Image Compositing
& Morphing

COS 426, Spring 2014
Princeton University
Image Processing Operations I

- Luminance
  - Brightness
  - Contrast
  - Gamma
  - Histogram equalization

- Color
  - Black & white
  - Saturation
  - White balance

- Linear filtering
  - Blur & sharpen
  - Edge detect
  - Convolution

- Non-linear filtering
  - Median
  - Bilateral filter

- Dithering
  - Quantization
  - Ordered dither
  - Floyd-Steinberg
Image Processing Operations II

• Transformation
  - Scale
  - Rotate
  - Warp

• Combining images
  - Composite
  - Morph
  - Computational photography

Last time

Today
Image Processing Operations II

• Transformation
  ▪ Scale
  ▪ Rotate
  ▪ Warp

}{ Last time

Combining images
  ▪ Composite
  ▪ Morph
  ▪ Computational photography

}{ Today
Image Composition

• Issues:
  ◦ Segmentation of image into layers/regions
  ◦ Blend into single image seamlessly

Smith & Blinn`84
Image Composition

• Issues:
  - Segmentation of image into layers/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 (draw matte, every frame)
  - Semi-automatic: graph min-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 Composition

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

• Ingredients
  - Background image
  - Foreground image

• Method
  - Foreground pixels overwrite background
Blending with Alpha

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

$\alpha = 1$

$0 < \alpha < 1$

$\alpha = 0$
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

or

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

- Suppose we put A over B over background G

- Final result?

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

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

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

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

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

### Operation Table

<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: $C = A$ 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

Stars

Planet

Image Composition Example

BFire

FFire

Image Composition Example

BFire out Planet

Composite

COS426 Examples

Einstein and me on the Beach

Darin Sleiter

Kenrick Kin
Beyond simple compositing

- Solve for image samples that follow gradients of source subject to boundary conditions imposed by dest
Poisson Image Blending

sources | destinations | cloning | seamless cloning
Poison Image Blending
Poisson Image Blending

http://www.csie.ntu.edu.tw/~r00944002/CPHW2/result.htm
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

\[ \begin{array}{ccc}
\text{src} & \text{warp} & \text{warp} \\
\text{cross-dissolve} & \text{warp} & \text{warp} \\
\text{dst} & \text{t} = 0.0 & \text{t} = 0.5 & \text{t} = 1.0
\end{array} \]
Beier & Neeley Example

Image\textsubscript{0} \rightarrow \text{Warp}\textsubscript{0} \rightarrow \text{Result} \rightarrow \text{Image}\textsubscript{1} \rightarrow \text{Warp}\textsubscript{1}
Beier & Neeley Example

Image₀ → Warp₀ → Result → Image₁ → Warp₁
Line Correspondence Mappings

- Beier & Neeley use pairs of lines to specify 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) = 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 = 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
COS426 Examples

ckctwo

Jon Beyer
Image Composition Applications

- *Computational photography*: enable new photographic effects that inherently use multiple images + computation + composition.
Image Composition Applications

• Extended depth-of-field
Image Composition Applications

- High dynamic range images
Image Composition Applications

• High dynamic range images
Image Composition Applications

- Flash / No flash

non-flash + flash = merged

[Michael Cohen]
Image Composition Applications

- Stoboscopic images
Image Composition Applications

- Photo montage

[Michael Cohen]
Image Composition Applications

- Photo montage
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
Texture synthesis result
Image Completion
Image Completion

2.3 Million unique images from Flickr
Scene Completion Result
Image Completion Algorithm

Input image → Scene Descriptor → Image Collection

20 completions ← Mosaicing ← 200 matches
Image Completion
Image Completion
Image Completion Result
Image Completion Results
Summary

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

• Image morphing
  ◦ Warping
  ◦ Compositing

• Computational photography
Next Time: 3D Modeling

Hoppe