Digital Geometry Processing

**Introduction**

**Objective**
- Theory and applications of 3D mesh processing
- Hands-on experience

**Requirements**
- Recommended Prerequisite: Computer Graphics (234325 or 046345)

**Grade:**
- Programming exercises (50%) - MeshMaker Hello World (3%)
- Compression (22%)
- Parameterization (25%)
- Modest final project (50%)
- Reconstruction

Work in pairs. Use MeshMaker API.

**Syllabus**
- Discrete Geometry
- Coding and compression
- Simplification
- Parameterization
- Filtering
- Reconstruction
- Editing
- Morphing

**Applications**
- Medical
- Engineering
- Topography
- Simulation
- Games
- E-commerce
  - A live e-commerce application
- Culture

Copyright 2008  C. Gotsman & A. Sheffer
Technion, Harvard, UBC
Digital Geometry Processing

Introduction

Mesh Construction

- from point clouds
- from contours

Simplification

- 20,000
- 8,000
- 2,000

Applications

- Oversampled 3D scan data
- ~150k triangles
- ~80k triangles

Applications

- Over tessellation: E.g., iso-surface extraction

Applications

- Multi-resolution hierarchies for
- efficient geometry processing
- level-of-detail (LOD) rendering

Applications

- Adaptation to hardware capabilities

Copyright 2008  C. Gotsman & A. Sheffer
Technion, Harvard, UBC
Digital Geometry Processing

Introduction

Size-Quality Tradeoff

Remeshing

More Remeshing

Compression

Parameterization

Application - Texture Mapping

Copyright 2008  C. Gotsman & A. Sheffer
Technion, Harvard, UBC
Introduction

Morphing

Virtual Reality Modeling Language
- Invented 1995
- Describes the geometry and properties of a 3D scene
- May be viewed (rendered) by many apps and browser plugins

#VRML V2.0 utf8
Shape {
  geometry IndexedFaceSet {
    point 0 0 0, 1.0 0 0, 0 1.0 0, 0 0 1.0,
    coordIndex 0 1 2 -1, 1 3 2 -1, 0 3 1 -1, 2 3 0 -1
  }
}

MeshMaker
- A MS Visual Studio C++ API for
  Mesh queries
  Geometry operations
  Rendering
- Based on
  MFC for GUI
  OpenGL for 3D rendering
- Current version: 5.3

What does MeshMaker do?
- Rendering
  rendering modes
  color control
  highlighting
  text
- I/O of VRML data files
- Scene interaction using mouse
- Efficient DCEL-type mesh queries
  vertex neighborhood
  validation
- Efficient mesh operations
  vertex/faces removal
  vertex split
  edge collapse

Questions?

Chapter 1
Introduction and Basic Concepts
Introduction

**Standard Graph Definitions**

- **Graph** \( G = \langle V, E \rangle \)
  - \( V \) = vertices \( \{A, B, C, D, E, F, G, H, I, J, K, L\} \)
  - \( E \) = edges \( \{(A, B), (B, C), (C, D), (D, E), (E, F), (F, G), (G, H), (H, A), (A, J), (A, G), (B, J), (K, F), (C, L), (C, I), (D, I), (D, F), (F, I), (G, K), (J, L), (K, L), (L, I)\} \)

- **Vertex degree (valence)**: number of edges incident on vertex
  - \( \text{deg}(J) = 4 \)
  - \( \text{deg}(H) = 2 \)

- **k-regular graph**: graph whose vertices all have degree \( k \)

- **Face**: cycle of vertices/edges which cannot be shortened
  - \( F = \{(A, H, G), (A, G, K, J), (B, A, J), (B, J, L), (C, L, C), (C, L, D), (D, F, E), (D, F, I), (L, K, F), (L, J, K), (K, G, F), (A, B, C, D, E, F, G, H)\} \)

- **Planarity**: graph can be drawn in \( \mathbb{R}^2 \) without edge crossings

**Connectivity**

- **Graph is connected** if there is a path of edges connecting every two vertices
- **Graph is k-connected** if between every two vertices there are \( k \) edge-disjoint paths

**Graph Embedding**

- **Graph is embedded in** \( \mathbb{R}^d \) if each vertex is assigned a position in \( \mathbb{R}^d \)

**Planar Graphs**

- **Planar graph**: graph whose vertices and edges can be embedded in \( \mathbb{R}^2 \) such that its edges do not intersect
- **Every planar graph can be drawn as a straight-line plane graph**

**Triangulation**

- **Triangulation**: straight line plane graph all of whose faces are triangles

**Meshes**

- **Closed mesh**: mesh with no boundary edges
- **Manifold mesh**: mesh with no singular edges
- **Non-Manifold**

---

Copyright 2008  C. Gotsman & A. Sheffer
Technion, Harvard, UBC
Planar Graphs and Meshes

Every (genus 0) manifold mesh with a boundary is planar!! Flatten!!

Topology

Euler-Poincare Formula

\[ v + f - e = 2(c - g) - b \]

\( v = \# \text{ vertices} \)
\( c = \# \text{ conn. comp} \)
\( f = \# \text{ faces} \)
\( g = \text{genus} \)
\( e = \# \text{ edges} \)
\( b = \# \text{ boundaries} \)

Examples

Genus 0
Genus 1
Genus 2

Exercise I

Theorem: Average vertex degree in closed manifold triangle mesh is \(~6\)

Proof: In such a mesh, \(3f = 2e\) by counting edges of faces.
By Euler's formula, \(v + f - e = 2 - 2g\).
Thus \(e = 3(v - 2 + 2g)\).
So Average(degree) = \(2e/v = 6v(v - 2 + 2g)/v = 6\) for large \(v\)

Corollary: Only toroidal (\(g=1\)) closed manifold triangle mesh can be regular (all vertex degrees are \(~6\))

Proof: In regular mesh average degree is exactly 6.
Can happen only if \(g=1\)

Exercise II

Theorem: In closed manifold mesh:

\(2e \geq 3f\) (equality for triangle mesh),
\(e \geq 3v\)

Corollary: No closed manifold mesh can have 7 edges

Corollary: \(2f - 4 \geq v\)

Euler-Poincare implies that any planar graph has an independent set of size at least \(v/3\).
Orientability

Orientation of a face is clockwise or anticlockwise order in which its vertices and edges are listed. This defines the direction of face normal. A plane graph is orientable if orientations of its faces can be chosen so that each edge is oriented in both directions. For example, the Möbius strip or Klein bottle is not orientable.

Orientation of a face

Oriented

F={L,J,B),(B,C,L),(L,C,I),
(L,K,L),(L,K,J)}

Not Oriented

F={(B,J,L),(B,C,L),(L,C,I),
(L,I,K),(L,K,J)}

Convexity

Set $C \subseteq \mathbb{R}^d$ is convex if for any two points $p,q \in C$ and any $\alpha \in [0,1]$, $\alpha p + (1-\alpha)q \in C$.

Convex hull of set $S \subseteq \mathbb{R}^d$ is the minimal convex set $C$ containing $S$.

Mesh is convex if all its vertices are on its convex hull.

Developability

Mesh is developable if it may be embedded in $\mathbb{R}^2$ without distortion.

Almost Developable

Non-Developable