INTERACTIVE SHAPE DESIGN

Synthesis Lectures on Computer Graphics and Animation

Editor
Brian A. Barsky, *University of California, Berkeley*

Interactive Shape Design
Marie-Paule Cani, Takeo Igarashi and Geoff Wyvill
2008

Real-Time Massive Model Rendering
Sung-eui Yoon, Enrico Gobbetti, David Kasik, and Dinesh Manocha
2008

Virtual Crowds: Methods, Simulation and Control
Nuria Pelechano, Jan. Allbeck, and Norman I. Badler
2008

High Dynamic Range Video
Karol Myszkowski, Rafał Mantiuk, and Grzegorz Krawczyk
2008

GPU-Based Techniques For Global Illumination Effects
László Szirmay-Kalos, László Szécsi and Mateu Sbert
2008

High Dynamic Range Imaging Reconstruction
Asla Sa, Paulo Carvalho, and Luiz Velho IMPA, Brazil
2007

High Fidelity Haptic Rendering
Miguel A. Otaduy, Ming C. Lin
2006

A Blossoming Development of Splines
Stephen Mann
2006

© Springer Nature Switzerland AG 2022
Reprint of original edition © Morgan & Claypool 2008

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopy, recording, or any other except for brief quotations in printed reviews—without the prior permission of the publisher.

Interactive Shape Design

Marie-Paule Cani, Takeo Igarashi, and Geoff Wyvill

ISBN: 978-3-031-79533-6 paperback
ISBN: 978-3-031-79534-3 ebook

DOI 10.1007/978-3-031-79534-3

A Publication in the Springer series
SYNTHESIS LECTURES ON COMPUTER GRAPHICS AND ANIMATION #6

Lecture #6

Series Editor: Brian A. Barsky, University of California, Berkeley

Series ISSN: 1933-8996 print
Series ISSN: 1933-9003 electronic

INTERACTIVE SHAPE DESIGN

Marie-Paule Cani
Grenoble Institute of Technology
France

Takeo Igarashi
University of Tokyo
Japan

Geoff Wyvill
University of Otago
New Zealand

SYNTHESIS LECTURES ON COMPUTER GRAPHICS AND ANIMATION #6

ABSTRACT

Providing an intuitive modeling system, which would enable us to communicate about any free-form shape we have in mind at least as quickly as with real-world tools, is one of the main challenges of digital shape design. The user should ideally be able to create, deform, and progressively add details to a shape, without being aware of the underlying mathematical representation nor being tied by any constraint on the geometrical or topological nature of the model.

This book presents the field of interactive shape design from this perspective. Since interactively creating a shape builds on the humans ability of modeling by gesture, we note that the recent advances in interactive shape design can be classified as those that rely on sculpting as opposed to sketching metaphors. Our synthetic presentation of these strategies enables us to compare the different families of solutions, discuss open issues, and identify directions for future research.

KEYWORDS

Geometric modeling, Digital shape editing, Geometric design, Surface representations, Sculpting, Sketching

Contents

List of Figures xi

Acknowledgments xv

1. Introduction 1
1.1 Introduction 1
1.2 Designing Shapes in the Real World 2
1.2.1 Sculpting 2
1.2.2 Sketching 3
1.3 A Brief History of Shape Representations 4
1.3.1 Seventies: From Solid Primitives to Free-Form Surfaces 5
1.3.2 Eighties: Constructive Implicit Surfaces 7
1.3.3 Nineties: Subdivision and Multiresolution Surfaces 9
1.3.4 Recently: Meshes Versus Meshless Representations 10
1.4 Shape Representations versus Shape Design 12

2. Sculpting Metaphors 15
2.1 Background: Deforming Digital Shapes 15
2.1.1 Model-Based Methods 15
2.1.2 Space Deformations 16
2.1.3 Surface-Based Deformations 18
2.1.4 Deforming versus Sculpting a Shape 19
2.2 Volumetric Sculpting 19
2.2.1 Geometric Representation for Clay 20
2.2.2 Providing a Multiresolution, Unbounded Clay Model 21
2.2.3 Adding, Carving, and Smoothing Clay 23
2.2.4 User Interaction: Visual and Haptic Feedback 24
2.2.5 Limitation: Volumetric Clay Does Not Deform! 25
2.3 Physically based Virtual Clay 27
2.3.1 Physical Models for Clay 27
2.3.2 A Layered Model for Virtual Clay 29
2.3.2.1 Large-Scale Deformations 29

2.3.2.2 Mass Conservation Layer 30
2.3.2.3 Surface Tension 31
2.3.3 Interacting with Virtual Clay 31
2.4 Sculpting with Sweepers 33
2.4.1 Sweepers: Space Deformations Controlled by Gesture 33
2.4.2 Constant Volume Swirling Sweepers 34
2.4.3 Vector Field–Based Deformation 36
2.4.4 Results and Discussion 36
2.5 Conclusion: Where Shall we Stop Toward Virtual Clay? 37

3. Sketching Systems 39
3.1 Background 39
3.1.1 2D Depiction of 3D Shapes 39
3.1.2 Human Perception of the Third Dimension 41
3.2 Sketching Mesh Models 42
3.2.1 User Interface 42
3.2.1.1 Creating a New Object 43
3.2.1.2 Extrusion 43
3.2.1.3 Cutting 44
3.2.1.4 Smoothing 44
3.2.1.5 Transformation 45
3.2.2 Algorithms 45
3.2.3 Results and Discussions 47
3.3 Sketching Implicit Models 48
3.3.1 User Interface 49
3.3.2 Extraction of a Skeleton From 2D Sketch 51
3.3.3 Construction of a Convolution Surface 52
3.3.4 Progressive Blending and Rendering 53
3.3.5 Results and Discussion 54
3.4 Sketching Optimization Surfaces 55
3.4.1 User Interface 56
3.4.1.1 Sketching Tool 56
3.4.1.2 Deformation Tool 57
3.4.1.3 Rubbing Tool 58
3.4.1.4 Erasing Tool and Type Change Tool 59
3.4.2 Algorithm 60
3.4.3 Results and Discussion 62

3.5 Conclusion: What is the Best Representation for Sketch-Based Modeling? 63

4. Future directions: Modeling by Gesture 65

4.1 Modeling through Adding and Erasing Gestures 65

4.2 Enabling Deformation Gestures 66

4.3 Toward Fast and Easy Creation of Digital Content 67

Bibliography 69

Author Biography 81

List of figures

1.1 Shape design in the real world: these examples of clay modeling, wood carving, and sketching show the importance of the artist's hand gestures in the design process. 2

1.2 Constructive solid geometry (CSG) expresses shapes from a construction tree, which shows how they are progressively built out of solid primitives. 5

1.3 Tensor product, parametric surfaces built from a grid of control points, are not convenient for representing branching structures, for which either n-sided patches or smooth junctions between an arbitrary number of square patches need to be modeled. 7

1.4 Constructive implicit surfaces are defined as isosurfaces of a scalar field function. The last is typically a decreasing function of the distance to the skeleton (left and middle). The most standard ones are distance surfaces (top right), for which the contribution of skeletal elements are computed using closest points and then summed, generating bulges at junctions. Convolution surfaces (bottom right) avoid the problem by integrating the field contribution along the skeleton curves or surfaces. 7

1.5 Subdivision surfaces are procedurally defined through an iterative subdivision scheme to be applied to their control mesh. 9

1.6 Point-based, or meshless representations, rendered using oriented splats, have proved to be a good alternative to standard mesh representations for visualizing huge unstructured data sets. 10

1.7 Differential mesh representations enable us to change global shape attributes while preserving surface details. 11

2.1 Space deformations warp space, deforming all the embedded objects. 16

2.2 Distance fields (left) require global edits and are discontinuous in concave areas, while density fields (middle) are local. They can be stored in multiresolution grids that are refined according to the tool's shape, leading to the easy sculpting and storage of large to fine structures (right). 21

2.3 Data structures for the virtual unbounded grid, stored in three hash tables: nonempty cells, cells crossing the surface, and surface triangles. These three structures are locally updated after each user edit, so rendering is real-time. 22

2.4 Sculpting tools are either analytic implicit primitives (e.g. spheres and ellipsoids) or discrete implicit shapes sculpted within the application. 23

2.5 Using haptic feedback accelerates volumetric sculpting, since it eases the positioning of the tools. The figure on the left (from [88]) was sculpted in 3 hours using a simple mouse, while the one on the right (from [90]) was sculpted in 1 hour using haptic feedback. 25

2.6 A geometric solution for mimicking the local deformation produced by a rigid tool: matter is suppressed inside the tool and added in an outside ring, forming a bulge. 26

2.7 Global deformations such as bending and twisting are of a different nature from local ones, since they can propagate outside of any predefined boundary of the tool's influence. 26

2.8 Viscous fluids have been modeled using particles surrounded by an implicit surface (picture from [92]). 27

2.9 A layered model for virtual clay. Large-scale deformations produced by the first layer modeling plasticity (left). Local deformations generated by the second layer insure constant volume (middle). The third layer generates an internal rigidity which prevents clay from spreading over space (right). 29

2.10 Comparison between real clay and Dewaele's virtual clay. Local and global deformations at constant volume are adequately modeled. They can be controlled by an arbitrary number of tools. 32

2.11 Three possible interfaces for interacting with virtual clay: using a single tool, moved in 3D using a force feedback device (left); capturing the deformations the user applies on a real object serving as an avatar (middle); and controlling the gestures of a virtual hand interacting with the clay (right). 32

2.12 Sweepers use the user's translation gestures to wrap space in a foldover. Small steps used to make the deformation foldover free (left). The user positions a tool in intersection with a shape, the tool's region of influence being depicted in yellow (right): when applied and moved, the tool sweeps a part of the shape with it while keeping it smooth. 34

2.13 Constant volume swirls (left) are placed along rings to move matter along a translation vector defined by the user's gesture (center). In practice, the use of swirls is transparent to the user. He or she simply sweeps constant volume clay by gesture (right). 35

2.14 Examples of shapes modeled from a ball of clay in a few minutes, using swirling sweepers. 36

3.1 Line drawing of real-world objects. 40
3.2 Silhouette rendering in object space method (left) and image space method (right). 40
3.3 3D surface reconstructions from 2D contours. 41
3.4 Creation. 43
3.5 Extrusion. 44
3.6 Cutting. 44
3.7 Smoothing. 45
3.8 Transformation. (Red lines are reference stroke and blue lines are target strokes.) 45
3.9 Internal representation: (a) after creation, (b) after extrusion, and (c) after cutting. 46
3.10 Finding the skeleton: (a) initial 2D polygon, (b) result of CDT, (c) chordal axis, (d) fan triangles, (e) resulting spline, and (f) final triangulation. 46
3.11 Polygonal mesh construction: (a) before, (b) elevate skeletons, (c) elevate edges, and (d) sew elevated edges. 47
3.12 3D models designed by novice users. 48
3.13 Unintuitive extrusion results. 48
3.14 Progressive implicit modeling by sketching: simple implicit primitives drawn from different viewpoints are blended using a zoom-dependent sharpness (from [120]). 50
3.15 (a), (b) Creating a part. (c), (d), (e) Adding a part to an object. (f),(g),(h) Carving. (i),(j),(k) Thickness control (side view). 50
3.16 Algorithm overview: (a) user strokes, (b) identification of the object boundaries, (c) WDT transform, (d) iteratively thinning with preservation of the CMDs, (e) iteratively thinning without preservation of the CMDs, (f) sampling the segments, (g) sampling and triangulating the polygons, (h) final skeleton graph, and (i) final convolution surface. 52
3.17 Objects modeled with the system. The user took 2–5 minutes for modeling each object and used three to nine strokes, each of them modeling a different object part. 55
3.18 Sketching operations of FiberMesh, modifying geometry. 57
3.19 Sketching operations of FiberMesh, adding control curves. 58
3.20 Pulling a curve. The deformed curve segment is determined by how much of it the user peels off. 58
3.21 Rubbing (smoothing) a curve. 59

3.22 Erasing a control curve: (left) before erasing, (middle) immediately after erasing, and (right) after optimization. 59
3.23 Changing curve type: (left) before the change, (middle) immediately after the change, and (right) after optimization. 59
3.24 3D models designed using FiberMesh. 62

Acknowledgments

A work of this kind depends on the cooperation of many people, students, colleagues, and researchers who have published with us. We would like to specially thank Marc Alexa, Anca Alexe, Alexis Angelidis, Grégoire Aujay, Loïc Barthe, Renaud Blanch, David Bourguignon, Gilles Debunne, Mathieu Desbrun, Guillaume Dewaele, Eric Ferley, Jean-Dominique Gascuel, Laurent Grisoni, John F. Hughes, Shin HyoJong, Yuki Igarashi, Takashi Ijiri, Scott King, Paul Kry, Satoshi Matsuoka, Tomer Moscovich, Andrew Nielen, Makoto Okabe, Shigeru Owada, Adeline Pihuit, Olga Sorkine, Kenshi Takayama, Hidehiko Tanaka, Emmanuel Turquin, Nayuko Watanabe, Jamie Wither, Brian Wyvill, and the anonymous reviewers from Morgan & Claypool who made so many helpful suggestions.