Image Compositing & Morphing

Felix Heide
COS 426, Spring 2020
Princeton University
### Digital Image Processing

- **Changing pixel values**
  - Linear: scale, offset, etc.
  - Nonlinear: gamma, saturation, etc.
  - Histogram equalization
- **Filtering over neighborhoods**
  - Blur & sharpen
  - Detect edges
  - Median
  - Bilateral filter
- **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

• Today: **compositing**
  - Separate image into layers with known order
  - Can generate layers independently
  - *Pixelwise* combination: each pixel in each layer can be transparent, opaque, or somewhere in between

Smith & Blinn`84
Image Composition

• Issues:
  ○ Segmenting image into regions
  ○ Blending into single image seamlessly
Image Composition

• Issues:
  • Segmenting image into regions
    ○ Blending 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 (draw matte, every frame)
  ◦ Semi-automatic: graph-cut (draw a few strokes)

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 Segmentation

- Portrait mode in Google Pixel Phone

© Markus Kohlpaintner

Wadhwa et al., 2018
Image Segmentation

• Portrait mode in Google Pixel Phone

Wadhwa et al., 2018
Image Composition

• Issues:
  ◦ Segmenting image into regions
  ➢ Blending into single image seamlessly
Image Blending

• Ingredients
  ◦ Background image
  ◦ Foreground image with blue background

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

Per-pixel “alpha” channel: controls the linear interpolation between foreground and background pixels when elements are composited.

\[ \alpha = 1 \]

\[ 0 < \alpha < 1 \]
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: $\alpha = 0.3$

[Diagram of partial coverage and semi-transparent]
Alpha Blending: “Over” Operator

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

\[ C = \alpha_A A + (1-\alpha_A) B \]

\[ 0 < \alpha < 1 \]
• 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 $B$?
    $$(1 - \alpha_A)(1 - \alpha_B)$$
Compositing Algebra

- Suppose we put A over B over background G

  - Final result?

\[
\alpha_A A + (1 - \alpha_A) \alpha_B B + (1 - \alpha_A)(1 - \alpha_B) G \\
= \alpha_A A + (1 - \alpha_A) [\alpha_B B + (1 - \alpha_B) G] \\
= A \overline{[B \overline{G}]} \\
\]

Must perform “over” back-to-front: right associative!
Other Compositing Operations

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

How many combinations?!!
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
Image Composition Example

Stars

Planet

Image Composition Example

Image Composition Example

Image Composition Example

“Genesis” sequence from Star Trek II: The Wrath of Khan (1982)
COS426 Examples

Einstein and me on the Beach

Darin Sleiter

Kenrick Kin
Poisson Image Blending

Beyond simple compositing

- 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
Poisson Image Blending
Digital Image Processing

- Changing pixel values
  - Linear: scale, offset, etc.
  - Nonlinear: gamma, saturation, etc.
  - Histogram equalization

- Filtering over neighborhoods
  - Blur & sharpen
  - Detect edges
  - Median
  - Bilateral filter

- Moving image locations
  - Scale
  - Rotate
  - Warp

- Combining images
  - Composite
  - Morph

- Quantization

- Spatial / intensity tradeoff
  - Dithering
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
Beier & Neeley Example

Image₀ → Warp₀ → Result → Image₁ → Warp₁
Beier & Neeley Example

Image$_0$

Warp$_0$

Result

Image$_1$

Warp$_1$
Beier & Neeley Example

Black or White, Michael Jackson (1991)
WarpImage(Image, L_{src}[,\ldots], L_{dst}[,\ldots])
begin
  foreach destination pixel p_{dst} do
    psum = (0,0)
    wsum = 0
    foreach line L_{dst}[i] do
      p_{src}[i] = p_{dst} transformed by (L_{dst}[i], L_{src}[i])
      psum = psum + p_{src}[i] * weight[i]
      wsum += weight[i]
    end
    p_{src} = psum / wsum
  Result(p_{dst}) = Resample(p_{src})
end
end
Warping Pixel Locations

This generates one warp per line, each of which is a simple rotation and non-uniform scale (scaling is only done along the axis of the line). These warps must then be averaged to get the final warp. In the original paper, the weights for the average are tuned with the formula below. The \textit{dist} variable is the distance of the point from the line segment, and the \textit{length} variable is the length of the line segment.

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

The equations give several parameters to tune, and I got the best results when \( a = 0.001 \), \( b = 2 \), and \( p = 0 \). Ignoring the length of the line segments (by setting \( p \) to zero) gave better results than when the length was taken into account. I used seven contours with 28 line segments to represent the features of each face.

Nice implementation notes from Evan Wallace, Brown University
http://cs.brown.edu/courses/csci1950-g/results/proj5/edwallac/
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 tᵗʰ of the way from L₀[i] to L₁[i]
    end
    Warp₀ = WarpImage(Image₀, L₀, L)
    Warp₁ = WarpImage(Image₁, L₁, L)
  end
  foreach pixel p in FinalImage do
    Result(p) = (1- t) Warp₀ + t Warp₁
  end
end
COS426 Example

Amy Ousterhout
COS426 Examples

ckctwo

Jon Beyer
COS426 Examples

Sam Payne
Matt Matl
Image Composition Applications

• “Computational photography”: new photographic effects that inherently use multiple images + computation

• Example: stitching images into a panorama
Image Composition Applications

• Flash / No flash

[Michael Cohen]
Image Composition Applications

- Photo montage
Image Composition Applications

- Photo montage
Stoboscopic images
Image Composition Applications

- Extended depth-of-field
Image Composition Applications

• Removing people
Scene Completion Using Millions of Photographs

James Hays and Alexei A. Efros
SIGGRAPH 2007

Slides by J. Hays and A. Efros
Image Completion
Image Completion

2.3 Million unique images from Flickr

Hays et al. SIGGRAPH 07
Scene Completion Result
Image Completion Algorithm

Input image

Scene Descriptor

Image Collection

Mosaicing

20 completions

200 matches
Summary

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

• Image morphing
  ◦ Warping
  ◦ Compositing

• Computational photography