Image Compositing & Morphing

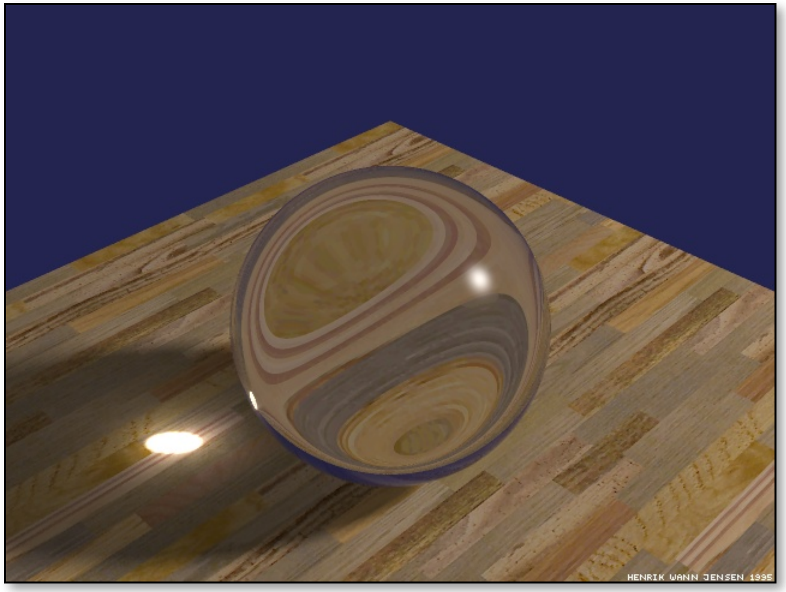
Adam Finkelstein
COS 426, Spring 2018
Princeton University

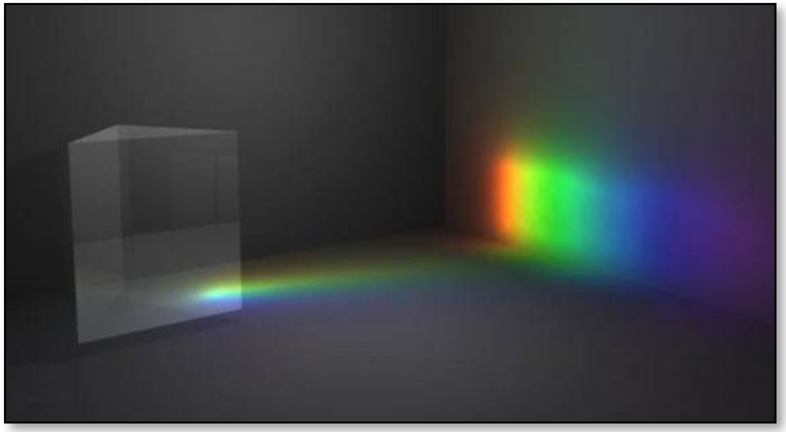
Digital Image Processing

- 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

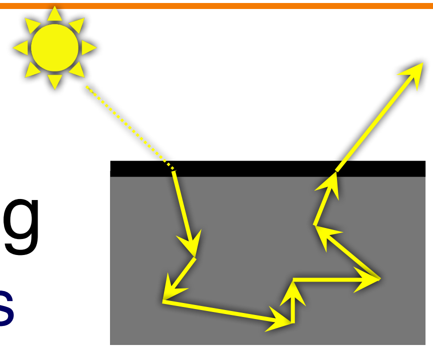
- Refraction
  - Light is bent as it goes through an object
  - Can focus light: caustics
  - Can be color-dependent: dispersion





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

Jurassic Park (1993)

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

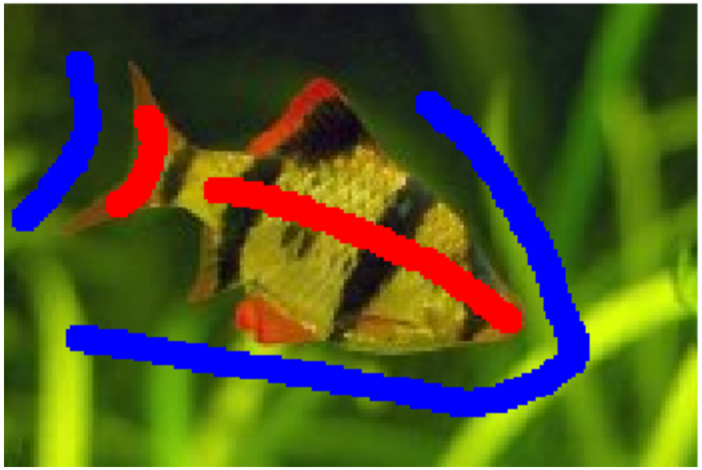
Image Segmentation

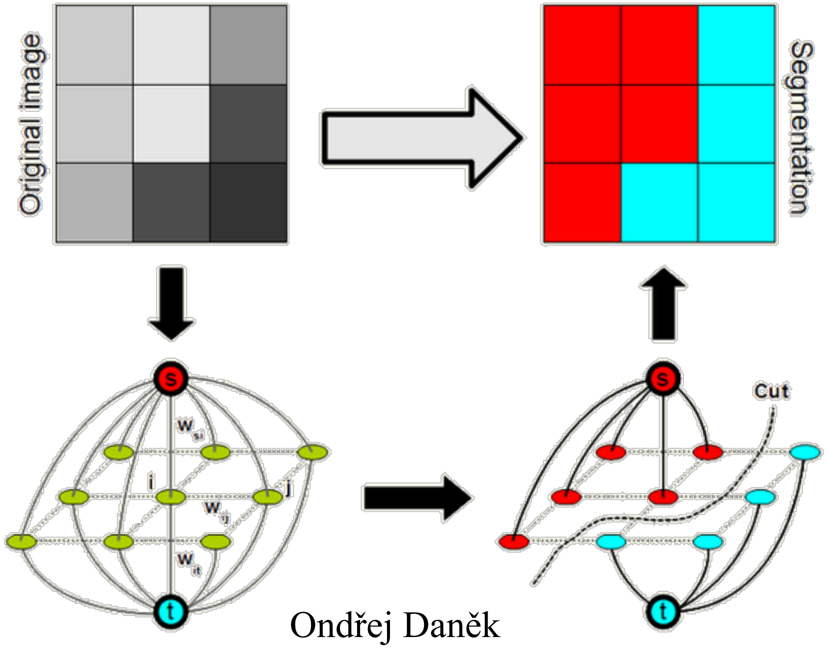
- Chroma keying (blue- or green-screen)
  - Photograph object in front of screen with known color

Rosco Spectrum

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)





Ondřej Daněk

Image Segmentation

- Novel methods, e.g. flash matting

Sun et al., 2006

Image Composition

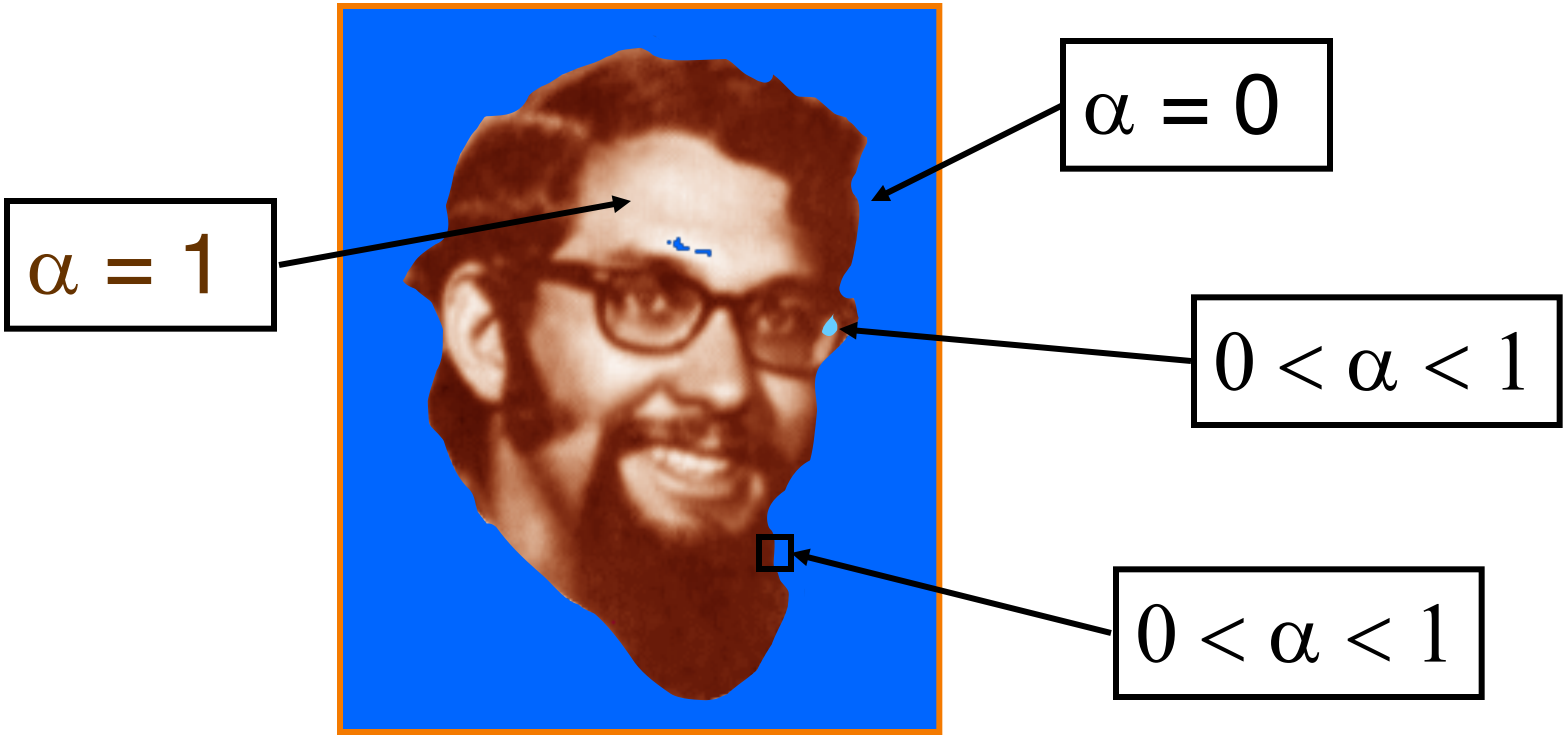
- Issues:
  - Segmenting image into regions
  - Blending into single image seamlessly

Image Blending

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

Blending with Alpha Channel

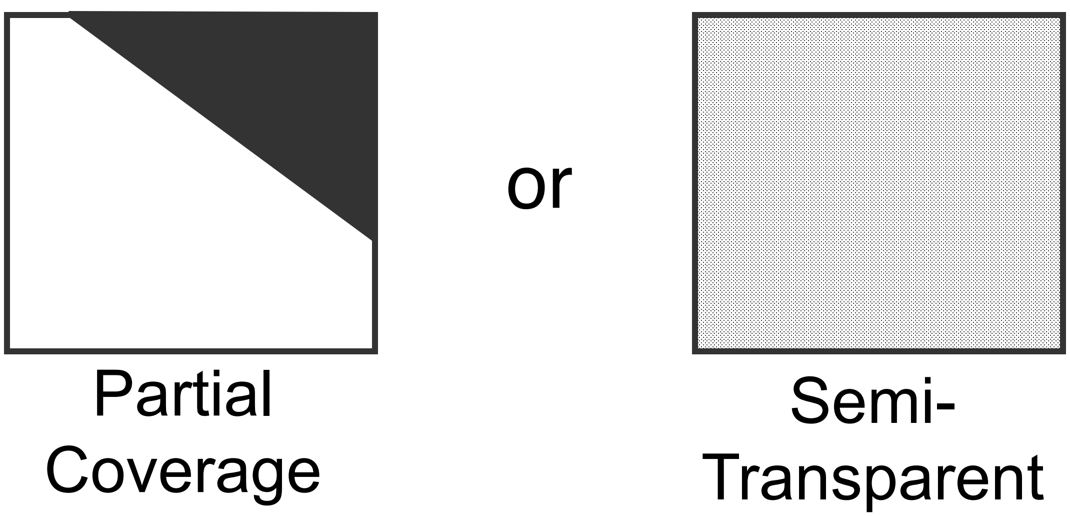
Per-pixel "alpha" channel: controls the linear interpolation between foreground and background pixels when elements are composited.



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)

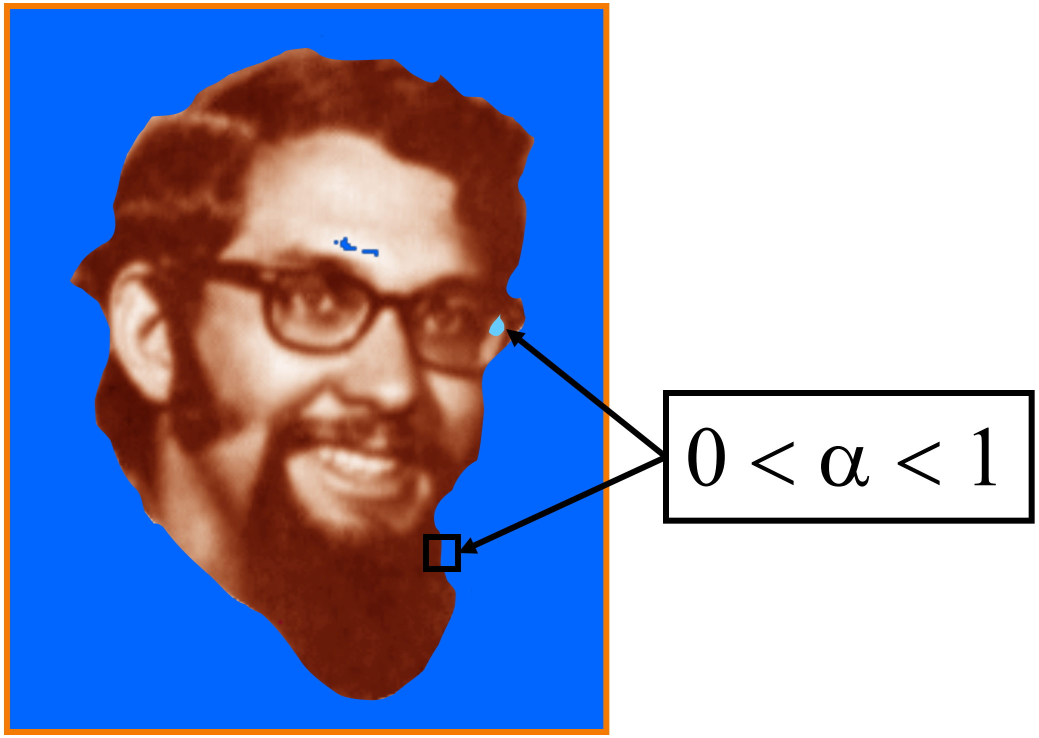
- Example: $\alpha = 0.3$



Alpha Blending: "Over" Operator

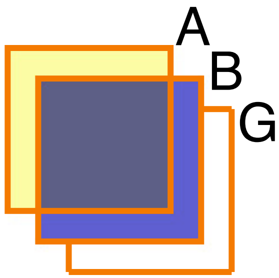
$C = A$ over B

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



Compositing Algebra

- Suppose we put A over B over background G



  - How much of B is blocked by A?

    α_A

  - How much of B shows through A

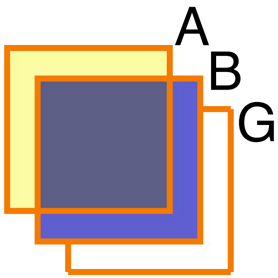
    $(1-\alpha_A)$

  - 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?

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

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

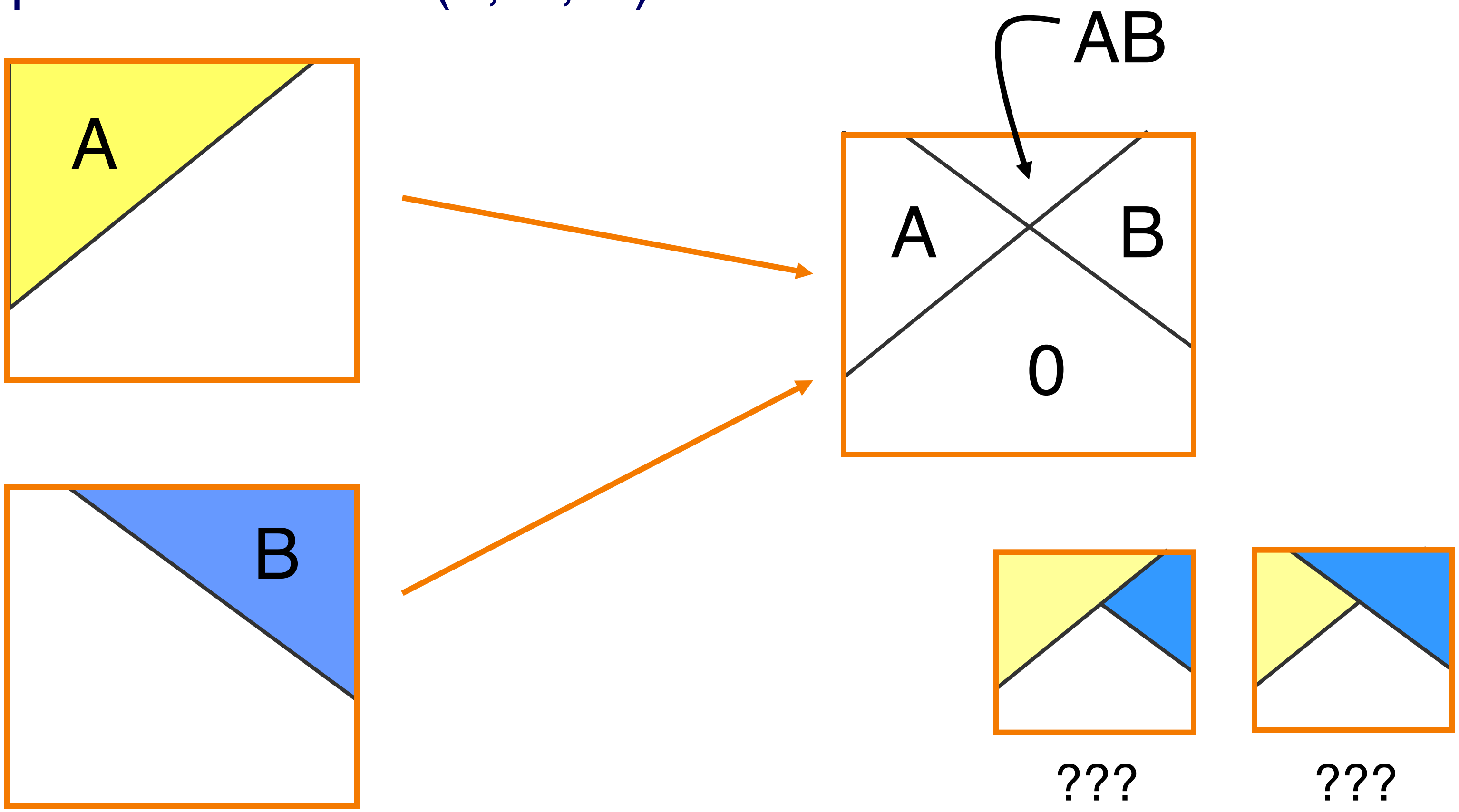
$$= A \text{ over } [B \text{ over } G]$$

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?!?

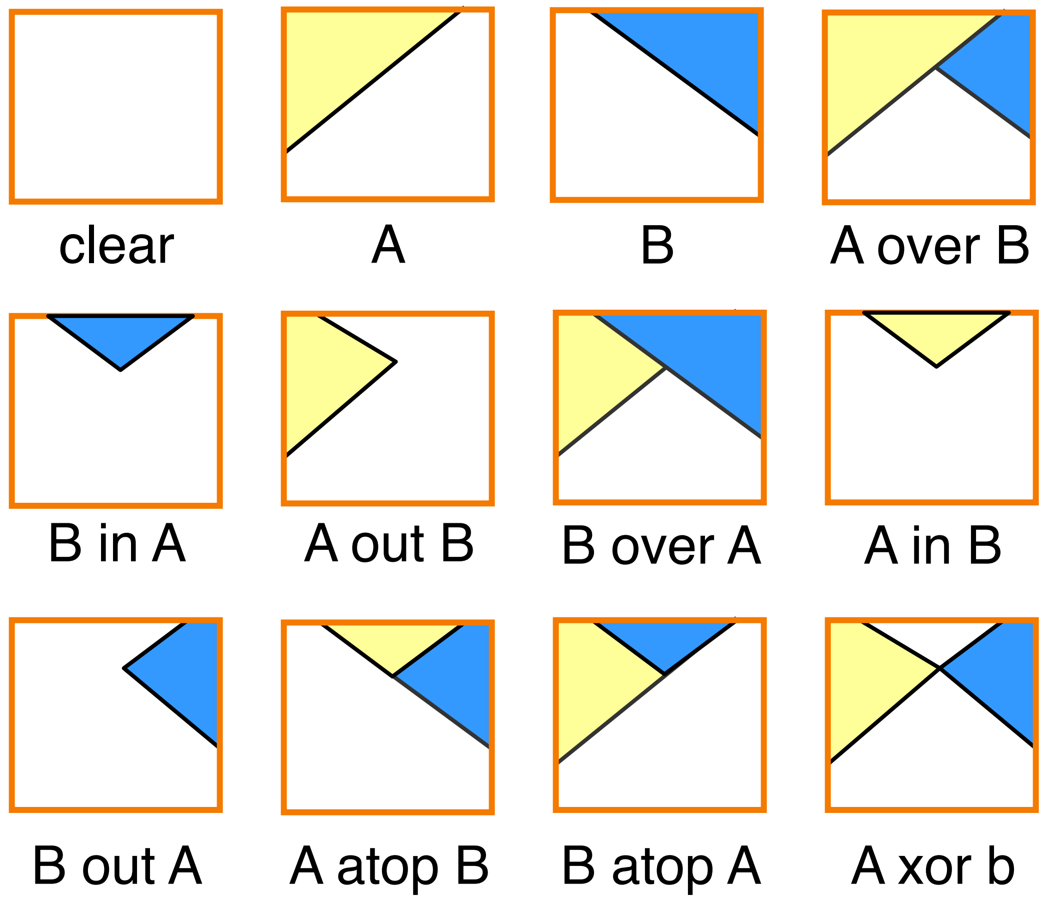


Blending with Alpha

Composition algebra – 12 combinations

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

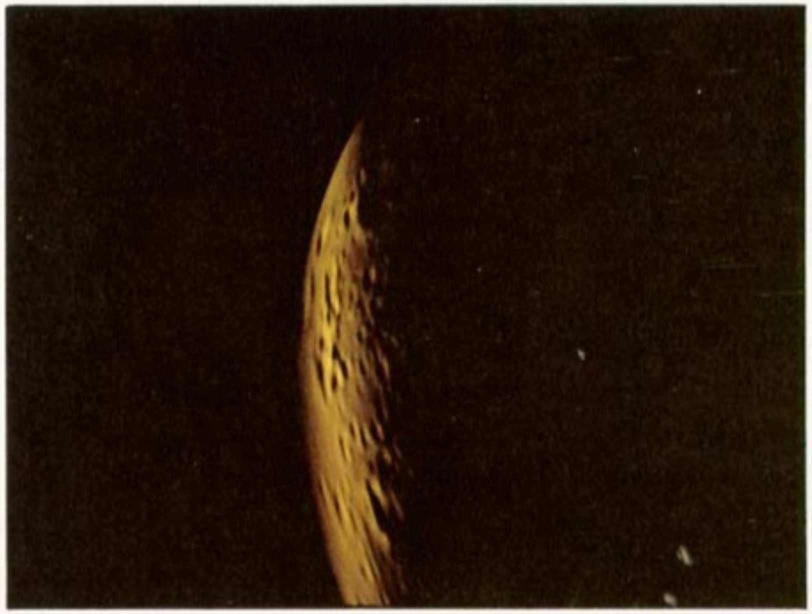
Operation	F_A	F_B
Clear	0	0
A	1	0
B	0	1
A over B	1	$1-\alpha_A$
B over A	$1-\alpha_B$	1
A in B	α_B	0
B in A	0	α_A
A out B	$1-\alpha_B$	0
B out A	0	$1-\alpha_A$
A atop B	α_B	$1-\alpha_A$
B atop A	$1-\alpha_B$	α_A
A xor B	$1-\alpha_B$	$1-\alpha_A$



Porter & Duff `84

Image Composition Example

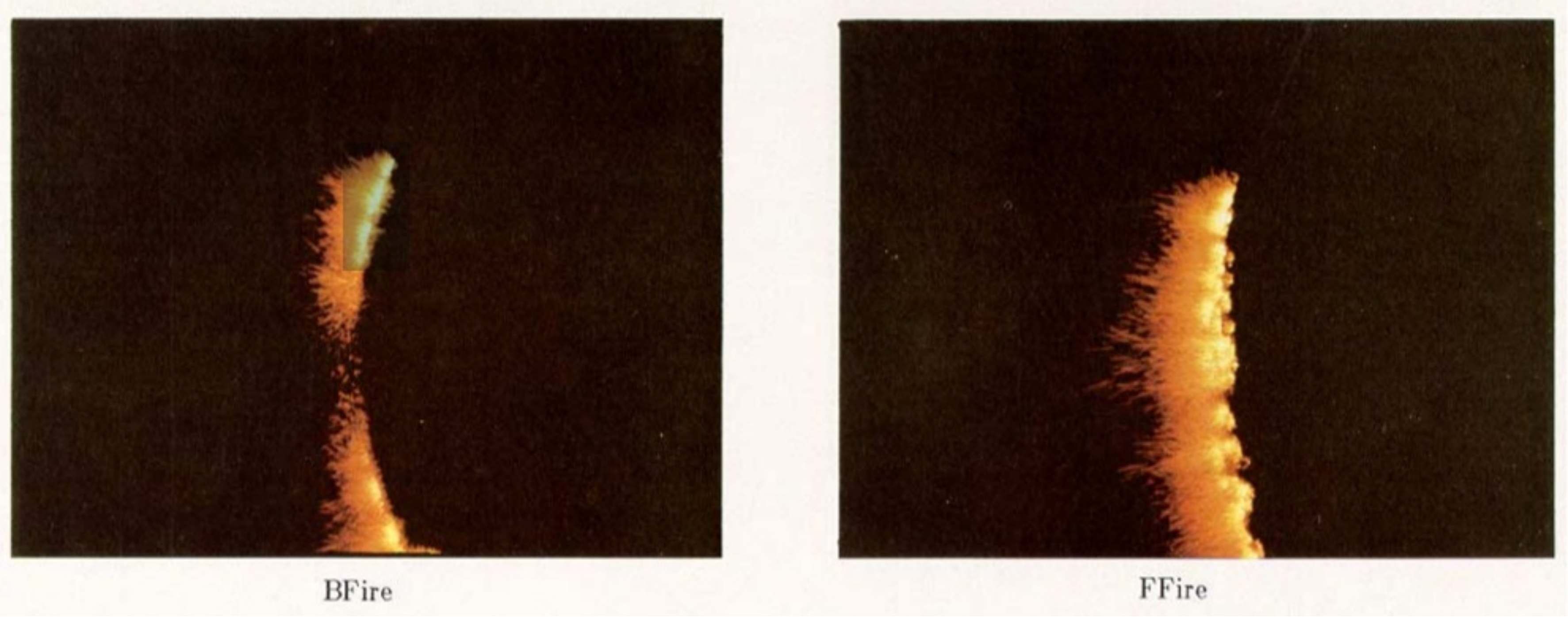
Stars



Planet

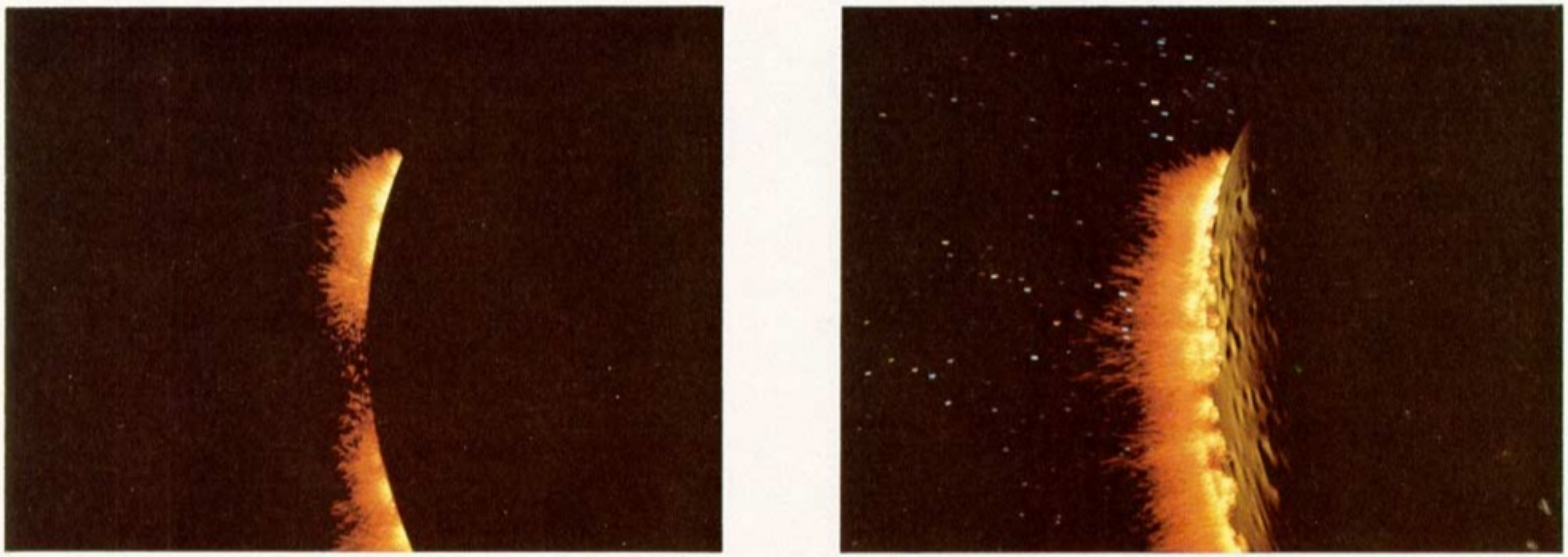
[Porter&Duff *Computer Graphics* 18:3 1984]

Image Composition Example



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

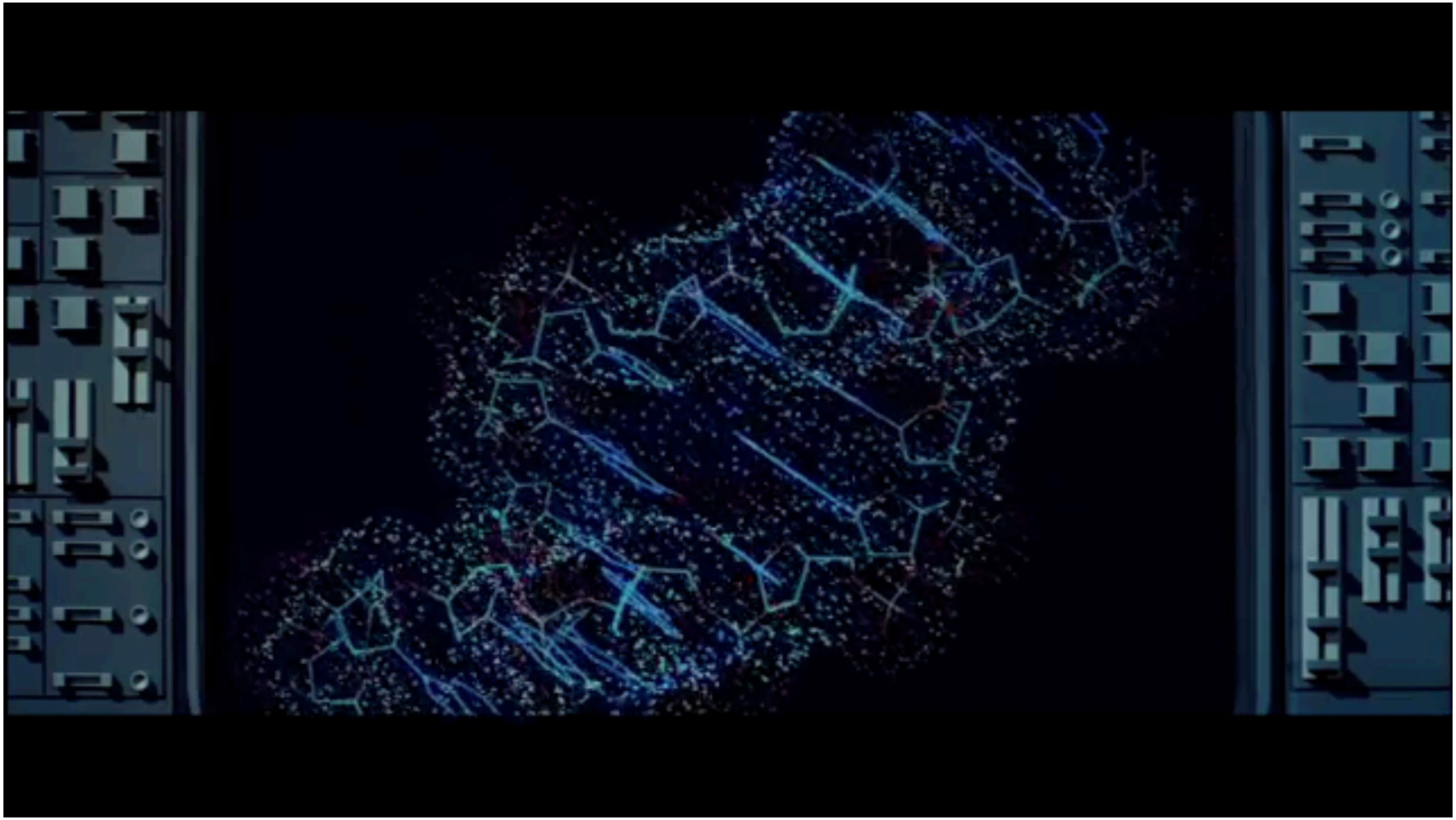
Image Composition Example



BFire **out** Planet Composite

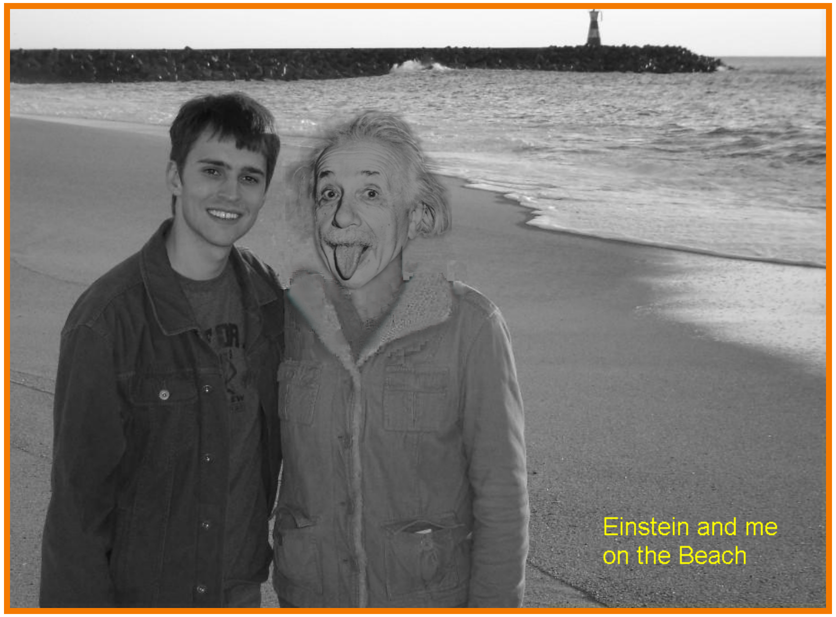
[Porter&Duff *Computer Graphics* 18:3 1984]

Image Composition Example



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

COS426 Examples



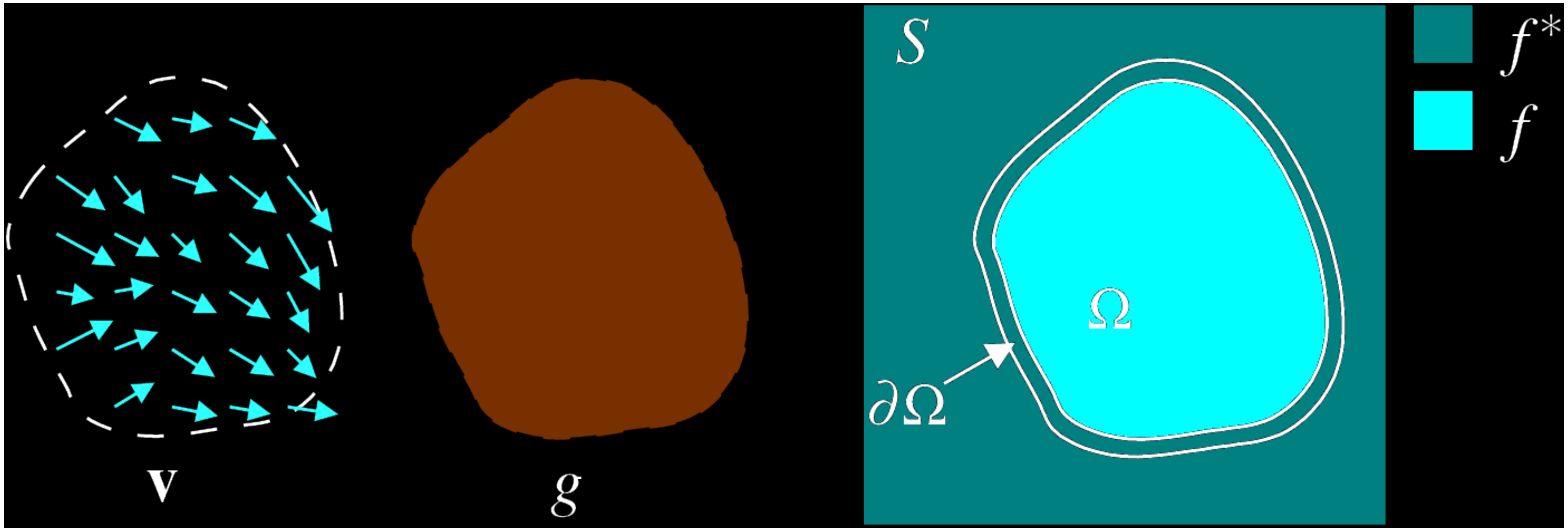
Darin Sleiter

Kenrick Kin

Poisson Image Blending

Beyond simple compositing

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

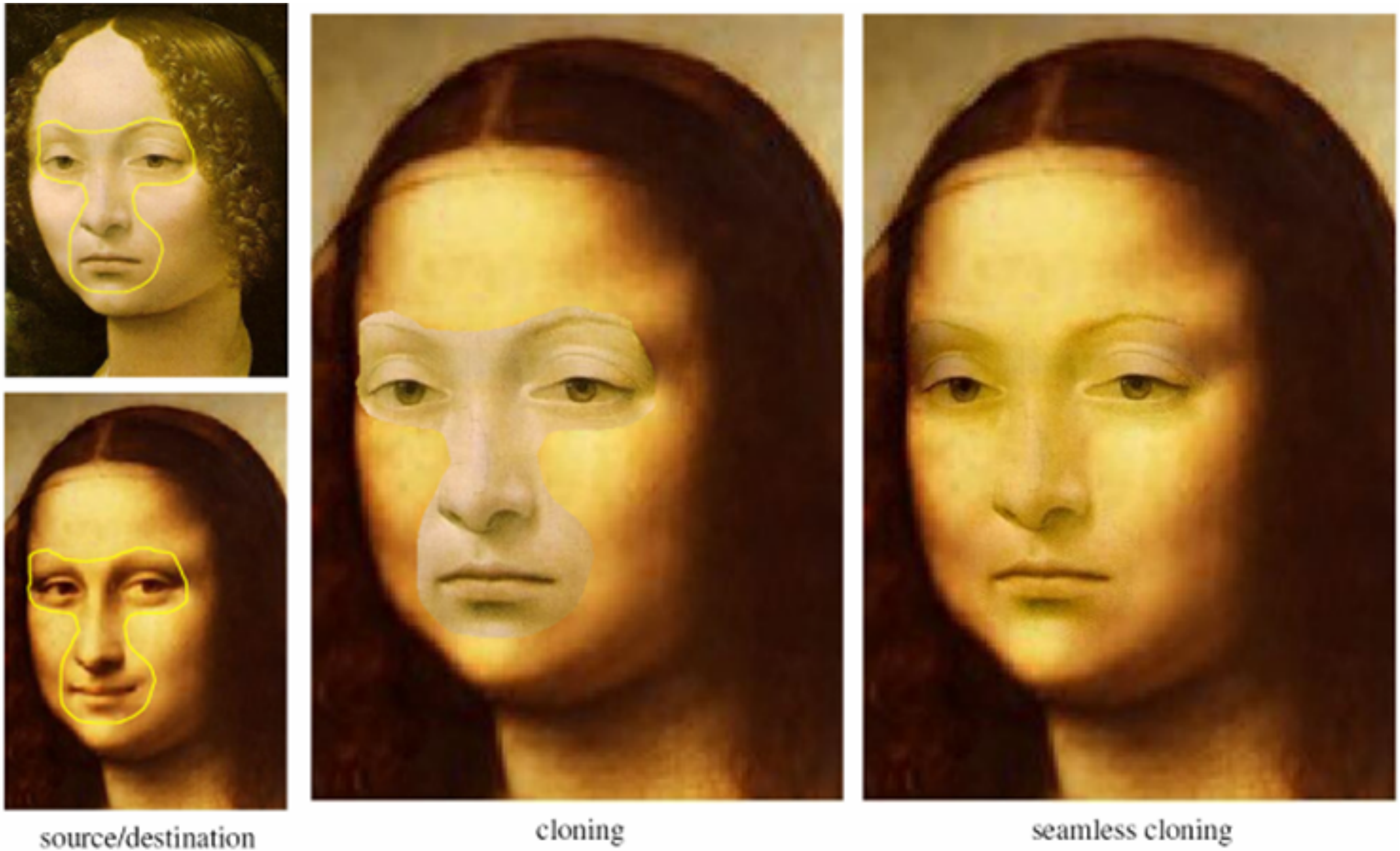


$$\begin{cases} \nabla^2 f = \nabla \cdot \mathbf{v} \\ f|_{\partial\Omega} = f^*|_{\partial\Omega} \end{cases}$$

Poisson Image Blending

sources destinations cloning seamless cloning

Poisson Image Blending



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

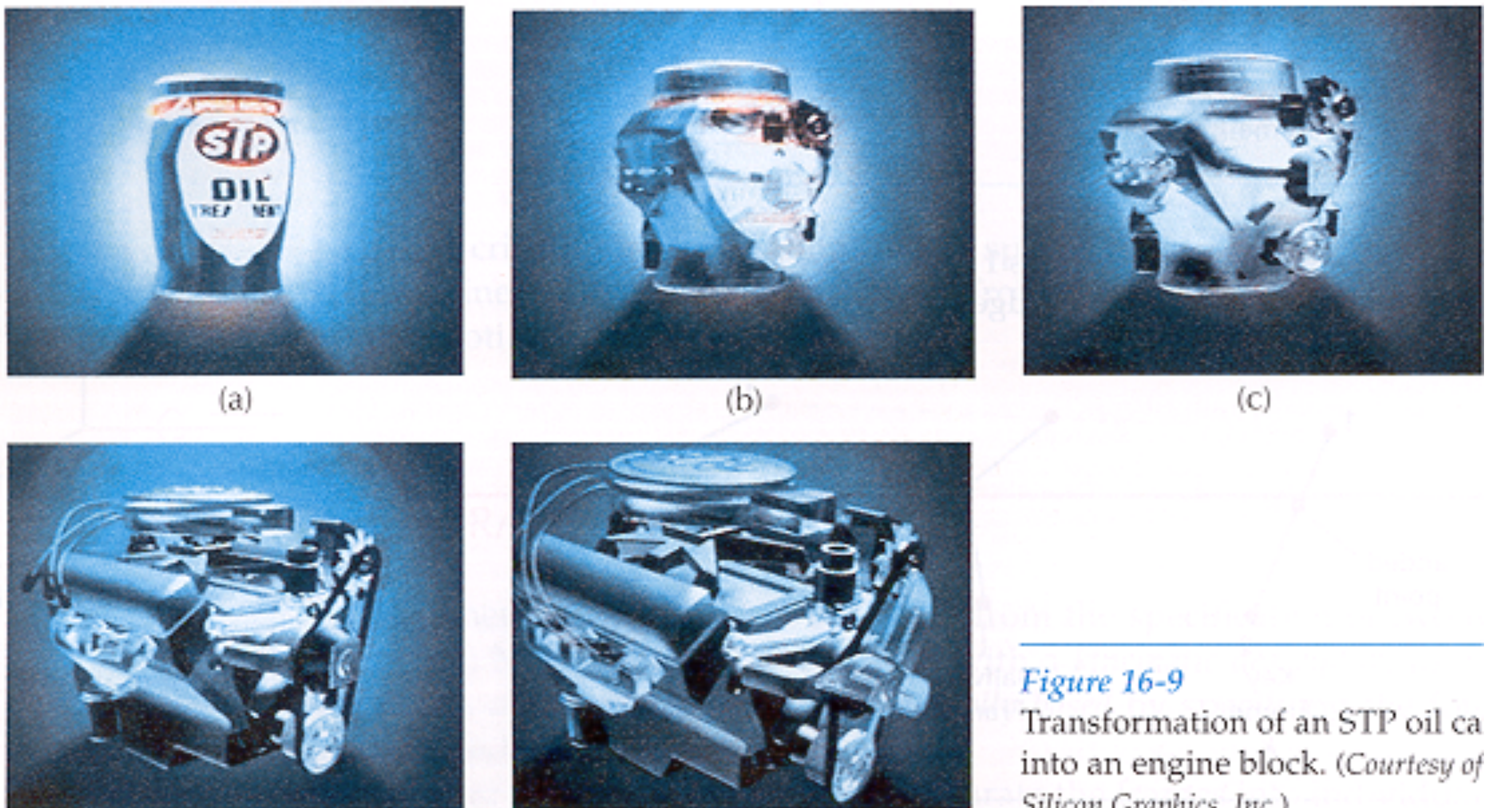


Figure 16-9
Transformation of an STP oil ca
into an engine block. (*Courtesy of Silicon Graphics, Inc.*)

H&B Figure 16.9

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

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

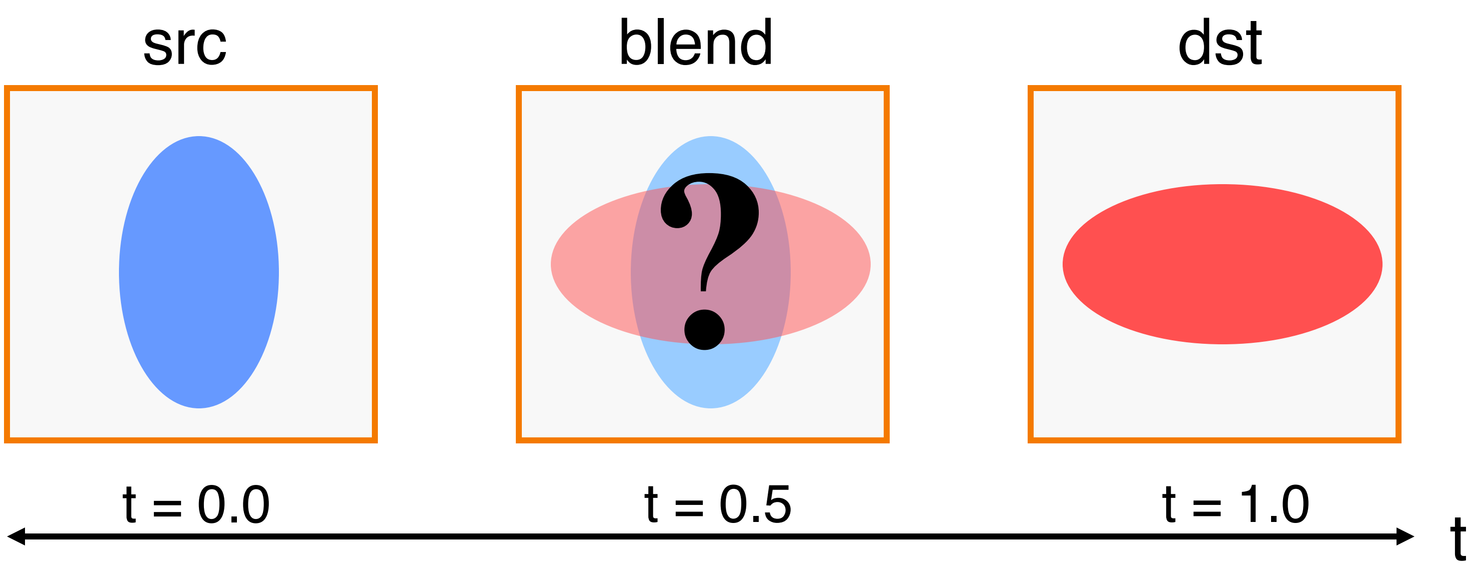
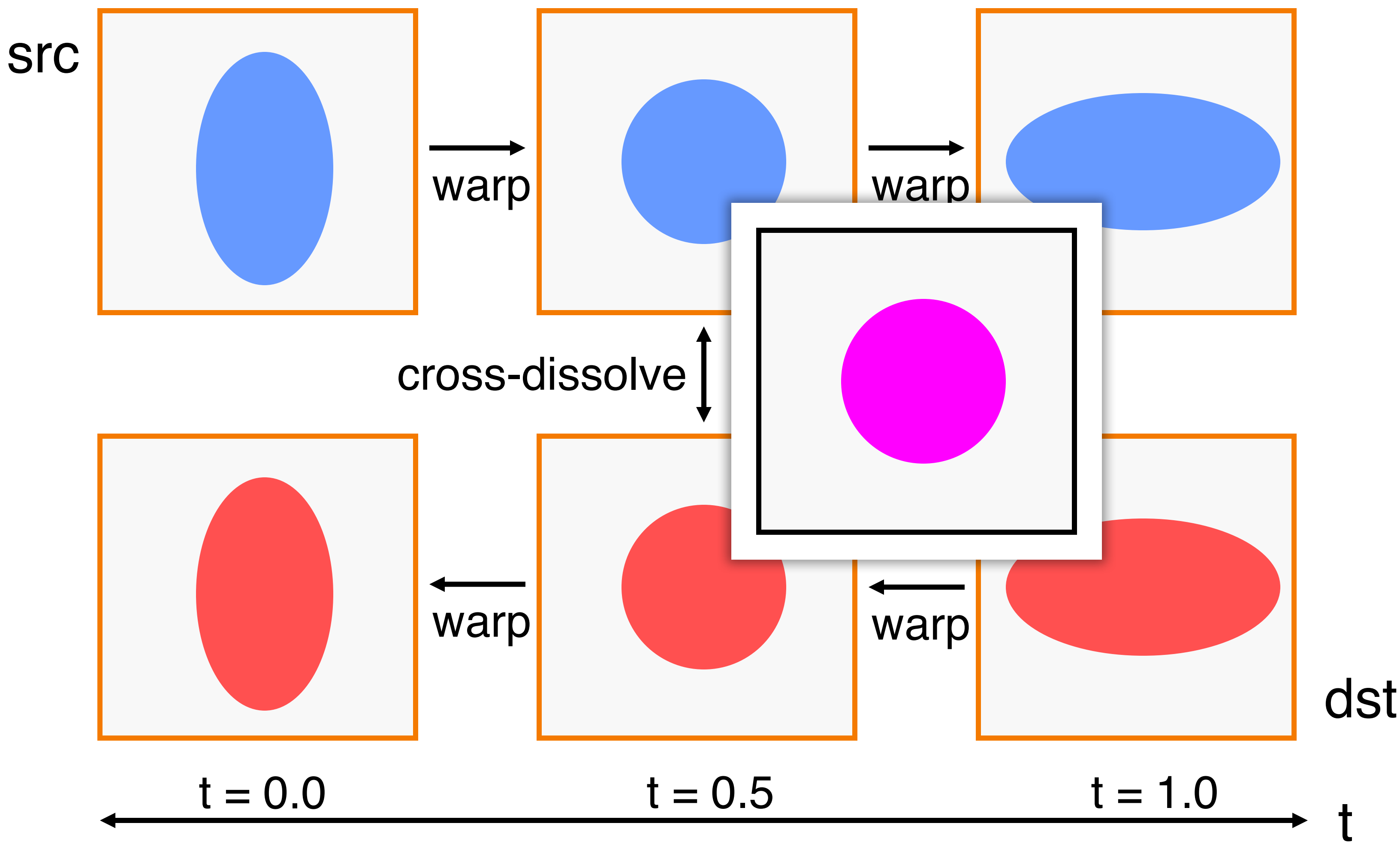


Image Morphing

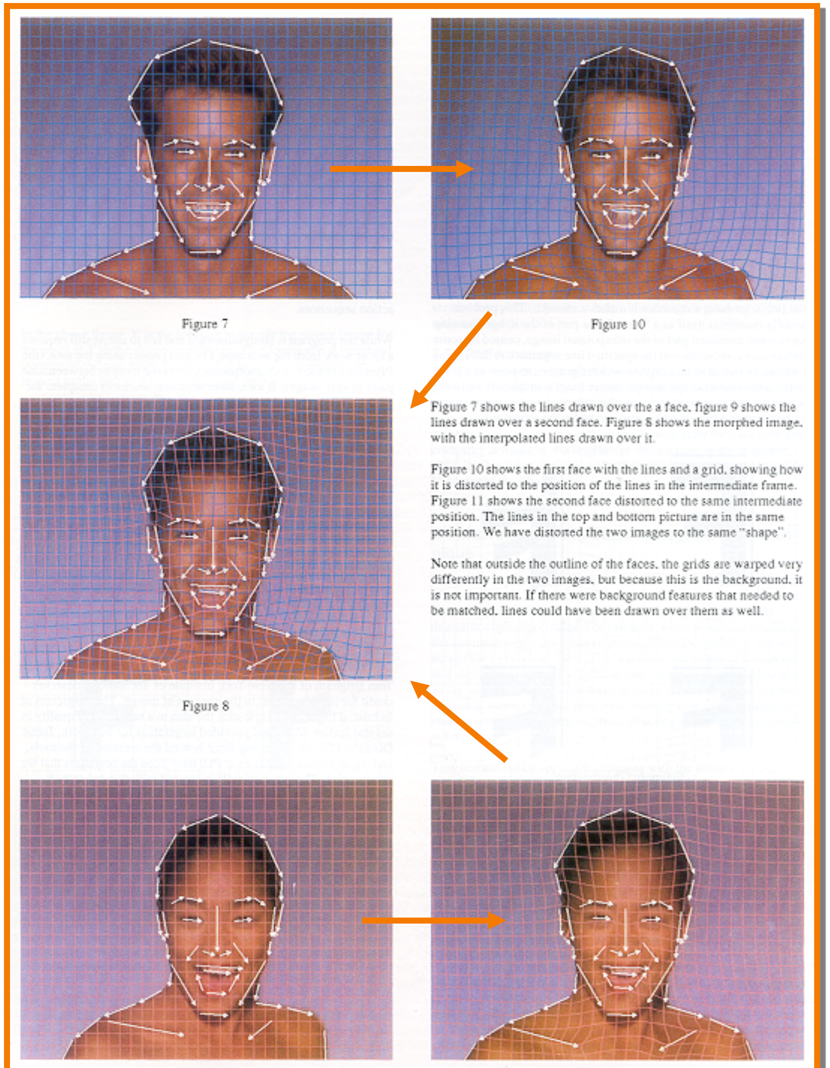
- Combines warping and cross-dissolving



Beier & Neeley Example

Image$_0$

Warp$_0$



Result

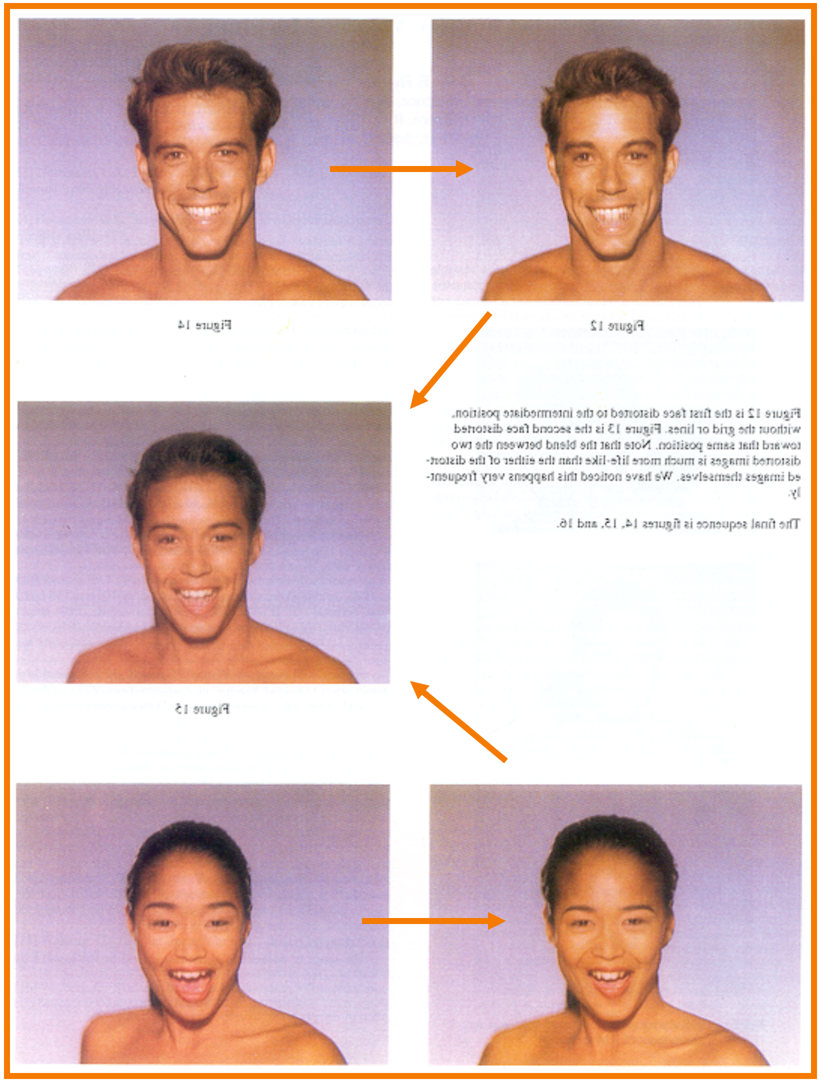
Image$_1$

Warp$_1$

Beier & Neeley Example

Image_0

Warp_0



Result

Image_1

Warp_1

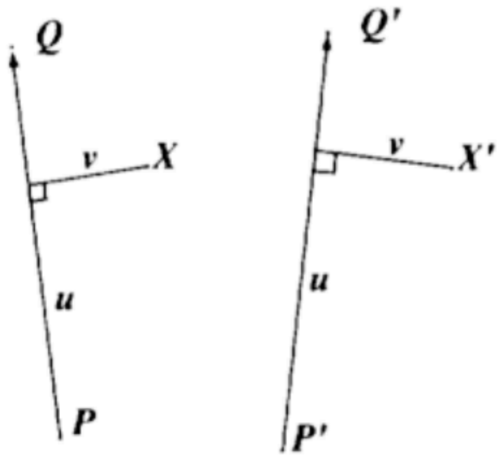
Beier & Neeley Example

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

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

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

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

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.

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

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.

Implementation notes from Evan Wallace, Brown University
http://cs.brown.edu/courses/csci1950-g/results/proj5/edwallac/

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)
        foreach pixel p in FinalImage do
            Result(p) = (1-t) Warp_0 + t Warp_1
        end
    end
end
```

COS426 Example

Amy Ousterhout

COS426 Examples

ckctwo

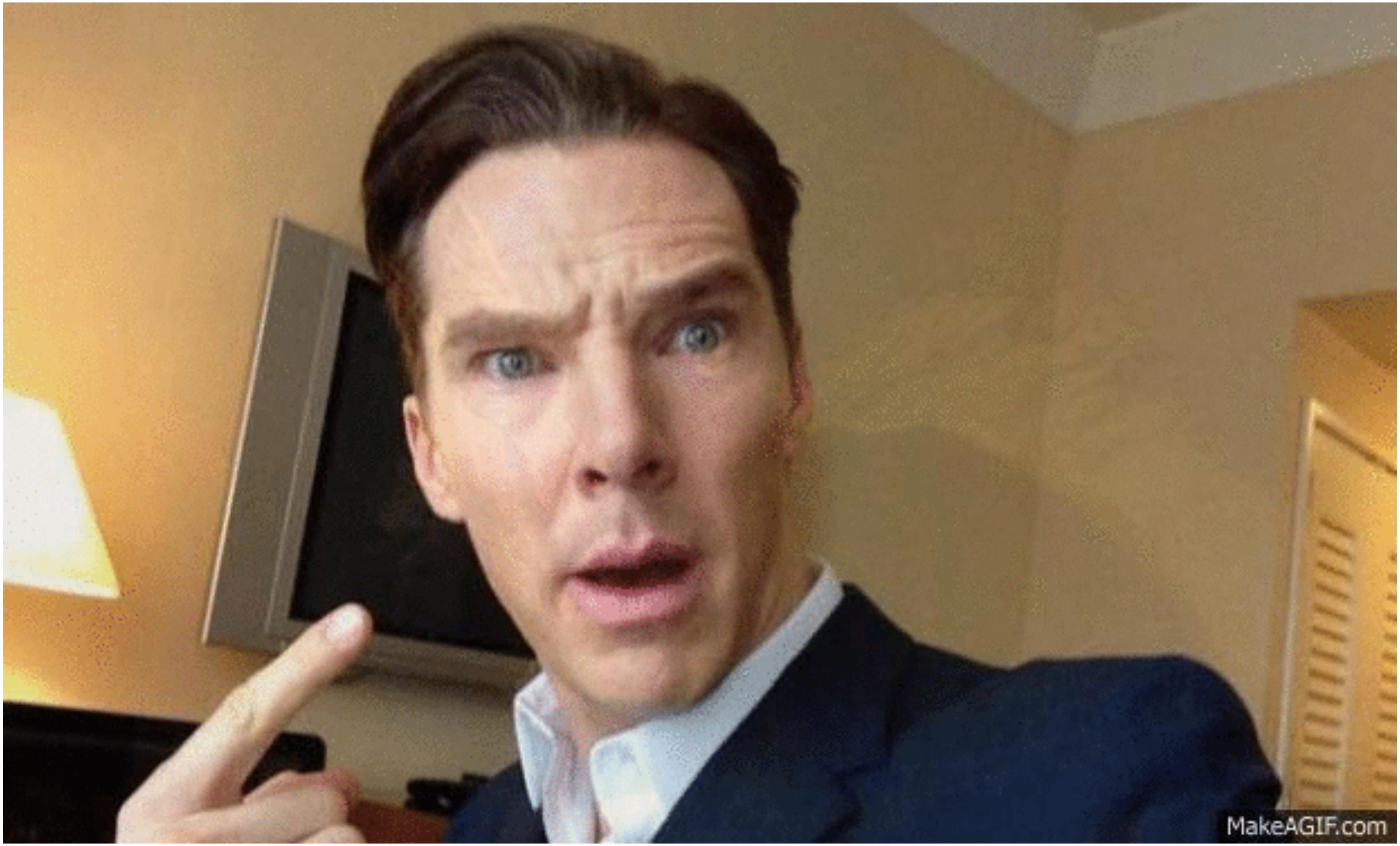
Jon Beyer

COS426 Examples

Sam Payne

Matt Matl

COS426 Examples



atran

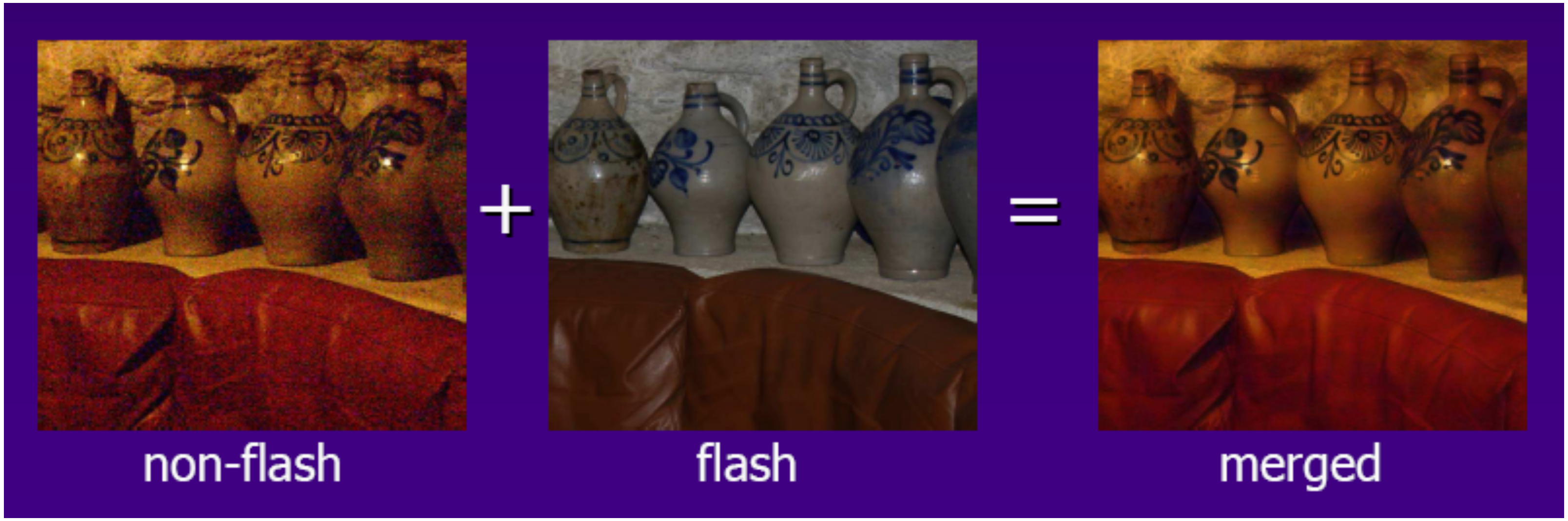
Image Composition Applications

- "Computational photography": enable new photographic effects that inherently use multiple images + computation
- Example: stitching images into a panorama

[Michael Cohen]

Image Composition Applications

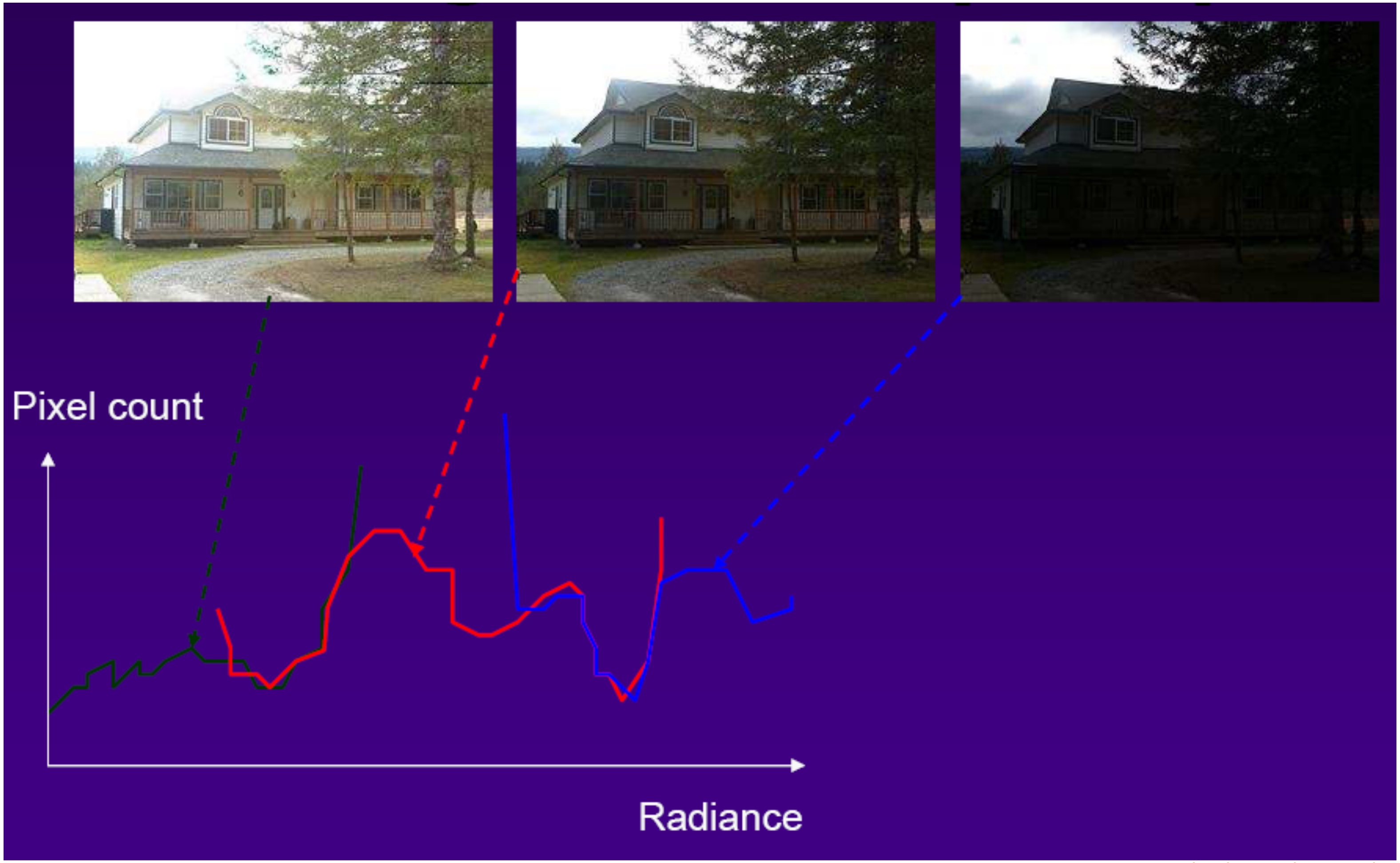
- Flash / No flash



[Michael Cohen]

Image Composition Applications

- High dynamic range images



[Michael Cohen]

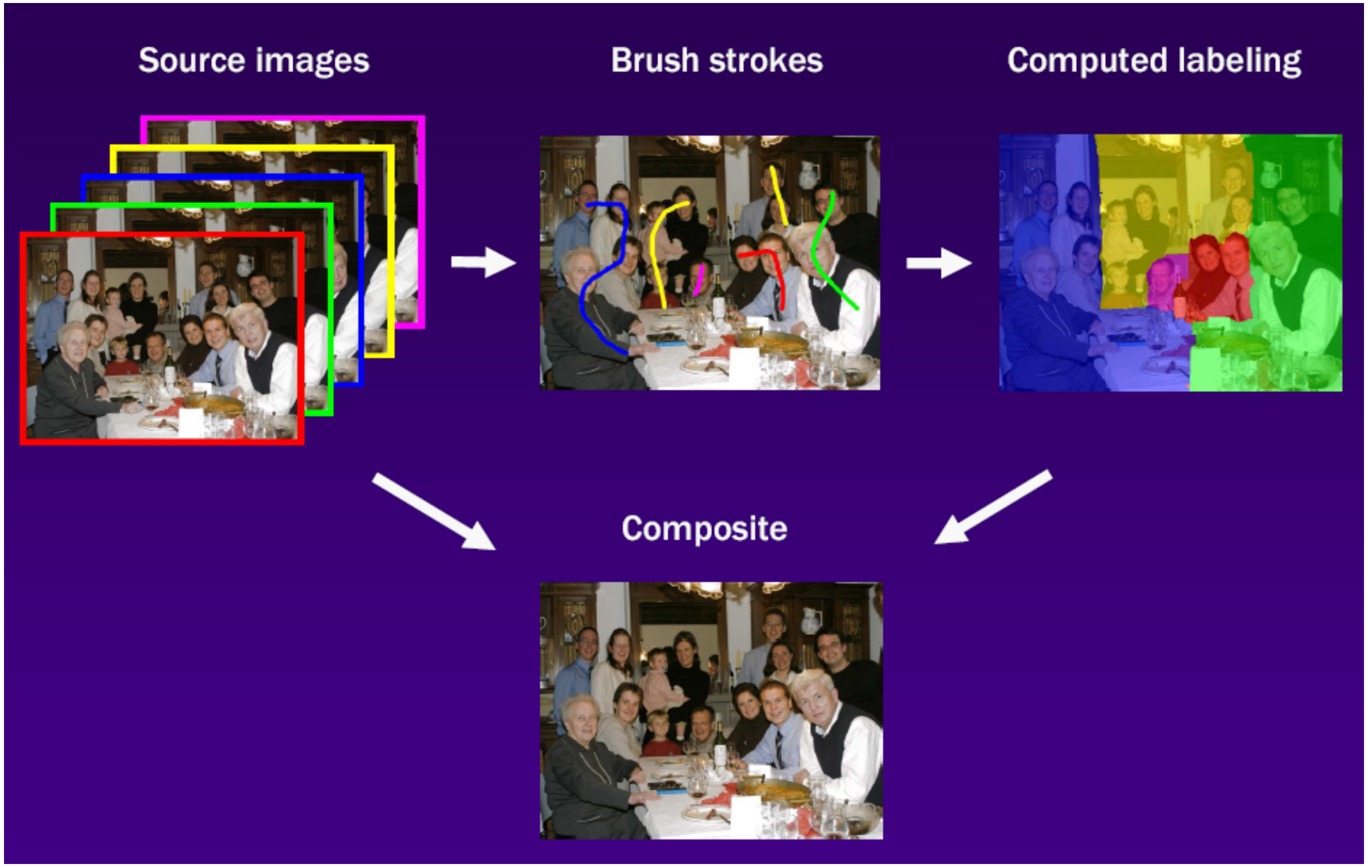
Image Composition Applications

- High dynamic range images

[Michael Cohen]

Image Composition Applications

- Photo montage



[Michael Cohen]

Image Composition Applications

- Photo montage

[Michael Cohen]

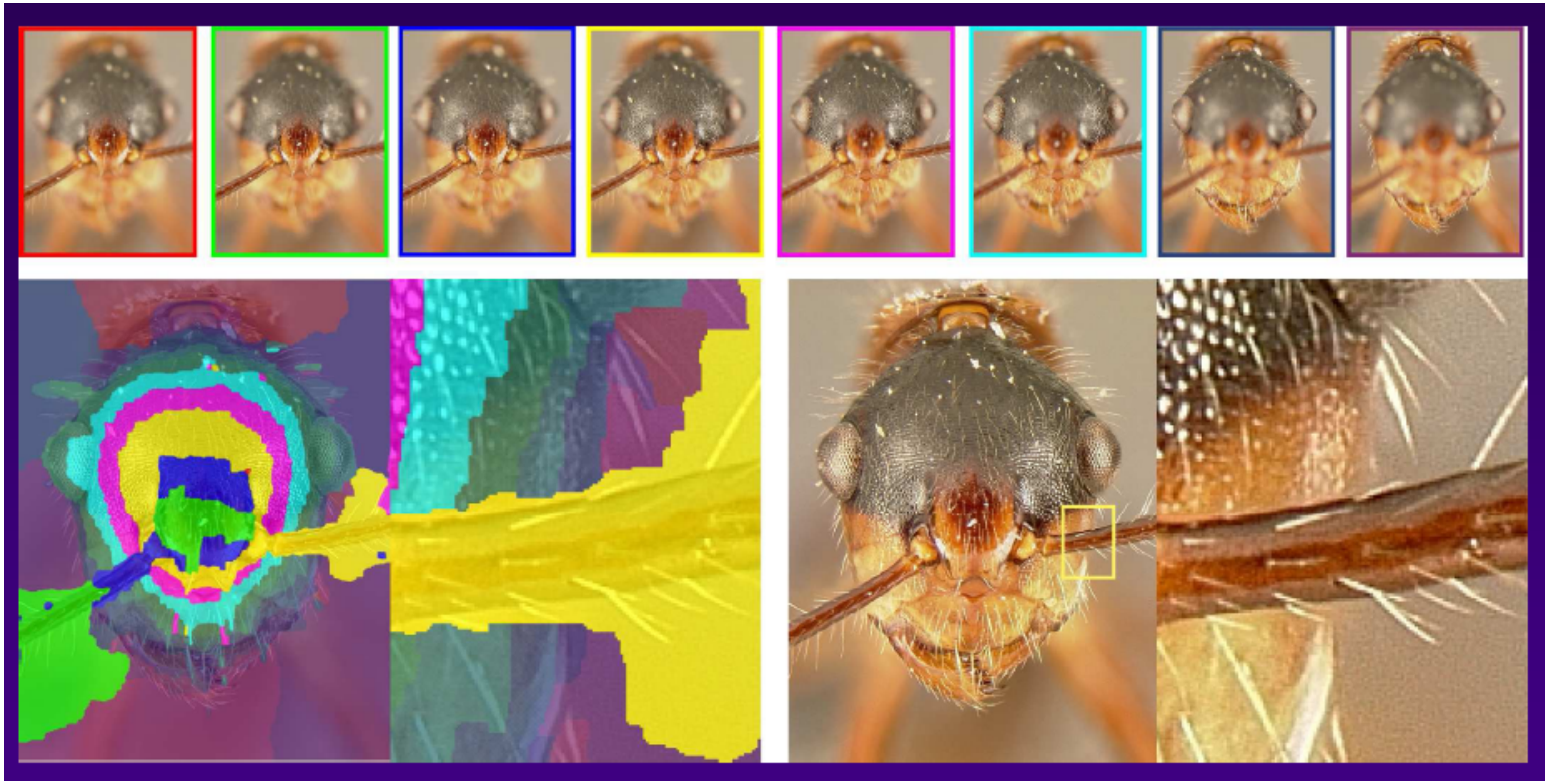
Image Composition Applications

- Stoboscopic images

[Michael Cohen]

Image Composition Applications

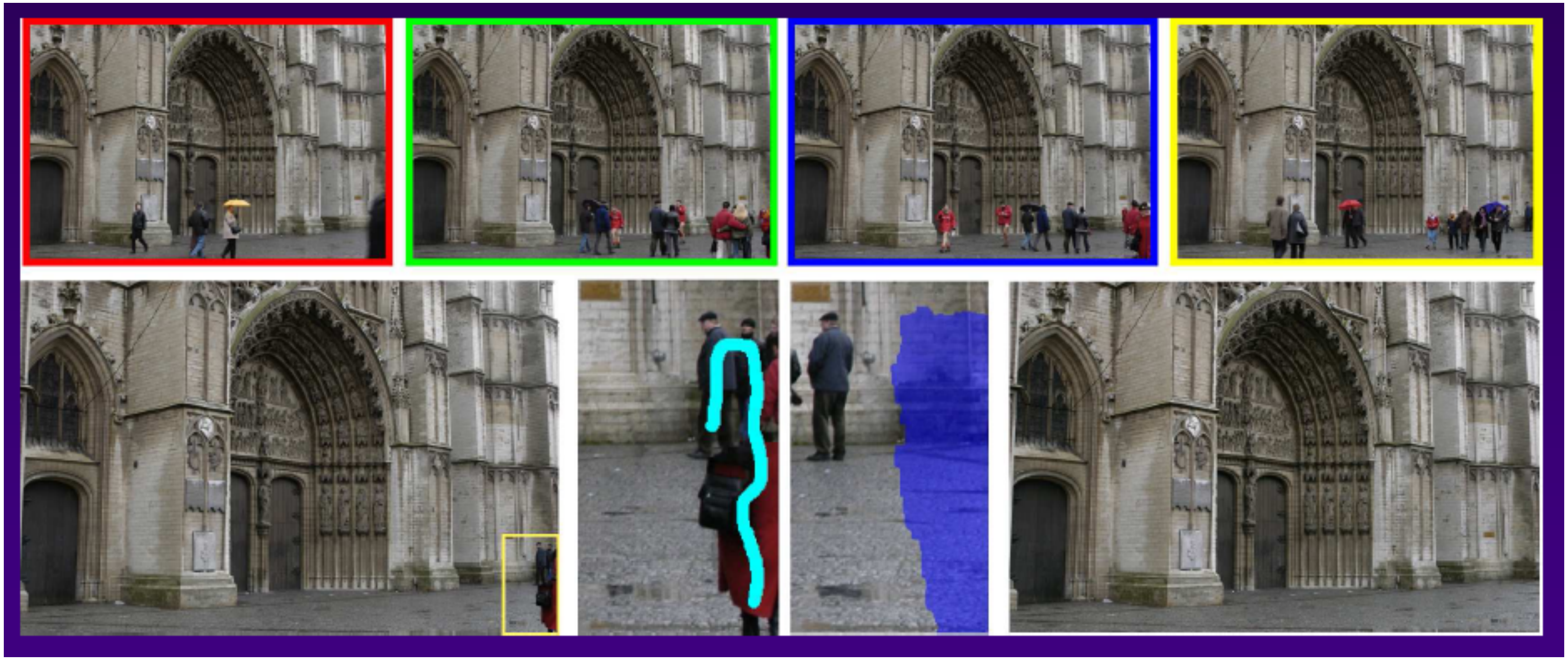
- Extended depth-of-field



[Michael Cohen]

Image Composition Applications

- Removing people



[Michael Cohen]

Scene Completion Using Millions of Photographs

James Hays and Alexei A. Efros

SIGGRAPH 2007

Slides by J. Hays and A. Efros

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Image Completion

Hays et al. SIGGRAPH 07

Image Completion

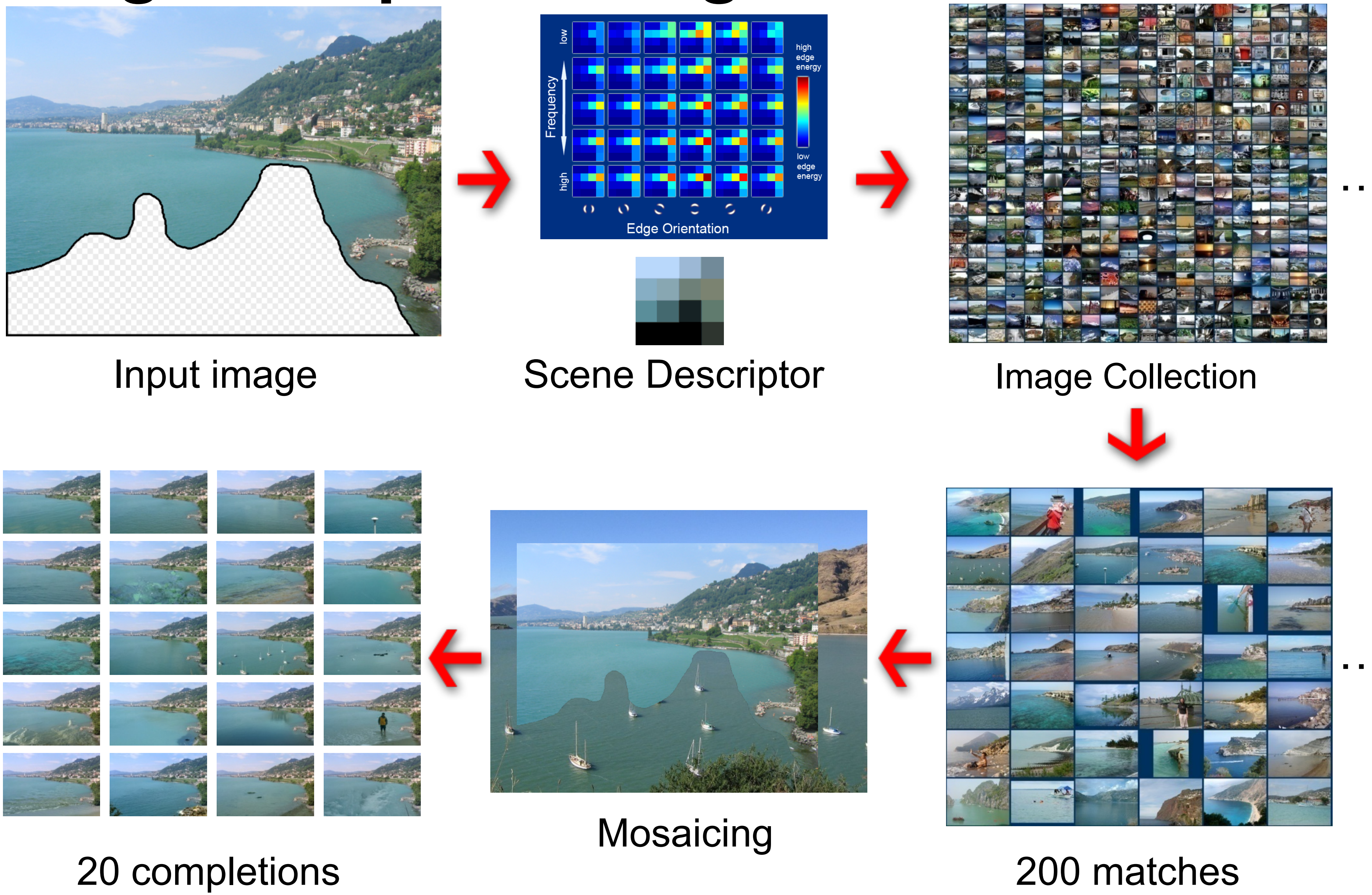
2.3 Million unique images from Flickr

Hays et al. SIGGRAPH 07

Scene Completion Result

Hays et al. SIGGRAPH 07

Image Completion Algorithm



Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

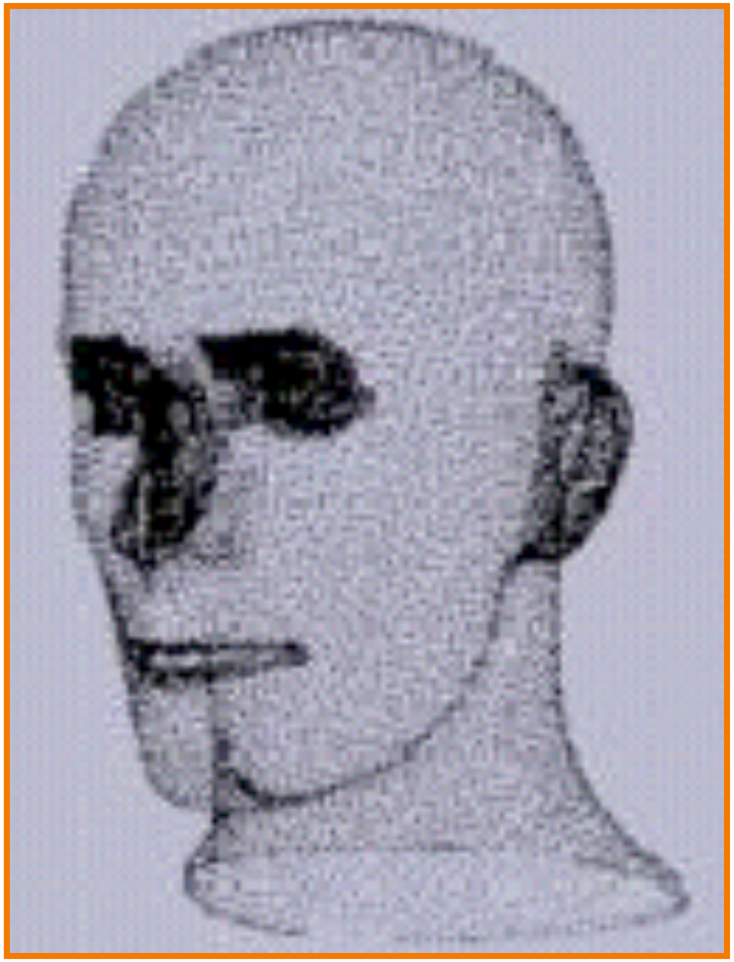
Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Summary

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

Next Time: 3D Modeling



Hoppe