Interactive Generation of Feature Curves on Surfaces

Stuart Andrews
University of Toronto
Advisor: Professor Demetri Terzopoulos

April 14, 2000

A thesis submitted in conformity with the requirements for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto
Overview of Talk

• Introduction
  1. Problem and Motivation
  2. Virtual Planning and Simulation of Cranio-Facial Surgery
  3. Goals

• Challenges

• Background

• Proposed Solution

• Contributions

• Live-Wire

• Paths of Minimal Action

• Results and Conclusions
Problem and Motivation

Curves on 3D surfaces
A surface $S$
A curve $C \subset S$

Applications

• surface-based measurement

• navigation

• geometric modeling

• 3D shape analysis

Problem
Specific desired curve qualities $\implies$ difficult to draw by hand
Cranio-Facial Surgery

Virtual Surgery - Planning and simulation

- make precise measurements
- delineate incisions
- extract feature curves
Goals

Provide a tool to compute certain types of curves upon user request

(Like ruler, protractor, or compass)

• general purpose

• explicit user control

• capacity of generating curves that identify specific features
Challenges

- Introduction
- Challenges
  1. Representation
  2. User Interaction
  3. Properties of Curves
- Background
- Proposed Solution
- Contributions
- Live-Wire
- Paths of Minimal Action
- Results and Conclusions
Representations

We need a parameterization for the surface $S$, and for curves $C$ on $S$.

- to perform traversals of the surface
- to enforce embedding property $C \subset S$
- to represent geometrically and topologically complex surfaces and curves
User Interaction

Drawing in 2D

- mouse-based drawing interfaces
- poor results
- Computer Aided Design systems are often cumbersome

For drawing on 3D surfaces, the situation is worse. We must project 2D mouse input onto the surface, which introduces new problems.

- jump discontinuities
- distortion of distance
- occlusions

A simple projection of the 2D mouse input to the surface does not work.
Properties of Curves

How do we specify, and then compute, useful feature curves?

- examples
  1. shortest or minimal cost path
  2. ridge line
  3. iso-contour

- smoothness of curve

- invariance to surface representation, scale, transformation

- reproducible
Background

2D image processing

- Snakes: Active contours models
  [Kass et al., 1987]

- Geodesic active contours
  [Caselles et al., 1997]

- Computing geodesics paths on manifolds
  [Kimmel et al., 1995]

- Interactive live-wire boundary extraction
  [Barrett and Mortensen, 1997]

- Global minimum for active contour models
  [Cohen and Kimmel, 1997]
Proposed Solution

- Introduction
- Challenges
- Background

- Proposed Solution
  1. Choice of Representation
  2. Feature Curves
  3. Interaction

- Contributions

- Live-Wire

- Paths of Minimal Action

- Results and Conclusions
Choice of Representation

- surface $S$ is represented by a triangle mesh

- the location of a point $x \in S$ is parameterized by the triangle index in which it lies, and within the triangle, by barycentric coordinates

- curve $C : \Omega \rightarrow S$ is represented by a sequence of control points

\[ C = \langle x^0, x^1, x^2, \ldots x^M \rangle \]

with the property that $ax^i + (1 - a)x^{i+1} \in S$ for $a \in [0, 1]$
Feature Curves

The curves that we generate are minimal geodesics.

In the Euclidean metric space, these are equivalent to shortest paths.

We obtain other types of curves by defining alternative Riemannian metric spaces.

- geodesic property - locally minimizes $\int \| C_s \| \, ds$

- redefine arclength - locally minimizes $\int P(C_s) \, ds$
Riemannian Metric

• scalar field $P$
  1. intrinsic surface properties (curvature)
  2. extrinsic properties (position, colour)
Interaction

• solves 2D to 3D mapping problem
  1. algorithm constructs embedded curves
  2. curve segments have specific properties
  3. projection & occlusion problems

• feature curves are \textit{selected} incrementally
  \[ ? \quad \Rightarrow \quad \text{restricted freedom} \]

• allows user to adaptively moderate precision

An intuitive form of 3D interaction generating curves on surfaces.
Contributions

We introduce the novel use of 2D image processing techniques for generating feature contours on a surface.

1. We extend the live-wire technique to graphs representing surfaces in $\mathbb{R}^3$.

2. We extend the minimal path framework [Cohen and Kimmel, 1997] to 3D, and couple it with live-wire interaction.
Live-Wire

- Introduction
- Challenges
- Background
- Proposed Solution
- Contributions
- Live-Wire
  1. 2D Live-Wire
  2. Extension to 3D
  3. Evaluation
- Paths of Minimal Action
- Results and Conclusions
Problem - locate object boundaries in images

Observation: the contours are linked to edge detectors (e.g., image gradient)

How do we generate a contour over these features?

Solution [Barrett and Mortensen, 1997]

1. represent image as a graph and curves as edge sequences
2. define an cost function on graph edges
3. identify feature contours $C$ in image as minimal paths in the graph

$$
\arg \min_C \sum_{e \in C} w(e),
$$

4. Use Dijkstra’s algorithm to compute the shortest paths to the source point ($\mathcal{O}(n \log n)$ time, where $n$ is the number of nodes).
How should we define the edge cost $w(e)$?

$$w(e) = \begin{cases} 
\text{small} & \text{if } e \text{ is near object boundary} \\
\text{large} & \text{otherwise}
\end{cases}$$

‘Near object boundary’ is defined with respect to the image gradient, or other edge detectors.
Live-Wire - Extension to 3D

1. represent surface $S$ as a graph

2. scalar field $P$ defined at the vertices

3. define the edge costs to reflect features of interest on the surface

\[ w(e) = \left( \alpha + \frac{P(p) + P(q)}{2} \right) \| e \| \]

where $e = \langle p, q \rangle$ connects vertices $p$ and $q$
Evaluation

Pros - direct, intuitive, and fast

Cons - paths are restricted to edge sequences in the graph

Can we implement this in a continuous manner?
Paths of Minimal Action

- Introduction
- Challenges
- Background
- Proposed Solution
- Contributions
- Live-Wire
- Paths of Minimal Action
  1. Intuition
  2. Paths of Minimal Action
  3. Wave Propagation
  4. Surface of Minimal Action
- Results and Conclusions
Intuition

Dijkstra’s algorithm mimics a *wave propagation*

Paths are reconstructed by following sequence of ‘previous’ pointers (a descent).
Paths of Minimal Action

- paths of minimal action [Cohen and Kimmel, 1997] are defined as geodesics

  \[ \arg\min_C \int_\Omega \alpha + P(C(s)) \, ds \]

- curves \( C \) are no longer restricted to the edges of the graph

- compare with discrete solution

  \[ \arg\min_C \sum_{e \in C} \left( \alpha + \frac{P(p) + P(q)}{2} \right) \| e \| \]

- Riemannian metric: \( \tilde{P} ds = (\alpha + P) ds \)
Wave Propagation

- use wave propagation to compute *geodesic circles* from which we obtain *geodesics*

- arrival time $t$ is proportional to distance

The parametric solution for advancing wavefront is the solution of the following boundary value PDE:

$$ \frac{\partial u(x, t)}{\partial t} = \frac{1}{\tilde{P}} (N(x) \times T(x)), $$

where $u(A, 0) = 0$. $T(x)$ is the unit tangent to the wavefront and $N(x)$ is the unit surface normal.
Surface of Minimal Action

Problem
solving this PDE is messy

Solution
embed the wavefront in a higher dimensional function $U$ that is easier to compute

compute geodesics using gradient decent on $U$.

• No additional complexity compared to Dijkstra’ algorithm

• Smooth results
Results
Elasticity
3D surfaces
3D surfaces
Conclusion

- we have extended two methods for generating curves on a triangle mesh surface

- interactive and intuitive easy to use

- easy designation of feature curves - especially ridge lines

- handles the underconstrained nature of 2D input to generate 3D surface curves

In the future, we would like to consider the following:

- apply minimal path techniques to the PETRA - Digital Archeology project

- automate the selection of start and end points
References


