

Overview

Image matting is used everywhere. It's how Superman flies, or how weathermen point to hurricanes on maps. The technique of blue screen matting has been around for ages. Blue screen matting allows for a foreground object (e.g. Superman), to be removed from his background and pasted onto another background. The method is relatively simple: you know the background color (blue, but increasingly green in movies), so anything that isn't the same as the background is part of the foreground. The problem with this technique, however, is that it's all or nothing: either you're part of the foreground, or not.

The Matting Equation

$$C = C_o + (1-\alpha_o)C_k$$

C = composite [R,G,B] values

C_o = foreground [R,G,B] values

α_o = foreground transparency value

C_k = background [R,G,B] values

There's more trouble when we try to use general matting, however. What you're trying to obtain is the RGB and alpha (transparency) values for the foreground. But all you have is the single composite RGB values. That gives you four unknowns, but only three equations

There are a few ways around this. The first is to have the background be only blue ($RGB = [0,0,x]$), and the foreground contain no blue ($RGB = [y,z,0]$). The limitations here are very strict: no gray, no white, and no hues that contain any blue. This effectively reduces the the number of unknowns from 4 to 3, allowing for a single solution.

Another solution is to have the foreground be gray ($R=G=B$). This drops the number of unknowns from 4 to 2, making the system overconstrained (2 unknowns, 3 equations).

The triangulation method tries to increase the number of equations, rather than try to decrease the number of unknowns. [Smith and Blinn](#)'s method requires more photographs, but provides and added bonus: partial transparencies. Another plus is that the backgrounds do not have to be uniform: they can be anything. One stipulation: the camera has to be perfectly still and all the images must be perfectly aligned. For their method, two different backgrounds and four images are required: one of each of the backgrounds, and one of the foreground object in front of each of the backgrounds. The algorithm for solving this system of equations is detailed below.

Algorithm

For every pixel in each set of images, triangulation matting gives you the following:

$$\text{Backing color } C_{k1} = [R_{k1}, G_{k1}, B_{k1}]$$

$$\text{Backing color } C_{k2} = [R_{k2}, G_{k2}, B_{k2}]$$

$$\text{Composite color } C_1 = [R_1, G_1, B_1]$$

$$\text{Composite color } C_2 = [R_2, G_2, B_2]$$

This allows for six equations with only 4 unknowns. We can solve a matrix system of equations to obtain the values of the foreground RGB and alpha.

$$\begin{bmatrix} R_{1\Delta} \\ G_{1\Delta} \\ B_{1\Delta} \\ R_{2\Delta} \\ G_{2\Delta} \\ B_{1\Delta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -R_{1k} \\ 0 & 1 & 0 & -G_{1k} \\ 0 & 0 & 1 & -B_{2k} \\ 1 & 0 & 0 & -R_{2k} \\ 0 & 1 & 0 & -G_{2k} \\ 0 & 0 & 1 & -B_{2k} \end{bmatrix} \begin{bmatrix} R_o \\ G_o \\ B_o \\ \alpha_o \end{bmatrix}$$

Where $C_\Delta = C - C_k$

Results

Here's what the program produced!

Some images are pulled from other sources ([Smith & Blinn](#), [Miklos Bergou](#))



