Image Compositing and Morphing

Thomas Funkhouser
Princeton University
C0S 426, Fall 2000

Image Processing

- Quantization
  - Uniform Quantization
  - Random dither
  - Ordered dither
  - Floyd-Steinberg dither
- Pixel operations
  - Add random noise
  - Add luminance
  - Add contrast
  - Add saturation
- Filtering
  - Blur
  - Detect edges
- Warping
  - Scale
  - Rotate
  - Warp
- Combining
  - Composite
  - Morph

Image Processing

- Quantization
  - Uniform Quantization
  - Random dither
  - Ordered dither
  - Floyd-Steinberg dither
- Pixel operations
  - Add random noise
  - Add luminance
  - Add contrast
  - Add saturation
- Filtering
  - Blur
  - Detect edges
- Warping
  - Scale
  - Rotate
  - Warp
- Combining
  - Composite
  - Morph

Overview

- Image compositing
  - Blue-screen mattes
  - Alpha channel
  - Porter-Duff compositing algebra
- Image morphing
  - Specifying correspondences
  - Warping
  - Blending

Image Compositing

- Separate an image into “elements”
  - Render independently
  - Composite together
- Applications
  - Cel animation
  - Chroma-keying
  - Blue-screen matting

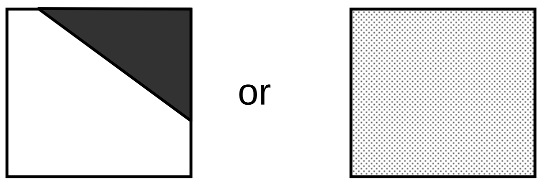
Dobkin meets Elvis

Blue-Screen Matting

- Composite foreground and background images
  - Create background image
  - Create foreground image with blue background
  - Insert non-blue foreground pixels into background

Alpha Channel

- Encodes pixel coverage information
  - $\alpha = 0$: no coverage (or transparent)
  - $\alpha = 1$: full coverage (or opaque)
  - $0 < \alpha < 1$: partial coverage (or semi-transparent)
- Example: $\alpha = 0.3$



Partial Coverage

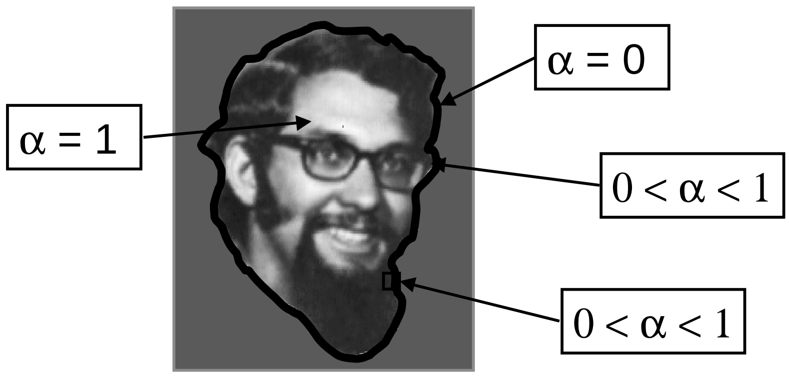
Semi-Transparent

Pixels with Alpha

- Alpha channel convention:
  - (r, g, b, α) represents a pixel that is α covered by the color $C = (r/\alpha, g/\alpha, b/\alpha)$
    - » Color comonents are premultiplied by α
    - » Can display (r,g,b) values directly
    - » Closure in composition algebra
- What is the meaning of the following?
  - (0, 1, 0, 1) = Full green, full coverage
  - (0, 1/2, 0, 1) = Half green, full coverage
  - (0, 1/2, 0, 1/2) = Full green, half coverage
  - (0, 1/2, 0, 0) = No coverage

Compositing with Alpha

- Controls the linear interpolation of foreground and background pixels when elements are composited



Semi-Transparent Objects

- Suppose we put A over B over background G
  - How much of B is blocked by A?

    α_A

  - How much of B shows through A?

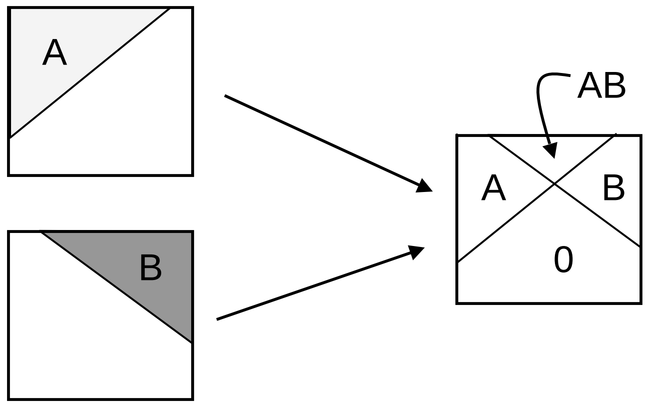
    $(1-\alpha_A)$

  - How much of G shows through both A and B?

    $(1-\alpha_A)*(1-\alpha_B)$

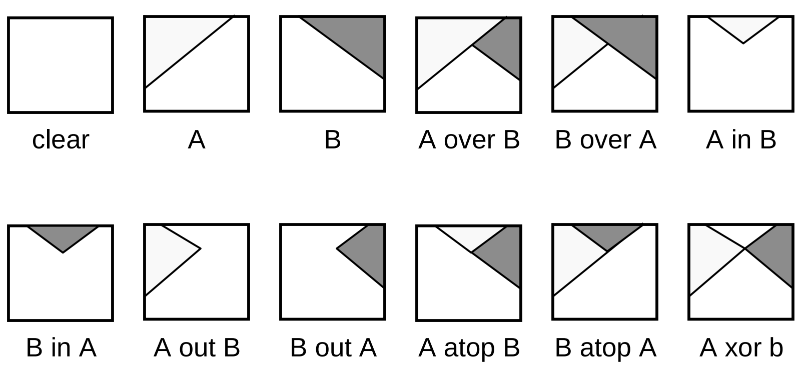
Opaque Objects

- How do we combine two partially covered pixels?
  - 3 possible colors (0, A, B)
  - 4 regions (0, A, B, AB)



Composition Algebra

- 12 reasonable combinations



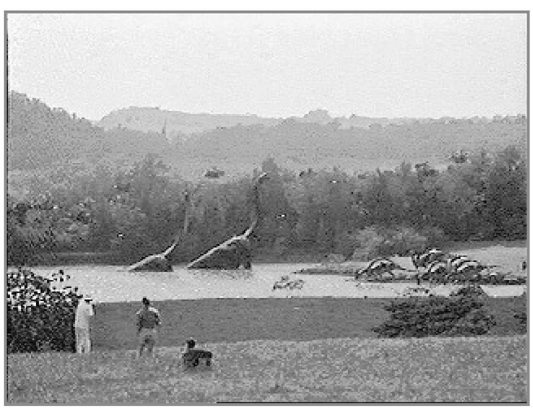
Porter & Duff '84

Over Operator

- For C_B and C_F, which are not premultiplied:
  - $C' = \alpha_B(1-\alpha_F)*C_B + \alpha_F*C_F$
  - $\alpha = \alpha_B(1-\alpha_F) + \alpha_F$
- For C'_B and C'_F, which are premultiplied:
  - $C' = (1-\alpha_B)*C'_F + C'_B$
  - $\alpha = \alpha_B(1-\alpha_F) + \alpha_F$

Assumption: coverages of B and F are uncorrelated for each pixel

Image Composition Example



Jurasic Park

Overview

- Image compositing
  - Blue-screen mattes
  - Alpha channel
  - Porter-Duff compositing algebra
- Image morphing
  - Specifying correspondences
  - Warping
  - Blending

Image Morphing

- Animate transition between two images

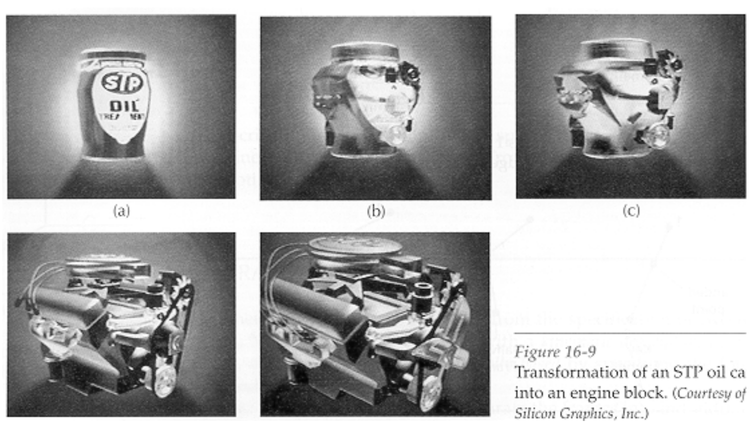


Figure 16-9
Transformation of an STP oil ca into an engine block. (*Courtesy of Silicon Graphics, Inc.*)

H&B Figure 16.9

Cross-Dissolving

- Blend images with "over" operator
  - alpha of bottom image is 1.0
  - alpha of top image varies from 0.0 to 1.0

$$blend(i,j) = (1\text{-}t)\ src(i,j) + t\ dst(i,j) \quad (0 \leq t \leq 1)$$

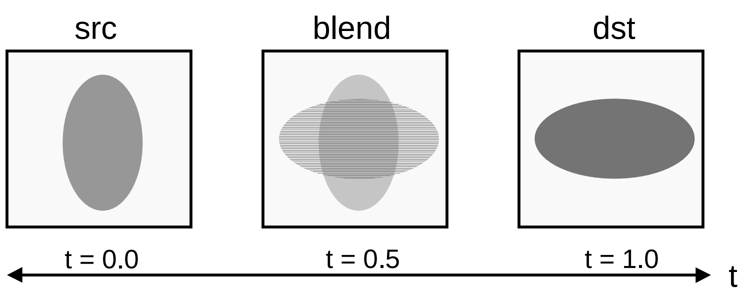


Image Morphing

- Combines warping and cross-dissolving

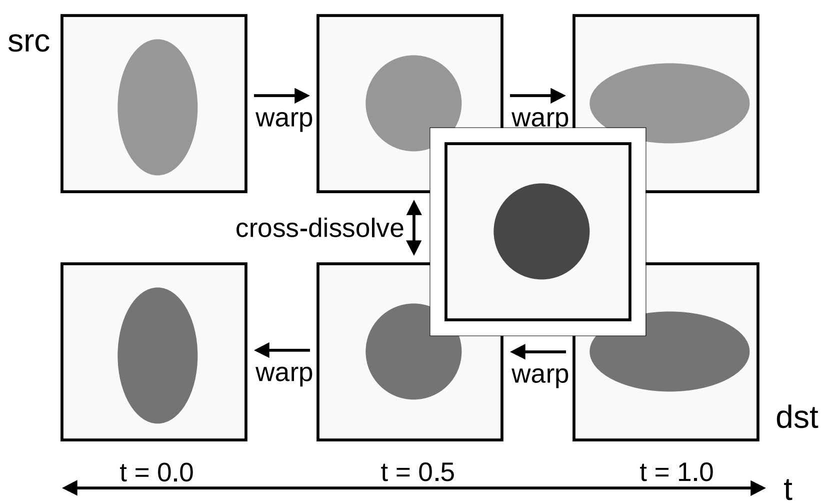
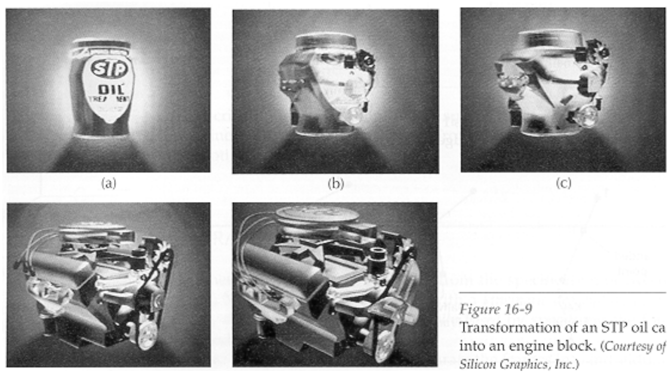


Image Morphing

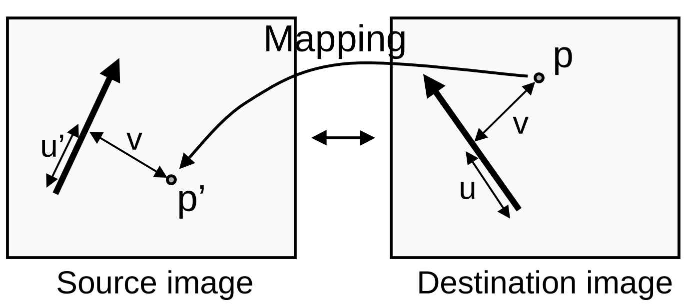
- The warping step is the hard one
  - Aim to align features in images



H&B Figure 16.9

Feature-Based Warping

- Beier & Neeley use pairs of lines to specify warp
  - Given p in dst image, where is p' in source image?



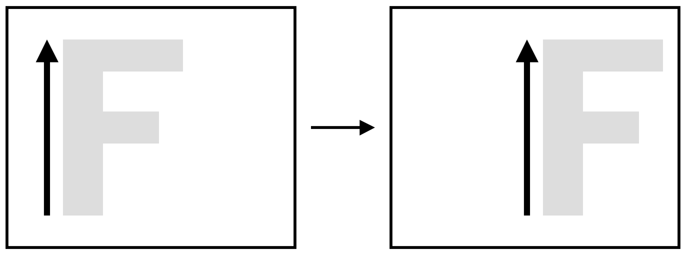
u is a fraction

v is a length (in pixels)

Beier & Neeley
SIGGRAPH 92

Warping with One Line Pair

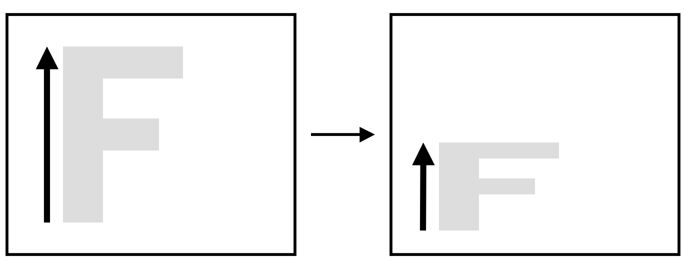
- What happens to the “F”?



Translation!

Warping with One Line Pair

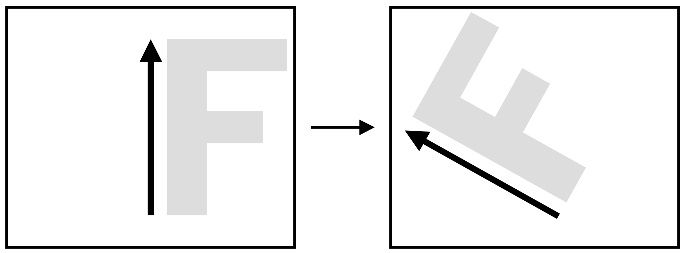
- What happens to the “F”?



Scale!

Warping with One Line Pair

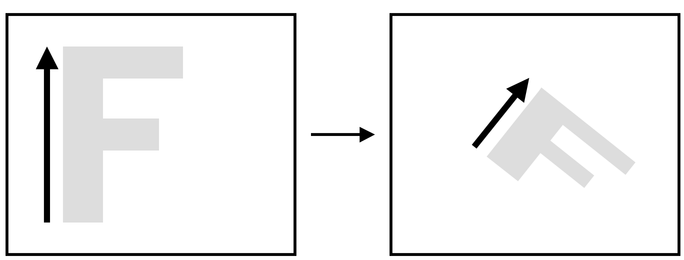
- What happens to the "F"?



Rotation!

Warping with One Line Pair

- What happens to the "F"?

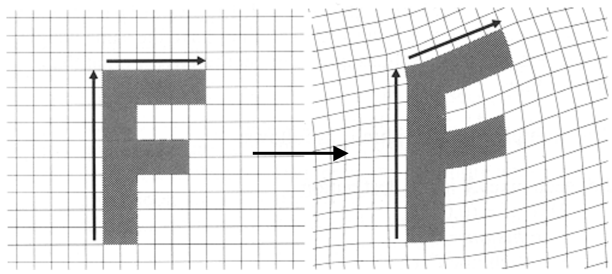


In general, similarity transformations

What types of transformations can't be specified?

Warping with Multiple Line Pairs

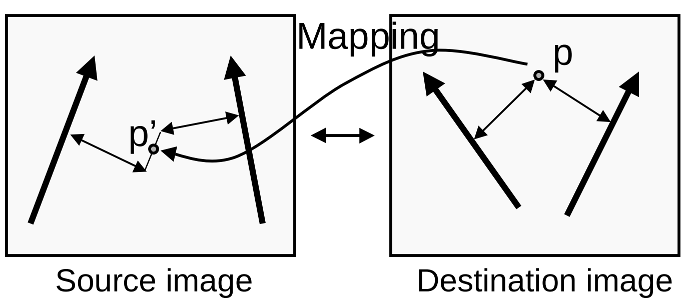
- Use weighted combination of points defined by each pair of corresponding lines



Beier & Neeley, Figure 4

Warping with Multiple Line Pairs

- Use weighted combination of points defined by each pair of corresponding lines



p' is a weighted average

Weighting Effect of Each Line Pair

- To weight the contribution of each line pair, Beier & Neeley use:

$$weight[i] = \left(\frac{length[i]^p}{a + dist[i]} \right)^b$$

Where:

- *length[i]* is the length of L[i]
- *dist[i]* is the distance from X to L[i]
- *a, b, p* are constants that control the warp

Warping Pseudocode

```
WarpImage(Image, L'[…], L[…])
begin
    foreach destination pixel p do
        psum = (0,0)
        wsum = 0
        foreach line L[i] in destination do
            p'[i] = p transformed by (L[i],L'[i])
            psum = psum + p'[i] * weight[i]
            wsum += weight[i]
        end
        p' = psum / wsum
        Result(p) = Image(p')
    end
end
```

Morphing Pseudocode

GenerateAnimation(Image$_0$, L$_0$[…], Image$_1$, L$_1$[…])
begin
    foreach intermediate frame time t do
        for i = 1 to number of line pairs do
            L[i] = line t-th of the way from L$_0$ [i] to L$_1$ [i]
        end
        Warp$_0$ = WarpImage(Image$_0$, L$_0$, L)
        Warp$_1$ = WarpImage(Image$_1$, L$_1$, L)
        foreach pixel p in FinalImage do
            Result(p) = (1-t) Warp$_0$ + t Warp$_1$

    end
end

Beier & Neeley Example

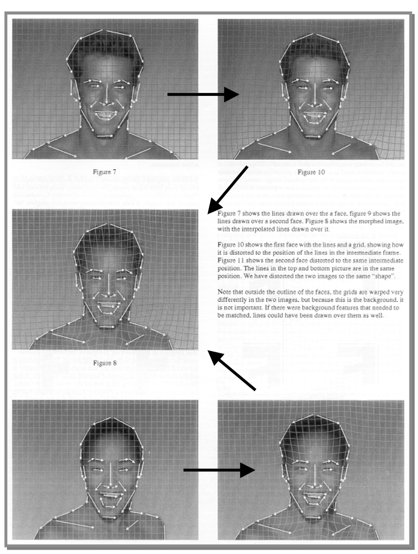
Image$_0$

Warp$_0$

Result

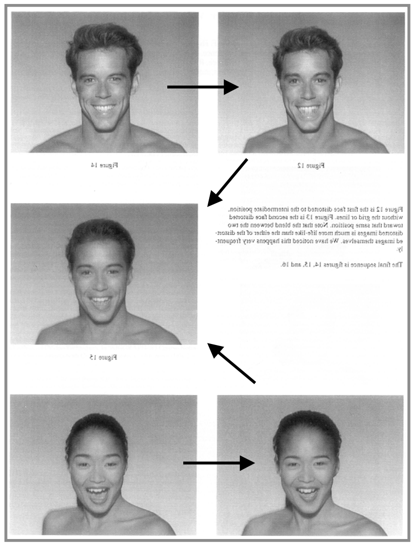
Image$_1$

Warp$_1$



Beier & Neeley Example

Image$_0$ Warp$_0$



Result

Image$_1$ Warp$_1$

CS426 Examples

CS426 Class, Fall98

Robert Osada, Fall00

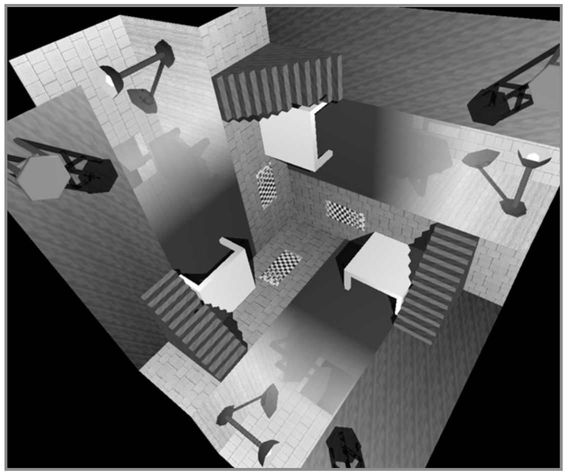
Summary

- Image compositing
  - Blue-screen mattes
  - Alpha channel
  - Porter-Duff compositing algebra
- Image morphing
  - Specifying correspondences
  - Warping
  - Blending

Image Processing

- Quantization
  - Uniform Quantization
  - Random dither
  - Ordered dither
  - Floyd-Steinberg dither
- Pixel operations
  - Add random noise
  - Add luminance
  - Add contrast
  - Add saturation
- Filtering
  - Blur
  - Detect edges
- Warping
  - Scale
  - Rotate
  - Warp
- Combining
  - Composite
  - Morph

Next Time: 3D Rendering



Misha Kazhdan,
CS426, Fall99