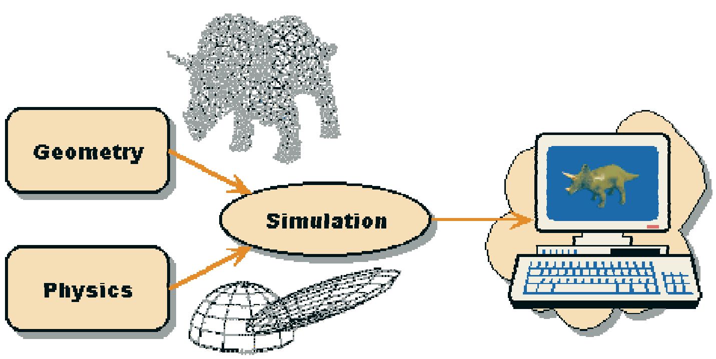
Image-Based Rendering Using Image Warping

Leonard McMillan
LCS Computer Graphics Group
MIT

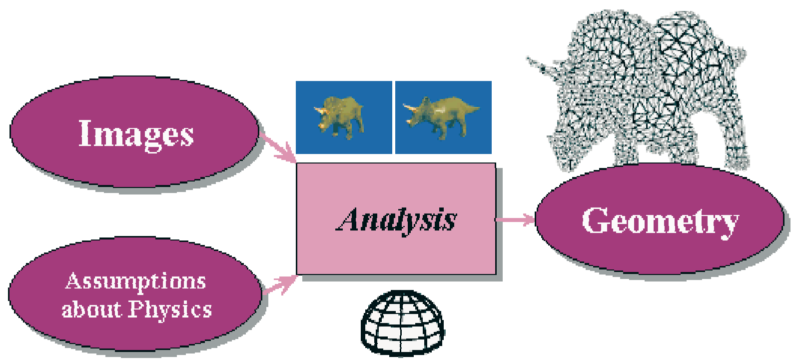
Conventional 3-D Graphics

- ✔ Simulation



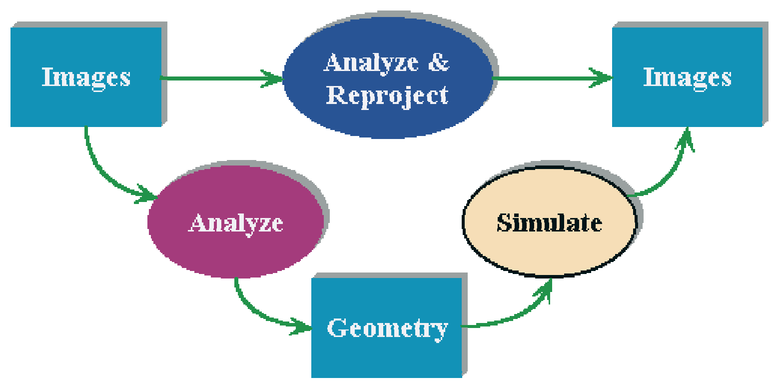
Computer Vision

✔ Analysis

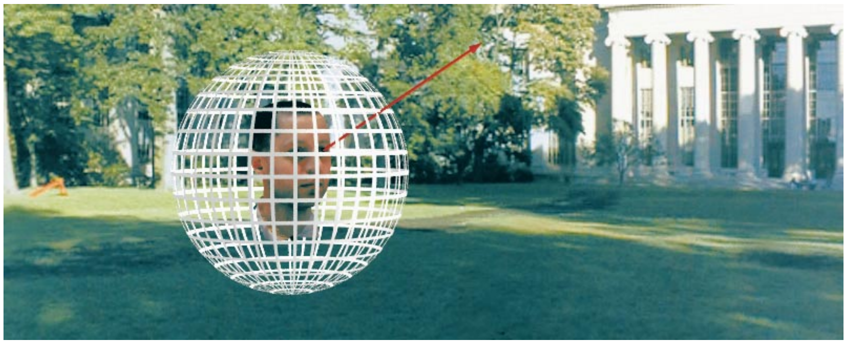


The Image-Based Approach

✔ Transformation



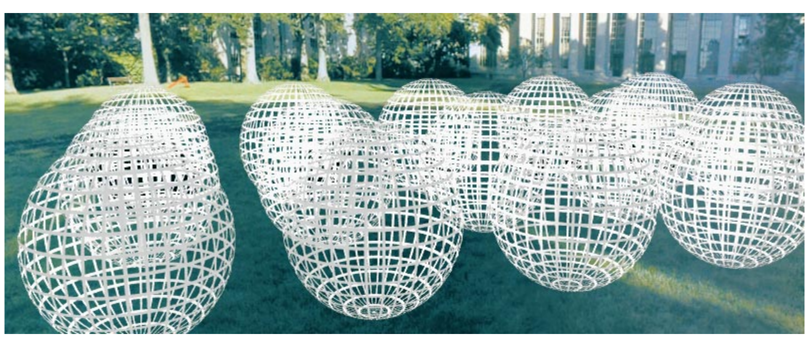
Images as a Collection of Rays



An image is a subset of the rays seen from a given point
- this "space" of rays occupies two dimensions

The Plenoptic Function

✔ The set of rays seen from all points ...



$$p = P(\theta, \phi, x, y, z, \lambda, t)$$

Image-based rendering is about

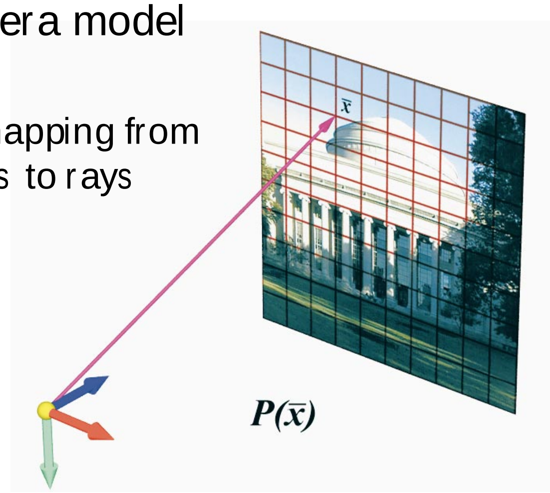
…reconstructing a plenoptic function from a set of samples taken from it.

✔ Ignoring time, and selecting a discrete set of wavelengths gives a 5-D plenoptic function

Where to Begin?

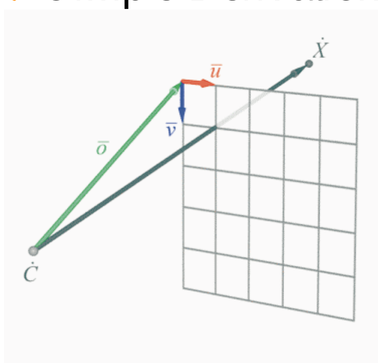
✔ Pinhole camera model

- Defines a mapping from image points to rays in space



Mapping from Rays to Points

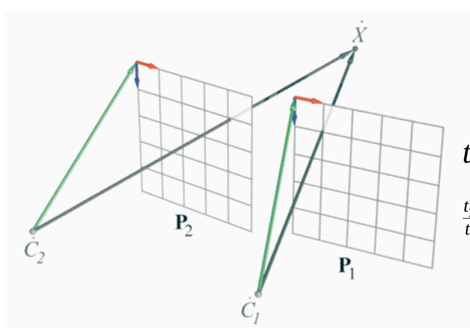
✔ Simple Derivation



$$P = \begin{bmatrix} u_x & v_x & o_x \\ u_y & v_y & o_y \\ u_z & v_z & o_z \end{bmatrix}$$

$$\dot{X} = \dot{C} + t\, P\, \vec{x}$$

Correspondence



$$\dot{C}_2 + t_2 P_2 \vec{x}_2 = \dot{C}_1 + t_1 P_1 \vec{x}_1$$

$$t_2 P_2 \vec{x}_2 = \dot{C}_1 - \dot{C}_2 + t_1 P_1 \vec{x}_1$$

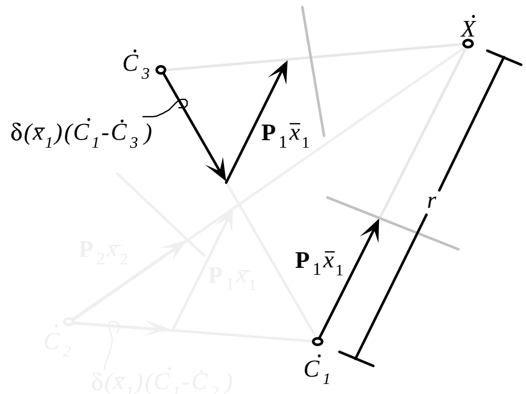
$$t_2 \vec{x}_2 = P_2^{-1}\left(\dot{C}_1 - \dot{C}_2\right) + t_1 P_2^{-1} P_1 \vec{x}_1$$

$$\tfrac{t_2}{t_1} \vec{x}_2 = \tfrac{1}{t_1} P_2^{-1}\left(\dot{C}_1 - \dot{C}_2\right) + P_2^{-1} P_1 \vec{x}_1$$

$$\vec{x}_2 \doteq \underbrace{\tfrac{1}{t_1}}_{\delta} \underbrace{P_2^{-1}\left(\dot{C}_1 - \dot{C}_2\right)}_{\vec{e}_{21}} + \underbrace{P_2^{-1} P_1}_{H_{21}} \vec{x}_1$$

Planar Warping Equation

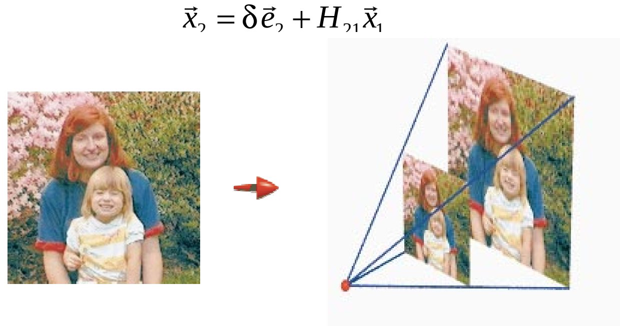
$$\vec{x}_2 \doteq \delta(\vec{x}_1) P_2^{-1}(\dot{C}_1 - \dot{C}_2) + P_2^{-1} P_1 \vec{x}_1$$



Resulting Warping Function

- ✔ A perturbed planar warp ...

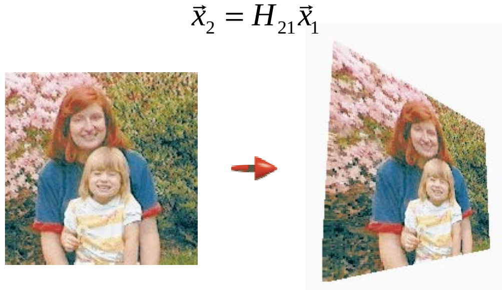
$$\vec{x}_2 = \delta \vec{e}_2 + H_{21} \vec{x}_1$$



Special Case

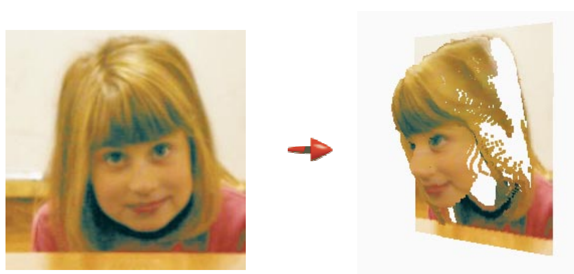
✔ A simple Planar warp

$$\vec{x}_2 = H_{21}\vec{x}_1$$



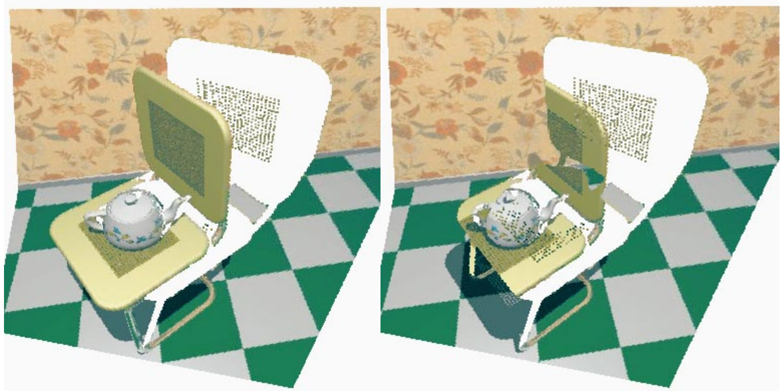
Warping in Action

✔ A 3D Warp



Visibility

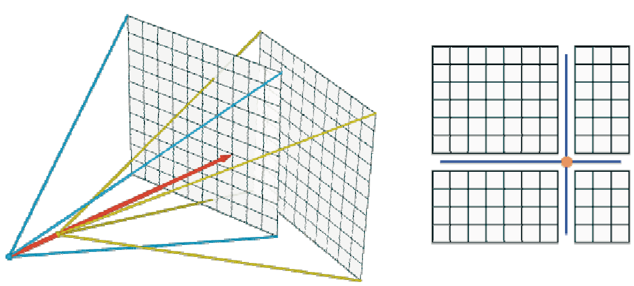
✔ The warping equation determines where points go…



… but that is not sufficient

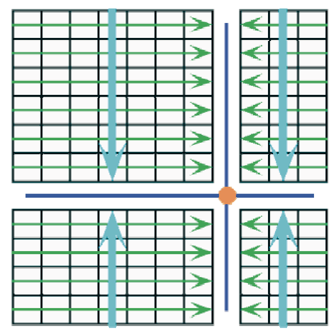
Partition Reference Image

✔ Project the *desired* center-of-projection onto the reference image



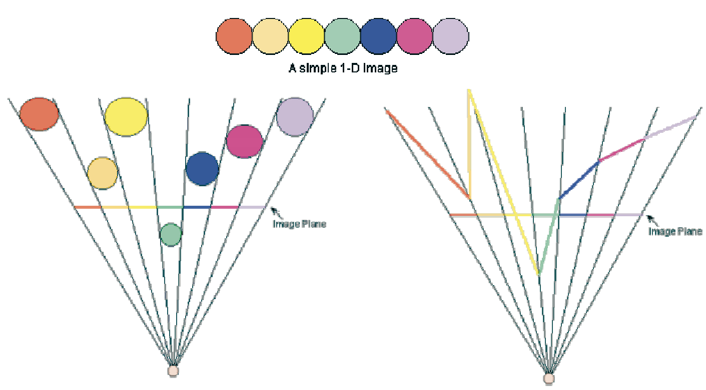
Enumeration

- Drawing toward the projected point guarantees an *occlusion compatible* ordering
- Ordering is consistent with a painter's algorithm
- Independent of the scene's contents
- Easily generalized to other viewing surfaces
- No auxiliary information required



Reconstruction

- Typical images are discrete, not continuous
- An image can be formed by different geometries



Gaussian Cloud Model

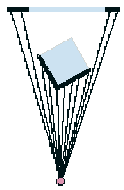
- Represents samples as Gaussian cloud densities
- Excessive exposure errors

Bilinear Patch Model

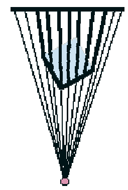
- Fits a bilinear patch through grid points in reference image
- Excessive occlusion errors

Comparison of Models

- Gaussian-Cloud Model



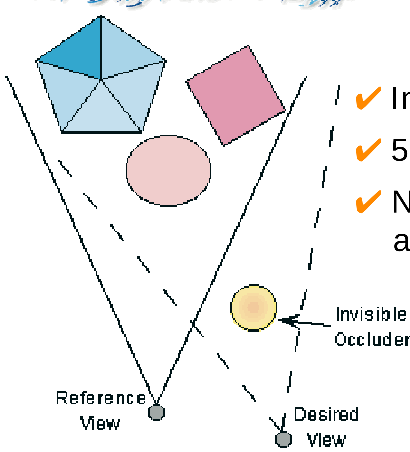
- Bilinear-Patch Model



Gaussian-Cloud Model:
- Excessive exposure errors
- Pinhole problems
- Generally preferred

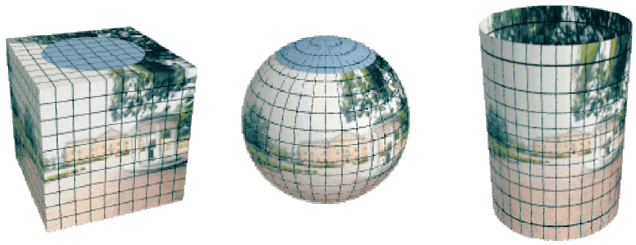
Bilinear-Patch Model:
- Excessive occlusion errors
- Rasterization H/W
- Difficult to navigate

Problems with Planar Cameras



- Invisible occluder problem
- 5 intrinsic parameters
- Non-uniform sampling of solid angle

Panoramic Cameras



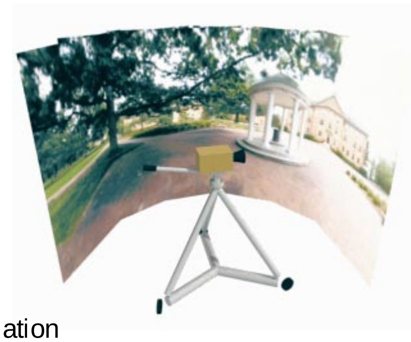
- ✔ Warping equation can be easily adapted
- ✔ Visibility algorithm works
- ✔ Nonlinear mapping functions

Examples

- ✔ Cylindrical camera

Constructing Panoramas

- ✔ Images are related by a projective transforms



$$\vec{x}_2 = H_{21}\vec{x}_1$$

- ✔ Optimization problem
  - maximize normalized correlation
  - minimize sum of squared error

Initial Guesses and Constraints

- ✔ Sum of angles is 2π
  - constrains focal length
- ✔ Skew of camera is near 0
- ✔ Aspect ratio near 1

Finding Disparity

- ✔ How to get it
  - 3-D laser scanners
  - Depth-from-stereo
  - Depth-from-motion
  - Depth-from-focus
  - Depth-from-light-fields
  - Manual layer segmentation
- ✔ How accurate must it be?

Visual Hulls

- ✔ Depth-from-silhouettes
- ✔ Simple computer vision methods
  - blue screening
  - image differencing
- ✔ Can be computed in image space

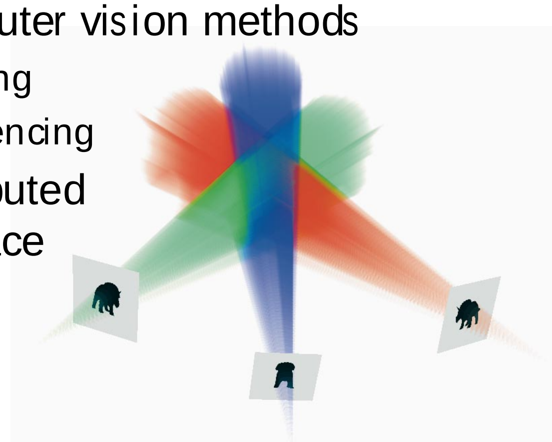
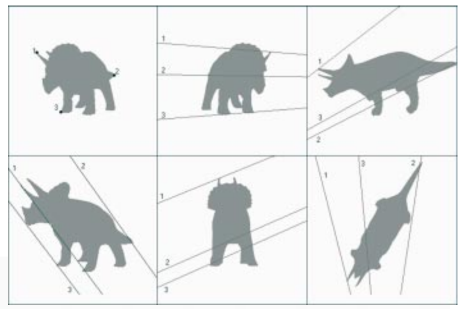
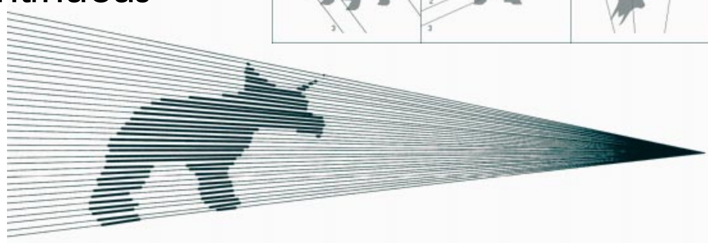


Image-based Visual Hulls

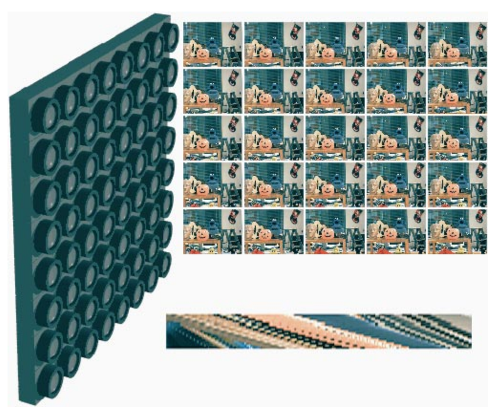
- ✔ Volume-like
- ✔ Self-consistent
- ✔ Discrete-discrete-continuous





Depth from Redundant Structure

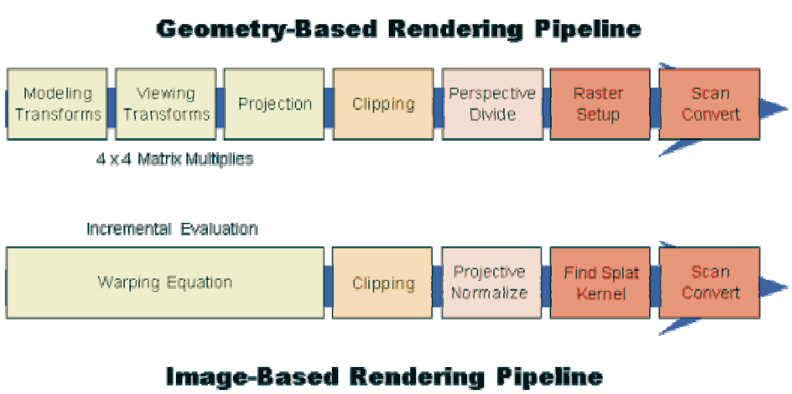
- ✔ Light fields for depth acquisition
  - Depth-from
    - stereo
    - motion
    - focus
    - silhouettes



Comparing Rendering Approaches

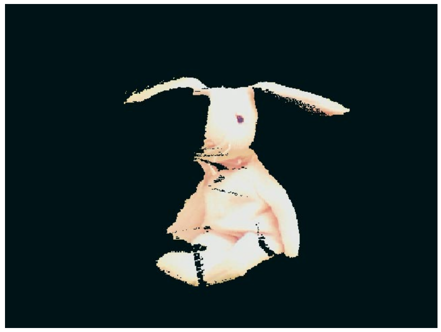
- Geometry Based
  - Forward Mapping (graphics pipeline)
  - Inverse Mapping (ray tracing)
- Image based
  - Greater spatial coherence
  - Lower depth complexity

Image-Based Pipeline



Forward-warping

✔ Single depth value per pixel



Forward-Mapped Visual Hull

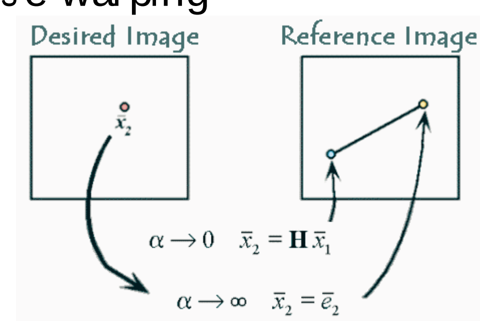
✔ Draws a line segment for each interval

Merging Forward Warps

- ✔ Draw textured line segments

Image-Based Ray Tracing

- ✔ Inverse warping



Linear Interpolation $x = a(1-t) + bt$

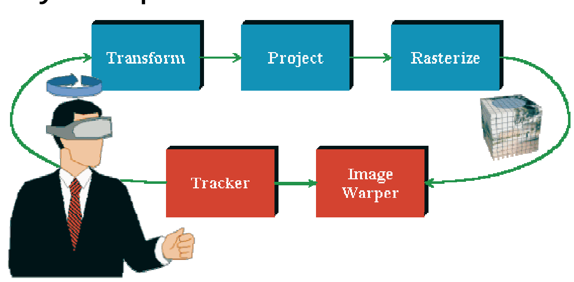
Harmonic Interpolation $x = a\left(\frac{\theta}{1-\theta}\right) + b\left(\frac{1}{1-\theta}\right)$

Algorithm Properties

- ✔Search is confined to a line
- ✔First intersection is closest point
- ✔Incremental line drawing
- ✔Reconstruction in reference image
- ✔Work is proportional to size of output image

Applications of IBR

- ✔IBR combined with traditional methods
- ✔Decouples rendering from interaction
- ✔Latency compensation



Conclusions

✔IBR provides

- new representations for 3D graphics
  - easy to acquire
  - allows efficient rendering
- scalable performance
  - depends on number of pixels rather than the number of geometric primitives
- amenable to HW acceleration