# More on Colors

## Printing and Compositing

Brent M. Dingle, Ph.D.
Game Design and Development Program
Mathematics, Statistics and Computer Science
University of Wisconsin - Stout

2015

# Lecture Objectives

- Previously
  - What a Digital Image is
    - Acquisition
    - Human Perception
    - Representation
  - HTML5 and JavaScript Code Examples
    - Pixel manipulation
    - Image Loading
    - Filtering
  - Color Spaces
  - Image Manipulation
    - Filtering
    - Enhancement
    - Convolutions

- Today
  - Colors for Printing
  - Colors for Compositing

# Lecture Objectives

- Previously
  - What a Digital Image is
    - Acquisition
    - Human Perception
    - Representation
  - HTML5 and JavaScript Code Examples
    - Pixel manipulation
    - Image Loading
    - Filtering
  - Color Spaces
  - Image Manipulation
    - Filtering
    - Enhancement
    - Convolutions

- Today
  - Colors for Printing
  - Colors for Compositing

# Outline

- Color
  - Subtractive Color Spaces CMY and CMYK
- Compositing
  - Example operations
  - Associated Images
  - Bluescreening

# Color Review

- Tri-Stimulus Theory of Color
  - Additive Color Systems (light emitting)
    - RGB
    - HSV
    - CIE xyY
  - Subtractive Color Systems (pigment/light reflecting)
    - CMY
    - CMYK

# Additive Was Discussed Before

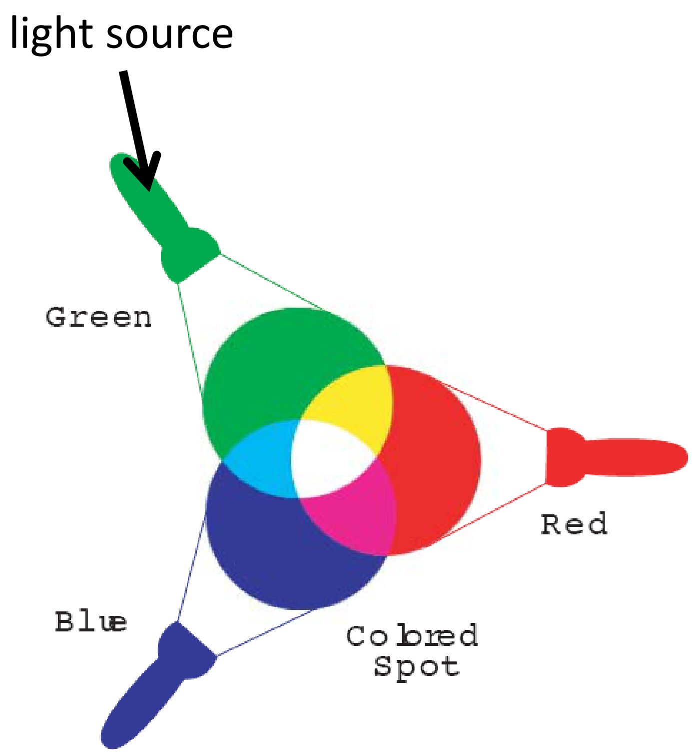
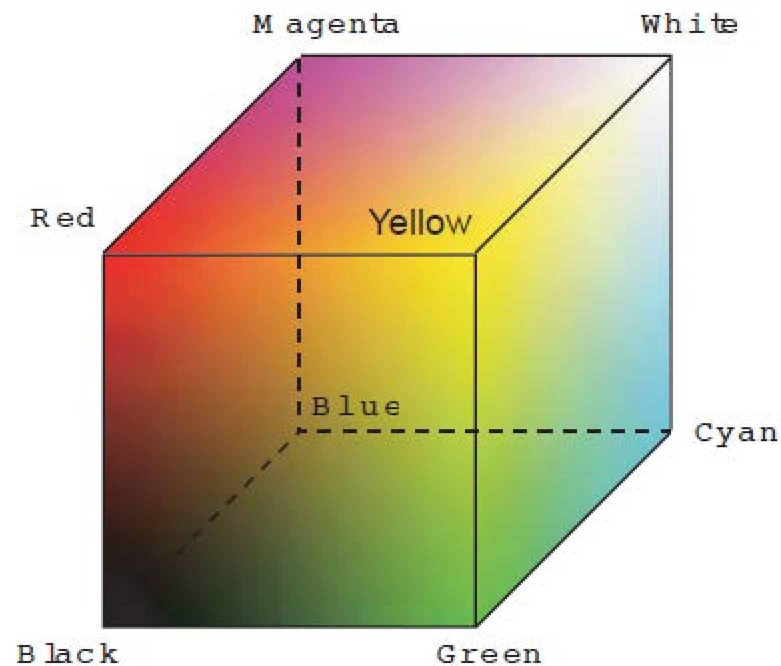



Figure 1.7: Additive Color Mixing for the Red-Green-Blue System




a) RGB

**Began with RGB then into HSV then into CIE xyY**
**→ skipped subtractive… saved it for now**

# Motivation

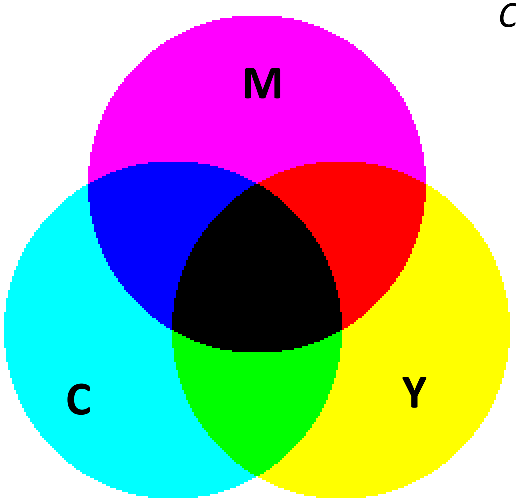
- We want to PRINT our nice color images
  - But RGB doesn’t seem to work
  - Not directly anyway

  - How do we solve this problem?

# Subtractive Color Systems

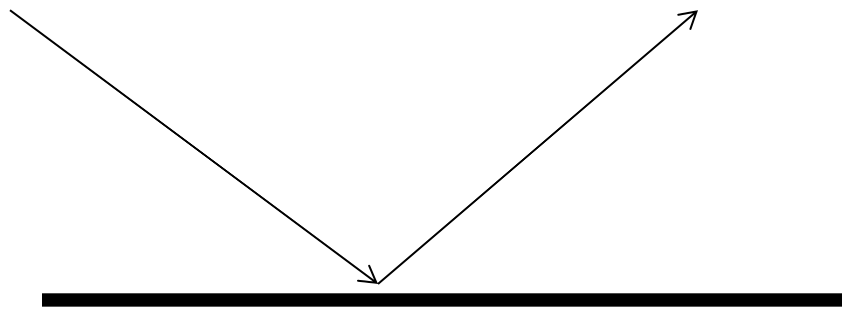
- CMY and CMYK (think printers/ink/pigment)

*Cyan, Magenta, and Yellow (and Black)*




light is additive
White = R + G + B

Yellow (R + G)



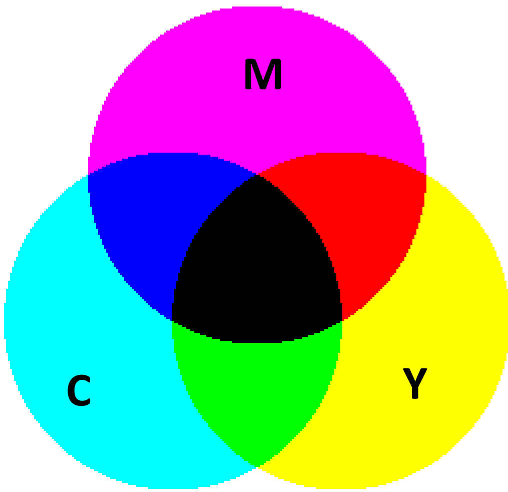
*Assume:*
*the surface here Absorbs all Blue light*
*(reflects no blue light, just red and green)*

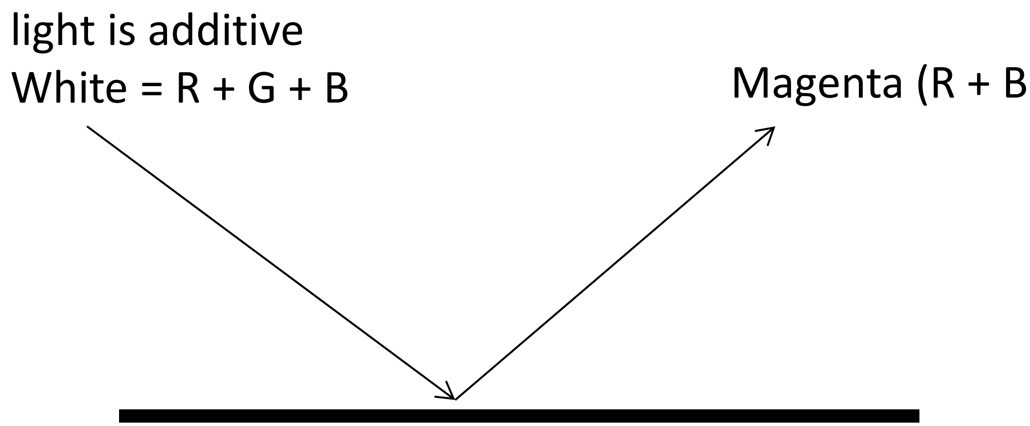
Y = white – blue
= 1 - B

Conclusion: A **surface** (pigment) that **appears yellow absorbs all blue light**

# Subtractive Color Systems

- Magenta



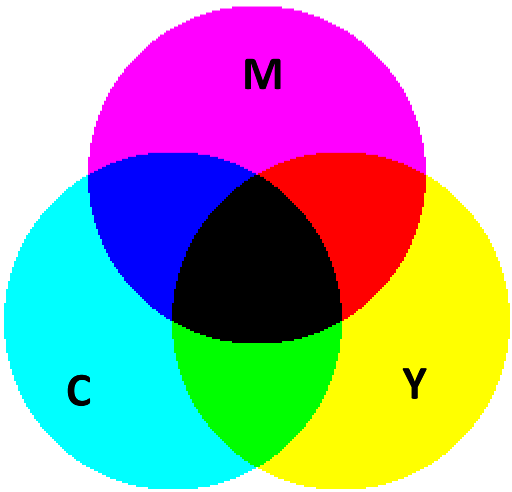

*Assume:*
*the surface here absorbs all GREEN light*
*(reflects only red and blue)*

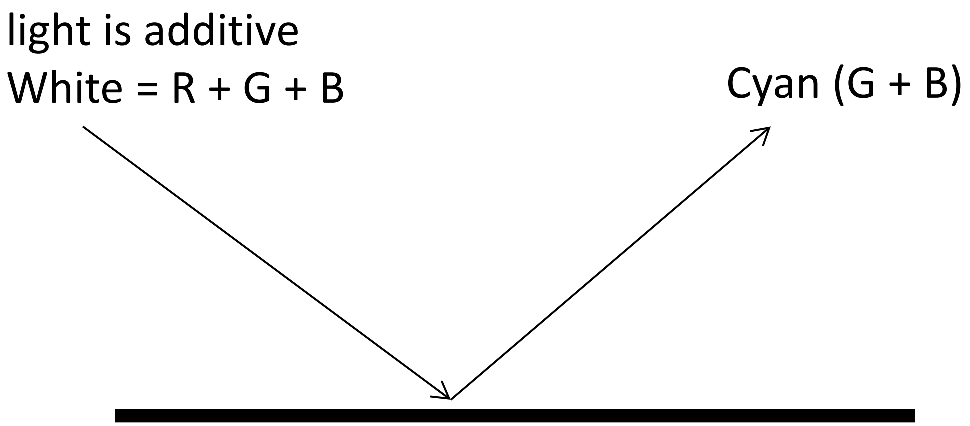
M = white – green
= 1 - G

Conclusion: A **surface** (pigment) that **appears magenta absorbs all green light**

# Subtractive Color Systems

- Cyan




*Assume:*
*the surface here absorbs all RED light*
*(reflects only green and blue)*

C = white – red
= 1 - R

Conclusion: A **surface** (pigment) that **appears Cyan absorbs all Red light**

# Given RGB calculate CMY

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Specifically assume we have R = 1, G = 1, and B = 0
Then the equivalent in CMY would be C = 0, M = 0, Y = 1

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

# Given RGB calculate CMY

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

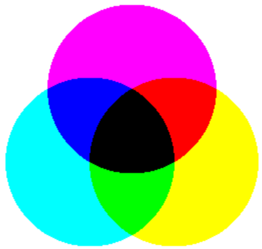
Specifically assume we have R = 1, G = 1, and B = 0
Then the equivalent in CMY would be C = 0, M = 0, Y = 1

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Also note

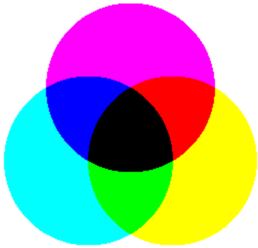
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

# CMY

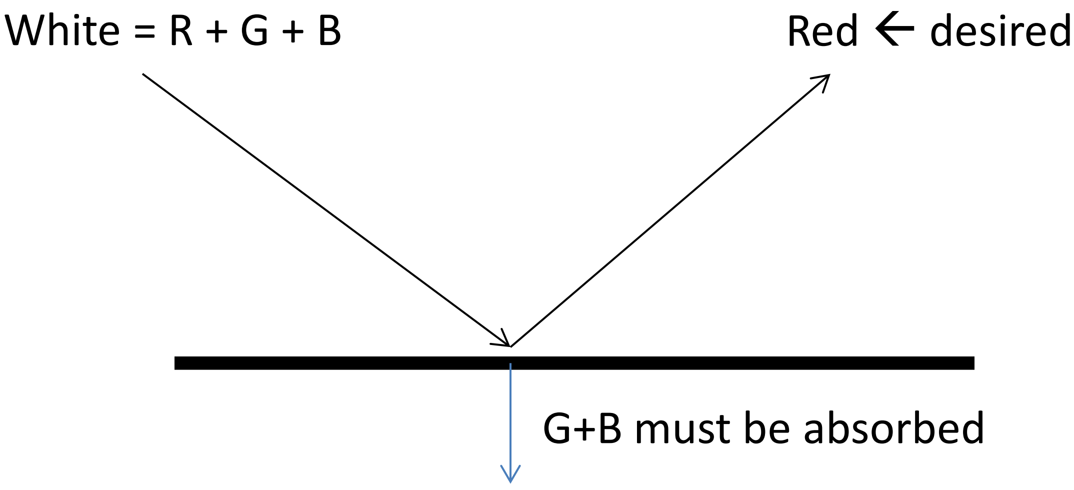


- How then do we get a Red color to print?
  - *i.e. we have an image with red in it, what CMY do we send to the printer to get it to be printed "correctly"*

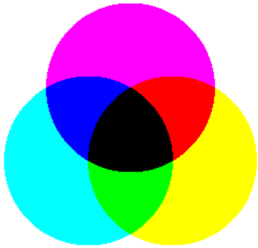
# CMY



- How then do we get a Red color to print?
  - *i.e. we have an image with red in it, what CMY do we send to the printer to get it to be printed "correctly"*

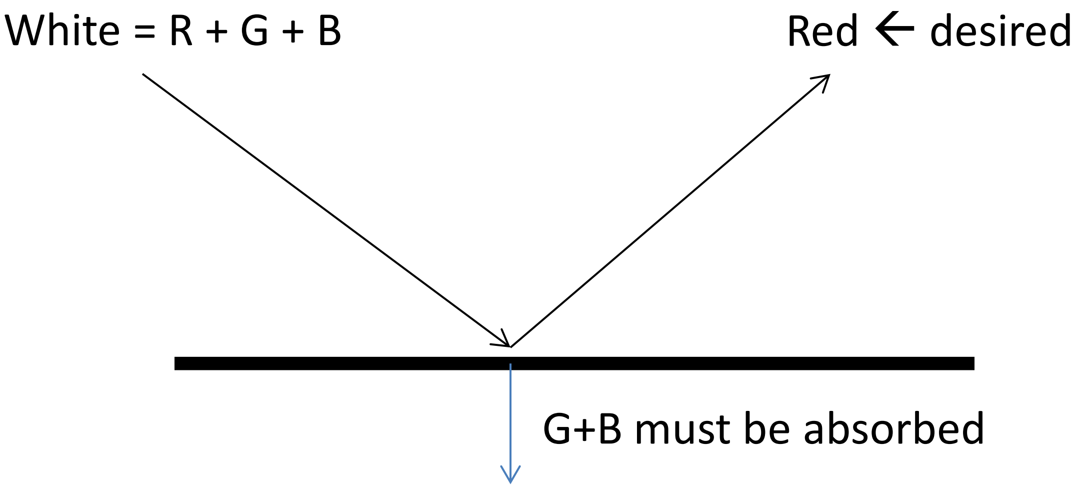


# CMY

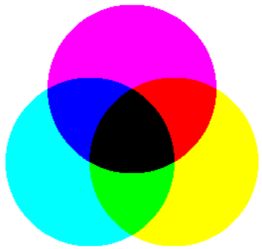


- How then do we get a Red color to print?
  - *i.e. we have an image with red in it, what CMY do we send to the printer to get it to be printed "correctly"*

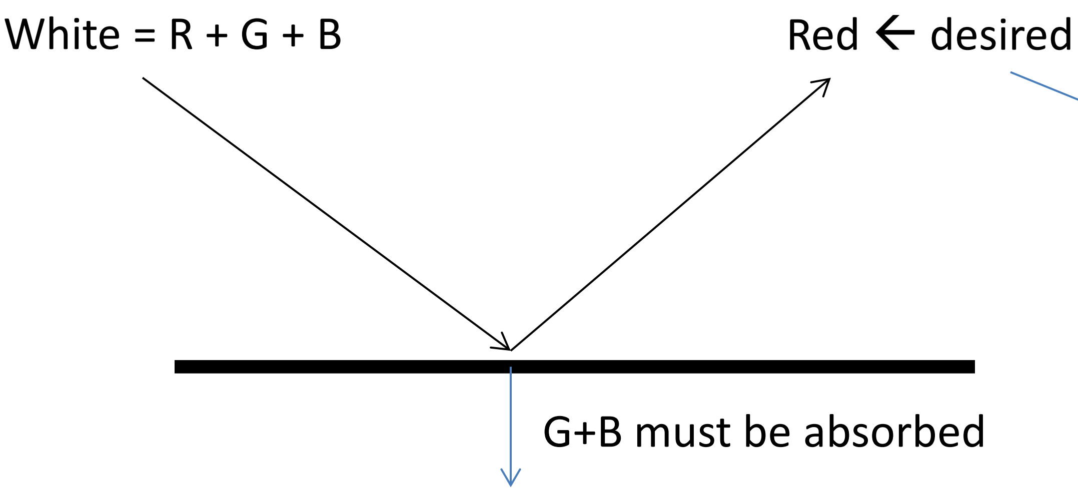
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$



# CMY



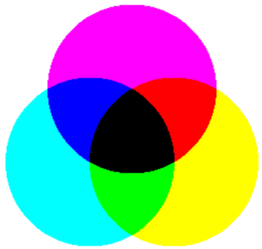
- How then do we get a Red color to print?
  - *i.e. we have an image with red in it, what CMY do we send to the printer to get it to be printed "correctly"*



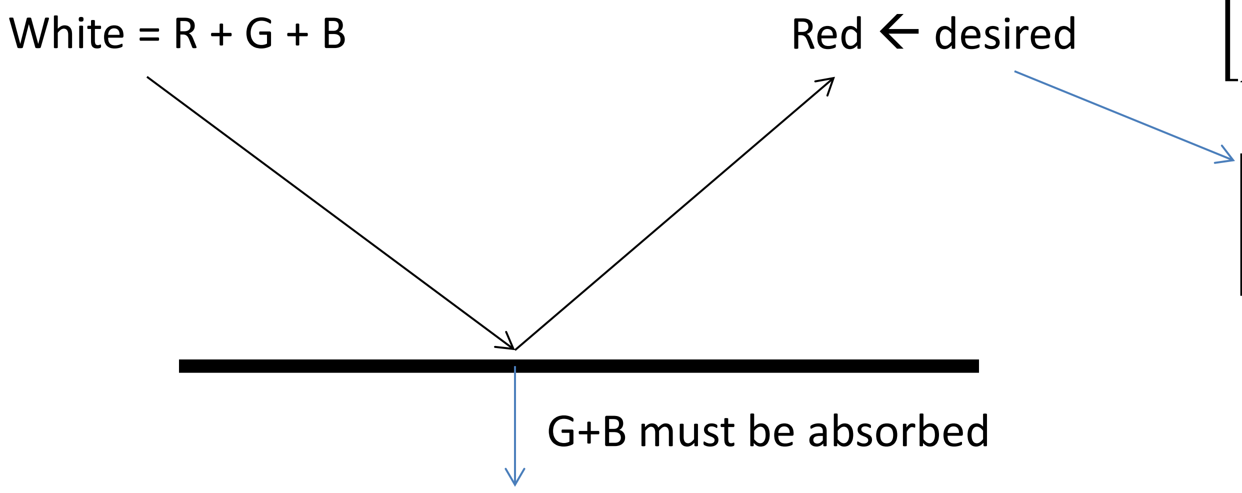

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

# CMY



- How then do we get a Red color to print?
  - *i.e. we have an image with red in it, what CMY do we send to the printer to get it to be printed "correctly"*

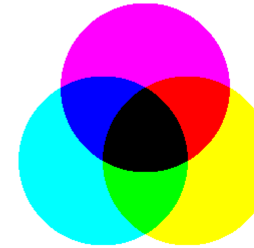



$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

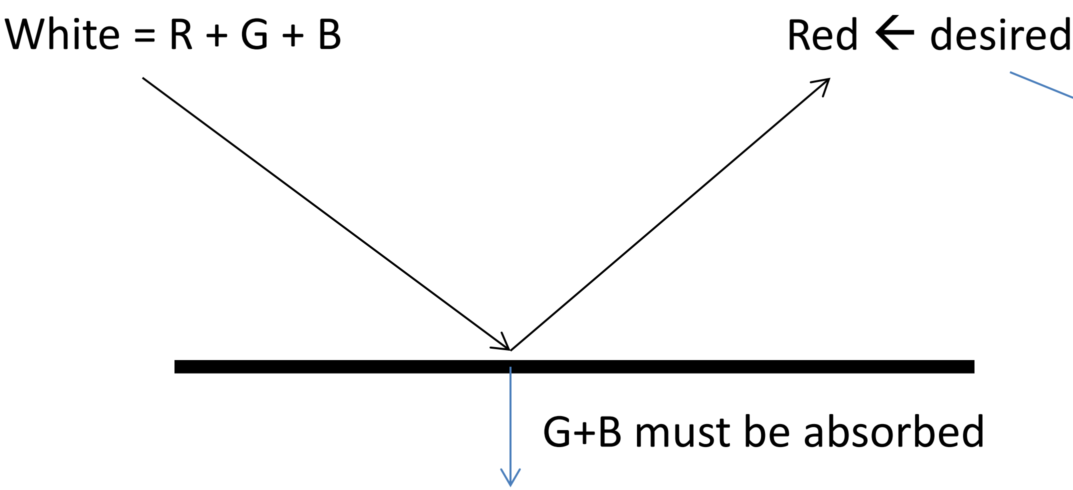
$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

# CMY



- How then do we get a Red color to print?
  - *i.e. we have an image with red in it, what CMY do we send to the printer to get it to be printed "correctly"*




$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

Solution:

Use M and Y ink
to absorb G and B

i.e. Mix M and Y ink to get Red

# CMY**K**

- How do we get BLACK to print?

# CMY**K**

- How do we get BLACK to print?

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \longrightarrow \begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

In theory this is true
But in practice it only makes a dark grey

and so in practice a black ink is used
→ CMYK, where K is the measure of black ink

*Why? the CMY inks are not pure enough to make black*
*It is too expensive to make them. Black in is cheaper to make.*

# CMY**K**

- Common practice for calculating K
  - First calculate the CMY
  - Identity which is smallest in value and set K to be that minimum
  - Then adjust to C'M'Y'

K = min(C, M, Y)

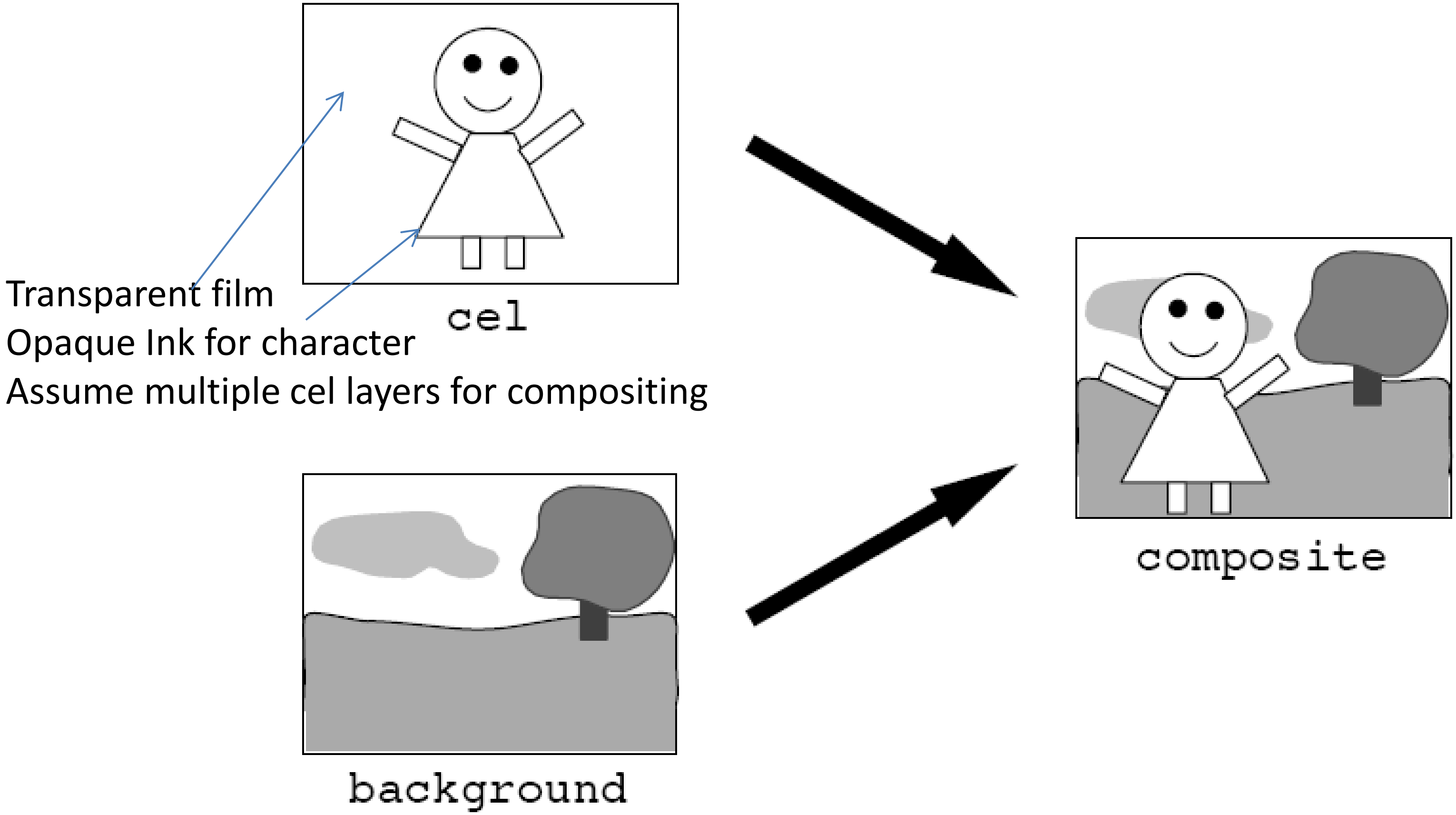
$$\begin{bmatrix} C' \\ M' \\ Y' \end{bmatrix} = \begin{bmatrix} C \\ M \\ Y \end{bmatrix} - \begin{bmatrix} K \\ K \\ K \end{bmatrix}$$

And printing is done with 4 inks
in the amounts of C' M' Y' and K

# Outline

- Color
  - Subtractive Color Spaces CMY and CMYK
- Compositing
  - Example operations
  - Associated Color Images
  - Bluescreening

# Compositing




Single background for entire animated sequence

Examples from: Donald House, Texas A&M University, 1999

# Use of Compositing

- Film Special Effects
- Morphing (turning one image into another)
- Integrating animated images with live action film
- **… but how/why can it be done digitally?**

# Alpha

- As already discussed we typically store an ALPHA value along with our RGB

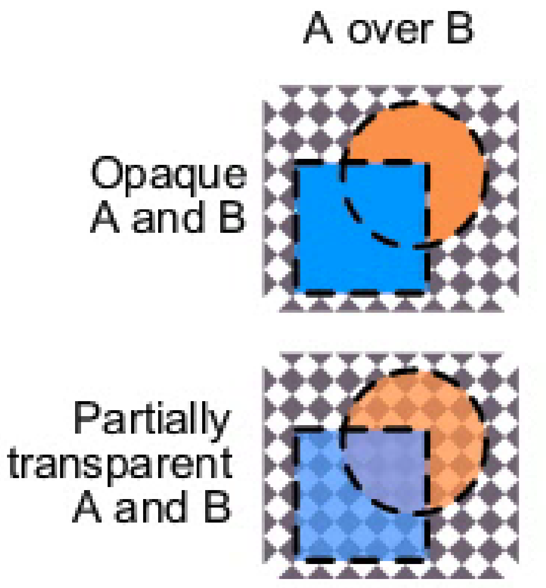
- Alpha is the pixel’s opacity
  - Typically on a [0, 1] scale
    - 0 means completely transparent
    - 1 means fully opaque

# Over Operator

- The OVER operator takes two images as input and composites them together
  - Foreground Image = F
  - Background Image = G
  - Resulting Composite Image = P

$$P = F \ \textbf{over} \ G$$

This is a pixel by pixel operation



# OVER operator: definition

- Assume F is transparent and G opaque

$$C_P = \alpha_F C_F + (1 - \alpha_F) C_G$$

$C_P$ is composite color channel value

$\alpha_F$ is the Foreground pixel alpha

$C_F$ is the Foreground pixel color channel value

$C_G$ is the backGround pixel color channel value

color channel = R or G or B
equation is same for each

# Associativity Challenge

- The process is repeatable
  - Composite each foreground image into the background image, one at a time
- BUT
  - It is desirable to composite multiple foreground images together then composite the result with a background
    - i.e. the OVER operator needs to be associative
      - can group the compositing operations in any way and still get the same result
        - » as with addition: (A+B)+C = A+(B+C)

# Associative: Explicit Example

How to define **over** operator such that the following is true

$$A\,\mathbf{over}\,(B\,\mathbf{over}\,G) = (A\,\mathbf{over}\,B)\,\mathbf{over}\,G$$

# Tasks at Hand

- Reformulate **over** operator to *use 2 alpha values* in the foreground images *to composite the color values* into the intermediate foreground image, and

- extend the **over** operator to *combine alpha values* from two foreground images *to provide a single alpha value* for the intermediate image

# Case of: A over B

Solution for
Combined alpha is

$$\alpha_H = \alpha_A + (1 - \alpha_A)\alpha_B$$

Solution for
Color Channel is

$$C_H = \frac{\alpha_A}{\alpha_H} C_A + \frac{(1-\alpha_A)\alpha_B}{\alpha_H} C_B$$

*ASIDE:*

*While designed for cases where both A and B are transparent*
*this will work for all cases of A and B alpha values*

*For example, say alpha of image A is 0.5 and B is fully opaque*
*The resulting alpha of combined images is 0.5 + (1-0.5)*0.25 = 0.625*

*Finish this and observe the resulting $C_H$ is as $C_P$ was in the earlier 'simple case' definition*

# Subtle Detail

- If we ASSUME each pixel color value is given to us as pre-multiplied with its associative value
  - Then some simplifications can occur

$$C_H = \frac{\alpha_A}{\alpha_H} C_A + \frac{(1 - \alpha_A)\alpha_B}{\alpha_H} C_B$$

# Subtle Detail

- If we ASSUME each pixel color value is given to us as pre-multiplied with its associative value
  - Then some simplifications can occur

$$C_H = \frac{\alpha_A}{\alpha_H} C_A + \frac{(1 - \alpha_A)\alpha_B}{\alpha_H} C_B$$

$$\alpha_H C_H = \alpha_A C_A + (1 - \alpha_A)\alpha_B C_B$$

# Subtle Detail

- If we ASSUME each pixel color value is given to us as pre-multiplied with its associative value
  - Then some simplifications can occur

$$C_H = \frac{\alpha_A}{\alpha_H} C_A + \frac{(1 - \alpha_A)\alpha_B}{\alpha_H} C_B$$

$$\alpha_H C_H = \alpha_A C_A + (1 - \alpha_A)\alpha_B C_B$$

$$Assume\text{: } C_{\hat{H}} = \alpha_H C_H$$

$$Assume\text{: } C_{\hat{A}} = \alpha_A C_A$$

$$Assume\text{: } C_{\hat{B}} = \alpha_B C_B$$

$$C_{\hat{H}} = C_{\hat{A}} + (1 - \alpha_{\hat{A}})\, C_{\hat{B}}$$

# OVER: Nice, Simple, Definition

$$C_{\hat{H}} = C_{\hat{A}} + (1 - \alpha_{\hat{A}})\, C_{\hat{B}}$$

***Important:*** Many images may have an alpha value.
This does NOT always mean
the alpha values have been pre-applied to their color channels (RGB)
--> context of where the image came from may help

So, if you use the above equation, you may need to apply the alpha values to your initial images (i.e. perform the pre-multiplication on the initial images before you begin your compositing)

# Outline

- Color
  - Subtractive Color Spaces CMY and CMYK
- Compositing
  - Example operations
  - Associated Color Images
  - Bluescreening

# Associated Color Images

- When an image is stored with associated color pixels it is called an ***Associated Color Image***

Associated color:

Instead of storing color and alpha in the frame buffer as

(**1**, **0**, **0**, 0.5)

We store them as

(**0.5**, **0**, **0**, 0.5)

or colors pre-multiplied with alpha

# Why Use Associated Color Images?

- Simplifies and generalizes
- Useful in compositing
- It makes the associative law work for image operations

***IMPORTANT:***

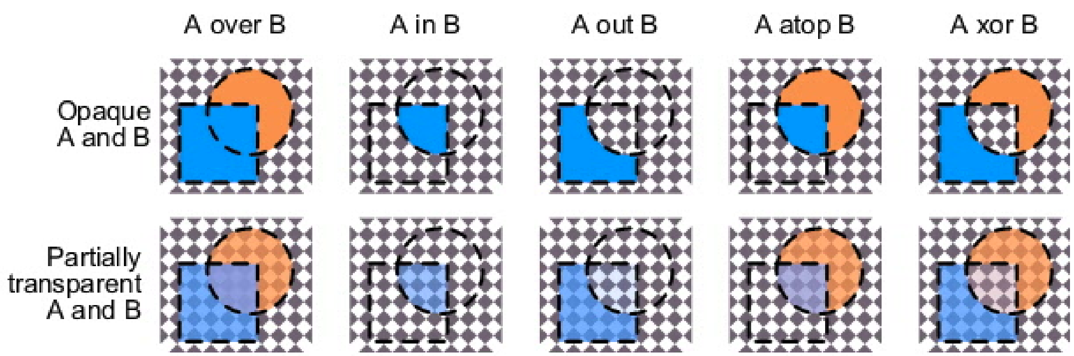
***Most file formats do NOT do this***

***BUT***

***It might be useful to do this immediately on images if you plan on doing operations such as compositing***

# Other Compositing Operations

| image operation | per pixel operation |
|---|---|
| A over B | $C_{\hat{H}} = C_{\hat{A}} + (1 - \alpha_{\hat{A}})\, C_{\hat{B}}$ |
| A in B | $C_{\hat{H}} = \alpha_{\hat{B}} C_{\hat{A}}$ |
| A out B | $C_{\hat{H}} = (1 - \alpha_{\hat{B}})\, C_{\hat{A}}$ |
| A atop B | $C_{\hat{H}} = \alpha_{\hat{B}} C_{\hat{A}} + (1 - \alpha_{\hat{A}})\, C_{\hat{B}}$ |
| A xor B | $C_{\hat{H}} = (1 - \alpha_{\hat{B}}) C_{\hat{A}} + (1 - \alpha_{\hat{A}})\, C_{\hat{B}}$ |




Examples from: Donald House, Texas A&M University, 1999

# Outline

- Color
  - Subtractive Color Spaces CMY and CMYK
- Compositing
  - Example operations
  - Associated Color Images
  - Bluescreening

# Bluescreening

- Bluescreening Filmmaking technique
  - Uses an evenly-lit monochromatic background for the purpose of replacing it with a different image or scene
    - Invented by Petro Vlahos
      - Technical Academy Award 1964

- Related terms
  - Luma keying
  - Chroma keying

# Algorithm

Capture the foreground image against a blue background

Image from advertisement at: http://www.telepresencecatalog.com/draper-vc/

# Algorithm

Capture the foreground image against a blue background

Set the $\alpha = 0$ for the blue pixels

Image from advertisement at: http://www.telepresencecatalog.com/draper-vc/

# Algorithm

Capture the foreground image against a blue background

Set the $\alpha = 0$ for the blue pixels

Set the $\alpha = 1$ otherwise

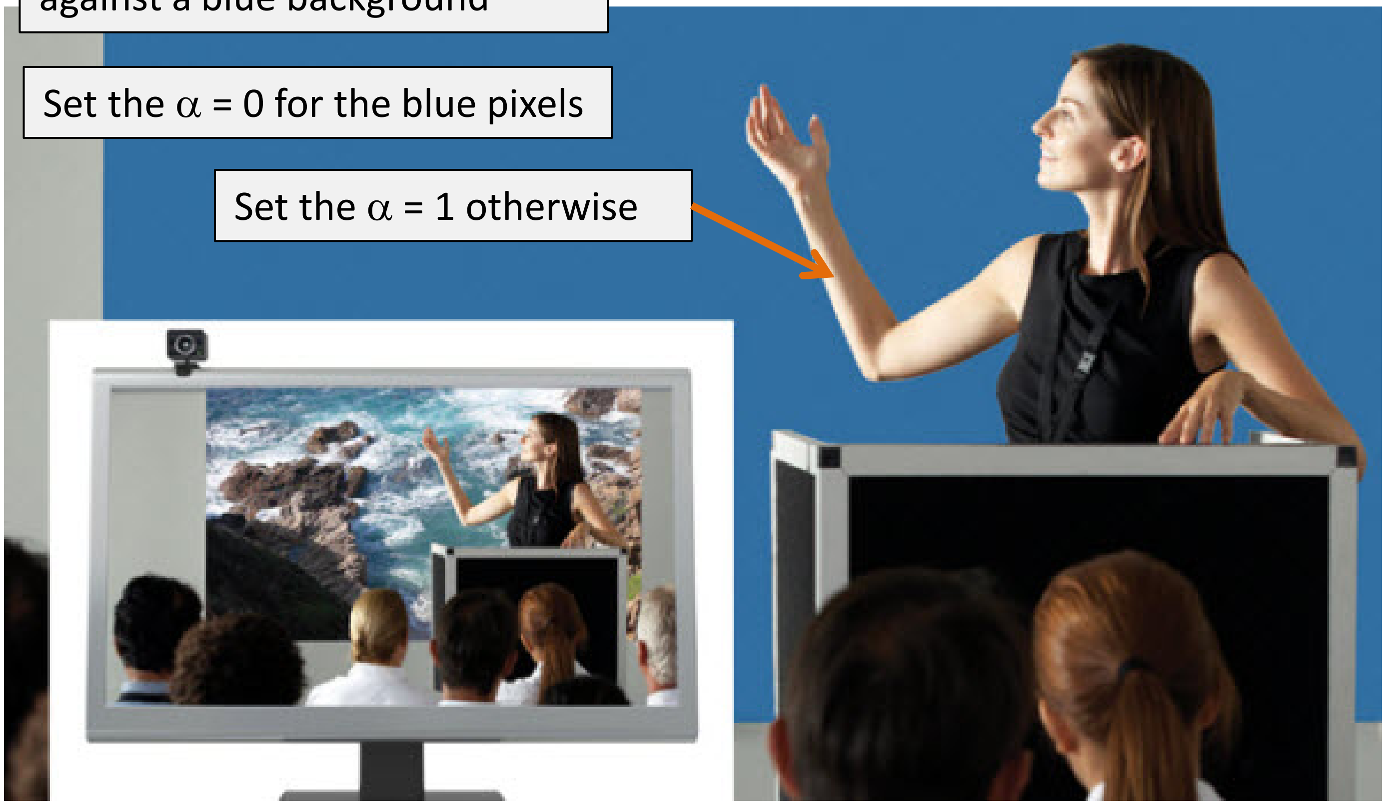


Image from advertisement at: http://www.telepresencecatalog.com/draper-vc/

# Algorithm

Capture the foreground image against a blue background

Set the $\alpha = 0$ for the blue pixels

Set the $\alpha = 1$ otherwise

Composite with desired background

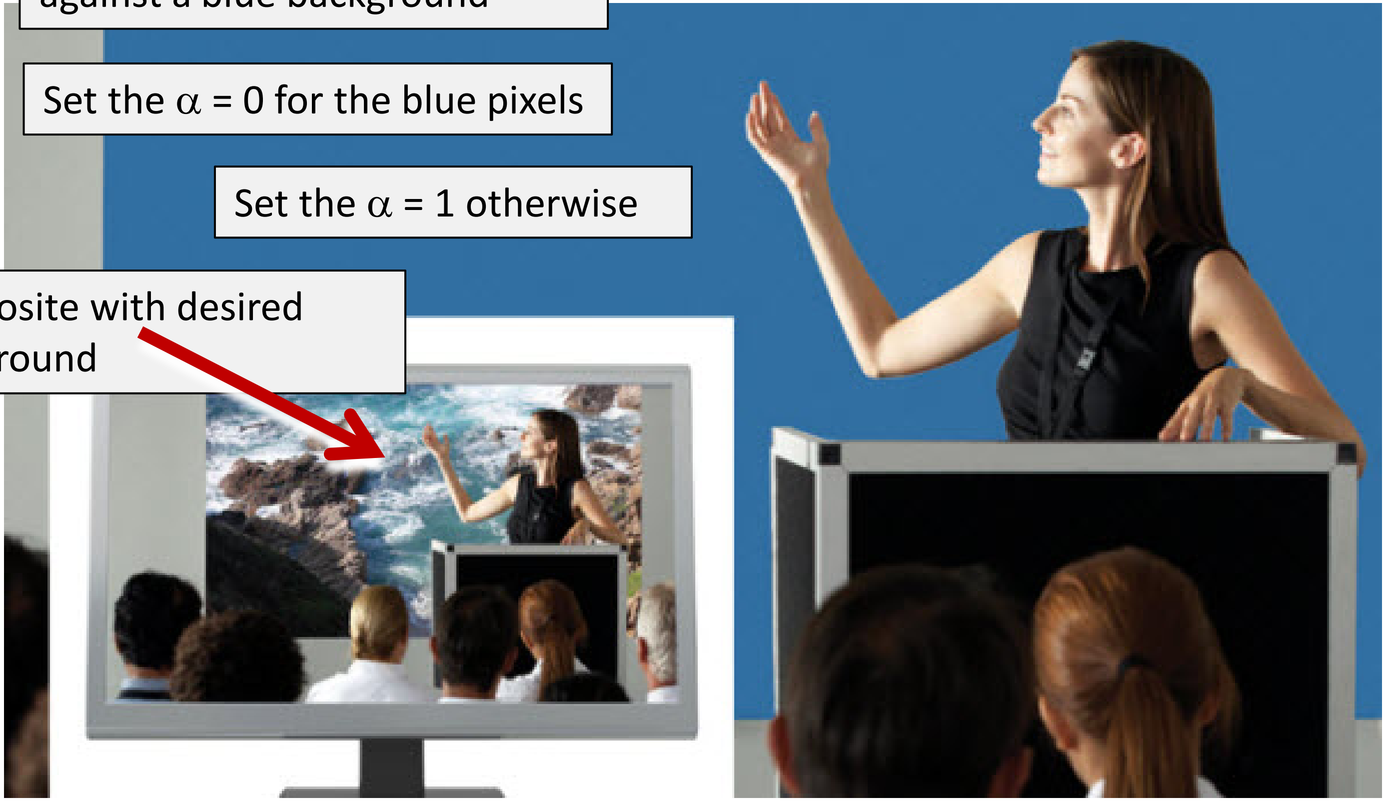


Image from advertisement at: http://www.telepresencecatalog.com/draper-vc/

# Bluescreening Algorithm

$$C = \alpha \times F + (1 - \alpha) \times B$$

unassociated color images, and × means times

$$\hat{C} = \hat{F} + (1 - \alpha) \times \hat{B}$$

if using associated color images

# Blue Screening: An Issue

- Keep the edges looking smooth in composite
- How?

| | | | |
|---|---|---|---|
| **1** | **0.7** | **0.2** | **0** |
| **1** | **1** | **0.8** | **0** |
| **1** | **1** | **1** | **0.4** |
| **1** | **1** | **1** | **0.8** |

alpha channel

Along edges of the alpha mask we use

$$0 < \alpha < 1$$

to reduce jaggies

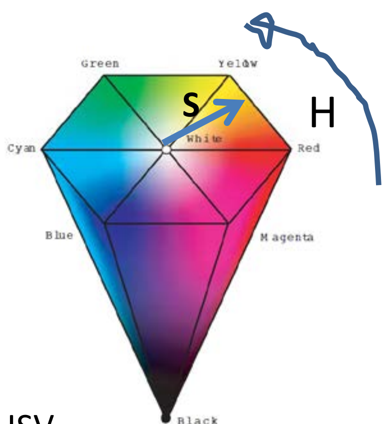
# Creating the Alpha Mask

- Keying Using Chroma (H) and Luma (V)

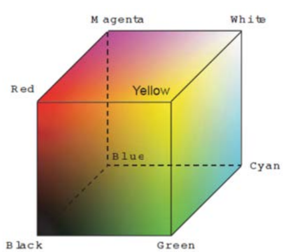
First: Convert RGB to HSV

Recall

$$H \in [0, 360]$$
$$S, V, R, G, B \in [0, 1]$$




HSV




RGB

Let

MAX = maximum(R, G, B);
MIN = minimum(R, G, B);

S and V are straightforward

$$S = \begin{cases} 0, & \text{if } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{otherwise} \end{cases}$$

$$V = MAX$$

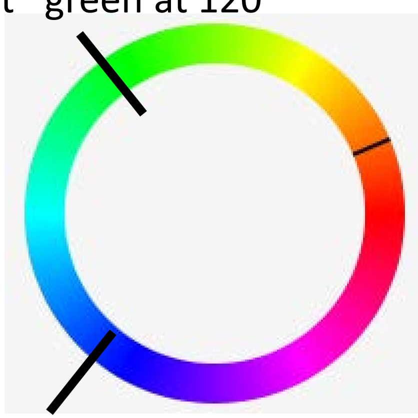
# Creating the Alpha Mask

- Keying Using Chroma (H) and Luma (V)

First: Convert RGB to HSV → now the H

Recall

$$H \in [0, 360]$$
$$S, V, R, G, B \in [0, 1]$$




Let

MAX = maximum(R, G, B);
MIN = minimum(R, G, B);

$$H = \begin{cases} \text{undefined}, & \text{if } MAX = MIN \\ 60 \times \frac{G-B}{MAX-MIN} + 0, & \text{if } MAX = R \\ & \text{and } G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & \text{if } MAX = R \\ & \text{and } G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{if } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{if } MAX = B \end{cases}$$

# Creating the Alpha Mask

Example H:

If MAX = B, then H is Blue (240 degrees).
If R=G, then it is perfect Blue (240)
else must move towards green or red whichever is larger
(bound the move to 60 degrees or less)

$$H \in [0, 360]$$
$$S, V, R, G, B \in [0, 1]$$

MAX = maximum(R, G, B);
MIN = minimum(R, G, B);

$$H = \begin{cases} \text{undefined}, & \text{if } MAX = MIN \\ 60 \times \frac{G-B}{MAX-MIN} + 0, & \text{if } MAX = R \\ & \text{and } G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & \text{if } MAX = R \\ & \text{and } G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{if } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{if } MAX = B \end{cases}$$

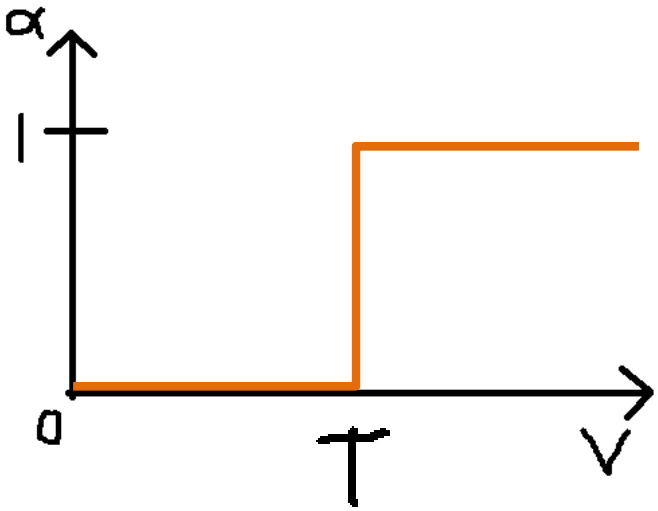
# Option: Luma Keying

- Use Value (V) in HSV for keying
  - works for well lit scene and dark background

Method 1

*(may cause jaggies)*

$$\alpha = \begin{cases} 1, & V > T \\ 0, & else \end{cases}$$

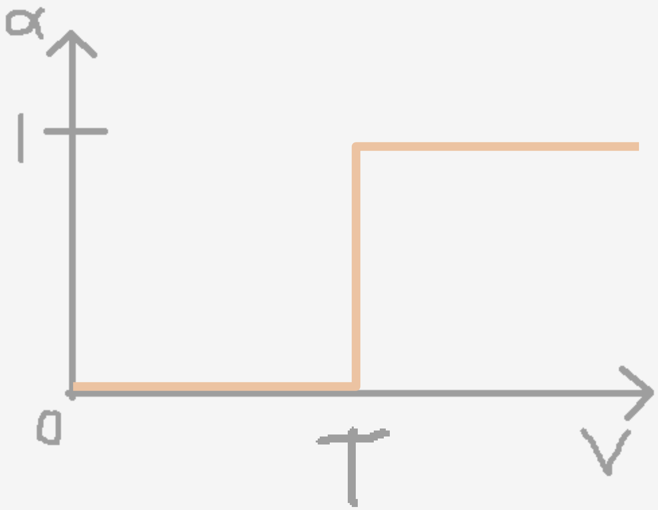


# Option: Luma Keying

- Use Value (V) in HSV for keying
  - works for well lit scene and dark background

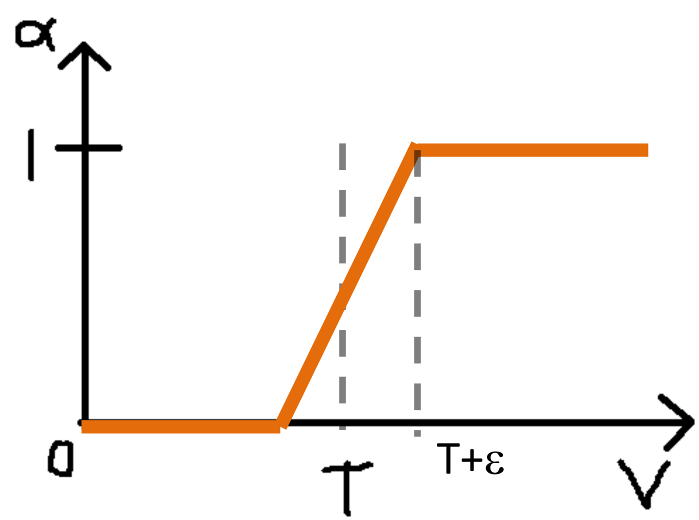
Method 1
*(may cause jaggies)*

$$\alpha = \begin{cases} 1, & V > T \\ 0, & else \end{cases}$$




Method 2

$$\alpha = \begin{cases} 1 & V > T + \varepsilon \\ \frac{V-(T-\varepsilon)}{2\varepsilon} & T - \varepsilon < V \leq T + \varepsilon \\ 0 & else \end{cases}$$



# More Problems: Blue Spilling

- Color Spill
  - Describes areas of the foreground subject that have absorbed or reflected color from the background
  - Commonly happens during shooting and is fixed in post-shooting


*Source: The Art & Science of Digital Compositing, by Ron Brinkmann*

# Reflection Blue Spill

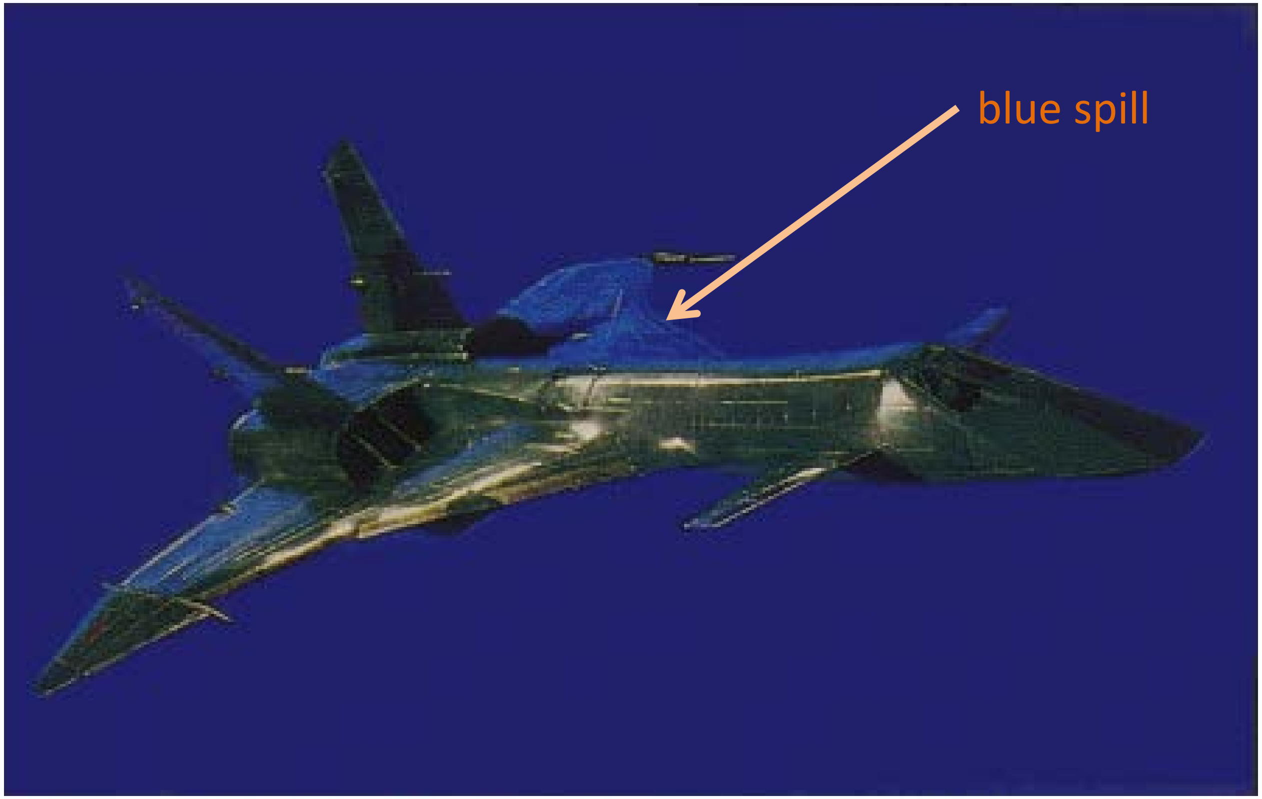



***Figure 3.*** *Firefox Blue Spill Matte Series 1, original shot. Note blue reflected on wing surfaces from bluescreen -- undesirable but unavoidable on such surfaces.*

http://www.digitalgreenscreen.com/figure3.html

# Suppression of Spill

- Modify the blue channel on ***associated*** color image

if ( B > G) then B = G
this will set B=0 on bluescreen and reduces blue otherwise

Problem?
a blue shirt can become green
if this is not done carefully

Can also try
if (B > R) then B = R

*Trick is to do this on the associated color image made from the foreground input image*
*Do not do this to the original input image – can try and see what problems occur =)*

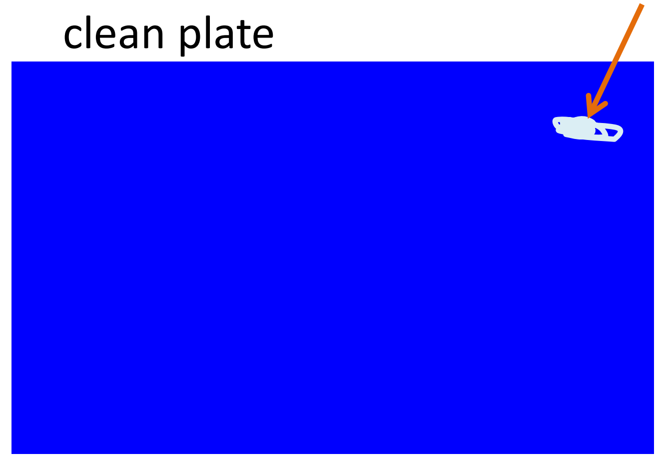
# Difference Matte

- Can be used for non-even blue screens or natural scene screens

non-even screen
overemphasized

clean plate

**A**



**B**

$$\alpha = \begin{cases} 1, & |A - B| > \varepsilon \\ 0, & else \end{cases}$$

# Garbage Matte

- Garbage matte
  - manual method to exclude parts of an image that a bluescreen would not remove
    - or include parts it did remove
      - such as reflective blue spill

For example:

Exclude parts outside of the dashed lasso region

# Challenge

- Composite your own set of images
  - i.e. Try doing this


$$C = \alpha \times F + (1 - \alpha) \times B$$

unassociated color images, and × means times

Suggest input of: 1 background image,

and 2 or more images taken in front of a bluescreen

(or green screen if easier)

output 2 or more composited images

# Trivia

- Bluescreening was used “long” ago as blue was useful to film

- As movies have become more “digital” Green screens are more common
  - BECAUSE
    - image sensors in digital video cameras are most sensitive to green
      - ***Bayer pattern*** allocates more pixels to the green channel
    - Green channel thus has least amount of noise for cleanest mask
    - Also requires less light because hardware sensors are more sensitive to green light

# Extra Research

- Compositing Digital Images SIGGRAPH paper by Porter and Duff, 1984
  - http://keithp.com/~keithp/porterduff/

- Look into Shapes from Silhouettes
  - Good presentation at:
    - http://www.sci.utah.edu/~gerig/CS6320-S2015/Materials/CS6320-S2015-Shape-from-Silhouttes-I.pdf

- Investigate Active Contour Models
  - Computer Vision
  - Relates to Blue Screening (Chroma Keying)

- Other (maybe) useful papers
  - https://www.cs.princeton.edu/courses/archive/fall00/cs426/papers/smith95a.pdf
  - Smith, A. R., AND Blinn, J. F. 1996. Blue screen matting.
    In Proceedings of ACM SIGGRAPH 1996, 259–268.
  - A. Blake & M. Isard (1998). Active Contours. Springer-Verlag.
  - M. Kass, A. Witkin & D. Terzopoulos (1988). Snakes: active contour models.
    In International Journal of Computer Vision (1988), pp. 321-331.

# Summary: Color and Compositing

- Colors for Printing
  - Subtractive Color Spaces CMY and CMYK

- Colors for Compositing
  - Example operations
  - Associated Images
  - Bluescreening

# Questions?

- Beyond D2L
  - Examples and information can be found online at:
    - *http://docdingle.com/teaching/cs.html*

    - *Continue to more stuff as needed*

# Extra Reference Stuff Follows

# Credits

- Much of the content derived/based on slides for use with the book:
  - *Digital Image Processing,* Gonzalez and Woods

- Some layout and presentation style derived/based on presentations by
  - Donald House, Texas A&M University, 1999
  - Bernd Girod, Stanford University, 2007
  - Shreekanth Mandayam, Rowan University, 2009
  - Igor Aizenberg, TAMUT, 2013
  - Xin Li, WVU, 2014
  - George Wolberg, City College of New York, 2015
  - Yao Wang and Zhu Liu, NYU-Poly, 2015
  - Sinisa Todorovic, Oregon State, 2015