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

3D Images (Depth Images)

Image has x and y resolution
Each sample has depth as well as color
### Acquiring 3D Images (Sampling)

- Range camera
- Overlapping images
- Camera rotation about tripod
- Conventional 3D rendering

### Rendering 3D Images

Think of each sample as a 3D point
- Transform each point according to viewing parameters
- Kind of slow
- Actually, point-based rendering has become more popular/feasible in recent years...

3D image warping
- Don’t transform each point independently
- Take advantage of the x and y coherence of the image representation

### 3D Image Warping as Forward Mapping

- Depth image is the source
- Generated image is the destination
- Very regular source image is warped to destination image
  - No longer regular in destination image
    - Similar to problem in texture mapping
  - Ultimately need to get regularly sampled destination

### 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
    - Often splat the samples to the screen with some appropriate splat size
    - Sometimes use samples as triangle vertices and draw warped triangles

### Some 3D Image Warping Based IBR Algorithms

- **View Interpolation**
  - Chen/Williams, *SIGGRAPH* 93
- **Post-Rendering Warping**
  - Mark et al., *I3DG* 97
- **QuickTime VR**
  - Chen, *SIGGRAPH* 95
- **Plenoptic Modeling**
  - McMillan/Bishop, *SIGGRAPH* 95
- **Layered Depth Images**
  - Shade et al., *SIGGRAPH* 98
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)
  • Think of image morphing, but in 3D

View Interpolation Examples

Fig. 2: Excerpts of scene movement for 2D receptive window, as compared to the moving plane.
Fig. 3: Excerpt of scene movement for 3D receptive window.


Correspondence Mappings

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

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 technique. The reference frames were generated at 3 frames/sec, and the average per-vertex position prediction error was 5.5 cm.
Plate 3: A particularly bad reference frame produced by our technique. Some areas of the image were occluded in both reference frames, resulting in prediction error.

from Mark, McMillan, and Bishop, “Post-Rendering 3D Warping”, Proceedings of 1997 Symposium on Interactive 3D Graphics

Video


John Hopkins Department of Computer Science
Course 480.490: Rendering Techniques, Professor: Jonathan Cohen
**Quick-Time VR**

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
Actually, doesn’t use depth images

---

**Quick-Time VR Examples**

![Quick-Time VR Examples](image)

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

---

**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
  • Based on following optical flow
Develop system for full 3D image warping of cylindrical panoramas

---

**Plenoptic Modeling Examples**

![Plenoptic Modeling Examples](image)

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

---

**Layered Depth Images**

Allow multiple samples per pixel in depth image
  • Each sample at different depth
  • All the front-most samples are first “layer”, etc.
Alleviates exposure artifacts
Often small average number of samples per pixel can remove most of the artifacts
Reduces the redundancy of multiple depth images

---

**Videos**