2 Matting

The key ingredient for many of the previous operations is an alpha channel, permitting synthetic rendered objects to be convincingly placed in real footage or mixed with other rendered images. However, there are also many reasons to go the other way: composite real footage over synthetic images or other real footage. A prime example is inserting a real actor into an environment that couldn't feasibly exist in real life (or would be far too dangerous to put an actor in).

The biggest problem with this is that real footage doesn't come with an alpha channel. An alpha channel must be created for the footage, which is a difficult or even impossible problem in general, but which can be convincingly solved under the right conditions. This process is generally called *matting*, and the alpha images produced called *mattes*.

The general model we use for matting is that the photo is the result of compositing a foreground image over a background image: we want to separate the foreground image with an alpha channel from the background image, allowing the foreground to be used over a different background. Mathematically, considering one pixel and assuming we will use premultiplied alpha, this is just a restatement of equation (3),

$$\vec{C}_{\text{photo}} = \vec{C}_{\text{fore}} + (1-\alpha)\vec{C}_{\text{back}}, \tag{8}$$

to be solved for $\vec{C}_{\text{fore}}$ and α, with $\vec{C}_{\text{photo}}$ known.

Clearly this is massively ill-posed as it stands: this provides only three equations (one each for R, G, and B) but there are seven unknowns (foreground colours, alpha, and background colours). More information is needed to make progress.

As with all film work, one option is of course to call on the artist: ask them to manually paint the alpha channel. This is potentially very tedious pixel-level work, although good paint software and special heuristics which automatically identify obvious edges (where colours change significantly from one pixel to the next) can help significantly. Where the artist specifies $\alpha = 0$, the premultiplied foreground image also has to have $\vec{C}_{\text{fore}} = 0$, and where $\alpha = 1$ the foreground colour is identical to the photo. For intermediate values of α, where a photo pixel is a mix of foreground and background colours, estimates of the background colour taken from nearby pixels where $\alpha = 0$ can be used. However, it would be much better if a more automatic approach could be used — even if manual artist involvement might still be necessary on occasion to fix mistakes made by an algorithm.[1]

One of the most successful approaches is to gain information about the background colour. If the camera is stationary, for example, a photo can be taken of just the background before the actors or other foreground elements move into place. By identifying which pixels in the final photo are the same as the background photo, you can get a pretty good binary matte:

$$\alpha = \begin{cases} 0 & : \quad \vec{C}_{\text{photo}} \approx \vec{C}_{\text{back}} \\ 1 & : \quad \text{otherwise.} \end{cases} \tag{9}$$

This is sometimes called *background subtraction*. One obvious problem that can happen is if the foreground has many of the same colours as the background: if a person wearing a black shirt stands in front of a black background, for example, this method would erroneously report the shirt as part of the background. Another problem is that it's not at all obvious how to estimate alpha values between zero and one, which is crucial for properly antialiased edges.

A careful choice of background helps immensely. Instead of just using whatever happens to be there, a special screen of a single carefully chosen colour can be used — in particular, a colour that's unlikely to appear in

[1] One common case where an artist steps in is the creation of a *garbage matte*, a very rough matte for removing objects like calibration targets or microphones and the like from a photo, far away from the desired foreground. Here just a rough scribble over the "garbage" is all that's needed, to direct subsequent automatic algorithms to ignore that part of the image in case they would get confused by it.

the foreground image, and which is nice and bright.[2] "Blue screens" and especially "green screens" are very popular, since human flesh and hair is unlikely to be either of these colours, and most clothes (apart from certain superhero costumes!) also eschew bright saturated blues or greens. With a uniformly-coloured screen as the background, $\vec{C}_{\text{back}}$ is known everywhere — even if the camera is moving. With that, the compositing equation (8) reduces to a system of linear equations which we can write in matrix-vector form:

$$\begin{pmatrix} 1 & 0 & 0 & -R_{\text{back}} \\ 0 & 1 & 0 & -G_{\text{back}} \\ 0 & 0 & 1 & -B_{\text{back}} \end{pmatrix} \begin{pmatrix} R_{\text{fore}} \\ G_{\text{fore}} \\ B_{\text{fore}} \\ \alpha \end{pmatrix} = \begin{pmatrix} R_{\text{photo}} - R_{\text{back}} \\ G_{\text{photo}} - G_{\text{back}} \\ B_{\text{photo}} - B_{\text{back}} \end{pmatrix}. \tag{10}$$

We now have three linear equations for just four unknowns — still ill-posed, but much more tractable. We just need to close up the system somehow to make it well-posed, at which point we can get the matte.

One ingredient we haven't mentioned yet is the intrinsic limits on the unknowns. With any interpretation of alpha — opacity, coverage, probability — reality demands that $\alpha \geq 0$ and $\alpha \leq 1$. Similarly, each RGB channel has to be nonnegative, though we can't a priori assume they are bounded by anything (light can't be negative, but it can be arbitrarily bright). These inequalities aren't in general enough to solve the problem, but they can help a lot in narrowing down the possibilities. The space of solutions to the system (10) is an infinite line in 4D space, but with the inequalities applied, it is restricted to a finite line segment.

To arrive at a solution, some heuristic equation needs to be incorporated. Exactly what equation to choose might depend on the scenario. Many foreground images may adequately satisfy some further constraint. For example, some analysis on typical foreground images might show an average

[2] Note that it's the colour of the screen in the image which is important: making it appear to be uniform and bright, without any shadows cast from the foreground, may require special efforts in setting up lights.

relationship between the R, G, and B channels, such as:[3]

$$0.3461 \cdot R_{\text{fore}} - 0.8213 \cdot G_{\text{fore}} + 0.4535 \cdot B_{\text{fore}} = 0. \tag{11}$$

Adding this equation to the system provides a 4×4 matrix which hopefully is invertible (it will be as long as the background colour is not black, so the fourth column of the matrix in equation (10) is not all zero); the linear system can then be inverted. Of course, this last equation is just a heuristic, so if it turns out the solution violates any of the inequalities, the nearest point satisfying the inequalities and the original three equations should be used instead.

Research into matting continues to this day: see at least the paper by Smith and Blinn [SB96] as a foray into the literature.

[3]This equation was derived by taking a portrait image of myself and using the Singular Value Decomposition to find the 3D vector most orthogonal to all pixel RGB values, i.e. the smallest singular vector

References

[Deb98] Paul Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 189–198, New York, NY, USA, 1998. ACM.

[SB96] Alvy Ray Smith and James F. Blinn. Blue screen matting. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 259–268, New York, NY, USA, 1996. ACM.