Image-based modeling (IBM) and image-based rendering (IBR)

CS 248 - Introduction to Computer Graphics
Autumn quarter, 2005
Slides for December 8 lecture
The graphics pipeline

modeling → animation → rendering
The graphics pipeline

the traditional pipeline

modeling → animation → rendering

the new pipeline?

3D scanning → motion capture → image-based rendering
“The study of image-based modeling and rendering is the study of sampled representations of geometry.”
Image-based representations: the classics

3D
- model + texture/reflectance map [Blinn78]
- model + displacement map [Cook84]
- volume rendering [Levoy87, Drebin88]

2D + Z
- range images [Binford73]
- disparity maps [vision literature]

2.5D
- sprites [vis-sim, games]

n 2D
- epipolar plane images [Bolles87]
- movie maps [Lippman78]

2D
- environment maps, a.k.a. panoramas [19th century]
Recent additions

- full model
  - view-dependent textures [Debevec96]
  - surface light fields [Wood00]
  - Lumigraphs [Gortler96]

- sets of range images
  - view interpolation [Chen93, McMillan95, Mark97]
  - layered depth images [Shade98]
  - relief textures [Oliveira00]

- feature correspondences
  - plenoptic editing [Seitz98, Dorsey01]

- camera pose
  - image caching [Schaufler96, Shade96]
  - sprites + warps [Lengyel97]
  - light fields [Levoy96]

- no model
  - outward-looking QTVR [Chen95]
Rangefinding technologies

• passive
  – shape from stereo
  – shape from focus
  – shape from motion, etc.

• active
  – texture-assisted shape-from-X
  – triangulation using structured-light
  – time-of-flight
Laser triangulation rangefinding

- Laser
- Laser sheet
- Cylindrical lens
- Direction of travel
- Object
- CCD image plane
- CCD
Post-processing pipeline

• steps
  1. aligning the scans
  2. combining aligned scans
  3. filling holes
Digitizing the statues of Michelangelo using laser scanners

- 480 individually aimed scans
- 2 billion polygons
- 7,000 color images
- 30 nights of scanning
- 22 people
Replica of Michelangelo’s David
(20 cm tall)
Solving the jigsaw puzzle of the Forma Urbis Romae
The puzzle as it now stands
Clues for solving the puzzle

- incised lines
- incision characteristics
- marble veining
- fragment thickness
- shapes of fractured surfaces
- rough / smooth bottom surface
- straight sides, indicating slab boundaries
- location and shapes of clamp holes
- the wall: slab layout, clamp holes, stucco
- archaeological evidence
Matching incised lines

fragment 156

fragment 167

fragment 134
Geometry-based versus image-based rendering

conceptual world

model construction

geometry

geometry-based rendering

flythrough of scene

real world

image acquisition

images

computer vision

rendering

flythrough of scene
Shortcutting the vision/graphics pipeline

real world

vision pipeline

geometry

graphics pipeline

views

image-based rendering

(from M. Cohen)
Apple QuickTime VR
[Chen, Siggraph ’95]

• outward-looking
  – panoramic views taken at regularly spaced points

• inward-looking
  – views taken at points on the surface of a sphere
View interpolation from a single view

1. Render object
2. Convert Z-buffer to range image
3. Tesselate to create polygon mesh
4. Re-render from new viewpoint
5. Use depths to resolve overlaps

Q. How to fill in holes?
View interpolation from multiple views

1. Render object from multiple viewpoints
2. Convert Z-buffers to range images
3. Tesselate to create multiple meshes
4. Re-render from new viewpoint
5. Use depths to resolve overlaps
6. Use multiple views to fill in holes
Post-rendering 3D warping
[Mark et al., I3D97]

- render at low frame rate
- interpolate to real-time frame rate
  - interpolate observer viewpoint using B-Spline
  - convert reference images to polygon meshes
  - warp meshes to interpolated viewpoint
  - composite by Z-buffer comparison and conditional write
Results

- rendered at 5 fps, interpolated to 30 fps
- live system requires reliable motion prediction
  - tradeoff between accuracy and latency
- fails on specular objects
Image caching
[Shade et al., SIGGRAPH 1996]

• precompute BSP tree of scene (2D in this case)
• for first observer position
  – draw nearby nodes (yellow) as geometry
  – render distant nodes (red) to RGB images (black)
  – composite images together
• as observer moves
  – if disparity exceeds a threshold, rerender image
Light field rendering
[Levoy & Hanrahan, SIGGRAPH 1996]

- must stay outside convex hull of the object
- like rebinning in computed tomography
The plenoptic function

Radiance as a function of position and direction in a static scene with fixed illumination

• for general scenes
  5D function
  \[ L ( x, y, z, \theta, \phi ) \]

• in free space
  4D function
  “the (scalar) light field”
The free-space assumption

• applications for free-space light fields
  – flying around a compact object
  – flying through an uncluttered environment
Some candidate parameterizations

Point-on-plane + direction

\[ L(x, y, \quad ) \]

• convenient for measuring luminaires
More parameterizations

Chords of a sphere

\[ L ( \quad , \quad ) \]

• convenient for spherical gantry
• facilitates uniform sampling
Two planes ("light slab")

- uses projective geometry
  - fast incremental display algorithms

\[ L( u, v, s, t ) \]
Creating a light field

- off-axis (sheared) perspective views
A light field is an array of images
Displaying a light field

foreach x,y
    compute u,v,s,t
    I(x,y) = L(u,v,s,t)
Devices for capturing light fields: Stanford Multi-Camera Array

- cameras closely packed
  - high-X imaging
  - synthetic aperture photography
- cameras widely spaced
  - video light fields
  - new computer vision algorithms
The BRDF kaleidoscope
[Han et al., SIGGRAPH 2003]

- discrete number of views
- hard to capture grazing angles
- uniformity?
Light field morphing
[Zhang et al., SIGGRAPH 2002]

UI for specifying feature polygons and their correspondences

- feature correspondences = 3D model
Autostereoscopic display of light fields
[Isaksen et al., SIGGRAPH 2000]

- image is at focal distance of lenslet (collimated rays)
- spatial resolution $\sim$ # of lenslets in the array
- angular resolution $\sim$ # of pixels behind each lenslet
- each eye sees a different sets of pixels (stereo)
End-to-end 3D television

[Matusik et al., SIGGRAPH 2005]

- 16 cameras, 16 video projectors, lenticular lens array
- spatial resolution $\sim$ # of pixels in a camera and projector
- angular resolution $\sim$ # of cameras and projectors
- horizontal parallax only
Why didn’t IBR take over the world?

• warping and rendering range images is slow
  – pixel-sized triangles are inefficient
  – just as many pixels need to be touched as in normal rendering

• arms race against improvements in 3D rendering
  – level of detail (LOD)
  – culling techniques
  – hierarchical Z-buffer
  – etc.

• visual artifacts are objectionable
  – not small and homogeneous like 3D rendering artifacts