# The αδγ's of Digital Media Convergence

Alvy Ray Smith
Microsoft Corporation
One Microsoft Way, 16/2356
Redmond WA 98052-6399, USA

Phone: 425-703-2185 Fax: 425-703-2015
e-mail: alvys@microsoft.com
http://www.research.microsoft.com/research/graphics/alvy/

*Abstract*

There is no theoretical roadblock obstructing the integration of different media types into a single digital medium—after all, bits are bits—but there are several real problems hindering the so-called digital convergence. The alpha problem is that between premultiplied and non-premultiplied alpha. The gamma problem concerns the nonlinearity that many of today's applications insist on burning into their image data. The delta problem is about the integration of the discrete and the continuous—eg, samples (pixels) and geometry. The subtleties of these are explored—eg, "square pixels" and non-rectangular images—and a current example of how wrong things can get—the US digital television transmission formats battle—is elaborated.

*Keywords: Digital Convergence, Integral Alpha, Premultiplied Alpha, Gamma Correction, Color Matching, Square Pixel Spacing, Digital TV, Progressive Scan.*

## Vision: The Single Creative App

A realizable vision that captures the digital media convergence is The Single Creative App. It might actually not be a single application, but to the user it appears to be one.

Any creative person utilizing the various forms of digital media today knows the nightmare resulting from one or more applications per media type plus file conversions between them. For example, a project might require a 3D modeling program, a 2D paint program, a 3D rendering program, a 2D drawing program, a sound program, an editing program, etc. Since these have historically risen as separate applications, created by distinct companies, or distinct groups within a company, they do not naturally know about one another, necesitating file conversions and other energy losses to mere friction.

The Single Creative App supplies one creative space to its user that seamlessly integrates 2D and 3D, the discrete and the continuous (for pictures, this means sampling-based and geometry-based pictures), sound and pictures, animation and interactivity.

A model that realizes the vision is presented after discussion of the alpha problem, because it is the profundity of the premultiplied alpha concept that makes the model possible.

## Alpha (α)

The concept of the *integral alpha* channel—eg, a fourth channel integral to each image pixel in addition to its red, green, and blue color channels—has been with the computer graphics community since 1977 [4]. This simple idea was augmented by the notion of *premultiplied* and *non-premultiplied alpha* in 1984 [2]. Neither it nor the notion of integral alpha were appreciated by their inventors for the ramifications implied.

The integral alpha channel reduced mental baggage by obviating the need for a separate entity called the matte. It is important because it permitted subdivision of a monolithic 3D rendering problems into lesser renderings which could later be composited simply in 2D.

Premultiplied alpha—the notion that the color channels of each pixel are premultiplied by the alpha channel of that pixel—was originally just a technique for dramatically decreasing the number of multiplies required for compositing, a requirement at the time when multiplies were so expensive.

The ***alpha problem*** is the confusion of the two types of alpha. Another form of it confuses the continuous with the discrete and is non-integral. Both forms are roadblocks to media convergence.

Some 2D imaging programs today continue to require the separate baggage of a geometrically defined alpha (a "path"). This made sense when memory was expensive because a geometric description is nearly always more concise than one defined by an array of samples. On the other hand, it is nearly always less subtle than an image-defined alpha.

The confusion of premultiplied and not premultipled alpha is the more difficult to eradicate. Although the 3D computer graphics community almost universally uses premultiplied alpha (what Porter and Duff called associated alpha), the 2D imaging world, particularly on personal computer platforms often uses the non-premultiplied variety.

**Overthrowing the Tyranny of the Rectangle**

The profundity of premultiplied alpha follows from the fact that a completely transparent pixel, with $\alpha = 0$, must have color channels also 0 in the premultiplied case. Inability to divide by 0 precludes ever recovering the color of a transparent pixel in this case. Thus, for all practical purposes, ***a transparent pixel ceases to exist***. Memory need never be allocated for transparent pixels.

What does this mean? Most importantly, it means that images do not have to be rectangular. To state it positively, ***images with integral premultiplied alpha have shape***. Shaped images are called *sprites* to emphasize the distinction.

Contrast this with the non-premultiplied alpha case. Since the color channels of a transparent pixel can hold any color, it is normal to think of a shape defined by the alpha channel as temporary, or not for real. The real image is rectangular—as we've all grown up believing anyway—and the truth can always be retrieved by simply setting the alphas all back to 1. So the alpha channel in this case appears to be integral but it is really just the separate entity occupying integral image space. That is, the color part is conceptually a rectangle, and the alpha part is a changeable shape—two separate notions.

Most imaging applications today are still written to the old rectangular mindset, and this greatly hampers the convergence of geometry and imaging. Let's see why.

The rectangular worldview says that the workspace is a rectangle, with edges, holding a rectangular image. A user "falls off" the edges (is cropped by them). Compositing is accomplished by the conceptual baggage of a set of "layers" holding the images to be composited over a special "background". The layers too are rectangles in register with the layer holding the background, with edges too. To change order, images are reslotted into different layers.

Contrast this with the world of a 2D drawing or illustration application. A conceptually infinite workspace (sometimes called a "desktop") contains several floating geometric objects that have shape (of course), can be moved about freely, grouped together in subsets or hierarchies, and have front to back order that is easily altered. The workspace is inaccessible and certainly isn't a geometric object.

Premultiplied alpha lets us use exactly this same model for shaped images, or sprites. They are shaped entities, just like 2D geometrical objects. They can float over an infinite "void" or workspace. It is not a special rectangular background image; it is just not there; it's not geometry nor an image. The sprites can be in any depth order and easily changed, with exactly the same interface as for 2D drawing objects. They can be grouped the same way. There is no preferred background sprite. (Of course, one can always use a rectangular sprite that way, but it is never a requirement.) The baggage of layers is unnecessary. There are no edges to fall off.

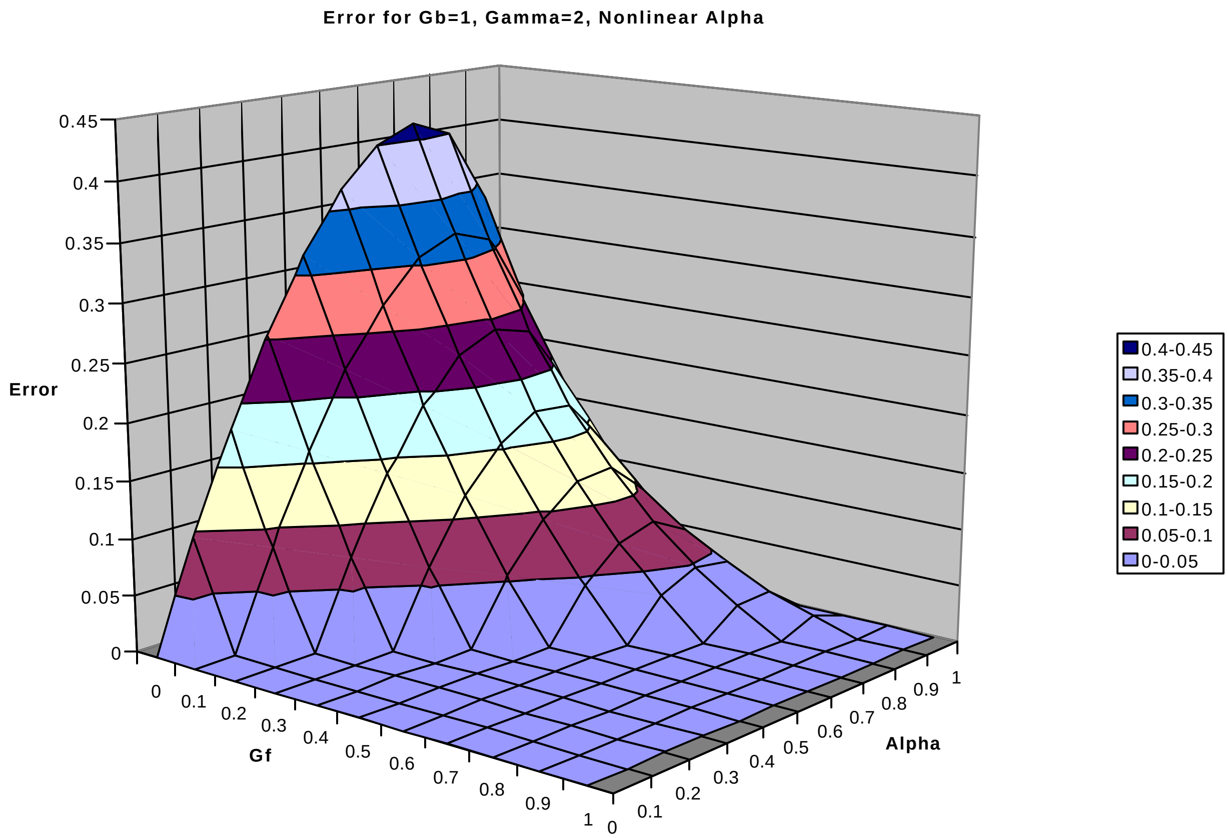
Most importantly, there is no longer any reason not to mix the media types in one creative space. The objects floating can be geometrically defined or sample defined. So this model, built on premultiplied alpha, accomplishes the true convergence of 2D geometry and 2D imaging.

That was the hard step in realizing The Single Creative App. Then it becomes easy—conceptually anyway—to add the third dimension for 3D, the fourth for time, to add sound, etc. There are still tricky issues—like the fact that the usual coordinate system used by imaging applications is not that typically used by 3D systems or 2D geometric ones—but these are not show-stoppers.

**Gamma ($\gamma$)**

The computer graphics community almost invariably assumes linear pixels—ie, that the numbers in the color and alpha channels are linear entities. For example, half red plus half red equals full red. But real display devices are notoriously nonlinear. Luckily, nonlinearity of the very common CRT-based video display can be described accurately enough with a single exponent, traditionally called gamma. Of course, different displays have different gammas. The computer graphics community has understood this for decades and compensated for it on output to a display by some gamma correction process. Stated another way, computations on images are assumed to occur in linear space. Antialiasing is a technique, for example, that relies on this.

But ordinary human beings, like the typical customers of personal computer applications, don't understand gamma and don't want to. The ***gamma problem*** arises because of this: Gamma is simply ignored in major personal computer imaging applications! More accurately, a single value of gamma is assumed (eg, 2.2, but typically one doesn't know and can't query) and this is "wired into" all images.




But the algorithms used by these applications assume linear data. The obvious solution of "degamma correction" doesn't work. Since almost universally these applications use 8-bit channels (24 for color and 8 for alpha if they have alpha), the correction of the nonlinear data back to linear throws away 1-2 bits per channel, and this is visible. Many applications simply compute on the nonlinear data as if it were linear! The surprising thing is that nobody seems to notice. But this is before our attempt to converge the 2D imaging world and the 3D modeling and rendering world (ie, 3D geometry world).

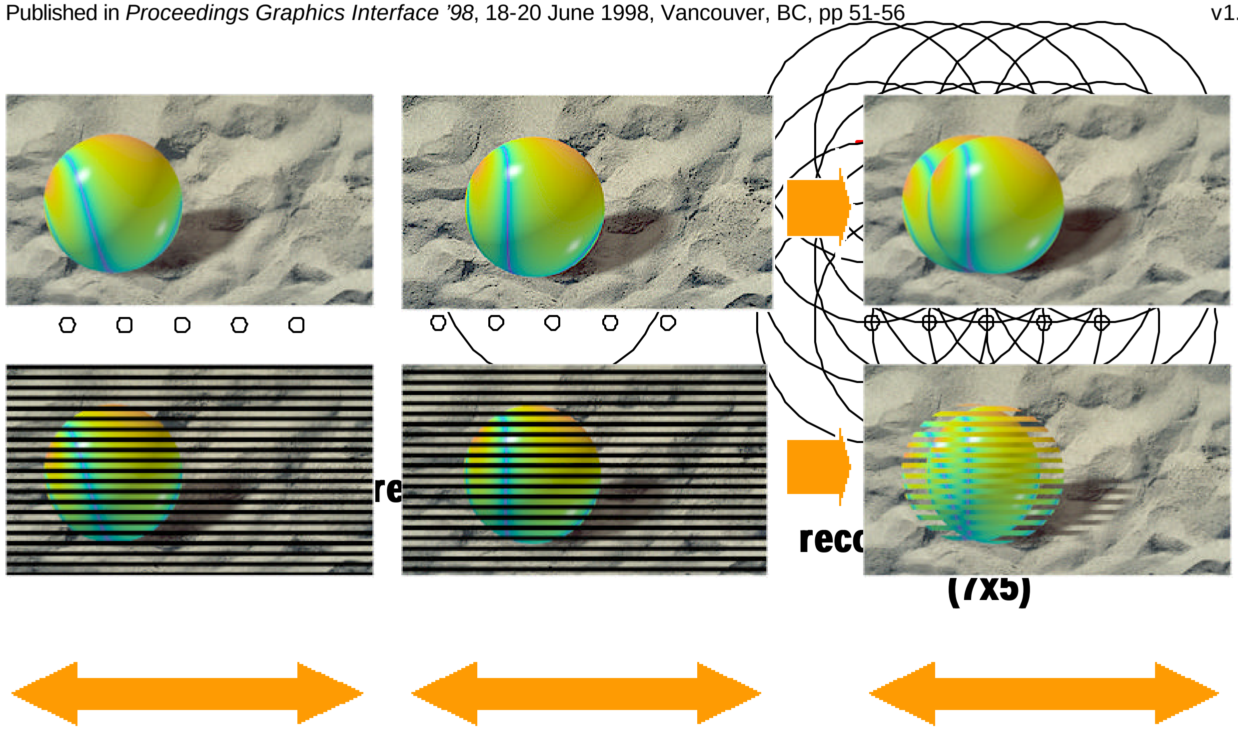
The figure above shows the results of a spreadsheet exercise on the common lerp (linear interpolation) operator. It plots the error, for operand pairs *Gf* and *Gb*, between lerp on nonlinear operands vs linear operands. A gamma of 2.0 is assumed for computational convenience. The lesson is that the worst-case error is 41%! This occurs for a black object over a white one. See [1] for an excellent analysis.

Apparently the only solution is to convert all imaging applications to 16-bit channels, which do have enough headroom for loss of bits in the nonlinearity conversions. At this writing, such doubling of memory is still not economically, or politically, realistic in the personal computing world.

**Color Matching**

The gamma problem is a special case of the broader *color matching* problem, sometimes called the *color constancy* or *color correction* problem. Some display devices, such as ink on paper printers, have very nonlinear colorspaces which cannot be simply described with a single exponent. The general problem is to supply nonlinearity corrections so that input colors match display colors match output colors, regardless of which input devices (eg, scanner, digital camera), which display devices (eg, CRT, liquid crystal, plasma, digital mirror devices), or which output devices (eg, printer, film, video) are used. And, of course, this must be invisible to the user since it is to hard to understand.

Missing from this usual description of the color matching problem is perhaps the most important colorspace, the internal or computational colorspace. The model proposed here assumes this colorspace is linear, so there must also be nonlinearity corrections between it and the input, display, and output colorspaces. And they must be fast! One of the main reasons software developers are still ignoring some of the color matching solutions available is that they are simply too slow.



**Delta (δ)**

One of the "fundamental tenets" of the vision espoused here is that the continuous and the discrete are equally important and equally supported. This most obviously means that image-based and geometry-based picture making are equally supported, but it also applies to discrete and continuous sound, animation, and interaction. The ***delta problem*** results from confusion at the boundary between the discrete and continuous domains, even among computer graphics sophisticates. Picturing will be used as the example here.

Historically, anyway, geometry specialists have tended to think of the rendering of their elaborate models into pixels as the plumbing at the end of the process. Imaging specialists have believed that nothing serious, with the richness of the real world, could be pictured with geometry. Hopefully, these attitudes are a thing of the past, but this is the context for the stress on equal importance.

To realize The Single Creative App, there must be a single model marrying the continuous and the discrete. There is. It is called the Sampling Theorem, but it is often subverted. The term "square pixels" is nearly always a red flag indicating the delta problem. ("Delta", by the way, is taken from the delta function used to sample continuous, but filtered, functions according to the Sampling Theorem.)

**A Pixel Is *Not* a Little Square!**
**A Voxel Is *Not* a Little Cube!**

Computer graphics would not be where it is today if its geometers had not modeled the pixel as a little square, a simplifying assumption that made rendering possible, especially in the early days of very slow machines. One form of the delta problem is the identification of this simplistic model of *the contributions to* a pixel with the pixel. In sampling theory terms, the mistake is confusion of the (dumb) box filter footprint with the sample taken. So to be very clear, a pixel is a sample (or a tuple of samples) and its geometry—if it's to be forced to have one—is simply a point, regardless of dimension.

The figure above illustrates how far from typical imaging practice the little square falls. An ordinary cubic filter is used for reconstruction of a set of samples into a continuous entity. Notice that the reconstructed entity is not rectangular (ie, where it departs from 0) and the footprint of no filter is a square. Furthermore the areas under each filter overlap highly. See [3] for full details.

Besides the little square model from geometrical computer graphics, another strong influence on people, seeming to enforce the notion of pixels as little squares, is video magnification. When one magnifies a screen of pixels, by 4 say, a field of little squares is displayed. But each square is not a visual magnification of the underlying pixel (which is just a point) but rather the representation of a magnification obtained by replicating the sample 4 times in both dimensions. The human eye integrates an array of 4x4 pixels of the same color, each spread by the cathode ray beam, into a little square. It is this array one sees, not the pixel up close.

Symptoms of the delta problem are such expressions as "the edge of the pixel" or "the center of the pixel". This appears in a recent image file specification, for example. The problem is often disguised as the question of where to place the "centers" of the pixels, on the integers or on the half-integers. Although it makes no difference where the sampling grid is located, so long as it is consistently placed, the sheer existence of this "problem" implies the questioner is seeing little squares. The figure shows that when thinking properly, one never asks about the half-integers. It is like asking should a matrix be indexed by integers or half-integers.

It probably goes without saying that to converge the discrete and continuous, a single model must be used. Luckily there is a very serviceable and respected model available, provided by the Sampling Theorem.

In digital TV, to be discussed next, the term "square pixels" is often misused to mean uniform sampling in both dimensions. Many in the business are now using the more appropriate term "square pixel spacing" to imply this meaning.

**The Digital TV Wars**

It is instructive to see what problems of a non-technical nature can be introduced into what is a straightforward technical issue. At this writing, the Digital TV (DTV) Wars rage in the US over what video formats should be used as the transmission standard in the new national digital TV system. The most contentious issue is progressive vs interlaced scanning.

Although the computer graphics community is very aware of the difference, the figure above is included to illustrate a naming problem in the wars. It illustrates the difference between the two scanning order proposals as well. The top row represents progressive scanning, the second interlaced. The left two columns represent two successive $60^{th}$s of a second. The right column shows what the eye integrates over a $30^{th}$ of a second. The Progressives call their system 720p because it presents 720 lines, successively (or progressively) scanned down the screen every $60^{th}$ second. The eye integrates two complete frames every $30^{th}$ second. The interlaced system presents 540 lines every $60^{th}$ second, every other line, then the missing 540 lines the next $60^{th}$ second. The Interlacers add these numbers together and call their system 1080i. The eye attempts to integrate these two "torn" fields every $30^{th}$ second.

For computer graphicists and those practiced in video recording of computer graphics, the thought that interlaced scanning could be done away with is uplifting. Since personal computers decided over a decade ago to go with progressive scanning—to make text readable—any simple convergence of television and computing—certainly part of our vision—demands progressive scanning be adopted.

Interlaced scanning was adopted about 50 years ago as a means of spacetime compression of the given analog signal into the given TV channel bandwidth. It was a clever solution then, but now we have much superior spacetime compression schemes for the digital domain. So it is a surprise to find that the US very nearly adopted (and may still adopt) an old-fashioned compression scheme for its supposedly modern digital TV system. The problem comes from people steeped in analog whose understanding of digital is only sufficient to digitize the analog process as they currently understand it. This thinking with "analog bits" brings us interlaced scanning again.

It is not difficult to argue successfully that progressive scanning wins over interlaced scanning in *any* technical sense. It also wins in economic arguments, when consumer economics are considered. The problem is that neither of these valid argument domains is paramount in the Wars. It is the sunk cost (billions of dollars) in research of very large companies into interlaced scanning formats and equipment that apparently drives the debate, not what is good or right.

An example of the technical disinformation being used is illustrated in the figure. As already mentioned, the Interlacers' system is called by them 1080i and the Progressives' system is called 720p. These are the two that are most nearly matched, but many non-technical executives and congressman have been lead to believe that they can ignore the *i* or *p* suffix and merely judge the resolution of the systems by looking at the prefix number. This "logic" leads to the belief that the 1080i system is truly "high defintion" while the 720p system is not. The figure shows that using the same reasoning as employed by the Interlacers that leads to 1080i tells us that the fair name for 720p is 1440p, alternatively that 1080i should be renamed 540i. One thing is clear, for still pictures 540 lines is less than 720 lines every $60^{th}$ of a second, and nobody buys TVs for still pictures. There is a perceived increase in resolution above 540 lines caused by interlacing, but it is only sufficient to raise the effective resolution to 600-650 lines of equivalent progressively scanned video, still less than 720.

The part of the technical argument that is never presented is the effect of compression on resolution. The 540i (aka 1080i) system and 720p systems both have to be compressed by about 50:1 to fit in the allotted digital broadcast channel. This is a terrific compression ratio that wipes out the high resolution available in the source. A 480p system, with a wide aspect ratio, has also been proposed by the Progressives. It requires an 18:1 compression which is much less severe. And the system is much more affordable than so-called "high definition" systems, both for consumers and producers.

This battle will be decided, in the short run anyway, by the large broadcasters who must soon write their checks for digital TV equipment to meet Congress's requirement of DTV broadcasts by summer 1999 if they are to hold on to their free slices of the digital broadcast spectrum. They must make these decisions in a highly charged environment where politics is more important than technology or consumer needs. Chances are high that there will be a mix of "standards" adopted, an oxymoron that will take years to settle. Unnecessarily and expensively.

**Conclusion**

The Single Creative App vision is within reach. A model that actually works to converge different media types has been presented. There are several simple but nasty technical problems to be resolved. And politics can be a substantial roadblock as well.

**References**


[1] BLINN, J. A Ghost in a Snowstorm. *IEEE Computer Graphics and Applications* 18, 1, (January-February 1998), pp. 79-84.

[2] PORTER, T. and DUFF, T. Compositing Digital Images. Proceedings of SIGGRAPH 84 (Minneapolis, Minnesota, July 23-27, 1984). In *Computer Graphics* 18, 3 (July 1984), pp. 253-259.

[3] SMITH, A. R. A Pixel Is *Not* a Little Square! A Pixel Is *Not* a Little Square! A Pixel Is *Not* a Little Square! (And a Voxel Is *Not* a Little Cube). Technical Memo 6, Microsoft Corporation, June 1995.

[4] SMITH, A. R. Alpha and the History of Digital Compositing. Technical Memo 7, Microsoft Corporation, August 1995.