Image Compositing & Morphing

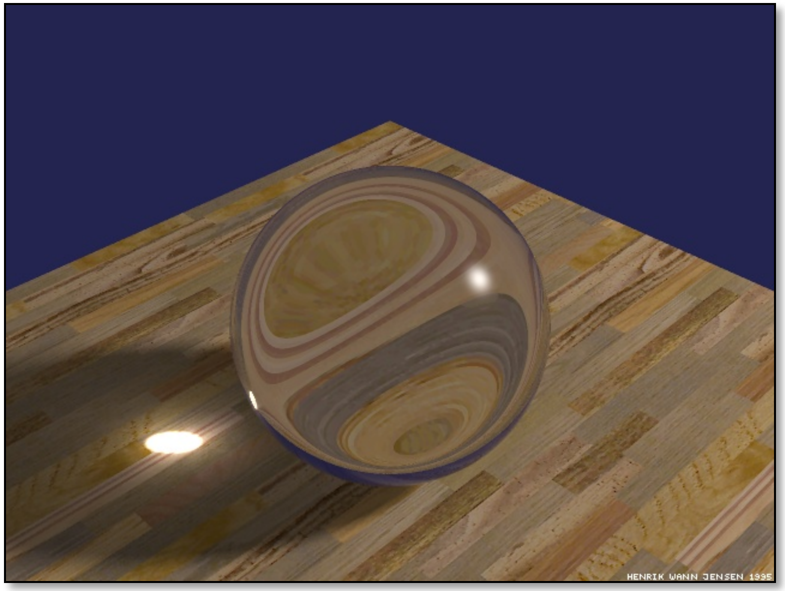
COS 426

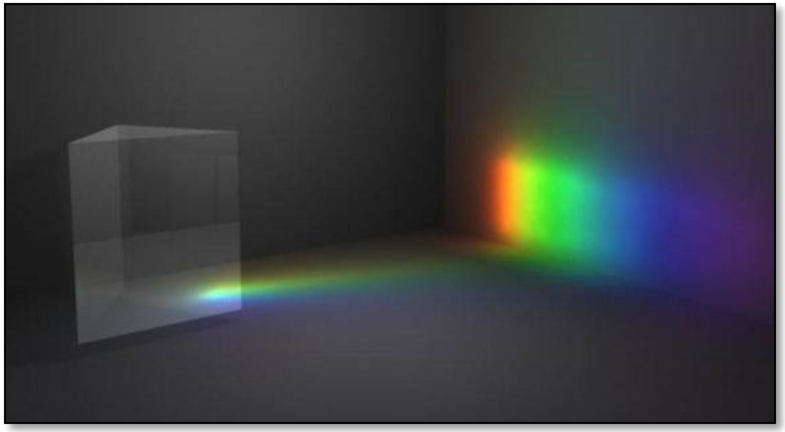
Digital Image Processing

- 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

Types of Transparency

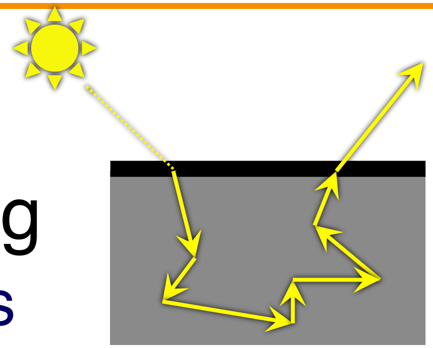
- Refraction
  - Light is bent as it goes through an object
  - Can focus light: caustics
  - Can be color-dependent: dispersion





Types of Transparency

- Refraction
- Subsurface scattering
  - Translucent materials
  - Light leaves at different position than it entered



Types of Transparency

- Refraction
- Subsurface scattering
  - Translucent materials
  - Light leaves at different position than it entered
- Today: nonrefractive transparency
  - Pixelwise composition
  - Separate image into “elements” or “layers”
  - Can generate independently
  - Composite together

Smith & Blinn`84

Example

Jurassic Park

Image Composition

- Issues:
  - Segmentation of image into regions
  - Blend into single image seamlessly

Image Composition

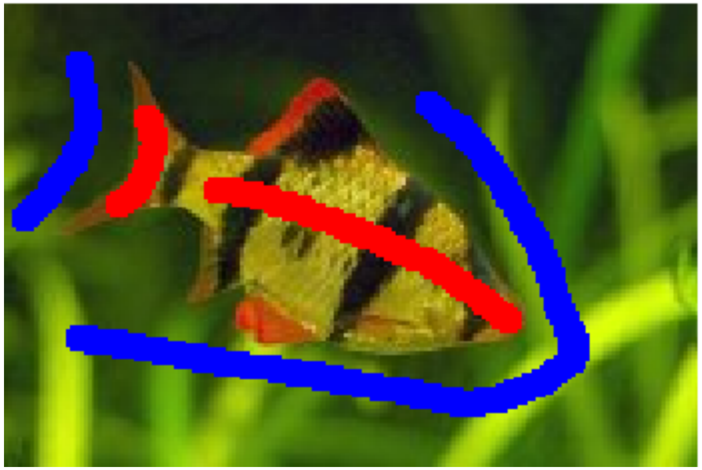
- Issues:
  - Segmentation of image into regions
  - Blend into single image seamlessly

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)



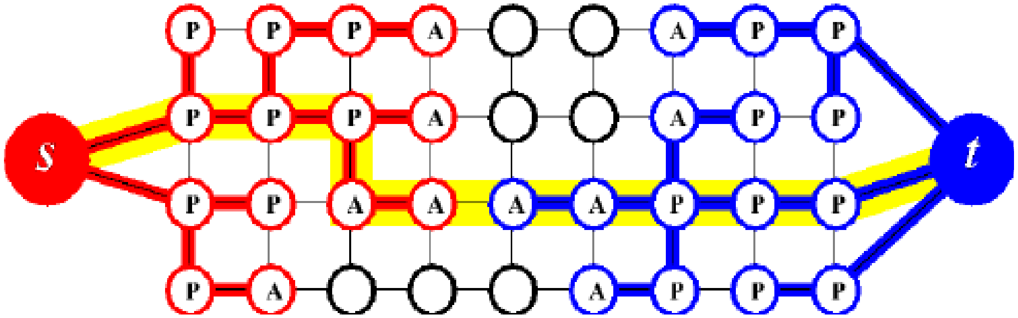


Image Segmentation

- Novel methods, e.g. flash matting

flash　　no flash　　matte　　composite

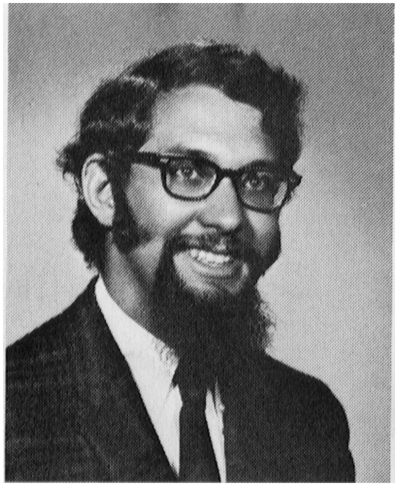
Sun et al., 2006

Image Composition

- Issues:
  - Segmentation of image into regions
  - Blend into single image seamlessly

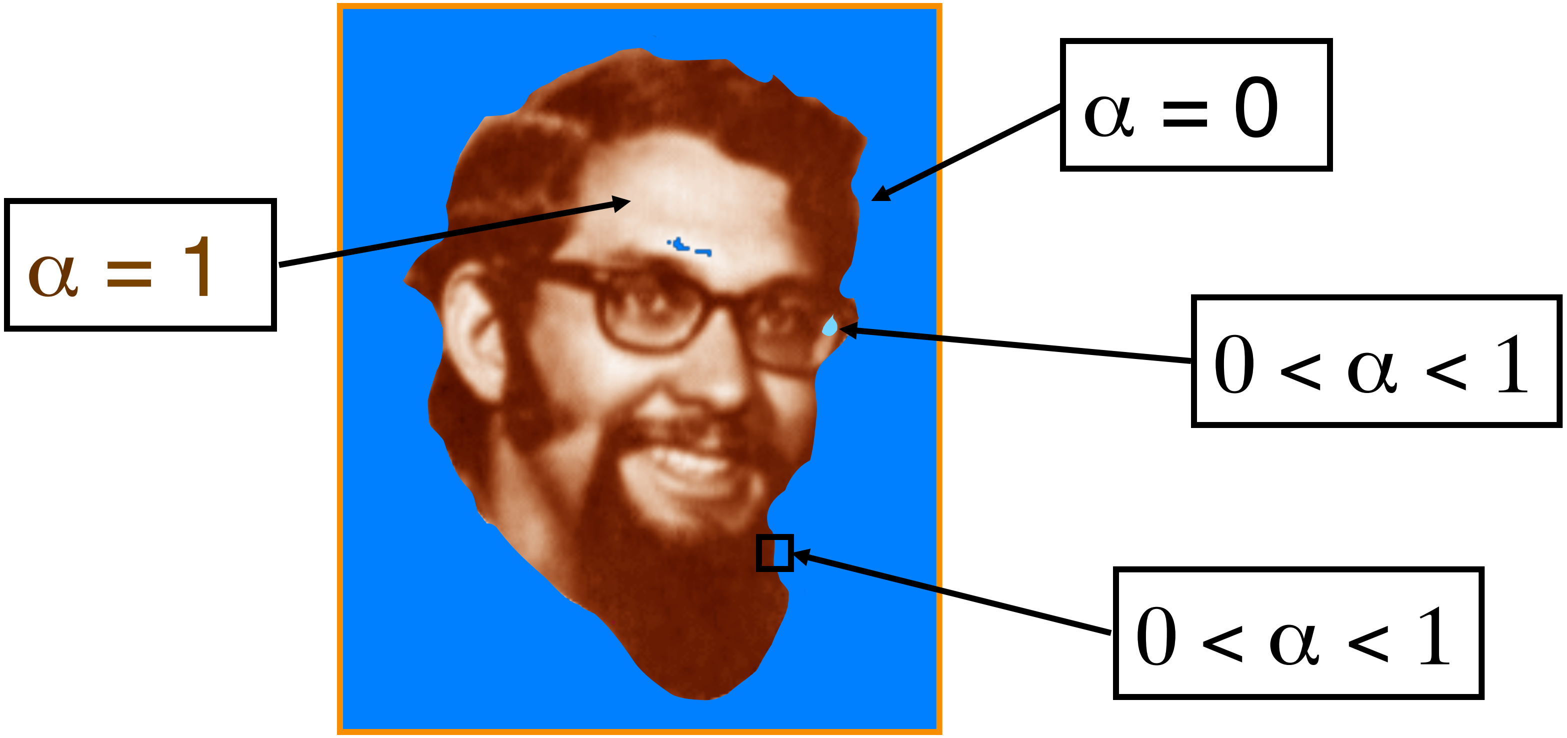
Image Blending

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



Blending with Alpha

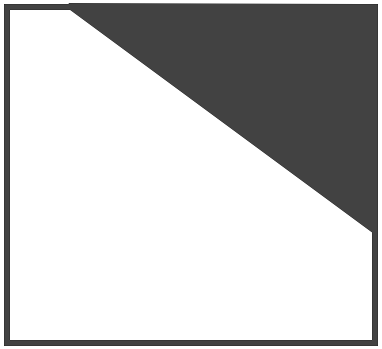
Controls the linear interpolation
of foreground and background pixels
when elements are composited.



Alpha Channel

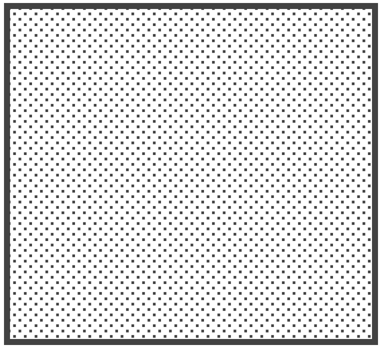
- Encodes pixel coverage information
  - $\alpha = 0$: no coverage (or transparent)
  - $\alpha = 1$: full coverage (or opaque)
  - $0 < \alpha < 1$: partial coverage (or semi-transparent)

- Example: $\alpha = 0.3$



Partial Coverage

or

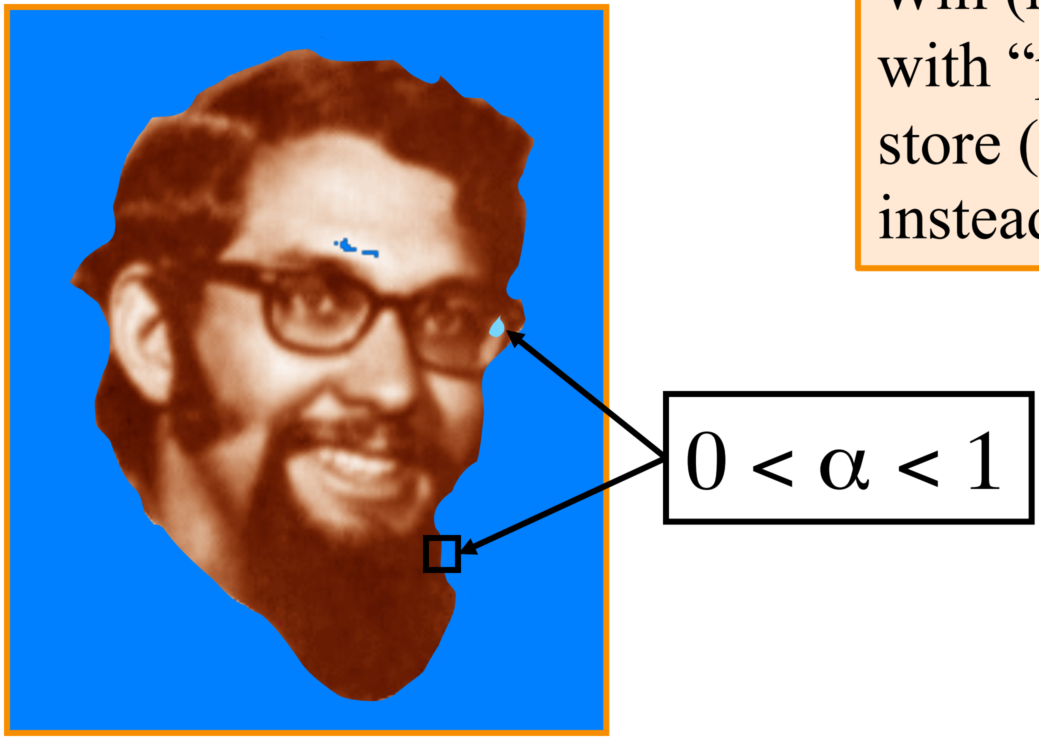


Semi-Transparent

Alpha Blending: "Over" Operator

$C = A$ over B

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

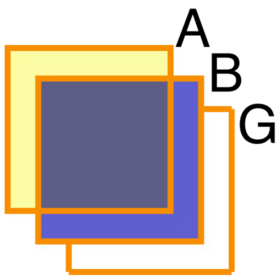


This assumes an image with "non-pre-multiplied" alpha.

Will (rarely) encounter images with "pre-multiplied" alpha: store $(\alpha R, \alpha G, \alpha B, \alpha)$ instead of (R, G, B, α)

Alpha Blending: "Over" Operator

- 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

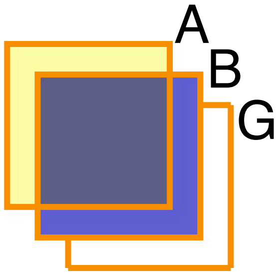
    $(1-\alpha_A)$

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

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

Alpha Blending: "Over" Operator

- Suppose we put A over B over background G



  - Final result?

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

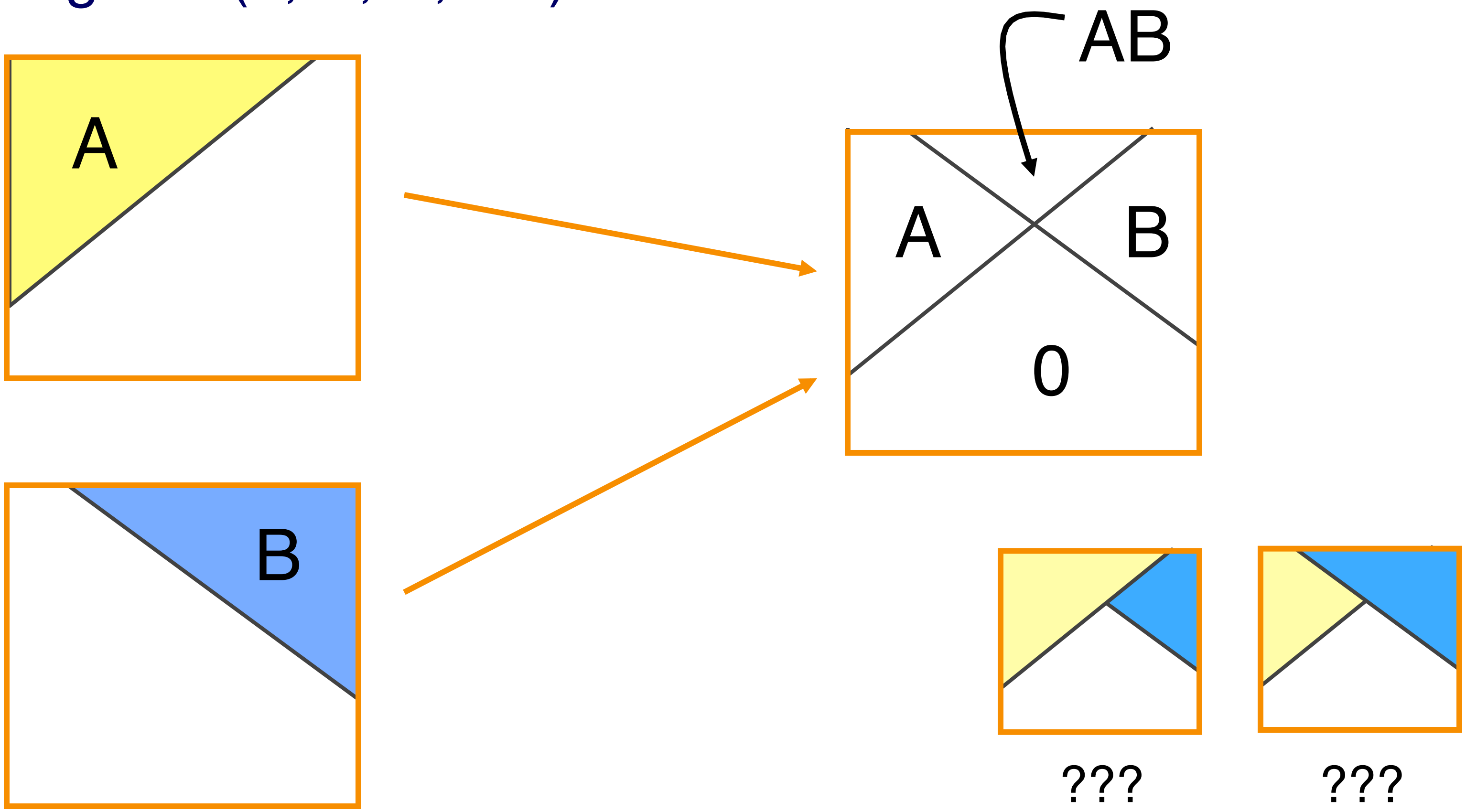
$$= \alpha_A A + (1-\alpha_A)\left[\alpha_B B + (1-\alpha_B)G\right]$$

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

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)

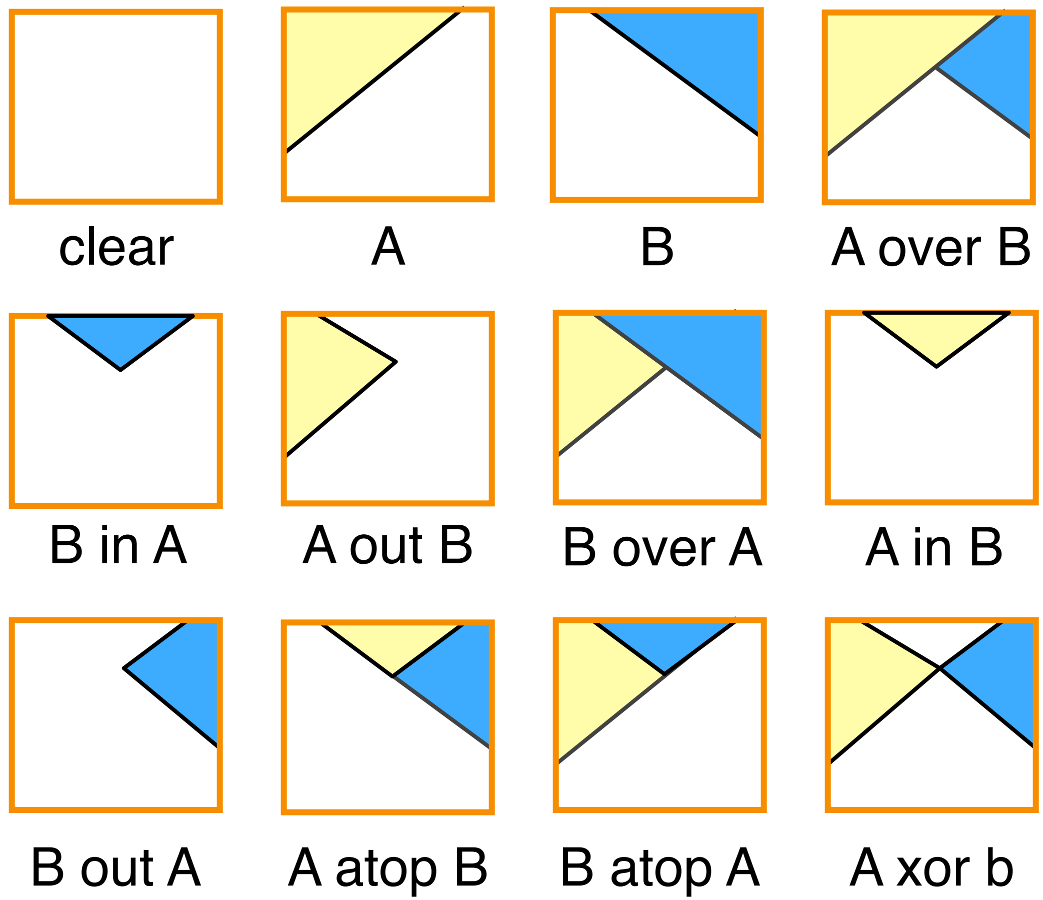


Blending with Alpha

Composition algebra – 12 combinations

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

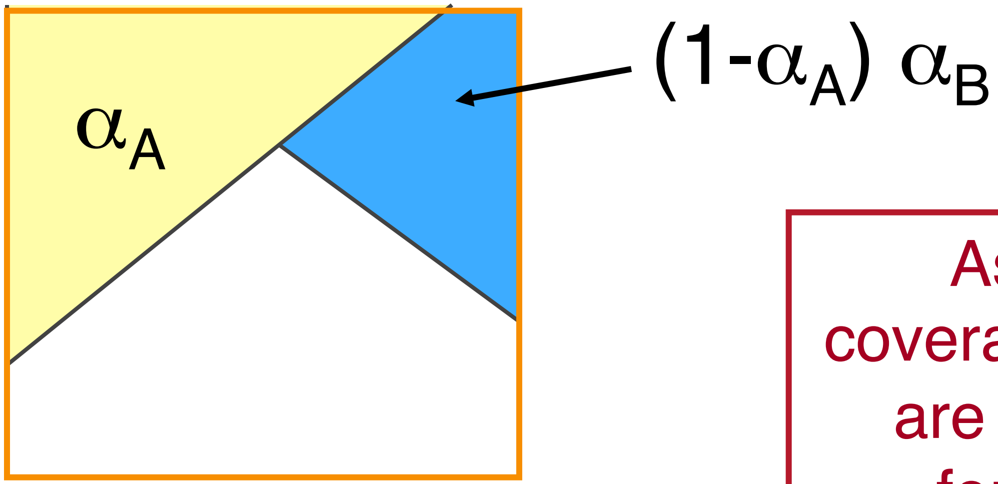
Operation	F_A	F_B
Clear	0	0
A	1	0
B	0	1
A over B	1	$1-\alpha_A$
B over A	$1-\alpha_B$	1
A in B	α_B	0
B in A	0	α_A
A out B	$1-\alpha_B$	0
B out A	0	$1-\alpha_A$
A atop B	α_B	$1-\alpha_A$
B atop A	$1-\alpha_B$	α_A
A xor B	$1-\alpha_B$	$1-\alpha_A$



Porter & Duff `84

Blending with Alpha

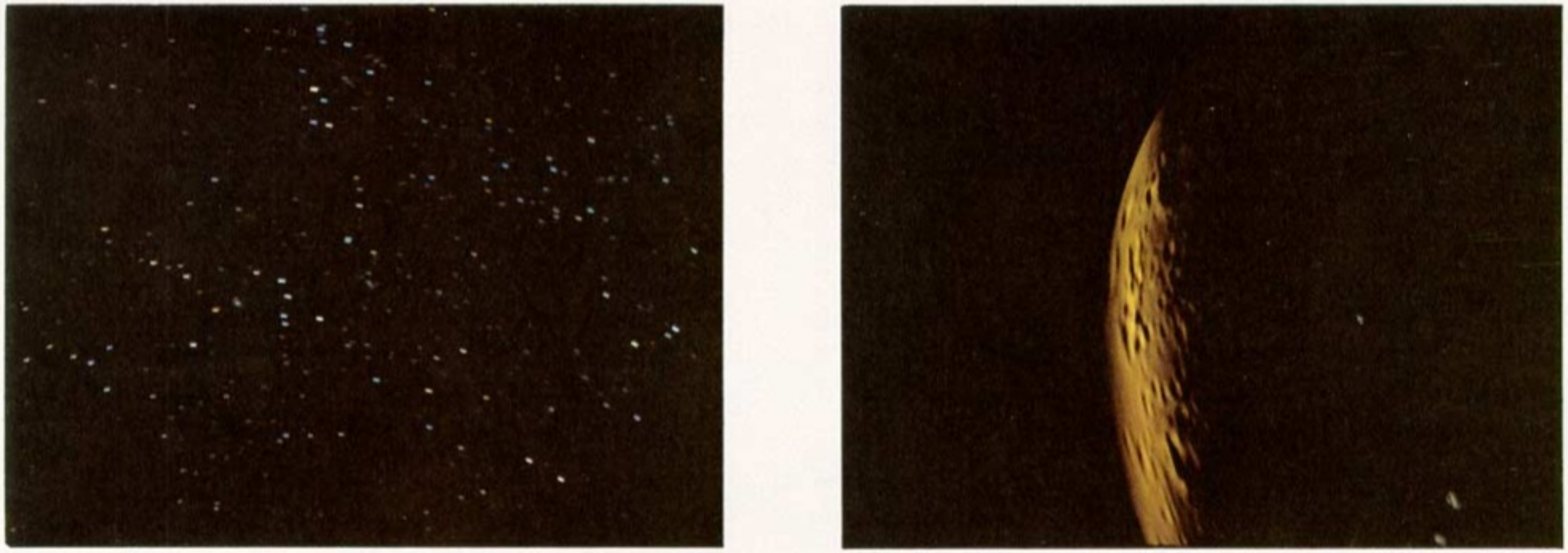
- Example: C = A Over B
  - $C' = \alpha_A A + (1-\alpha_A)\,\alpha_B B$
  - $\alpha = \alpha_A + (1-\alpha_A)\,\alpha_B$



A over B

Assumption:
coverages of A and B
are uncorrelated
for each pixel

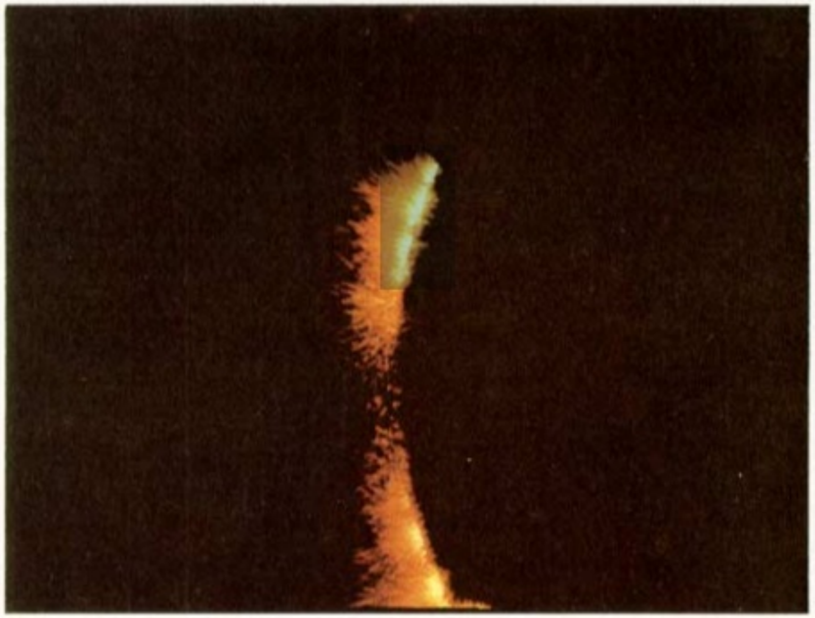
Image Composition Example



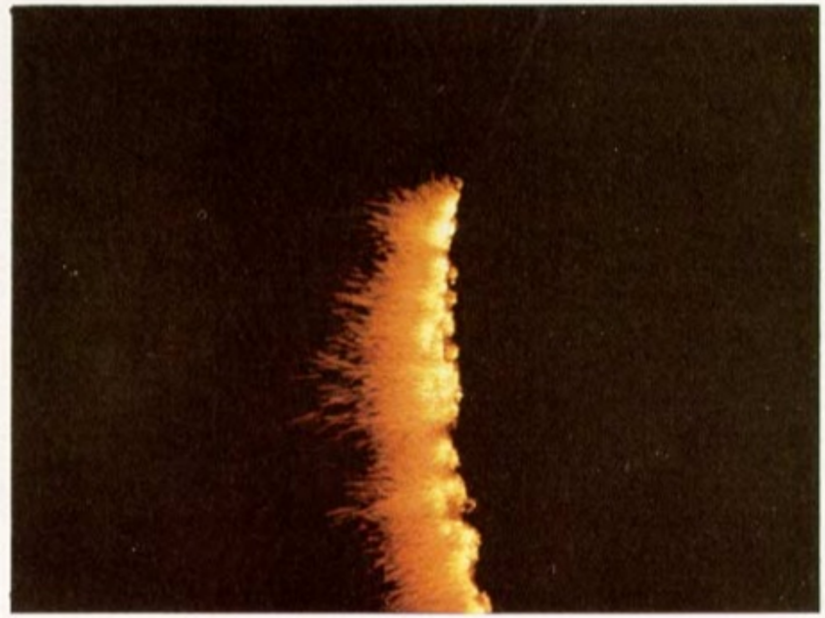
Stars   Planet

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

Image Composition Example



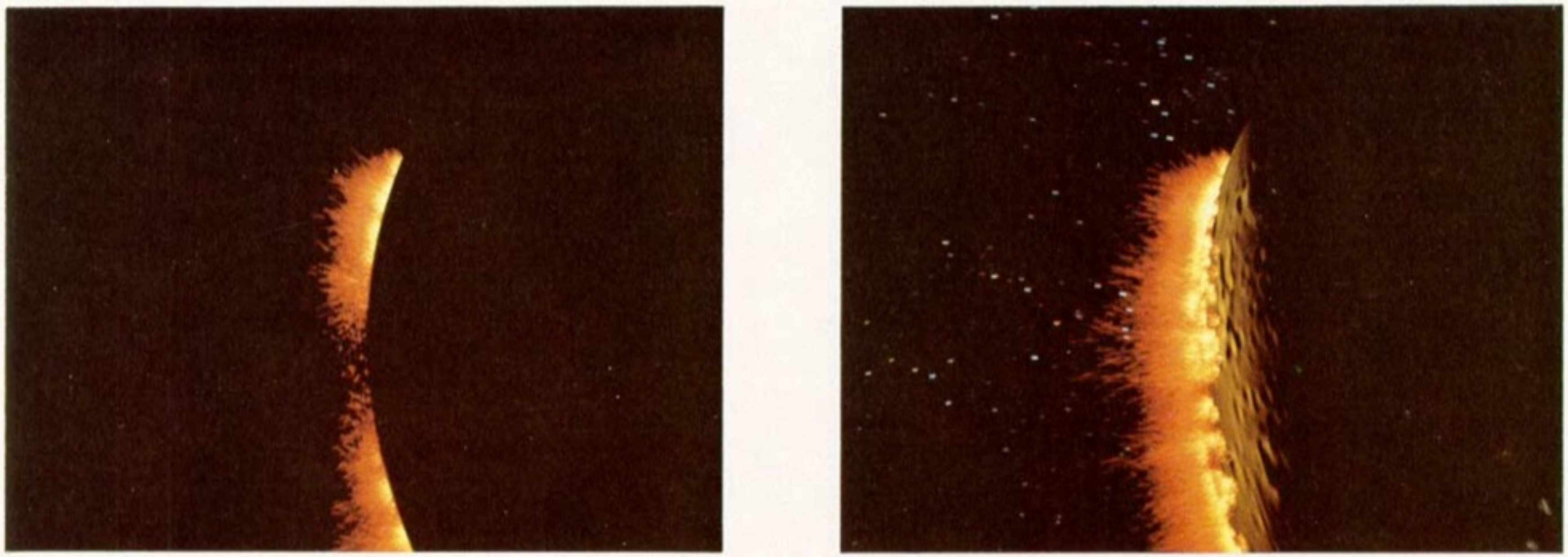
BFire



FFire

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

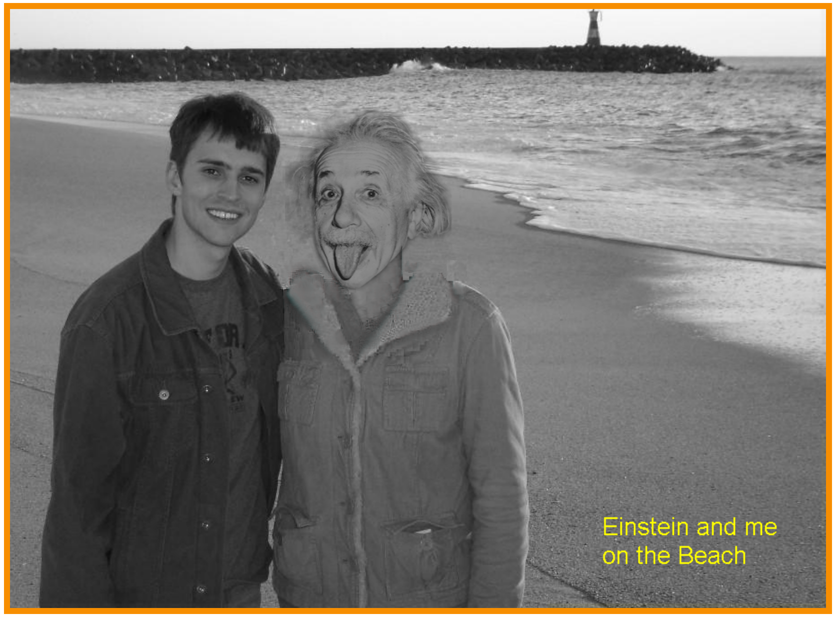
Image Composition Example



BFire **out** Planet   Composite

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

COS426 Examples



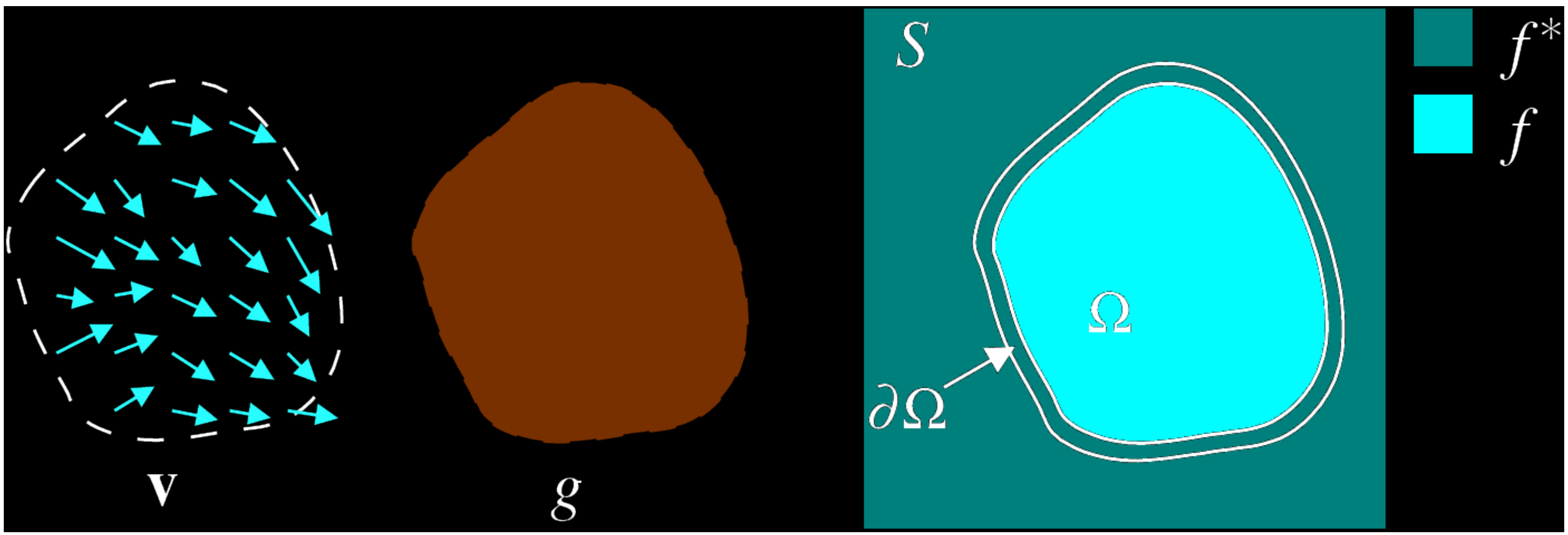
Darin Sleiter

Kenrick Kin

Poisson Image Blending

Beyond simple compositing

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

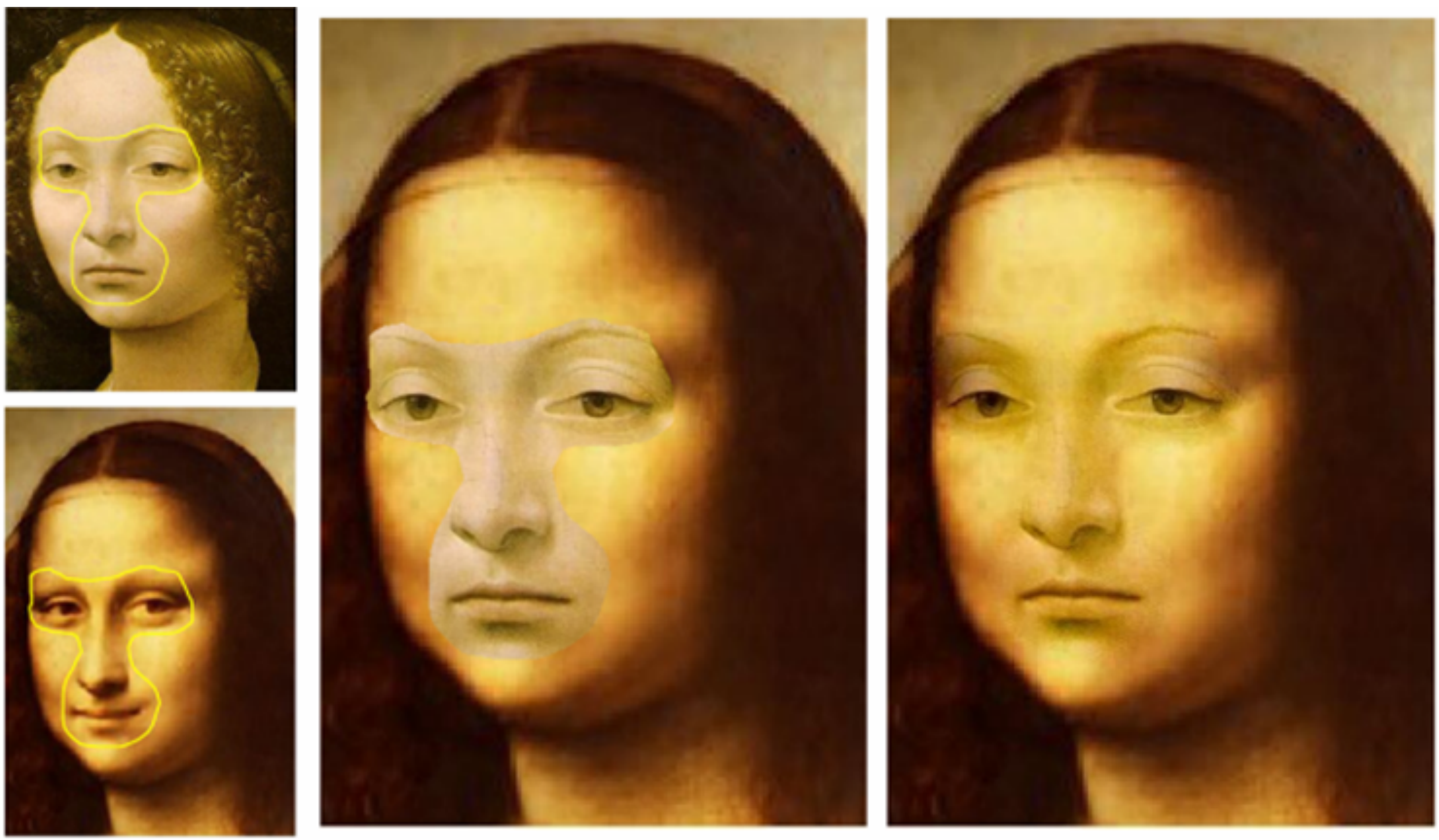


$$\begin{cases} \nabla^2 f = \nabla \cdot \mathbf{v} \\ f|_{\partial\Omega} = f^*|_{\partial\Omega} \end{cases}$$

Poisson Image Blending

sources destinations cloning seamless cloning

Poisson Image Blending



source/destination   cloning   seamless cloning

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

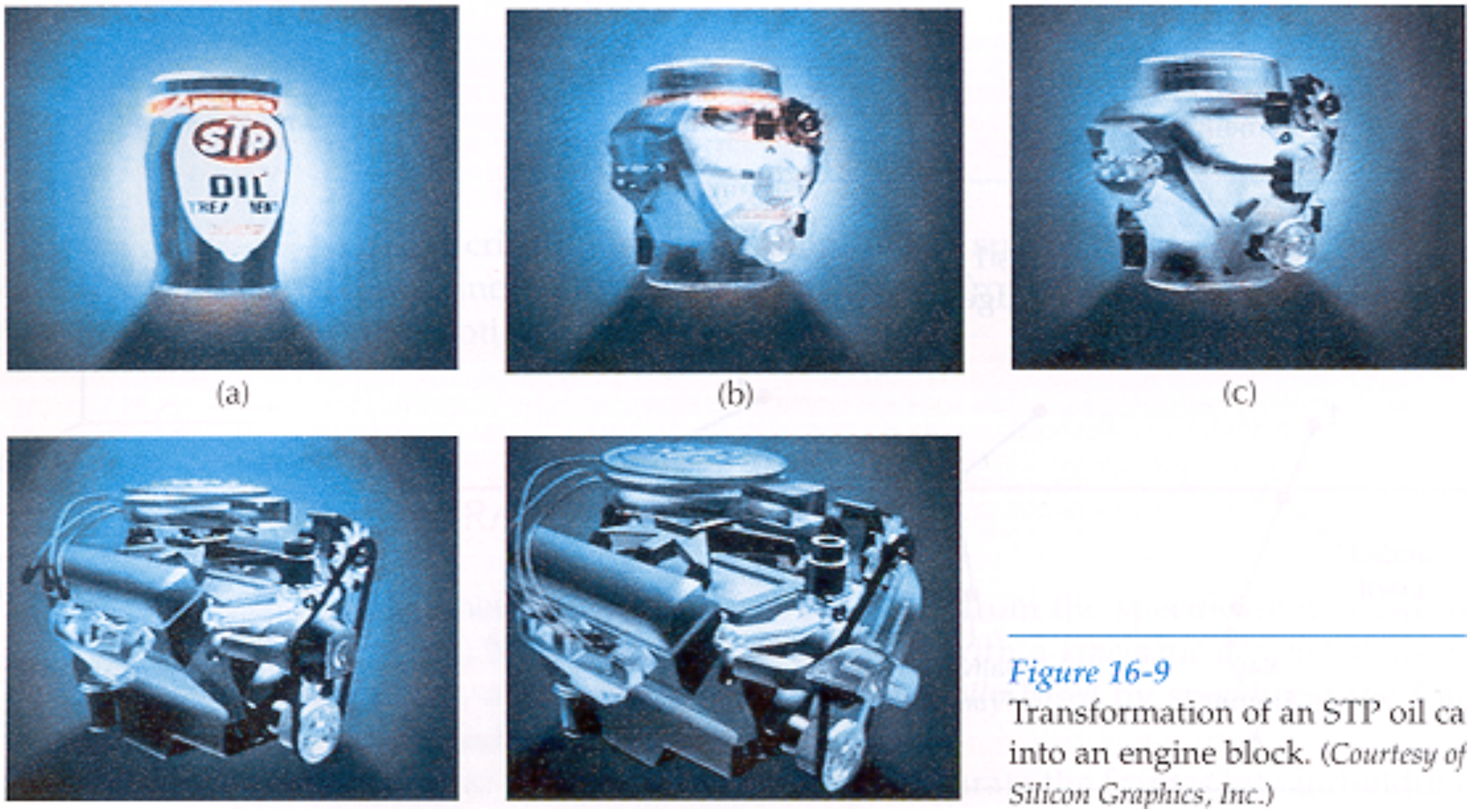


Figure 16-9
Transformation of an STP oil ca into an engine block. (*Courtesy of Silicon Graphics, Inc.*)

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

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

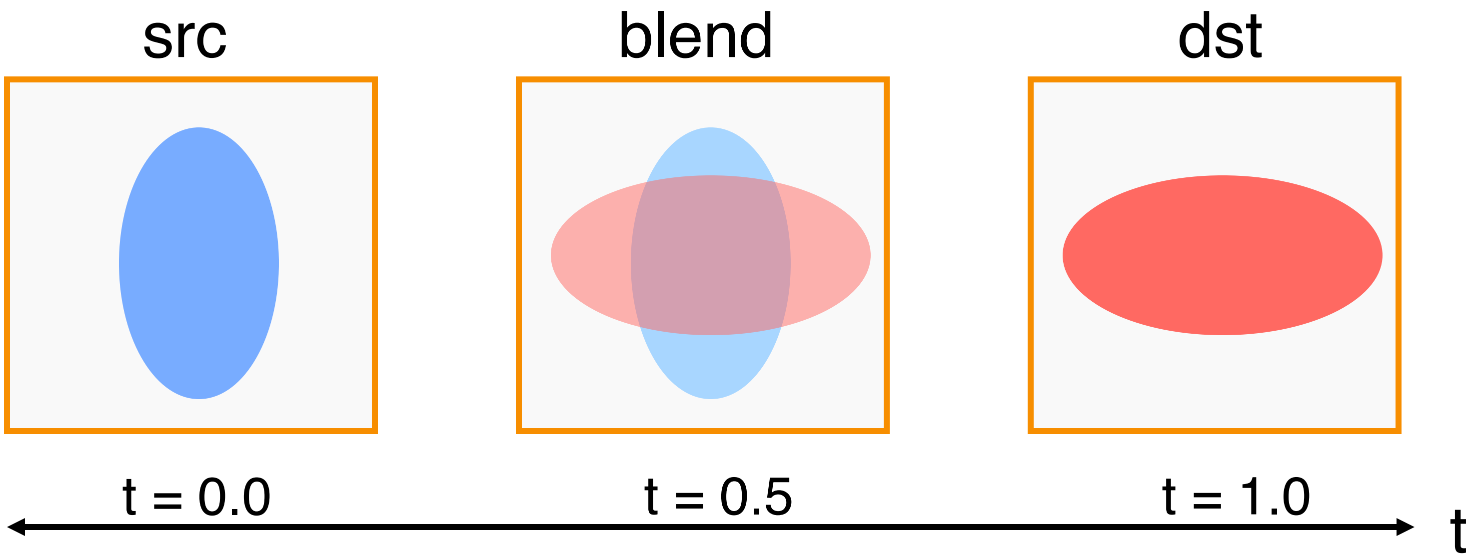
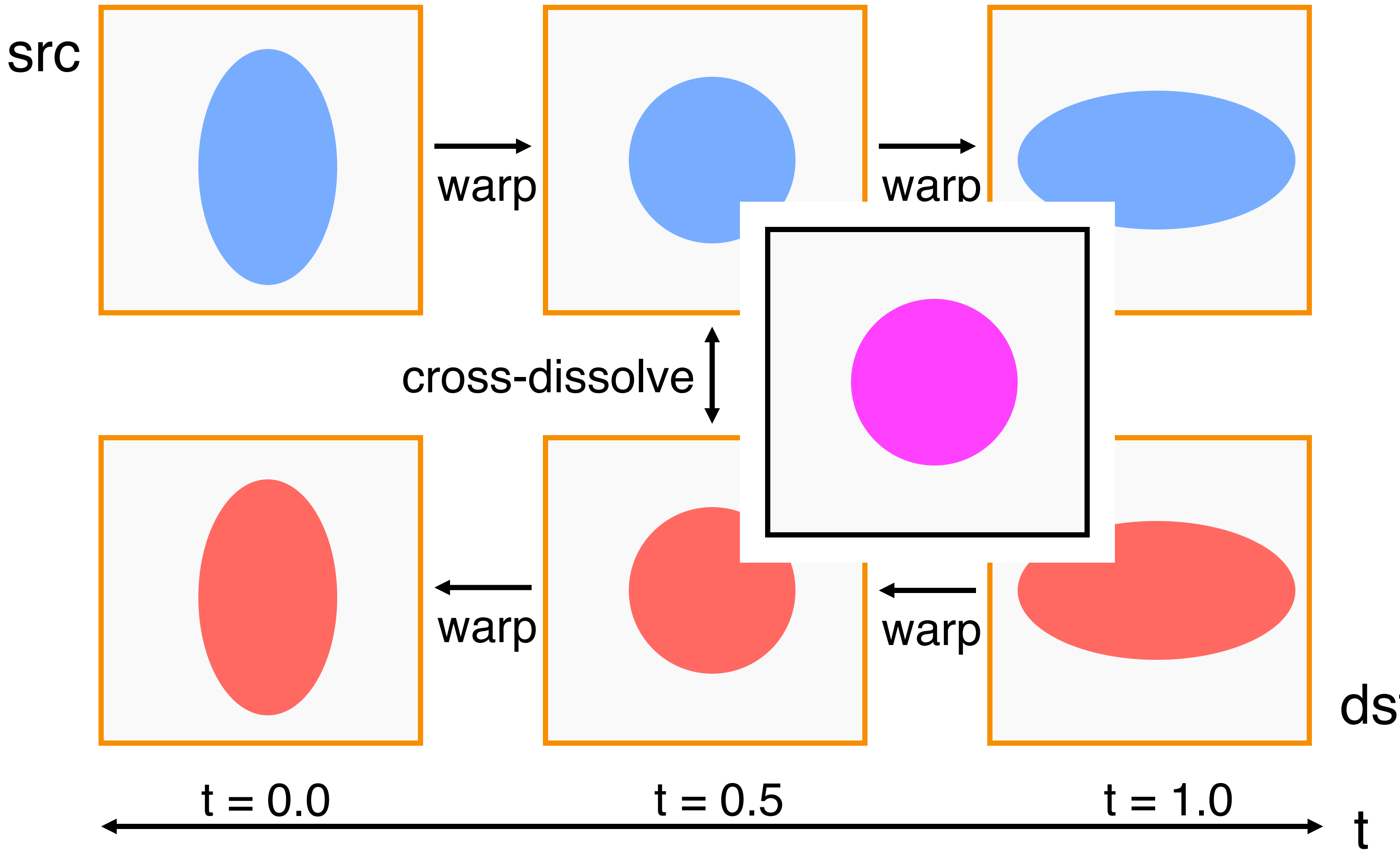


Image Morphing

- Combines warping and cross-dissolving



Beier & Neeley Example

$Image_0$

$Warp_0$

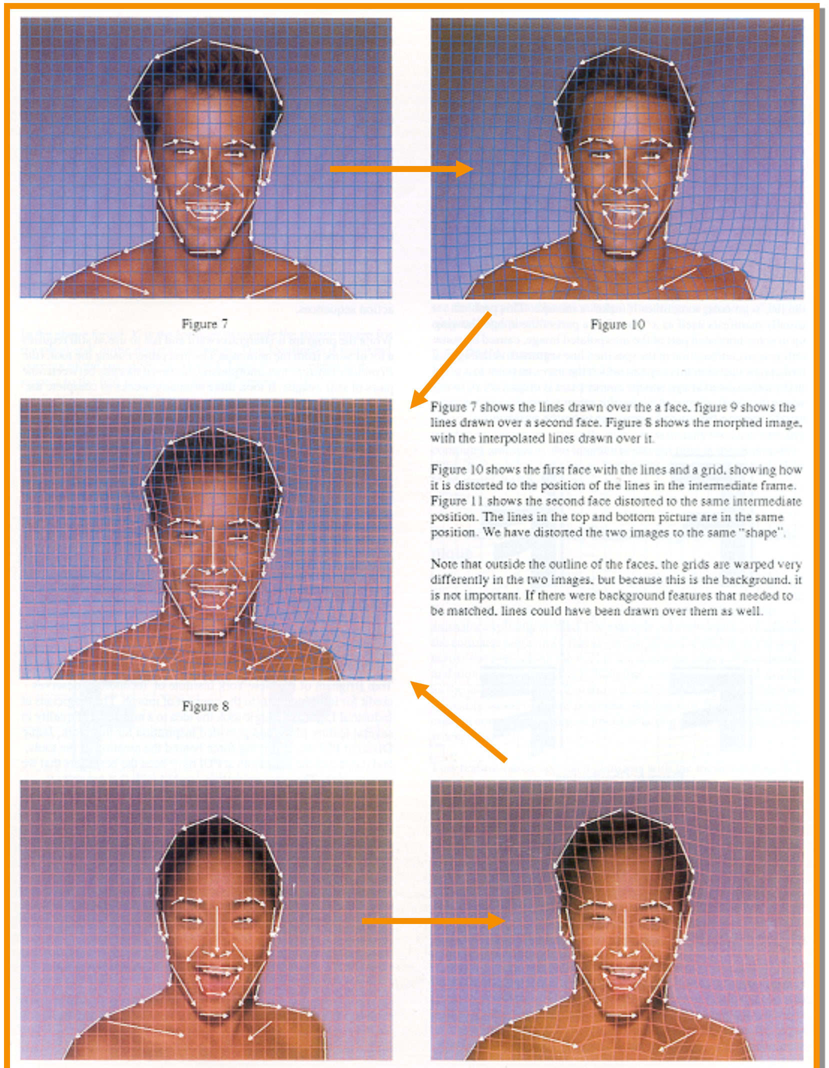


Figure 7

Figure 10

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.

Figure 8

Result

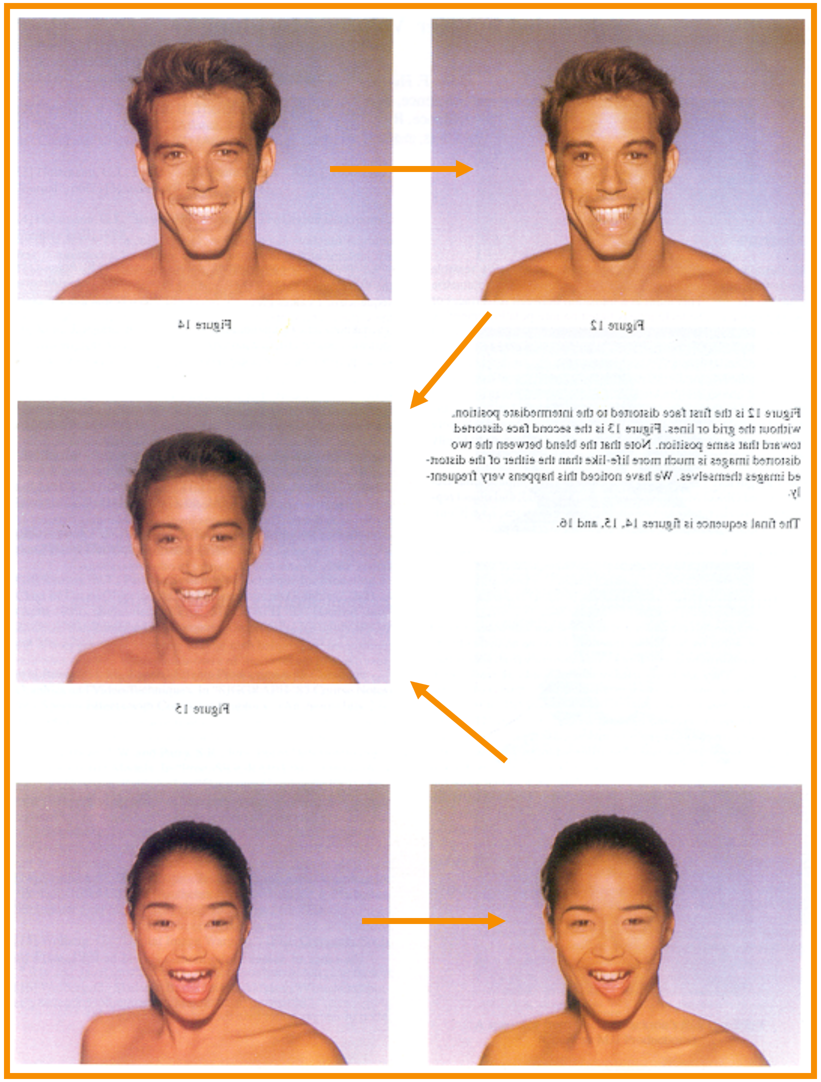
$Image_1$

$Warp_1$

Beier & Neeley Example

Image_0

Warp_0



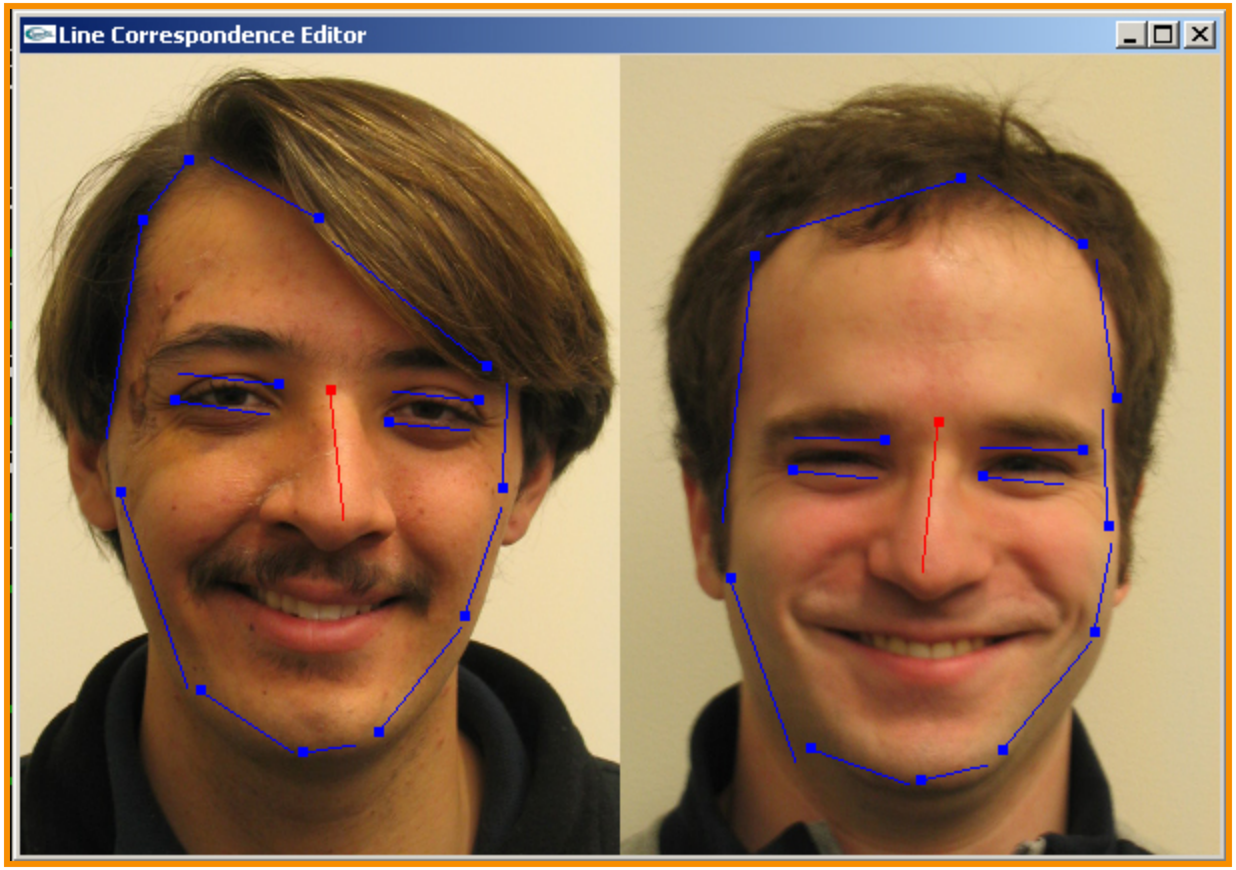
Result

Image_1

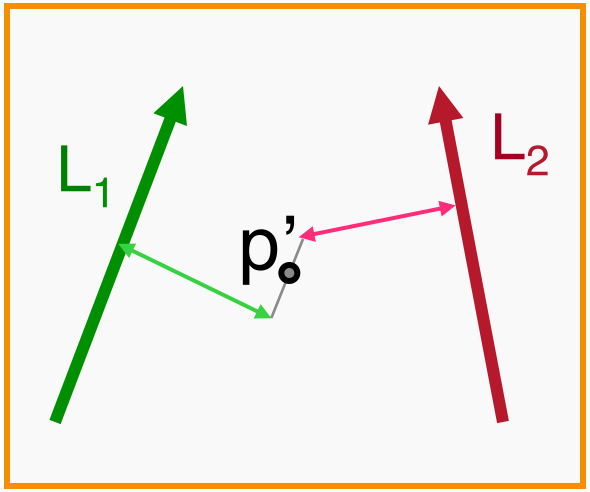
Warp_1

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 = WarpImage(Image0, L0, L)
        Warp1 = WarpImage(Image1, L1, L)
        foreach pixel p in FinalImage do
            Result(p) = (1-t) Warp0 + t Warp1

    end
end
```

COS426 Example

Amy Ousterhout

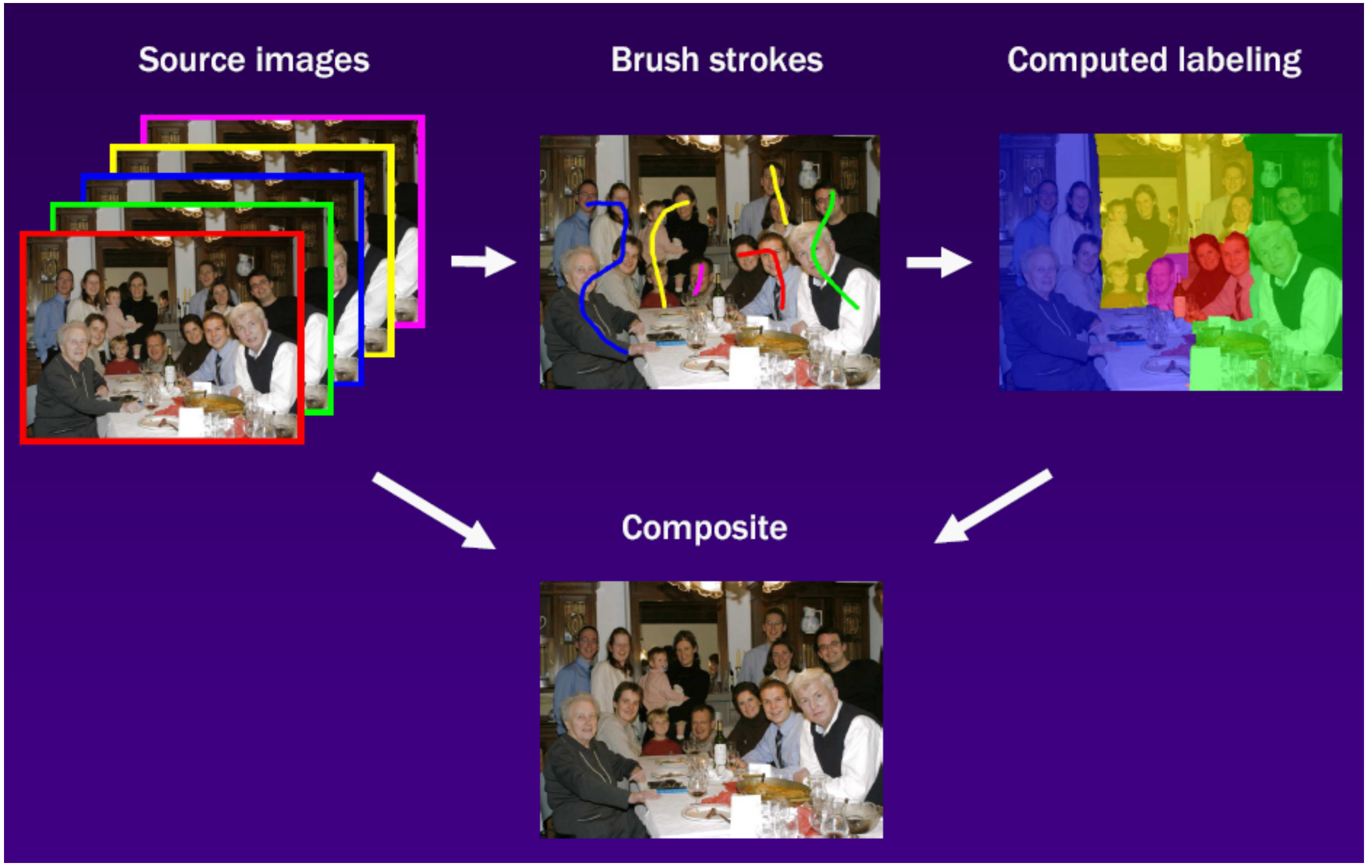
Image Composition Applications

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

[Michael Cohen]

Image Composition Applications

- Photo montage



[Michael Cohen]

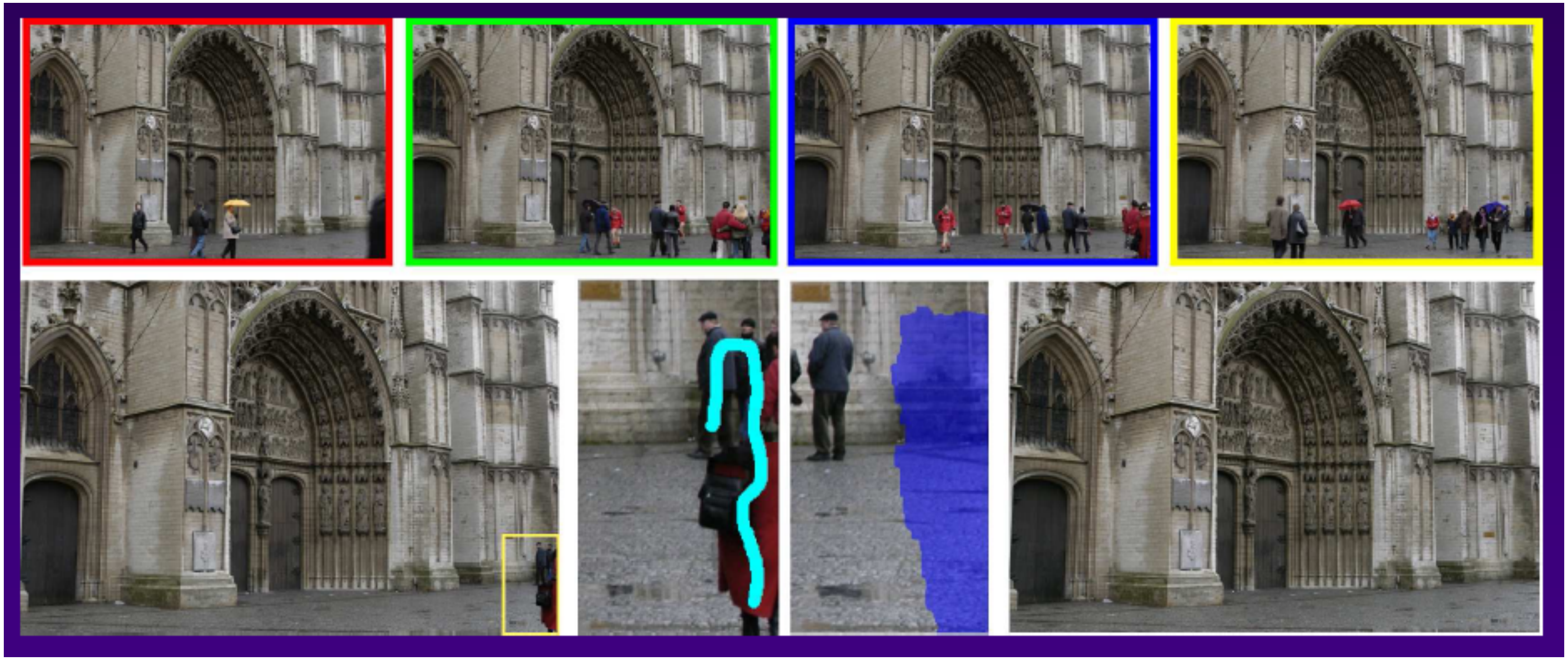
Image Composition Applications

- Photo montage

[Michael Cohen]

Image Composition Applications

- Removing people



[Michael Cohen]

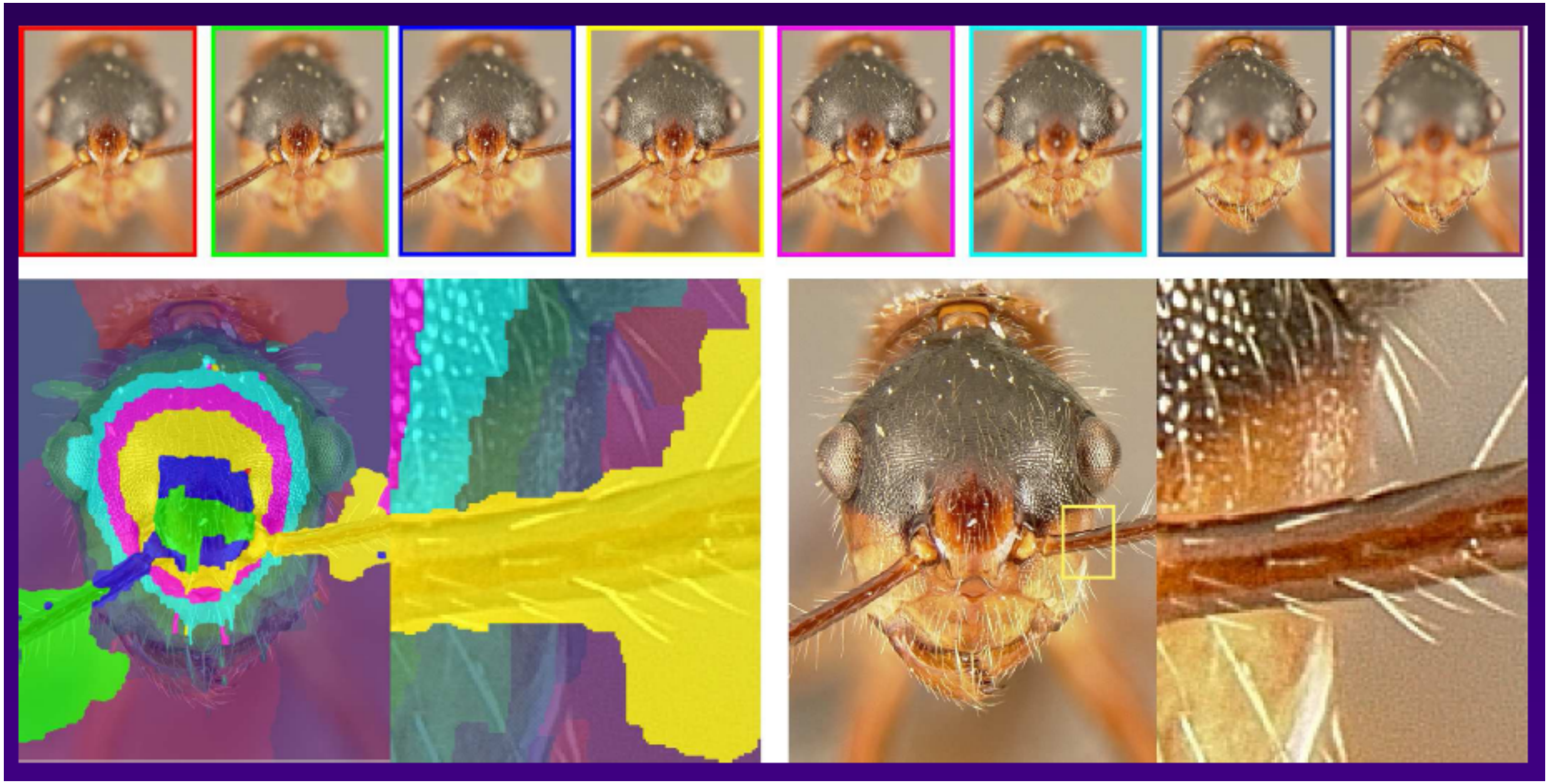
Image Composition Applications

- Stoboscopic images

[Michael Cohen]

Image Composition Applications

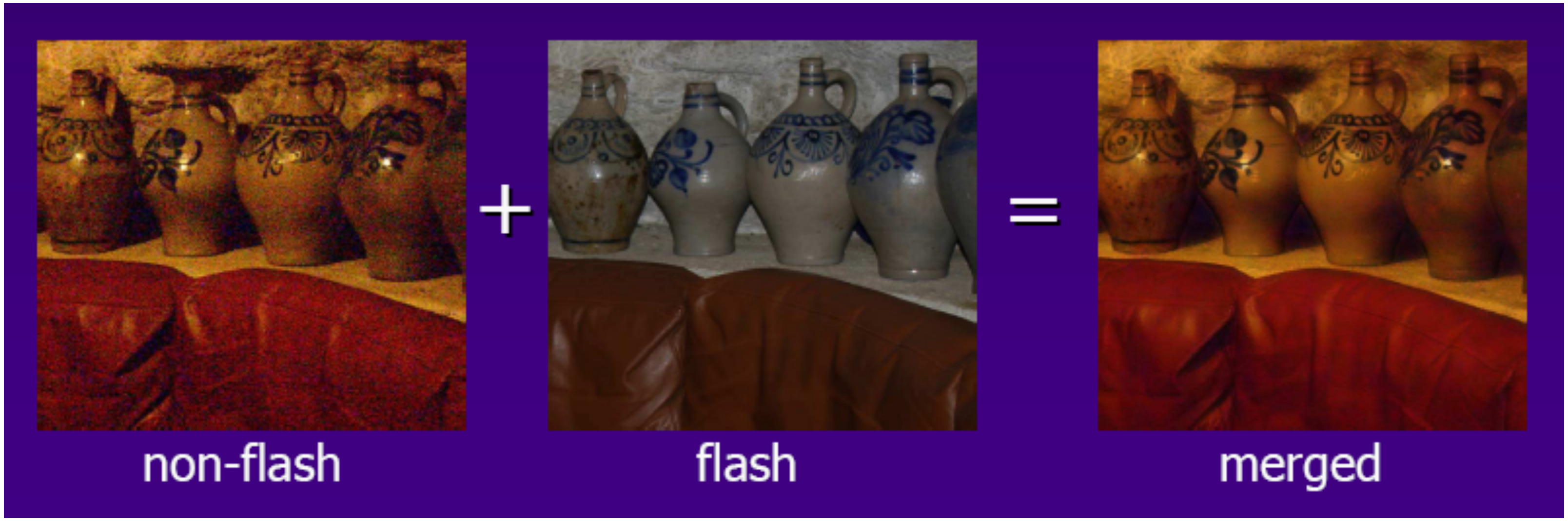
- Extended depth-of-field



[Michael Cohen]

Image Composition Applications

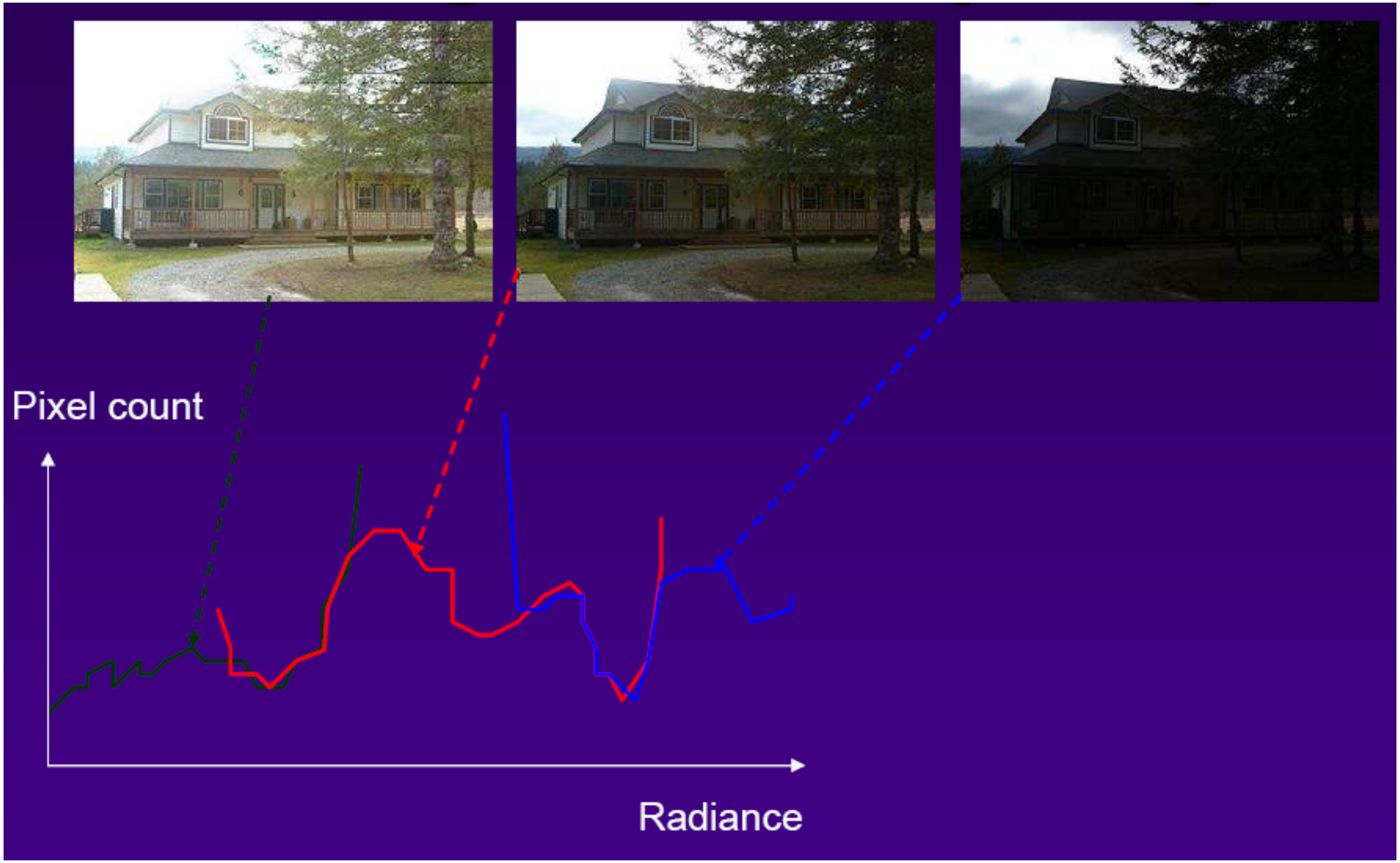
- Flash / No flash



[Michael Cohen]

Image Composition Applications

- High dynamic range images



[Michael Cohen]

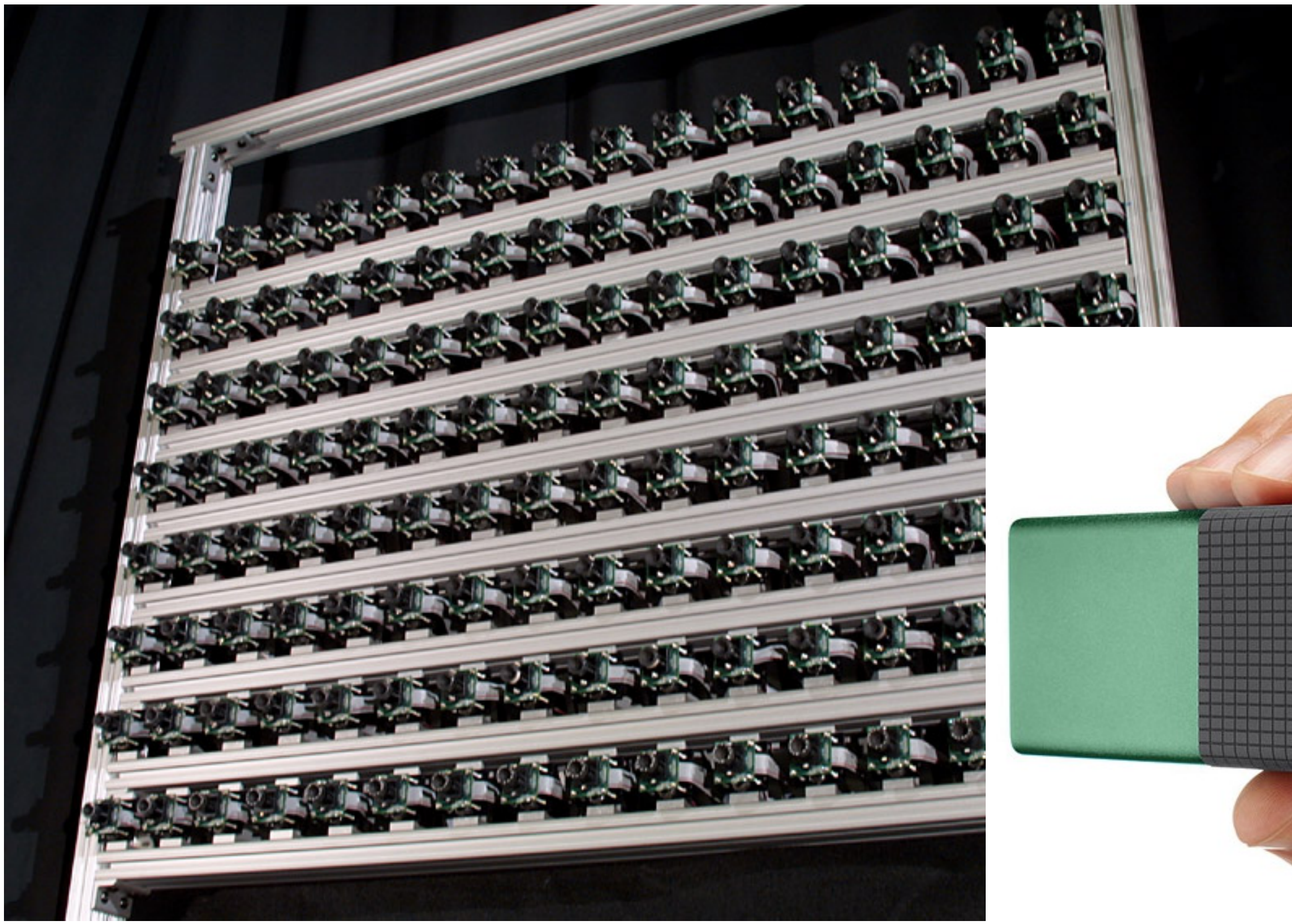
Image Composition Applications

- High dynamic range images

[Michael Cohen]

Image Composition Applications

- Multi-camera array



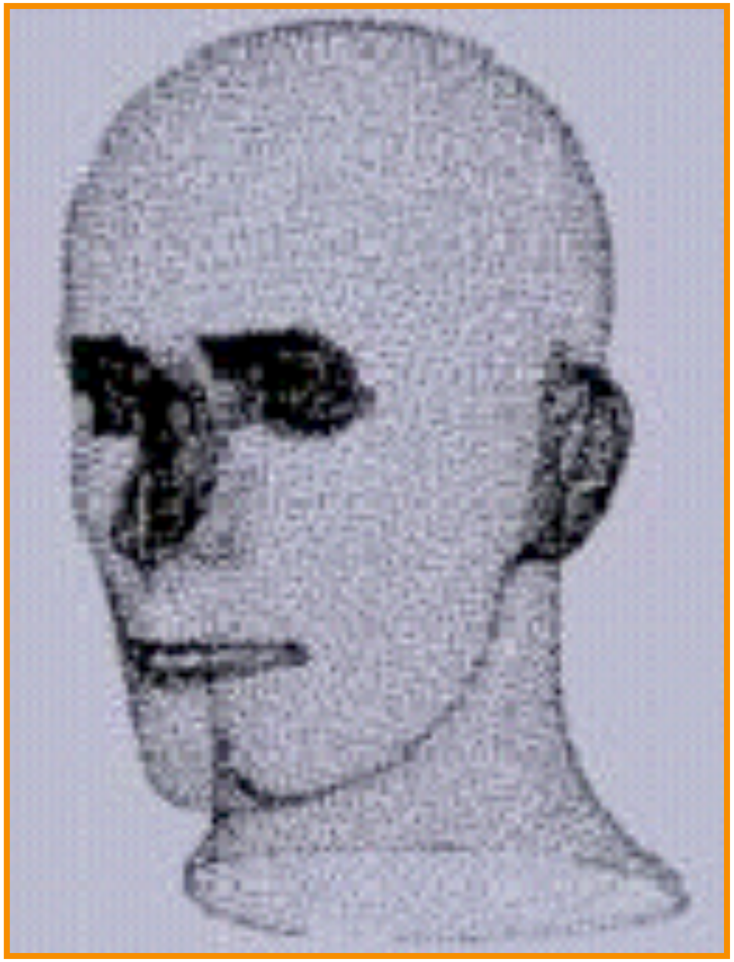
[Marc Levoy]

lytro.com

Summary

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

Next Time: 3D Modeling



Hoppe