



Image Compositing & Morphing

Adam Finkelstein
COS 426, Spring 2019
Princeton University

Digital Image Processing

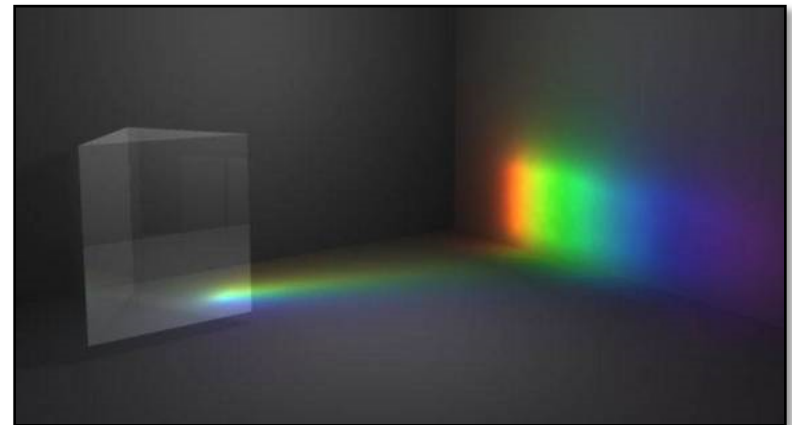
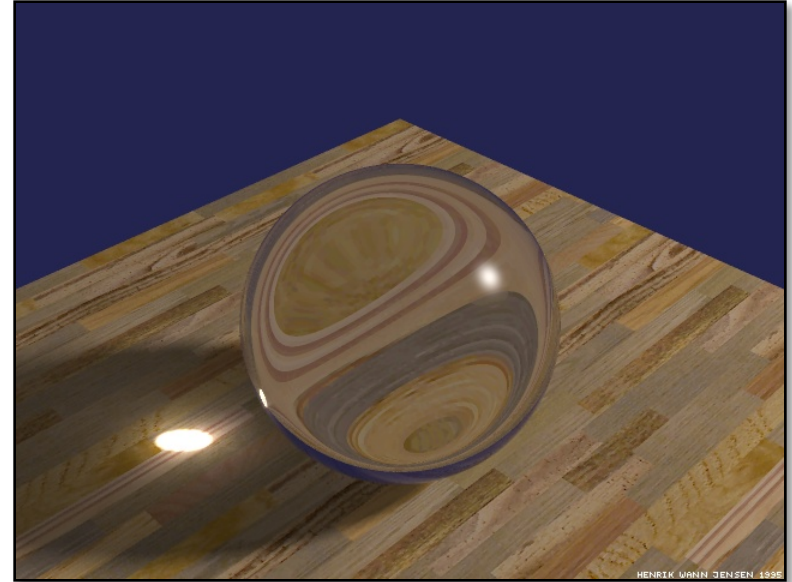


- Changing pixel values
 - Linear: scale, offset, etc.
 - Nonlinear: gamma, saturation, etc.
 - Histogram equalization
- Filtering over neighborhoods
 - Blur & sharpen
 - Detect edges
 - Median
 - Bilateral filter
- Moving image locations
 - Scale
 - Rotate
 - Warp
- Combining images
 - Composite
 - Morph
- Quantization
- Spatial / intensity tradeoff
 - Dithering

Types of Transparency



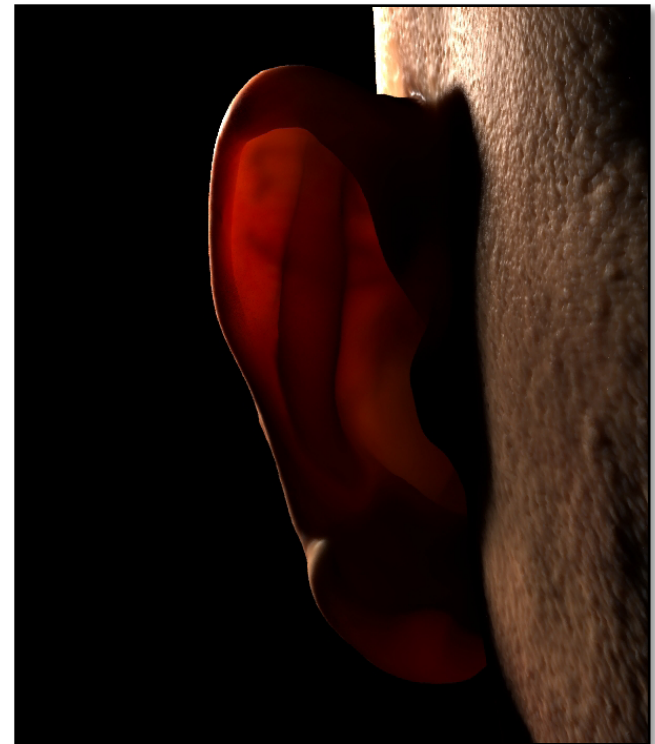
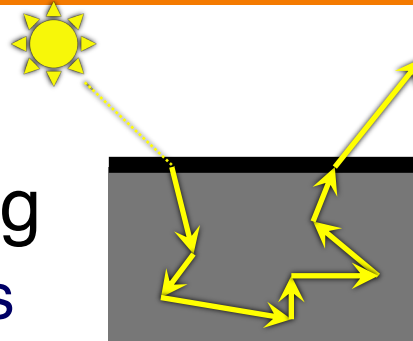
- Refraction
 - Light is bent as it goes through an object
 - Can focus light: caustics
 - Can be color-dependent: dispersion



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

Example



MOVIECLIPS.COM

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 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)

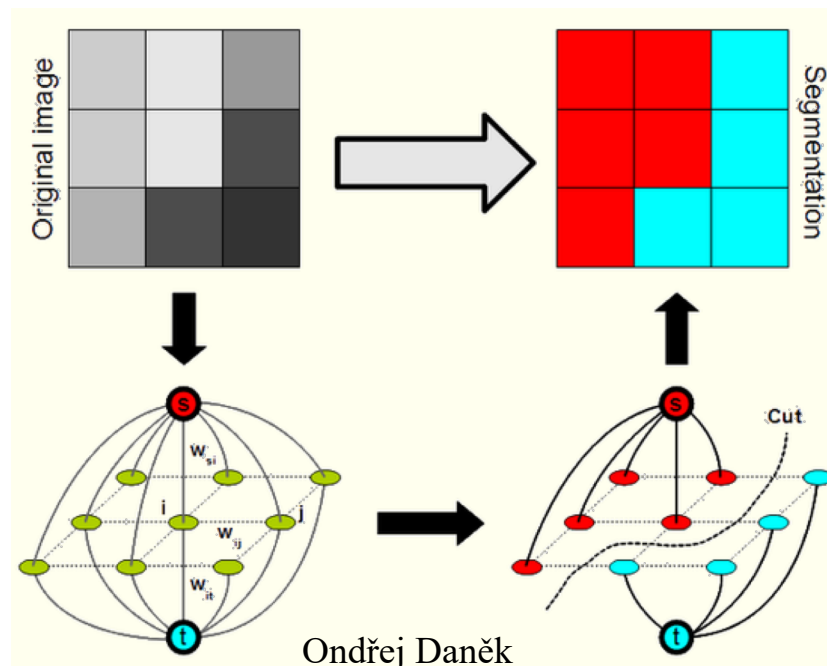


Image Segmentation



- Novel methods, e.g. flash matting



Image Composition



- Issues:
 - Segmenting image into regions
 - Blending into single image seamlessly

Image Blending



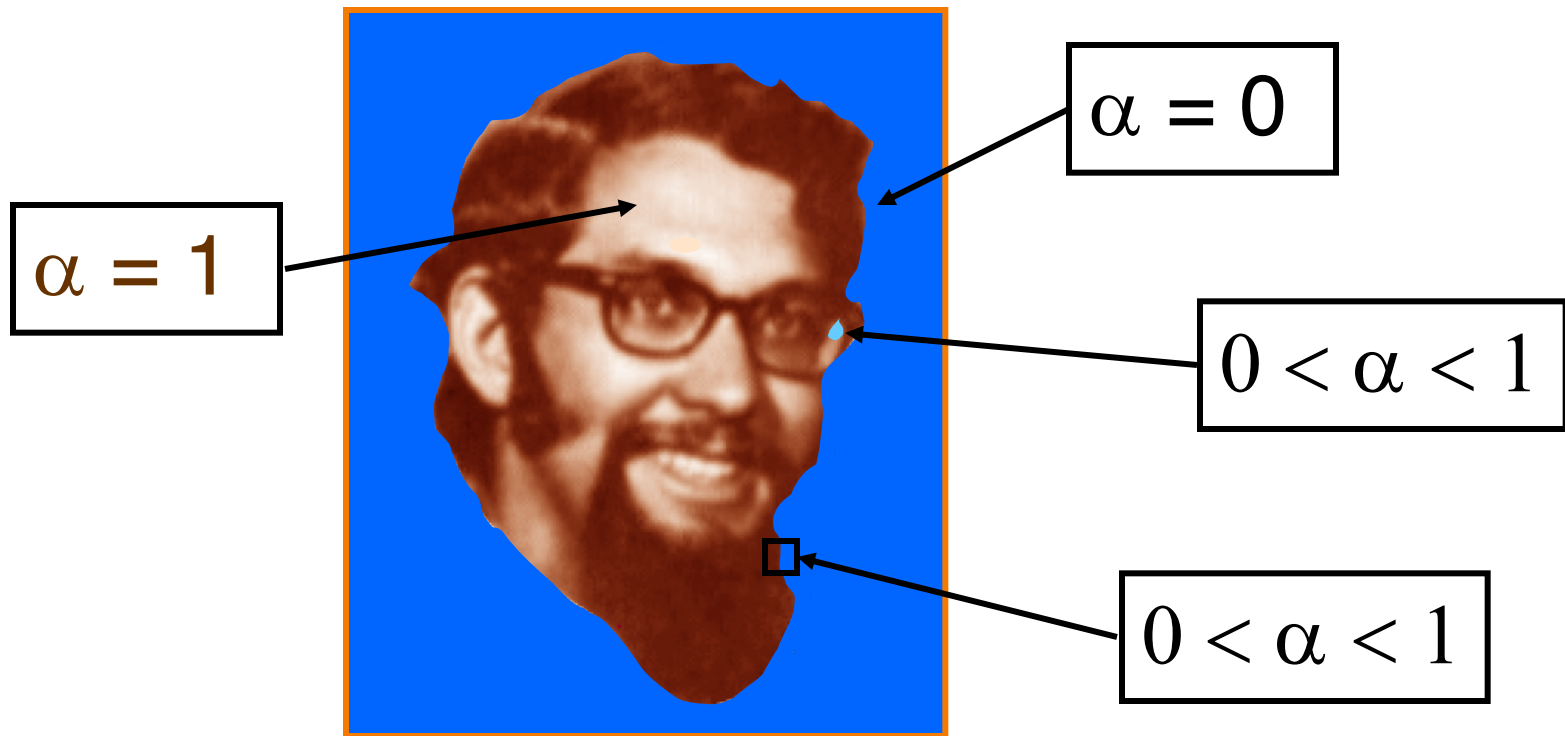
- Ingredients
 - Background image
 - Foreground image with blue background
- Method
 - Non-blue foreground pixels overwrite background





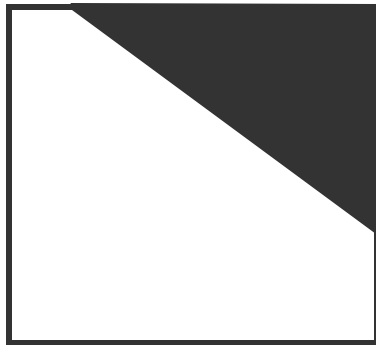
Blending with Alpha Channel

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



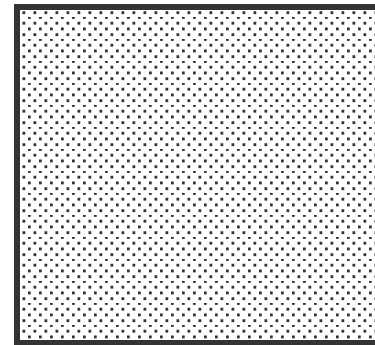
Alpha Channel

- 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



$$C = A \text{ over } B$$

$$C = \alpha_A A + (1 - \alpha_A) B$$

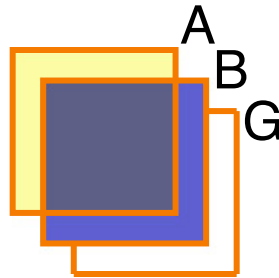


$$0 < \alpha < 1$$

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

$$(1 - \alpha_A)$$

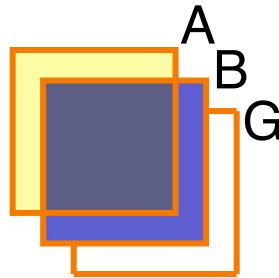
- How much of G shows through both A and B?

$$(1 - \alpha_A)(1 - \alpha_B)$$

Compositing Algebra



- Suppose we put A **over** B **over** background G



- Final result?

$$\alpha_A A + (1 - \alpha_A) \alpha_B B + (1 - \alpha_A) (1 - \alpha_B) G$$

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

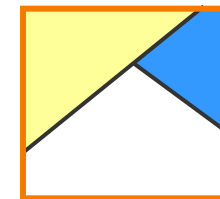
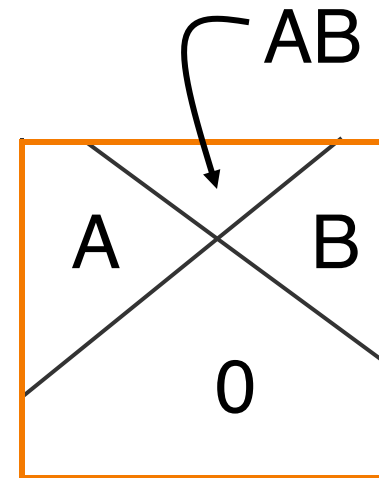
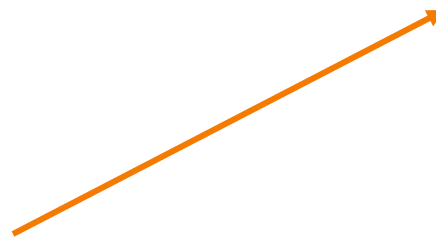
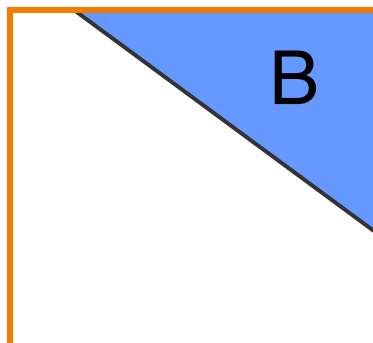
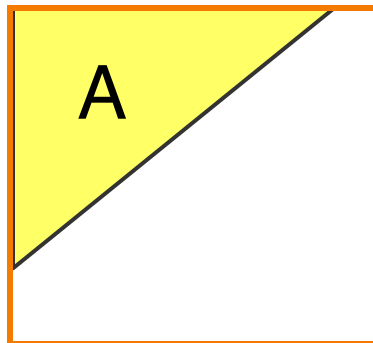
$$= A \text{ over } [B \text{ over } G]$$

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

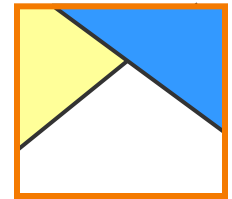
Other Compositing Operations



- 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

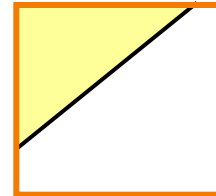
Composition algebra – 12 combinations

$$C' = F_A \alpha_A A + F_B \alpha_B B$$

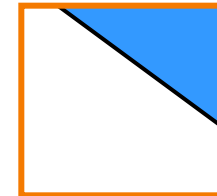
Operation	F_A	F_B
Clear	0	0
A	1	0
B	0	1
A over B	1	$1 - \alpha_A$
B over A	$1 - \alpha_B$	1
A in B	α_B	0
B in A	0	α_A
A out B	$1 - \alpha_B$	0
B out A	0	$1 - \alpha_A$
A atop B	α_B	$1 - \alpha_A$
B atop A	$1 - \alpha_B$	α_A
A xor B	$1 - \alpha_B$	$1 - \alpha_A$



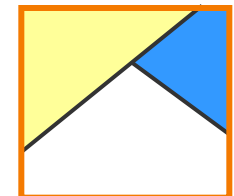
clear



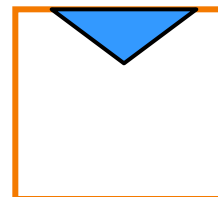
A



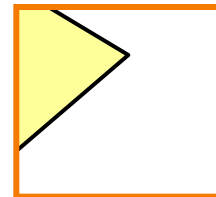
B



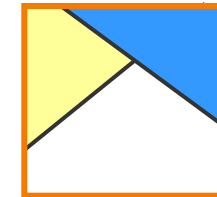
A over B



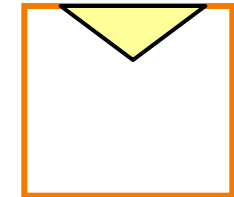
B in A



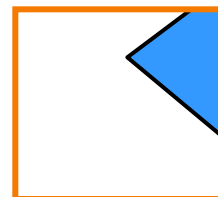
A out B



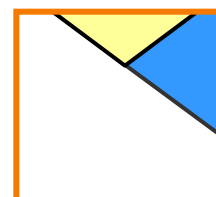
B over A



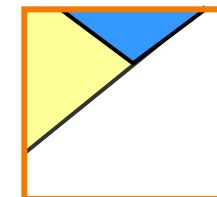
A in B



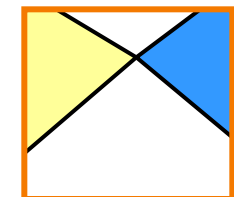
B out A



A atop B



B atop A

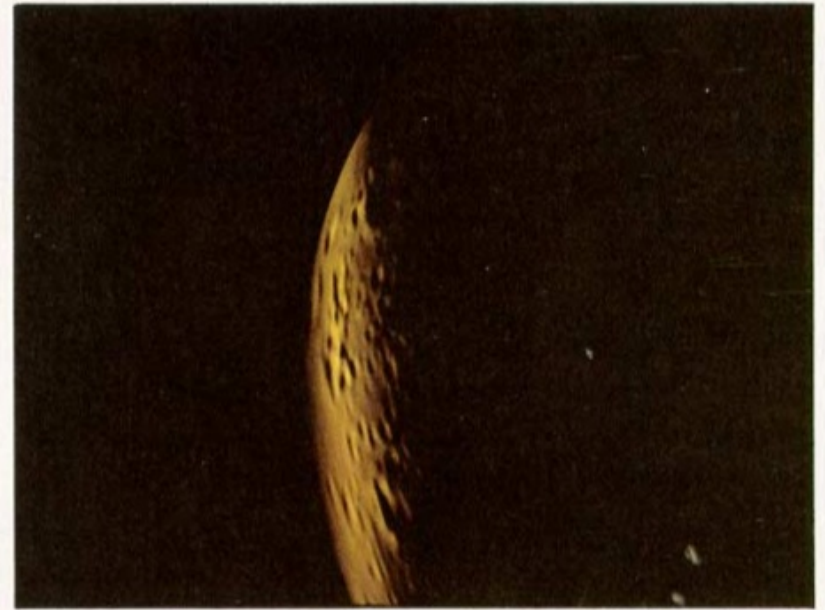


A xor b

Image Composition Example



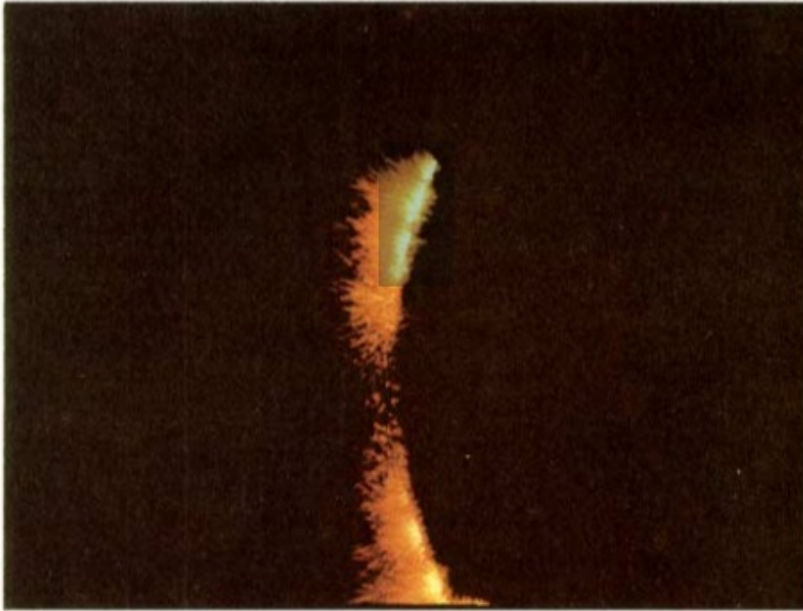
Stars



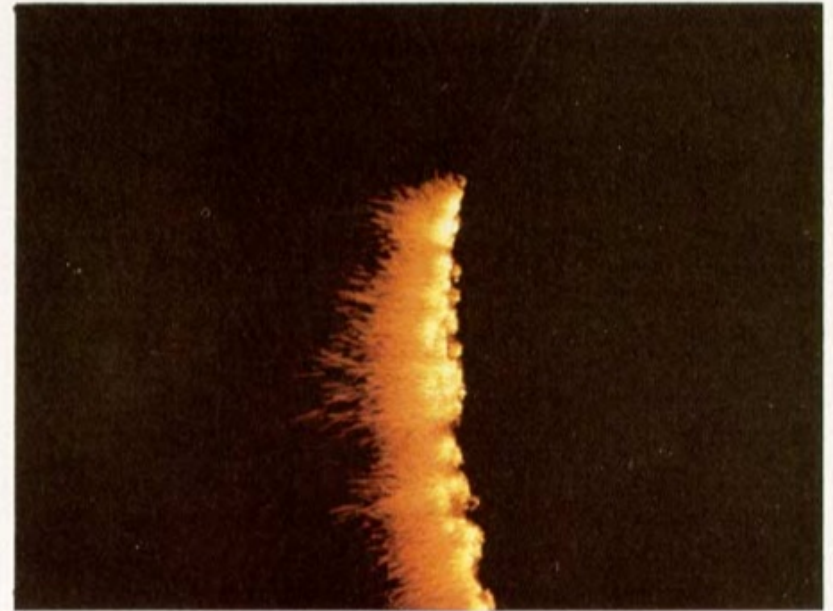
Planet

[Porter&Duff *Computer Graphics* 18:3 1984]

Image Composition Example



BFire



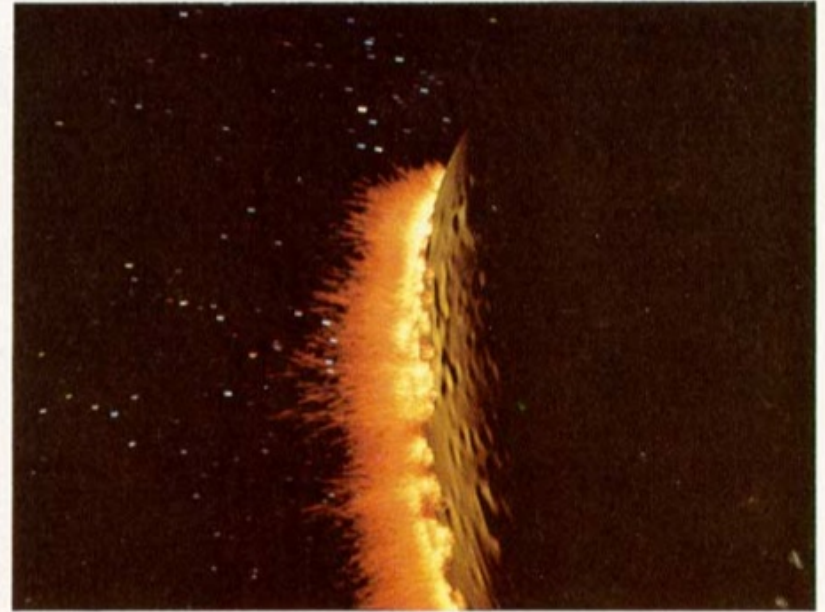
FFire

[Porter&Duff *Computer Graphics* 18:3 1984]

Image Composition Example



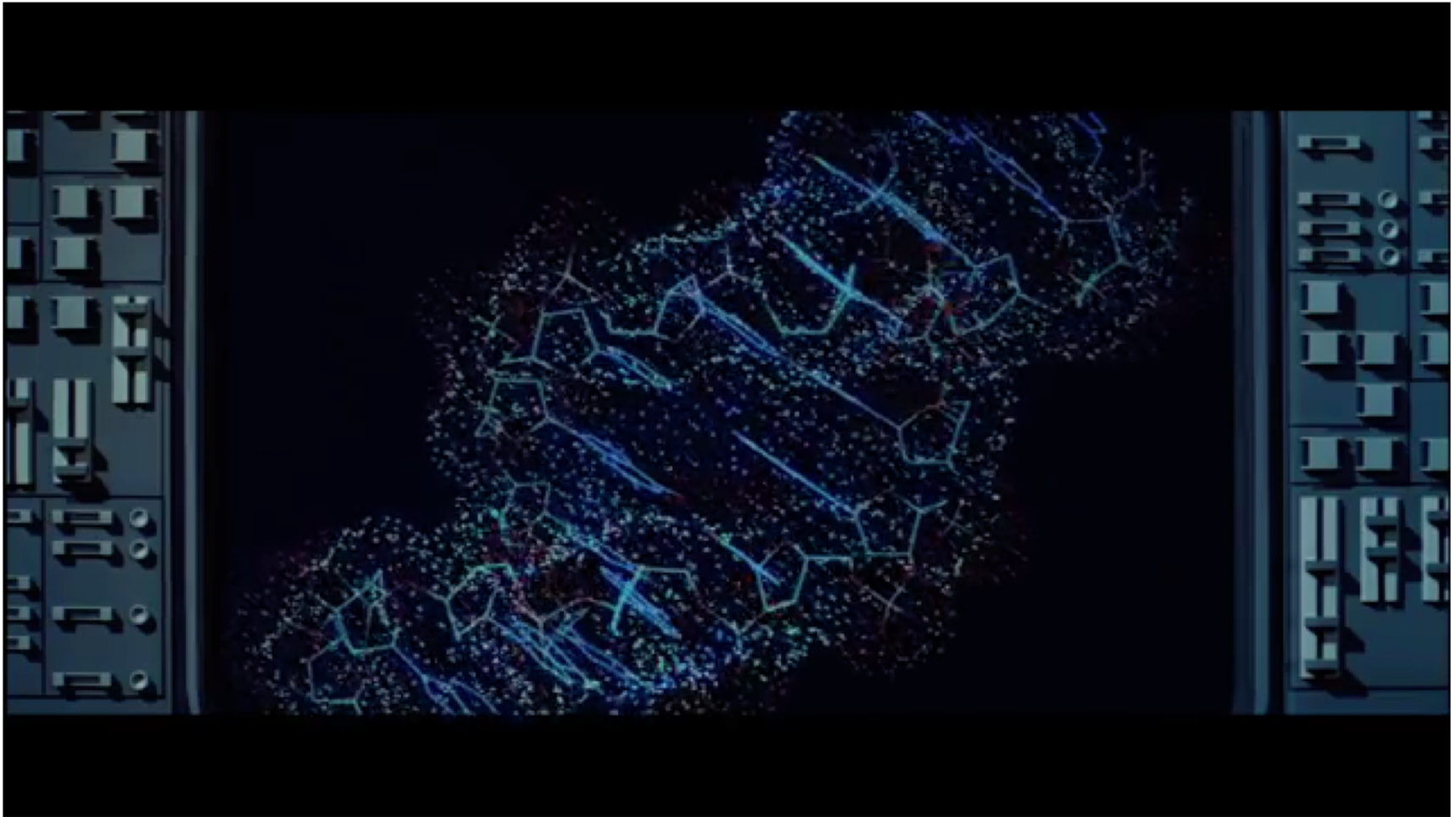
BFire out Planet



Composite

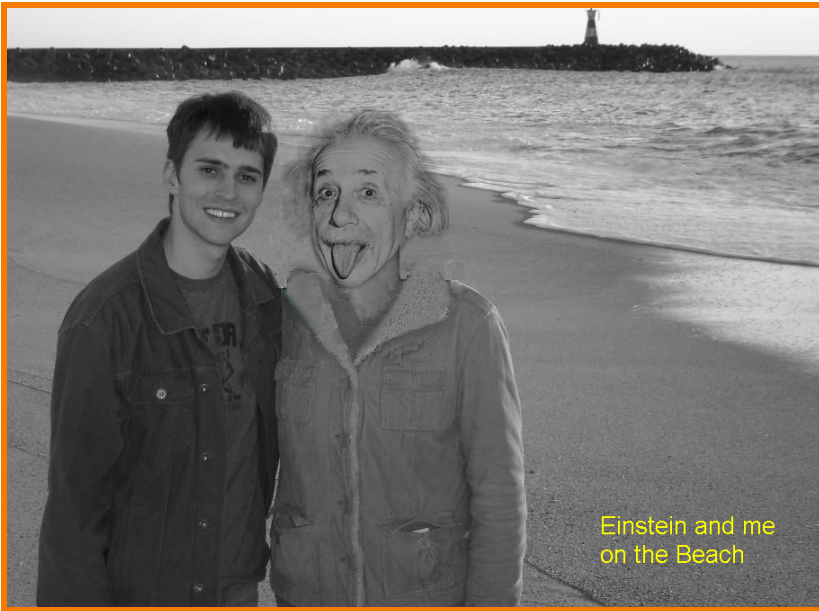
[Porter&Duff *Computer Graphics* 18:3 1984]

Image Composition Example



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

COS426 Examples



Darin Sleiter

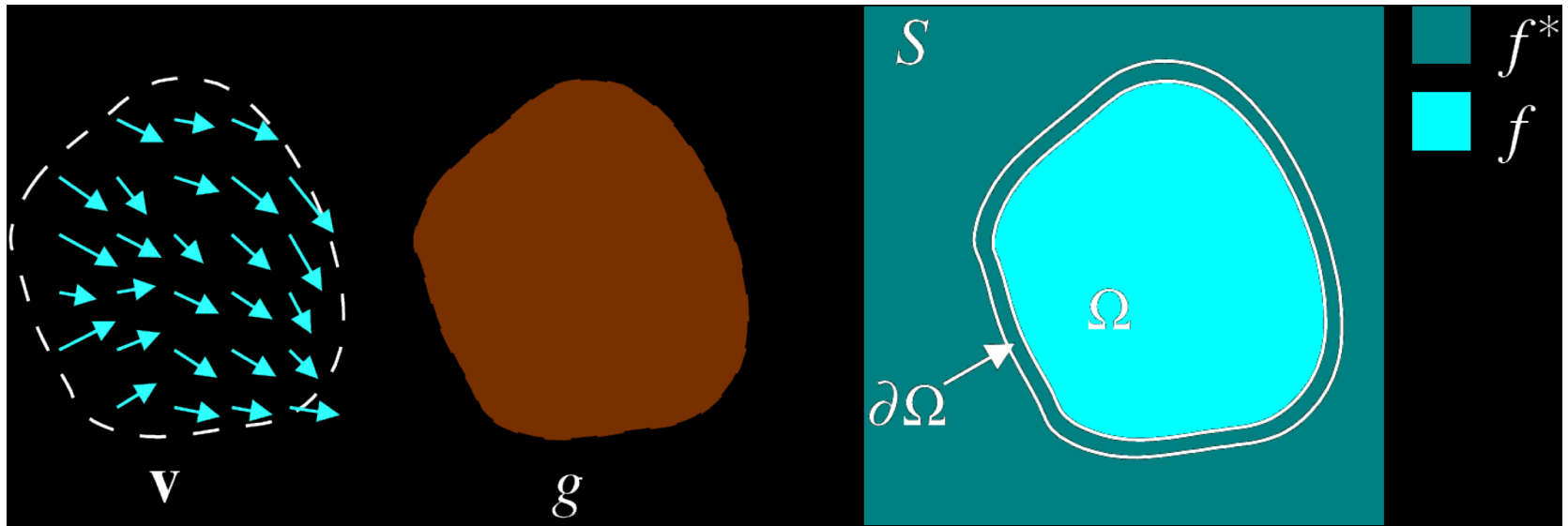


Kenrick Kin

Poisson Image Blending

Beyond simple compositing

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

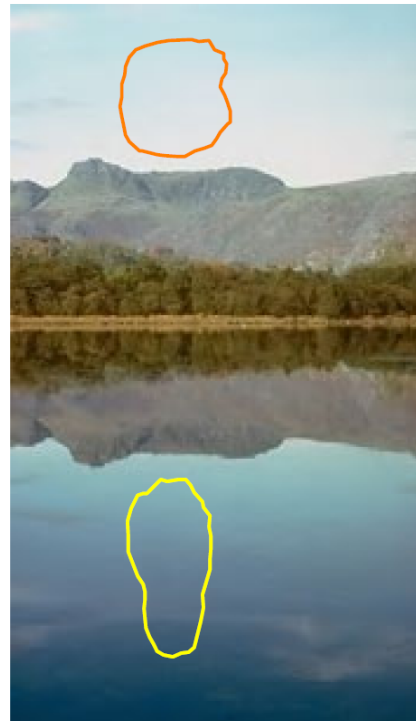


$$\begin{cases} \nabla^2 f = \nabla \cdot \mathbf{v} \\ f|_{\partial\Omega} = f^*|_{\partial\Omega} \end{cases}$$

Poisson Image Blending



sources



destinations



cloning



seamless cloning

Poisson Image Blending



source/destination



cloning



seamless cloning

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



(a)



(b)



(c)

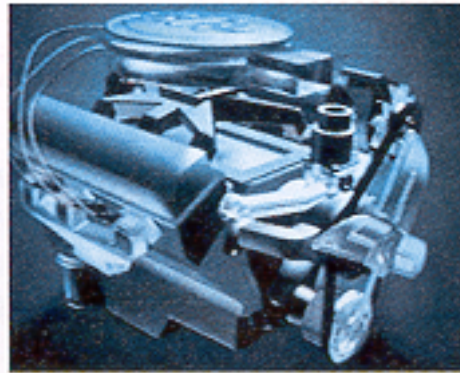
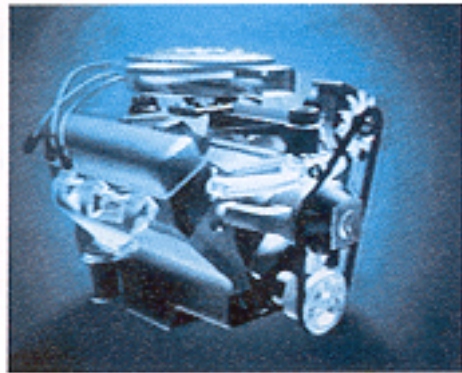


Figure 16-9

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

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

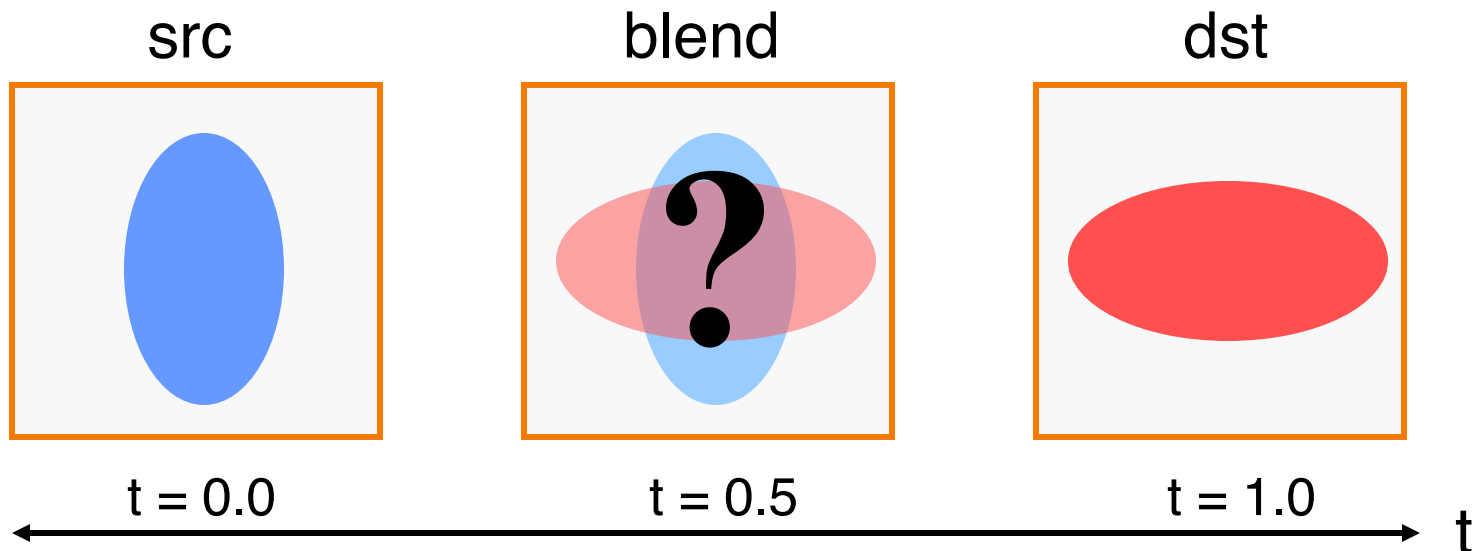
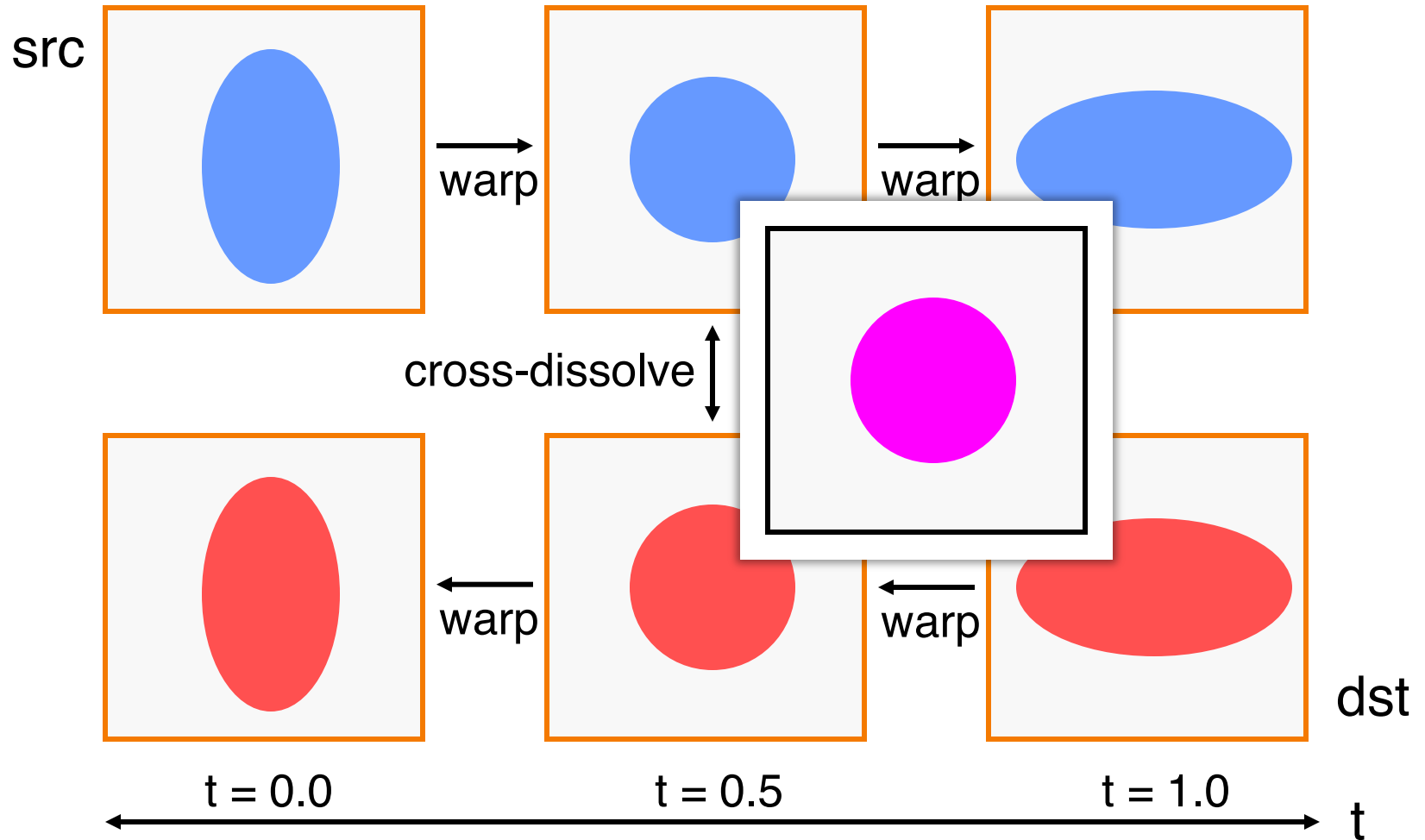


Image Morphing



- Combines warping and cross-dissolving



Beier & Neeley Example



Image₀

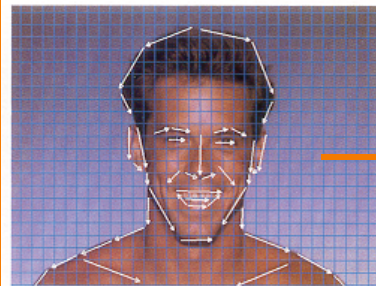


Figure 7

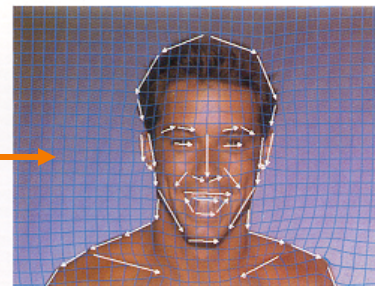


Figure 10

Warp₀

Result

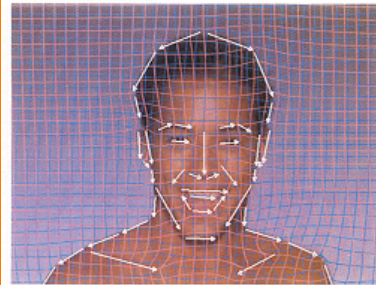


Figure 8

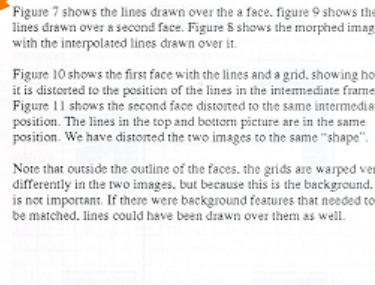
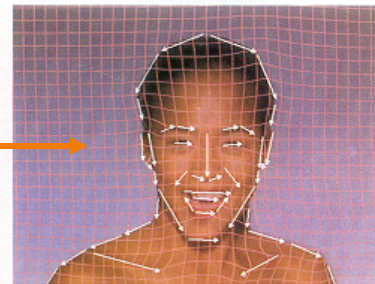
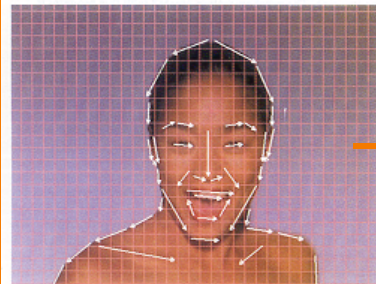


Figure 7 shows the lines drawn over the a face. figure 9 shows the lines drawn over a second face. Figure 8 shows the morphed image, with the interpolated lines drawn over it.

Figure 10 shows the first face with the lines and a grid, showing how it is distorted to the position of the lines in the intermediate frame. Figure 11 shows the second face distorted to the same intermediate position. The lines in the top and bottom picture are in the same position. We have distorted the two images to the same "shape".

Note that outside the outline of the faces, the grids are warped very differently in the two images, but because this is the background, it is not important. If there were background features that needed to be matched, lines could have been drawn over them as well.

Image₁

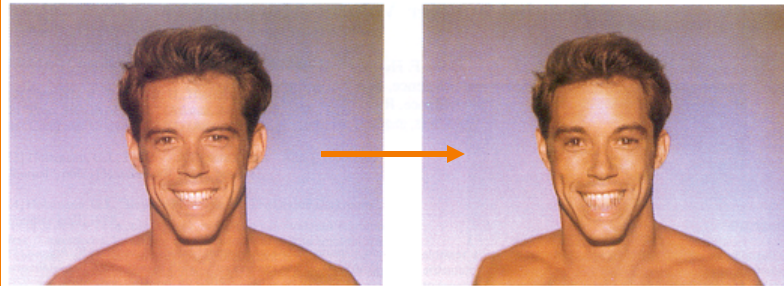


Warp₁

Beier & Neeley Example



Image₀



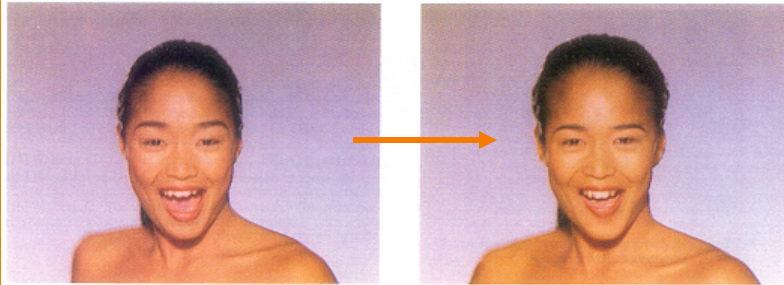
Warp₀

Result



The final sequence is figures 14, 12, and 10.
Figure 12 is the first face distorted to the intermediate position without the grid or lines. Figure 13 is the second face distorted toward that same position. Note that the blend between the two distorted images is much more like than the either of the distorted images themselves. We have noticed this happens very frequently.

Image₁



Warp₁

Beier & Neeley Example



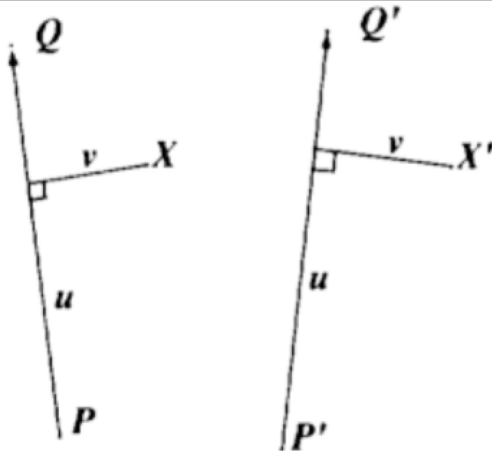
Black or White, Michael Jackson (1991)

Warping Pseudocode



```
WarpImage(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
```

Warping Pixel Locations



The original basis

The warped basis

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

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

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

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.

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

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 in to 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)
    foreach pixel p in FinalImage do
      Result(p) = (1-t) Warp0 + t Warp1
    end
  end
end
```

COS426 Example

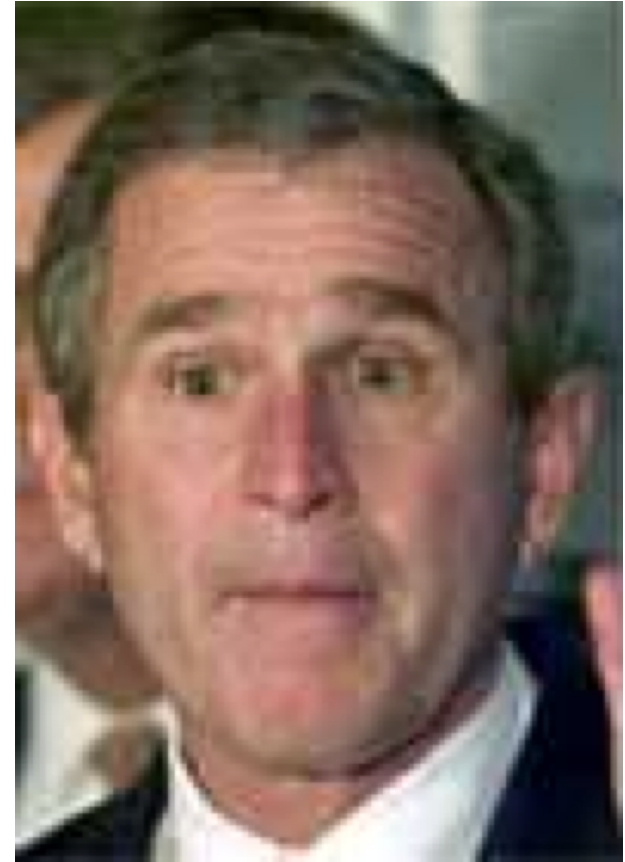


Amy Ousterhout

COS426 Examples



ckctwo



Jon Beyer

COS426 Examples



Sam Payne



Matt Matl

COS426 Examples



MakeAGIF.com

atran

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

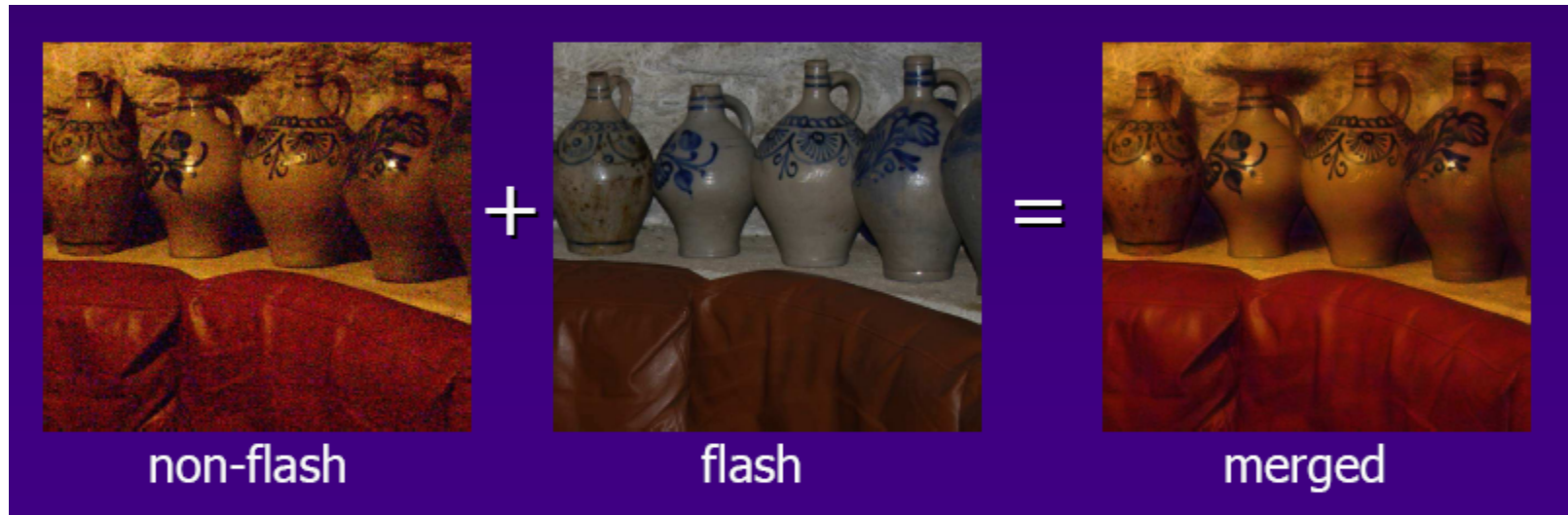


Image Composition Applications



- Photo montage

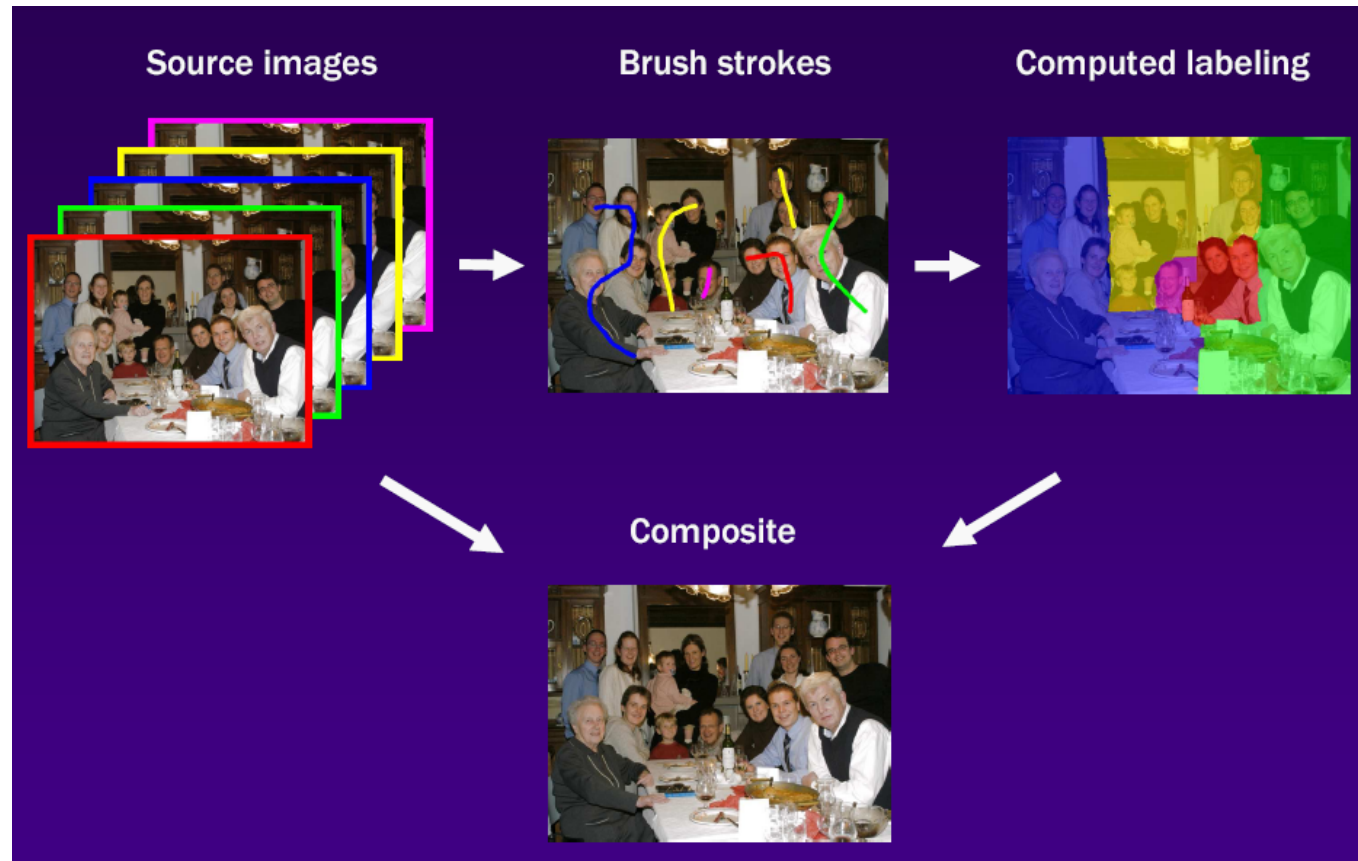


Image Composition Applications



- Photo montage



[Michael Cohen]

Image Composition Applications



- Stoboscopic images



[Michael Cohen]

Image Composition Applications



- Extended depth-of-field

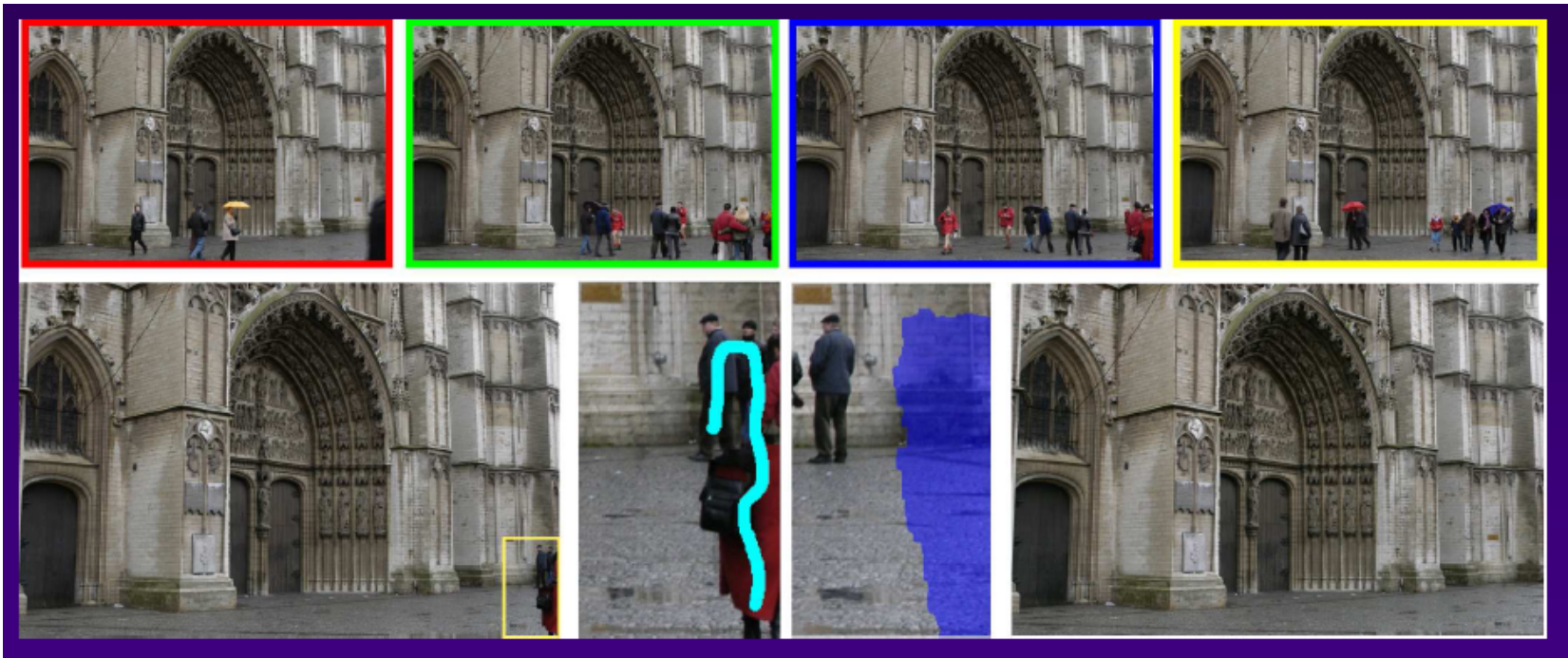


[Michael Cohen]

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

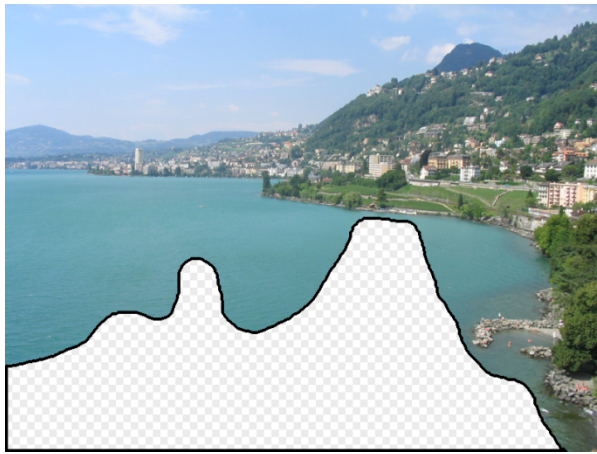
2.3 Million unique images from Flickr



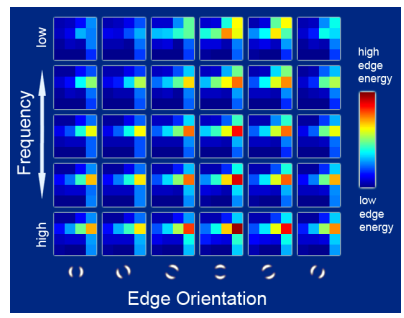


Scene Completion Result

Image Completion Algorithm



Input image



Scene Descriptor

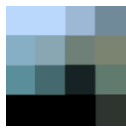
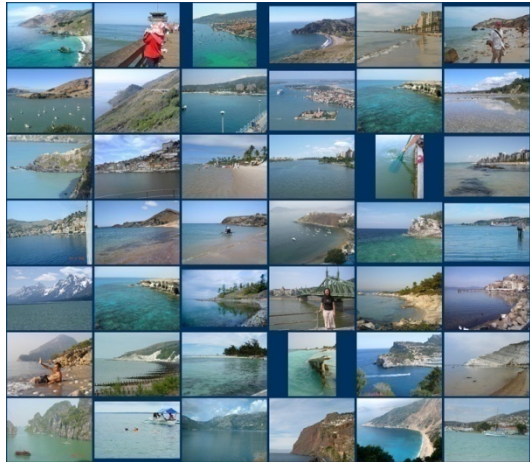


Image Collection



200 matches



Mosaicing



20 completions











Summary



- Image compositing
 - Alpha channel
 - Porter-Duff compositing algebra
- Image morphing
 - Warping
 - Compositing
- Computational photography

Next Time: 3D Modeling



Hoppe