Intro to Image Based Rendering

Working with images

- realistic appearance
  - textures
- inflexible?

References

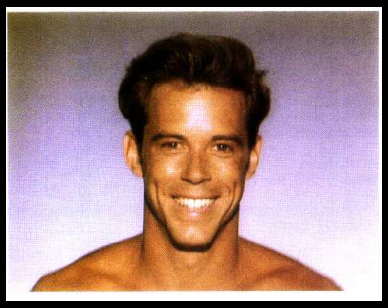
Two classic papers (available through the ACM web site: www.acm.org/dl)

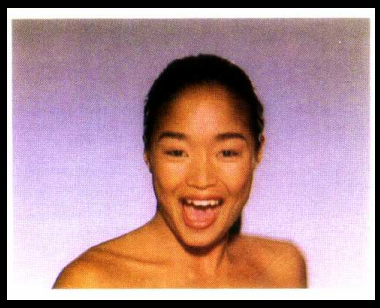
Thaddeus Beier and Shawn Neely, “Feature-Based Image Metamorphosis”, SIGGRAPH ‘92

Shenchang Eric Chen and Lance Williams, “View Interpolation for Image Synthesis,” SIGGRAPH ’93.

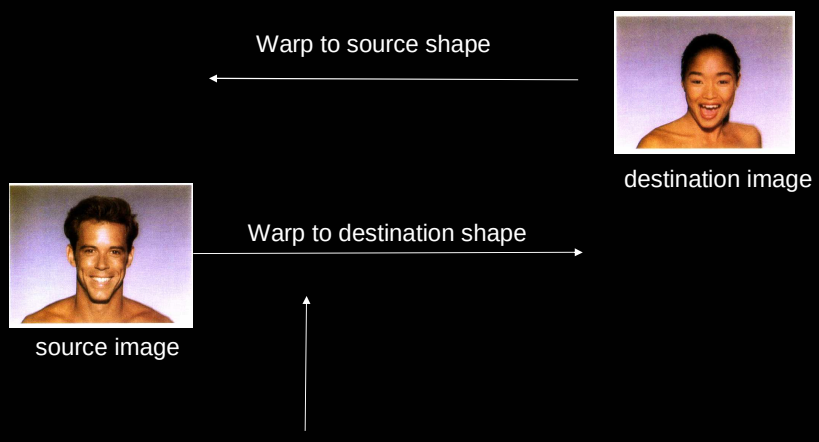
Feature Based Morphing

- Goal: create an animated morph from person A to person B (while they are singing!)



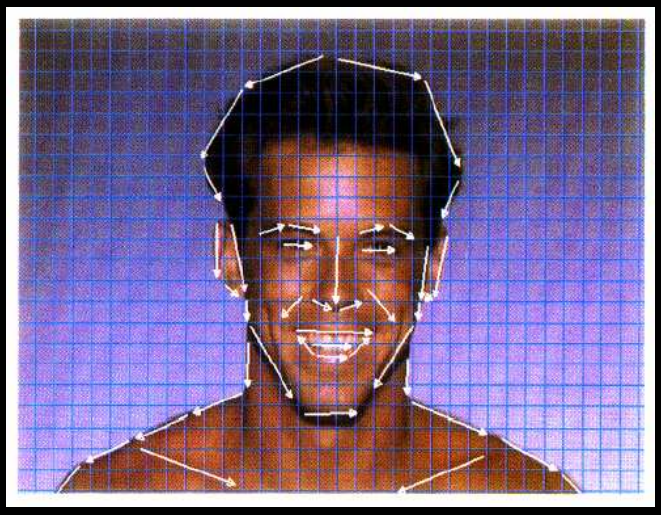


Main Idea



Controlling the Warp

- Idea: Feature lines establish correspondence



Feature Lines

- A feature line defines a coordinate system

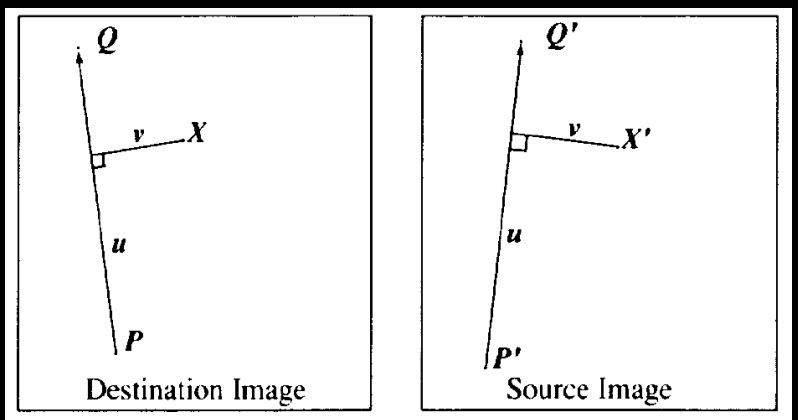


Figure 1: Single line pair

Feature Lines

- Changing the feature line changes how the image is drawn

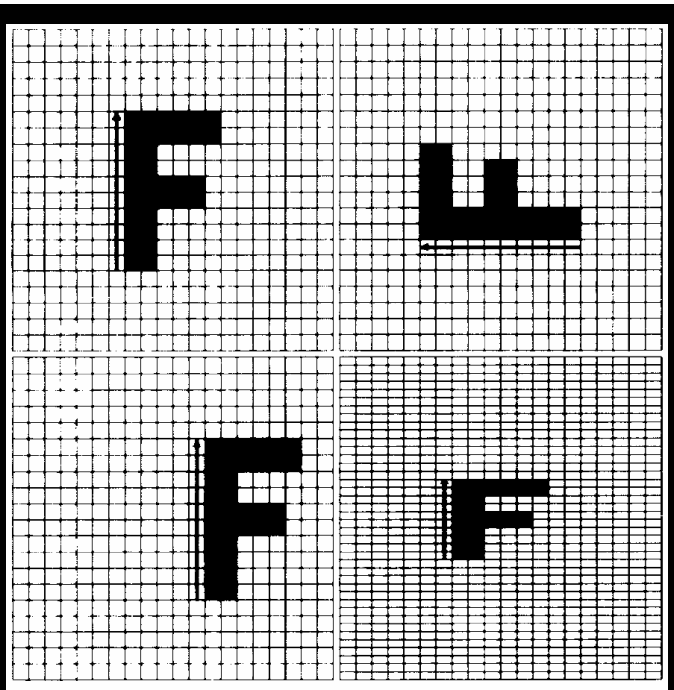


Figure 2: Single line pair examples

Filling in Pixels

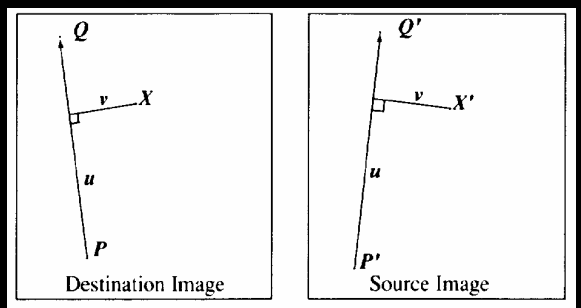
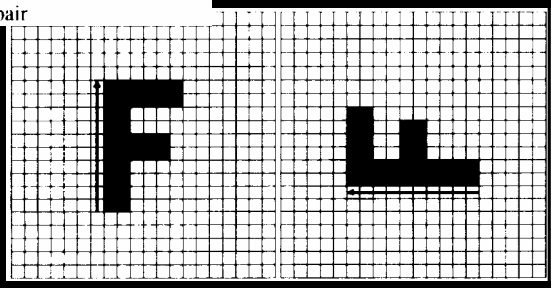


Figure 1: Single line pair



Multiple Feature Lines

- Weighted average of displacement vectors (destination to source pixel)

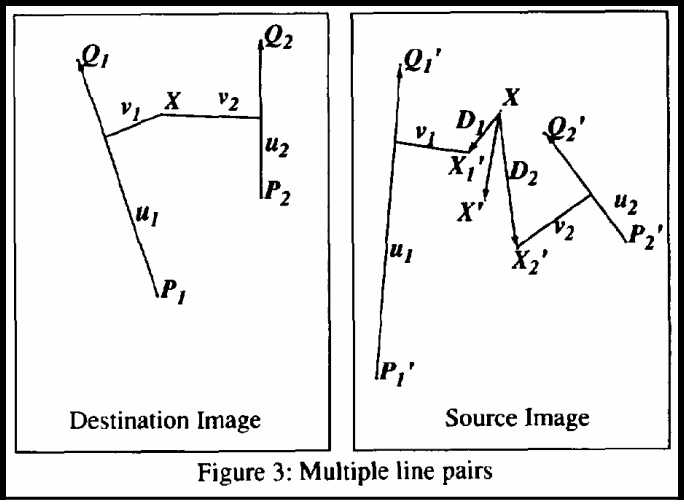


Figure 3: Multiple line pairs

Multiple Feature Lines Example

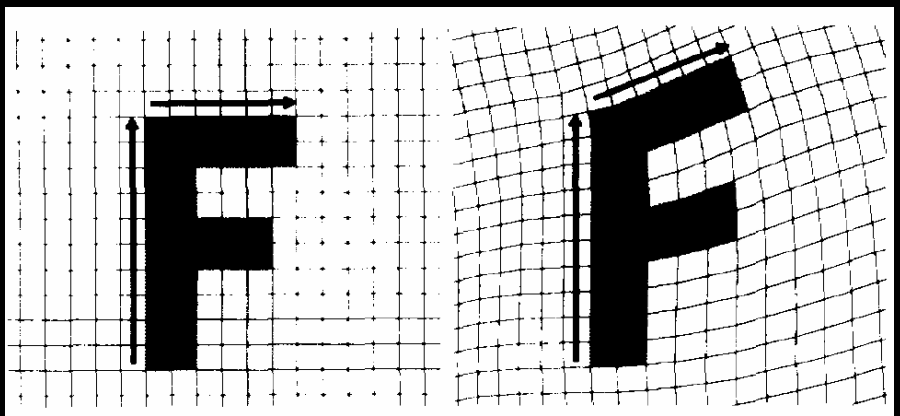


Figure 4: Multiple line pair example

Problems with the technique

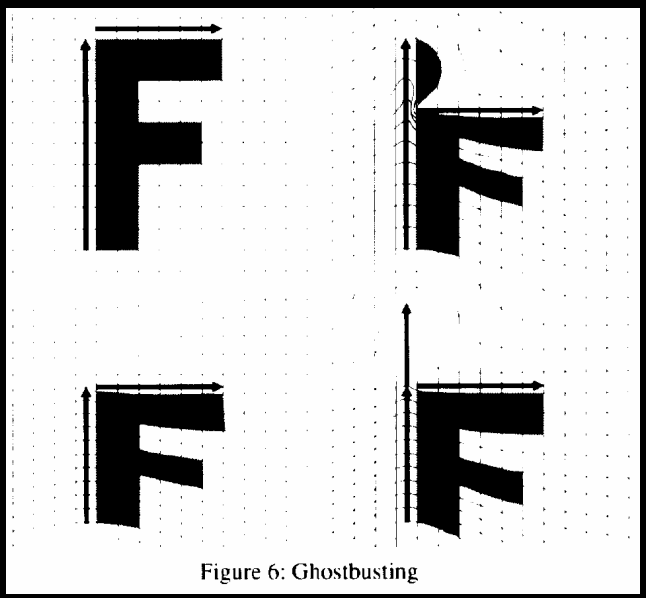
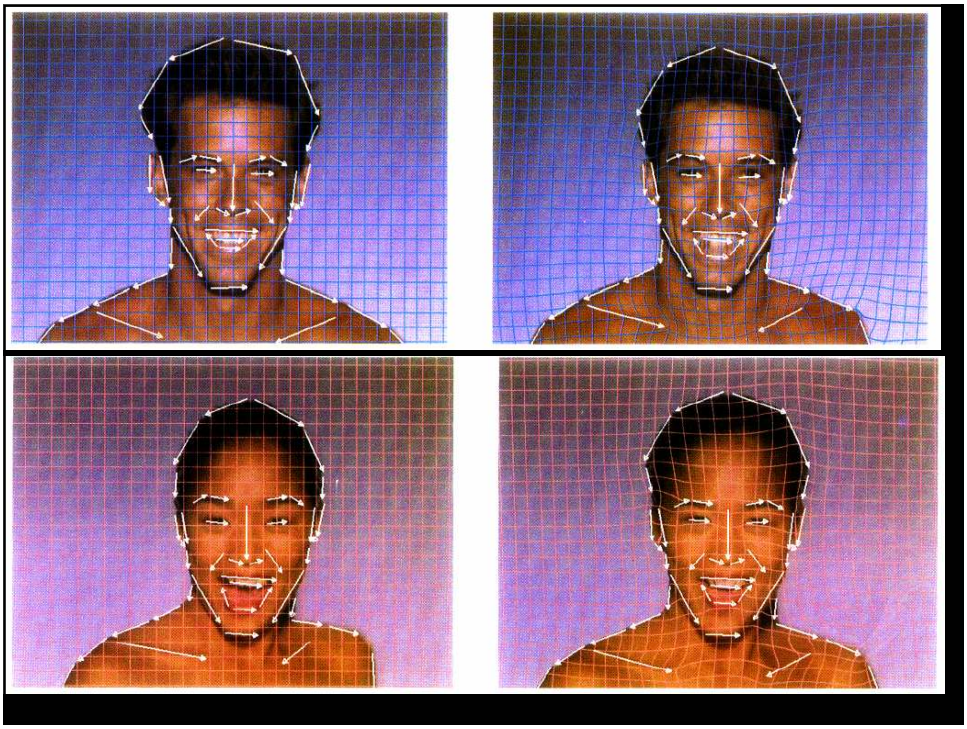
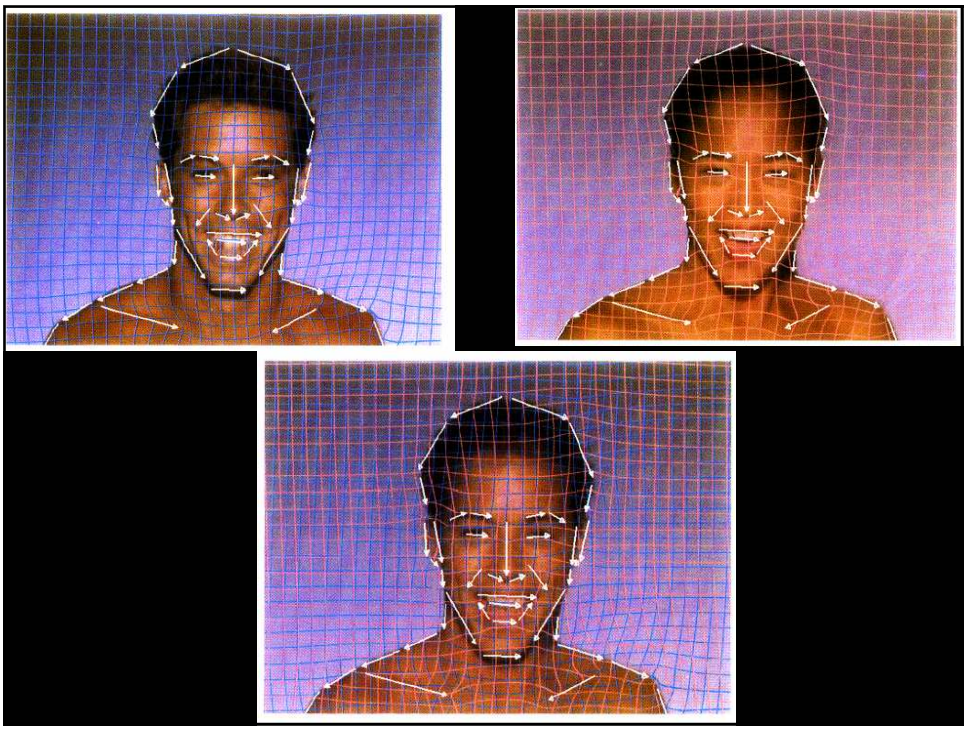
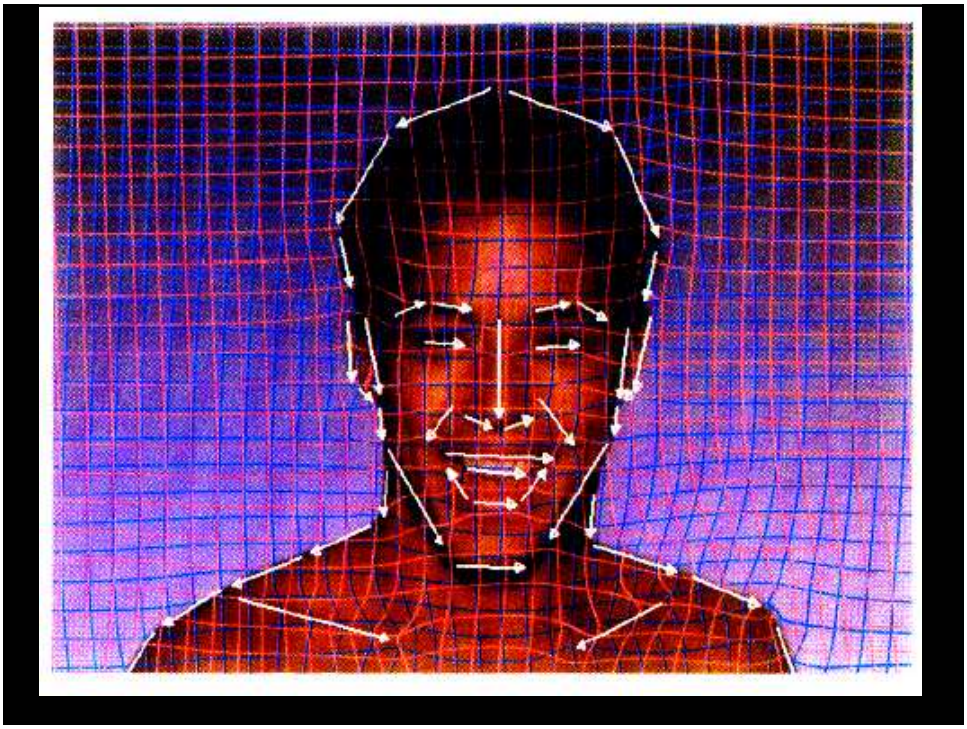
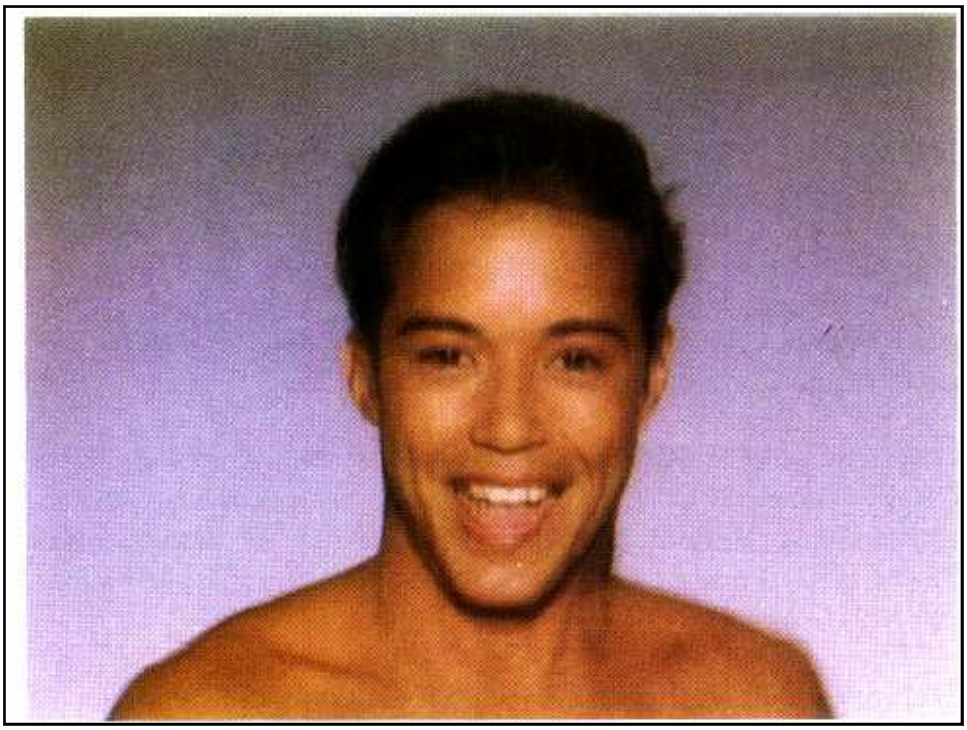


Figure 6: Ghostbusting









Video

View Interpolation

- Goal: Given photographs, generate new views

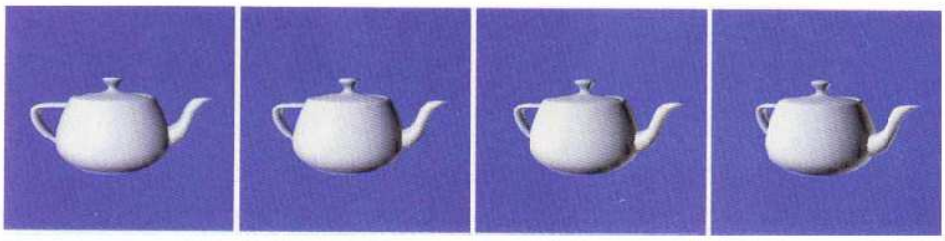
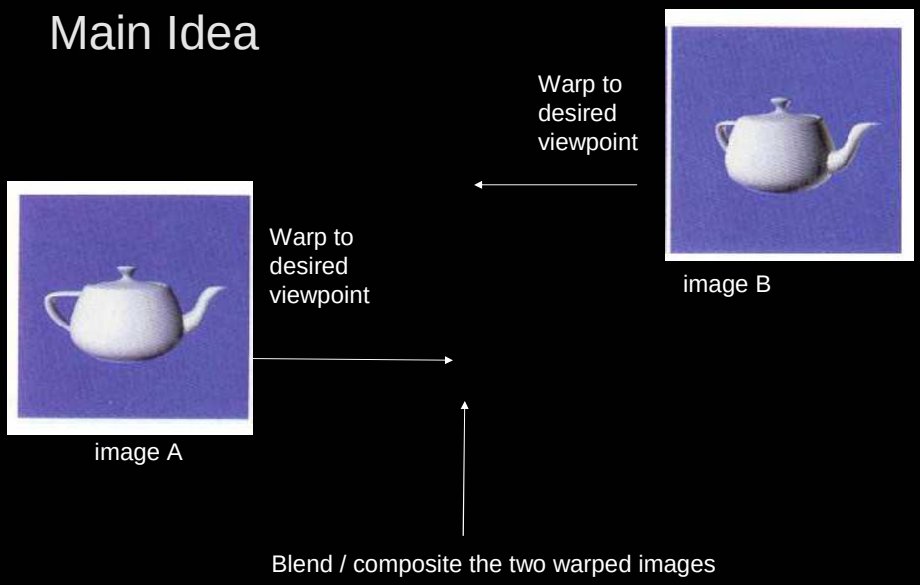


Fig. 14 Teapot images generated by interpolation (two middle ones).

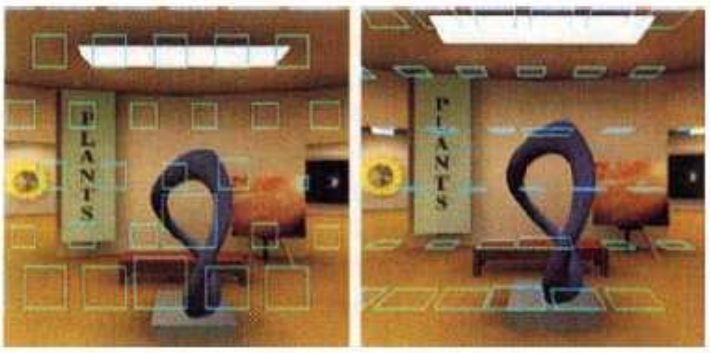
Main Idea



Controlling the Warp

- Key: we know the depth of every pixel
- Result is *dense correspondence* between images
- Forward mapping is used to create warped image
  - why?
- (Example on the board)

Boring pixel displacement fields



(a) (b)

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.)

More general displacement field

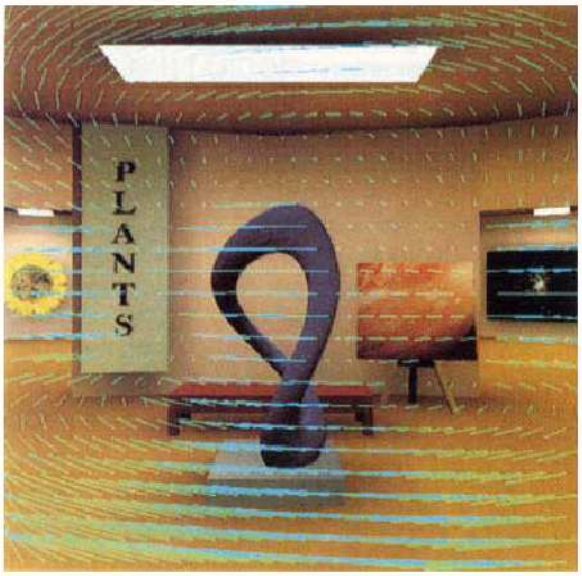


Fig. 1 Offset vectors for the camera motion in Figure 3.

Issues: Overlaps

- Many pixels map to one
  - large area becomes smaller → blend
  - multiple layers map to single point → z-buffer

Issues: Holes

- No pixels map to a pixel in the new image

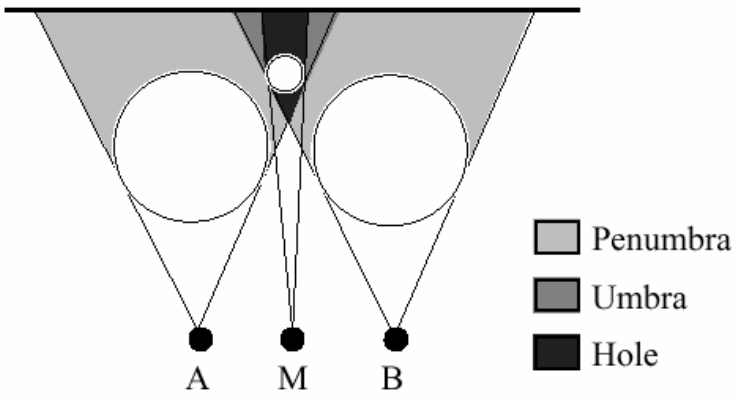


Fig. 4 Penumbra, umbra and hole regions

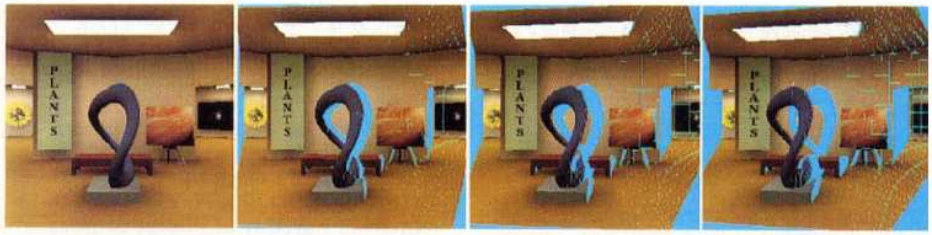
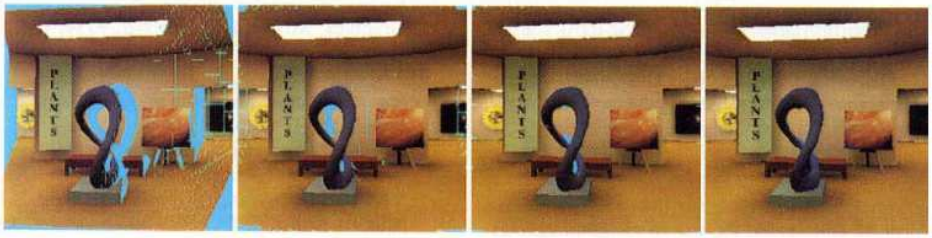
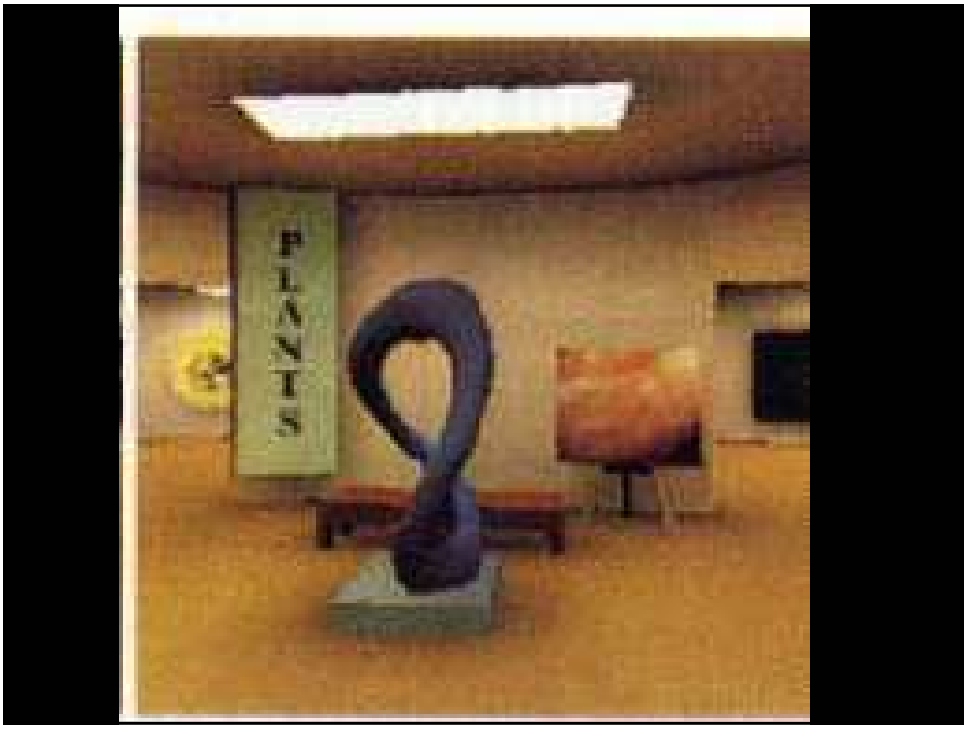


Fig. 3 *A source image viewed from a camera rotated to the right.*

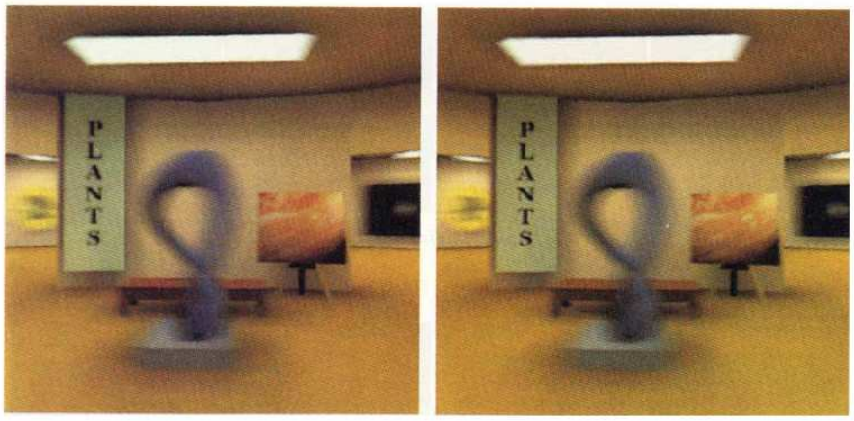


(a) *(b)* *(c)* *(d)*

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.*



Application: Motion Blur w/out Rerendering



(a) (b)

Fig.11 (a) Motion blur computed from source images, (b) motion blur computed from interpolated images[2].

Application: Soft shadows w/out rerendering

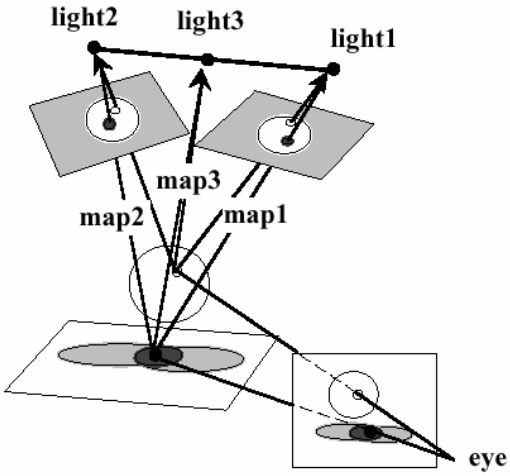


Fig. 12 Shadow buffer interpolation for a linear light source

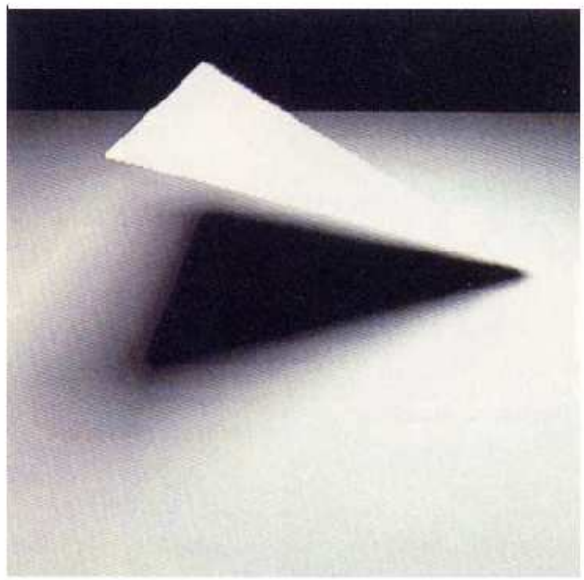


Fig. 13 Shadow from a rectangular area light computed with the shadow map interpolation.

Primary Limitation

- depth is required for every pixel!

Some more recent work:

- get the user involved in specifying depth
  - Debevec, SIGGRAPH ’96
  - http://athens.ict.usc.edu/Research
- use color and many cameras to get a good guess at depth .. use texture synthesis to refine that guess
  - Fitzgibbon et al., ICCV ’03
  - http://www.robots.ox.ac.uk/~awf/ibr/

Some more recent work:

- Fitzgibbon et al., ICCV ‘03

Input images

Results

Results 2

Announcements

- Homework due Thursday just before class
- Project 4 goes out Thursday
  - Texture synthesis!
  - James Hays will give the lecture on Thursday