Imaging Models

Abstract images and imaging models

Image composition

Mattes and the alpha channel

Fragments

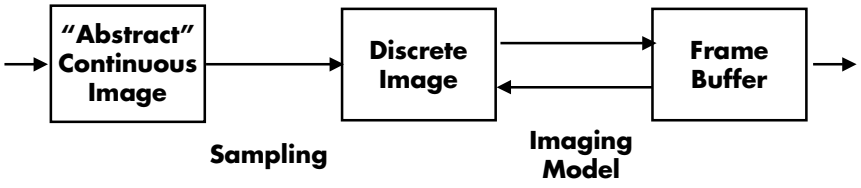
Compositing operators

Painting

Matte extraction

Copyright © Pat Hanrahan

Imaging Model



Imaging operations

- **Functional**
  - **user: erase, draw, reverse, highlight**
- **Bitmaps**
  - **bitblt: clr, mov, set, xor, and, or**
- **Arithmetic**
  - **blend: cross-dissolve, accumulation, compositing**

Copyright © Pat Hanrahan

Image Composition

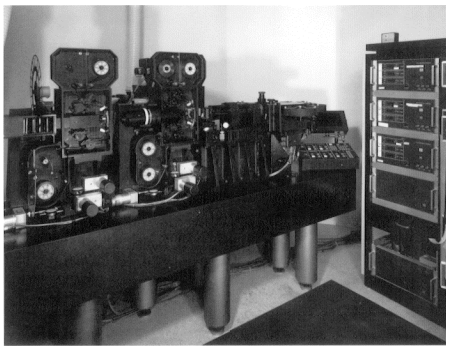
Defn: Combine *foreground* element with *background*

Examples:

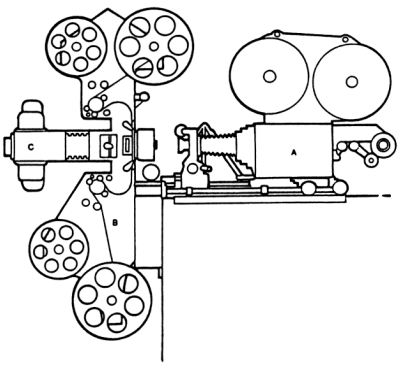
- **Film: optical compositing and blue screen matting**
- **Video: video-keying, chroma-keying**
- **Graphics arts: masks, friskets, stencils**
- **Animation: cels**
- **Computer graphics:**
  - **emulate film, video, graphics arts, animation techniques**
  - **plus: painting, sprites (cursor), multi-pass rendering, ...**

Copyright © Pat Hanrahan

Optical Printing



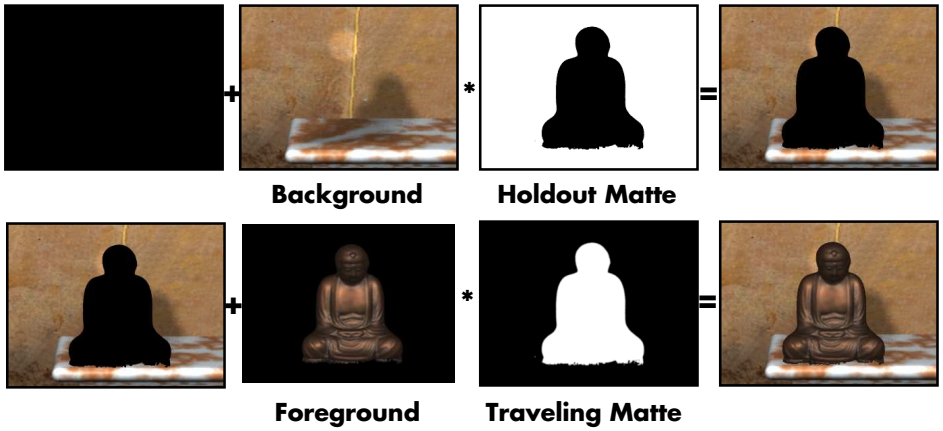
From: "Industrial Light and Magic," Thomas Smith (p. 181)



From: "Special Optical Effects," Zoran Perisic

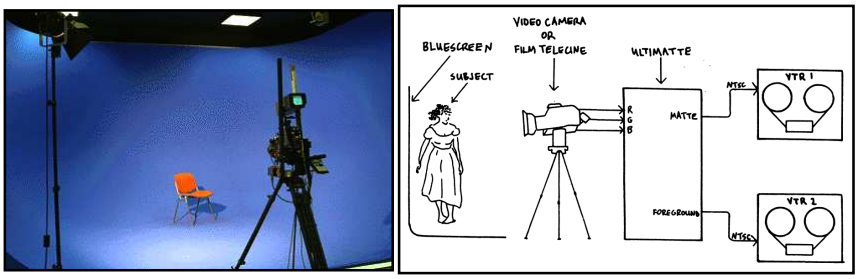
Copyright © Pat Hanrahan

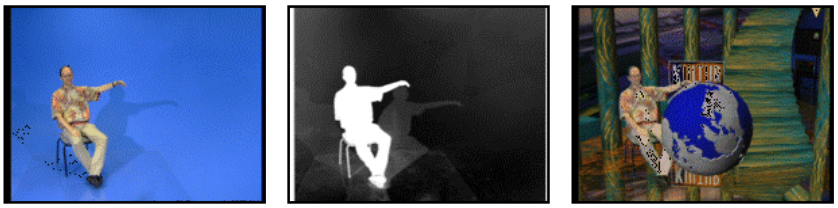
Composing Two Elements



Copyright © Pat Hanrahan

Blue Screen





Copyright © Pat Hanrahan

"Pulling a Matte" - Matte Creation

From digitized images

- Blue-screen matting (Petro Vlahos)
  - Separate blue background from foreground image
- Video or chroma-keying
  - Range of luminances marked transparent
- Image processing
  - Set of colors marked transparent, region growing ...

From computer generated images

- Coverage
- Transparency

Copyright © Pat Hanrahan

Mattes & Masks: Alpha Channel

A alpha channel is an additional image that defines:

- The presence or absence of imagery
  - Geometric coverage: soft-edge
- The transparency or opacity of an image
- Or both coverage and transparency

Alpha channels may be

- Masks: all or none, binary
- Mattes: 0 to 1, n-ary

Combination of color and opacity

- Conceptually tied together -> like a digital bipack
- Integral alpha channel - RGBA

Copyright © Pat Hanrahan

Pixel Fragment: Color + Coverage

Pixel

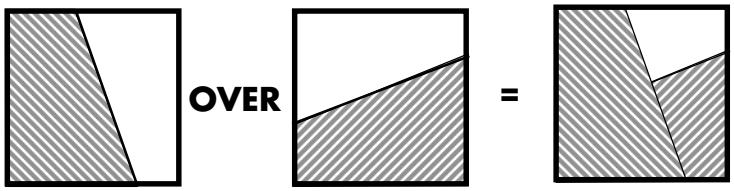
α = A
= Coverage
= Area
= Opacity
= 1 - Transparency

Final color C′ of pixel is a area-weighted average of C

$$C' = \alpha C$$

Copyright © Pat Hanrahan

OVER Operator



Composite color: $C' = A_F C_F + A_B C_B = (\alpha_F C_F) + (1-\alpha_F)(\alpha_B C_B)$

Composite alpha: $\alpha = A_F + A_B = \alpha_F + (1-\alpha_F)\ \alpha_B$

Copyright © Pat Hanrahan

Pre-multiplied Alpha

Represent as C′ = αC = (αr,αg,αb,α)

- **One formula for compositing color and alpha**

  $C' = C'_F + (1-\alpha_F)\, C'_B$

- **More efficient**

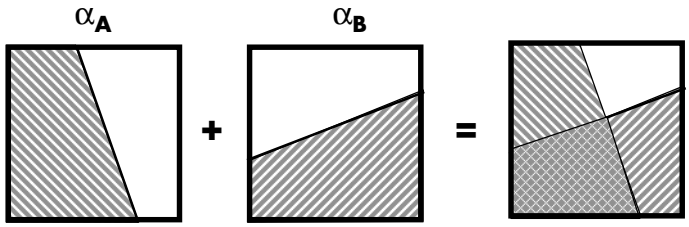
  Associated: OVER (1 sub, 4 muls, 4adds)

  Unassociated: OVER (1 sub, *7* muls, 4 adds)

- **Closure**
- **Display C′ ; C′ over K = C′ + (1-α_C) K = C′**
- **Rarely need to recover unassociated color**

Copyright © Pat Hanrahan

Region Coverage

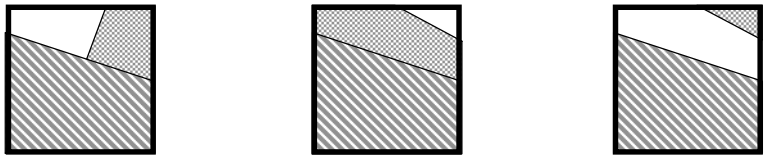


4 Regions

Region	Fraction
A ∩ B	$\alpha_A\ \alpha_B$
~A ∩ B	$(1-\alpha_A)\alpha_B$
A ∩ ~B	$\alpha_A\ (1-\alpha_A)$
~A ∩ ~B	$(1-\alpha_A)\ (1-\alpha_B)$

Copyright © Pat Hanrahan

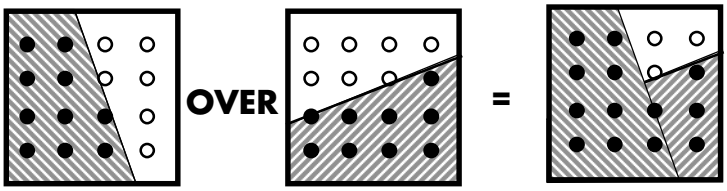
Assumptions



Uncorrelated Correlated Anticorrelated

Copyright © Pat Hanrahan

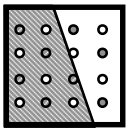
A-Buffer: Bitmask Fragments



C_A C_B $C' = bits(M_A)C_A + bits(M_B - M_A)C_B$

M_A M_B $M = M_A \mid M_B$

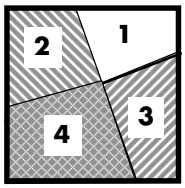
$\alpha_A = bits(M_A)$ $\alpha_B = bits(M_B)$ $\alpha = bits(M)$

Note: Screen door transparency 

Copyright © Pat Hanrahan

Porter-Duff Compositing Algebra

Enumerate the number of ways to combine two images



4 Regions

Region1:	**1 possibility - 0**
Region2:	**2 possibilities - A or 0**
Region3:	**2 possibilities - B or 0**
Region4:	**3 possibilities - A, B or 0**
Operators:	**12 total possibilities**

Copyright © Pat Hanrahan

Porter-Duff Compositing Algebra

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

$$C' = F_A C'_A + F_B C'_B$$

OpenGL `blendfunction`
Specify src and dst F's

0, 1,
As, Ad, 1-As, 1-Ad,
min(As,1-Ad),
Cs, Cd, 1-Cs, 1-Cd,

Copyright © Pat Hanrahan

Painting

Canvas **: Current image representing the painting**

Brush **: Small matte specifying the brush**

Paint **: Color and texture properties of the paint**

Paint equation

paint = color* IN *texture

compose = canvas OP paint

canvas += compose IN brush

Masks: Current selection, frisket

Copyright © Pat Hanrahan

Blue-Screen Matte Extraction

Given:

C - Observed color at each pixel

C_B - Backing color (perhaps at each pixel)

Compute:

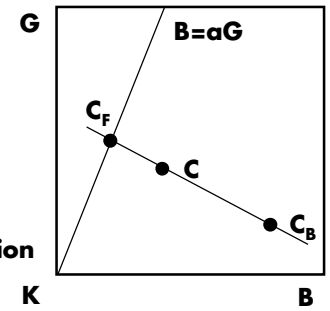
$C'_F = (\alpha_F R_F, \alpha_F G_F, \alpha_F B_F, \alpha_F)$

Matte Equation

$C = C_F + (1-\alpha_F) C_B$

(3 equations in 4 unkowns)

Vlahos Assumption



Copyright © Pat Hanrahan