Computers & Graphics 54 (2016) 127–134

Contents lists available at ScienceDirect

Computers & Graphics

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

Special Issue on CAD/Graphics 2015

Making and animating transformable 3D models

Yi-Jheng Huang, Shu-Yuan Chan, Wen-Chieh Lin, Shan-Yu Chuang

Department of Computer Science, National Chiao Tung University, 1001 University Rd., Hsinchu City 300, Taiwan, ROC

ARTICLE INFO

Article history:
Received 12 April 2015
Received in revised form
12 July 2015
Accepted 13 July 2015
Available online 31 July 2015

Keywords:
Transformable models
Segmentation
Character animation

ABSTRACT

Transformable models are 3D models whose shapes can be changed by rotating or translating their component parts. They have a variety of applications in our daily lives, being used in film props and sets, robots, furniture, tools, and toys. Successful transformable models, however, are challenging to create. In this paper, we present a new approach to designing and animating a transformable model, in which a source model is optimally segmented based on a target model and skeleton provided by users, and the motion of transformation is mapped from the source to target models. Our experimental results indicate that our system can transform a 3D model plausibly.

1. Introduction

Transformable models are 3D models whose overall shapes can be changed merely by rotating or translating their component parts. A good example of this is provided by the robots in the movie *Transformers*, but transformable models also exist in our daily lives, in the form of foldable furniture, tools, and toys whose shapes can be altered to save space or enhance functionality. Nevertheless, the successful creation of a transformable model is not an easy task. Designers have to consider how to shape and arrange each component part so that the model can take on two very different shapes by only translating and rotating its parts. They also need to carefully plan a transformation process, including the transformation order of the component parts and the type of joints between them, to ensure that the parts do not collide with each other and cause damage or even the failure of the entire system.

In this paper, we present a system that can generate a transformable model and its associated transformation motion. Taking two inputs—a source model and a target skeleton, with the latter representing a user's desired figure—the system first adjusts the target skeleton and embeds it into the source model. Then, the simulated annealing (SA) is utilized to optimally segment the source model into parts. Finally, the transformation of these parts is animated based on the results of a two-level motion-planning process. The experimental results discussed below indicate that our system has considerable promise for assisting novice users to create transformable models easily, as well as providing professionals with feasible options for references.

The contributions of this paper can be summarized as follows: (1) We introduce the transformable-model problem to the fields of modeling and animation. (2) We propose an optimization-based segmentation approach to generate suitable parts for the construction of transformable models. (3) We propose a two-level motion-planning process to animate transformable models plausibly.

2. Related work

Making a transformable model is related to the geometric dissection problem that originated in ancient Greece [1]. Essentially, it is a segmentation problem. This section reviews related work including the dissection puzzles in mathematics and segmentation in computer graphics, as well as research that applies to 3D models in other applications.

Dissection puzzles: A dissection puzzle, also called a transformation puzzle, is a set of pieces that can be rearranged into two or more distinct geometric shapes. An excellent survey of which was provided by Frederickson [2,3], though study of such puzzles was undertaken by ancient Greek mathematicians more than two millennia ago [1]. The ancient Chinese also invented the tangram, a puzzle consisting of seven pieces [4], the objective of which is to use all the pieces to form different specific shapes.

Recently, research on geometric dissection has taken two main directions. The first is a search for optimal dissections that minimize the number of pieces. Cohen [5], for instance, studied economical dissections of a triangle into squares, and Akiyama et al. [6] proposed an efficient dissection method for dividing a square into pieces that can be rearranged into two or more smaller squares. The other focus of research has been hinged dissections [3], in which all the pieces are linked at certain points. Piano hinges [7] and twisted hinges [8] have also been studied. However, the above-named studies only solved 2D hinged-dissection problems, whereas the focus of the present work will be on 3D hinged dissections.

http://dx.doi.org/10.1016/j.cag.2015.07.014
0097-8493/

With regard to 3D, Zhou and Wang [9] proposed a method for creating geometric dissection puzzles. Taking two regular grid figures of equal area, they partitioned one into separate pieces and reassembled them to form the other figure. Later, Zhou et al. [10] improved upon this work such that the pieces remained linked together while transforming into the alternative shape. Lo et al. [11] introduced a system which generated a polyomino puzzle by parameterizing the input to a quad-based surface then tiling the surface with polyominoes. Interlocking puzzles were discussed in [12,13], in which an input mesh is segmented into pieces that interlock with each other.

The key differences between this paper and the above 3D puzzle studies are as follows. (1) Unlike prior researchers [9–13] whose methods limited the solution space to discrete regular grids, we link parts together and seek solutions in a continuous space. (2) With regard to research purposes, previous studies of interlocking puzzles [11–13] focused on cutting a model into a buildable, interlocking and maintainable puzzle, whereas the goal of our study is the building of a transformable model. (3) Our study takes into account the process of transformation, utilizing a rapidly-exploring random tree (RRT) method to generate transformation animation.

Mesh segmentation: The problem of 3D model segmentation has been extensively studied. Many methods have been proposed, including K-means [14], core extraction [15], primitive fitting [16], randomized cuts [17], and random walks [18]. Chen et al. [19] proposed a set of manually generated benchmarks for evaluating 3D segmentation. In contrast to traditional segmentation algorithms, which mainly consider the segmenting of mesh surfaces, we treat a 3D model as a solid object and cut it into blocks that are also 3D. Moreover, our goal is to generate parts that can roughly represent the figures before and after transformation.

Computational methods for recreation: There has been a great deal of recent work on applying 3D models to recreational purposes. Huang et al. [20] generated 3D mechanical collages by automatically assembling mechanical elements. Mitani and Suzuki [21] constructed papercraft toy models based on a particular mesh. (Papercraft puzzles are generated so that a 3D shape is converted into planar slices [22–24].) Kilian et al. [25] constructed an elegant surface via curved folding from a planar sheet. Mori and Igarashi [26] proposed a sketch system that helps users design plush toys. Bacher et al. [27] generated articulated deformable toys by estimating the joint positions of input meshes. Zhu et al. [28] presented a system that can plan an assembly mechanism to control the motion of a mechanical toy whose motion is specified by the designer; and Koo et al.'s [29] system assists users in generating prototypes of mechanical objects.

3. Approach

Fig. 1 is an overview of our system. The user provides a source model and a target skeleton as inputs, after which the system generates the output model and the motions of the transformation process in three stages: preprocessing, parts optimization, and transformation computation. We describe the first and third stages in this section, and the second stage in Section 4, below. We assume that all input models are manifold and symmetric (to a sagittal plane). This symmetry assumption saves computational time, in that only half of each model needs to be processed, because optimized segmentation and planned transformation can both be mirrored.

3.1. System input

Though the usual inputs of our system are a source model and a target skeleton, a user can also opt to provide a target model if she/he has a more specific shape in mind. Our system generates *parts* by segmenting the source model, and then rotates and/or translates the parts to create a *transformed model*. The source model is treated as the initial shape of a transformable model and the shape of the transformed model is determined by the target skeleton. Additionally, if a target model is given, our system will try to match the shape of the transformed model to that of the target model.

A target skeleton consisting of bones and joints not only depicts a transformed figure, but also indicates information on the connections between parts. There are two types of bones: bone and virtual bone. A bone has a corresponding part, while a virtual bone does not, as it merely denotes a connection relationship. We use virtual bones because they are more flexible for users assigning the orientation of parts. A joint may connect two parts, or more than two. Each joint has one translational degree of freedom (DOF) l and three rotational DOFs (r_x, r_y, r_z). We adopt one translational DOF because we assume the bones in our system to be stretched along the bone's orientation, i.e., they can be lengthened but not widened. It is a common setting in computer animation. However, if necessary, 3D translational DOFs also can be decomposed into three one DOF in our system. Also, we assume that the target model and its skeleton are both symmetrical. When creating the target skeleton, users are required to distinguish *central joints*, which are to be placed on the sagittal plane, from *symmetric joints*. They do not need to set the joint position precisely to make the target skeleton symmetric, however. Rather, the users just need to build a skeleton along the target model and our system will automatically create a symmetrical skeleton by adjusting joint positions during the preprocessing stage.

Guidelines for skeleton setting: To execute a design using our system, users should ideally provide a source model, a target model, and a skeleton rigged in the target model. The guidelines for users setting skeletons are as follows. First, the orientation of bones is important because our optimal segmentation (Q_1 term in Section 4) will produce a cylindrical part, which is corresponding to a bone, aligned with the orientation of the bone. Second, the computational time mainly depends on the number of parts, which is equal to the number of bones. As the number of virtual bones does not influence the computational time, users could

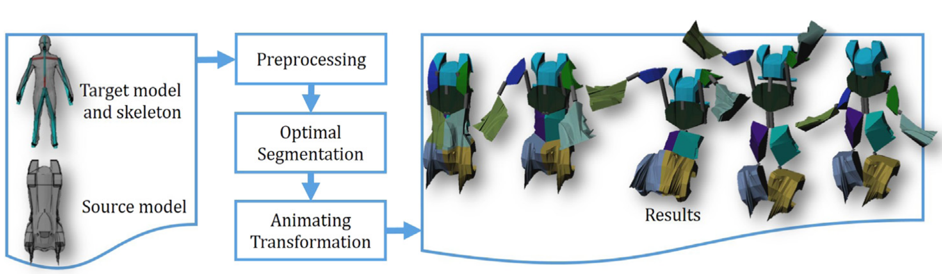

Fig. 1. System overview.