# 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
  - Light leaves at different position than it entered
  - Translucent materials
Types of Transparency

- Refraction
- Subsurface scattering
- Today: compositing
  - Nonrefractive (partial) transparency
  - Separate image into layers with known order
  - \textit{Pixelwise} combination: each pixel in each layer can be transparent, opaque, or somewhere in between

Smith & Blinn`84
Example

Jurassic Park (1993)
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 Matting

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

• Specify segmentation by hand
  ◦ Purely manual: draw matte every frame
  ◦ Semi-automatic: graph-cut (draw a few strokes)
    Implemented using min-cut algorithm: separate regions along minimal cuts (where edges measure differences between adjacent pixels)
Image Matting

• Portrait mode in Google Pixel Phone

Wadhwa et al., 2018
Image Matting

- Portrait mode blur in Google Pixel Phones

Wadhwa et al., 2018
Image Matting

- Portrait mode blur in Google Pixel Phones

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

\[\alpha = 1\]

\[\alpha = 0\]

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: $\alpha = 0.3$

![Partial Coverage](image1)

![Semi-Transparent](image2)

or
Alpha Blending: “Over” Operator

• If background B is opaque:
  ○ $C = A \text{ over } B$
  ○ $C = \alpha_A A + (1-\alpha_A)B$

• If background B has its own $\alpha$:
  ○ $C = A \text{ over } B$
  ○ $C = \alpha_A A + (1-\alpha_A)\alpha_B B$
  ○ $\alpha_C = \alpha_A + (1-\alpha_A)\alpha_B$

$0 < \alpha < 1$
Suppose we put $A$ over $B$ over background $G$.

- How much of $B$ is blocked by $A$?

$\alpha_A$
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?
  \[ (1 - \alpha_A) \]
Compositing Algebra

- 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?
    \[ (1 - \alpha_A)(1 - \alpha_B) \]
Suppose we put \( A \) over \( B \) over background \( G \)

Final result?

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

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

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>
</tbody>
</table>
Other Compositing Operations

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>
</tbody>
</table>

Porter & Duff `84
Other Compositing Operations

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>
</tbody>
</table>

Porter & Duff '84
Other Compositing Operations

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>
</tbody>
</table>

Source

Over

In

Out

Dest

Dest Over

Dest In

Dest Out

Porter & Duff `84
Other Compositing Operations

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_A )</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

BFire

FFire

Porter & Duff `84
Image Composition Example

Porter & Duff `84
“Genesis” sequence from Star Trek II: The Wrath of Khan
COS426 Examples

Darin Sleiter

Einstein and me on the Beach

Kenrick Kin
Beyond simple compositing

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

sources  destinations  cloning  seamless cloning
Beyond simple compositing

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

\[ \min_f \iint_{\Omega} |\nabla f - v|^2 \quad \text{with} \quad f|_{\partial \Omega} = f^*|_{\partial \Omega} \]
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 1.0 to 0.0

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

• Combines warping and cross-dissolving
Beier & Neeley Example

Image_0

Warp_0

Result

Image_1

Warp_1
Black or White, Michael Jackson (1991)
Warping Pixel Locations

The original basis

The warped basis

Pixel Position

Pixel Position
Warping Pixel Locations

$$u = \frac{(X - P) \cdot (Q - P)}{||Q - P||^2}$$

$$v = \frac{(X - P) \cdot \text{Perpendicular} (Q - P)}{||Q - P||}$$

$$X' = P' + u \cdot (Q' - P') + \frac{v \cdot \text{Perpendicular} (Q' - P')}{||Q' - P'||}$$
Warping Pixel Locations

\[ u = \frac{(X - P) \cdot (Q - P)}{\|Q - P\|^2} \]

\[ v = \frac{(X - P) \cdot \text{Perpendicular}(Q - P)}{\|Q - P\|} \]

\[ X' = P' + u \cdot (Q' - P') + \frac{v \cdot \text{Perpendicular}(Q' - P')}{\|Q' - P'|} \]
Warping Pixel Locations

The original basis

The warped basis

Pixel Position Pixel Position

\[
\begin{align*}
    u &= \frac{(X - P) \cdot (Q - P)}{\|Q - P\|^2} \\
    v &= \frac{(X - P) \cdot \text{Perpendicular}(Q - P)}{\|Q - P\|} \\
    X' &= P' + u \cdot (Q' - P') + \frac{v \cdot \text{Perpendicular}(Q' - P')}{\|Q' - P'\|}
\end{align*}
\]
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 dist variable is the distance of the point from the line segment, and the 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/
Warping Pseudocode

Warplmage(Image, L_{src}[…], L_{dst}[…])
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
Morphing Pseudocode

\[
\text{GenerateAnimation}(\text{Image}_0, L_0[...], \text{Image}_1, L_1[...])
\begin{align*}
&\text{begin} \\
&\quad \text{foreach intermediate frame time } t \text{ do} \\
&\quad \quad \text{for } i = 1 \text{ to number of line pairs do} \\
&\quad \quad \quad L[i] = \text{line } t^{th} \text{ of the way from } L_0[i] \text{ to } L_1[i] \\
&\quad \quad \text{end} \\
&\quad \text{Warp}_0 = \text{WarpImage}(\text{Image}_0, L_0, L) \\
&\quad \text{Warp}_1 = \text{WarpImage}(\text{Image}_1, L_1, L) \\
&\quad \text{foreach pixel } p \text{ in FinalImage do} \\
&\quad \quad \text{Result}(p) = (1-t) \text{ Warp}_0 + t \text{ Warp}_1 \\
&\quad \text{end} \\
&\text{end} \\
\end{align*}
\]
COS426 Example

Amy Ousterhout
Image Composition Applications

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

• Example: stitching images into a *panorama*
Image Composition Applications

• Photo montage
Stoboscopic images
Image Composition Applications

- Extended depth-of-field
Scene Completion Using Millions of Photographs

James Hays and Alexei A. Efros

SIGGRAPH 2007

Slides by J. Hays and A. Efros
Image Completion

Hays et al. SIGGRAPH 07
Image Completion

2.3 Million unique images from Flickr

Hays et al. SIGGRAPH 07
Image Completion Algorithm

Input image

Scene Descriptor

Image Collection

20 completions

Mosaicing

200 matches
Summary

- Image compositing
  - Alpha channel
  - Porter-Duff compositing algebra
- Image morphing
  - Warping
  - Compositing
- Compositing in Computational Photography
Next Time: 3D Modeling

Hoppe