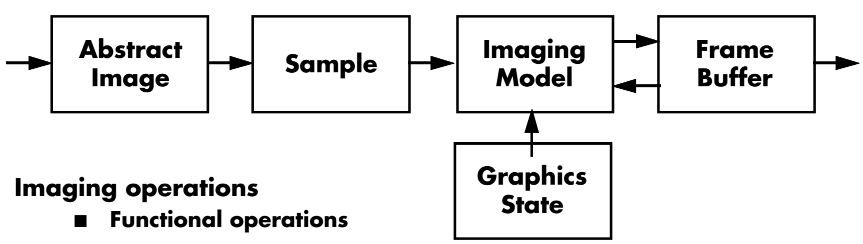
# Compositing and Painting

**Abstract images**
**Imaging models**
**Image composition**
**Mattes and alpha channel**
**Compositing operators**
**Painting**




# Imaging Model




**Imaging operations**

- Functional operations
  user: erase, draw, reverse, highlight
- Logical operations
  bitblt: clr, mov, set, xor, and, or
- Arithmetic operations
  blend: cross-dissolve, painting and compositing

**Graphics state (context)**

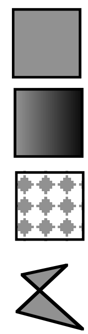
- color, texture, pattern, clipping ...

## Abstract Images

**Definition: Returns color as a function of x and y**

**Examples**

- **card**
- **ramp**
- **pattern or texture**
- **text**
- **lines and polygons**
- **"gleams" and "glows"**
- **sampled image arrays**






## Image Composition

**Combine *foreground* with *background***

**Examples:**

- **Mounting a photograph**
- **Graphics arts: masks, friskets, stencils**
- **Special Effects in Movies: optical printer**
- **Animation: cels**
- **Television: video keying**
- **Computer graphics: alpha compositing**

## Optical Printing




## Video Keying

## Mattes and The Alpha Channel

**Mattes are used to control the composition of multiple images or elements.**

**A alpha channel is image that defines the presence or absence of imagery, or more generally whether another image is *transparent* or *opaque.***

**Alpha channels may be n-ary, not just binary**

- **Soft-edge composites at edges**
- **Transparency**

**RGBA pixel format**




## "Pulling a Matte" - Matte Creation

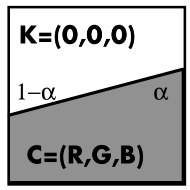
**From digitized images**

- **Blue-screen matting (Petro Vlahos)**
  **Separate blue background from foreground image**
- **Luminance**
  **Range of lumiances marked transparent - video keying**
- **Others ...**
  **Set of colors marked transparent, region growing ...**

**From computer generated images**

- **Coverage**
- **Transparency**

## Fragment Coverage




Pixel

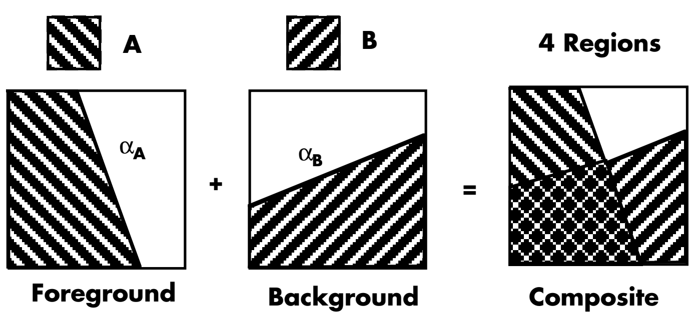
**$\alpha$ = Area of Fragment**

**Final color of pixel is a weighted average of C and K (Black)**

$$c = \alpha C + (1-\alpha) K = \alpha C$$




## Multiple Fragments




**Assume: Uncorrelated**

| Region | Fraction |
|---|---|
| A * B | $\alpha_A \alpha_B$ |
| ~A * B | |
| A * ~B | |
| ~A * ~B | |

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

## Porter-Duff Compositing Algebra

- Composite foreground with background element
- 12 possible cases determined by the fractions

$$C = F_A C_A + F_B C_B$$
$$\alpha = F_A \alpha_A + F_B \alpha_B$$

- Over is the most important!

$$F_A = \alpha_A$$
$$F_B = (1-\alpha_A)\alpha_B$$




## Color and Matte Representation

**Distinct image and matte (unassociated)**
- Color channel: (r,g,b)
- Matte channel: ($\beta$)

**Integral image and matte (associated)**
- Conceptually tied together -> like a digital bipack
- Premultiplied alpha: ($\alpha$r,$\alpha$g,$\alpha$b,$\alpha$)
  - black = (0,0,0,1); white = (1,1,1,1); clear = (0,0,0,0)
- More efficient
  - OVER operator (1 sub, 4 muls, 4 adds)
  - $c = C_f + (1-\alpha_f)\, C_b$

## Painting

***Canvas*** **: Current image representing the painting**

***Brush*** **: Small matte specifying the brush**

***Paint*** **: Color and texture properties of the paint**

**Paint equation**

> ***paint*** **=** ***color*** **IN** ***texture***
>
> ***canvas*** **+=(*canvas* OP *paint*) IN *brush***

**Paint modes**

- **Smudge, dissolve, sharpen, blur, saturate, desaturate, dodge/burn, tint, erase, ...**