Geometric Modeling
Project Ideas

University of Victoria | Computer Science

Final Project

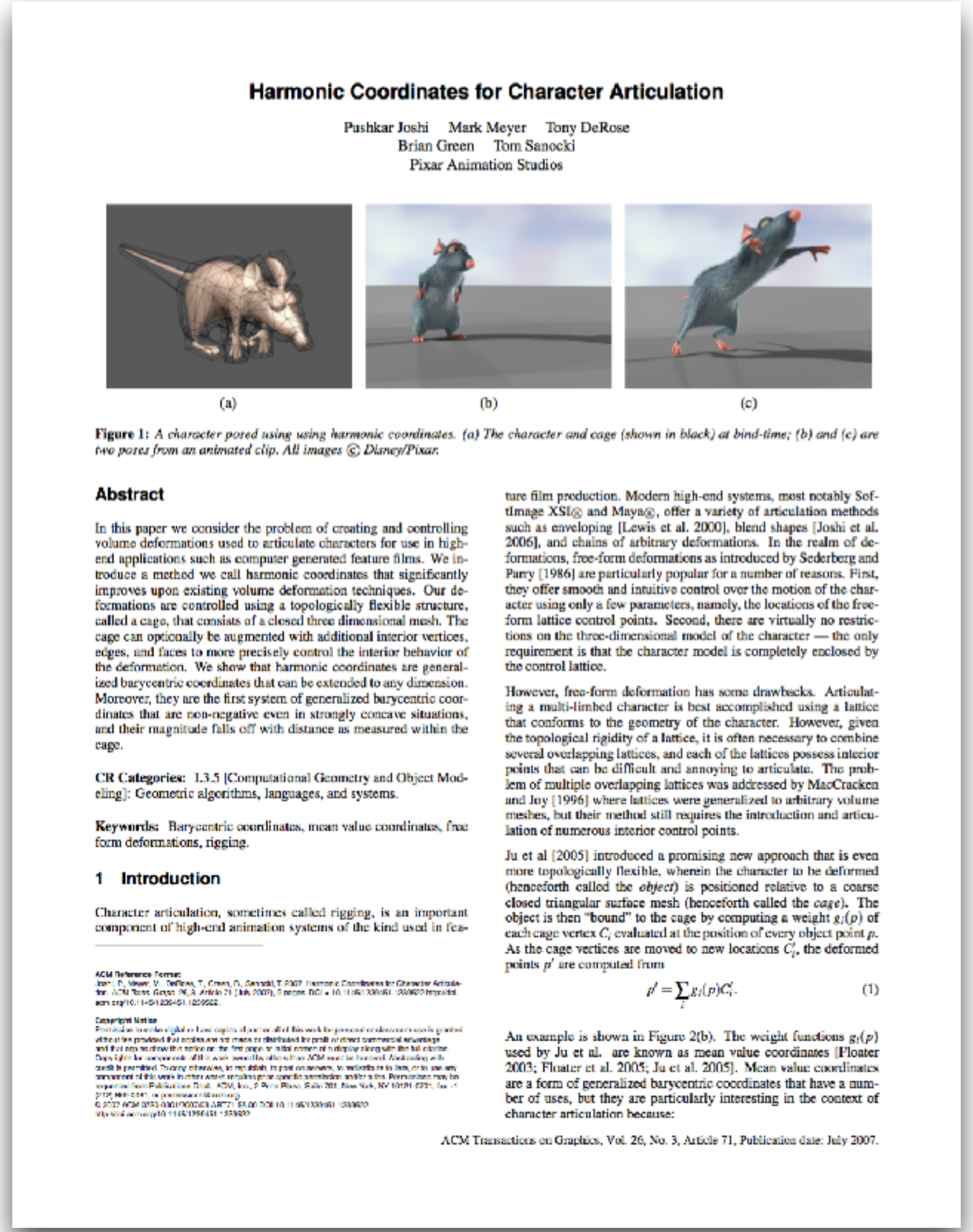
- Implement a research paper from scratch (no provided code)
- Using libigl is fine, other dependencies must be pre-approved

Project Ideas

- Choose from 5 **suggestions**
  - Harmonic Coordinates for Character Animation
  - Smooth Feature Lines on Surface Meshes
  - Interactive Geometry Remeshing
  - Diffusion Curves
  - As-Rigid-As-Possible Shape Interpolation
- … **or propose your own**

Project 1:
Harmonic Coordinates

Harmonic Coordinates



Harmonic Coordinates for Character Articulation

Pushkar Joshi Mark Meyer Tony DeRose
Brian Green Tom Sanocki
Pixar Animation Studios

(a) (b) (c)

Figure 1: *A character posed using using harmonic coordinates. (a) The character and cage (shown in black) at bind-time; (b) and (c) are two poses from an animated clip. All images © Disney/Pixar.*

Abstract

In this paper we consider the problem of creating and controlling volume deformations used to articulate characters for use in high-end applications such as computer generated feature films. We introduce a method we call harmonic coordinates that significantly improves upon existing volume deformation techniques. Our deformations are controlled using a topologically flexible structure, called a cage, that consists of a closed three dimensional mesh. The cage can optionally be augmented with additional interior vertices, edges, and faces to more precisely control the interior behavior of the deformation. We show that harmonic coordinates are generalized barycentric coordinates that can be extended to any dimension. Moreover, they are the first system of generalized barycentric coordinates that are non-negative even in strongly concave situations, and their magnitude falls off with distance as measured within the cage.

CR Categories: I.3.5 [Computational Geometry and Object Modeling]: Geometric algorithms, languages, and systems.

Keywords: Barycentric coordinates, mean value coordinates, free form deformations, rigging.

1 Introduction

Character articulation, sometimes called rigging, is an important component of high-end animation systems of the kind used in feature film production. Modern high-end systems, most notably SoftImage XSI® and Maya®, offer a variety of articulation methods such as enveloping [Lewis et al. 2000], blend shapes [Joshi et al. 2006], and chains of arbitrary deformations. In the realm of deformations, free-form deformations as introduced by Sederberg and Parry [1986] are particularly popular for a number of reasons. First, they offer smooth and intuitive control over the motion of the character using only a few parameters, namely, the locations of the free-form lattice control points. Second, there are virtually no restrictions on the three-dimensional model of the character — the only requirement is that the character model is completely enclosed by the control lattice.

However, free-form deformation has some drawbacks. Articulating a multi-limbed character is best accomplished using a lattice that conforms to the geometry of the character. However, given the topological rigidity of a lattice, it is often necessary to combine several overlapping lattices, and each of the lattices possess interior points that can be difficult and annoying to articulate. The problem of multiple overlapping lattices was addressed by MacCracken and Joy [1996] where lattices were generalized to arbitrary volume meshes, but their method still requires the introduction and articulation of numerous interior control points.

Ju et al [2005] introduced a promising new approach that is even more topologically flexible, wherein the character to be deformed (henceforth called the *object*) is positioned relative to a coarse closed triangular surface mesh (henceforth called the *cage*). The object is then "bound" to the cage by computing a weight $g_i(p)$ of each cage vertex C_i evaluated at the position of every object point p. As the cage vertices are moved to new locations C_i', the deformed points p' are computed from

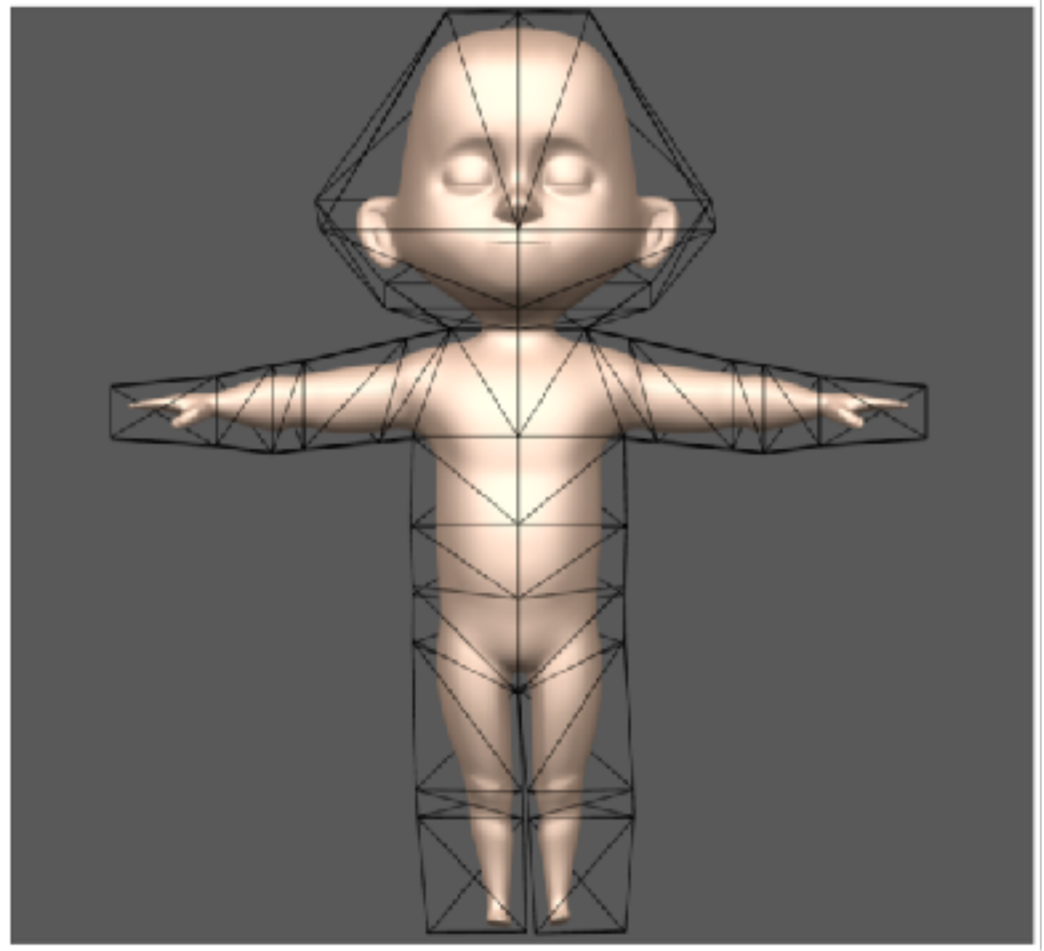
$$p' = \sum_i g_i(p) C_i'. \tag{1}$$

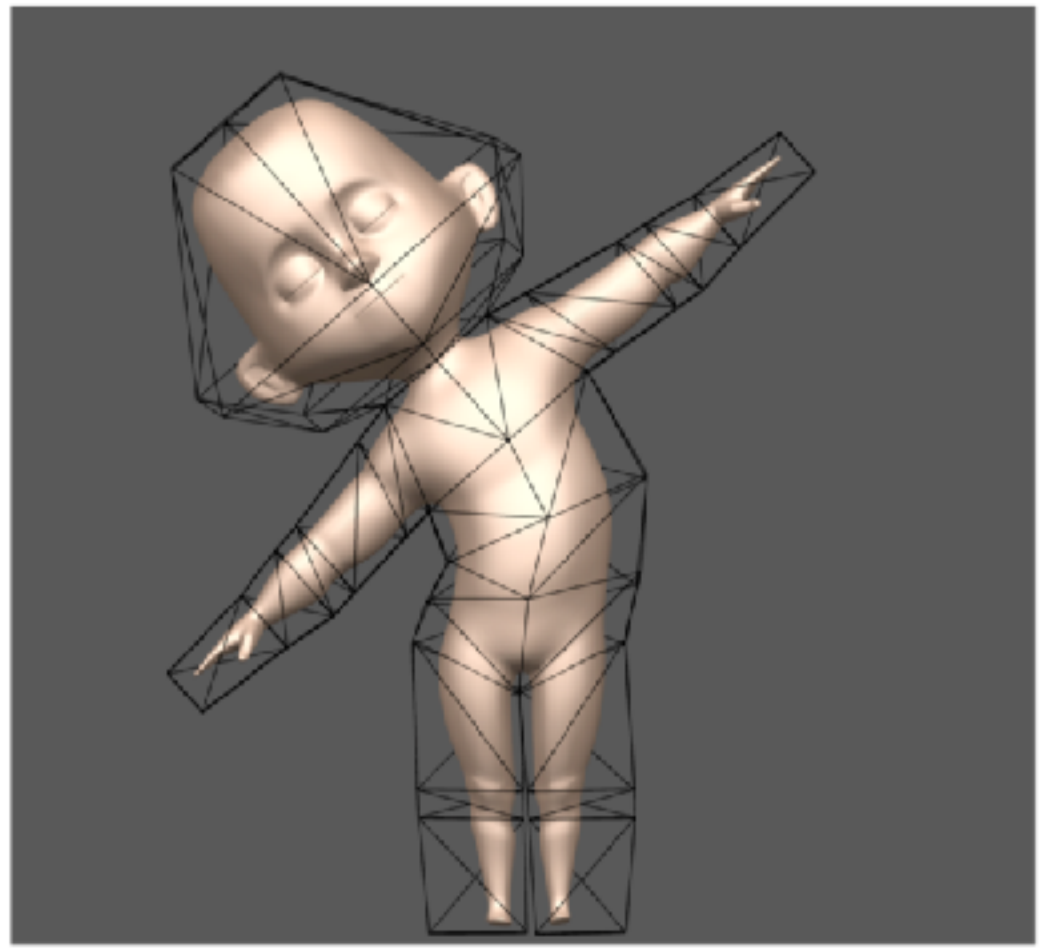
An example is shown in Figure 2(b). The weight functions $g_i(p)$ used by Ju et al. are known as mean value coordinates [Floater 2003; Floater et al. 2005; Ju et al. 2005]. Mean value coordinates are a form of generalized barycentric coordinates that have a number of uses, but they are particularly interesting in the context of character articulation because:

ACM Reference Format:

University of Victoria | Computer Science

Harmonic Coordinates

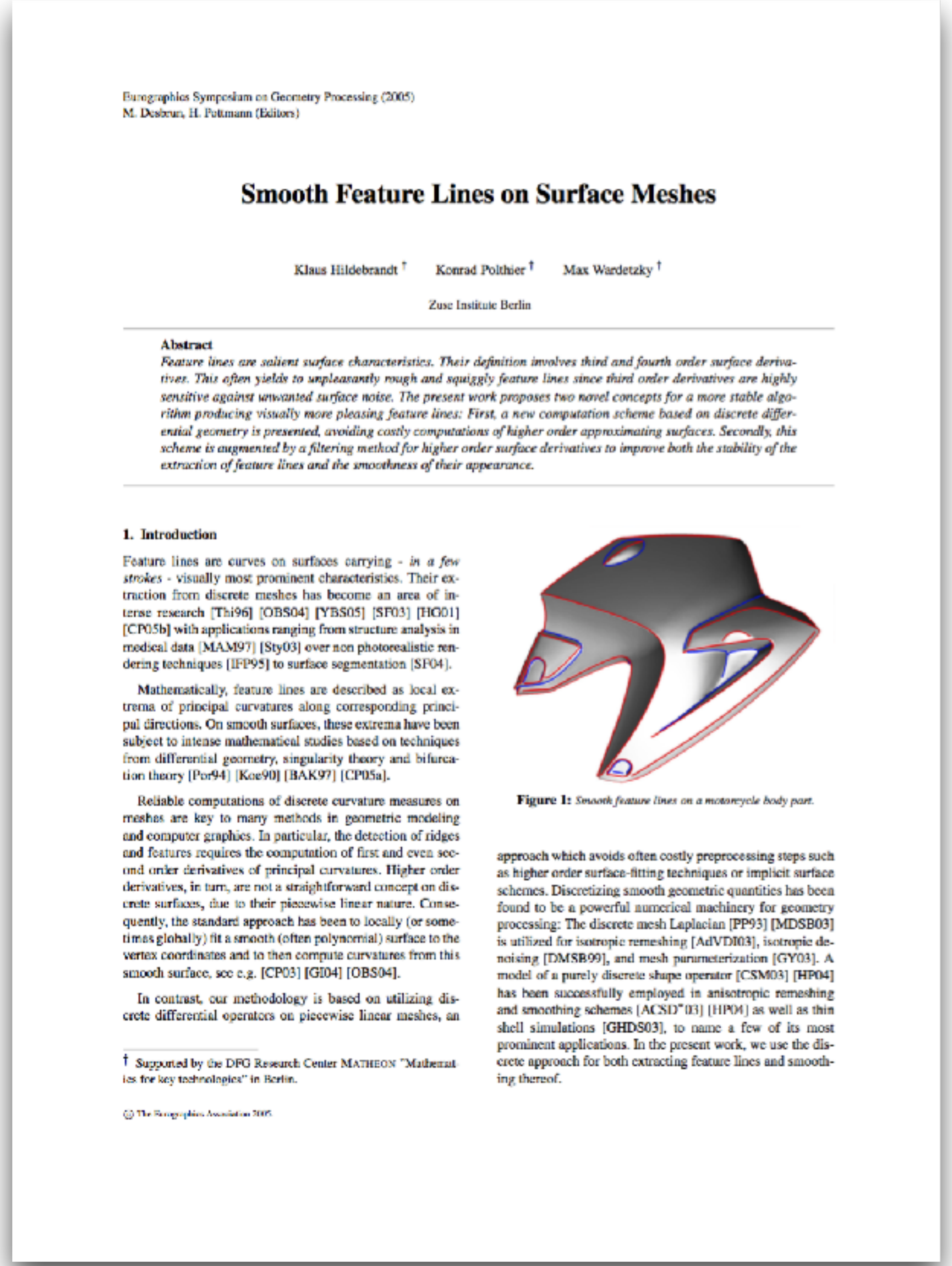




- Draw a cage around an image
- Construct "coordinate functions" for each cage vertex
  - Smooth functions over the mesh
  - 1 on a particular cage vertex, 0 on all others
  - Compute them as solutions to Laplace equation!
- Use these coordinate functions to interpolate a cage deformation

Project 2:
Feature Lines

Smooth Feature Lines on Surface Meshes



Eurographics Symposium on Geometry Processing (2005)
M. Desbrun, H. Pottmann (Editors)

Smooth Feature Lines on Surface Meshes

Klaus Hildebrandt [†] Konrad Polthier [†] Max Wardetzky [†]

Zuse Institute Berlin

Abstract

Feature lines are salient surface characteristics. Their definition involves third and fourth order surface derivatives. This often yields to unpleasantly rough and squiggly feature lines since third order derivatives are highly sensitive against unwanted surface noise. The present work proposes two novel concepts for a more stable algorithm producing visually more pleasing feature lines: First, a new computation scheme based on discrete differential geometry is presented, avoiding costly computations of higher order approximating surfaces. Secondly, this scheme is augmented by a filtering method for higher order surface derivatives to improve both the stability of the extraction of feature lines and the smoothness of their appearance.

1. Introduction

Feature lines are curves on surfaces carrying - *in a few strokes* - visually most prominent characteristics. Their extraction from discrete meshes has become an area of intense research [Thi96] [OBS04] [YBS05] [SF03] [HG01] [CP05b] with applications ranging from structure analysis in medical data [MAM97] [Sty03] over non photorealistic rendering techniques [IFP95] to surface segmentation [SF04].

Mathematically, feature lines are described as local extrema of principal curvatures along corresponding principal directions. On smooth surfaces, these extrema have been subject to intense mathematical studies based on techniques from differential geometry, singularity theory and bifurcation theory [Por94] [Koe90] [BAK97] [CP05a].

Reliable computations of discrete curvature measures on meshes are key to many methods in geometric modeling and computer graphics. In particular, the detection of ridges and features requires the computation of first and even second order derivatives of principal curvatures. Higher order derivatives, in turn, are not a straightforward concept on discrete surfaces, due to their piecewise linear nature. Consequently, the standard approach has been to locally (or sometimes globally) fit a smooth (often polynomial) surface to the vertex coordinates and to then compute curvatures from this smooth surface, see e.g. [CP03] [GI04] [OBS04].

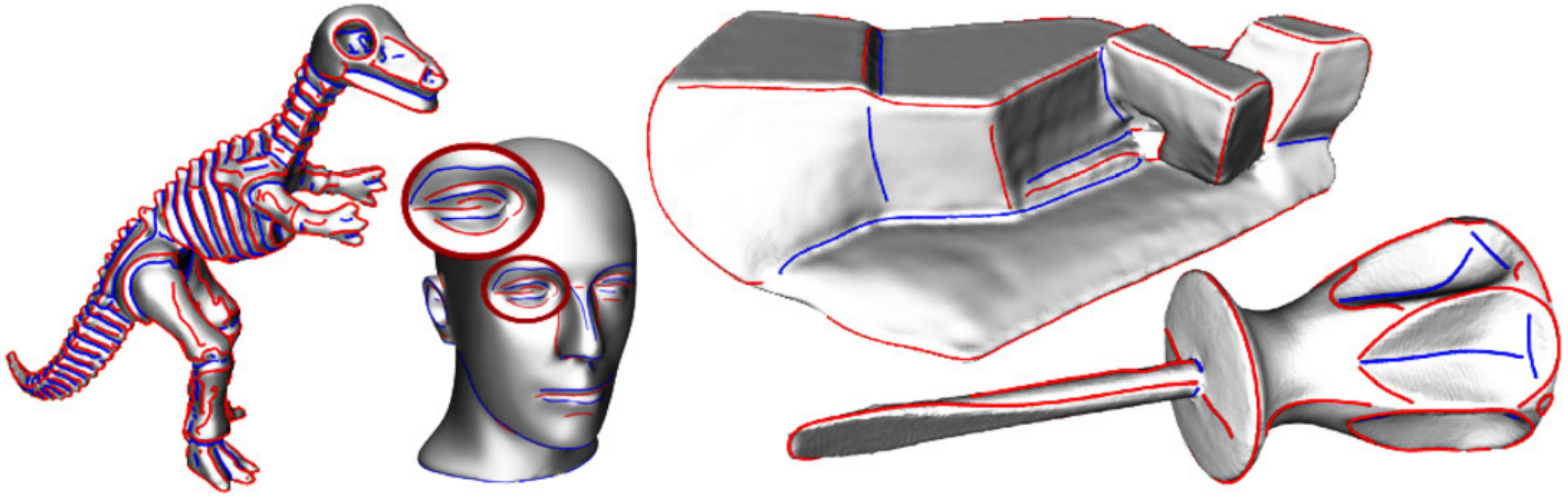
In contrast, our methodology is based on utilizing discrete differential operators on piecewise linear meshes, an approach which avoids often costly preprocessing steps such as higher order surface-fitting techniques or implicit surface schemes. Discretizing smooth geometric quantities has been found to be a powerful numerical machinery for geometry processing: The discrete mesh Laplacian [PP93] [MDSB03] is utilized for isotropic remeshing [AdVDI03], isotropic denoising [DMSB99], and mesh parameterization [GY03]. A model of a purely discrete shape operator [CSM03] [HP04] has been successfully employed in anisotropic remeshing and smoothing schemes [ACSD*03] [HP04] as well as thin shell simulations [GHDS03], to name a few of its most prominent applications. In the present work, we use the discrete approach for both extracting feature lines and smoothing thereof.

Figure 1: *Smooth feature lines on a motorcycle body part.*

[†] Supported by the DFG Research Center MATHEON "Mathematics for key technologies" in Berlin.

University of Victoria | Computer Science

Smooth Feature Lines on Surface Meshes



- Detect “salient ridges” on meshes
- Compute points with extreme principal curvatures
  - Under certain conditions, these form feature curves
  - Checking these conditions requires computing 1st and 2nd derivatives of curvature

Project 3:
Remeshing

Interactive Geometry Remeshing

Interactive Geometry Remeshing

Pierre Alliez, USC / INRIA
Mark Meyer, Caltech
Mathieu Desbrun, USC

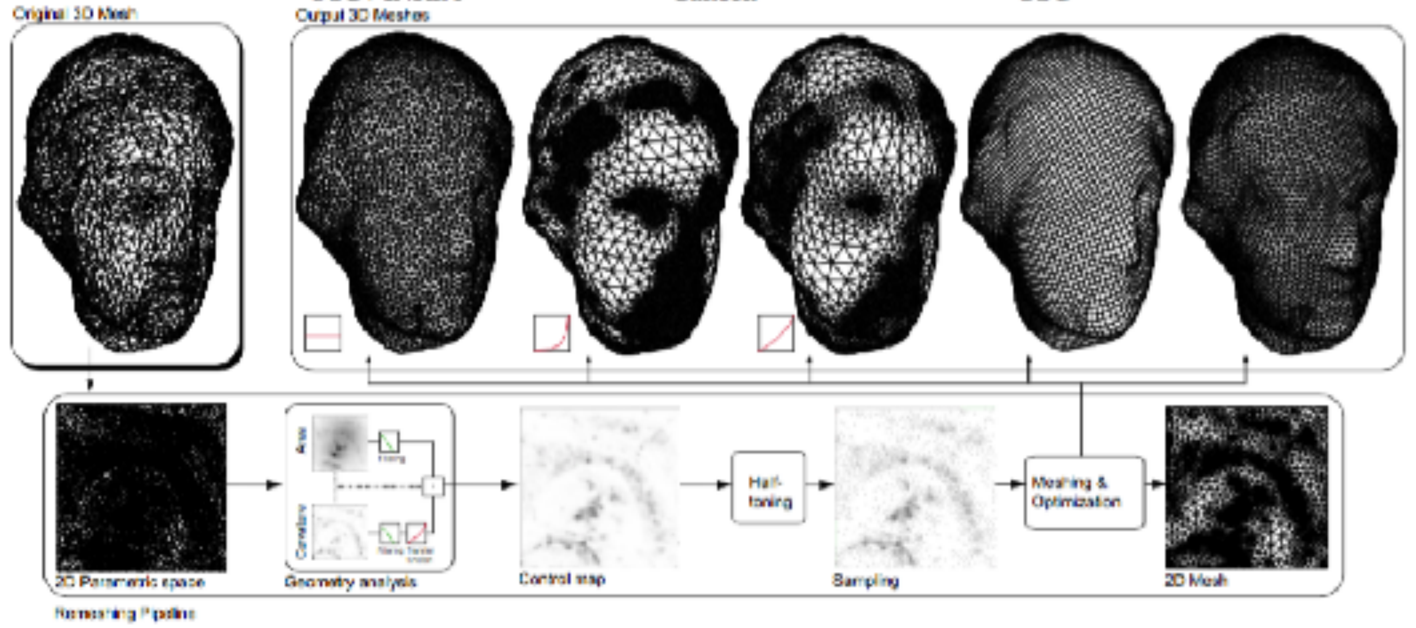


Figure 1: *A brief overview of our remeshing process: The input surface patch (top left) is first parameterized; Then geometric quantities are computed over the parameterization and stored in several 2D maps; These maps are combined to produce a control map, indicating the desired sampling distribution; The control map is then sampled using a halftoning technique, and the samples are triangulated, optimized and finally output as a new 3D mesh. A few examples of the various types of meshes our system can produce are shown (top, from left to right): uniform, increased sampling on higher curvature, the next with a smoother gradation, regular quads, and semi-regular triangles. After an initial pre-processing stage (~1s), each of these meshes was produced in less than 2 seconds on a low-end PC.*

Abstract

We present a novel technique, both flexible and efficient, for interactive remeshing of irregular geometry. First, the original (arbitrary genus) mesh is substituted by a series of 2D maps in parameter space. Using these maps, our algorithm is then able to take advantage of established signal processing and halftoning tools that offer real-time interaction and intricate control. The user can easily combine these maps to create a control map – a map which controls the sampling density over the surface patch. This map is then sampled at interactive rates allowing the user to easily design a tailored resampling. Once this sampling is complete, a Delaunay triangulation and fast optimization are performed to perfect the final mesh.

As a result, our remeshing technique is extremely versatile and general, being able to produce arbitrarily complex meshes with a variety of properties including: uniformity, regularity, semi-regularity, curvature sensitive resampling, and feature preservation. We provide a high level of control over the sampling distribution allowing the user to interactively custom design the mesh based on their requirements thereby increasing their productivity in creating a wide variety of meshes.

1 Introduction

As 3D geometry becomes a prevalent media, a proliferation of meshes are readily available, coming from a variety of sources including 3D scanners, modeling software, and output from computer vision algorithms. Although these meshes capture geometry accurately, their sampling quality is usually far from ideal for subsequent applications. For instance, these (sometimes highly) irregular meshes are not appropriate for computations using Finite Elements, or for rapid, textured display on low-end computers. Instead, meshes with nearly-equilateral triangles, a smooth gradation of sample density depending on curvatures, or even uniform sampling are preferable inputs to most existing geometry processing algorithms. *Remeshing*, i.e., modifying the sampling and connectivity of a geometry to generate a new mesh, is therefore a fundamental step for efficient mesh processing.

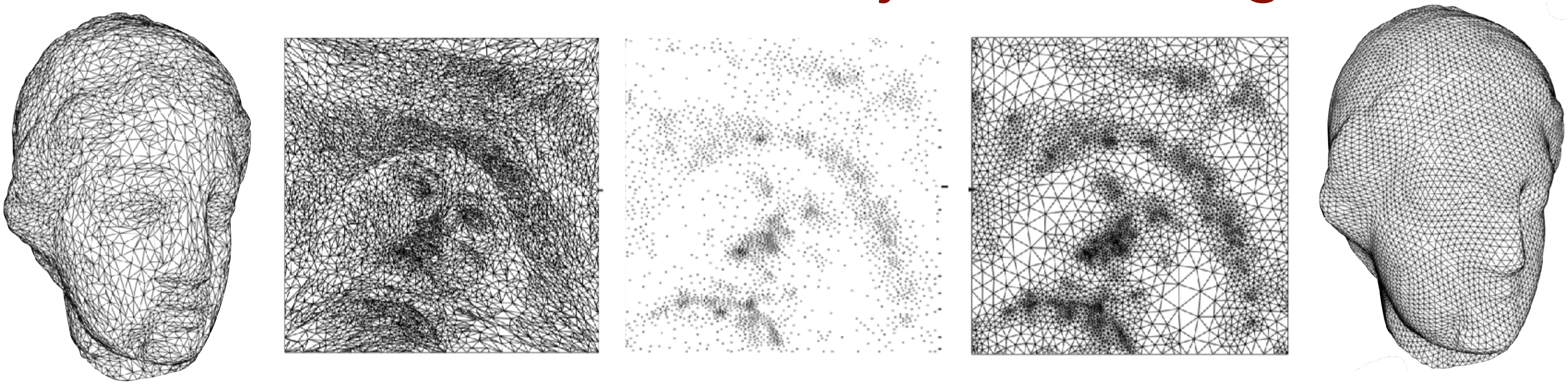
We propose a precise and flexible remeshing technique for arbitrary geometry. Unlike previous techniques, we offer a high level of control over the sampling quality of the output mesh, as well as an unprecedented speed of execution. We will show that our remeshing engine can accurately generate any "tailored" sampling at interactive rates, and, if necessary, quickly optimize the quality of the resulting mesh, allowing the user to easily design a resampled geometry conforming to her requirements.

1.1 Background

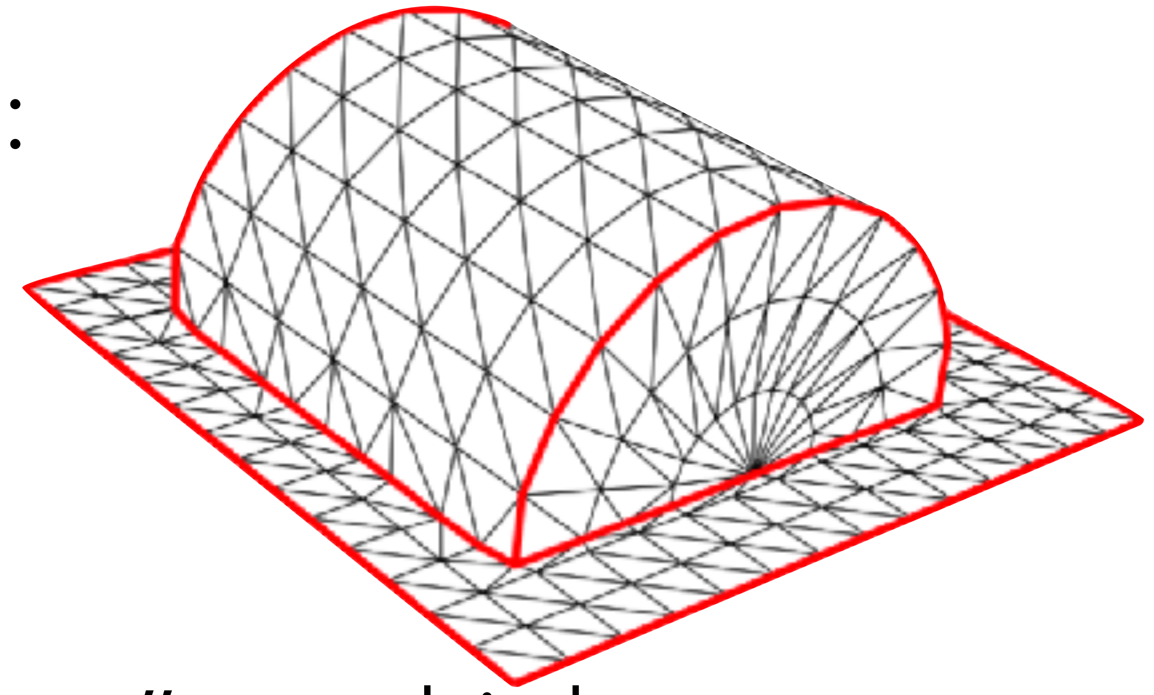
Although studied in Computer Graphics for obvious reasons, surface remeshing has also received a lot of attention from various non-CG fields interested in mesh generation — mainly Computational Fluid Dynamics, Finite Element Methods, and Computational Geometry. However, the diverging goals resulted in vastly different, non-overlapping solutions as we now briefly review.

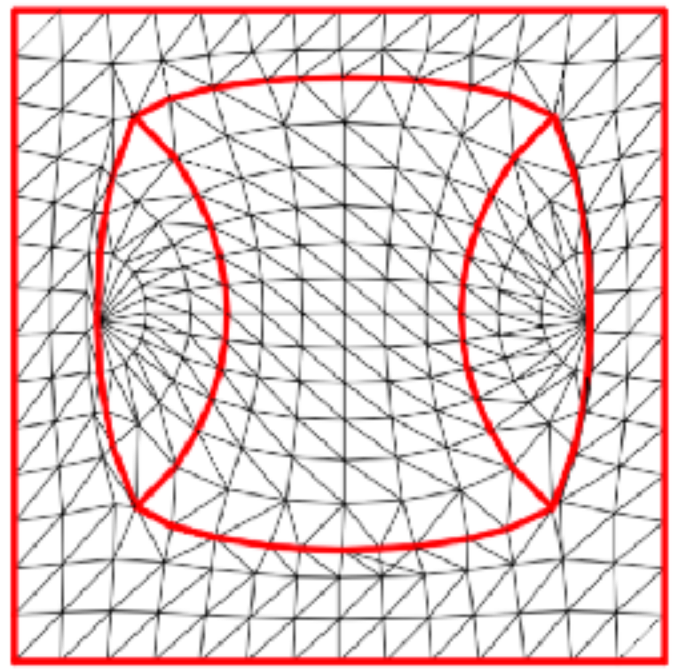
Mesh Generation Community Since the emphasis is generally on numerical accuracy, most of the tools developed in the non-CG communities focus on mesh quality. Remeshing procedures

Interactive Geometry Remeshing



- **Parametrize mesh** into (UV) plane, **triangulate** UV plane nicely, **transfer mesh back to 3D**.
- Design plane triangulation to **minimize post-transfer distortion**
  - Compensate for parametrization's area distortion: **vertex density proportional to area scaling**.
  - Detect creases in original mesh, constrain these to be edges in triangulation.
  - Optionally increase sampling in "important regions," e.g. high curvature.



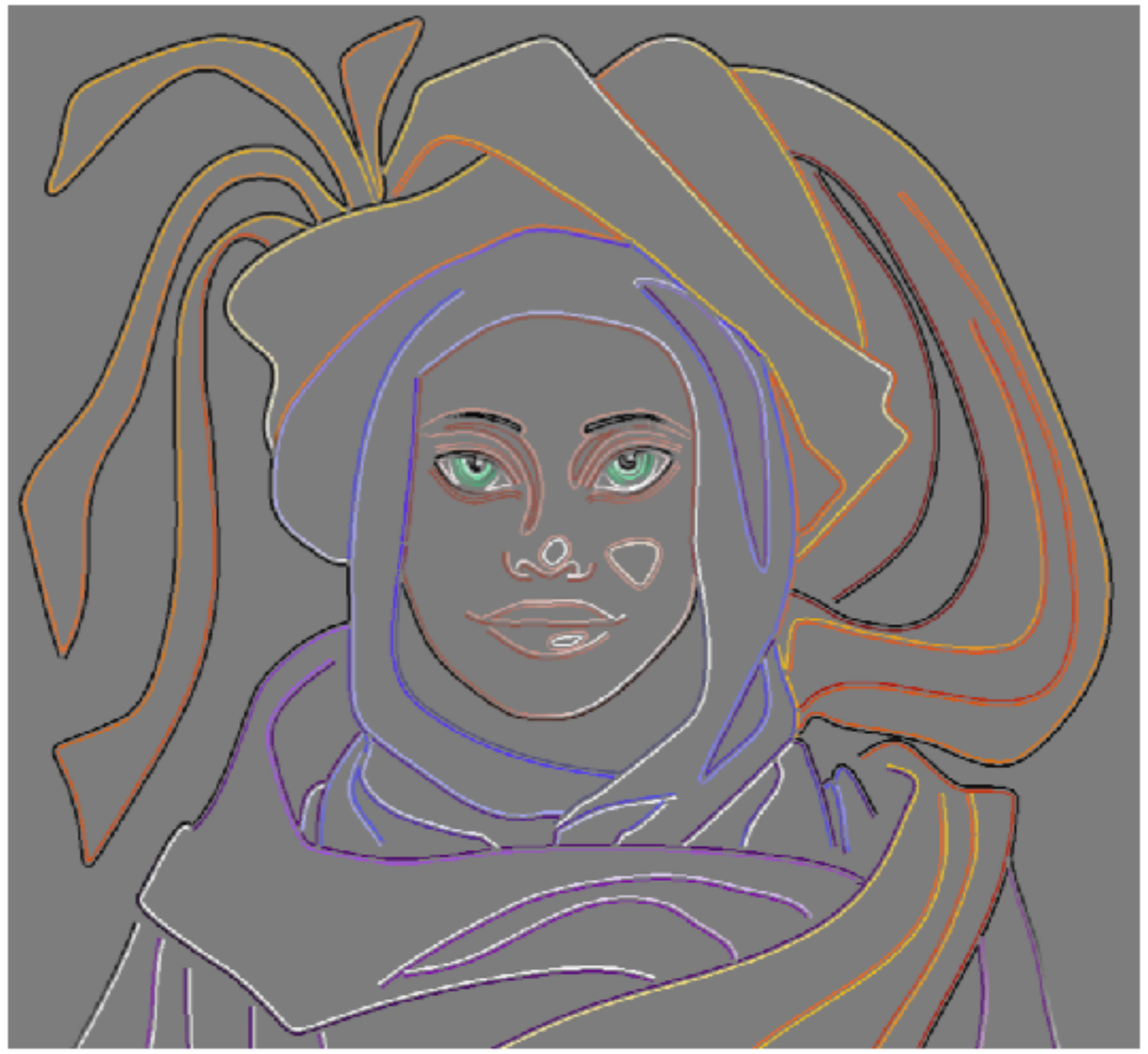


University of Victoria | Computer Science

Project 4:
Diffusion Curves

University of Victoria | Computer Science

Diffusion Curves



Diffusion Curves: A Vector Representation For Smooth-Shaded Images
Alexandrina Orzan, Adrien Bousseau, Pascal Barla, Holger Winnemöller, Joëlle Thollot, David Salesin

You can do it in 3d too!

Project 5: ARAP Deformation

University of Victoria | Computer Science

Shape Interpolation

Marc Alexa, Daniel Cohen-Or, and David Levin. 2000. As-rigid-as-possible shape interpolation