Image Compositing
& Morphing

COS 426
Digital Image Processing

- Changing intensity/color
  - Linear: scale, offset, etc.
  - Nonlinear: gamma, saturation, etc.
  - Add random noise

- Filtering over neighborhoods
  - Blur
  - Detect edges
  - Sharpen
  - Emboss
  - Median

- Moving image locations
  - Scale
  - Rotate
  - Warp

- Combining images
  - Composite
  - Morph

- Quantization

- Spatial / intensity tradeoff
  - Dithering
Types of Transparency

- Refraction
  - Light is bent as it goes through an object
  - Can focus light: caustics
  - Can be color-dependent: dispersion
Types of Transparency

• Refraction

• Subsurface scattering
  ○ Translucent materials
  ○ Light leaves at different position than it entered
Types of Transparency

• Refraction

• Subsurface scattering
  ○ Translucent materials
  ○ Light leaves at different position than it entered

• Today: nonrefractive transparency
  ○ Pixelwise composition
  ○ Separate image into “elements” or “layers”
  ○ Can generate independently
  ○ Composite together

Smith & Blinn `84
Example

Jurassic Park
Image Composition

• Issues:
  ◦ Segmentation of image into regions
  ◦ Blend into single image seamlessly
Image Composition

• Issues:
  ➢ Segmentation of image into regions
    ◦ Blend into single image seamlessly
Image Segmentation

- Chroma keying (blue- or green-screen)
  - Photograph object in front of screen with known color
Image Segmentation

• Specify segmentation by hand
  ◦ Purely manual: rotoscoping
  ◦ Semi-automatic: graph min-cut
    Separate image regions along minimal cuts (where edges measure differences between adjacent pixels)
Image Segmentation

- Novel methods, e.g. flash matting

Sun et al., 2006
Image Composition

• Issues:
  ◦ Segmentation of image into regions
    ➢ Blend into single image seamlessly
Image Blending

• Ingredients
  ○ Background image
  ○ Foreground image with blue background

• Method
  ○ Non-blue foreground pixels overwrite background
Blending with Alpha

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

$\alpha = 1$

$\alpha = 0$

$0 < \alpha < 1$

$0 < \alpha < 1$
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
```
Alpha Blending: “Over” Operator

\[ C = A \text{ over } B \]
\[ C = \alpha_A A + (1-\alpha_A) B \]

This assumes an image with “non-pre-multiplied” alpha.

Will (rarely) encounter images with “pre-multiplied” alpha: store \((\alpha R, \alpha G, \alpha B, \alpha)\) instead of \((R, G, B, \alpha)\)

\[ 0 < \alpha < 1 \]
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 \( B \)?
  \[ (1 - \alpha_A)(1 - \alpha_B) \]
Alpha Blending: “Over” Operator

• Suppose we put A over B over background G

  \[
  \alpha_A A + (1-\alpha_A)\alpha_B B + (1-\alpha_A)(1-\alpha_B)G
  \]

  \[
  = \alpha_A A + (1-\alpha_A)\left[\alpha_B B + (1-\alpha_B)G\right]
  \]

  \[
  = A \text{ over } [B \text{ over } G]
  \]

  Must perform “over” back to front!
Other Compositing Operations

• How can we combine 2 partially covered pixels?
  ◦ 3 possible colors (0, A, B)
  ◦ 4 regions (0, A, B, AB)
Blending with Alpha

Composition algebra – 12 combinations

\[ C' = F_A \alpha_A A + F_B \alpha_B B \]

<table>
<thead>
<tr>
<th>Operation</th>
<th>( F_A )</th>
<th>( F_B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>A over B</td>
<td>1</td>
<td>1 - ( \alpha_A )</td>
</tr>
<tr>
<td>B over A</td>
<td>1 - ( \alpha_B )</td>
<td>1</td>
</tr>
<tr>
<td>A in B</td>
<td>( \alpha_B )</td>
<td>0</td>
</tr>
<tr>
<td>B in A</td>
<td>0</td>
<td>( \alpha_A )</td>
</tr>
<tr>
<td>A out B</td>
<td>1 - ( \alpha_B )</td>
<td>0</td>
</tr>
<tr>
<td>B out A</td>
<td>0</td>
<td>1 - ( \alpha_A )</td>
</tr>
<tr>
<td>A atop B</td>
<td>( \alpha_B )</td>
<td>1 - ( \alpha_A )</td>
</tr>
<tr>
<td>B atop A</td>
<td>1 - ( \alpha_B )</td>
<td>( \alpha_A )</td>
</tr>
<tr>
<td>A xor B</td>
<td>1 - ( \alpha_B )</td>
<td>1 - ( \alpha_A )</td>
</tr>
</tbody>
</table>

Porter & Duff `84
Blending with Alpha

- Example: $C = A \text{ Over } B$
  
  - $C' = \alpha_A A + (1-\alpha_A) \alpha_B B$
  
  - $\alpha = \alpha_A + (1-\alpha_A) \alpha_B$

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

Image Composition Example

Image Composition Example

BFire out Planet

Composite

COS426 Examples

Darin Sleiter

Einstein and me on the Beach

Kenrick Kin
Poisson Image Blending

- General idea
  - Solve for image samples that follow gradients of source subject to boundary conditions imposed by dest

\[
\begin{align*}
\nabla^2 f &= \nabla \cdot \mathbf{v} \\
\left. f \right|_{\partial\Omega} &= \left. f^* \right|_{\partial\Omega}
\end{align*}
\]
Poisson Image Blending

sources  destinations  cloning  seamless cloning
Poisson Image Blending
Digital Image Processing

- Changing intensity/color
  - Linear: scale, offset, etc.
  - Nonlinear: gamma, saturation, etc.
  - Add random noise
- Filtering over neighborhoods
  - Blur
  - Detect edges
  - Sharpen
  - Emboss
  - Median
- Moving image locations
  - Scale
  - Rotate
  - Warp
- Combining images
  - Composite
  - Morph
- Quantization
- Spatial / intensity tradeoff
  - Dithering
Image Morphing

- Animate transition between two images
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
Beier & Neeley Example

Image_0

Warp_0

Result

Image_1

Warp_1
Beier & Neeley Example

Image$_0$

Warp$_0$

Result

Image$_1$

Warp$_1$
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(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)
  end
  foreach pixel p in FinalImage do
    Result(p) = (1-t) Warp_0 + t Warp_1
  end
end
COS426 Examples

CS426 Class, Fall98

Jon Beyer
Image Composition Applications

• Stitching images into a panorama
Image Composition Applications

- Photo montage
Image Composition Applications

- Photo montage
Image Composition Applications

- Removing people
Image Composition Applications

- Stroboscopic images
Image Composition Applications

- Extended depth-of-field
Image Composition Applications

- Flash / No flash
Image Composition Applications

- High dynamic range images
Image Composition Applications

• High dynamic range images
Image Composition Applications

- Multi-camera array
Summary

- Image compositing
  - Alpha channel
  - Porter-Duff compositing algebra

- Image morphing
  - Warping
  - Compositing

- Computational photography
Next Time: 3D Modeling

Hoppe