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](image1.png) or ![Semi-Transparent](image2.png)

**Pixels with Alpha**

- Alpha channel convention:
  - $(r, g, b, \alpha)$ represents a pixel that is
    $\alpha$ covered by the color $C = (r/\alpha, g/\alpha, b/\alpha)$
    - Color components are premultiplied by $\alpha$
    - 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

![Image showing different values of \( \alpha \)]

\( \alpha = 1 \)
\( 0 < \alpha < 1 \)
\( \alpha = 0 \)

Semi-Transparent Objects

- Suppose we put A over B over background G
  - How much of B is blocked by A?
    \( \alpha_A \)
  - 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
  - clear
  - A over B
  - B over A
  - A in B
  - B in A
  - A out B
  - B out A
  - A atop B
  - B atop A
  - A xor b

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

Jurassic 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 can into an engine block (Courtesy of Silicon Graphics, Inc.)
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

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

Image Morphing

- Combines warping and cross-dissolving
Image Morphing

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

![Image Morphing](image1.png)

Feature-Based Warping

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

![Feature-Based Warping](image2.png)
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”?

[Diagram showing the transformation of the letter F]

Rotation!

Warping with One Line Pair

- What happens to the “F”?

[Diagram showing the transformation of the letter 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
Weighting Effect of Each Line Pair

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

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

Where:
- \( \text{length}[i] \) is the length of \( L[i] \)
- \( \text{dist}[i] \) is the distance from \( X \) to \( L[i] \)
- \( a, b, p \) are constants that control the warp

Warping Pseudocode

```
Warplmage(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(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 = WarplImage(Image₀, L₀, L)
    Warp₁ = WarplImage(Image₁, L₁, L)
    foreach pixel p in FinalImage do
      Result(p) = (1-t) Warp₀ + t Warp₁
    end
  end
end

Beier & Neeley Example

Image₀ → Warp₀

Result

Image₁ → Warp₁
Beier & Neeley Example

Image₀ → Warp₀

Result

Image₁ → Warp₁

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