Alpha and the History of Digital Compositing

Technical Memo 7

Alvy Ray Smith
August 15, 1995

Abstract

The history of digital image compositing—other than simple digital implementation of known film art—is essentially the history of the alpha channel. Distinctions are drawn between digital printing and digital compositing, between matte creation and matte usage, and between (binary) masking and (subtle) matting. The history of the *integral alpha* channel and *premultiplied alpha* ideas are presented and their importance in the development of digital compositing in its current modern form is made clear.

Basic Definitions

Digital compositing is often confused with several related technologies. Here we distinguish compositing from printing and matte creation—eg, blue-screen matting.

Printing v Compositing

Digital film printing is the transfer, under digital computer control, of an image stored in digital form to standard chemical, analog movie film. It requires a sophisticated understanding of film characteristics, light source characteristics, precision film movements, film sizes, filter characteristics, precision scanning devices, and digital computer control. We had to solve all these for the Lucasfilm laser-based digital film printer—that happened to be a digital film input scanner too. My colleague David DiFrancesco was honored by the Academy of Motion Picture Art and Sciences last year with a technical award for his achievement on the scanning side at Lucasfilm (along with Gary Starkweather). Also honored was Gary Demos for his CRT-based digital film scanner (along with Dan Cameron). *Digital printing* is the generalization of this technology to other media, such as video and paper.

Digital film compositing is the combining of two or more strips of film—in digital form—to create a resulting strip of film—in digital form—that is the composite of the components. For example, several spacecraft may have been filmed, one per film strip in its separate motion, and a starfield may have also been filmed. Then a digital film compositing step is performed to combine the separate spacecrafts over the starfield. The important point is that none of the technology mentioned above for digital film printing is involved in the digital compositing process. The separate spacecraft elements are digitally represented, and the starfield is digitally represented, so the composite is a strictly digital computation. *Digital compositing* is the generalization of this technology to other media.

This only means that the digital images being combined are represented in resolutions appropriate to their intended final output medium; the compositing techniques involved are the same regardless of output medium being, after all, digital computations.

No knowledge of film characteristics, light sources characteristics, film movements, etc. is required for digital compositing. In short, **the technology of digital film printing is completely separate from the technology of digital film compositing**. The technology of digital film scanning is required, perhaps, to get the spacecrafts and starfield into digital form, and that of digital film printing is required to write the composite of these elements out to film, but the composite itself is a computation, not a physico-chemical process. This argument holds regardless of input or output media. In fact, from hereon I will refer to film as my example, it being clear that the argument generalizes to other media.

Matte Creation v Matte Usage

The general distinction drawn here is between the technology of *pulling mattes*, or *matte creation*, and that of *compositing*, or *matte usage*. To perform a film composite of, say a spacecraft, over, say a starfield, one must know where on an output film frame to write the foreground spacecraft and where to write the background starfield—that is, where to expose the foreground element to the unexposed film frame and where to expose the background element. We will ignore for the moment, for the purpose of clarity, the problem of partial transparencies of the foreground object that allow the background object to show through partially.

In classic film technology, predating the computer by decades ([Beyer64], [Fielding72], [Vlahos80]), the required spatial information is provided by a (*traveling*) *matte*, another piece of film that is transparent where the spacecraft, for example, exists in the frame and opaque elsewhere. This can be done with monochrome film. It is also easy to generate the complement of this matte, sometimes called the *holdout matte*, by simply exposing the matte film strip to an unexposed strip of monochrome film. So the holdout matte film strip is placed up against the background film strip, in frame by frame register, called a *bipack* configuration of film, and exposed to a strip of unexposed color film. The starfield, for example, gets exposed to this receiving strip where the holdout matte does not hold out—that is, where the holdout matte is transparent. Then the same strip of film is re-exposed to a bipack consisting of the matte and the foreground element. This time the spacecraft, for example, gets exposed exactly where the starfield was not exposed.

Digital film compositing technology is, in its simplest implementation, the digital version of this process, where each strip of film is replaced with a digital equivalent, and the composite is done with a digital computation. Once the foreground and background elements are in digital form and the matte is in digital form, then digital film compositing is a computation, not a physico-chemical process. As we shall see, the computer has caused several fundamentally new

ideas to be added to the compositor's arsenal that are not simply simulations of known analog art.

The question becomes: Where does the matte come from? There are several classic (pre-computer) answers to this question. One set of techniques (at least one of which, the sodium vapor technique, was invented by Petro Vlahos [Vlahos58]) causes the generation of the matte strip of film simultaneously with the foreground element strip of film. So this technique simultaneously generates two strips of film for each foreground element. Then optical techniques are used, as described above, to form the composite. Digital technology has nothing new to contribute here; it simply emulates the analog technique.

Another technique called *blue-screen matting* provides the matte strip of film after the fact, so to speak. Blue-screen matting (or more generally, constant color matting, since blue is not required) was also invented by Petro Vlahos [Vlahos64]. It requires that a foreground element be filmed against a constant-color, often bright ultramarine blue, background. Then with a tricky set of optical and film techniques that don't need to concern us here, a matte is generated that is transparent where the the foreground film strip is the special blue color and opaque elsewhere, or the complement of this. There are digital simulations of this technique that are complicated but involve nothing more than a digital computer to accomplish.

The art of generating a matte when one is not provided is often called, in filmmaking circles, *pulling a matte*. It *is* an art, requiring experts to accomplish[1]. I will generalize this concept to all ways of producing a matte, and term it *matte creation*. The important point is that matte creation is a technology separate from that of compositing, which is a technology that *assumes* a matte already exists. In short, **the technology of matte creation is completely separate from the technology of digital film compositing.** Petro Vlahos has been awarded by the Academy of Motion Picture Arts and Sciences for his inventions of this technology, a lifetime achievement award in fact. The digital computer can be used to simulate what he has done and for relatively minor improvements. At Lucasfilm, my colleague Tom Porter and I implemented digital matte creation techniques and improved them, but do not consider this part of our compositing technology. It is part of our matte creation technology.

It is time now to return to the discussion of transparency mentioned earlier. One of the hardest things to accomplish in matte creation technology is the representation of partial transparency in the matte. Transparencies are important for foreground elements such as glasses of water, windows, hair, halos, filmy clothes, motion blurred objects, etc. I will not go into the details of why this is difficult or how it is solved, because that is irrelevant to the arguments here. The important points are (1) partial transparency is fundamental to convincing com-

[1] I have proved, in fact, in [Smith82b] that blue-screen matting is an underspecified problem in general and therefore *requires* a human in the loop.

posites, and (2) representing transparencies in a matte is part matte creation technology, not the compositing technology, which just uses the result.

Masking v Matting

The distinction to be made here is between simple-minded binary matting that we will call *bitmasking*—or *masking*, for short—and fully subtle blending for which we will reserve the respectful term *matting*. So masking is the special case of matting where there are only two possibilities: An image is either included or excluded from a final composite, no other possibilities allowed. Matting (the general case) allows mixtures of images at each point, one can show through the other with varying amounts of transparency.

The most simple-minded technique is to generate a *bitmask* ("binary mask", either 0 or 1, on or off, densely exposed or unexposed) to shield the exposed film from previously exposed elements. We shall call this a *mask* for short, and preserve *matte* for the general case. A (bit)mask is a simulation of the classic analog optical printer techniques described earlier, where we disregard partial transparencies. One can think of it as creating a very high contrast matte.

What the bitmask technique does not accomplish, however, is partial transparencies. In particular, no partial transparencies are provided at the edges of objects. For computer generated images, this results in what is known in the digital world as "jaggies" or "aliasing" at the edges of objects. In the early days it was often thought that a bitmask generated at sufficiently high resolution did not exhibit jaggies, or at least that they were invisible. But this is not true. I recall sitting in the Lucasfilm screening room in the early 1980s with Richard Edlund, Academy-Award winning special-effects director then at Industrial Light & Magic division of Lucasfilm, while he watched a submission from an outside computer graphics firm using the high-resolution trick. Richard instantly spotted the jaggies running along the edges of the elements—spacecraft, of course, at this time at Lucasfilm—and rejected the work or proposed work. They were very high-resolution jaggies, but still jaggies nevertheless.

Matting, on the other hand, preserves a range of transparencies at each point. We shall see in a moment that the digital representation of a matte came to be called an *alpha channel*, and the composition of two images with a full matte, or alpha channel, is sometimes called *alpha blending*. In computer circles, the "alpha" terminology thus became interchangeable with the "matte" terminology, even for those cases where the matte was not mathematically created by a computer rendering program.

Let's take care of one detail before continuing. To those practiced in the art of computer generated imagery, it is obvious how a mask can be created from a strictly mathematical definition of an element. A bitmask can be generated while the computer generated image element is being generated. A 1 is written in the mask at every pixel that holds an output pixel and a 0 is written everywhere else. A computer rendering program can also generate a matte, with say 256 or more levels of transparency, with almost as much ease as it can generate a mask, again

when an element is geometrically represented. In a sense this is, in either case, a digital simulation of the classic film techniques that generate matte films simultaneously with foreground element films. Generation of a mask or matte as an adjunct to computer generation of an image should be thought of as yet another technique for creating a matte and not part of the technology of compositing—which assumes a matte.

It is worth noting that, up to this point in our discussion, nothing really innovative has been introduced to the technology of composition by digital techniques. All digital techniques presented so far are simply obvious digital representations of known techniques. I have mentioned one contribution to matte creation technology, however, due to computers, and this is the generation of a (binary) mask or (full valued) matte simultaneously with the rendering of a 3D geometrical model of a film element into an image. Let's now turn our attention to true digital contributions to the technology of compositing.

The Invention of Alpha

Ed Catmull and I invented the notion of the *integral alpha* in the 1970s at New York Tech. This is the notion that opacity (or, equivalently, transparency) of an image is as fundamental as its color and should therefore be included as part of the image, not as a secondary accompaniment. To be very clear, we did not invent digital mattes or digital compositing. These were obvious digital adaptations of known analog techniques. We invented the notion of the alpha channel as a fundamental component of an image. We coined the term "alpha" for the new channel. We called the resulting full-color pixel an "RGBA" pixel.

Thus RGB images (Red, Green, Blue) became RGBA images (Red, Green, Blue, Alpha) in all work done by the Catmull/Smith team from that point forward, including Lucasfilm and Pixar. Red, Green, and Blue obviously are the three color channels of a full-color image, and Alpha is the transparency (equivalently, opacity) channel. The alpha channel typically contains as many bits as a color channel. So, for example, an 8-bit alpha channel can represent 256 levels of transparency, from 0 (completely transparent) to 255 (completely opaque). Or 10 bits can represent 1024 levels of transparency. The actual number of bits is immaterial to the technology of compositing (but it may affect the quality of digital printing tremendously). The RGBA image has been fundamental to New York Tech, Lucasfilm, Pixar, Altamira, and Disney and is broadly supported by the graphics community today.

Hundreds of thousands, if not millions, of images have been created in the last two decades with alpha channels. Many, many films have been made using them—all those of Lucasfilm and its special effects division, Industrial Light & Magic, after 1982 with digital elements, all those of Pixar after 1986, and all Disney animated films after 1990.

It is not hard to understand why no one had leapt to the concept of the integral alpha before we did. Recall that at the time memory was still very expensive.

Our first video framebuffer, 640x480x8 bits, cost $80,000 and the next five cost $60,000 each. So an RGB framebuffer cost us $200,000 and an RGBA framebuffer (with an alpha channel storage) cost $260,000 (in 1975 dollars)[2]. It was nontrivial to increase memory usage by 25%. And we were the only facility in the world that had 24-bit and 32-bit framebuffers, and one of only three or four places that had even an 8-bit framebuffer.

I can remember the moment of this invention very clearly. Ed Catmull was working on his sub-pixel hidden surface algorithm for SIGGRAPH paper submission (published eventually as [Catmull78]). He was generating images of objects using this technique over different backgrounds. I was working with him to make these pictures since I was the local expert on image file formats and knew where all of the interesting background images were stored in our file system. I would position an image in the framebuffer that he would then render over, using his new technique. The compositing would happen as the rendering occurred. As you can see, this was tedious. A different background required a new rendering, then a very slow process. Ed mentioned that it certainly would make life easier if, instead of re-rendering the same image over different backgrounds, he rendered the opacity information *with* the color information at each pixel into a file and *then* the file could be composited over different backgrounds without re-rendering, as it was read pixel-by-pixel from the file. I immediately said that this would be extremely easy to accomplish. I could say this confidently because I had written the image file saving and restoring programs that we used. I already had versions for saving and restoring 8-bit and 24-bit images, and I knew exactly how to write a version that saved and restored 32-bit images. I started right then and by the next morning had the full package, complete with Unix-style manual pages using the “alpha” and “RGBA” terminology, ready for use. All Ed had to do was write the alpha information—we called it that because of the classic linear interpolation formula $\alpha A + (1-\alpha)B$ that uses the Greek letter α (alpha) to control the amount of interpolation between, in this case, two images A and B—into a fourth framebuffer (we had six 8-bit framebuffers at New York Tech at this time). Then I would save the four framebuffers (Red, Green, Blue, and the new Alpha framebuffers[3]) into a file with the new code, called *savpa4*[4]. Then Ed or I or anybody could use the newly revised restore routine (called *getpa*) to composite the file image over an arbitrary image already in the frame-

[2] About $1 million in 1995 dollars!

[3] We called three 8-bit framebuffers ganged together an RGB framebuffer and four an RGBA framebuffer.

[4] The earliest dated documentation I have for this code is dated January 13, 1978. Ed was preparing for SIGGRAPH 78. SIGGRAPH typically has a paper due date of early January of the corresponding year, so this is probably about when the invention actually occurred although it might have happened in December 1977, to avoid the last minute crunch against the paper deadline. I was also preparing a paper for SIGGRAPH 78. The date on the submission is January 6, 1978, and the code I used to generate figures for the paper is dated December 28, 1977.

buffers. *getpa* would detect that the enclosed image had a fourth channel and use it to do compositing, as the image was read from the file. That was it. The integral alpha channel had been born. The "or anybody could use" above is a key phrase. The integral alpha channel severed the image synthesis step from the compositing step, and this changed how digital compositing was done forever. When we started Lucasfilm graphics later, it was RGBA from the outset. The original framebuffers there were RGBA and all software was written to honor RGBA.

In film terms, the alpha channel is exactly the matte needed to composite one image with another. As opposed to the simple bitmask, the alpha channel inherently supports partial transparencies. Given a matte with subtle transparencies—and recalling that compositing technology does not care how this matte was derived—the integral alpha approach stores this matte in the fourth channel of the foreground image that it serves to define the shape of. There is no "second strip of film" for the matte. In other words, the digital contribution here is not simply a simulation of the classic film techniques; it is something new: a single concept incorporating both color and transparency—much like the human mind perceives an object. In yet other words**, the matte ceases to exist conceptually**. An image partially exists at a point depending on itself alone, namely its alpha channel[5]. As we shall see, this slight conceptual change led to a sequence of further changes with profound effects upon the industry.

The notion of the integral alpha next led next to the notion of *premultiplied alpha* , and it led to a complete modeling of the human perception of an object. Tom Porter and Tom Duff of the Catmull/Smith team, now at Lucasfilm and Pixar, first drew the distinction between premultiplied and non-premultiplied alpha[6] and showed the relative benefits of premultiplied alpha. Another way to say this is that although we had added the integral alpha channel to our thoughts and computations and hardware, we still did not fully understand it until Porter and Duff wrote their classic paper[7].

Notice that the classic linear interpolation or compositing formula above[8] is equivalent to $\alpha A + B - \alpha B$. Notice that if A is premultiplied by α—that is, its colors are premultiplied by α—then one multiplication is removed from this for-

[5] To emphasize this point a bit further: Notice that we could think of an image as four separate entities: a red one, a green one, a blue one, and a transparency one. We have gained great conceptual ease by combining the red, green, and blue entities into a single colored thing. The alpha channel carried this one more step to reduce the mental load even further by adding transparency to the colored entity.

[6] They often call this *associated* and *unassociated* alpha, but I always forget which is which so prefer the more descriptive terms used here.

[7] In fact, I have only recently come to believe I have finally understood all the profundity of the alpha concept. See [Smith95] for details.

[8] The full argument is more complex than that presented here. It takes into account the partial transparencies of both images. The simplified argument here carries the gist, nevertheless. See [Smith95] for the full argument, or of course, the classic [PorterDuff84].

mula (actually three multiplications since this formula has to be applied to the three color channels). Thus Porter and Duff observed that a great many multiplies could be avoided at compositing time by figuring them into the image, as it was computed, to form an image with so-called premultiplied alpha. At the time, multiplies were very expensive and this was a large saving in computation time.

Thus premultiplied alpha is efficient. But it is more than efficient. It is as conceptually fundamental as the integral alpha to which it is intimately related. To see this, notice that the color channels of a completely transparent pixel must be 0. This is because premultiplication by 0 (recall that alpha 0 means transparent) must result in 0 colors. Once you have 0 colors and 0 alpha, any information about non-0 color that might have existed at the point previously is lost. Thus, for all practical purposes, **a transparent pixel ceases to exist conceptually**[9]. This is profound because suddenly images change from rectangular items to shaped objects with partial transparencies. This is largely what humans mean by a visual object. And compositing shaped image elements is how modern film compositing is done. There is no longer the notion of a traveling matte—a separate shape descriptor somehow synchronized with the thing being given shape. The shape is integral to the image. This is the legacy of the NYIT/Lucasfilm/Pixar group. All modern digital filmmaking is done this way—including all of Disney's blockbusters (eg, *Beauty and the Beast*, *Aladdin*, *The Lion King*, *Pocahontas*), Pixar's new movie *Toy Story*, and dozens of special effects in movies such as Paramount's *Star Trek II—The Wrath of Khan* (1984) and Amblin's *The Young Sherlock Holmes*.

It is important to notice that notions of resolution, logarithmic curves, and bit depth are not relevant to the technology of compositing; they do not enter into the discussion of compositing above at all. These are notions about scanning and printing, not compositing.

The digital compositing technology founded on the alpha channel fundamentally supports subtle transparency. The Catmull/Smith team has never resorted to the bitmask (binary masking) technology often argued in the early days to be sufficient if high resolution images were used.

The Catmull/Smith team made profound contributions to digital compositing, that are standard in today's digital filmmaking world. Nobody in the computer graphics world invented digital compositing, an obvious simulation of known film techniques. The Catmull/Smith team invented the integral alpha (Catmull and Smith) and premultiplied alpha (Porter and Duff), concepts that are not simulations of known film technique but true additions to the art and science of compositing made possible by the computer and the basis of nearly all modern digital compositing.

---

[9] They may still occupy memory space, but this is only convenient. There is no need for memory space to be allocated if an appropriate storage model is provided. That is, the information stored in those pixels, if any, is never used.

Summary of Points

Here is a summary of points argued in this paper:

- Digital compositing is distinct from digital printing.
- Digital compositing is distinct from digital matte creation.
- Binary masking is distinct from full matting, or alpha blending.
- Digital simulation of known film compositing techniques is easy and obvious, and therefore not a contribution.
- The integral alpha channel and premultiplied alpha are fundamentally new compositing concepts, intrinsically supporting full matting, that are due to the computer, and they have become the basis of essentially all modern digital compositing.

References

[Beyer64] Beyer, Walter, *Traveling Matte Photography and the Blue Screen System*, **American Cinematographer**, May 1964, 266. The second of a four-part series. Pre-digital technology.

[Catmull78] Catmull, Edwin, *A Hidden-Surface Algorithm with Anti-Aliasing*, **Computer Graphics**, Vol 12, No 3, Jul 1978, 6-11. SIGGRAPH'78 Conference Proceedings.

[Fielding72] Fieldling, Raymond, **The Technique of Special Effects Cinematography**, Focal/Hastings House, London, 3rd edition, 1972, 220-243. Pre-digital technology.

[PorterDuff84] Porter, Thomas, and Duff, Tom, *Compositing Digital Images*, **Computer Graphics**, Vol 18, No 3, Jul 1984, 253-259. SIGGRAPH'84 Conference Proceedings.

[Smith78] Smith, Alvy Ray, *Color Gamut Transform Pairs*, **Computer Graphics**, Vol 12, No 3, Jul 1984, 12-19. SIGGRAPH'78 Conference Proceedings.

[Smith82a] Smith, Alvy Ray, *Analysis of the Color-Difference Technique*, Tech Memo 30, Computer Division, Lucasfilm Ltd., Mar 1982.

[Smith82b] Smith, Alvy Ray, *Math of Mattings*, Tech Memo 32, Computer Division, Lucasfilm Ltd, Apr 1982. Reissue of tech memo of Dec 30, 1980.

[Smith95] Smith, Alvy Ray, *Image Compositing Fundamentals*, Tech Memo 4, Microsoft, Jun 1995. The argument for sprites and premultiplied alpha becomes rock solid.

[Vlahos58] Vlahos, Petro, *Composite Photography Utilizing Sodium Vapor Illumination*, US Patent 3,095,304, May 15, 1958. A two-film technique that generates a matte strip with an element strip.

[Vlahos64] Vlahos, Petro, *Composite Color Photography*, US Patent 3,158,477, Nov 24, 1964. The classic color-difference blue screen compositing technique.

[Vlahos78] Vlahos, Petro, *Comprehensive Electronic Compositing System*, US Patent 4,100,569, Jul 11, 1978. The (analog) electronic equivalent of blue-screen—the UltiMatte.

[Vlahos80] Vlahos, Petro, *Traveling Matte Composite Photography*, American **Cinematographer Manual**, American Society of Cinematographers, Hollywood, 5th edition, 1980, 530-539.