Computer Graphics in Court: The Adobe/Quantel Case

Richard L. Phillips
RL Phillips Consulting Inc.

The Adobe/Quantel case that Dick Phillips discusses below is especially interesting and some of the landmark paint programs he mentions are being demonstrated in the '70s equipment exhibit on the SIGGRAPH 98 show floor.

— Carl Machover

Introduction

By SIGGRAPH reckoning, computer graphics is 25 years old. Although the field is older than that, a quarter of a century has nonetheless proven to be ample time to involve computer graphics in one of America's favorite pastimes, litigation. Interestingly enough, many technologies that have been around since the early days of graphics have only recently become the subjects of court action. Who would have thought that old war horse topics like display list traversal, three dimensional viewing algorithms and font definition technology would have been the subject of litigation in just the last few years?

In fact, it was as recent as 1996 that digital paint systems became the subject of a lawsuit. This is surprising, especially when you consider that [Smith97] points out that the first digital paint program can be traced back to 1969. But it was indeed in January 1996 that Adobe Systems Inc. was sued by Quantel Ltd. for alleged infringement of five of their patents by Adobe's Photoshop product. The stakes were huge; Quantel was seeking damages of $138 million, to be trebled if willful infringement was determined. Moreover, Quantel was seeking an injunction to stop Adobe from selling Photoshop.

I was hired by Adobe as an expert to help out with various aspects of the lawsuit. It is the story of the case, of the ensuing trial and the role played by computer graphics history that I'll relate in this article.

The Patents

Quantel Ltd. is a British media and communications company that specializes in providing digital effects hardware to the television and movie industries. One of Quantel's early products is Paintbox, a system that many SIGGRAPH exhibition attendees may have seen in the late 1980s. Among other things, a Paintbox user can do digital painting and cut-and-paste-type of digital compositing. From 1985 to 1995, Quantel was issued five United States patents covering various ways of using computers for graphics artwork, especially painting and compositing. For those who are interested, the patent names and numbers are cited below, but I'll shortly summarize the relevant aspects of the patents.

Before I jump into details, it's important to know that the date of issue of a patent is not necessarily pertinent in determining whether there was relevant prior art, i.e. somebody thought of the idea earlier and the patent should never have been issued. What's important here is that Quantel first received equivalent British patents, and a reciprocity agreement observed by most major countries endows the U.S. patent with a priority date equal to the date of filing in the U.K. Thus, the priority date is December 4, 1980 for the two earliest patents in this case. That means only work that was known, published or in use before that date is relevant prior art.

The rest of this article will focus mostly on those two early patents, cited as [Walker93] and [Walker94]. Together they describe aspects of a digital paint system whose characteristics will be familiar to anyone who has used one of several commercially available paint systems. But wait! If you know something about Quantel's Paintbox or read the patents in detail, you'll observe that Paintbox uses dedicated special purpose hardware while Adobe's Photoshop is a software package designed to run on a wide range of general-purpose computers. How can there be any issue of Photoshop infringing Quantel's patents? Well, depending on how the patent and its claims are drafted, the law doesn't necessarily recognize the hardware/software distinction. In this case Quantel argued that because Photoshop performs the same operations described in the patents, the fact that it does so entirely in software is irrelevant. So, on with the story.

The Painting Patents

As I mentioned earlier, I'll focus mainly on the [Walker93] and [Walker94] patents, which together I'll refer to as "the painting patents." From these patents I've distilled the following six elements which represent the essence of the claims asserted against Adobe:

1. A pressure-sensitive pen (stylus) and tablet that mimics an artist's paint brush.
2. A soft-edge brush shape (more on this later).
3. A background image (canvas) to which paint is electronically applied.
4. Applying paint to the background image using a read-modify-write process, which, according to the shape of the brush, performs a proportional blend of paint data with canvas data.
5. Continuously repeating the read-modify-write process as the artist moves the brush, thereby creating a succession of overlapping stamps.
6. Applying a certain sub-pixel positioning technique to painting operations.

Given this information, one of my jobs was to determine whether any prior art existed that taught or suggested these six elements. Was there a publication that described a system that had these attributes? Had a system been implemented that exhibited this behavior, or both, all prior to December 4, 1980? Indeed, there were both publications and systems. I'll discuss some of those shortly, but first I'll describe some of the legal ground rules that I had to follow.

Infringement and Validity

As is usual in defending against patent infringement, I was asked to help with a two-pronged investigation. The first task was to determine whether Photoshop in fact infringed the Quantel patents. At the same time I conducted a search for prior art that would render Quantel's patents invalid. I'll spend just a little time on the infringement issue before continuing with more details of the validity story. It's not as though the infringement study was a minor part of my job, however. It required, after all, examining the Photoshop source code to see just how Photoshop performed the relevant operations. While I didn't have to become familiar with all Photoshop functions, being faced with navigating through about a half million lines of C++ code was nonetheless a bit daunting.

Determining the absence of literal infringement was quite straightforward. It involved looking at each element in an asserted claim and seeking just *one* element that was not present in Photoshop, i.e. Photoshop either didn't do what the element described or didn't do it in the way described. Of course, finding more than one such element was good but only one was needed to refute infringement of the entire claim. By progressively refuting infringement of each asserted claim in a patent, I was able to argue that Photoshop infringed none of the asserted patents. As we'll see later, the jury agreed.

Prior art can invalidate a patent claim in one of two different ways. Prior art can "anticipate" a claim, meaning that the art in question practices or teaches *exactly* the same subject matter as recited by the claim. In other words, all of the elements of an asserted claim, and the manner in which they are combined, are found in a single item of prior art. If the prior art does not quite anticipate the patent claim, the art may nonetheless render the claim invalid if the subject matter described by the claim is an "obvious" variant of the prior art. In that case, one tries to find as many elements as possible in one prior art item in order to conclude that it would have been obvious to a person with ordinary skill in the art at the time to modify the prior art to arrive at the claimed invention. I found historical SIGGRAPH material to be a fertile ground for potentially invalidating prior art. After providing a little background information, I'll discuss some of that material in detail.

Relevant Technology and Background

In 1975 Evans and Sutherland Computer Corp., in conjunction with the University of Utah, developed a device known as a video frame buffer. The original frame buffer comprised sufficient computer memory to represent a single video frame of 512x480 pixels having 8 bits of intensity resolution, and a display controller that read the memory contents at a rate consistent with display on a video monitor. When attached to a general-purpose digital computer, the frame buffer contents could be read and written by an application program to produce graphical images with a degree of complexity and realism not previously possible. By the end of 1975 one notable application of the frame buffer was as the "canvas" in a digital paint program. In addition to the frame buffer, all that was needed was a graphics tablet and stylus to mimic drawing or painting, a video display and software that could translate the user's gestures into marks on the canvas.

Because the basic paint program is so simple, many painting applications appeared rapidly and at many locations. Notable is the work of Alvy Ray Smith at New York Institute of Technology, Marc Levoy at Cornell University and Jim Blinn at the University of Utah and JPL. By 1978 all the cited researchers had developed highly sophisticated paint programs which incorporated features the painting patents claimed to be novel, like soft-edge brushes. As promised earlier, I'll soon describe what a soft-edge brush is and the role it plays in digital paint systems. But first, some background information.

A Brief Tutorial on Painting

Since the soft-edge brush and proportional blending concepts figure so prominently in the painting patents, I'll start with a brief discussion of the meaning of these terms. It's important to know that a brush, in the digital paint system sense, is actually a checker-

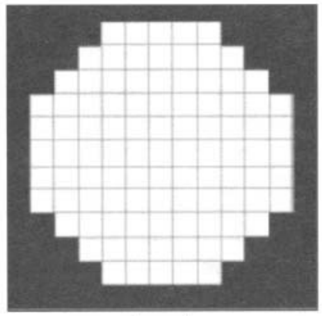

Figure 1

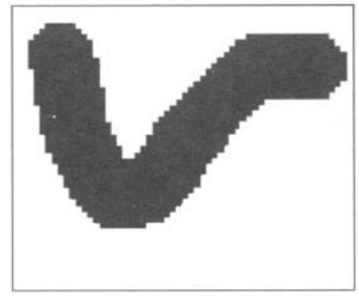

Figure 2

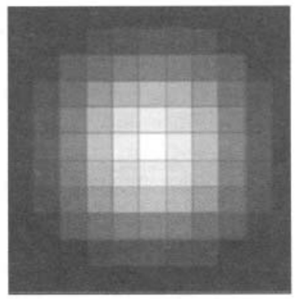

Figure 3

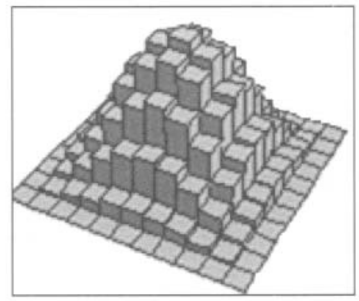

Figure 4

board-like collection of (8 bit) pixels. The value of each brush pixel defines its *weight*, which represents its relative contribution to the painting process. Depending on the desired shape of the brush, the collection of pixels might be square, circular or elliptical. Moreover, the larger the desired brush, the more pixels needed to define it.

Let's consider first the simplest kind of brush. Figure 1 shows a (greatly magnified) collection of pixels that defines a circular brush with an 11-pixel diameter. For this simple brush all pixels in the brush have a weight of 255, the largest number possible for an 8-bit pixel. In the digital painting process, the user steers this collection of pixels (the brush) over the canvas (the frame buffer) by moving a mouse or pen along successive positions on the tablet. At every point in a brush stroke each pixel in the brush, because it has the maximum possible value, replaces each pixel in the canvas that lies directly beneath it with a value that corresponds to the brush color. It is important to note that I have used white pixels in Figure 1 to denote that each pixel in the brush has the maximum value of 255; the brush color itself can be any color we choose.

Figure 2 shows a (greatly) magnified stroke painted with this simple brush on a white canvas and I have selected black for the brush color. It's clear that the edges of the stroke are quite rough and do not blend smoothly into the background. This condition is known by various terms — *staircasing*, the *jaggies*, and as one type of *aliasing* probler

Because aliasing produces an unpleasing visu effect, one usually takes steps to diminish i impact, sometimes by producing soft ed strokes.

But it's not the simplest brush that inte ests us; it's the so-called soft-edge brush. Th kind of brush doesn't have all pixels set to t same weight. Rather, it's a brush whe central pixel weights are high, pixels furth out have lesser weights and those at the ed of the brush have very low weights. Such brush is shown in Figure 3. It's called a *so edge brush* or sometimes a *shaped brush*. T pixels in Figure 3 are shown in differe shades of gray. This coloration denotes t weight of each pixel in the brush. The whit the pixel, the higher its weight.

It's also helpful to think of the brush as three-dimensional entity, where the weight each pixel is thought of as a height. Figure shows how the brush in Figure 3 wou appear in 3D. From this figure one can s that the brush is shaped to have a maximu contribution to a paint stroke at its cent with the contribution falling off to zero at edges. This shaped character of a brush what allows us to paint a stroke whose edg are smoothly blended with the backgrour thereby reducing the unwelcome aliasi effects discussed earlier.

Unlike the simple brush I discussed abo the role each pixel in a soft-edge brush pl in modifying the corresponding canvas pixel not one of just simple replacement with t brush color. With a soft-edge brush, what happens to a canvas pixel depends on the weight of the brush pixel located above it. Rather than just write the current brush color into the canvas pixel, the weight of the brush pixel is used as a blending factor to produce a weighted average of the current canvas color and the brush color. The process is to read the canvas pixel data from its location in the image, modify it and write it back to the same location in the image.

Here's what happens. At every brush position on the canvas, for each pixel in the brush, the painting process is:

1. Read the value of the canvas pixel under the current brush pixel.
2. Create a new color by blending the canvas color with the brush color in proportion to the weight of the current brush pixel.
3. Replace the canvas pixel under the current brush pixel with a canvas pixel having the newly created color.

We can represent this three-step process with a simple algebraic expression:[1]

$$nC=bW(bC+(1-bW)oC)$$

which is:

newCanvasColor = brushWeight (brushColor + (1 - brushWeight) (oldCanvasColor)

The formula on the right hand side of the equation is simply the weighted average of the brush color and the old canvas color. This formula is also called linear interpolation, or sometimes *lerping*, for short.

It's easy to see from the lerping expression how brush color and canvas color are blended. If brushWeight is 1, newCanvasColor is just the brush color because the oldCanvasColor is multiplied by 0. If brushWeight is 0, no change is made to the canvas color. For small intermediate values of brushWeight, newCanvasColor contains more oldCanvasColor than brushColor, while for larger values of brushWeight newCanvasColor contains more brushColor than oldCanvasColor.

Referring to Figure 3 or 4, it's clear that the brush shown there will contribute only a small fraction of its color to the underlying canvas at the edge of the brush. That behavior ensures that there will be no radical changes in color at the edge of the brush, thus reducing any aliasing artifacts. Hence the term *soft edge brush*. Figure 5 shows a magnified stroke painted with a shaped brush, again choosing a black brush color to paint on a white background. The reduction of sharp edges and the smooth blending with the background are apparent.

Notice in Figure 5 that individual applications (called stamps) of the 11-pixel-wide brush successively overlap each other during the stroke. This results in the center of the stroke being solid while the edges remain soft.

Invalidating Prior Art for the Painting Patents

Remember that the painting patents were entitled to a date of invention of December 4, 1980. Several paint systems, however, were in active use in the United States well before that date. Moreover, those systems had basically the functionality claimed by the Quantel painting patents, summarized by the six elements I identified above. I mentioned earlier that Alvy Ray Smith, Marc Levoy and Jim Blinn all had developed paint systems that were described and functional prior to December 1980. Some references to their work are, for example, [Smith78], [Levoy78] and [Blinn79]. In addition, and of critical importance, was the discovery of vintage software that implemented the three paint systems. While the work of all three researchers was important for my prior art studies, I'll concentrate on just the work of Smith for the rest of the article.

The Smith System: Relevant Publications

The [Smith78] and [Smith79] references pertain to the Paint and Paint3 systems developed by Alvy Ray Smith at New York Institute of Technology. Both systems — Paint3 was a 24-bit/pixel version of the 8 bit/pixel Paint — were fully operational by late 1977.

In [Smith78] a painting technique — called wet paint — is described. The [Smith78] reference, which describes Paint and Paint3, was widely distributed both informally and at SIGGRAPH conferences from 1978-1982. Thus, having a personal knowledge of SIGGRAPH attendance in those years, it is safe to say that thousands of people received copies of the [Smith78] reference from 1978 to 1980.

[Smith78] describes clearly aspects of the Paint and Paint3 systems. The reference provides sufficient information, including possible equipment configurations, to enable a person with knowledge of the technology of the era to reproduce the systems in their entirety. With specific regard to the brush patents, Smith describes a soft-edge brush painting technique, which he calls wet paint. His description of it is:

> *"wet paint – The (8-bit) values of the brush are used as 'wetness' weights. The higher the weight w in the brush the more dominant is the color A, the color selected to paint with, over the color B at a pixel under the brush. The computation is the familiar 'lerp' function (linear interpolation); w*A+(1-w)*B. If the brush is shaped as a random distribution over a circular radius, clustering toward the center with higher weights there, then wet painting simulates airbrushing."*

Specific brush wetness/weight distributions that were available are listed in [Smith78]. The cone family of brushes is one that was available; each brush in the family represented different sizes. I'll revisit these shaped brushes when I describe the actual Smith software.

Although his description of wet paint is brief, it is clear that he is describing precisely the type of brush required by the painting

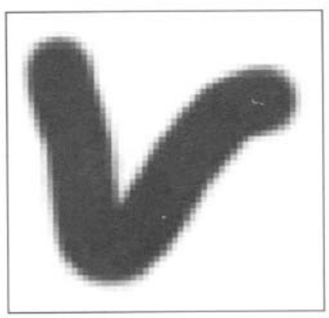

Figure 5

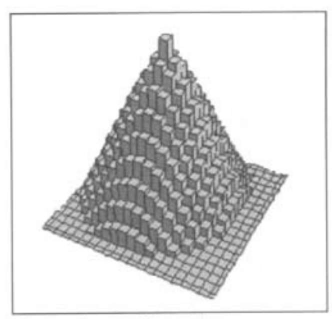

Figure 6

Figure 7

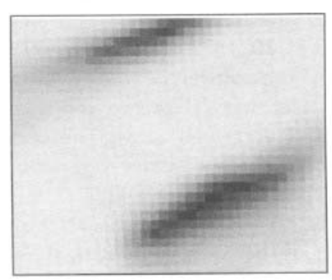

Figure 8

[1] In order to simplify this computation, brush pixel values of 0 to 255 have been normalized to decimal values between 0.0 and 1.0.

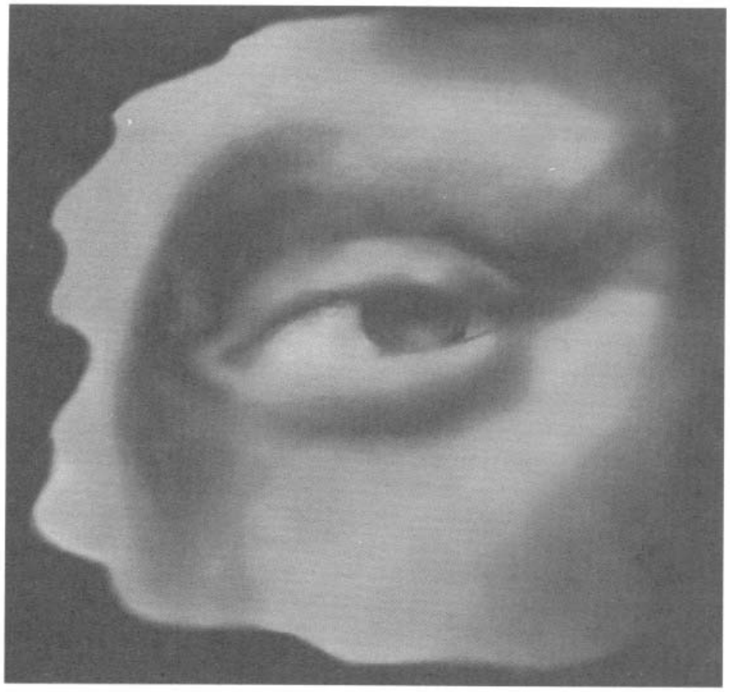

Figure 9: ***This image appears in color on page 94.***

patents. Moreover, any person knowledgeable in the technology of the era would have had no trouble implementing the wet paint technique in a replicated system. In fact, while a professor at the University of Michigan, I taught a class in 1984 [Phillips84] to a group of engineering students (programmers) and a group of (both student and professional) artists. The goal of the course was to implement a professional quality paint program on a Raster Technologies 24-bit/pixel frame buffer. The programmers — third and fourth year undergraduate computer engineering students — successfully implemented wet paint from its description in [Smith78].

Relevant Software

While the Smith publications make a persuasive argument for prior art, the discovery of the actual Paint3 software which Smith developed made the argument compelling. My job was to study this software and draw conclusions on its consistency with its description in [Smith78]. According to the dates of the relevant files, I found that all software was extant and working in the time frame between September 30, 1977 and May 4, 1979. One aspect of the software I studied particularly closely was that related to the wet paint capability I described above. The code that performed the wet paint operation was clearly present and with a little study it was possible to follow the brush-weighted blending of the paint color with the underlying canvas colors. Then I studied the brush data that pertained to the cone family mentioned earlier. I was able to decode the brush data and portray the brush's shape. The shape of the *cone9* brush, the largest member of the cone series, is shown in Figure 6. Its shape is clearly conical, with a maximum weight in the center and tapers off linearly to zero at the edges.

I can't go into all the details here, but a thorough study of the Smith code showed that it had basically the essential elements of the Quantel painting patents. Other aspects of the patents such as the use of a pressure-sensitive stylus and painting with a brush with sub-pixel resolution were found in other bodies of work (including the work of Smith and others at NYIT) and could be incorporated into my prior art arguments as being obvious.

Finally, a consultant actually compiled Smith's Paint3 software under Windows 95. He made minimal changes to the source code in order to make it compile on a modern C compiler and to produce its output via a modern VRAM, instead of the Evans & Sutherland frame buffers that were used in the mid to late 1970s. Execution of the program built from this source code produces brush strokes that have the exact appearance you'd expect to see from reading [Smith78]. Figures 7 and 8 are screen shots of typical strokes painted with the *cone9* brush using wet paint mode.

The Other Patents

Two of the other patents in this case, [Kellar86] and [Searby95], pertained to cutting and pasting of images, i.e. digital compositing. The subject of the fifth patent, [Searby85], is a palette for custom color mixing and selection for a digital paint system. For each of these patents I performed a infringement and validity study, just as described earlier. As with the painting patent historical SIGGRAPH material proved to b crucial for showing that there was prior a for the three remaining patents as wel Armed with all this information we went t trial.

The Trial

The Adobe/Quantel trial was held during tw weeks in September 1997 in the Feder District Court in Wilmington, DE. At the tri Quantel introduced a new interpretation the painting patents. They argued that the system was capable of producing "fine ar whereas prior systems — including Smith's b inference — could only produce "compute looking impressionistic art." The specific tec nical issue Quantel cited was that the system guaranteed that individual brus stamps would overlap, regardless of how fa the artist painted. It turned out that the judg disallowed that interpretation of the paten (on the grounds that the patents containe no such information), but a dramatic rebutt of the "fine art" shortfall of the Smith syste was provided by David Em, an artist who h worked with many digital paint system [Em88]. David Em was a witness for Adob and demonstrated the use of Smith's Pain system in court. Figure 9, although not a image shown at trial, speaks eloquently to t issue of "fine art." It was painted by David E using Paint3.

On Friday, September 19, 1997 the ju returned a verdict finding that all five Quant patents were invalid and not infringed.

Acknowledgments and Aftermath

I'm pleased to acknowledge the relentle pursuit of prior art, especially the releva software, by David Barkan and John Gartm of Fish & Richardson. Thanks to David Em f permission to include his painting, shown Figure 9 and to Douglas Brotz of Adob Systems for information on digital font defi tion litigation. Finally, thanks to Ja Newlevant for his meticulous job of reviving 20-year-old program to run on a mode platform.

On February 28, 1998 Alvy Ray Smit Dick Shoup and Tom Porter received a tec nical Academy Award from the Academy Motion Picture Arts and Sciences (AMPA for "pioneering inventions in Digital Pai Systems."

The April 6, 1998 issue of *The National L Journal* described the Adobe/Quantel case an article titled "Year's Top Wins for t Defense."

References

Patents

[Kellar86] U.S. Patent 4,602,286, Video Processing for Composite Images, July, 22, 1986.

[Searby85] U.S. Patent 4,525,421, Computerized Graphics System and Method Using an Electronically Synthesized Palette, June, 18, 1985.

[Searby95] U.S. Patent 5,459,529, Video Processing for Composite Images, October 17, 1995.

[Walker93] U.S. Patent 5,216,755, Video Image Creation System Which Proportionally Mixes Previously Created Image Pixel Data With Currently Created Data, June, 1, 1993.

[Walker94] U.S. Patent 5,289,566, Video Image Creation, February 22, 1994.

Publications

[Blinn79] Blinn, James F., "Raster Graphics', Tutorial: Computer Graphics, Compcon, Spring 1979, pp. 150-156. Reprinted in *Tutorial: Computer Graphics*, edited by Kellogg S. Booth, IEEE Computer Society Press, February 1979.

[Em88] Em, David, *The Art of David Em*, Harry N. Abrams, Inc., Publishers, New York, 1988.

[Levoy78] Levoy, Marc, "Computer-Assisted Cartoon Animation," MS Thesis, Cornell University, August 1978.

[Phillips84] Phillips, Richard L., "Kaleidoscope II," class project at the University of Michigan, 1984.

[Smith78] Smith, Alvy Ray, *Paint*, Technical Memo 7, Computer Graphics Lab, New York Institute of Technology, Old Westbury, NY, July 1978. Issued as tutorial notes at SIGGRAPHs 78-82. Reprinted in *Tutorial: Computer Graphics*, edited by John C. Beatty and Kellogg S. Booth, IEEE Computer Society Press, 2nd edition, 1982, pp. 501-515.

[Smith79] Smith, Alvy Ray, *Painting Tutorial Notes*, Issued as tutorial notes at SIGGRAPH 79, August 1979.

[Smith97] Smith, Alvy Ray, "Digital Paint Systems Historical Overview," Microsoft Corporation, May 30, 1997.

Richard L. Phillips
RL Phillips Consulting Inc.
1291 Lejano Lane
Santa Fe, NM 87501
Tel: +1-505-982-3647
Fax: +1-505-989-5085
Email: **pappa@ix.netcom.com**

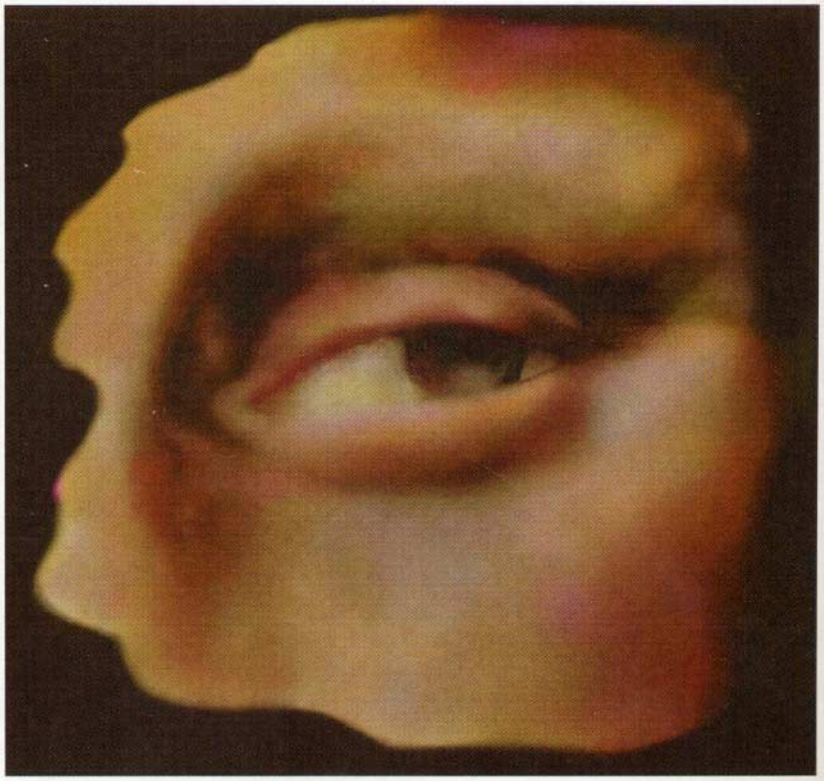

Figure 9: "Computer Graphics in Court: The Adobe/Quantel Case" by Richard L. Phillips. See pages 51-55.