Image Compositing & Morphing

Adam Finkelstein & Tim Weyrich
Princeton University
COS 426, Spring 2008

Image Processing

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

Image Processing

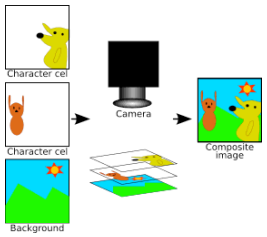
- Pixel operations
  - Add random noise
  - Add luminance
  - Add contrast
  - Add saturation
- Filtering
  - Blur
  - Detect edges
  - Sharpen
  - Emboss
  - Median
- Quantization
  - Uniform Quantization
  - Floyd-Steinberg dither
- Warping
  - Scale
  - Rotate
  - Warp
- Combining
  - Composite
 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

Image Compositing

- Combine images
  - Separate image into “elements”
  - Generate independently
  - Composite together
- Applications
  - Cel animation
  - Chroma-keying
  - Blue-screen matting

Cel Animation

- Classical animation technique (Disney)
- Superimposition of different layers on translucent films



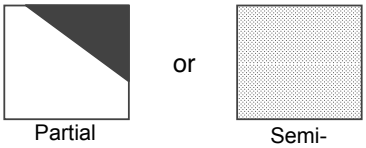
Blue-Screen Matting

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

Problem: no partial coverage!

Alpha Channel

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



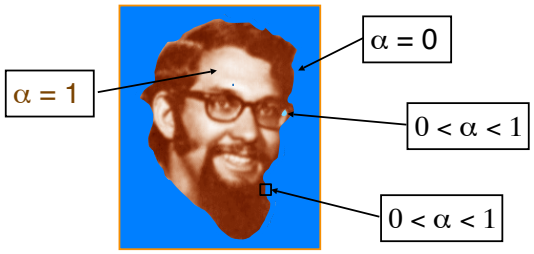
Partial Coverage

or

Semi-Transparent

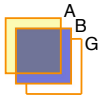
Compositing with Alpha

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



  - o How much of B is blocked by A?

$$\alpha_A$$

  - o How much of B shows through A

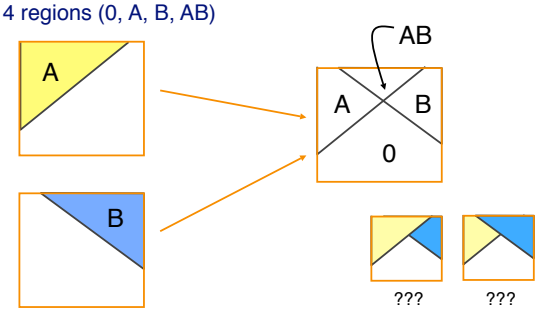
$$(1-\alpha_A)$$

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

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

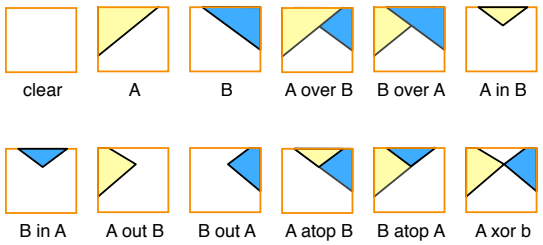
Opaque Objects

- How do we combine 2 partially covered pixels?
  - o 3 possible colors (0, A, B)
  - o 4 regions (0, A, B, AB)



Composition Algebra

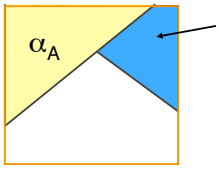
- 12 reasonable combinations



Porter & Duff `84

Example: C = A Over B

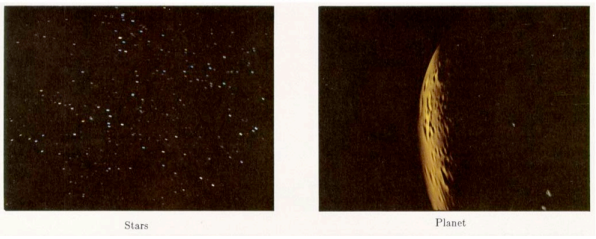
- Consider the areas covered:
  - o $C = \alpha_A A + (1-\alpha_A)\,\alpha_B B$
  - o $\alpha = \alpha_A + (1-\alpha_A)\,\alpha_B$



A over B

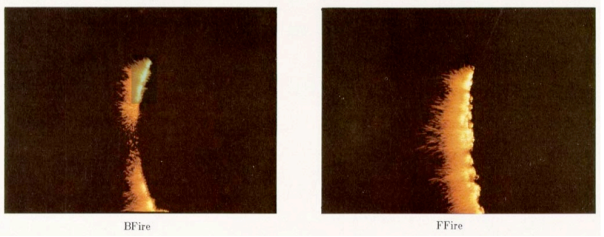
Assumption: coverages of A and B are uncorrelated for each pixel

Image Composition Example



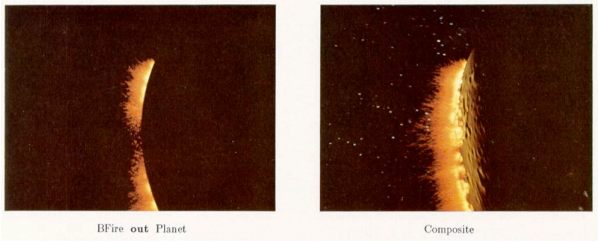
[Porter&Duff *Computer Graphics* 18:3 1984]

Image Composition Example



[Porter&Duff *Computer Graphics* 18:3 1984]

Image Composition Example

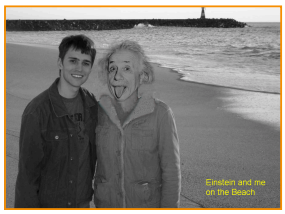


[Porter&Duff *Computer Graphics* 18:3 1984]

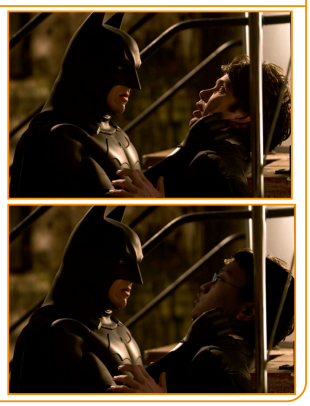
Image Composition Example

Jurassic Park

COS426 Examples



Darin Sleiter



Kenrick Kin

Even CG folks Can Win an Oscar

Smith Duff Catmull Porter

Image Processing

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

Image Morphing

- Animate transition between two images

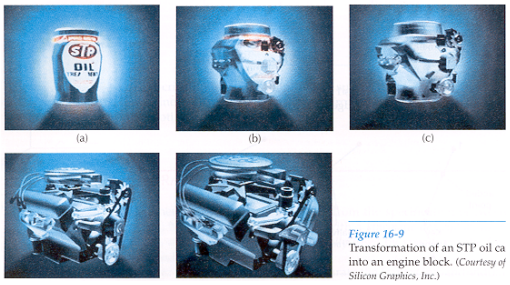


Figure 16-9 Transformation of an STP oil ca into an engine block. (*Courtesy of Silicon Graphics, Inc.*)

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

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

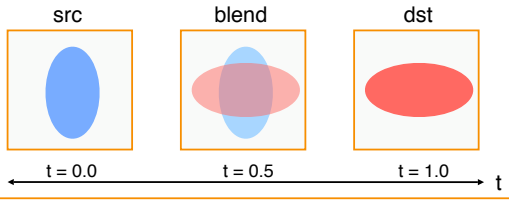


Image Morphing

- Combines warping and cross-dissolving

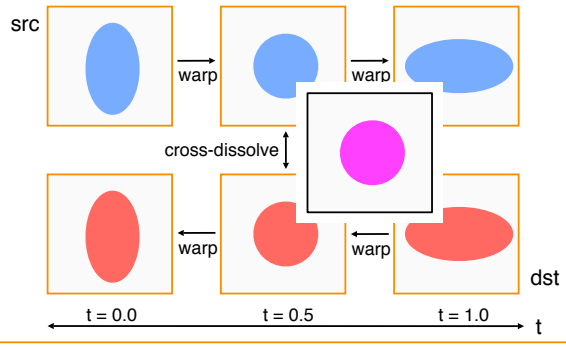
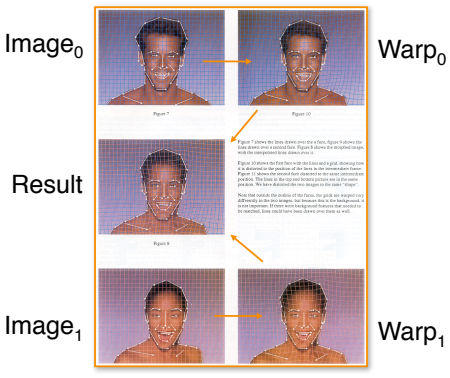


Image Morphing

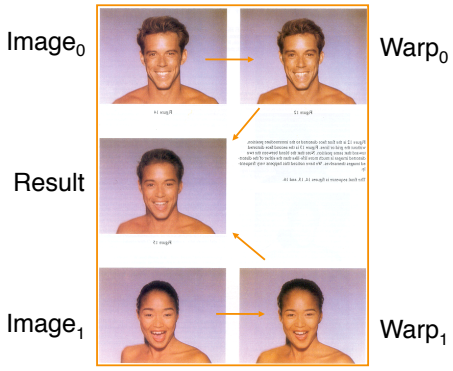
- Hard part is defining warp to intermediate images
  - Aim to align features
  - Aim to preserve shapes

Alexa

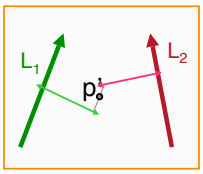
Beier & Neeley Example



Beier & Neeley Example



Warping Pseudocode



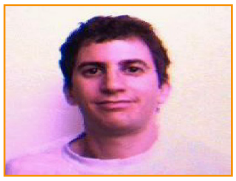

```
WarpImage(Image, L'[…], L[…])
begin
    foreach destination pixel p do
        psum = (0,0)
        wsum = 0
        foreach line L[i] in destination do
            p'[i] = p transformed by (L[i],L'[i])
            psum = psum + p'[i] * weight[i]
            wsum += weight[i]
        end
       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


Morphing Pseudocode

```
GenerateAnimation(Image0, L0[…], Image1, L1[…])
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 L0 [i] to L1 [i]
        end
        Warp0 = WarpImage(Image0, L0, L)
        Warp1 = WarpImage(Image1, L1, L)
        foreach pixel p in FinalImage do
            Result(p) = (1-t) Warp0 + t Warp1

    end
end
```


COS426 Examples

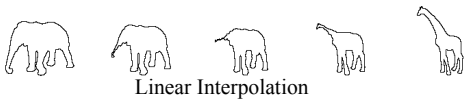


CS426 Class, Fall98

Jon Beyer

More Advanced Image Morphing

- Automatic methods to avoid shape deformations during interpolating warp



Linear Interpolation

As Rigid As Possible Shape Interpolation

Alexa

More Advanced Image Morphing

- "As rigid as possible" shape interpolation

Alexa

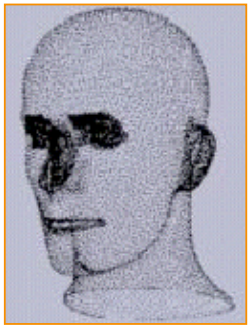
Summary

- Image compositing
  - Alpha channel
  - Porter-Duff compositing algebra
- Image morphing
  - Warping
  - Compositing

Summary

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

Next Time: 3D Modeling



Hoppe