Camera Model Design

A Novel Paradigm for Graphics, Visualization, and Vision

Voicu Popescu

Computer Graphics and Visualization Laboratory

PURDUE UNIVERSITY

Planar pinhole cameras dominate

Planar pinhole camera limitations

- Field of view
- Rays pass through common point

Panoramic pinhole cameras

Panoramic pinhole cameras

Panoramic pinhole cameras

How about pinhole constraint?

- Useful
  - Any pinhole image can be re-sampled to planar pinhole camera image
  - Simple, fast SW and HW implementation
- Restrictive

PURDUE UNIVERSITY

7

Why are we stuck on pinholes?

- Misconceptions
  - Such as the belief that non-pinhole camera (NPHC) images are not useful
  - Or that rendering with an NPHC is inherently inefficient

PURDUE UNIVERSITY

8

NPHC images *are* useful

- Coherence is sufficient for us to make sense of an image
- Not all images in nature are pinhole images
  - Reflections, refractions
- NPHC images can be used as intermediate images, in IBR

PURDUE UNIVERSITY

9

NPHCs *can* be efficient

- Provide fast projection and
- Scene can be rendered with efficient feed-forward pipeline, w/ HW support

PURDUE UNIVERSITY

10

Freedom

- There are many, many efficient NPHCs possible

PURDUE UNIVERSITY

11

Camera Model Design (CMD) Paradigm

- Instead of using one of the few “off the shelf” cameras, the camera model should be designed to best fit the needs of the application, and it should be dynamically optimized according the scene or data set of interest.

PURDUE UNIVERSITY

12

CMD: two steps

- Define rays of interest according to app.
  - General ray: locus of 3D points projecting at a given pixel
- Implement rays such that resulting camera has efficient projection

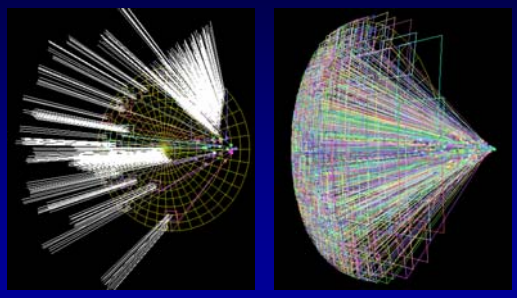
PURDUE UNIVERSITY

Example 1: Sample-Based Cameras (SBCs) for feed-forward reflections

- Step 1: Rays of interest
  - Reflected rays
- Step 2: Efficient Projection
  - Reflected rays $O(10^5)$ are replaced by simple cameras $O(10^3)$
  - Simple cameras stored at leafs of BSP tree

PURDUE UNIVERSITY

Example 1: SBCs



PURDUE UNIVERSITY

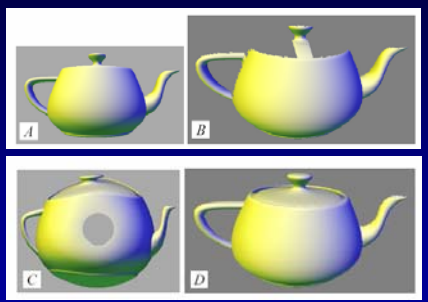
SHOW VIDEO

Example 2: occlusion cameras for disocclusion-error-resistant reference images

- Step 1: Rays of interest
  - Reference view rays
  - Rays that go around occluder to gather barely hidden samples
- Step 2: Efficient Projection
  - PPHC projection followed by 3D distortion

PURDUE UNIVERSITY

Example 2: occlusion cameras

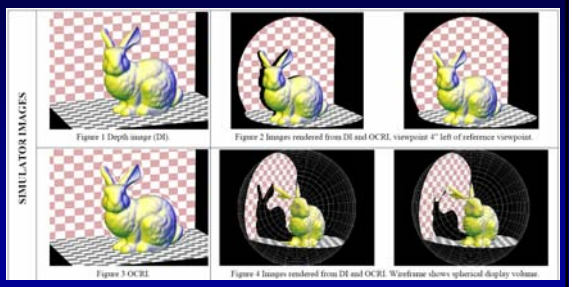


PURDUE UNIVERSITY

Example 2: occlusion cameras

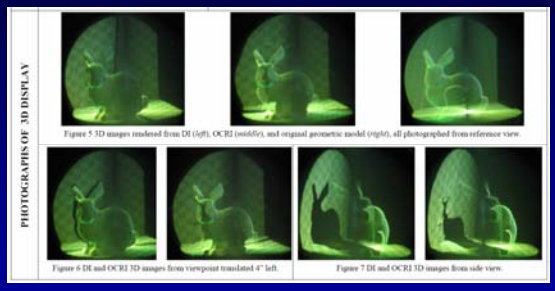
PURDUE UNIVERSITY

Example 2: occlusion cameras



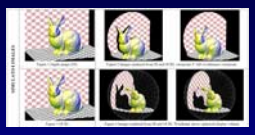
PURDUE UNIVERSITY

Example 2: occlusion cameras

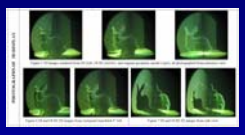


PURDUE UNIVERSITY

Example 2: occlusion cameras



SHOW
VIDEO



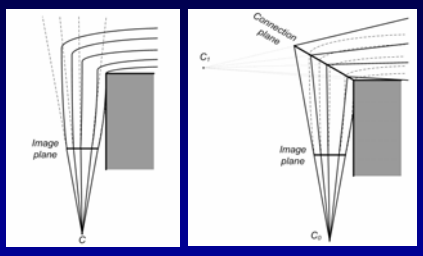
PURDUE UNIVERSITY

Example 3: graph cameras for comprehensive single-image visualization at interactive rates

- Step 1: Rays of interest
  - Start with PPHC and design rays at will
  - Each ray is C^0 continuous
- Step 2: Efficient projection
  - Each ray is piecewise linear (chain of segmnts.)
  - Segments grouped in *general* pinhole cameras

PURDUE UNIVERSITY

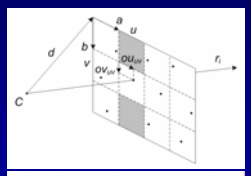
Example 3: graph cameras

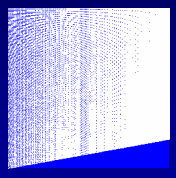


PURDUE UNIVERSITY

General pinhole camera

- Gathers any subset of 2D ray space





$$C+d+au+bv+ax_{uv}+by_{uv}$$

PURDUE UNIVERSITY

Example 3: graph cameras

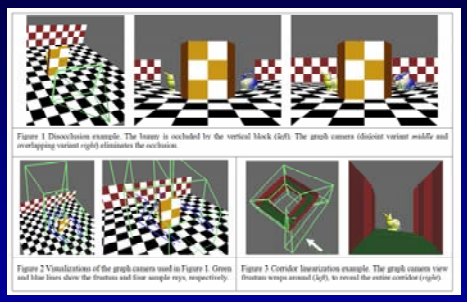


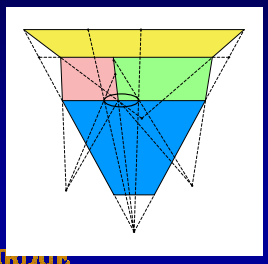
Figure 1 Disocclusion example. The bunny is occluded by the vertical block (left). The graph camera (disjoint variant *middle* and overlapping variant *right*) eliminates the occlusion.

Figure 2 Visualizations of the graph camera used in Figure 1. Green and blue lines show the frustum and four sample rays, respectively.

Figure 3 Corridor linearization example. The graph camera view frustum wraps around (*left*), to reveal the entire corridor (*right*).

PURDUE UNIVERSITY

Example 3: graph cameras



PURDUE UNIVERSITY

Example 3: graph cameras

SHOW VIDEO

PURDUE UNIVERSITY

Future work

- More camera families for other problems
- Investigate physical implementations

PURDUE UNIVERSITY

p_2 p_1

p_6 p_5 p_7