CPSC 4040/6040 Computer Graphics Images

Joshua Levine
levinej@clemson.edu

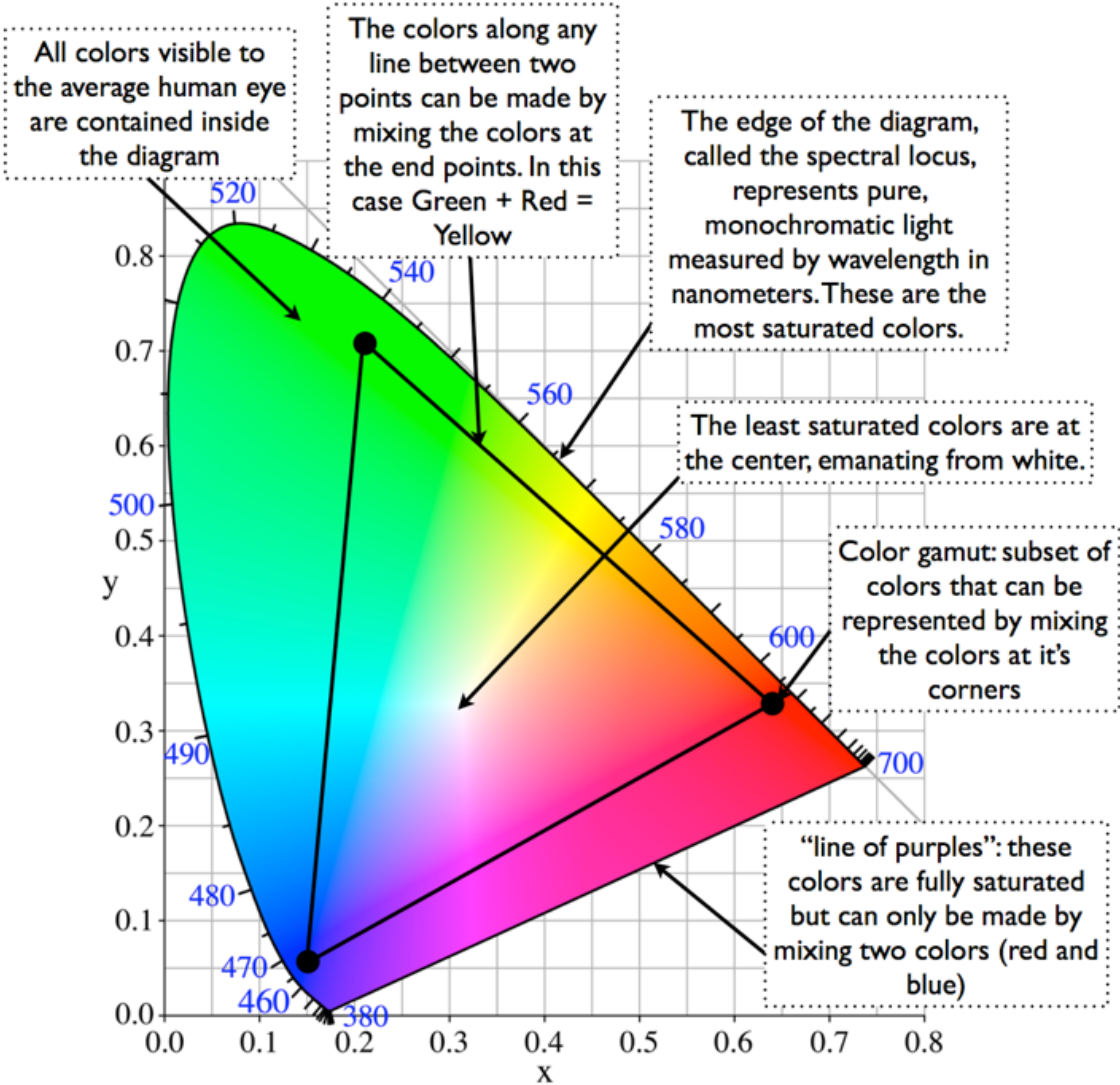
Lecture 06
Compositing

Sept. 8, 2015

Agenda

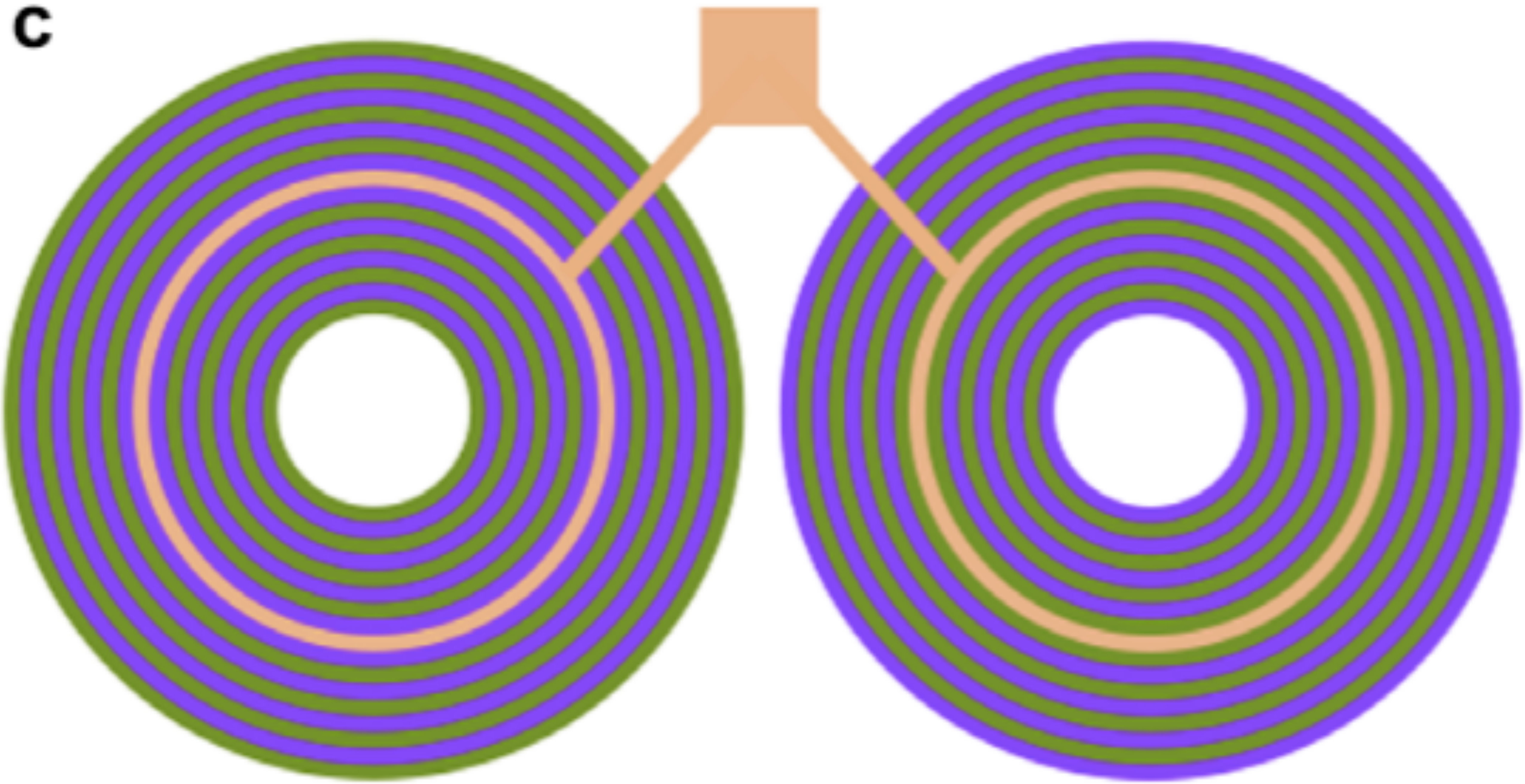
- Quiz 1 Due Thurs.
- PA02 Assigned, due Tues. Sept.15, 11:59pm
- PA01 Due at 11:59pm tonight: questions?

Continuing from Lec06

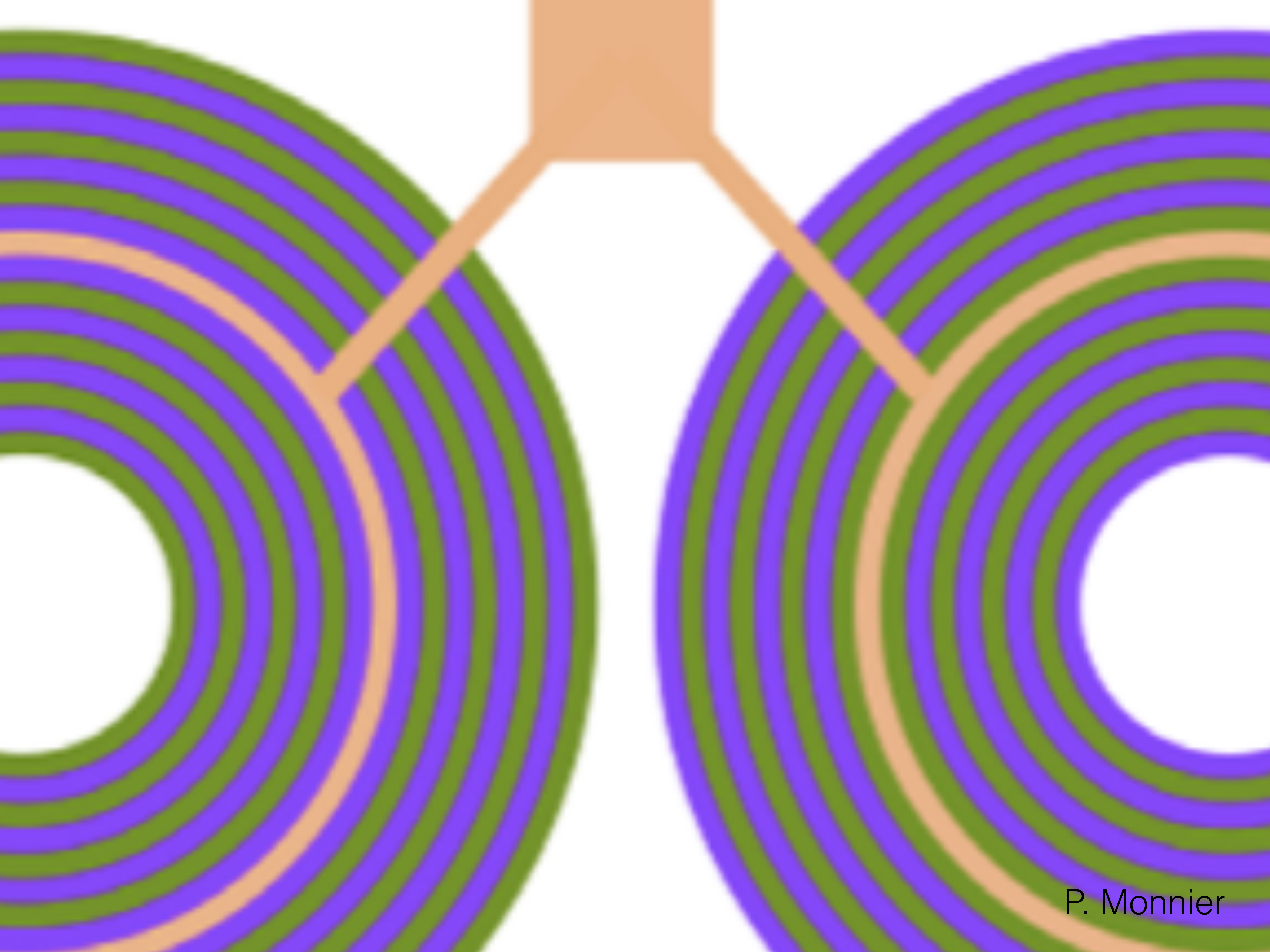


Anatomy of a CIE Chromaticity Diagram

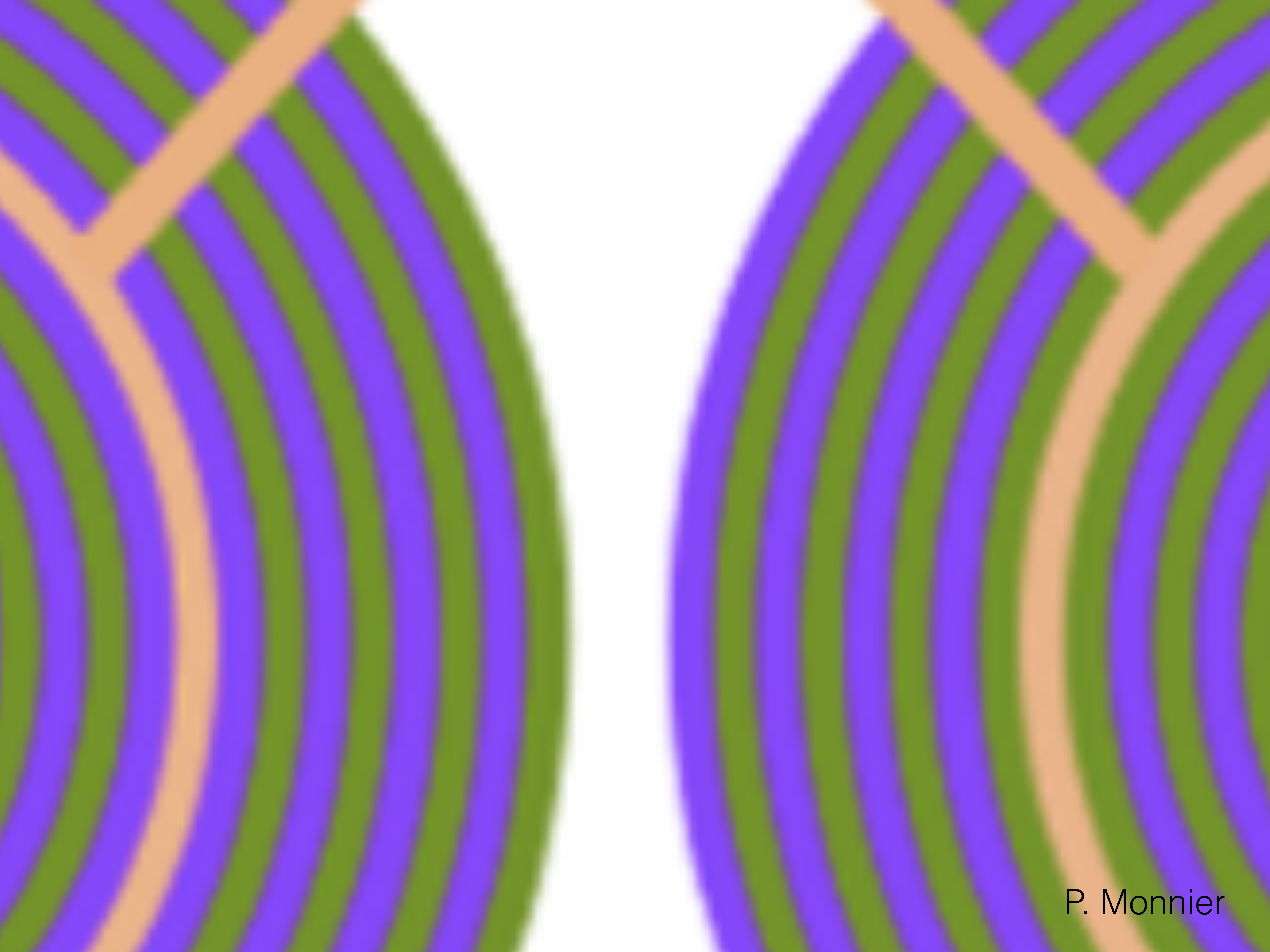
Chromatic Induction



P. Monnier



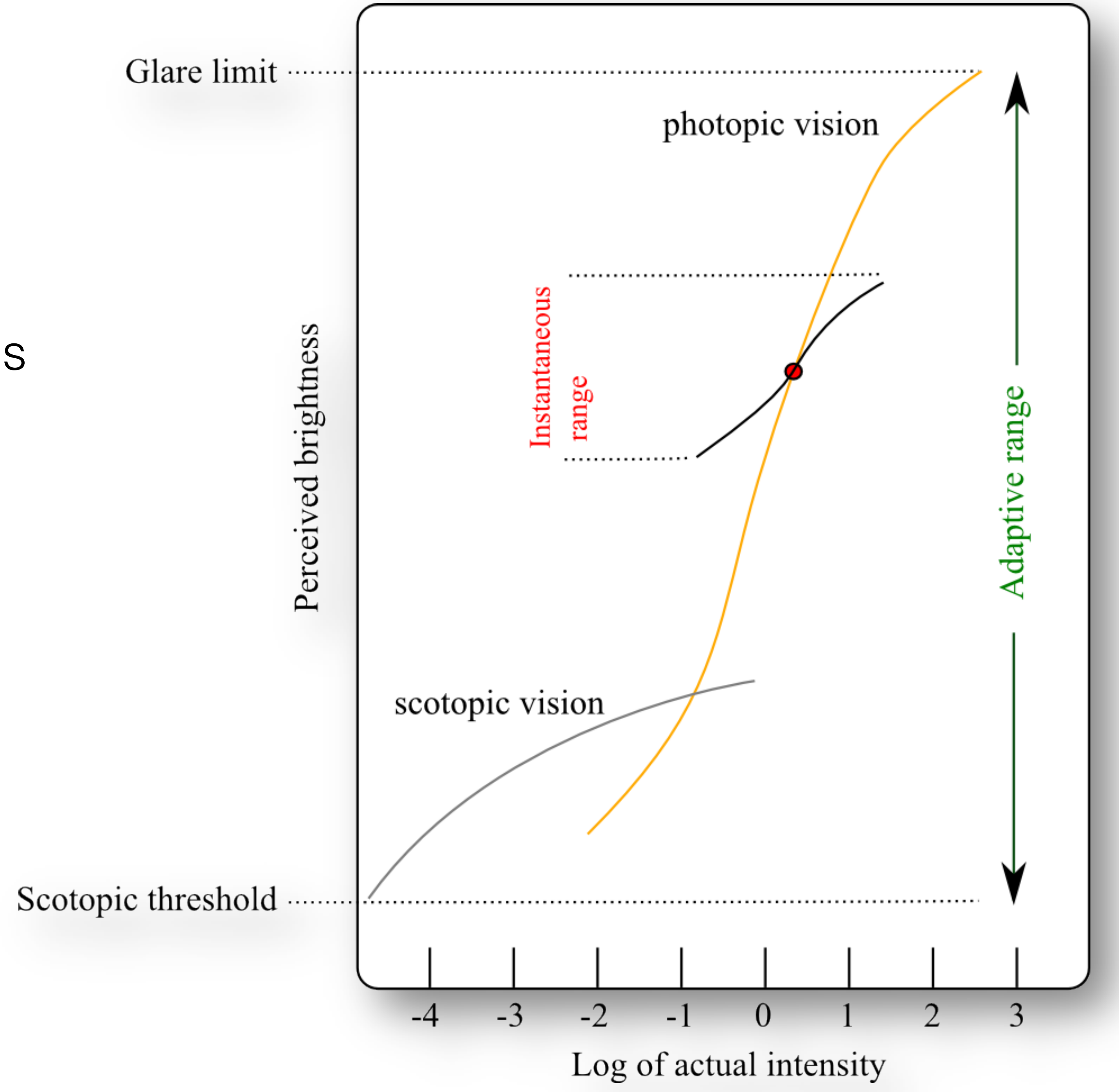
P. Monnier



P. Monnier

Brightness Adaptation

- Actual light intensity is (basically) log-compressed for perception.
- Human vision can see light between the glare limit and scotopic threshold but not all levels at the same time.
- The eye adjusts to an average value (the red dot) and can simultaneously see all light in a smaller range surrounding the adaptation level.
- Light appears black at the bottom of the instantaneous range and white at the top of that range.



Compositing Basics

A short history of compositing…

- Concept existed since the early days of film and art
- Lumière brothers, Georges Méliès, etc. used various matte techniques at the end of the 19th century
- Digital Compositing revolutionized this (see Porter and Duff, Compositing Digital Images, SIGGRAPH 1984)

Compositing Digital Images

Thomas Porter
Tom Duff †

Computer Graphics Project
Lucasfilm Ltd.

ABSTRACT

Most computer graphics pictures have been computed all at once, so that the rendering program takes care of all computations relating to the overlap of objects. There are several applications. however, where elements must be rendered separately, relying on compositing techniques for the anti-aliased accumulation of the full image. This paper presents the case for four-channel pictures, demonstrating that a matte component can be computed similarly to the color channels. The paper discusses guidelines for the generation of elements and the arithmetic for their arbitrary compositing.

CR Categories and Subject Descriptors: I.3.3 [**Computer Graphics**]: Picture/Image Generations — Display algorithms; I.3.4 [**Computer Graphics**]: Graphics Utilities — Software support; I.4.1 [**Image Processing**]: Digitization — Sampling.

General Terms: Algorithms

Additional Key Words and Phrases: compositing, matte channel, matte algebra, visible surface algorithms, graphics systems

1. Introduction

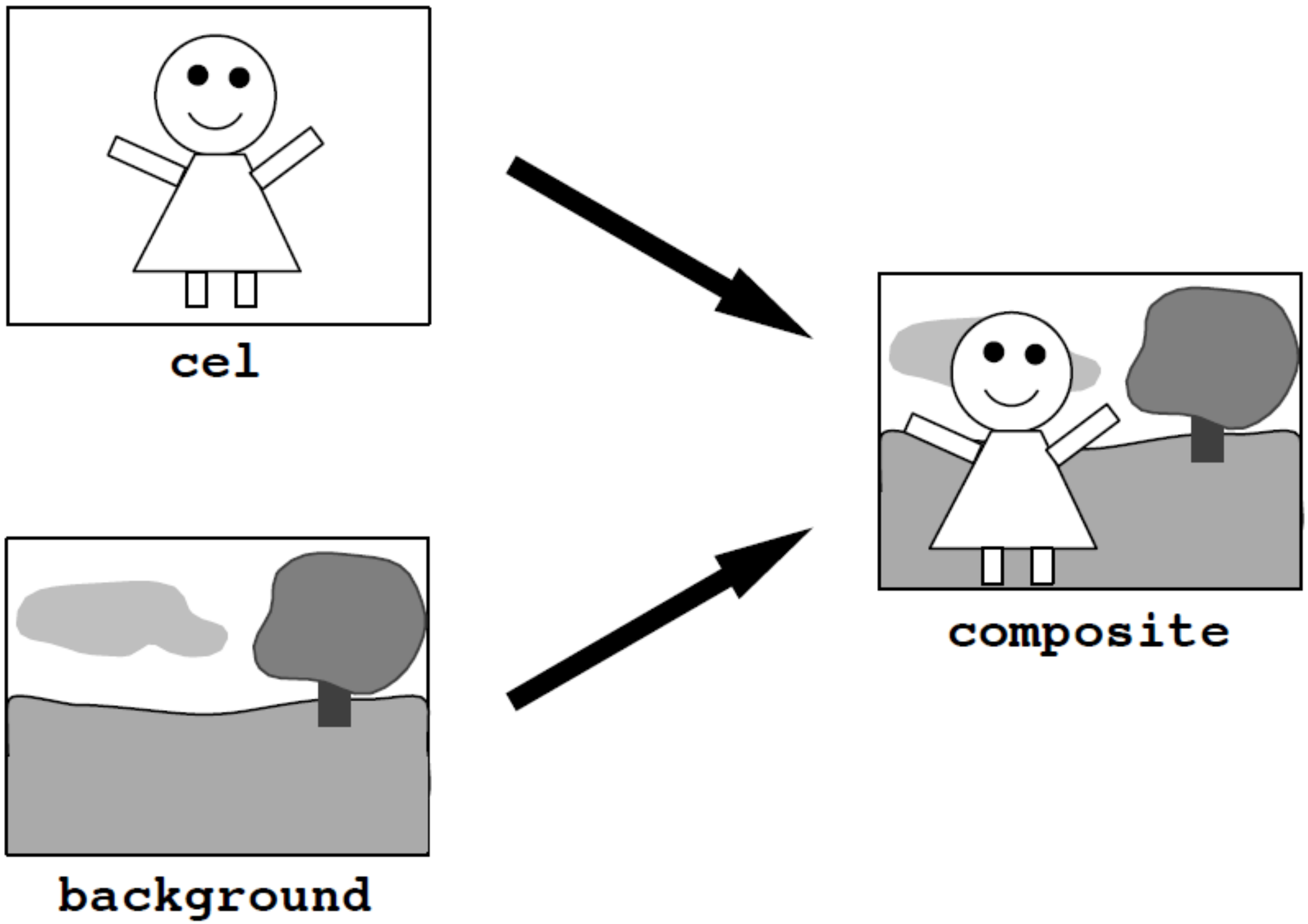
Increasingly, we find
scene cannot be fully
wealth of literature
surfaces, handling the
and quadrics and tria
texture mapping and
the basis of coherenc
suggests that multiple

In fact, reliance on
entire scene is a poor
small modeling errors
down large bodies of
order to save compila
forces only the recom
tively quick reloading
small errors in colora
not force the "recomp

Separating the ima
independently render
ment has an associ
which designates the
siting of those elemen
mulate the final imag

Basic Idea: Layering Cels

- Cels: Transparent sheets with (hand) drawings on them. (from “celluloid”)

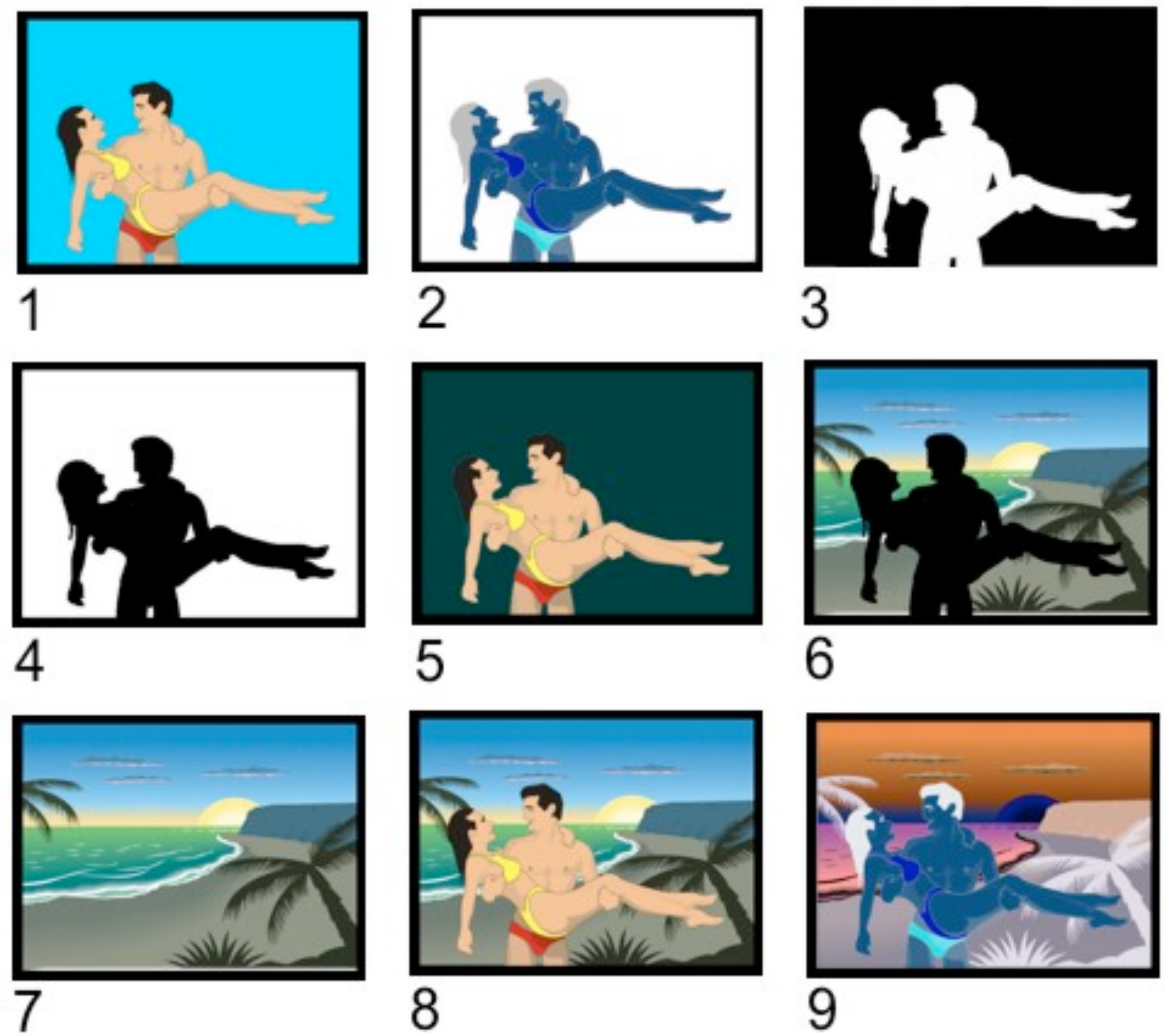


Animation Cels

$10k+ on Antique Roadshow: http://www.pbs.org/wgbh/roadshow/archive/200804A11.html

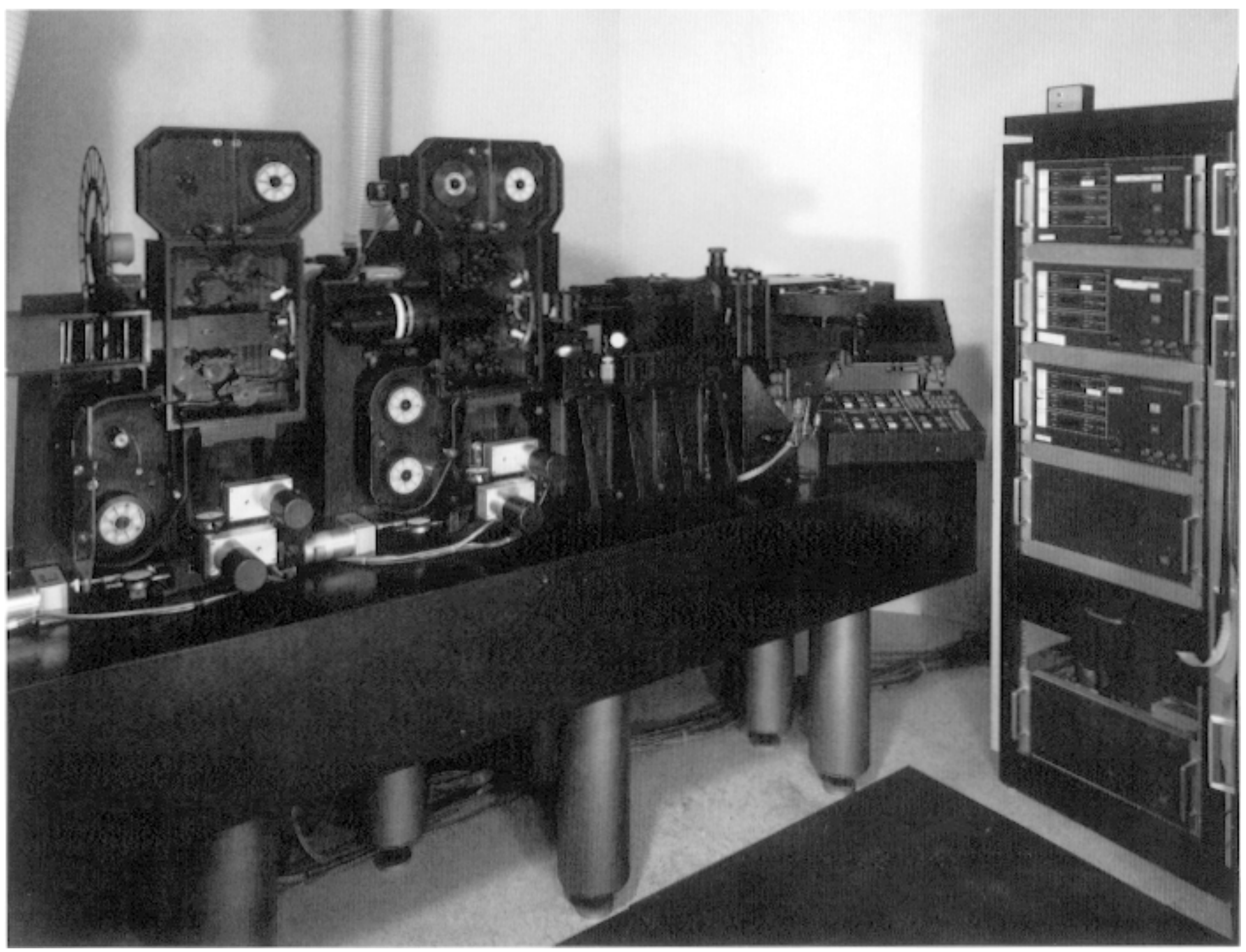
Compositing Using Mattes

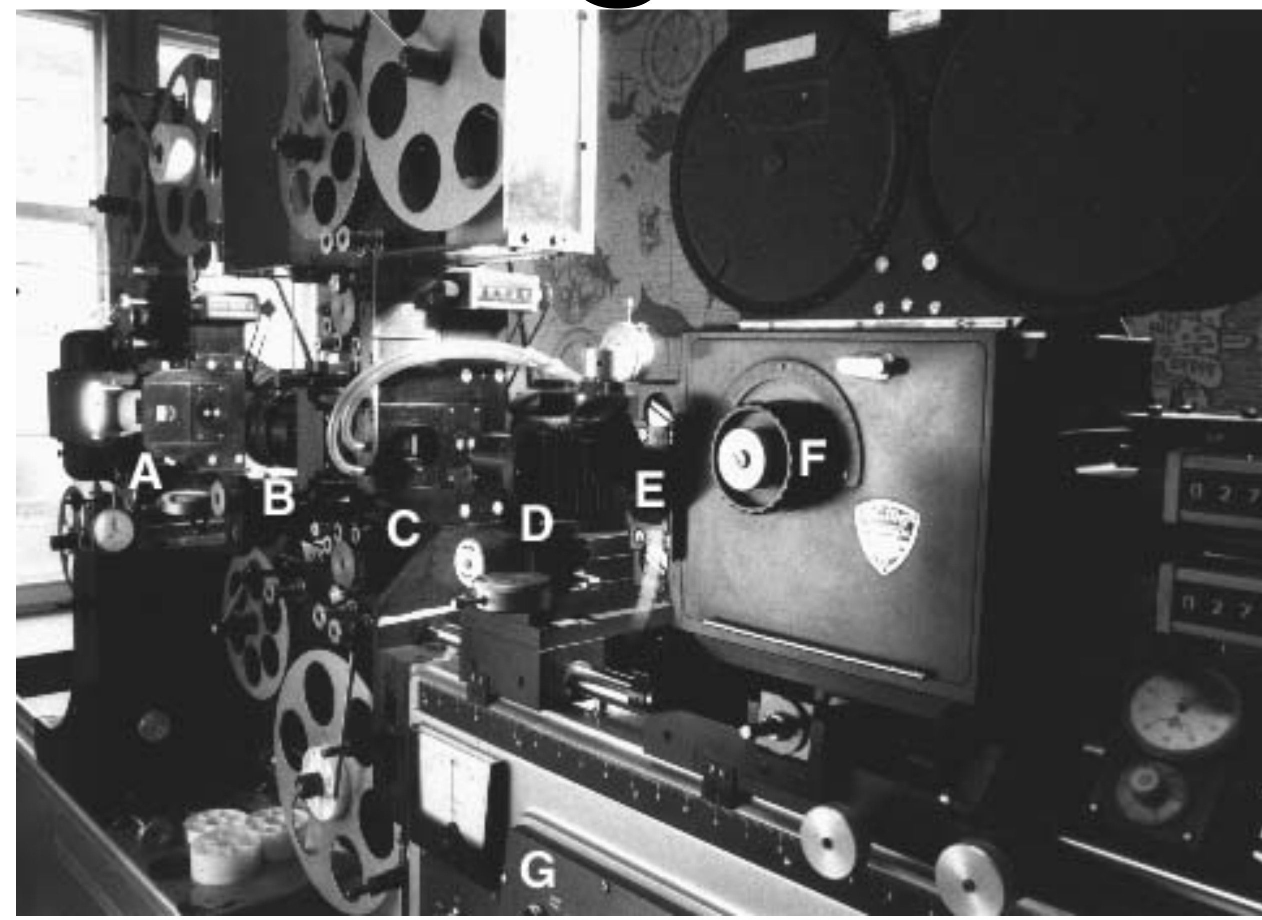
- Process by which film is partially exposed.
- Allows one to copy, create, merge, combine, different films.
- Example, filtering out the blue produces 5, reverse matte is used to remove from background and then they are combined



http://en.wikipedia.org/wiki/Compositing

Optical Printing





http://en.wikipedia.org/wiki/Optical_printing

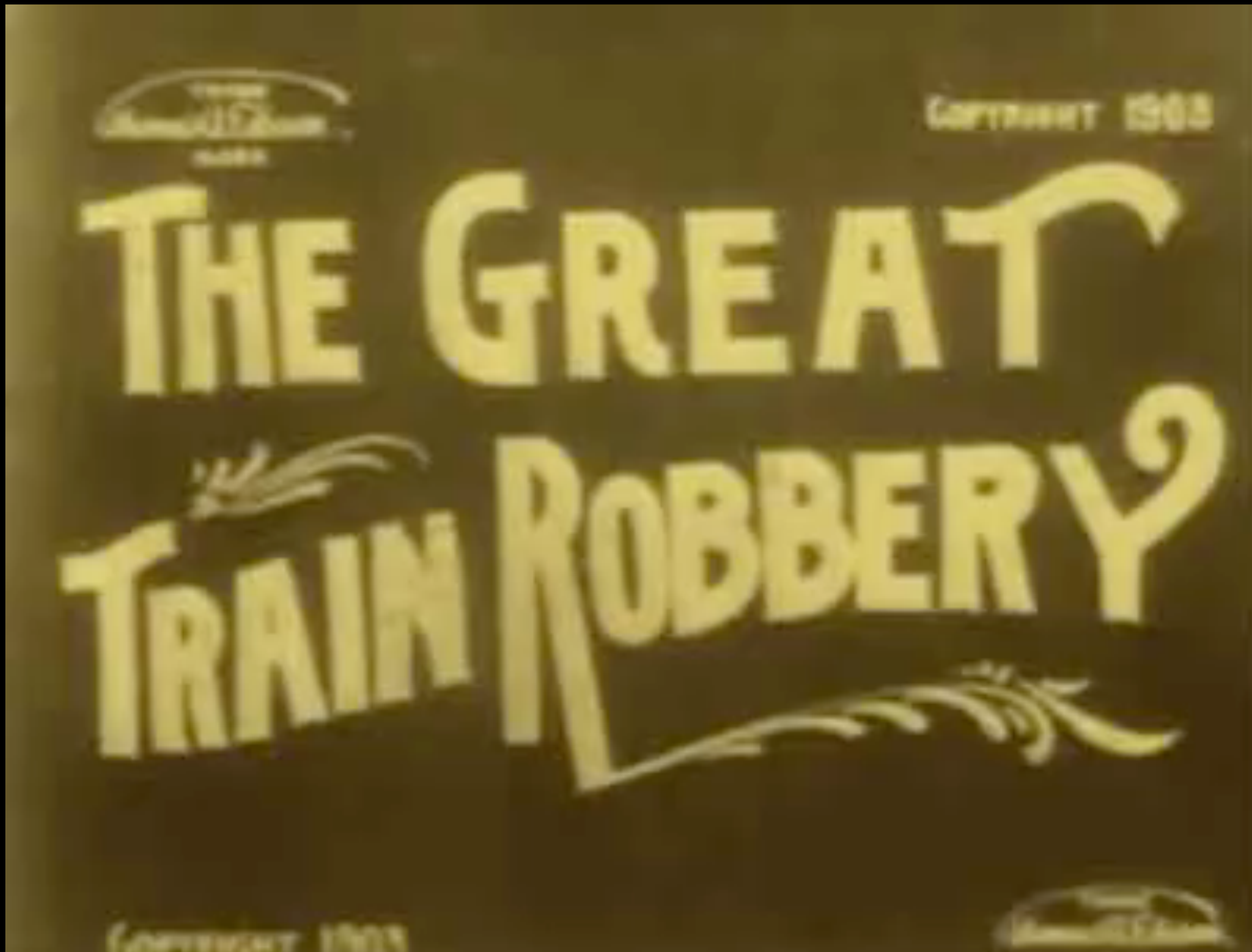
Source: “Industrial Light and Magic,” Thomas Smith (p. 181)

Digital Compositing

http://www.hongkiat.com/blog/30-nicest-photoshop-photo-effects-part-ii/

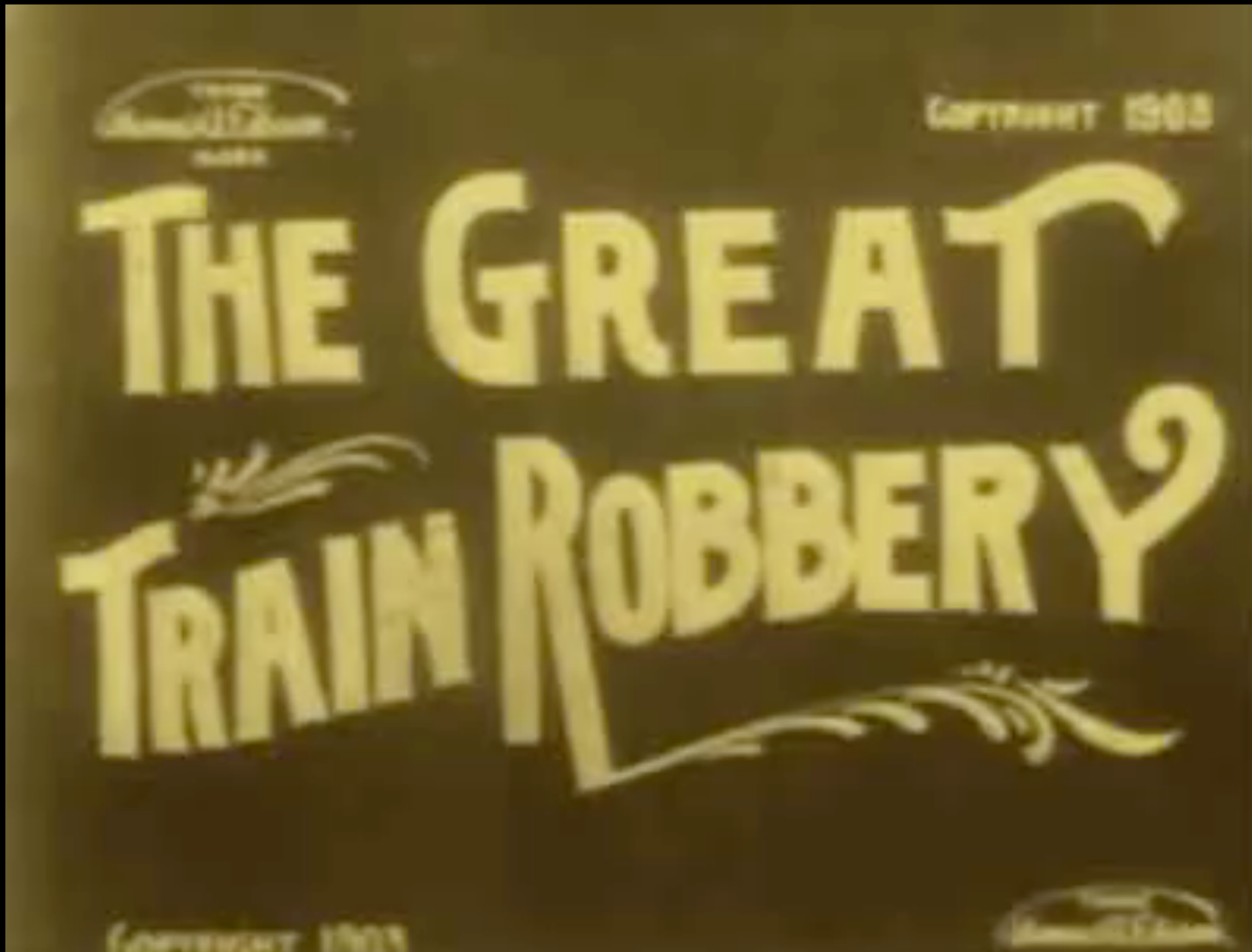
Used in Magazine Covers

Early Example from Film



http://archive.org/details/TheGreatTrainRobbery_555

Early Example from Film



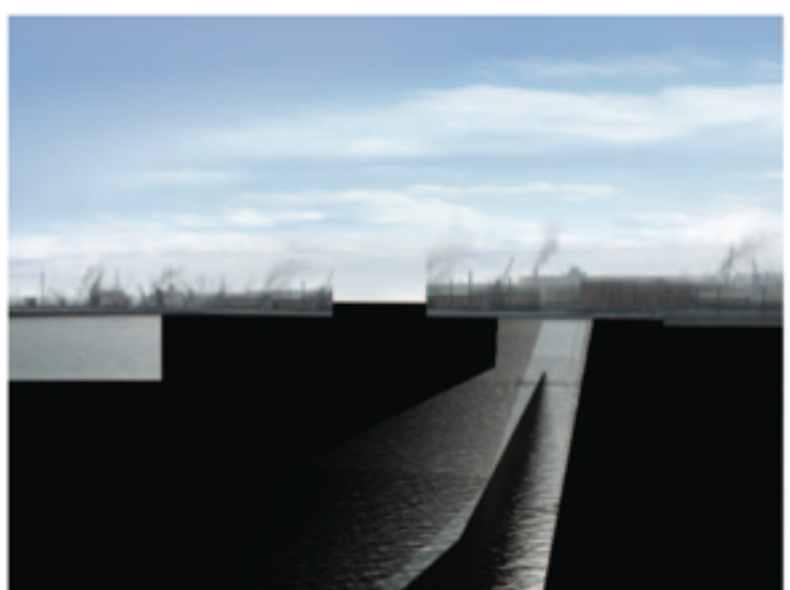
http://archive.org/details/TheGreatTrainRobbery_555

More Film Examples

Forrest Gump

Even More Film Examples

Figure 15.20 A composite image created for the film *Titanic*. Images from *Titanic* courtesy Twentieth Century Fox Film Corporation. © 1997 by Twentieth Century Fox Film Corporation. All rights reserved.



The Art and Science of Digital Compositing, Brinkmann

Even More Film Examples

Figure 15.20 A composite image created for the film *Titanic*. Images from *Titanic* courtesy Twentieth Century Fox Film Corporation. © 1997 by Twentieth Century Fox Film Corporation. All rights reserved.

Figure 15.27 The compositing script used to create the image shown in Figure 15.20. Sections are magnified to show detail. © 1997 by Twentieth Century Fox Film Corporation.

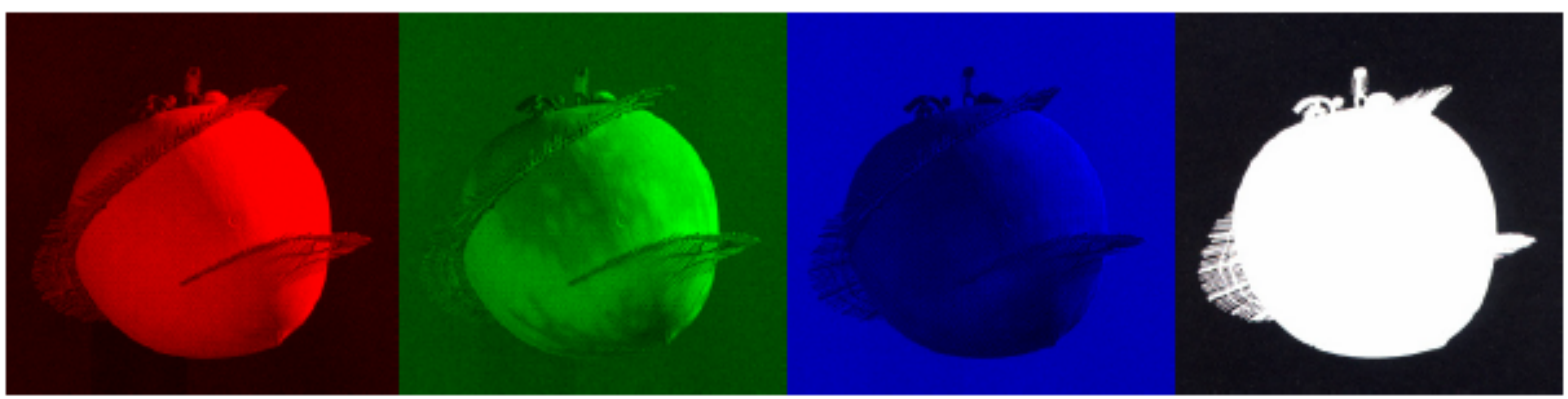
The Art and Science of Digital Compositing, Brinkmann

Four Channel Images

What is α (alpha)?

- α is a measurement of the opacity.
- α=1 means fully opaque.
- α=0 means fully transparent.

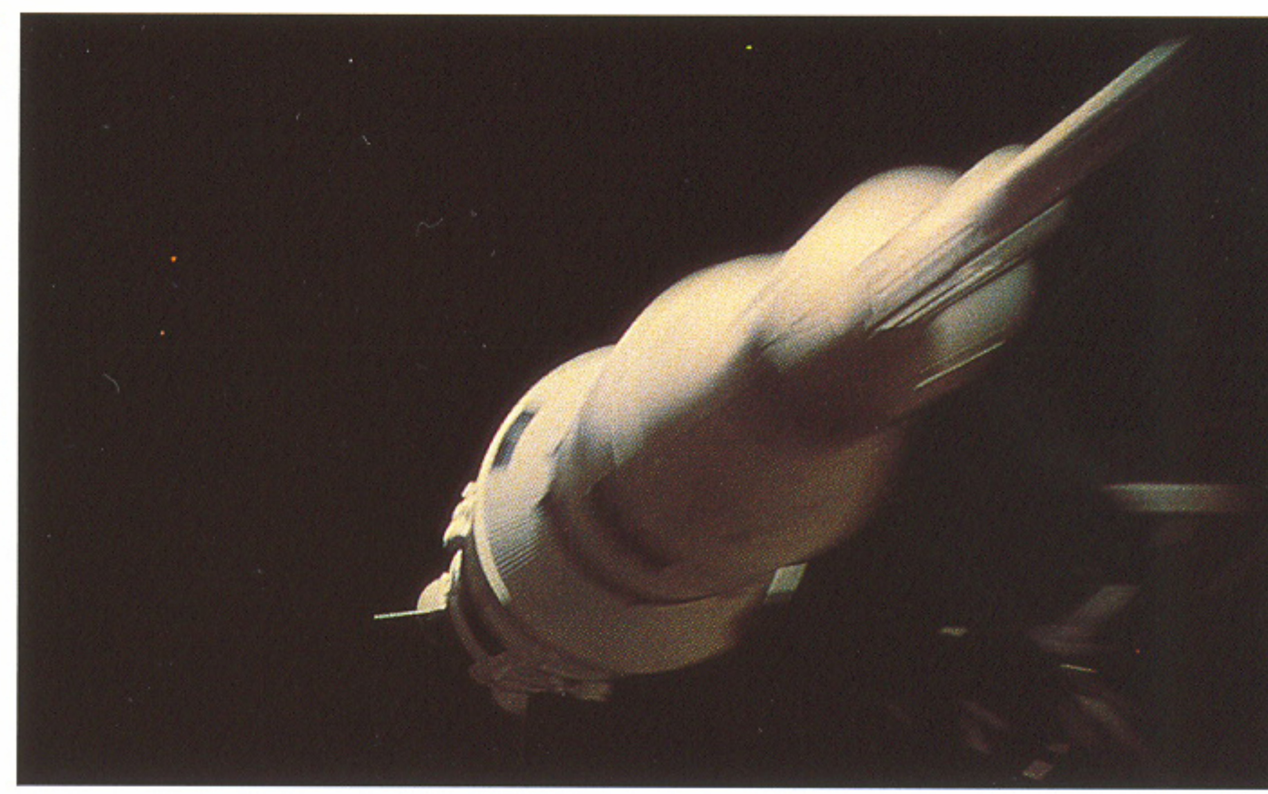
Red Green Blue α



The Art and Science of Digital Compositing, Brinkmann

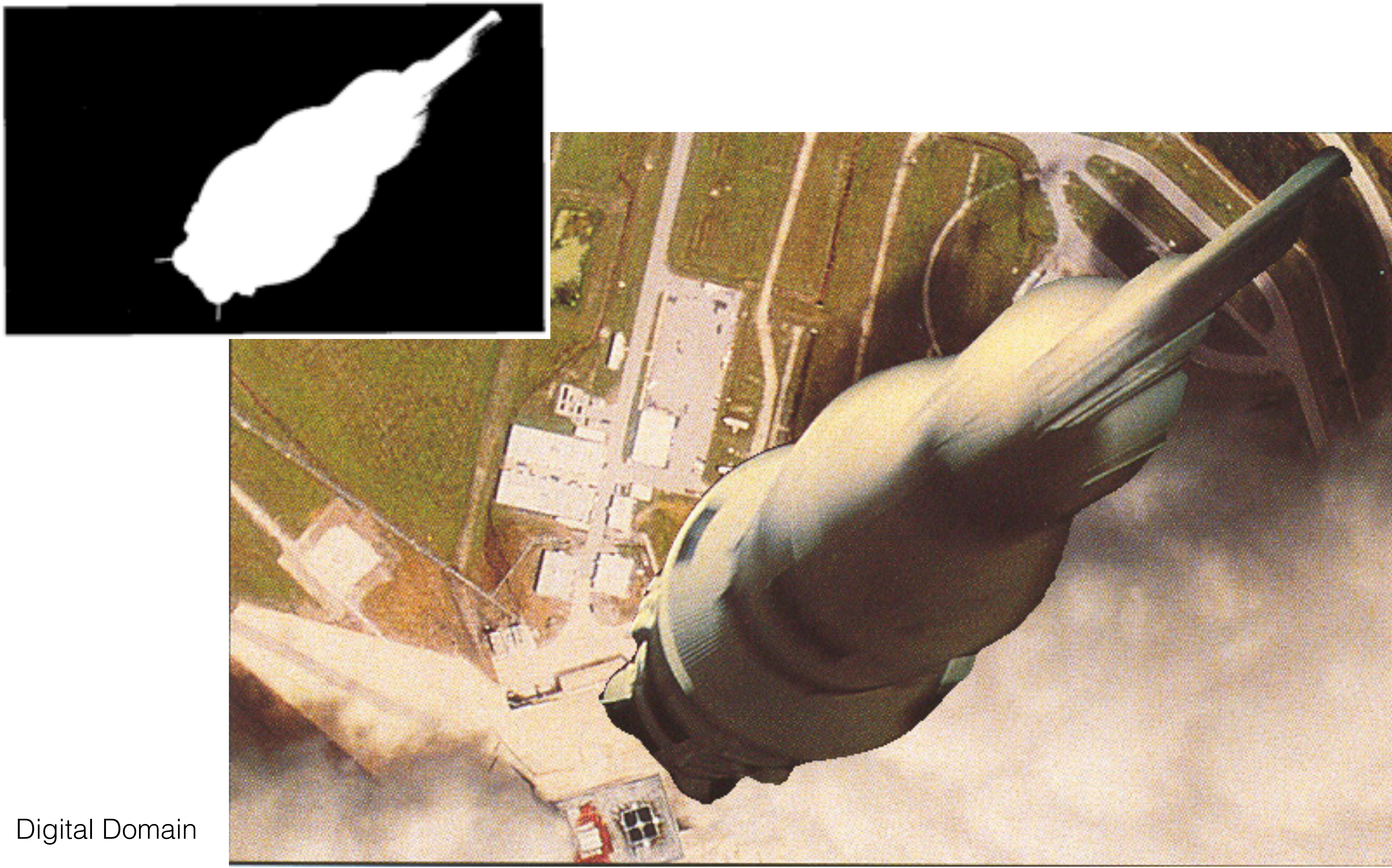
Interpreting Fractional α

- $0 < \alpha < 1$ means some percentage of coverage
  - Thus, some amount of light penetrates
- Useful for hair, motion, partial occlusion, and more



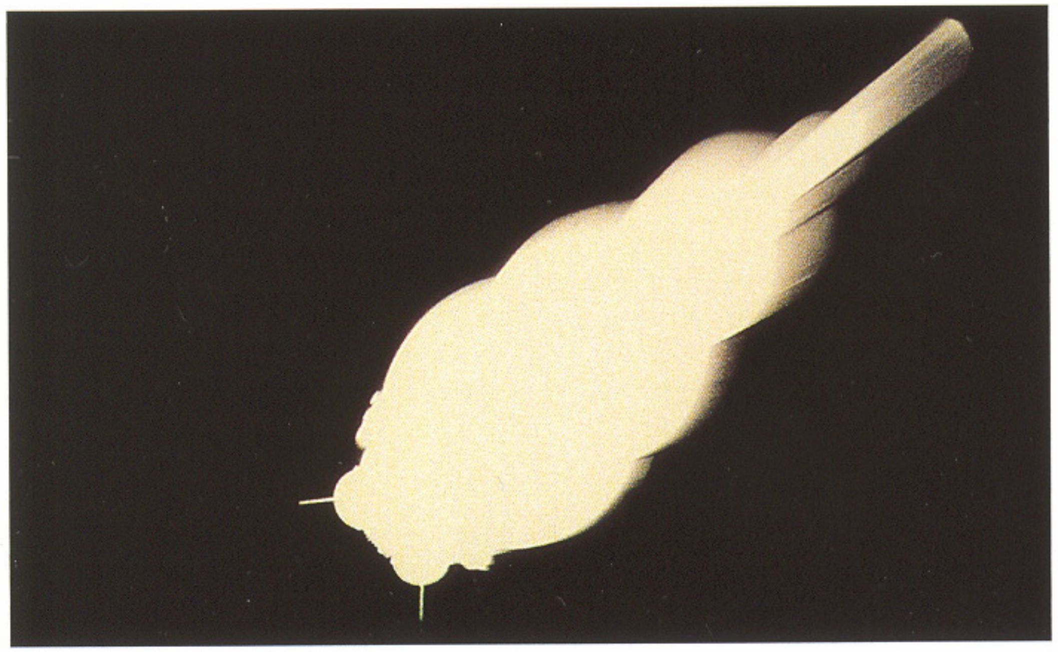
Digital Domain

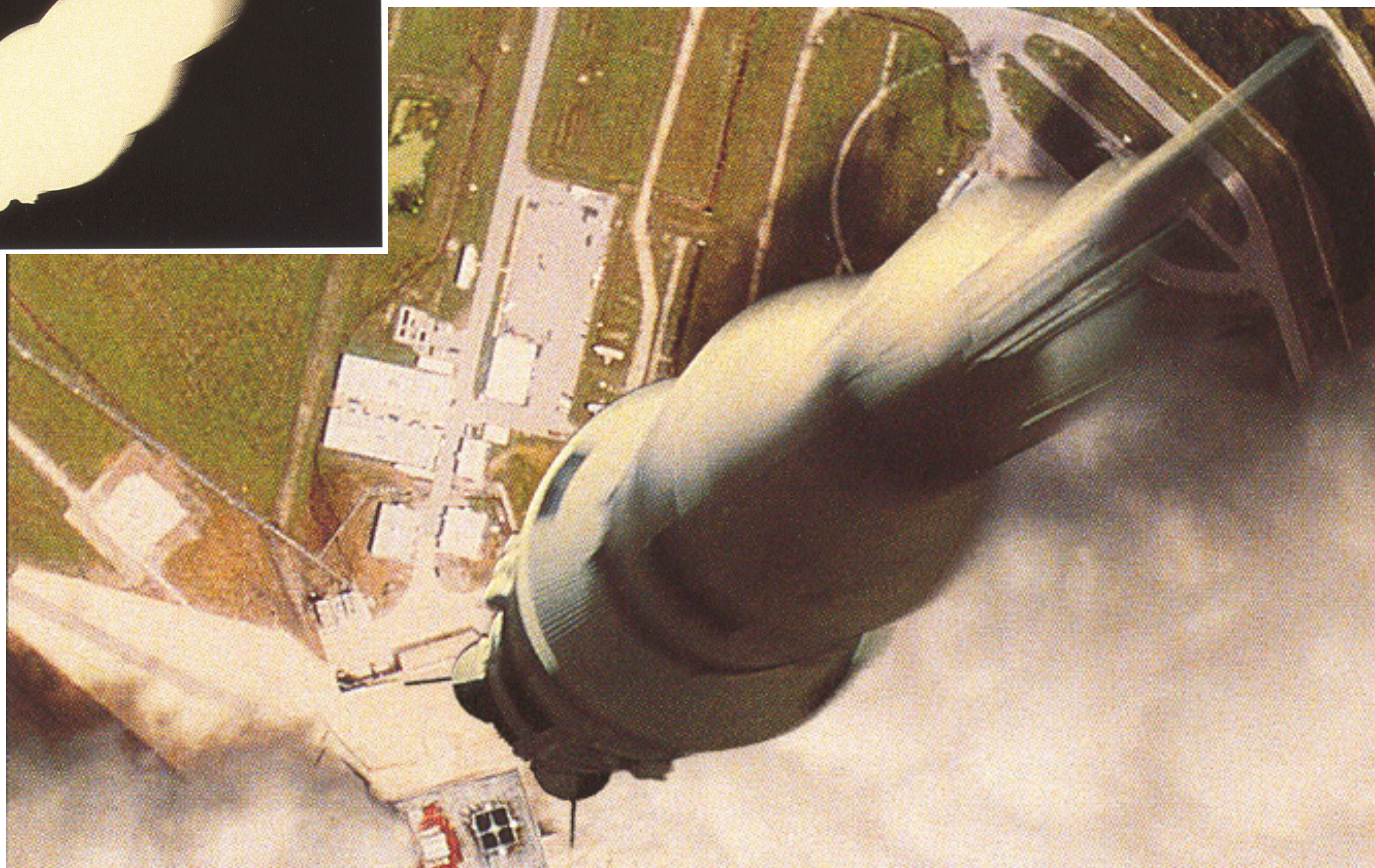
Compositing with Binary Alpha



Digital Domain

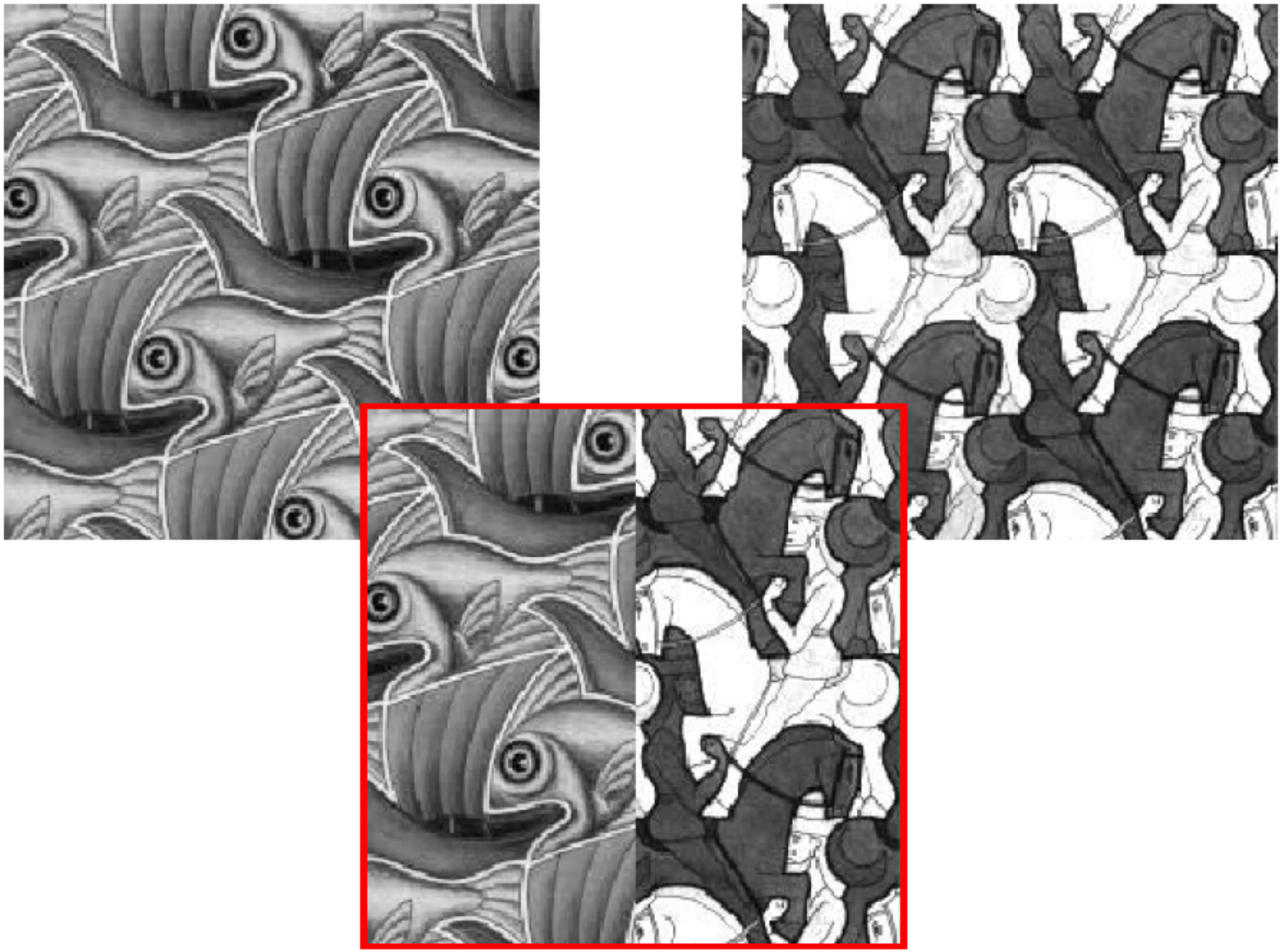
Compositing with Fractional Alpha





Digital Domain

Other Uses of Alpha: Image Editing

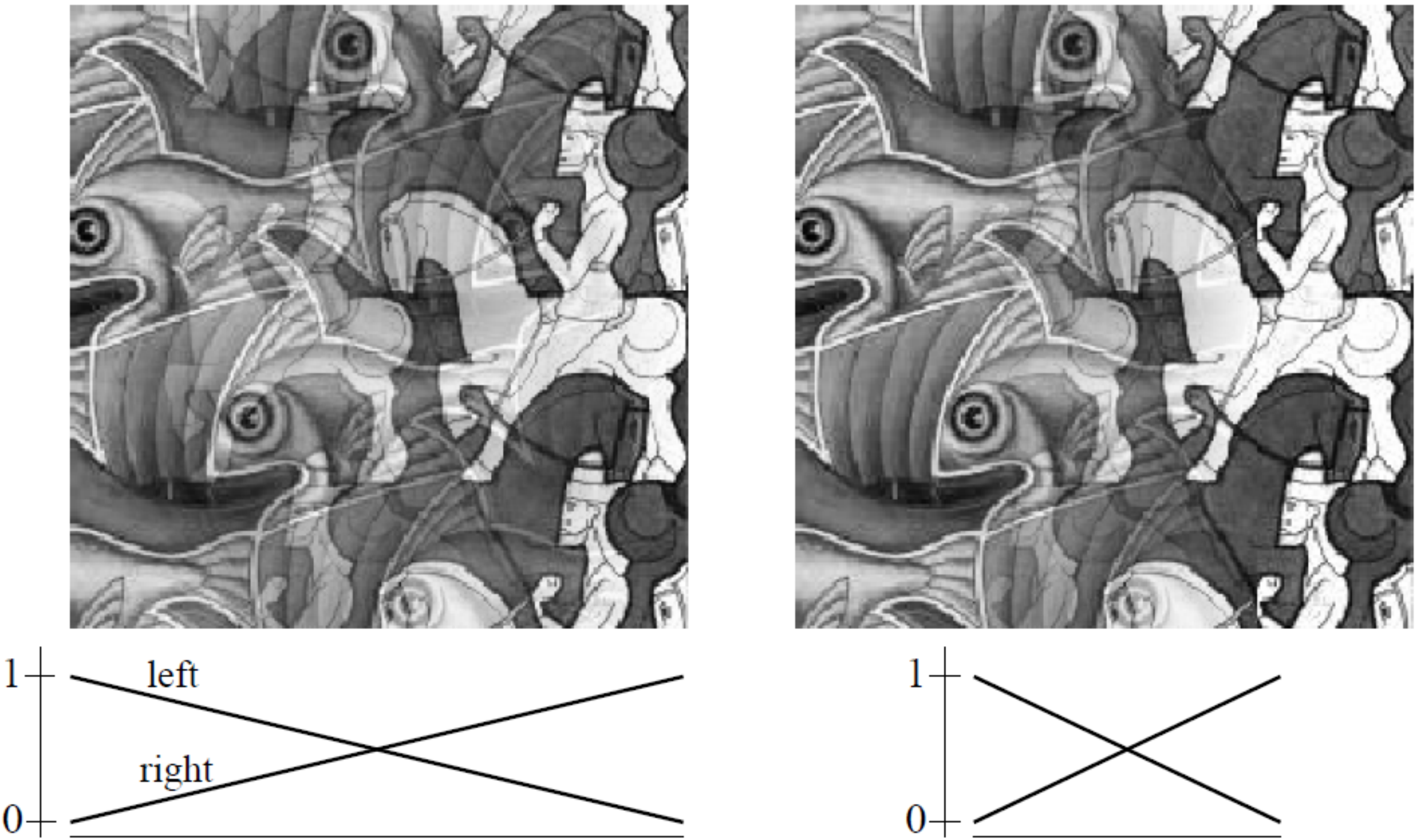


How to smooth the seam, even for non-physical images?

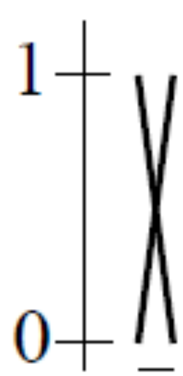
Feathering / Alpha Blending

- Encode distance in the transparency channel
- $C_{blend} = \boldsymbol{\alpha} C_A + (1-\boldsymbol{\alpha}) C_B$

Affect of Window Size (ghosting)



Affect of Window Size (seam visibility)

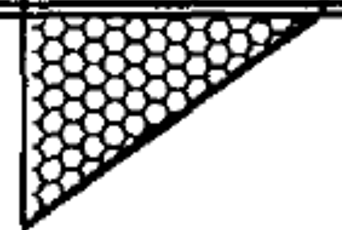
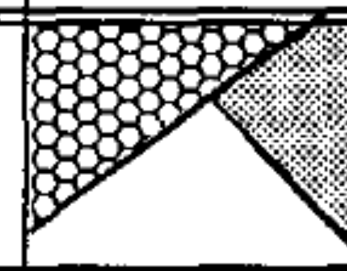


“Arithmetic” of Compositing

Over Operator

- “Painters’ Algorithm”
- Porter and Duff, 1984: “The paper discusses guidelines for the generation of elements and the arithmetic for their arbitrary compositing.”
- **over** is one of many such operators in this arithmetic

The Over Operator

clear	(0,0,0,0)		0	0
A	(0,A,0,A)		1	0
B	(0,0,B,B)		0	1
A **over** *B*	(0,A,B,A)		1	$1-\alpha_A$

- Written “A **over** B”
- B, background, painted first
- A, foreground, painted on top of B
- Where $\boldsymbol{\alpha}_A < 1$, view penetrates through A

Porter & Duff, 1984

Over Operator Defined

- Each color channel C treated separated:

$$C_P = \boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)C_B$$

- Linear interpolation of colors
- More generally (if background is transparent):

$$C_P = \boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B$$

Order Dependence (Associativity)

- Say we have three images, A, B, D:
- What about A **over** B **over** D?
- Two choices:
  - A **over** (B **over** D)
  - (A **over** B) **over** D
- Are they the same?

A **over** (B **over** D)

- A **over** (B **over** D)
- B **over** D => $\boldsymbol{\alpha}_B C_B + (1-\boldsymbol{\alpha}_B) C_D$
- $C_P = \boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)[\boldsymbol{\alpha}_B C_B + (1-\boldsymbol{\alpha}_B) C_D]$
- Set final $\boldsymbol{\alpha}_P = 1$ (assuming $\boldsymbol{\alpha}_D = 1$)

(A **over** B) **over** D

- (A **over** B) **over** D
- H = A **over** B, next compute H **over** D
- $C_P = \boldsymbol{\alpha}_H C_H + (1-\boldsymbol{\alpha}_H)C_D$
    - Does this equal (copied and expanded from last slide)?

      $$C_P = \boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B + (1-\boldsymbol{\alpha}_A)(1-\boldsymbol{\alpha}_B)C_D$$
- If it does, it must be that:

(A **over** B) **over** D

- (A **over** B) **over** D
- H = A **over** B, next compute H **over** D
- $C_P = \boldsymbol{\alpha}_H C_H + (1-\boldsymbol{\alpha}_H) C_D$
    - Does this equal (copied and expanded from last slide)?

$$C_P = \boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B + (1-\boldsymbol{\alpha}_A)(1-\boldsymbol{\alpha}_B) C_D$$

- If it does, it must be that:

1. $\boldsymbol{\alpha}_H C_H = \boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B$

(A **over** B) **over** D

- (A **over** B) **over** D
- H = A **over** B, next compute H **over** D
- $C_P = \boxed{\boldsymbol{\alpha}_H C_H} + (1-\boldsymbol{\alpha}_H)C_D$
    - Does this equal (copied and expanded from last slide)?

$$C_P = \boxed{\boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B} + (1-\boldsymbol{\alpha}_A)(1-\boldsymbol{\alpha}_B)C_D$$

- If it does, it must be that:

1. $\boldsymbol{\alpha}_H C_H = \boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B$

(A **over** B) **over** D

- (A **over** B) **over** D
- H = A **over** B, next compute H **over** D
- $C_P = \boxed{\boldsymbol{\alpha}_H C_H} + (1-\boldsymbol{\alpha}_H)C_D$
    - Does this equal (copied and expanded from last slide)?

$$C_P = \boxed{\boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B} + (1-\boldsymbol{\alpha}_A)(1-\boldsymbol{\alpha}_B)C_D$$

- If it does, it must be that:

1. $\boldsymbol{\alpha}_H C_H = \boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B$
2. $(1-\boldsymbol{\alpha}_H) = (1-\boldsymbol{\alpha}_A)(1-\boldsymbol{\alpha}_B) \Rightarrow \boldsymbol{\alpha}_H = \boldsymbol{\alpha}_A + (1 - \boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B$

(A **over** B) **over** D

- (A **over** B) **over** D
- H = A **over** B, next compute H **over** D
- $C_P = \boxed{\boldsymbol{\alpha}_H C_H} + \boxed{(1-\boldsymbol{\alpha}_H) C_D}$
  - Does this equal (copied and expanded from last slide)?

    $$C_P = \boxed{\boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B} + \boxed{(1-\boldsymbol{\alpha}_A)(1-\boldsymbol{\alpha}_B) C_D}$$

- If it does, it must be that:

  1. $\boldsymbol{\alpha}_H C_H = \boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B$
  2. $(1-\boldsymbol{\alpha}_H) = (1-\boldsymbol{\alpha}_A)(1-\boldsymbol{\alpha}_B) \Rightarrow \boldsymbol{\alpha}_H = \boldsymbol{\alpha}_A + (1 - \boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B$

(A **over** B) **over** D

- (A **over** B) **over** D
- H = A **over** B, next compute H **over** D
- $C_P = \boxed{\boldsymbol{\alpha}_H C_H} + \boxed{(1-\boldsymbol{\alpha}_H)C_D}$
  - Does this equal (copied and expanded from last slide)?

$$C_P = \boxed{\boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B} + \boxed{(1-\boldsymbol{\alpha}_A)(1-\boldsymbol{\alpha}_B)C_D}$$

- If it does, it must be that:

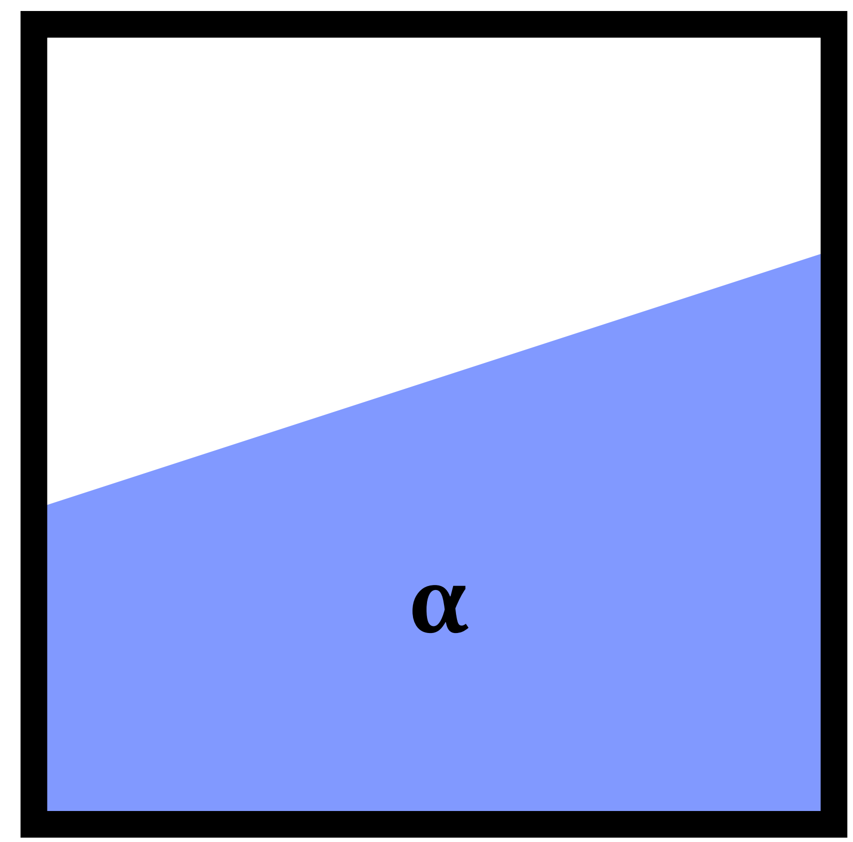
  1. $\boldsymbol{\alpha}_H C_H = \boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B$
  2. $(1-\boldsymbol{\alpha}_H) = (1-\boldsymbol{\alpha}_A)(1-\boldsymbol{\alpha}_B) \Rightarrow \boldsymbol{\alpha}_H = \boldsymbol{\alpha}_A + (1 - \boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B$

- Can then solve for $C_H = \boldsymbol{\alpha}_A/\boldsymbol{\alpha}_H C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B/\boldsymbol{\alpha}_H C_B$

Associated Color Images

Premultiplying Color

- Problem: have to compute C and $\boldsymbol{\alpha}$ separately
- Instead, use a **premultiplied** color, c = $\boldsymbol{\alpha}$C
- Idea: alpha is some percentage of (subpixel) color
- $\boldsymbol{\alpha}$ = area of blue = pixel coverage
- Convention: use lowercase letters for premultiplied colors and UPPERCASE for normal



A Hypothetical Pixel

over with Premultiplied Colors

- Without Premultiplication: $C_P = \boldsymbol{\alpha}_A C_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B C_B$
- Premultiplied: $c_P = c_A + (1-\boldsymbol{\alpha}_A)c_B$
  - Same for alpha: $\boldsymbol{\alpha}_P = \boldsymbol{\alpha}_A + (1-\boldsymbol{\alpha}_A)\boldsymbol{\alpha}_B$
- Interpretation: a premultiplied $(r,g,b,\boldsymbol{\alpha})$ means that the real color is $(R,G,B) = (\boldsymbol{\alpha}r, \boldsymbol{\alpha}r, \boldsymbol{\alpha}g)$

Arithmetic with RGBA vs. rgb

- Two ways of operating on pixels (note: ending up with premultiplied colors)
  1. Separately operate on $\boldsymbol{\alpha}$ and C; then multiply to get c = $\boldsymbol{\alpha}$*C
  2. Operate on premultiplied colors c (multiply first, then operate)
- These lead to *different* answers!!

- For example, average the pixels $(C_1,\boldsymbol{\alpha}_1) = (1,0,0,0)$ and $(C_2,\boldsymbol{\alpha}_2) = (0,1,0,1)$:

Arithmetic with RGBA vs. rgb

- Two ways of operating on pixels (note: ending up with premultiplied colors)
    1. Separately operate on $\boldsymbol{\alpha}$ and C; then multiply to get $c = \boldsymbol{\alpha}^*C$
    2. Operate on premultiplied colors c (multiply first, then operate)
- These lead to *different* answers!!

- For example, average the pixels $(C_1,\boldsymbol{\alpha}_1) = (1,0,0,0)$ and $(C_2,\boldsymbol{\alpha}_2) = (0,1,0,1)$:
    1. R = ½(1+0); G = ½(0+1); B = ½(0+0); $\boldsymbol{\alpha}$ = ½(0+1). So C = (1/2,1/2,0,1/2); and then we premultiply to get (1/4,1/4,0,1/2)

Arithmetic with RGBA vs. rgb

- Two ways of operating on pixels (note: ending up with premultiplied colors)

    1. Separately operate on $\boldsymbol{\alpha}$ and C; then multiply to get $c = \boldsymbol{\alpha}^*C$

    2. Operate on premultiplied colors c (multiply first, then operate)

- These lead to *different* answers!!

- For example, average the pixels $(C_1,\boldsymbol{\alpha}_1) = (1,0,0,0)$ and $(C_2,\boldsymbol{\alpha}_2) = (0,1,0,1)$:

    1. R = ½(1+0); G = ½(0+1); B = ½(0+0); $\boldsymbol{\alpha}$ = ½(0+1). So C = (1/2,1/2,0,1/2); and then we premultiply to get (1/4,1/4,0,1/2)

    2. First premultiply to get (0,0,0,0) and (0,1,0,1); then average to get (0,1/2,0,1/2)

Arithmetic with RGBA vs. rgb

- Two ways of operating on pixels (note: ending up with premultiplied colors)
    1. Separately operate on $\boldsymbol{\alpha}$ and C; then multiply to get c = $\boldsymbol{\alpha}$*C
    2. Operate on premultiplied colors c (multiply first, t
- These lead to *different* answers!!

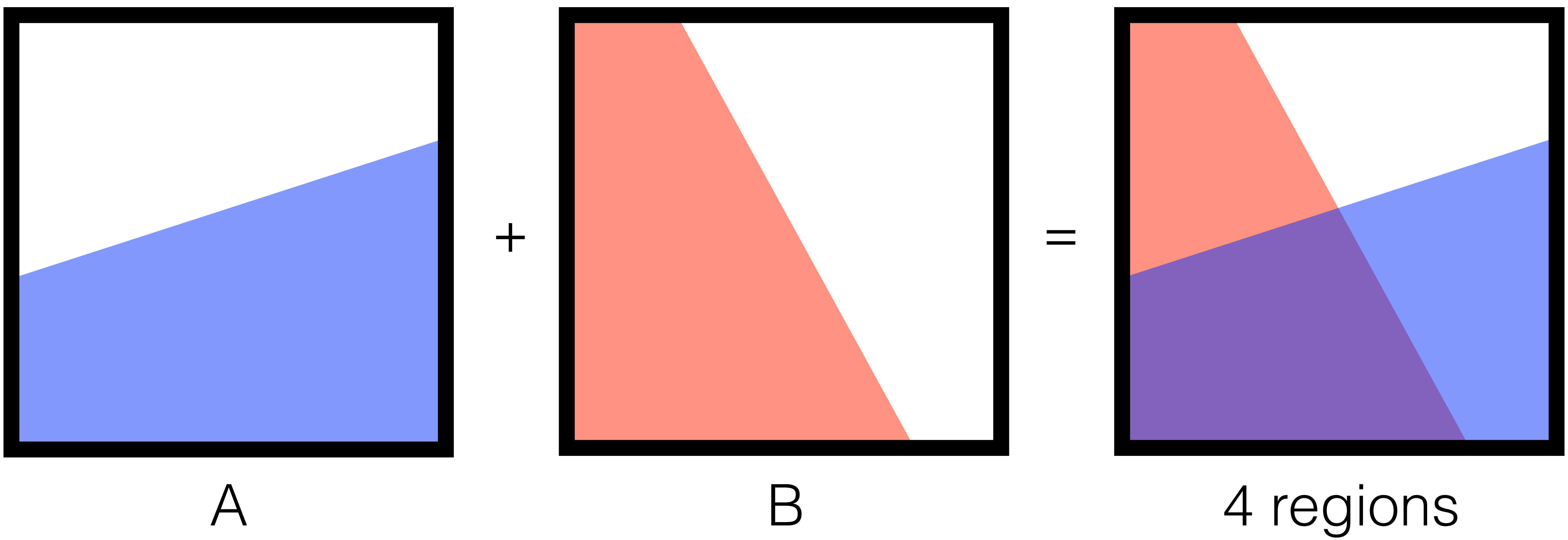
The R channel "leaked" out of transparent pixel

- For example, average the pixels $(C_1,\boldsymbol{\alpha}_1) = (1,0,0,0)$ and $(C_2,\boldsymbol{\alpha}_2) = (0,1,0,1)$:
    1. R = ½(1+0); G = ½(0+1); B = ½(0+0); $\boldsymbol{\alpha}$ = ½(0+1). So C = (1/2,1/2,0,1/2); and then we premultiply to get (1/4,1/4,0,1/2)
    2. First premultiply to get (0,0,0,0) and (0,1,0,1); then average to get (0,1/2,0,1/2)

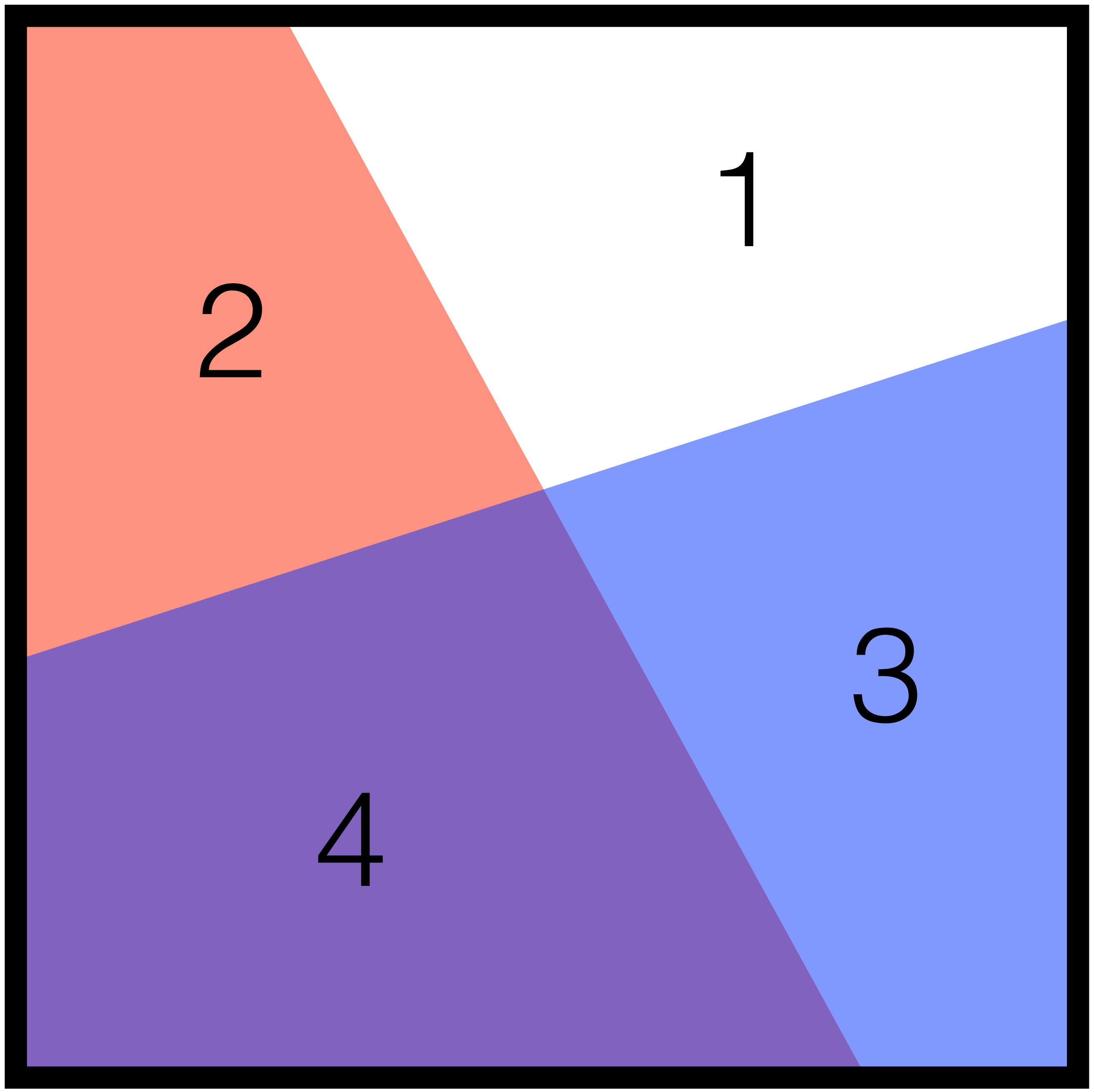
Why is Option #2 Correct?

- We want a system where it is identical to either:
  - Composite foreground over the background and then operate, or
  - Operate on foreground and background and then composite
- This is only possible with #2, where we operate c instead of C

Operations on Associated Colors



Porter-Duff Composition



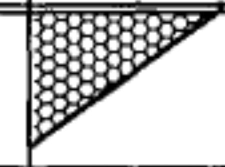
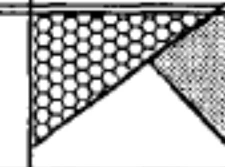
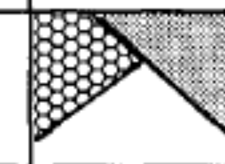
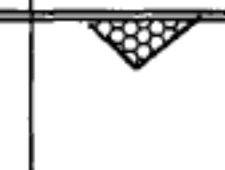
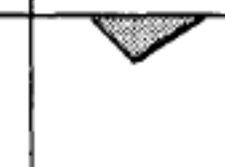
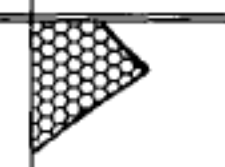
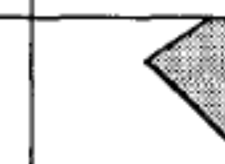
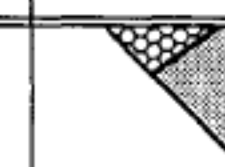
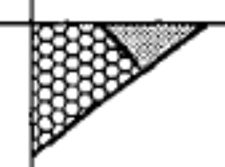
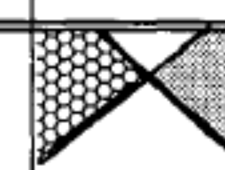
4 regions

- Region 1: 1 possibility - 0
- Region 2: 2 possibilities - A or 0
- Region 3: 2 possibilities - B or 0
- Region 4: 3 possibilities - A, B or 0

- Operators: 12 total possibilities

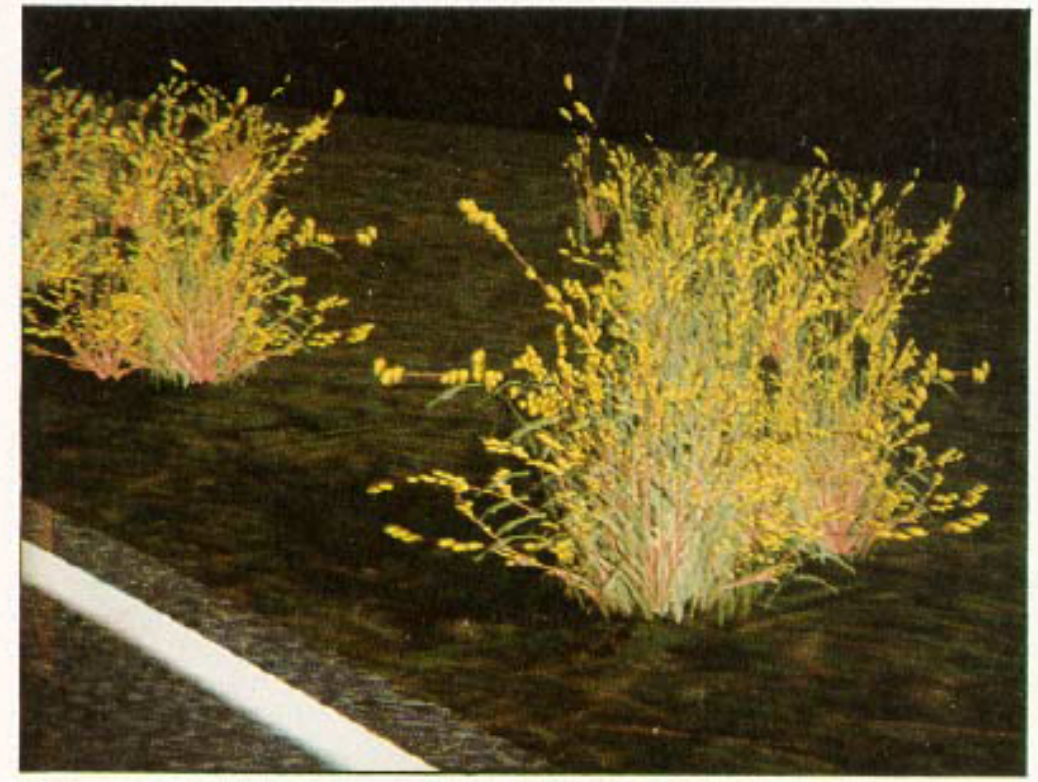
12 Operators

- $c_P = F_A c_A + F_B c_B$
- Various F functions dictate how to blend
- Always assumes color-associated values
- What commonly used operation does **in** do?

operation	quadruple	diagram	F_A	F_B
clear	(0,0,0,0)		0	0
A	(0,A,0,A)		1	0
B	(0,0,B,B)		0	1
A **over** B	(0,A,B,A)		1	$1-\alpha_A$
B **over** A	(0,A,B,B)		$1-\alpha_B$	1
A **in** B	(0,0,0,A)		α_B	0
B **in** A	(0,0,0,B)		0	α_A
A **out** B	(0,A,0,0)		$1-\alpha_B$	0
B **out** A	(0,0,B,0)		0	$1-\alpha_A$
A **atop** B	(0,0,B,A)		α_B	$1-\alpha_A$
B **atop** A	(0,A,0,B)		$1-\alpha_B$	α_A
A **xor** B	(0,A,B,0)		$1-\alpha_B$	$1-\alpha_A$

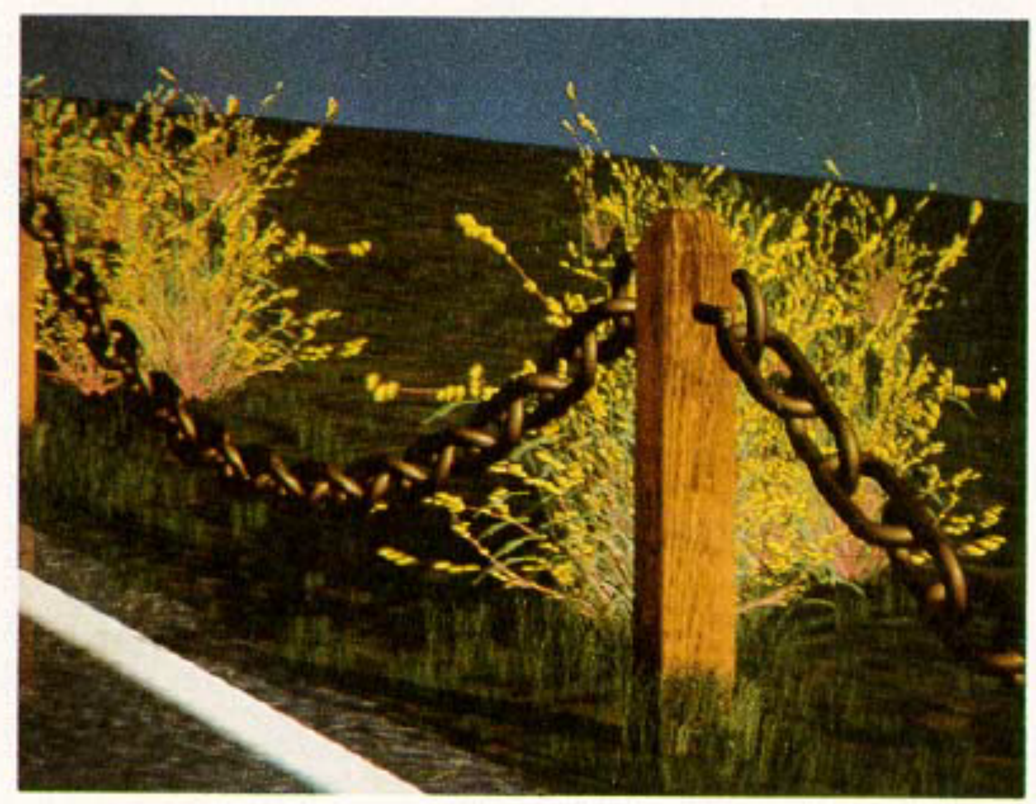
Porter & Duff, 1984

Foreground = *FrgdGrass* **over** *Rock* **over** *Fence* **over** *Shadow* **over** *BkgdGrass*;



Hillside = *Plant* **over** *GlossyRoad* **over** *Hill*;

Background = *Rainbow* **plus** *Darkbow* **over** *Mountains* **over** *Sky*;



Pt.Reyes = *Foreground* **over** *Hillside* **over** *Background*.

Porter & Duff, 1984

An Advanced Example

- Two-level operation, consider blending two adjacent pixels and then compositing (for example, if you "minified" two images)
- Blend P,Q = $\frac{1}{2}(p + q)$
- Blend R,S = $\frac{1}{2}(r + s)$
- Then **over**, $\frac{1}{2}(p + q)$ **over** $\frac{1}{2}(r + s)$

$$= \tfrac{1}{2}[(p+q) + (1-\boldsymbol{\alpha})(r+s)]$$

What if order switches?

- **over**, then blend
- ½[p **over** r + q **over** s]

$$= \tfrac{1}{2}[(p + (1-\boldsymbol{\alpha})r) + (q + (1-\boldsymbol{\alpha})s)]$$

$$= \tfrac{1}{2}[(p+q) + (1-\boldsymbol{\alpha})(r+s)]$$

=> The same!

Summary: Premultiplied Colors Help Make Compositing Operators Associative!

Without Pre-multiplication

- ½ $\boldsymbol{\alpha}$(P + Q) **over** ½$\boldsymbol{\alpha}_{RS}$(R + S)

  = ½[$\boldsymbol{\alpha}$(P + Q) + (1-$\boldsymbol{\alpha}$)$\boldsymbol{\alpha}_{RS}$(R + S)]

  => extra $\boldsymbol{\alpha}$ terms

Limitations of Relying on $\boldsymbol{\alpha}$ for Compositing

- Hard to represent certain types of transparency, like stained glass
  - Focus only on the subpixel occlusion (of all colors)
- Does not model more complex optical effects (magnifying glasses, lighting, etc.)
- Size of individual pixels

Cinefex

Lec07 Required Reading

- Brinkmann, sections from Ch.6 (see Blackboard)