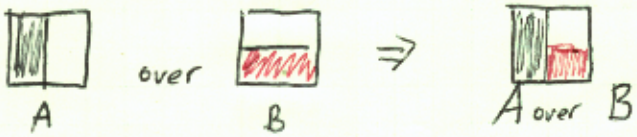
Compositing 101

Lecture 9 and 10

Over operator



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

$$C_A + (1-\alpha_A)\,C_B \qquad \alpha = \alpha_A + \alpha_B - \alpha_{AB}$$

How else to combine

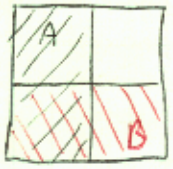
Porter / Duff image algebra

4 regions

0	- could be	0
A		0, A
B		0, B
AB		0, A or B (or AB) ?

$2 \cdot 2 \cdot 3$ possibilities

	0	A	B	AB	F_A	F_B
A over B	0	A	B	A	1	1-a
B over A	0	A	B	B	1-b	1
clear	0	0	0	0	0	0
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-α_B	0
B out A	0	0	B	0	0	1-a
A	0	A	0	A	1	0
B	0	0	B	B	0	1
A atop B	0	0	B	A	α_B	1-α_A
B atop A	0	A	0	B		
XOR	0	A	B	0		



$0 = (1-a)(1-b)$

$a = a(1-b)$

$b = b(1-a)$

$ab = ab$

Blending Functions

$C = F_A C_A + F_B C_B$

(C_A, C_B: premult!)

$\alpha = F_A \alpha_A + F_B \alpha_B$

What's with Alvy's approximations

divide (integer) is VERY expensive

Project -

Canvas

keep adding objects

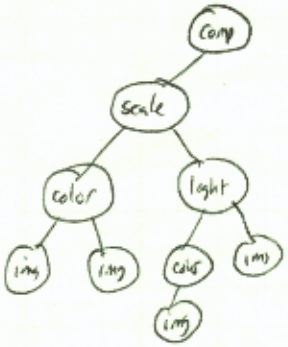
Other Image operators

place
scale

perspective correct
lighting adjust

color correct

graph / tree



← can be huge - thousands! of nodes (Titanic)

vs. Flame

How to get Mattes

Difference Matting $|A-B| > thresh$

registration, lock down matching colors

Chroma key

Simple - look for color

inexact - color range

shadows, creases, highlights, ... ⇒ might actually want shadows!

spill, reflections

anti-aliasing, clear objects, ...

Why Blue? Why not?

most different than common colors

least eye sensitive

first done w/ photographic film

Petro Vlahos – winner of several Academy Awards

Blue Screen Patent 1964.

Ultimatte

continued to invent - Ultimatte is still innovator

Basic patents on video expired late 80s' (still newer ones apply)

allowed for mathematical scrutiny

BASIC IDEA:

Assume ratio of key color / other colors is portion that is clear

$\alpha \approx \frac{B}{G}$ or $\frac{B}{G+R}$ – not Vlahos

Assumption, objects generally have ~~less "Blue" than other colors~~ blue in proportion to other colors

deviation from that may mean backing

e.g. $B \leq \mu_1 G \quad .5 \leq \mu \leq 1.5$

$\alpha = 1 - (B - \mu G)$ (clamped)

(where's the threshold)

or, an improvement

$\alpha = 1 - \mu_1 \left(\min \left(B_f, B_K \right) + \mu_2 G_K \right)$ ⇒ or add red and green

B_f: pixel's blue; B_K: background blue (minimum)

$\mu_2 G_K$: $\max(r,g)$ or max + min

WAIT! THIS IS IMPOSSIBLE!

~~is~~ it half red, or a quarter purple?

4 unknowns (R, G, B, α), 3 knowns (the color at the pixel)

Vlahos is a hack?
yes! - but it works

Blinn and Smith -
4 unknowns, 3 knowns - need either

① more knowns
② less unknowns (or) - assume no blue in object
assume red (or green) is related to Blue

→ 2 different backgrounds
tough! can't change object color (lighting) ← backgrounds always reflect
spill
can't move things

Writing in MATRIX FORM -
Why? Deal with C_K ← the background != $0, 0, B_K$
generalize to any color, not just pure blue

Math is elegant

$$C_O \begin{bmatrix} 1 & & & t_1 \\ & 1 & & t_2 \\ & & 1 & t_3 \\ -R_K & -G_K & -B_K & t_4 \end{bmatrix} = [R_A \; G_A \; B_A \; T]$$

$$\alpha_O R + (1-\alpha_O) R_K = R_F \quad \text{(Matting, equation, over operator)}$$

What is Blinn / Smith Good For?

Vlahos in equation form (not circuits)

Triangulation - tough, but doable in practice (see...)

Vlahos for arbitrary colors

Little tricks to make Vlahos work

$0 < \begin{matrix} R \\ G \\ B \end{matrix} < \alpha$ since premult

What you need to know:

Why matte extraction is hard

Some simple things to do -

OK for projects but understand

Think of other solutions

Convert to HSV?

Use edge detection?

Experiment!

Vlahos works because tuned, and gives knobs