CS4405

Compositing

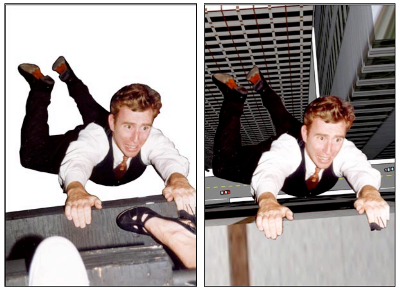
Digital Compositing

- Digital compositing is the digitally manipulated integration of at least two source images to produce a new image
- The new image usually appears realistic
  - Completely and seamlessly integrated as if it were actually photographed by a single camera

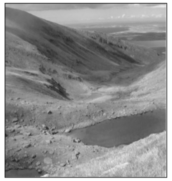
Main Topics

- Alpha blending is blending foreground and background
- Keying is separating foreground and background
  - Techniques luma, chroma and difference keying
- Rig removal is removing unwanted elements

Example

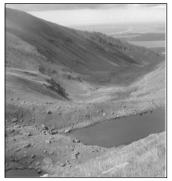


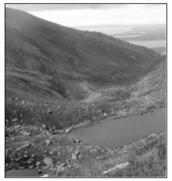
RGB Channels



red

green





blue

RGB and Alpha Channels

- RGB images contain three channels of colour information
  - Manipulated separately
  - In combination to produce various effects
- RGB images can also contain a fourth channel called an alpha channel
  - Controls transparency
- The alpha channel of an image enables you to create areas of transparency
  - Enables a wide range of effects

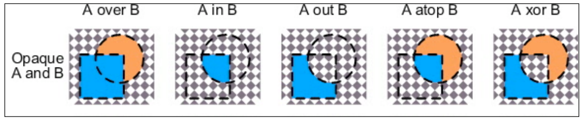
Alpha Channel

- The alpha component specifies the degree of transparency for each pixel
  - An image format that has an alpha component for each pixel is said to have an alpha channel
- In order to correctly combine compound images it is necessary to keep an associated matte for each element
- To distinguish between the parts of the image where
  - It is drawn
  - It is not drawn

Alpha Channel

- The alpha channel contains a value of either 0 or 1
  - A value of 0 means that the pixel does not have any coverage information
  - A value of 1 means that the pixel is fully opaque
- With an alpha channel it is easy to express useful compositing image operations

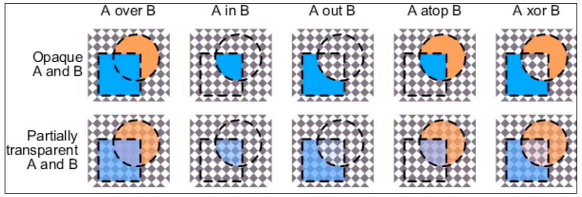
Compositing Operations



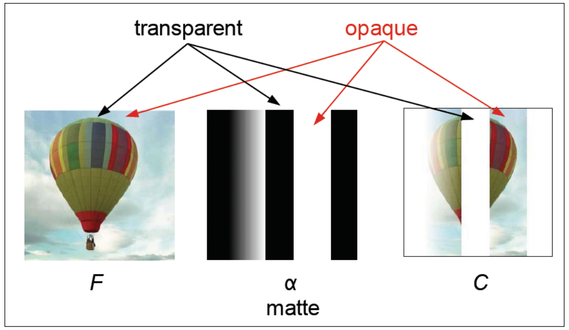
Alpha Blending

- In this case the value of alpha in the colour channel ranges from 0.0 to 1.0
  - 0.0 represents a fully transparent colour
  - 1.0 represents a fully opaque colour
- The value of the resulting colour C when colour F with an alpha value of α is drawn over an opaque background of colour B is
  - C = α * F + (1.0 - α) * B

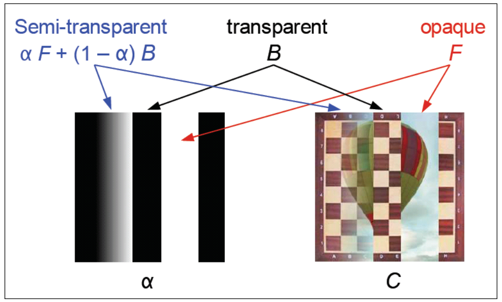
Compositing (Alpha Blending)



Example: No Background



Example: With Background



Semi-transparent

$\alpha F + (1 - \alpha) B$

transparent

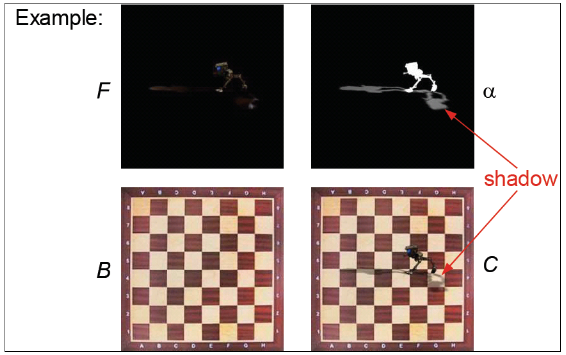
B

opaque

F

α

C



Example:

F

α

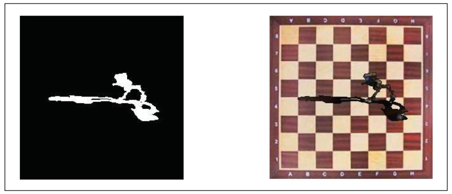
shadow

B

C

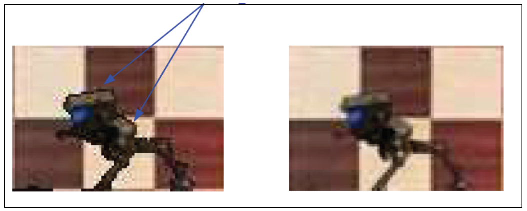
Shadows

- For shadows α must take fractional value (0 < α < 1)
  - Otherwise the shadow looks unreal



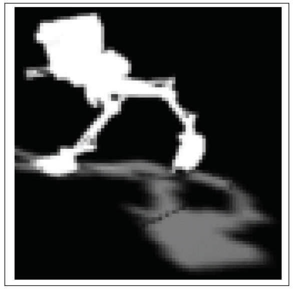
Boundary area

- α at the boundary area should also be fractional
  - Otherwise the composition will have dark fringes



Matte Regions

- A good matte has fractional α
  - In shadow
  - Along object boundaries
  - Along shadow boundaries

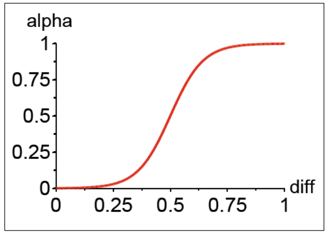


Keying

- Separating foreground from background means creating a matte of the foreground
  - Also called pulling a matte (of foreground) or keying out (making transparent) the background
- A good matte has fractional α in shadow and along object boundaries and shadow boundaries
- In general this is a hard problem
  - Need to make simplifying assumptions

Basic Idea

- Compute difference between foreground and background
- Very small difference α = 0
- Very large difference α = 1
- Intermediate difference intermediate α



Keying Methods

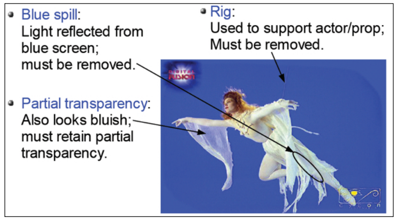
- Luma keying is based on luminance (intensity)
- Chroma keying is based on colour (blue screen, green screen)
- Difference keying requires a clean plate
  - A background image without the foreground elements

Luma Keying

- Key out the background based on luminance
  - Useful when background has a uniform luminance that is very different from foreground luminance (fire, smoke)
- This technique is less controllable than using colour

Chroma Keying

- Key out the background based on colour (or more usually a limited colour range)
  - Useful when background has a uniform colour that is very different from foreground colour

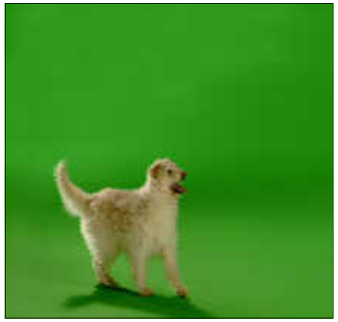


- Blue spill: Light reflected from blue screen; must be removed.
- Rig: Used to support actor/prop; Must be removed.
- Partial transparency: Also looks bluish; must retain partial transparency.

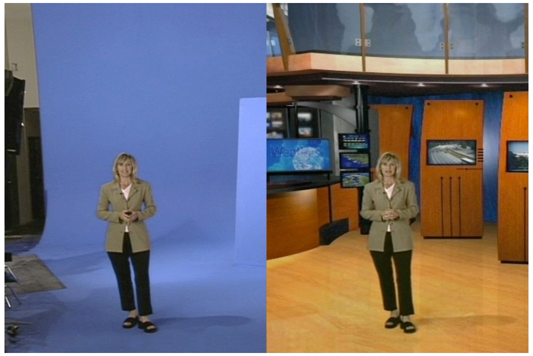
Chroma Background Colour

- Blue is complementary to human skin tone
- Green has become widely used because digital cameras retain more detail in the green channel
  - So green requires less lighting than blue
- Note that any colour can be used
  - You want to avoid colours that are in the scene

Green Screen Example



Blue Screen Example



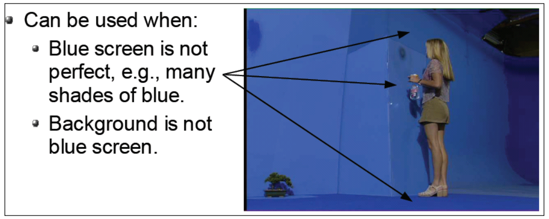
Not Just Backgrounds

Difference Key

- A difference key uses a background image of the scene that the foreground object is being keyed out of
  - The difference key examines the source image for any pixels that don't match
  - The background image must be taken from the same camera position (angle, focus and distance)
- Compare the image to be keyed with the background image and generate a mask based on the difference

Difference Keying

- More general than luma and chroma keying
- Key out background based on the per-pixel colour difference between foreground and background



- Can be used when:
  - Blue screen is not perfect, e.g., many shades of blue.
  - Background is not blue screen.

Final Composition

Rig Removals

- Rigs are equipment that support the actors or the props
- Sometimes rigs cannot be removed by keying alone
  - Apply masking technique to remove rigs
- If the camera moves then need a motion-controlled camera
- Computer controls camera to move the same way twice
  - Without foreground objects to get a clean plate
  - With foreground objects

Rig Removal Tasks

- Apply a mask to mask out the rig
- Replace pixels in masked area by corresponding pixels in clean plate background

The Matting Problem

- The colour $C = [R\ G\ B\ \alpha]$ for every pixel in the composite image is a function of
  - The foreground colour $C_f = [R_f\ G_f\ B_f\ \alpha_f]$
  - The new background colour $C_b = [R_b\ G_b\ B_b\ \alpha_b]$
  - Assume $\alpha_f = \alpha_b = 1$
- The uncomposited foreground colour $C_o = [R_o\ G_o\ B_o\ \alpha_o]$
  - The foreground object in isolation from any background

The Matting Problem

- Given C_f and C_b at corresponding points, the known background C_k and assuming
  - $C_f = C_o + (1 – \alpha_o)\ C_k$
- Determine C_o the uncomposited foreground colour
- Which then gives the composite colour C at the corresponding point from
  - $C = C_o + (1 – \alpha_o)\ C_b$

Matting is a Hard Problem

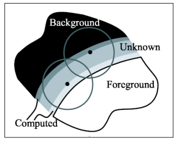
- The matting equation
  - $C_f = C_o + (1 – \alpha_o) C_k$
- Expands into three equations for [R G B]
  - $R_f = \alpha_o R_o + (1 – \alpha_o) R_k$
  - $G_f = \alpha_o G_o + (1 – \alpha_o) G_k$
  - $B_f = \alpha_o B_o + (1 – \alpha_o) B_k$
- But we have four unknowns (R_o, G_o, B_o, α_o)
  - Infinite number of solutions can be found
  - Need to constrain the problem to find a solution

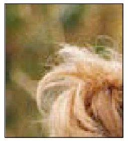
Matting Applications

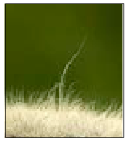
- Known background
  - Controlled studio environment
- Natural image matting
  - Image analysis required

Matting Problems

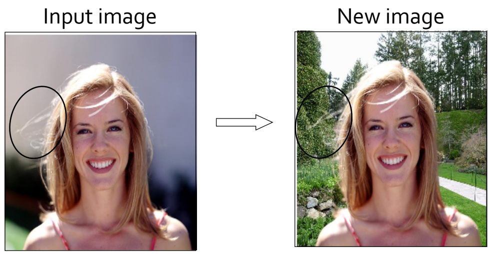
- Regardless of the environment certain types of content cause problems when computing a matte
- Fine detail such as hair
- Difficult to decide what belongs to
  - The background
  - The foreground



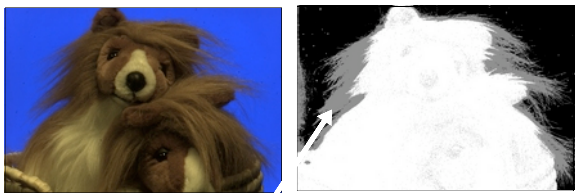




Matting Problems Natural Images



Mention necessity of alpha matte (for complex foreground)



Uncertain α values

Assume No Blue in Foreground

- The known background contains only blue [0, 0, B_k]
- If the uncomposited foreground does not contain blue [R_o G_o 0]
  - $R_f = R_o + (1 - \alpha_o) R_k$
  - $G_f = G_o + (1 - \alpha_o) G_k$
  - $B_f = 0 + (1 - \alpha_o) B_k$
- $C_o = [R_f\ G_f\ 0,\ 1\text{-}B_f/B_k]$
  - Note: $B_k \neq 0$

Assume No Blue in Foreground

- Three equations with three unknowns so only one solution exists
- Ideal situation but contains a very restrictive assumption (e.g. no grey)
  - Eliminates most colours (e.g. no grey or white, hues, skin tone)
- However, the solution can be extended to give a solution for grey and flesh tone

Grey in Foreground

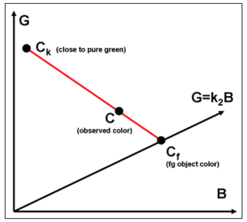
- If the foreground contains grey then
  - $R_o = G_o = B_o$
  - $R = R_o + (1 – \alpha_o) R_k$
  - $G = R_o + (1 – \alpha_o) G_k$
  - $B = R_o + (1 – \alpha_o) B_k$
- Three equations with two unknowns (R_o, α_o)
  - Over-determined system
  - Either one equation is redundant or the system is inconsistent

Flesh in Foreground

- If the foreground contains flesh then
  - $C_o = [d\ 0.5d\ 0.5d]$
  - d depends on the lightness or darkness of the skin
  - $R = d + (1 – \alpha_o) R_k$
  - $G = 0.5d + (1 – \alpha_o) G_k$
  - $B = 0.5d + (1 – \alpha_o) B_k$
- Three equations with two unknowns (d, α_o)

Solution (Vlahos Assumption)

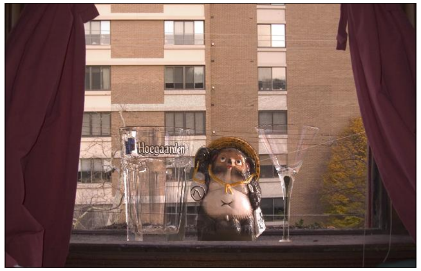
- Assume that all foreground colours have the magnitude of their blue component related to the magnitude of the green by some ratio k_2 ($G_f = k_2 * B_f$)
  - Or all foreground colours lie in a plane in colour space
- The observed colour C is a linear combination of the background colour and a foreground colour lying on this plane
- The weighting of the combination is given by the opacity of the foreground surface



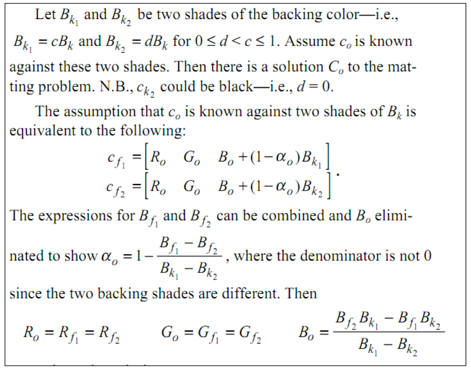
Triangulation Matting (Smith & Blinn)

- Instead of reducing the number of unknowns increase the number of equations
- Capture an image of an object against two known backgrounds of different colour
- Results in six equations and four unknowns
  - Three equations for each composite

Triangulation Matting Example



The Algorithm



Let B_{k_1} and B_{k_2} be two shades of the backing color—i.e., $B_{k_1} = cB_k$ and $B_{k_2} = dB_k$ for $0 \le d < c \le 1$. Assume c_o is known against these two shades. Then there is a solution C_o to the matting problem. N.B., c_{k_2} could be black—i.e., $d = 0$.

The assumption that c_o is known against two shades of B_k is equivalent to the following:

$$\begin{matrix} c_{f_1} = \begin{bmatrix} R_o & G_o & B_o + (1-\alpha_o)B_{k_1} \end{bmatrix} \\ c_{f_2} = \begin{bmatrix} R_o & G_o & B_o + (1-\alpha_o)B_{k_2} \end{bmatrix} \end{matrix}.$$

The expressions for B_{f_1} and B_{f_2} can be combined and B_o eliminated to show $\alpha_o = 1 - \frac{B_{f_1} - B_{f_2}}{B_{k_1} - B_{k_2}}$, where the denominator is not 0 since the two backing shades are different. Then

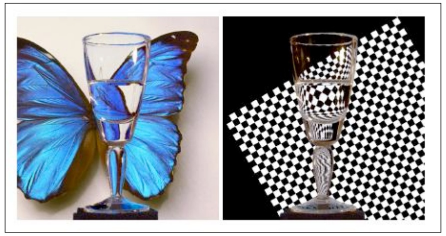
$$R_o = R_{f_1} = R_{f_2} \qquad G_o = G_{f_1} = G_{f_2} \qquad B_o = \frac{B_{f_2}B_{k_1} - B_{f_1}B_{k_2}}{B_{k_1} - B_{k_2}}$$

Smith, Blinn. Blue screen matting. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH '96). ACM

Problems with Matting

- Images do not look realistic
- Lack of Refracted Light
- Lack of Reflected Light
- Composited objects do not affect the image

Environmental Matting



http://grail.cs.washington.edu/projects/envmatte/

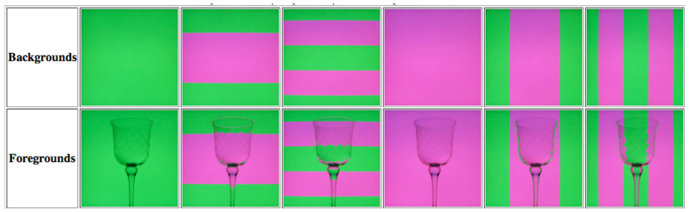
Environment Matting and Compositing Zongker, Werner, Curless, and Salesin. SIGGRAPH 99, pages 205-214, August 1999

Environment Matting Equation

- C = F + (1- α)B + Φ
  - C – composited colour
  - F – foreground colour
  - B – background colour
  - α – amount of light that passes through the foreground
  - Φ – contribution of light from the environment that travels through the object

Alpha holds values from zero to one…

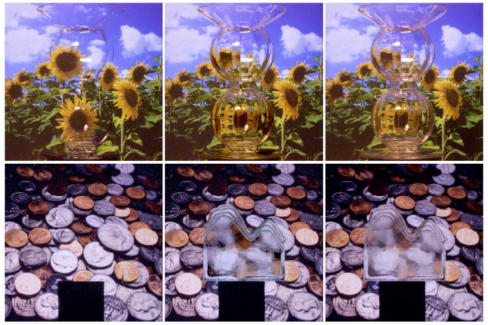
Environmental Matting



Ten backdrops (1 stripe to 256 stripes) for each alignment (vertical or horizontal), so each foreground object needed 20 pictures

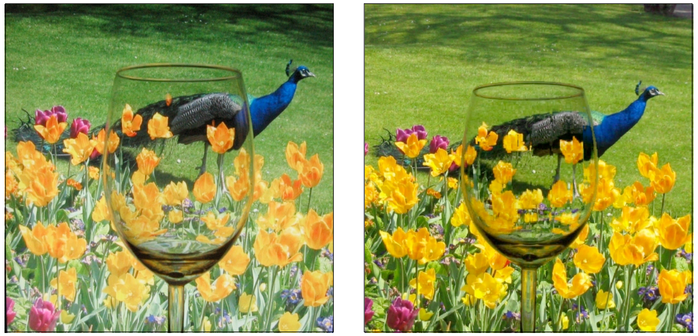
http://page.math.tu-berlin.de/~wotzlaw/ogweb/

Examples of Environment Matting



Alpha Matte Environment Matte Photograph

More Examples



More Examples