Lecture Notes #3 -- Image Compositing

- Reading:
  - Foley et al., Section 17.6 --or--
  - Thomas Porter and Tom Duff. Compositing digital images. *Computer Graphics* (SIGGRAPH '84) 18(3), 253--259.

Motivation

Often small programs work better than big ones:

The same is true for rendering pictures:

- Want different small programs for different objects.
- Small errors should not force "recompilation" of entire image.

Motivation, cont.

- Therefore, you want to separate an image into "elements":
  - rendered independently
  - **composited** together
- Each element has its own **matte** with coverage information.

The alpha channel

- We encode the matte (or coverage information) in a separate channel, called the **alpha** channel.

$\alpha = 0$	no coverage (totally transparent)
$\alpha = 1$	full coverage (totally opaque)
$0 < \alpha < 1$	partial coverage

The alpha value will control the linear interpolation of foreground and background colors when one element is placed over another.

Relationship between alpha and RGB:

- Alpha computed at same time
- Alpha resolution needs to be comparable
- Alpha treated in almost exactly the same way

The meaning of alpha

Q: How do we express a pixel that is half covered by a full red object?

Porter-Duff solution: Use pre-multiplied value in the color component.

The meaning of alpha, cont.

- In general, (r,g,b,α) indicates a pixel that is α covered by the color $(r/\alpha, g/\alpha, b/\alpha)$.
- This convention allows to draw the (r,g,b) components directly and see antialiased edges.

Q: What is the meaning of the values?

- (0, 0, 0, 1)?
- (0, 0, 0, 0)?
- (1, 1, 1, 1/2)?

The compositing algebra: Assumptions

- We now want to start mixing images together, based on alpha
- We don't have exact coverage information in each pixel, just overall alpha.

So we'll make some assumptions.

We'll consider two cases:

1. The overlap of two semi-transparent objects like haze.
2. The overlap of two opaque hard-edged elements.

Case 1: Semi-transparent objects

Let α_A and α_B be the opacities of A and B.

Suppose we put A and B over a background G.

Questions:

- How much of G shows through both A and B?
- How much of G is blocked by A and passed by B?
- How much of G is blocked by B and passed by A?
- How much of G is blocked simultaneously by A and B?

Case 2: Opaque objects

Let α_A and α_B be the coverages of two opaque objects A and B within a pixel.

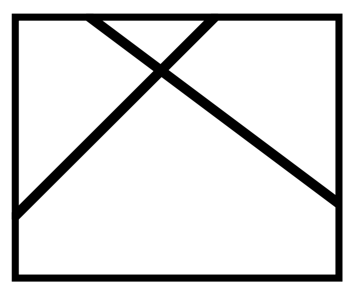
Compositing assumption:

The portions of the pixel covered by A and B are random.

(No correlation between A and B)

Case 2, cont.

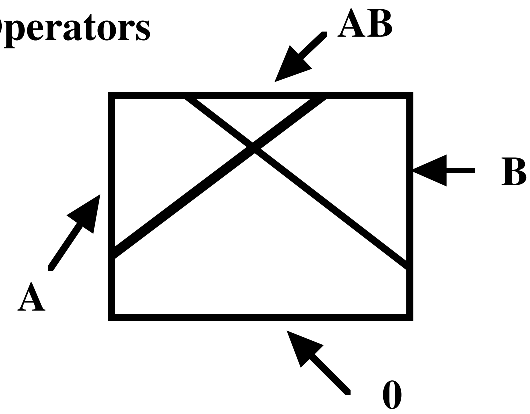
Q: What are the resulting areas of:



Q: When do these assumptions break down?

Compositing Operators

Two pictures divide a pixel into 4 subareas:



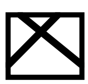
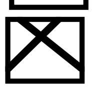
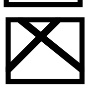
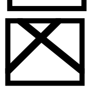
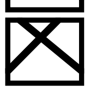
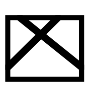
Name	*Choices*
0	0
A	0,A
B	0,B
AB	0,A,B

Each of these areas can be filled in with different choices, listed above.

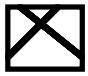
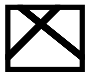
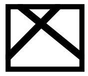
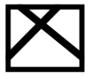
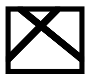
Q: How many choices in all?

Compositing operators, cont.

Let's look at the 12 choices for filling in regions 0, A, B, AB.

1. (0, 0, 0, 0) 
2. (0, A, 0, A) 
3. (0, 0, B, B) 
4. (0, A, B, A) 
5. (0, A, B, B) 
6. (0, 0, 0, A) 

Compositing Operators, cont.

7. (0, 0, 0, B) 
8. (0, A, 0, 0) 
9. (0, 0, B, 0) 
10. (0, 0, B, A) 
11. (0, A, 0, B) 
12. (0, A, B, 0) 

Blending alphas

Let $F_{\mathbf{A}}$ and $F_{\mathbf{B}}$ be the fractions of the original mattes of A and B that prevail in the composite.

Q: What is the alpha value of the composite?

Blending color components

Let C_A, C_B, C_O be the "true" color components of A, B and the composite, and let c_A, c_B, c_O be the pre-multiplied versions (which we want to compute.

Then, c_O =

Finding the Fractions

- Now we just need to find F_A and F_B
- These are easily found by inspection, from the compositing diagrams.

Examples:

Operation	F_A	F_B
clear		
A		
B		
A over B		
A in B		
A "plus" B		

Unary operators

- A couple of unary operators are also useful:

Example

(*FFire* **plus** (*BFire* **out** *Planet*)) **over** **darken**(*Planet,* 0.8) **over** *Stars*