliminate redundant and inconsistent constraints. The remaining constraints then define the dimensions of the model fully and consistently. Results from our implementation show that the method can beautify and dimension recovered 3D models correctly at acceptable speed.

Keywords: Sketching; 3D reconstruction; Beautification; Dimensioning; Constraints

1. Introduction

One active research area in computer-aided conceptual design concerns a designer sketching a design in 2D and a 3D model recovered automatically from the sketch [1–4]. Existing work is mainly restricted to polyhedral objects with planar faces only, which is also the case for this paper. Here all the edges of the object are present in the sketch and all the lines in the sketch represent edges. The sketch is first cleaned up to remove gaps at the vertices. A 3D model as perceived in human minds is then recovered, but it is inaccurate because the sketch, by its very nature, is inaccurate. The recovery method also affects the accuracy of the model. For example, the vertices in one face may not lie exactly on one plane and parallel faces may not be exactly parallel. Also, the size of the recovered model usually bears no resemblance to the actual size the designer has in mind, as a 2D sketch carries no dimensions. Improvement on the recovered model is therefore necessary to enforce proper geometric relationships and give it proper dimensions before the model can be used further. Fig. 1 illustrates the main stages.

It is neither practical nor desirable to require the user to provide the data for every entity – vertex, edge or face – present. One solution is to set up the constraints between the entities, and resolve the dimensions based on these constraints computationally. Many of the constraints can be established automatically, but there may be some that need to be given by the user. Clearly, it is desirable to minimize the user input on such a potentially tedious and error-prone task.

A similar problem arises in reverse engineering where a 3D model is recovered from a point cloud, although the inaccuracy here is less severe. Langbein [5] presented a method in which potential geometric constraints, such as parallelism, planarity and orthogonality, present in a 3D reverse-engineered model are found approximately at first, followed by a graph-based method to detect inconsistencies between these constraints. The result is a consistent subset of the potential constraints on the initial model. This subset, when resolved numerically, leads to an improvement in the geometry of the model. Langbein’s study shows that the graph-based method works well and fast in detecting inconsistencies.

One fast and simple step in the beautification of an object recovered from a sketch is to enforce planarity on its faces. Chen [6] solves this problem using a least-square method to adjust vertices of a face on to the same plane. But he did not

* Corresponding author. Tel.: +65 6790 5493; fax: +65 6791 1859.
E-mail addresses: zouhongliu@pmail.ntu.edu.sg (H.L. Zou), mytlee@ntu.edu.sg (Y.T. Lee).

0010-4485/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2007.08.002

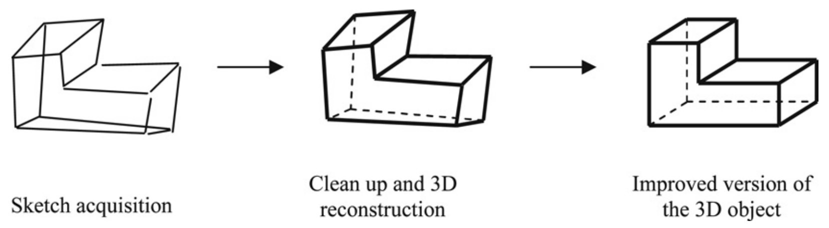
Sketch acquisition
Clean up and 3D reconstruction
Improved version of the 3D object

Fig. 1. Conceptual design by sketching.

take into account the effects of other constraints, such as faces being parallel.

Varley [4] beautified the 3D model reconstructed from a line drawing by determining the face normals and distances between faces separately using an optimization method.

Wilczkowiak et al. incorporated camera calibration and 3D reconstruction of 3D models built from images in computer vision [7]. The constraints, stored in a graph, are used to find a smaller set of input parameters and functions that yield all the parameters which are used in a subsequent refinement step. The approach is based on a dictionary of rules, called r-methods, that relate geometric entities through constraints, represented as a graph. Attempts are made iteratively to match the graph of an r-method to a subgraph of the constraint graph of the object, in which the entities are the nodes and the constraints are the edges. Then a reverse sequence of the r-methods identified is executed to solve the constraints. Implicit constraints are not considered because the system cannot contain redundancies.

The approaches described above beautify a 3D model, but the improved model carries no proper dimension. Martínez and Félez presented a constraint-based solver to provide a completely dimensioned 2D part [8]. Their method establishes the constraints from two or three sketches of different views of a model, and chooses a set of independent constraints of the system by determining if the system is over-constrained. The constraints form a system of equations, the Jacobian of which, when solved, reveals the redundant constraints. The work does not deal with dimensioning a 3D model.

A set of constraints needs to be sufficient to describe a model completely. There can be over-constraint or under-constraint, and there may also be redundancies, which include structural and numerical redundancies. A structural redundancy over-constrains the system. For instance, $f(x_1, x_2) = 0$, which constrains two variables x_1 and x_2 in a system, will lead to structural redundancy if two other constraints $g_1(x_1, x_2) = 0$ and $g_2(x_1, x_2) = 0$ are also present, since the values of x_1 and x_2 are implicitly determined already by g_1 and g_2. The problem can be rectified by discarding one of the constraints. A system can be numerically inconsistent or redundant. For example, two constraints expressed by $x_1 + x_2 = 1$ and $x_1 + x_2 = 0$ are inconsistent and one of $x_1 + x_2 = 1$ and $2x_1 + 2x_2 = 2$ is redundant.

Buchanan [9] determined whether a system of equations is inconsistent by using the Gröbner basis. Gao [10] gave a complete method for deciding whether the constraints in a set are independent and whether they are numerically inconsistent based on Wu–Ritt's [11] decomposition algorithm. Despite their advantages, both the Gröbner basis and the Wu–Ritt method require exponential time complexity. It is not uncommon for them to take tens of minutes or even hours, which is not acceptable in a real time interactive system.

Numerous researchers have addressed the problem of structural redundancies. In 2D, the triangle decomposition method was used by some researchers in geometric constraint solving [10,12–15]; their methods for identifying over-constrained subgraphs were based on triangle decomposition too. For example, Fudos's method [12] concludes that if two well-constrained clusters share more than one geometric element, then over-constraint exists. But the method can be used only within a limited domain and cannot be applied in 3D. Latham [16] proposed a method based on the maximum b-matching algorithm to decompose a constraint problem into a sequence of solvable subgraphs. The crux of this algorithm is the detection and correction of over-constraints and under-constraints. Maximum b-matching is a special case in generalized maximum matching (MM), of which Hoffmann stated [17], "the MM method may or may not detect over-constrained subgraphs, depending on the initial choice of vertices for reducing weights". He proposed a method MM1 to correct this drawback, but did not deal with 3D cases. Li's method [18] can detect over-constraint in 3D, but all the entities in the constraint graph must have six degrees of freedom. So his algorithm cannot be used in cases where the entities are points, lines and planes, which do not have six degrees of freedom. Recently, Langbein [5] proposed a method of identifying over-constraint based on Kramer's degree of freedom analysis [19] and Li's method of analyzing dependencies between geometric objects [18]. Jermann described a new concept of extensive structure rigidity [20,21], which is useful and is used in our algorithm.

There are limitations in redundant constraint detection using graph-based methods. Graphs have been used to detect structurally redundant constraints without solving the constraint system [5,16,18,22,23]. However, some constraints are implicit; they can be inferred from a set of constraints in a graph, but are not explicitly represented. Graph-based methods therefore cannot be used to detect all redundant constraints. We demonstrate it using two examples.

Fig. 2 illustrates Pappus's Theorem in 2D: If A, B, and C are three points on one line, D, E, and F are three points on another line, and AE meets BD at X, AF meets CD at Y, and BF meets CE at Z, then the three points X, Y, and Z are collinear [24]. The collinear constraint can be inferred from the given set of constraints. The constraint graph of the theorem, which includes all the points, lines, intersections and collinear constraints, is an under-constrained system, and there are no