Image Matting

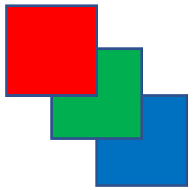
Topic 2

Week 2 – Jan. 16th, 2019

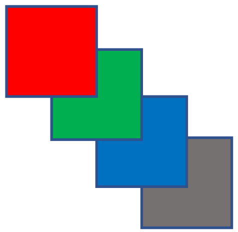
Topic 2: Image Matting

- **Alpha Masks**
- The Matting Equation
- Four Ways to Solve the Matting Equation
- The Triangulation Matting algorithm

Alpha Channel



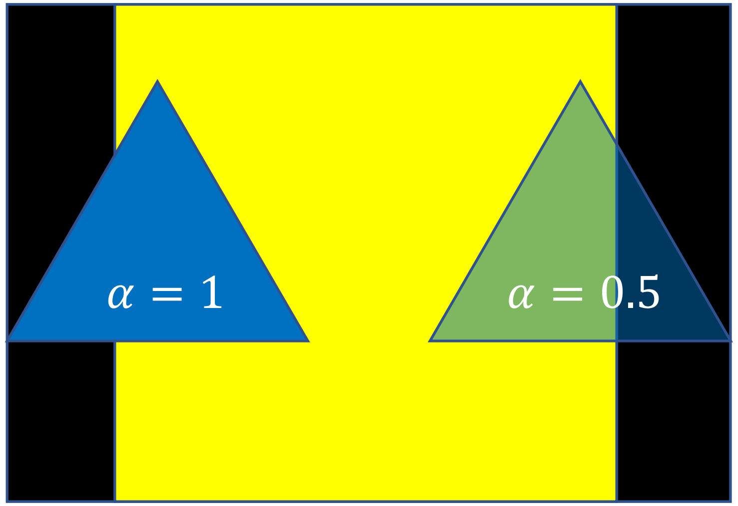
- Typical images from cameras are captured with 3 channels: RGB
  - i.e. every image pixel has 3 values: R, G, B
- Useful to have a 4th channel: α (alpha)
  - Has same representation as pixels (i.e. 0-255 for 8-bit image), but assumed to be fractional value between 0-1



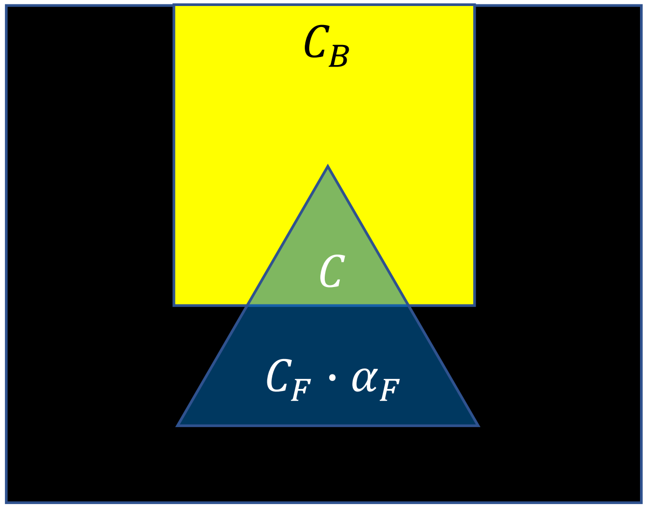
- Alpha channel is pixel "transparency”

$$\begin{bmatrix} R \\ G \\ B \\ \alpha \end{bmatrix} \rightarrow \begin{bmatrix} \alpha R \\ \alpha G \\ \alpha B \\ \end{bmatrix}$$

Alpha Channel: Compositing



- Alpha channel is pixel "transparency”
  - $\alpha = 0$ ➔ pixel is transparent
  - $\alpha = 1$ ➔ pixel is opaque
- When representing RGBA pixel as RGB, we calculate the alpha composite,
  - Foreground pixel (RGBA): $C_F = [R_F, G_F, B_F]$, α_F and,
  - Background pixel (RGB): $C_B = [R_B, G_B, B_B]$



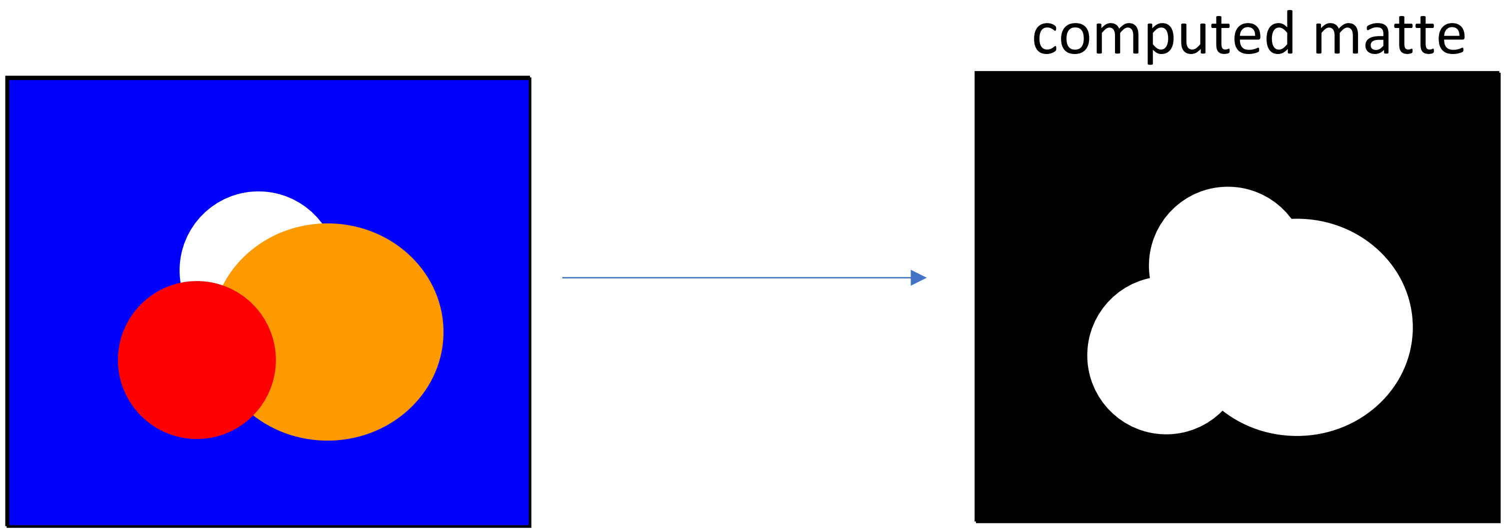
$$C = \alpha_F \cdot C_F + (1 - \alpha_F) \cdot C_B$$

Overlaying Graphics on Reality

- Alpha masks are often used to combine graphics and reality
  - i.e. “Green Screen”
- Applications:
  - TV weather reporting
  - Sports advertising
  - Special visual effects

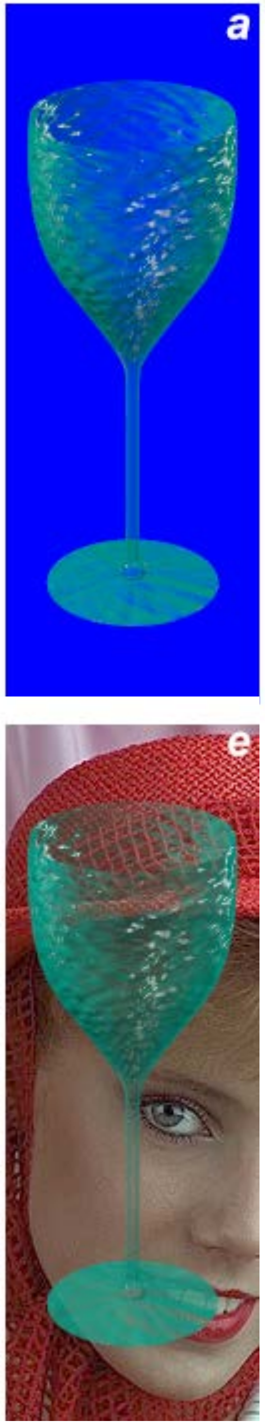
Constant-Colour Image Matting

- We want to separate “foreground” from ”background” – ill-defined
- Assume pixels of a given constant colour belong to background
- i.e. blue-screen matting:



Semi-Transparent Matte

- Constant colour matting doesn't look very realistic
- Better results if we can obtain semi-transparent matte
- Uses alpha channel for matte – i.e. matte is a greyscale image



Topic 2: Image Matting

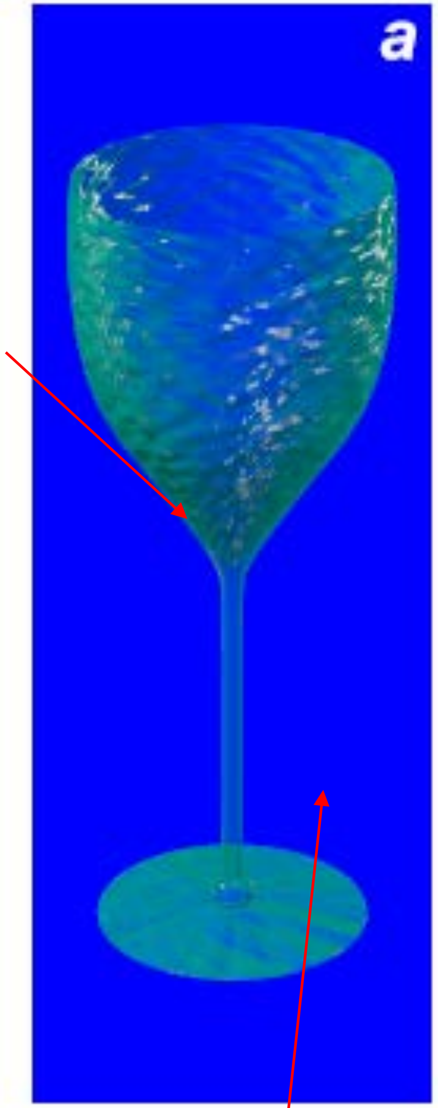
- Alpha Masks
- **The Matting Equation**
- Four Ways to Solve the Matting Equation
- The Triangulation Matting algorithm

Matting Equation

- Matting Problem: We want to extract all the foreground pixels $C_0 = [\mathrm{R}_0, \mathrm{G}_0, \mathrm{B}_0]$, and matte α_0
- Given (for every pixel):
  - Background pixel colour $B = [\mathrm{R_k}, \mathrm{G_k}, \mathrm{B_k}]$
  - Composite pixel colour $C = [\mathrm{R}, \mathrm{G}, \mathrm{B}]$
- We can consider the composite pixel C to be a mixture of the foreground (F) and background (B) pixels:

$$C = \alpha_F F + (1 - \alpha_F)B$$

$\alpha_o \cong 1$



$\alpha_o = 0$

Matting Equation: Why it's Hard

- Matting Equation: $C = \alpha_F F + (1 - \alpha_F)B$
- i.e.

$$C_r = \alpha_F F_r + (1 - \alpha_F)B_r$$
$$C_g = \alpha_F F_g + (1 - \alpha_F)B_g$$
$$C_b = \alpha_F F_b + (1 - \alpha_F)B_b$$

- 3 equations, 7 unknowns
- More unknowns than equations!
- We call this an underdetermined (or underconstrained) system

Topic 2: Image Matting

- Alpha Masks
- The Matting Equation
- **Four Ways to Solve the Matting Equation**
- The Triangulation Matting algorithm

Approaches to Underdetermined Systems

- An under-determined linear system of equations has infinitely many solutions (or no solutions if it is inconsistent)
  - e.g. $\mathrm{x} + \mathrm{y} = 1$ (one equation, two unknowns)
  - There are infinitely many values of x, y that satisfy this!

General approaches to these systems:

- Reduce the number of unknowns, i.e. assume values for some unknowns
  - e.g. if we assume $\mathrm{y} = 0.3$, can solve for x
  - makes sense if y is a real-world variable we can control, and we care about x
- Increase the number of equations (introduce more constraints)
  - e.g. if we assume that $\mathrm{x} + 2\mathrm{y} = 1.3$, we can solve for x

Approach 1: Known Background

- Reduce the number of unknowns
- Assume background $B = [B_r, B_g, B_b]$ is known

  - Then alpha matte also known, $\alpha_F = \begin{cases} 0, & C = B \\ 1, & C \neq B \end{cases}$

$$C_r = \alpha_F F_r + (1 - \alpha_F) B_r$$
$$C_g = \alpha_F F_g + (1 - \alpha_F) B_g$$
$$C_b = \alpha_F F_b + (1 - \alpha_F) B_b$$

  - 3 equations, 3 unknowns

Approach 1: Downsides

- Background colour must be known very accurately, and must be constant
- Foreground subject cannot have pixels similar to the background colour
- Alpha is binary ➔ no transparency or smooth transitions between objects

Approach 2: Blue Screen Matting

- Assume background contains only blue, i.e. $B = [B_r = 0, B_g = 0, B_b]$
  - Note: allows lighting to vary, because blue brightness not fixed, i.e. could be [0, 0, 1] or [0, 0, 0.5]
  - Matting equations simplify:

$$
\begin{aligned}
C_r &= \alpha_F F_r \cancel{+ (1 - \alpha_F) B_r} \\
C_g &= \alpha_F F_g \cancel{+ (1 - \alpha_F) B_g} \\
C_b &= \cancel{\alpha_F F_b} + (1 - \alpha_F) B_b
\end{aligned}
$$

  - 3 equations, 3 unknowns, and can still use α

Approach 2: Downsides

- Having no blue in foreground in practice is difficult!
- Not good for people with blue eyes (i.e. Hollywood movie actors/actresses)
- Excludes any colour with blue component – i.e. 2/3 of all hues, including gray, pastels and white!
- ”Blue/green spilling”: light reflected off background onto foreground, making it have some blue component

Approach 3: Gray or Skin Coloured Foreground

- Still blue screen, i.e. $B = [B_r = 0, B_g = 0, B_b]$, but *generalize a little*
- Assume foreground colour is either:
  - Gray: i.e. $F_r = F_g = F_b$
  - "Flesh": $F = [F_r = d, F_g = \frac{d}{2}, F_b = \frac{d}{2}]$ where d depends on skin tone
- Equations simplify (e.g. gray):

$$\begin{aligned} C_r &= \alpha_F F \cancel{+ (1 - \alpha_F) B_r} \\ C_g &= \alpha_F F \cancel{+ (1 - \alpha_F) B_g} \\ C_b &= \alpha_F F + (1 - \alpha_F) B_b \end{aligned}$$

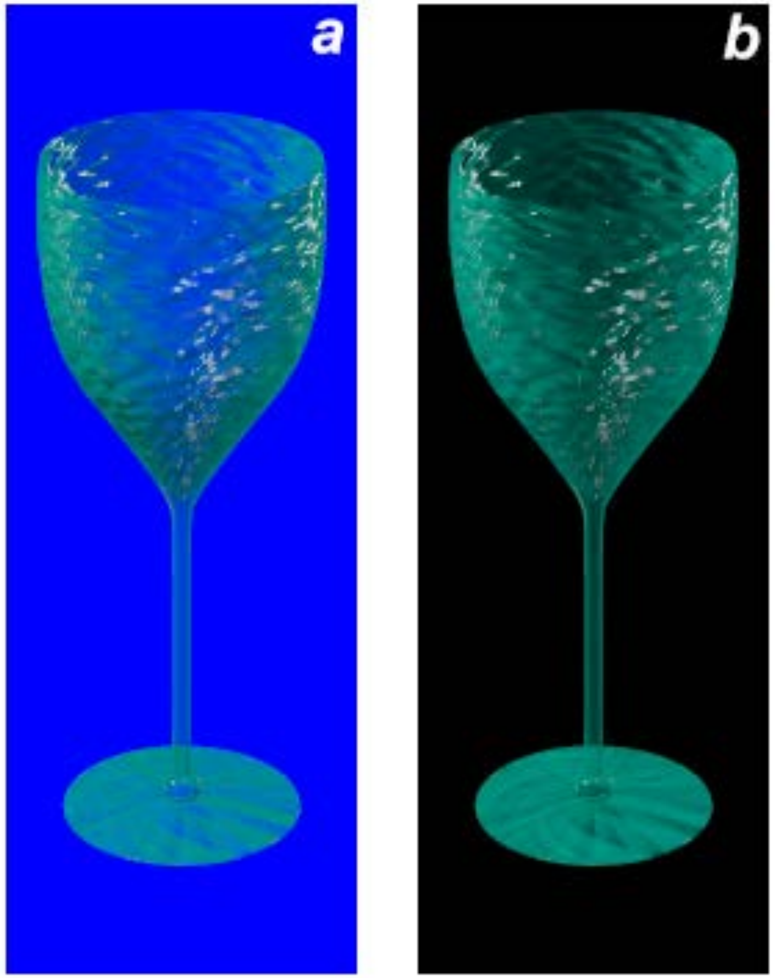
- 2 unknowns, 3 equations

Approach 4: Triangulation Matting

- Increase number of equations, by using two different backgrounds
- Assume backgrounds $B_0 = [B_{0,r}, B_{0,g}, B_{0,b}]$ and $B_1 = [B_{1,r}, B_{1,g}, B_{1,b}]$ known

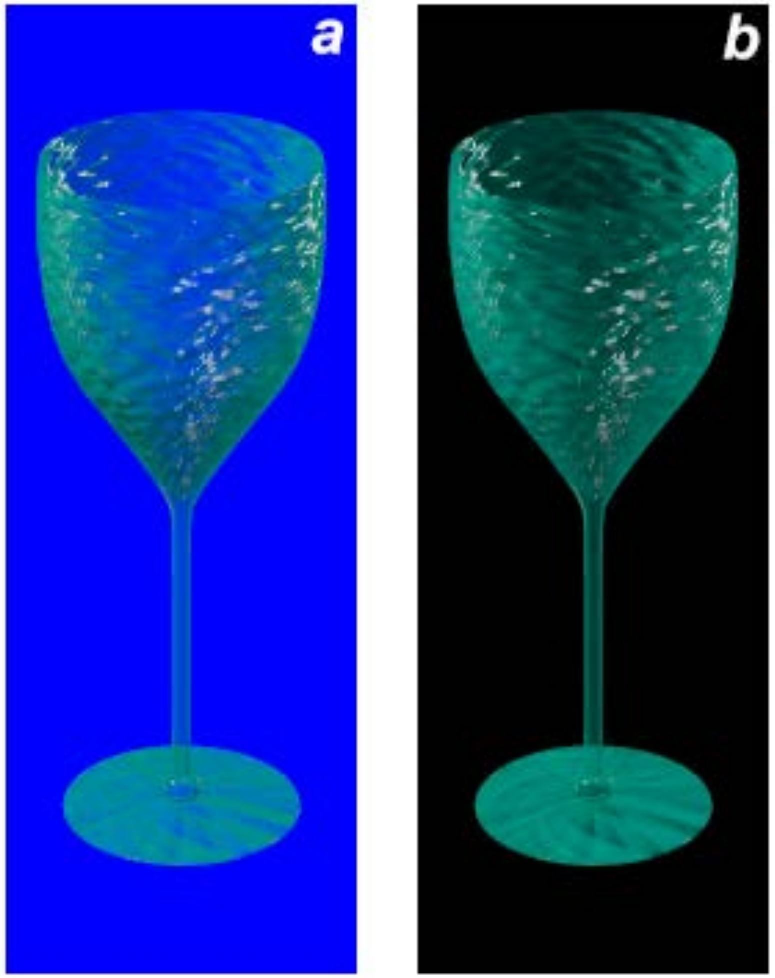
$$
\begin{aligned}
C_{0,r} &= \alpha_F F_r + (1 - \alpha_F) B_{0,r} \\
C_{0,g} &= \alpha_F F_g + (1 - \alpha_F) B_{0,g} \\
C_{0,b} &= \alpha_F F_b + (1 - \alpha_F) B_{0,b} \\
C_{1,r} &= \alpha_F F_r + (1 - \alpha_F) B_{1,r} \\
C_{1,g} &= \alpha_F F_g + (1 - \alpha_F) B_{1,g} \\
C_{1,b} &= \alpha_F F_b + (1 - \alpha_F) B_{1,b}
\end{aligned}
$$

  - 6 equations, 4 unknowns

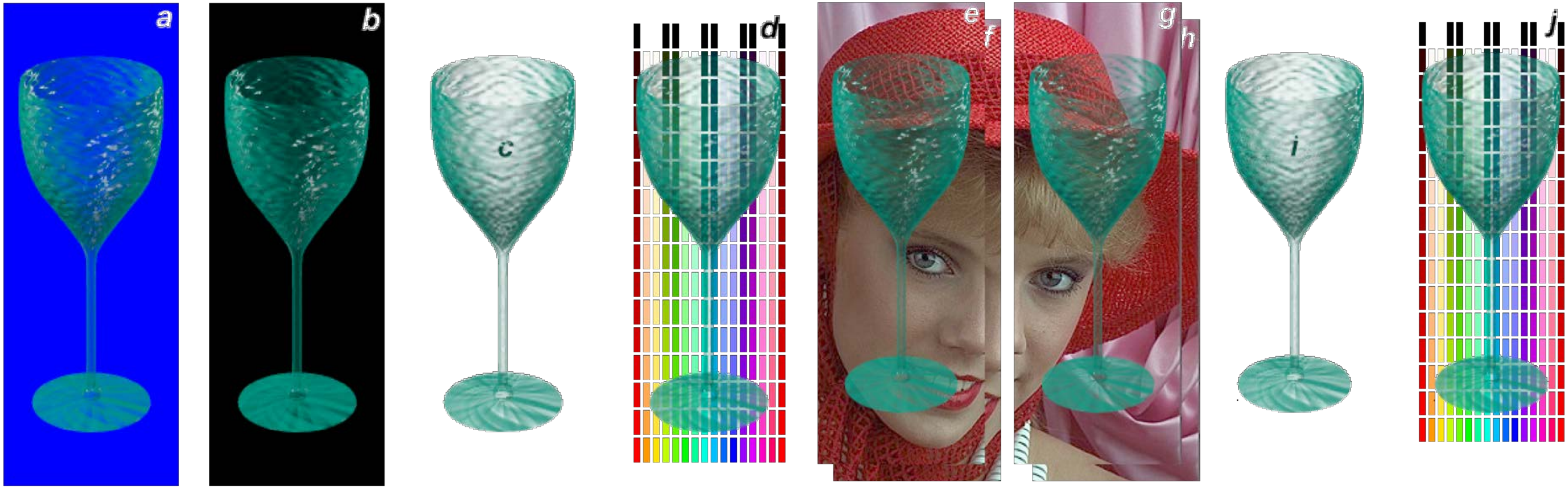


Approach 4: Downsides

- Need two photos with two different known backgrounds for each object
- Object must have same lighting in both images
- Object must be static



Approach 4: Examples



Matting Equation: In Matrix Form

Non-linear

Matting Equation: $C = \alpha_F F + (1 - \alpha_F)B$, define $F' = \alpha_F F$

i.e.

$$\left.\begin{aligned} C_r &= F_r' + (1-\alpha_F)B_r \\ C_g &= F_g' + (1-\alpha_F)B_g \\ C_b &= F_b' + (1-\alpha_F)B_b \end{aligned}\right\} \text{Linear}$$

Define $C_\Delta = C - B$

Then

$$\begin{aligned} C_{\Delta r} &= F_r' - \alpha_F B_r \\ C_{\Delta g} &= F_g' - \alpha_F B_g \\ C_{\Delta b} &= F_b' - \alpha_F B_b \end{aligned} \Rightarrow \begin{bmatrix} C_r - B_r \\ C_g - B_g \\ C_b - B_b \end{bmatrix} = \boxed{\begin{bmatrix} C_{\Delta r} \\ C_{\Delta g} \\ C_{\Delta b} \end{bmatrix} = \begin{bmatrix} 1 & & & -B_r \\ & 1 & & -B_g \\ & & 1 & -B_b \end{bmatrix} \begin{bmatrix} F_r' \\ F_g' \\ F_b' \\ \alpha_F \end{bmatrix}} = \begin{bmatrix} 1 & & & -B_r \\ & 1 & & -B_g \\ & & 1 & -B_b \end{bmatrix} \begin{bmatrix} \alpha_F F_r \\ \alpha_F F_g \\ \alpha_F F_b \\ \alpha_F \end{bmatrix}$$

Triangulation Matting : In Matrix Form

Similarly, define $F' = \alpha_F F$, and Define $C_\Delta = C - B$

$$
\begin{aligned}
C_{0,r} &= \alpha_F F_r + (1-\alpha_F)B_{0,r} \\
C_{0,g} &= \alpha_F F_g + (1-\alpha_F)B_{0,g} \\
C_{0,b} &= \alpha_F F_b + (1-\alpha_F)B_{0,b} \\
C_{1,r} &= \alpha_F F_r + (1-\alpha_F)B_{1,r} \\
C_{1,g} &= \alpha_F F_g + (1-\alpha_F)B_{1,g} \\
C_{1,b} &= \alpha_F F_b + (1-\alpha_F)B_{1,b}
\end{aligned}
\quad \Rightarrow \quad
\begin{bmatrix} C_{\Delta 0,r} \\ C_{\Delta 0,g} \\ C_{\Delta 0,b} \\ C_{\Delta 1,r} \\ C_{\Delta 1,g} \\ C_{\Delta 1,b} \end{bmatrix}
=
\begin{bmatrix}
1 & & & -B_{0,r} \\
 & 1 & & -B_{0,g} \\
 & & 1 & -B_{0,b} \\
1 & & & -B_{1,r} \\
 & 1 & & -B_{1,g} \\
 & & 1 & -B_{1,b}
\end{bmatrix}
\begin{bmatrix} F_r{}' \\ F_g{}' \\ F_b{}' \\ \alpha_F \end{bmatrix}
$$

Pseudo-inverse to compute the solution.

Next
Topic 3: HDR