Point-Based Rendering

September 30, 2004

Announcement

- No class next Monday.
- Volunteers wanted for early paper presentations.

Review of 3D Warping

- Input: Depth Images.
- No 3D model is constructed.
- Each pixel is often considered as a point sample in object space.

Artifacts of 3D Warping

- Due to incorrect reconstruction.
- Due to occlusion (or visibility).

Question: how to fill the holes?

Quick fix: draw larger points.

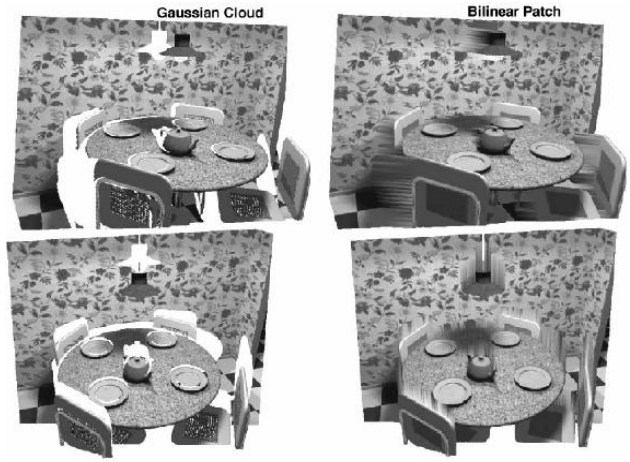
Next thought: draw some larger points and some smaller points (based on distance).

Another thought: draw elliptical points (based on neighbors and surface orientation)

Reconstruction

- Solving it by meshing?
  - –Must know the connectivity between points.
- Solving it by splatting?
  - –Size determined by viewing distance.
  - –Shape determined by surface normal.
  - –References: LDI [SG98], Surfel [SG2000], Surface Splatting [SG2001].

Meshing Artifacts



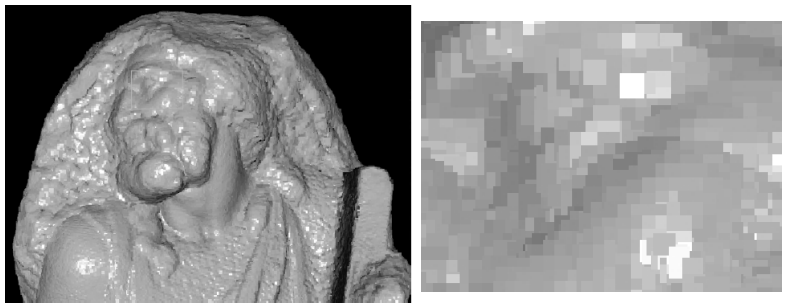
Picture source: Leonard McMillan

Simple Splats

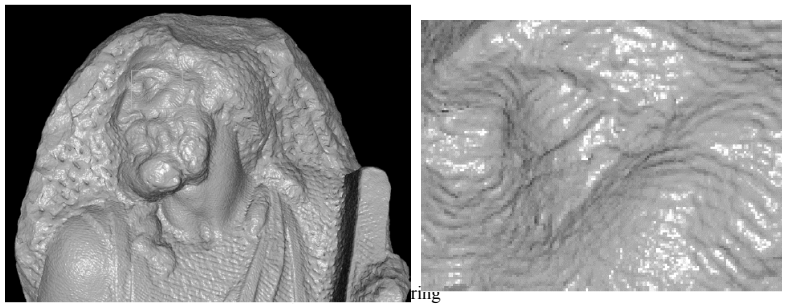
- Fixed, oversized splats
- Calculation of splat size based on:
  - –Distance
  - –Normal
- QSplat's approach
- LDI's approach

LOD Control

Threshold:
15 pixels
Points:
130,712
Rendering Time:
132 ms



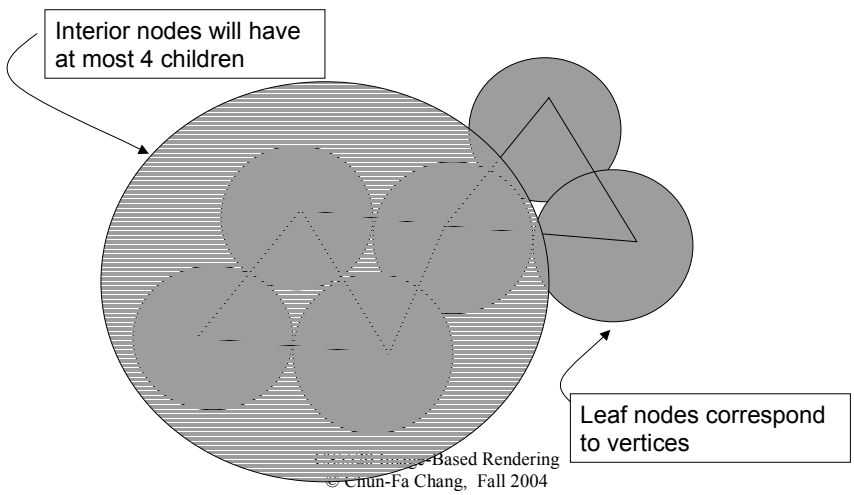
Threshold:
1 pixel
Points:
14,835,967
Rendering Time:
8308 ms



Michelangelo's statue of St. Matthew

Preprocessing

- Building the Hierarchy tree…
  What do the nodes look like?



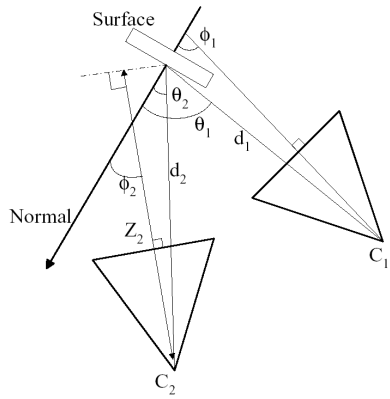
LDI's Approach

- Consider both the distance and the normal.

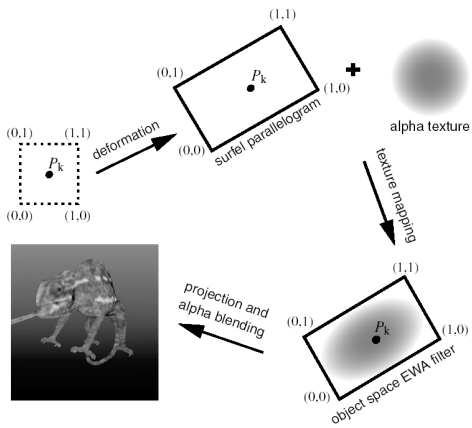
$$\sqrt{size} = \frac{1}{d_2} \cdot \frac{d_1\sqrt{\cos(\theta_2)res_2\tan(fov_1/2)}}{\sqrt{\cos(\theta_1)res_1\tan(fov_2/2)}}$$

$$\approx \frac{1}{Z_2} \cdot \frac{d_1\sqrt{\cos(\phi_2)res_2\tan(fov_1/2)}}{\sqrt{\cos(\phi_1)res_1\tan(fov_2/2)}}$$

$$\approx z_2 \cdot \frac{d_1\sqrt{\cos(\phi_2)res_2\tan(fov_1/2)}}{\sqrt{\cos(\phi_1)res_1\tan(fov_2/2)}}$$

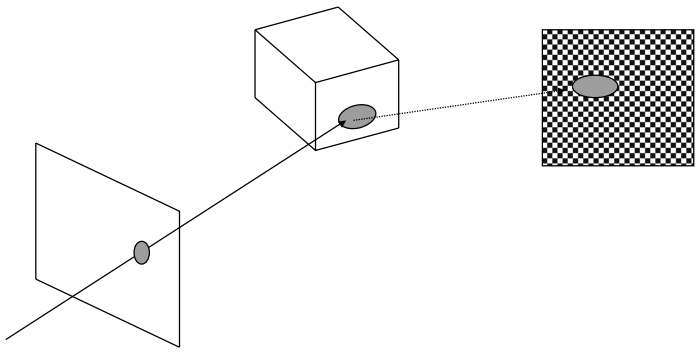


Surface Splatting

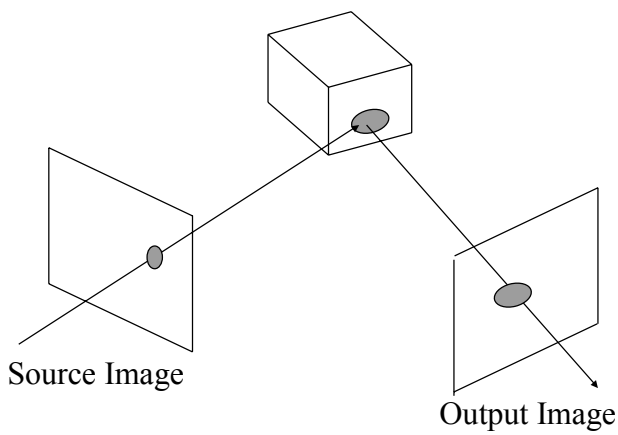


Source: Ren et al, "Object-Space EWA Surface Splatting", EG 2002

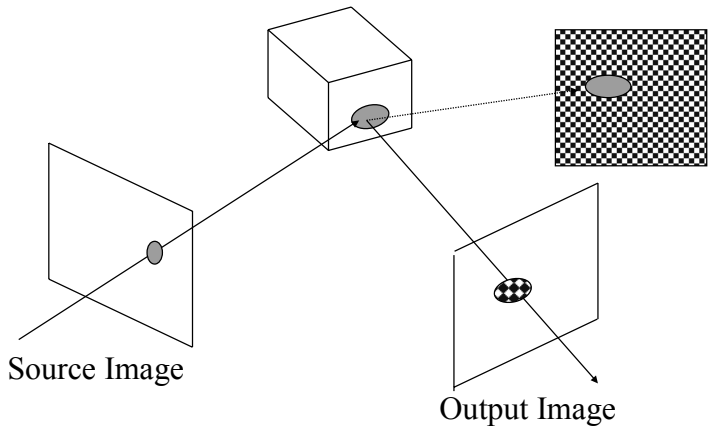
EWA Texture Filtering



Surface Splatting



Surface Splatting of Textured Surfaces

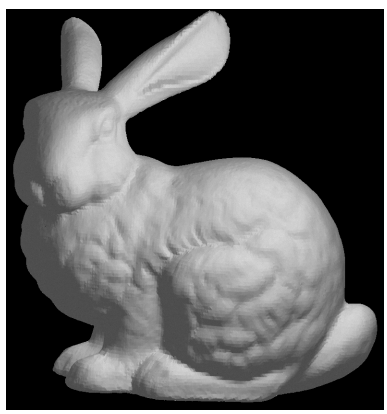


Screen-Space vs. Object-Space

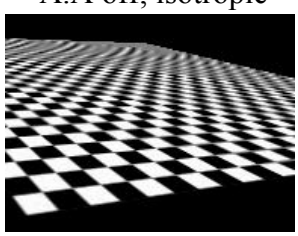
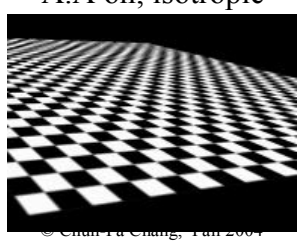
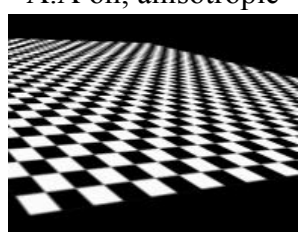
- Screen space:
  - PFISTER et al., Surfels: surface elements as rendering primitives. SIGGRAPH 2000.
  - ZWICKER et al., Surface splatting. SIGGRAPH 2001.
- Object-Space:
  - REN et al., Object space EWA surface splatting. Eurographics 2002.

Hardware Features

- What happens if the many tiny splats are mapped to a pixel?
  - Solution 1: Prefiltering
  - Solution 2: Turn on antialiasing!
- Textured surfaces: why is it an issue?
  - Can't we store the texture color with the point data?
  - No! If a splat covers more than 1 pixel, then it should trigger multiple texture lookups.
  - Solution: Turn on multi-texture and anisotropic texturing!



A.A off; isotropic | A.A on; isotropic | A.A on; anisotropic

Data Representation

- The simplest form is a depth image.
- Layered Depth Image (LDI):
  - –By combining multiple depth images.
- Layered Depth Cube (LDC):
  - –By using 3 LDIs.
- The above may be extended to include multi-resolution.
  - –Examples: LDI Tree, QSplats, Surfels.

The End