Computers & Graphics 35 (2011) 532–541

Contents lists available at ScienceDirect

Computers & Graphics

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

SMI 2011: Full Paper

Displaced subdivision surfaces of animated meshes

Hyunjun Lee [a], Minsu Ahn [b], Seungyong Lee [a,*]

[a] Pohang University of Science and Technology (POSTECH), Republic of Korea
[b] Samsung Advanced Institute of Technology (SAIT), Republic of Korea

ARTICLE INFO

Article history:
Received 13 December 2010
Received in revised form
14 March 2011
Accepted 14 March 2011
Available online 2 April 2011

Keywords:
Displaced subdivision surfaces
Animated meshes

ABSTRACT

This paper proposes a novel technique for converting a given animated mesh into a series of displaced subdivision surfaces. Instead of independently converting each mesh frame in the animated mesh, our technique produces displaced subdivision surfaces that share the *same* topology of the control mesh and a *single* displacement map. We first propose a conversion framework that enables sharing the same control mesh topology and a single displacement map among frames, and then present the details of the components in the framework. Each component is specifically designed to minimize the shape conversion errors that can be caused by enforcing a single displacement map. The resulting displaced subdivision surfaces have a compact representation, while reproducing the details of the original animated mesh. The representation can also be used for efficient rendering on modern graphics hardware that supports accelerated rendering of subdivision surfaces.

1. Introduction

Modern video games, immersive applications, and computer-based movies make extensive use of complex and highly detailed animated models to achieve realistic results. A variety of methods have been developed to efficiently represent and render animated models. Multiresolution subdivision surfaces provide an effective approach for handling animated models [7], and have been implemented as popular tools in the state-of-the-art modelers. They have also become a format of choice with the emergence of GPU-based tessellation methods in modern graphics hardware. However, although the design of new subdivision-based animated models is relatively simple, no effective method has been developed to extract them from existing animated meshes, which were acquired using 3D scanning of dynamic objects [1,18].

An animated mesh consists of a sequence of meshes, where the meshes have the same topology (vertex connectivity) but changing vertex positions. A displaced subdivision mesh consists of a control mesh and a displacement map, where the final shape is determined by adding the displacement map after recursive subdivision of the control mesh [12]. In this paper, we consider the conversion of an animated mesh into a series of displaced subdivision surfaces. Surprisingly, this subject has received little attention from research community despite potential advantages of the conversion. In addition to the rendering efficiency [24,2], displaced subdivision surfaces can be used as a compact representation of animated models which reduces storage requirement.

To understand the problem, it is worth pointing out that simply running the displaced subdivision surface technique for a *static* mesh [12] on *each* mesh of the animated mesh is not sufficient. The control meshes from different meshes would have different topologies and different displacement maps. Consequently, rendering and storage of the resulting displaced subdivision surfaces would be inefficient. This problem can be resolved if we enforce the resulting displaced subdivision surfaces to share the *same* topology of the control mesh and a *single* displacement map. This representation can save the amount of storage and allow effective rendering of animated meshes.

In this paper, we propose a conversion framework from an animated mesh into a series of displaced subdivision surfaces that share the same control mesh topology and a single displacement map. However, using a single displacement map may cause shape conversion errors in the resulting displaced subdivision surfaces. Different mesh frames may need different displacements and a single displacement map may not be enough to reproduce all the details of the meshes. Our framework is based on the previous work [12] for static meshes, but we extend it to handle animated meshes while specifically minimizing the conversion errors that can be caused by a single displacement map.

Our contributions can be summarized as follows.

- We propose a conversion framework for animated meshes to displaced subdivision surfaces. The output displaced subdivision surfaces share the *same* connectivity of control meshes and a *single* displacement map, which requires less storage and is easy to render with modern graphics hardware.

* Corresponding author.
E-mail addresses: crowlove@postech.ac.kr (H. Lee), minsu.ahn@samsung.com (M. Ahn), leesy@postech.ac.kr (S. Lee).

0097-8493/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2011.03.032

- We analyze the sources of shape conversion errors caused by using a single displacement map, and design each component of our framework to minimize the conversion errors. Experimental results demonstrate that our conversion framework generates small conversion errors.

2. Related Work

Displaced subdivision surfaces: Lee et al. introduced displaced subdivision surfaces [12]. They proposed a method to approximate a highly detailed mesh using a coarse mesh, called the *control mesh*, and a displacement map. A good approximation of the original mesh can be reconstructed from the data in two steps: A smooth mesh is defined as the limit surface of the control mesh by Loop subdivision [14], and vertices are then translated in the direction of their normals by the distance stored in the displacement map. Displaced subdivision surfaces are a powerful tool for modeling large and highly detailed objects. Furthermore, methods have recently been introduced to directly render displaced subdivision surfaces on GPU [26,15,4].

Simplification of animated meshes: To simplify deforming meshes, Mohr and Gliecher [16] proposed an extension of the QEM algorithm using the sum of quadrics of all frames in the animation. Kircher and Garland [10] used a progressive approach to simplify deforming meshes, and proposed a reclustering technique to improve simplification results. DeCoro and Rusinkiewicz [6] proposed a method for simplifying articulated meshes, and Landreneau and Schaefer [11] further improved the technique by optimizing vertex positions. In this paper, we propose a simplification technique for an animation mesh which considers the conversion error due to a single displacement map.

Global mesh parameterization: Lee et al. proposed a global mesh parameterization technique, which is called MAPS [13]. The original mesh is first partitioned into a set of triangular charts, and then the vertices of the original mesh are parameterized onto the charts. By regularly resampling the triangular charts, the original mesh can be remeshed and approximated. Several methods have been proposed to refine the parameterization, improving inter-chart continuity and smoothness [9,20,19]. While previous methods use local and global optimization techniques to optimize the parameterization, we use simple filtering to post-process the parameterization. We show our technique effectively smooths the parameterization while reflecting motion information of a given animated mesh.

3. Overview

In this paper, we assume an animated mesh is represented by a sequence of triangle meshes. From an input mesh sequence, we extract a series of control meshes having the same vertex connectivity which share a single displacement map. Our goal is to minimize the shape conversion errors of the resulting displaced subdivision meshes from the original mesh sequence. To achieve the goal, we first analyze the source of conversion errors, and then present our conversion framework with components specifically handling the errors.

3.1. Basic approach

For a static mesh, Lee et al. [12] minimized the shape conversion error by first making the control mesh similar to the original, and then using a displacement map to express remaining details. Our conversion framework is based on Lee et al.'s algorithm.

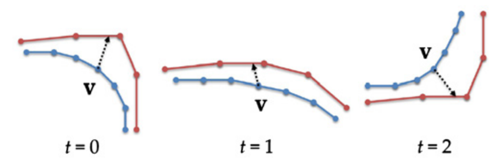

Fig. 1. One-dimensional illustration of motion-based errors. Red dots are vertices of original meshes and blue dots are vertices of subdivided meshes. A vertex **v** in the subdivided mesh is projected onto the original mesh. Displacements from three frames are different and they cannot be expressed with a single displacement map. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

However, we cannot directly apply this algorithm to generate displaced subdivision meshes when an input is a sequence of meshes, not a static mesh.

If we use only one displacement map shared among all frames, shape conversion errors will occur between the original meshes and the displaced subdivision meshes, because we cannot express all the details of all mesh frames with only one displacement map (see Fig. 1). We can divide the errors into two types:

Geometry-based errors: Differences between the input meshes and the subdivision surfaces of the control meshes. By minimizing these errors, we can reduce the displacements needed to recover the original shapes after subdivision.

Motion-based errors: Differences of displacements among all frames. To express displacements of all frames as one value, which will be saved in the single displacement map, we have to minimizing these errors.

We should minimize both geometry- and motion-based errors to accurately reproduce the original shapes in the input mesh sequence.

3.2. Overall process

We first define the notations. Let $\mathbf{M}_o$, $\mathbf{M}_c$, $\mathbf{M}_s$, and $\mathbf{M}_d$ be the original, control, subdivided, and resulting displaced subdivision mesh sequences, respectively. Let $\mathbf{D}$ be the shared single displacement map. A specific frame is referred by a parameter t. For example, $\mathbf{M}_o(t)$ and $\mathbf{M}_c(t)$ denote the original and control meshes at frame t, respectively.

For the displaced subdivision mesh obtained from a static mesh, Lee et al.'s algorithm [12] can be briefly described as follows. First, a simplification process is used to obtain an initial control mesh $\mathbf{M}_c$. The second stage optimizes the location of vertices in $\mathbf{M}_c$ in order to minimize the geometry-based error. Finally, a displacement map is computed to capture the remaining details.

Similar to Lee et al.'s algorithm, we use the geometry-based errors during the simplification and optimization processes. In addition, we propose motion-aware simplification and parameterization smoothing, which reflect motion information to reduce the required displacements for shape recovery. The following steps summarize the process of obtaining the displaced subdivision meshes from an input sequence of meshes.

1. We simplify the original mesh sequence $\mathbf{M}_o$ into a sequence of initial control meshes $\mathbf{M}_c$ by considering geometry- and motion-based errors, and build the initial mapping from $\mathbf{M}_o$ onto $\mathbf{M}_c$ (Section 4).