Image-Based Rendering

What is it?

• Still a difficult question to answer

• Uses images (photometric info) as key component of model representation
What’s Good about IBR

Model acquisition

- Detailed 3D geometry difficult to construct
- Images relatively easy to acquire

Model quality

- If you want photo-realistic output, start with photo-realistic input

Rendering complexity

- dependent on resolution of images and screen, not 3D geometry
Defining the Problem

Plenoptic function

\[ p = P(\theta, \phi, \lambda, V_x, V_y, V_z, t) \]

“Given a set of samples (complete or incomplete) from the plenoptic function, the goal of image-based rendering is to generate a continuous representation of that function”
Accomplishing IBR

Sampling

Reconstruction

Re-sampling
Some IBR Algorithms

View Interpolation
- Chen/Williams, *SIGGRAPH 93*

Post-Rendering Warping
- Mark et al., *I3DG 95*

QuickTime VR
- Chen, *SIGGRAPH 95*

Plenoptic Modeling
- McMillan/Bishop, *SIGGRAPH 95*

Light Fields (discuss tomorrow)
- Levoy/Hanrahan, *SIGGRAPH 96*
View Interpolation

Sample a number of depth images

Build adjacency graph of images

- nodes are images
- edges are mappings between them

Interpolate pixels to construct in-between images (i.e. - 3D image warping)
View Interpolation Examples

Fig. 2  Extents of pixel movement for 2D viewpoint motions:  
(a) viewpoints parallel to the viewing plane,  
(b) viewpoints parallel to the ground.  (Source pixels are in the lower right corner of each extent.)

Fig. 5  (a) Holes from one source image, (b) holes from two source images,  
(c) holes from two closely spaced source images, (d) filling the holes with interpolation.

Sampling

Range camera

Overlapping images

Camera rotation about tripod

Conventional 3D rendering
Correspondence Mappings

Use “forward mapping” algorithm

Apply 4x4 transformation to source pixels to determine location in destination frame

Approximate transformation by per-pixel linear interpolation

For each graph edge, construct two mappings, one for each direction
Difficulties of Forward Mappings

Mappings are “many to one”

Some destination pixels may be multiply-covered

Some destination pixels may not be covered at all
Dealing with Difficulties

Multiple coverage

- Z-buffering
- back-to-front traversal

Holes

- alleviated by warping multiple images
- hole-filling interpolation possible
Post-Rendering Warping

Render conventional 3D graphics images slowly, on-the-fly

Apply 3D image warping to generate in-between images quickly

Use view prediction to guess future view to start rendering conventionally
Post-Rendering Warping Example

Plate 1: A typical derived frame produced by our test-bed. The reference frames were generated at 5 frames/sec, and the average per-axis position prediction error was 5.0 cm.

Plate 3: A particularly bad reference frame produced by our test-bed. Some areas of the image near the door were occluded in both reference frames, mostly because of prediction error.

Choose key eye positions to sample

Capture/create cylindrical panoramic image for each eye position

Allow users to “hop” among eye positions and rotate/zoom at each position

• Fairly simple computation to map panorama to screen
Quick-Time VR Examples

from Chen, “Quick-Time VR: An Image-Based Approach to Virtual Environment Navigation,” *Proceedings of SIGGRAPH 95*, page 38

Johns Hopkins Department of Computer Science
Course 600.456: Rendering Techniques, Professor: Jonathan Cohen
Plenoptic Modeling

Provides mathematical framework for analyzing IBR algorithms with respect to plenoptic function

Presents algorithm for visibility-preserving (back-to-front) traversal in 3D image warping

Develop system for full 3D image warping of cylindrical panoramas
Plenoptic Modeling Examples

from McMillan and Bishop, “Plenoptic Modeling: An Image-Based Rendering System”, page 45.