

ECE 595 / CS 491 / CS 591
**Real-Time Rendering &
Graphics Hardware**

Pradeep Sen
Advanced Graphics Lab

Class 20
April 9, 2007

Announcements

- I will speak at this week's GFX Café
- Capturing light fields and refocusing images

AGL Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 20 - April 9, 2007

Last time

- Advanced topics on Precomputed Radiance Transfer (PRT)

AGL Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 20 - April 9, 2007

Today

- Real-time geometrical models
- Level of detail

AGL Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 20 - April 9, 2007

Sources of 3-D data

- Typing in the geometric representation
- Writing a program to generate this data
- Output from a modeling program
- Sampling a real model at various points using a 3-D digitizer
- Reconstruction from photographs of an object (photogrammetry)
- Using a 3-D scanner which gathers depth info

AGL Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 20 - April 9, 2007

Point clouds

- Scanners give us point clouds, we need to tessellate them in order to get triangular meshes we can render



point cloud



high-res triangular mesh

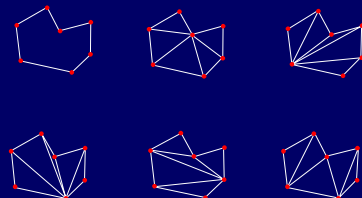


simplified triangular mesh

AGL Real-time Rendering & Graphics Hardware
Pradeep Sen

Images courtesy of Hugues Hoppe
Class 20 - April 9, 2007

Triangulation or tessellation



AGL Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 20 - April 9, 2007

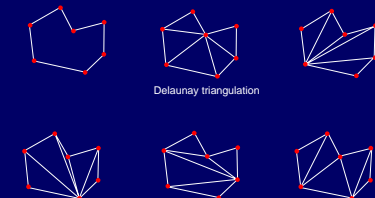
Delaunay Triangulation

- Triangulation of set of points P such that no point in P is inside the circumcircle of any triangle in the triangulation
- Delaunay triangulation maximizes the minimum angle of all the triangles (avoids thin, sliver triangles)
- There is a unique Delaunay triangulation for a general set of points P (no three points co-linear, no 4 points on the same circle)
- Equivalent to the dual of the Voronoi diagram

AGL Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 20 - April 9, 2007

Triangulation or tessellation



Delaunay triangulation

AGL Real-time Rendering & Graphics Hardware
Pradeep Sen

Class 20 - April 9, 2007

Triangulation or tessellation

Real-time Rendering & Graphics Hardware
Pradeep Sen
Class 20 - April 9, 2007

Triangulation or tessellation

Real-time Rendering & Graphics Hardware
Pradeep Sen
Class 20 - April 9, 2007

Triangulation or tessellation

Real-time Rendering & Graphics Hardware
Pradeep Sen
Class 20 - April 9, 2007

Delaunay triangulation algorithms

- Most Delaunay algorithms use the fact that we can quickly determine if a point D is inside the circumcircle defined by points A, B, C:

$$\begin{vmatrix} A_x & A_y & A_x^2 + A_y^2 & 1 \\ B_x & B_y & B_x^2 + B_y^2 & 1 \\ C_x & C_y & C_x^2 + C_y^2 & 1 \\ D_x & D_y & D_x^2 + D_y^2 & 1 \end{vmatrix}$$

If A, B, and C lie counterclockwise, this determinant is positive only if D is inside the circumcircle.

Real-time Rendering & Graphics Hardware
Pradeep Sen
Class 20 - April 9, 2007

After we have the triangle mesh

- We typically want to do several things to it:
 - Animate it
 - Simplify it for level of detail

Real-time Rendering & Graphics Hardware
Pradeep Sen
Class 20 - April 9, 2007

Animation of triangular meshes

- In real-time rendering, the animation of triangular meshes is typically done with the use of *skinning*
- In skinning, a skeleton is attached to the model and matrices are defined that transform each of the bones in the skeleton
- Vertices of the mesh are transformed by a weighted sum of the matrices of the bones that affect them

Real-time Rendering & Graphics Hardware
Pradeep Sen
Class 20 - April 9, 2007

Skinning example

skeleton of articulated arm
triangular mesh for arm
bones undergoing transformation
resulting mesh deformation

Images courtesy of Jeff Lander
Real-time Rendering & Graphics Hardware
Pradeep Sen
Class 20 - April 9, 2007

Mathematics of skinning

- A vertex v is transformed by a weighted sum of the transformations by each of the matrices associated with individual bones:

$$f(v) = \sum_{i=1}^n w_i M_i v$$

where

$$1 = \sum_{i=1}^n w_i$$

Real-time Rendering & Graphics Hardware
Pradeep Sen
Class 20 - April 9, 2007

After we have the triangle mesh

- We typically want to do several things to it:
 - Animate it
 - Simplify it for level of detail

Real-time Rendering & Graphics Hardware
Pradeep Sen
Class 20 - April 9, 2007

Mesh simplification

[Schroeder-etal92]
[Turk92]
[Hoppe-etal93]
[Rossignac-Borrel93]
[Cohen-etal96]
...

Slide courtesy of Hugues Hoppe
Class 20 - April 9, 2007

Level-of-detail (LOD)

distance from viewer?

close far

[Clark76]
[Funkhouser93]

Concern: transitions may "pop"
Would like smooth LOD

Slide courtesy of Hugues Hoppe
Class 20 - April 9, 2007

Mesh simplification operations

- Edge collapse
- Edge split
- Edge swap

Slide courtesy of Hugues Hoppe
Class 20 - April 9, 2007

Edge collapse

$ecol(v_s, v_t, v_s')$

(optimization)

Slide courtesy of Hugues Hoppe
Class 20 - April 9, 2007

Progressive meshes

- Hugues Hoppe "Progressive meshes" SIGGRAPH 1996

Slide courtesy of Hugues Hoppe
Class 20 - April 9, 2007

Simplification process

$\hat{M} = M^n \xrightarrow{ecol_{n-1}} M^{175} \xrightarrow{ecol_i} M^1 \xrightarrow{ecol_0} M^0$

Slide courtesy of Hugues Hoppe
Class 20 - April 9, 2007

Edge collapse can be inverted

Vertex split transformation:

$vspl(v_s, v_l, v_r, v_s', v_t', \dots)$

attributes

Slide courtesy of Hugues Hoppe
Class 20 - April 9, 2007

Mesh reconstruction

$M^0 \xrightarrow{vspl_0} M^1 \xrightarrow{\dots} M^{175} \xrightarrow{vspl_{n-1}} M^n = \hat{M}$

progressive mesh (PM) representation

Slide courtesy of Hugues Hoppe
Class 20 - April 9, 2007

Create a model of the desired resolution

From PM, extract M_i of any desired complexity.

$M^0 \xrightarrow{vspl_0} M^1 \xrightarrow{vspl_1} \dots \xrightarrow{vspl_n} M^n = \hat{M}$

3,478 faces?

Slide courtesy of Hugues Hoppe
Class 20 - April 9, 2007

Vertex correspondence

