CSC 433/533 Computer Graphics

Alon Efrat
Credit: Joshua Levine

Lecture 07 Image Compositing

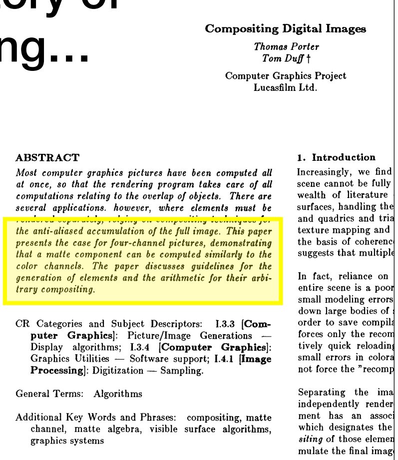
Today's Agenda

- Reminders:
  - A02 questions?
- Goals for today:
  - Discuss image compositing, alpha channel, Porter + Duff arithmetic, and green screening

Image Compositing

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

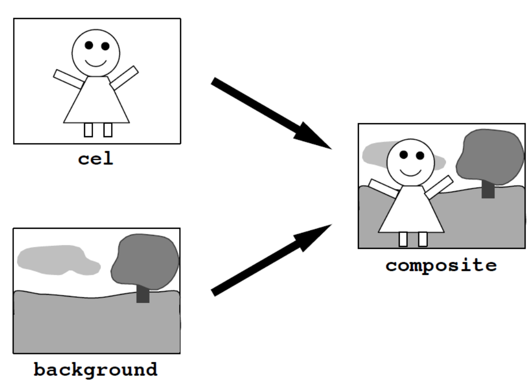
Increasingly, we find
scene cannot be fully
wealth of literature
surfaces, handling the
and quadrics and tria
texture mapping and
the basis of coherenc
suggests that multiple

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

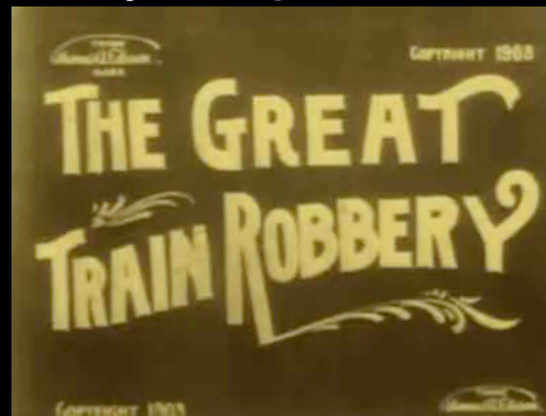
- Cels: Transparent sheets with (hand) drawings on them. (from "celluloid")



Animation Cels

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

Early Example from Film



http://archive.org/details/TheGreatTrainRobbery_555

More Film Examples

Forrest Gump

Even More Film Examples

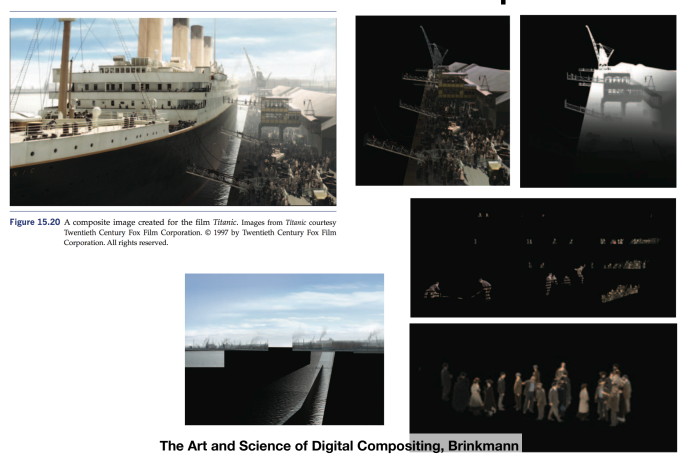
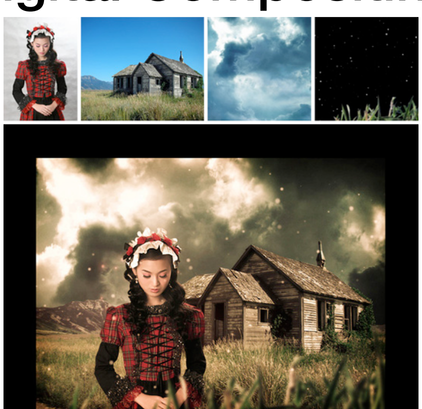


Figure 15.20 A composite image created for the film *Titanic*. Images from *Titanic* courtesy Twentieth Century Fox Film Corporation. © 1997 by Twentieth Century Fox Film Corporation. All rights reserved.

The Art and Science of Digital Compositing, Brinkmann

Digital Compositing

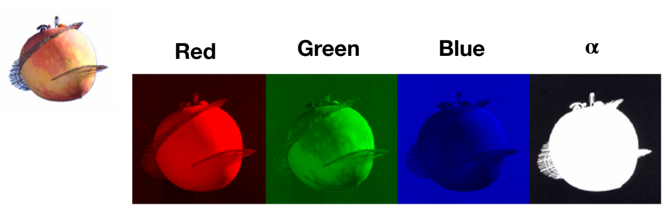


http://www.hongkiat.com/blog/30-nicest-photoshop-photo-effects-part-ii/

Four Channel Images

What is α (alpha)?

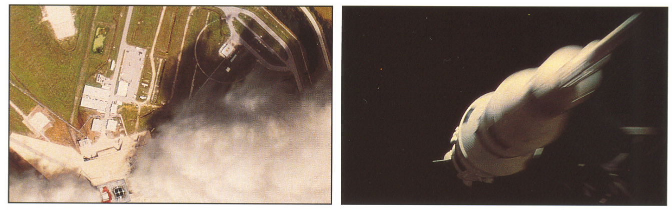
- α is a measurement of the opacity.
- α=1 means fully opaque.
- α=0 means fully transparent.



The Art and Science of Digital Compositing, Brinkmann

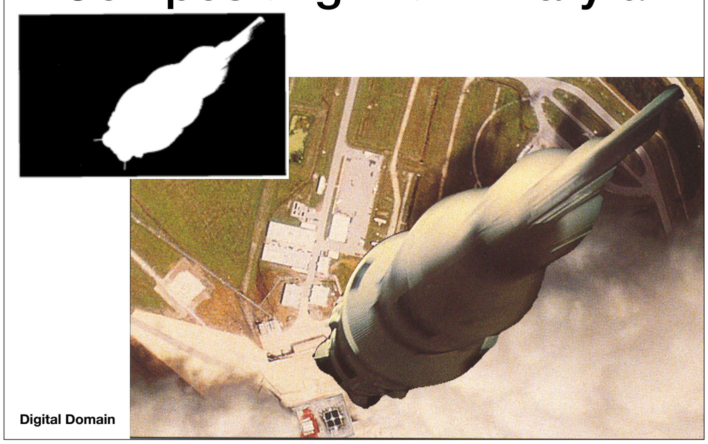
Interpreting Fractional α

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



Digital Domain, Apollo 13 (1995)

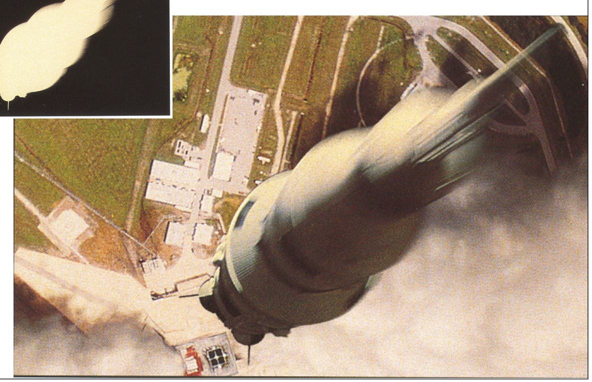
Compositing with Binary α



Digital Domain

Compositing with Fractional α

Compare at:
https://youtu.be/1-JdqHxqkHA?t=4m10s



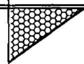
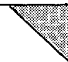
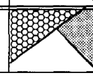
Digital Domain

Arithmetic of Compositing

over Operator

- Extension of the "Painter's" 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 $\alpha_A < 1$, view penetrates (partially) through A

Porter & Duff, 1984

over Operator Defined

- Each color channel C treated separated:

$$C_P = \alpha_A C_A + (1-\alpha_A) C_B$$

- Linear interpolation of colors, assumes that $0 < \alpha, C < 1$
- More generally (if background has transparency):

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

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

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 => $\alpha_B C_B + (1-\alpha_B) C_D$
- $C_P = \alpha_A C_A + (1-\alpha_A)[\alpha_B C_B + (1-\alpha_B) C_D]$
  $C_P = \alpha_A C_A + (1-\alpha_A)\alpha_B C_B + (1-\alpha_A)(1-\alpha_B) C_D$

(A over B) over D

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

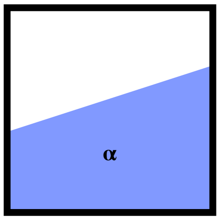
$$C_P = \boxed{\alpha_A C_A + (1-\alpha_A)\alpha_B C_B} + \boxed{(1-\alpha_A)(1-\alpha_B) C_D}$$

- If it does, it must be that:
  1. $\alpha_H C_H = \alpha_A C_A + (1-\alpha_A)\alpha_B C_B$
  2. $(1-\alpha_H) = (1-\alpha_A)(1-\alpha_B)$ => $\alpha_H = \alpha_A + (1 - \alpha_A)\alpha_B$
- If we solve for $C_H = (\alpha_A/\alpha_H) C_A + (1-\alpha_A)(\alpha_B/\alpha_H) C_B$, there is a problem since $C_H = \alpha_A C_A + (1-\alpha_A)\alpha_B C_B$ because H = A **over** B

Associated Colors

Premultiplying Color

- Problem: have to compute C and α separately
- Instead, use a **premultiplied** color, $c = \alpha C$
- Idea: alpha is some percentage of (subpixel) color
- α = 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: $C_P = \alpha_A C_A + (1-\alpha_A)\alpha_B C_B$
- Premultiplied: $c_P = c_A + (1-\alpha_A)c_B$
  - Same for alpha: $\alpha_P = \alpha_A + (1-\alpha_A)\alpha_B$
- Interpretation: a premultiplied (r,g,b,α) means that the real color is $(R,G,B) = (r/\alpha, g/\alpha, b/\alpha)$

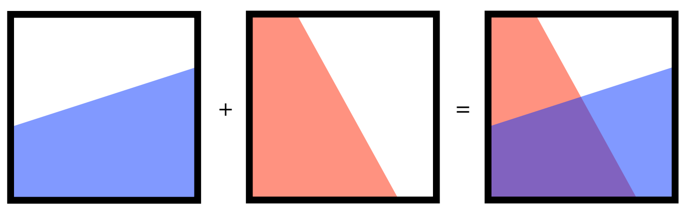
Arithmetic with RGB vs. rgb

- Two ways of operating on pixels (note: ending up with premultiplied colors)
  1. Separately operate on α and C; then multiply to get $c = \alpha^* C$
  2. Operate on premultiplied colors c (multiply first, then operate)
- These may not be consistent.

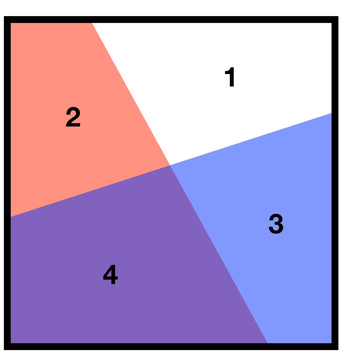
The R channel "leaked" out of transparent pixel

- For example, average the pixels $(C_1,\alpha_1) = (1,0,0,0)$ and $(C_2,\alpha_2) = (0,1,0,1)$:
  1. R = ½(1+0); G = ½(0+1); B = ½(0+0); α = ½(0+1). So C = (1/2,1/2,0,1/2); and then we multiply 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)

Operations on Associated Colors



Porter-Duff Composition



4 regions

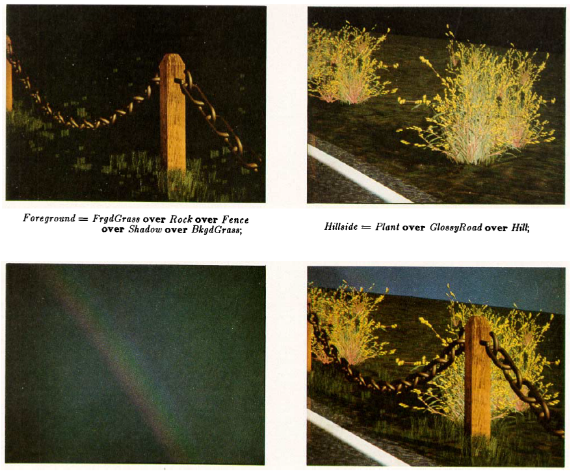
- Region 1: 1 possibility - 0
- Region 2: 2 possibilities - A or 0
- Region 3: 2 possibilities - B or 0
- Region 4: 3 possibilities - A, B or 0
- Operators: 12 total possibilities

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.

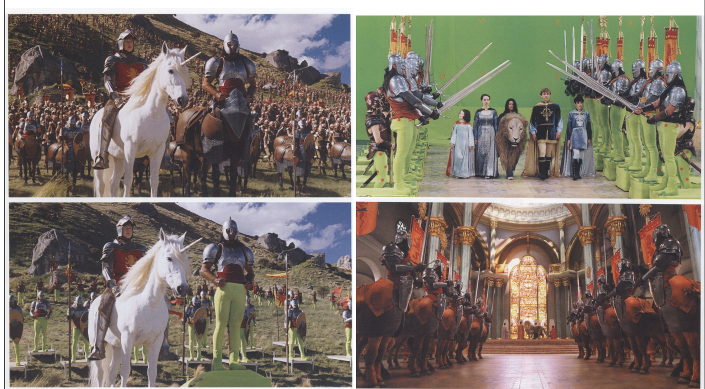
Porter & Duff, 1984

Green Screening

Example: King Kong

Cinefex

Example: Chronicles of Narnia



Cinefex

Or…a Motaur

https://www.adweek.com/agencies/progressives-perplexing-new-mascot-is-a-motaur-half-man-half-motorcycle/

Example: The Dark Crystal: Age of Resistance

https://www.instagram.com/p/BrCybSzg8tf/

Problem: “Pulling a Matte”

- Problem definition, Separate an image C into:
  1. A foreground object image C_F,
  2. a background image C_B,
  3. and an alpha matte α
  - C_F and α can then be used to composite the foreground object into a different image
- This is a HARD problem
  - Even if you only require α to be binary, this is hard to do automatically (why?)
  - For movies/TV, manual segmentation of each frame is infeasible
  - Need to do something procedural (or at least semi-automatic)

Some Simple Solutions

Idea #1: Luma Keying

- If the object is significantly brighter/darker, could look for high luminance (e.g. Y in xyY color space, V in HSV, etc.)

The Art and Science of Digital Compositing, Brinkmann

Idea #1: Luma Keying

- Usually, one needs to modify α to be a function of luminance, it cannot be used directly

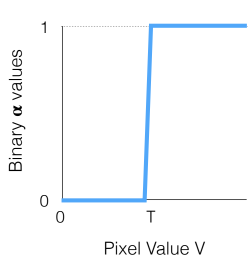
The Art and Science of Digital Compositing, Brinkmann

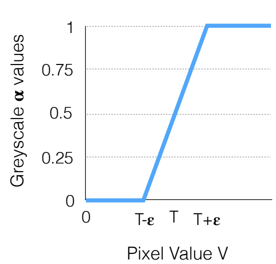
Modifying Luma-Keying

- Instead of a binary matte, can use a smoother function to define α based on a threshold T and a window size ε:

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

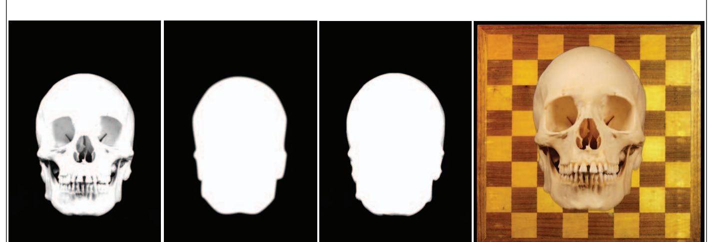
$$\alpha = \begin{cases} 1, & V > T + \varepsilon \\ \frac{V-(T-\varepsilon)}{2\varepsilon}, & T-\varepsilon < V \leq T+\varepsilon \\ 0, & \text{otherwise} \end{cases}$$





Garbage Matting

- Quickly generated user selections can approximate mattes that help distinguish what to include/exclude spatially.



Lumakey | Garbage Matte | Lumakey + Garbage | Composite

The Art and Science of Digital Compositing, Brinkmann

Comparison

Lumakey | Lumakey + Garbage

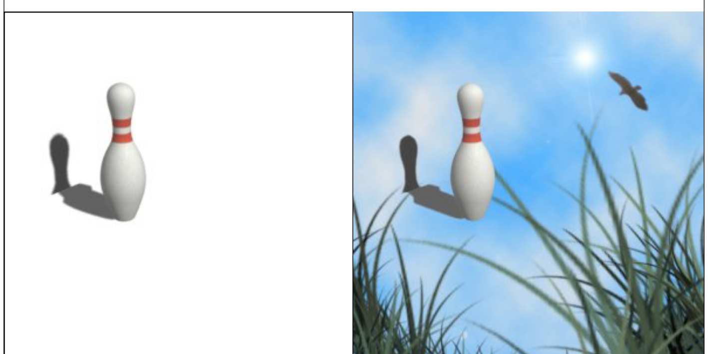
The Art and Science of Digital Compositing, Brinkmann

Idea #2: Difference Matte

- Shoot the scene twice, with and without the object, then subtract

Difference Matte

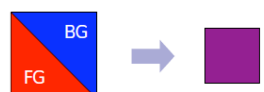
Difference Matte: Example



Matte Applied to Original | Composite

Idea #3: Chroma-Keying

- If the object has a different spectrum of color then the background (e.g. comparing H in HSV color space)
- Ideally, using a range of H or H+S, etc.
- Basis of Blue/Green Screening
  - why blue?, why green?, why not red?
- What about partial coverage in pixels?



Coverage of single pixel | Final pixel appearance

Math of Matte Extraction

- Given an image C, assume it is a composite of a foreground C_F and background C_B.
- Recall the compositing **over** equation (assuming background has α_B=1):
  $C = C_F$ **over** $C_B = \alpha_F C_F + (1-\alpha_F) C_B$

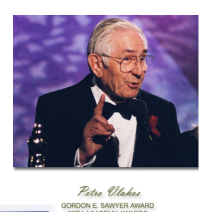
  $R = (\alpha_F R_F) + (1-\alpha_F) R_B$

  $G = (\alpha_F G_F) + (1-\alpha_F) G_B$

  $B = (\alpha_F B_F) + (1-\alpha_F) B_B$
- Knowns (R,G,B) and (R_B, G_B, B_B) Unknowns (R_F, G_F, B_F, α_F)
- 3 equations in 4 unknowns!

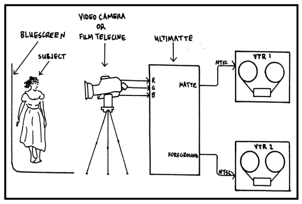
Traditional Blue Screen Matting

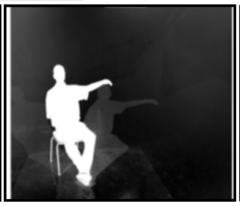
- Invented by Larry Butler in 1940, refined by Petro Vlahos in the 50s (Won Oscar in 1964)
- Initially for film, then video, then digital
- Assume that the foreground has no blue
- Note that computation of α was originally done in analog, required a simple solution.



Cinefex

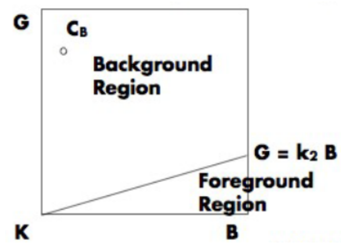
The Ultimatte (Technical Academy Award 1995)





Green Screen Algorithm

- Idea: Divide the color space into two regions
- Separating line at $G = k_2B$ (for greenscreen)
- Colors with high green are background
- $0.5 < k_2 < 1.5$

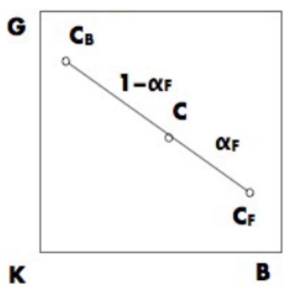


Green Screen Algorithm

- Any background color C has been composited with C_B, this means C is on some line between C_B and C_F

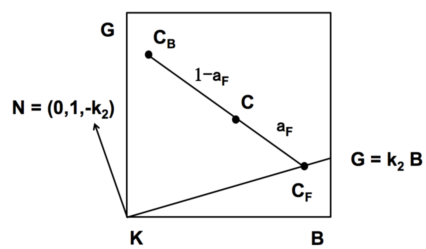
$C = \alpha_F C_F + (1- \alpha_F)C_B \Rightarrow c_F = C - (1- \alpha_F)C_B$

- But we don't know α_F!



Green Screen Algorithm

- Pure background colors are assumed to be above the plane $G = k_2B$
- This fourth equation allows us to solve for the four unknowns.



Blue/Green Screen Matting Issues

- Color limitations
  - Annoying for blue-eyed people
  - So adapt screen color (in particular green)
- Blue/Green spilling and reflection
  - The background illuminates the foreground, blue/green at silhouettes
  - Modify blue/green channel, e.g. set to $\min(b, \alpha_G)$
- Shadows
  - How to extract shadows cast on background?

Blue/Green Screen Matting Issues

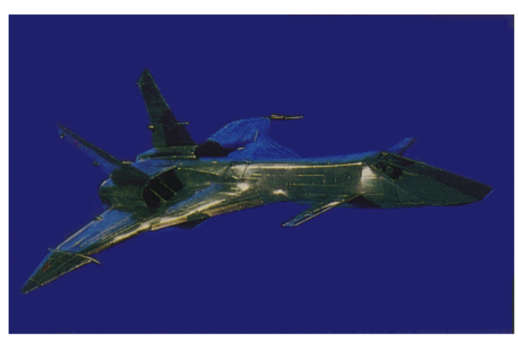


Figure 3. *Firefox Blue Spill Matte Series 1, original shot. Note blue reflected on wing surfaces from bluescreen -- undesirable but unavoidable on such surfaces.*

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

Suppressing Blue Spill

If (B > kG) B = G

Matte = B - max(R,G)
α = 1 - Matte

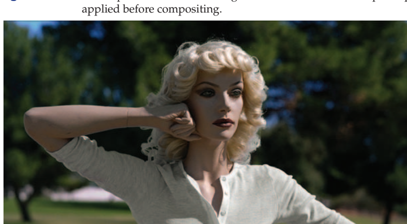
The Art and Science of Digital Compositing, Brinkmann

Blue/Green Screen Matting Issues

The Art and Science of Digital Compositing, Brinkmann

Blue/Green Screen Matting Issues

Figure 13.13c A composite where the foreground element has had spill-suppression applied before compositing.



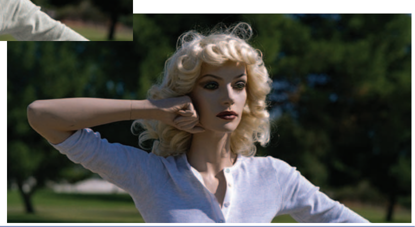


Figure 13.14 The foreground object photographed in the actual scene (not a composite).

The Art and Science of Digital Compositing, Brinkmann

Smith-Blinn Formalization

Blue Screen Matting

Alvy Ray Smith and James F. Blinn[1]
Microsoft Corporation

ABSTRACT

A classical problem of imaging—the *matting problem*—is separation of a non-rectangular foreground image from a (usually) rectangular background image—for example, in a film frame, extraction of an actor from a background scene to allow substitution of a different background. Of the several attacks on this difficult and persistent problem, we discuss here only the special case of separating a desired foreground image from a background of a constant, or almost constant, *backing* color. This backing color has often been blue, so the problem, and its solution, have been called *blue screen matting*. However, other backing colors, such as yellow or (increasingly) green, have also been used, so we often generalize to *constant color matting*. The mathematics of constant color matting is presented and proven to be unsolvable as generally practiced. This, of course, flies in the face of the fact that the technique is commonly used in film and video, so we demonstrate constraints on the general problem that lead to solutions, or at least significantly prune the search space of solutions. We shall also demonstrate that an algorithmic solution is possible

the color film image that is being matted is partially illuminated.

The use of an *alpha channel* to form arbitrary compositions of images is well-known in computer graphics [9]. An alpha channel gives shape and transparency to a color image. It is the digital equivalent of a holdout matte—a grayscale channel that has full value pixels (for opaque) at corresponding pixels in the color image that are to be seen, and zero valued pixels (for transparent) at corresponding color pixels not to be seen. We shall use 1 and 0 to represent these two alpha values, respectively, although a typical 8-bit implementation of an alpha channel would use 255 and 0. Fractional alphas represent pixels in the color image with partial transparency.

We shall use "alpha channel" and "matte" interchangeably, it being understood that it is really the holdout matte that is the analog of the alpha channel.

The video industry often uses the terms "key" and "keying"—as in "chromakeying"—rather than the "matte" and "matting" of the film industry. We shall consistently use the film terminology, after first pointing out that "chromakey" has now taken on a more sophisticated meaning (e.g., [8]) than it originally had (e.g., [19]).

Big Picture Idea: Matting is Ill-posed

- An infinite number of solutions exist to the matting equations:

  $R = (\alpha_F R_F) + (1-\alpha_F)R_B$

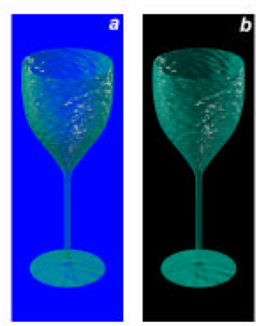
  $G = (\alpha_F G_F) + (1-\alpha_F)G_B$

  $B = (\alpha_F B_F) + (1-\alpha_F)B_B$

- Four unknowns (R_F, G_F, B_F, α_F), but 3 equations
- Need to the problem somehow:
  - e.g. Petro Vlahos approach: G = kB, but could we constrain the system otherwise?

Triangulation Matting

- Instead of reducing the number of unknowns, we could instead increase the number of equations.
- One way to do this is photograph the object in front of two different backgrounds
- This is six equations (3 for each image) but only four unknowns.
- Note: the backgrounds need not be constant, you process this independently per pixel.



Triangulation Matting Algorithm

- For every pixel p in the composite image, given:
  - Backing color $C_{B1} = (R_{B1}, G_{B1}, B_{B1})$ at p
  - Backing color $C_{B2} = (R_{B2}, G_{B2}, B_{B2})$ at p
  - Composited pixel color $C_1 = (R_1, G_1, B_1)$ at p, and
  - Composited pixel color $C_2 = (R_2, G_2, B_2)$ at p, and
- Solve the system of 6 equations:

$$R_1 = \alpha_F R_F + (1-\alpha_F) R_{B1} \qquad R_2 = \alpha_F R_F + (1-\alpha_F) R_{B2}$$

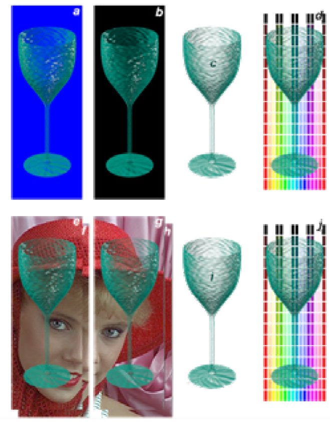
$$G_1 = \alpha_F G_F + (1-\alpha_F) G_{B1} \qquad G_2 = \alpha_F G_F + (1-\alpha_F) G_{B2}$$

$$B_1 = \alpha_F B_F + (1-\alpha_F) B_{B1} \qquad B_2 = \alpha_F B_F + (1-\alpha_F) B_{B2}$$

- For unknowns $(R_F, G_F, B_F, \alpha_F)$

Triangulation Matting Examples

From Smith & Blinn's SIGGRAPH'96 paper

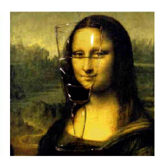


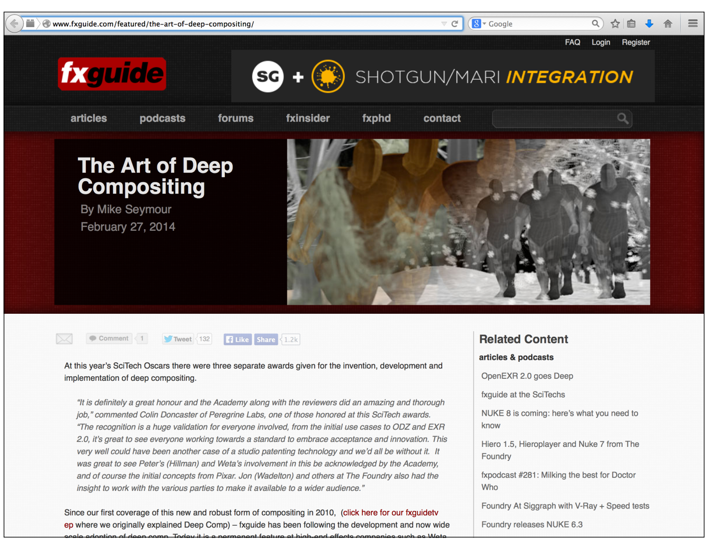
More Examples

Limitations of Relying on α 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.)

Cinefex





Lec08 Required Reading

- FOCG, Ch. 2 (particularly 2.1-2.4)
- See also optional readings!