

# Color in Web-based 3D Graphics

*Maureen C. Stone*

*Xerox Palo Alto Research Center  
Palo Alto, CA*

## Abstract

The World Wide Web has become a widely used multi-media platform. The Virtual Reality Modeling Language (VRML) was created to be a standard representation for interactive 3D graphics on the web. Like all web-based publishing, making 3D worlds that are visually similar and effective across browsers and across computing platforms is difficult.

The VRML Color Fidelity Working group was formed in early 1997 to try and address some of the color fidelity problems in the VRML specification and implementation. This paper will describe the goals and process the group is using to make VRML worlds "look the same" across all browsers and all platforms. While this work is targeted towards VRML, many of the issues are common to desktop graphics and web-based color in general.

## Introduction

The World Wide Web<sup>1</sup> is widely used to distribute multi-media documents and presentations. Web browsers are available for all major computing platforms, and references to websites by publishing URLs (Universal Resource Locators) have become common in all aspects of science and industry.

Content on web pages is defined by the Hypertext Markup Language<sup>2</sup> (HTML), which in its most standard form supports text, images and hyperlinks. This is extended by the browser and plug-in \* manufacturers to include features such as sound, video, 3D graphics and embedded programs. The result is that content on web pages can include almost anything that can be generated by a computer program. This richness has created a challenging new frontier for distributed media, especially since the target audience is viewing content on general purpose, unstandardized, desktop computers.

Color fidelity problems on the web are rampant. Many desktop systems have a limited color palette (8 bits/pixel), which is further limited by the peculiarities of browser implementations. Even for systems that support a larger color space (16 or 24 bits/pixel), differences in display systems and viewing conditions can create dramatic differences in color and image appearance. However, the web has enormous momentum and seems to be creating an important new form of information distribution. Defining standards

for color fidelity, even though full conformance may be impossible, seems an important and worthwhile enterprise. The Virtual Reality Modeling Language<sup>3</sup> (VRML) was created to be a standard representation for interactive 3D graphics on the web. Its current version, VRML 97, has been submitted to the International Standards Organization (ISO/IEC DIS 14772-1), and is supported by the two major browser manufacturers, Netscape and Microsoft. Figure 1 shows an image taken from an animated sequence showing the Mars rover rolling past a giant can of Pepsi. This sequence encoded as VRML is much smaller than an equivalent sequence of animated images.

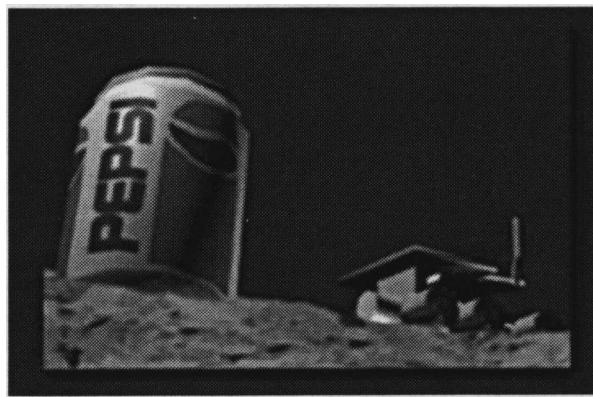


Figure 1. Image from the Pepsi VRML banner ad. Courtesy of Pepsi Corporation.

The VRML Color Fidelity Working Group was formed early in 1997 to address part of the problem of making VRML worlds "look the same" across all browsers on all computing platforms. This paper will describe the group's process and goals as of late summer 1997. While VRML's long-term dominance of 3D web-based graphics is not guaranteed, the issues raised with respect to color fidelity should apply to other web-based 3D graphics representations.

## VRML Color Fidelity Goals

The VRML 97 specification defines colors simply in terms of RGB values. This is obviously not a precise color specification that could be used for interchange with other standards. A primary goal of the Color Fidelity Working Group is to create a precise definition of RGB for VRML. Qualitatively, the goal is to enable authors to create content that can be viewed effectively on a wide variety of desktop systems. Without some formal specification, content authored for one workstation will look radically different on another.

\* A program that runs embedded in a browser window, making it possible to display non-standard content on a web page. Plug-ins must be installed in a browser to function, and may not operate on all platforms.

VRML, and web-based content in general, is very much like television in that it is easy for viewers to make radical changes to the display system by adjusting the knobs on their monitors and placing their workstations in highly variable lighting situations. Therefore, even if there is a precise definition for color, the actual display will vary. We want a specification that both creates a precise match between controlled environments and degrades gracefully over a wide range of typical situations.

Unlike television, the equipment needed to author VRML is very similar to that needed to view it. Anyone with a workstation, a browser, and some talent and enthusiasm for 3D graphics can create VRML content. Therefore, the specification cannot be built into the creation tool, as in a television camera, but must integrate well with other tools in the desktop computing environment.

The output of the VRML Color Fidelity working group will be an annex to the VRML 97 specification that defines its RGB color space precisely. We intend to provide two sets of tutorial documents, one for browser writers and one for authors. These will most likely be implemented as websites. The specification will also be encoded as a standard color profile that can be used by color management systems.

## Approach

Our basic approach is to follow the broadcast television model of defining a *reference display* (studio monitor) and a *reference set of viewing conditions* (studio setup). We then work backwards to define the precise specification for RGB in the VRML file format. This is similar to the model proposed for the sRGB<sup>4</sup> color space and the PNG<sup>5</sup> format for images on the web, and has the advantage that it is easy to explain and implement.

If we define a reference display that matches a typical desktop display system, default implementations can display the RGB pixels created by rendering a VRML scene without modification. This is an important issue for performance. Different grades of quality can be achieved by different degrees of conformance to the standard references. Professional content creators can achieve high-quality results by working in carefully controlled “studio environments” on calibrated displays. Less critical authors can get less accurate but still acceptable results simply by authoring on their own work-stations.

### The Reference Display

We base our reference on a computer monitor because that is by far the most common display system used for viewing VRML. The next most critical is the flat panel displays used in notebook computers. Practically speaking, film and print are of less immediate interest, though providing a precise definition for VRML color will facilitate conversions to these media.

A monitor is precisely defined by the chromaticity of its phosphors, its white point, and the function that maps voltage (expressed as normalized pixel values) to intensity. This function is sometimes referred to as its *gamma curve*

because it is typically a power function with an exponent of “gamma.” For VRML, it seems obvious to follow sRGB and other image standards by using the ITU-R BT 709 phosphors and a white point of D65, shown in Table I.

**Table I. CIE chromaticities for ITU-R BT.709 reference primaries and CIE standard illuminant D65**

	Red	Green	Blue	D6 5
x	0.6400	0.3000	0.1500	0.3127
y	0.3300	0.6000	0.0600	0.3290
z	0.0300	0.1000	0.7900	0.3583

These match modern computer monitors well, and while many computer monitors are sold with a default white point of D93, it is straightforward on modern systems for the user to change it.

Our primary challenge is defining the gamma curve because the tradition in the computer graphics community is to assume an effective gamma of 1.0 (linear intensity). Other standards, however, assume a higher value, such as 2.2, which is closer to the default behavior of a CRT-based display system. This issue will be discussed in detail in a later section of this paper.

### Reference Viewing Environment

With respect to the reference viewing environment, we once again look to the sRGB standard and choose an ambient luminance level similar to a dimly lit office, 64 lux. The proposed sRGB illuminant white point is CIE D50, presumably to facilitate viewing prints. This seems unimportant for VRML and does not match most office lighting. But, it is not clear whether advantages of recommending a different white point, such as D65, outweigh the disadvantages of having conflicting standards. The primary issue is the amount of light falling on the monitor, which strongly affects the perception of the displayed graphics.

## Rendering 3D Graphics

VRML supports general, real-time interactive graphics, which is difficult to achieve without hardware acceleration.<sup>†</sup> Therefore, whatever we define for color fidelity must have minimal impact on the rendering performance.

A VRML scene contains objects defined as collections of polygons. These objects can be colored either with solid colors or *textures* (images or video mapped onto a polygon surface). The scene can contain *light sources* that shine on the objects and affect their colors. There is also facility for sound and interaction.

Figure 2 describes the process of converting a VRML file to a visible image. The scene is read in from a file and converted to a data structure called a *scene graph*. To view it, the scene graph is *rendered* by projecting it through a virtual camera to a rectangle that represents the region on the digital desktop where it will be displayed. Then, the transformed geometric description is converted to pixels based on the visible shapes, colors and lights in the scene. The rendered pixels are passed to the display system for viewing.

<sup>†</sup> Games are special-purpose graphics that have been preprocessed to improve performance.

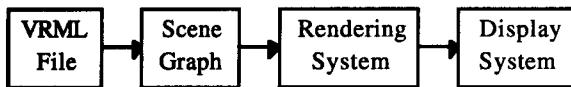


Figure 2. The process of rendering a VRML file. The file is parsed and converted to the scene graph data structure. This is rendered and displayed, potentially at interactive speeds.

In desktop graphics display architectures, the rendering process is split between *geometric processing* and *rasterization* steps as shown in Figure 3. In most VRML implementations, the geometric processing includes all of the viewing and lighting calculations. The result is represented as a collection of overlapping shaded triangles. These are rasterized by low-level primitives supplied by the operating system or by specialized graphics hardware. The triangles are specified in display system coordinates with RGB color values defined for each vertex. These colors are interpolated to shade the triangle. If the triangles are small, this approximation to the actual model will be very good. If they are too large, the result will have visible rendering artifacts.

The rasterization step results in RGB pixels that, for monitor viewing, are stored in a *frame buffer*. Colors in a typical workstation frame buffer are stored at a maximum of 8 bits/color channel. In most, but not all, computer graphics hardware, there is a *lookup table (LUT)* in the display hardware that can remap each color channel before it is sent to the monitor hardware. The frame buffer is typically part of the hardware display controller that drives the monitor electronics.

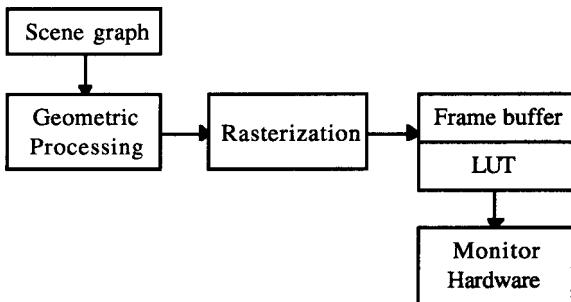


Figure 3. The process of rendering a scene graph for display on a monitor in a typical workstation. The hardware LUT is missing in low-end display controllers.

For many VRML implementations, rasterization is provided by low-level libraries or specialized hardware. Control of the LUT is typically left to the operating system. This creates a substantial systems challenge for VRML color fidelity.

## All About Gamma

There are many different uses of the term “gamma” in graphics and imaging systems. Therefore, discussions about the correct value for gamma are often very heated and confusing. This section will define our terminology for gamma, and describe our progress on defining “the right gamma” for VRML.

### Terminology

Computer monitors convert voltage to intensity via the following equation:  $I = A(k_1 D + k_2)^{\gamma}$  where  $k_1$  and  $k_2$  are

the system gain and offset,  $D$  is the normalized pixel value,  $A$  is the maximum luminance of the CRT and  $I$  is the resulting luminance. This is usually simplified by assuming the black level is 0.0 ( $k_1 = 0.0$ ) and the absolute values for illuminance do not matter, giving:  $I = V^{\gamma}$  We call the value for gamma in this equation the *monitor gamma*. Typical values for a workstation monitor gamma are 2.2–2.6.

Some workstation operating systems provide a form of “gamma correction” by inserting an inverse gamma function in the LUT:  $RGB_{out} = RGB_{in}^{(1/\gamma)}$  We call the gamma term in this equation the *OS gamma*, and typical values for different workstations are summarized in Table II.

**Table II: Default values for OS gamma**

P C	MAC	SGI
1.0	1.4	1.7

The important value for viewing is the product of the OS gamma with the monitor gamma, which we call the *display system gamma*. This reflects the transfer function from RGB pixels in the frame buffer to intensity values viewed on the monitor. Assuming a monitor gamma of 2.2 and the default OS gamma values in Table II, the default display system gamma for each platform is shown in Table III.

**Table III: Default display system gamma**

P C	MAC	SGI
2.2	1.57	1.29

### Evaluating Monitor Gamma

It seems generally agreed that modern computer monitors have similar values for monitor gamma except for the effect of the brightness and contrast settings. At this writing, however, I am still trying to get the experts to agree on what this value is. Poynton<sup>6</sup> claims 2.5, assuming careful setting of the black level to 0.0 and a dark room. The sRGB proposal references Sony data claiming 2.2. This data was generated in lighting more typical of office environments, which may have affected the measurements. Also, in brighter environments the preferred brightness setting may not be 0.0, which will change the measured value of gamma if the simplified equation is used.

It is hard to measure the gamma curve directly because few instruments can measure the darkest part of the monitor response curve—I personally have never gotten any instrument to read reliably below pixel value 60 out of 255. Therefore, the error in the most sensitive part of the curve is the highest. Fortunately, for this application the actual value for gamma does not really matter as long as it can be reliably achieved by some measurement process. For this discussion, I will follow the sRGB spec and Sony data and assume a default monitor gamma of 2.2.

### Tradition of Linearity in Computer Graphics

Computer graphics lighting and shading models are designed to mimic the physical interaction of light with surfaces. Therefore, there is a tradition of treating pixel values as linearized intensity values and displaying them on a monitor “gamma corrected” to be linear in intensity (display system gamma of 1.0). When the lighting calculations mimic reality, and the scene rendered is similar in absolute

intensity to the monitor, this can give excellent results.<sup>7</sup> More recent work in this domain attempts to model the response of the eye to absolute luminance to more accurately rendered models that are very dimly or brightly lit, or have very high dynamic range.<sup>8,9</sup>

Shading models for desktop systems such as VRML, however, are based on simplified lighting and shading models developed by Gouraud and Phong in the 70's.<sup>10</sup> These are composed mostly of interpolated ramps of color that can be combined to give a 3D shaded appearance. There is no modeling of light scattering from surfaces, or even of shadows. Dramatically beautifully pictures and special effects have been created with such systems. However, a great deal of artistry is needed to produce interesting and natural-looking scenes.

Linearity is also an issue for anti-aliasing,<sup>10</sup> which blends edges of objects with the background to remove quantization artifacts or "jaggies." Most algorithms assume computation on linear-intensity pixel values, as do transparency and other blending functions.

### Display System Gamma for VRML

From a traditional computer graphics perspective, the correct display system gamma for VRML should be 1.0. Looking at Table III, however, we see that no workstations, not even SGI graphics systems, are setup with a default display system gamma of 1.0. Yet, people create graphics on all of these systems. Is it possible that linearity is not really important for the simplified shading models used in VRML and similar systems?

There are significant system advantages to choosing a non-linear display system gamma for VRML. Few standard image formats encode pixels as linear intensity values, as this is an inefficient way to encode them. Therefore, it is more common to assume a display system gamma more like the natural monitor response, so that the pixels represent intensity values raised to a power of approximately 1/gamma. VRML needs to consider these formats because textures are represented as images or even as video. Consider also the case where a VRML world exists on a web page with an image that also appears in the VRML world as a texture. It would be nice if there was not a dramatic difference between the two renderings of the image.

No matter what the display system gamma, it is theoretically possible to precompensate the rendered VRML pixels before placing them in the frame buffer. However, the typical display architecture described in Figure 3 does not make this practical today. It would be much simpler if the rendered pixels could be displayed without modification.

The VRML rendering model is based on a shading model created by Paul Strauss<sup>11</sup> at Brown University that eventually became part of the Silicon Graphics *Open Inventor™* format.<sup>12</sup> It is based on the Phong shading model, but provides a simpler way to adjust the basic lighting and surface parameters than using the original equations. The goal was to make it easier for artists and designers to get the visual effect they want. This suggests that the VRML model is not really physically based, but artistically based. VRML worlds are designed; is it possible to design them so that they look good on a non-linear display?

To help answer this question, we performed a set of informal experiments comparing simple VRML scenes ren-

dered on two systems with different display system gamma curves. One was setup to mimic a typical SGI workstation (display system gamma of 1.29) and the other was setup with a display system gamma of 2.2, as in the sRGB standard. The two monitors were set side-by-side in a lab under controlled viewing conditions. The colors of the digital desktop and the scene backgrounds were hand-corrected to match so that there were no clues as to which system was which. A small group of engineers and 3D graphics designers were asked to compare simple VRML worlds that were simultaneously displayed on the two screens. Almost universally, the group felt the results were more attractive on the monitor with the 2.2 display system gamma. This was a big surprise, especially since the viewers were primarily SGI system users.

These results suggest that a default display system gamma of 2.2 would be fine for VRML 97. Before finalizing this, we need to do more work to discover if there are unsolvable problems with anti-aliasing or other blending functions. We also need to do further testing of more complex worlds, and to think about the transition strategy to more complex shading models where linear intensity is important.

### Visual Evaluation of Gamma

For web-based graphics, it is important that users be able to adjust their monitors to a reliable state without instruments, which suggests the use of a visual test pattern for evaluating gamma. Such a pattern compares gray values to patterns of black and white pixels. Figure 4 shows an example of such a pattern. The value for gamma is read from the bar where the top and bottom appear to match. Informal tests with such patterns give results similar to those shown in Table III.

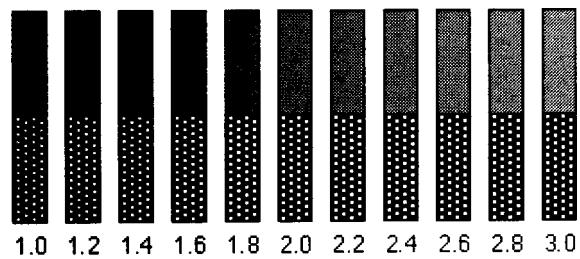


Figure 4. Gamma test pattern, courtesy of Robert Berger,<sup>13</sup> Carnegie Mellon University.

Creating a reliable, standard visual test pattern for evaluating display system gamma is crucial for all web-based color standards. We hope to work with the sRGB group to develop a common set of test patterns. By their nature, these patterns measure not only the hardware, but also the effect of the viewing environment.

### Calibrating the File Format

Given a precise definition of a monitor's phosphors, white point and gamma curve, it is possible to convert between any RGB triple and its corresponding tristimulus value using a table lookup and a matrix multiply. For VRML, the main advantage of calibration would be to enable encoding measured colors in the VRML file format.

Colors in a VRML file are defined as three floating point values, one for each primary. By definition, only those in the range of [0..1] are defined, though values outside of that range can be allowed. Therefore, to encode a measured color, we simply work backwards from the reference monitor specification. If we do choose a non-linear value for the gamma curve, then the calculated RGB values must be pre-compensated accordingly.

Calibrating the VRML file format allows the use of measured color for commerce and scientific exchange. Colors in a VRML file are used to describe the color of a light source, components of the material (diffuse and specular), vertex colors and textures. Commercial applications may want to match these colors to physical objects. Scientific applications may want to precisely encode information in the value of the color.

Objects can either be simply colored, or colored and lit. If an object is lit, its color is defined algorithmically from the interaction between its material properties and any lights shining on it. If it is unlit, the surface colors define its appearance.

For VRML 97, measured colors are most likely to be specified for unlit colors and textures. Shining lights on the colors adds enough variability (both due to the shading model itself and to the approximations to the shading model needed for performance) that color fidelity would be difficult to evaluate in the lit cases. However, as VRML and similar standards evolve towards more realistic lighting and shading models, the ability to measure, model and light real objects will become more important.

## Status and Future Work

As of late summer 1997, the primary issue is determining whether adopting a display system gamma of 2.2 will work for VRML 97. Besides the issues discussed above, we need to consider the implications for non-CRT displays such as flat panels and head-mounted displays.

Once the proposal is finished, it will be presented to the VRML Review Board and VRML Consortium 14 for approval. See the VRML website for our current status and results:

<http://www.vrml.org/WorkingGroups/>

## Conclusions

VRML and other web-based graphics standards are creating an exciting new genre for publication of digital color imagery. Color fidelity on the web is currently a disaster. There is hope, however, that efforts like the VRML Color Fidelity Working Group, the sRGB color space and the PNG image format can help bring some degree of reliability to color on the web.

Unlike most image reproduction technologies and physically-based computer graphics, VRML does not attempt to reproduce reality. VRML worlds are designed to be displayed on computer monitors. Give an experienced designer a stable, flexible platform and wonderful things will be created. This distinguishes computer graphics, both 2D and 3D, from other digital imaging technologies; the

focus is on creation, not reproduction.

VRML worlds are more than just colored pictures. They move, make sounds and provide means for user interaction. Color fidelity is only a small part of making VRML worlds look and act the same on all platforms, but it is a necessary piece of the foundation. This work, we hope, will contribute to the success of VRML and other similar formats on the web.

## Acknowledgments

I want to thank Mark Callow and Todd Newman, both of Silicon Graphics Inc., for their contributions to the VRML color fidelity effort. Mark and I co-chair the working group, Todd is an expert color consultant and vice-chair of the International Color Consortium<sup>15</sup> (ICC). Michael Stokes of Hewlett-Packard Laboratories hosted the gamma evaluation experiments in his color laboratory. Michael and Todd designed and ran the experiments. Finally, I want to thank the management of Xerox PARC and the Xerox Corporation for supporting my involvement in this effort.

## References

1. *The World Wide Web Consortium*. <http://www.w3.org/>
2. *HyperText Markup Language (HTML)*; <http://www.w3.org/MarkUp/>
3. *The Virtual Reality Modeling Language*, (April 1997); <http://www.vrml.org/Specifications/VRML97/DIS/>
4. M. Stokes, M. Anderson, S. Chandrasekar, R. Motta. *A Standard Default Color Space for the Internet-sRGB*, (Nov. 1996). <http://www.color.org/contrib/sRGB.html>; (see page 198, this publication).
5. *Portable Image Format (PNG)*, (Oct. 1996); <http://www.w3.org/TR/REC-png-multi.html>
6. C. Poynton, *A Technical Introduction to Digital Video*. John Wiley & Sons, New York, NY, (1996).
7. G. Meyer, H. Rushmeier, M. Cohen, D. Greenberg and K. Torrance, An Experimental Evaluation of Computer Graphics Imagery. *ACM Transactions on Graphics*, **5**, 1, (Jan. 1995) 30-50.
8. J. Tumblin and H. Rushmeier, Tone Reproduction for Realistic Images. *IEEE Computer Graphics & Applications*, **13**, 6, (Nov. 1993) 42-48.
9. J. Ferwerda, S. Pattanaik, P. Shirley, D. Greenberg, A Model of Visual Adaptation for Realistic Image Synthesis. *Proceedings of ACM SIGGRAPH '96*, (Aug. 1996) 249-258.
10. J. Foley, A. van Dam, S. Feiner, J. Hughes, *Computer Graphics Principles and Practice, Second Edition*. Addison Wesley, Reading MA, (1990).
11. P. Strauss, A Realistic Lighting Model for Computer Graphics Animators. *IEEE Computer Graphics & Applications*, **10**, 6, (Nov. 1990) 56-64.
12. *Open Inventor*. Silicon Graphics, Inc., (1994); <http://www.sgi.com/Technology/Inventor.html>
13. R. Berger, Why Do Images Appear Darker on Some Displays? An Explanation of Monitor Gamma; <http://www.vtiscan.com/~rwb/gamma.html>
14. *The VRML Consortium*. <http://www.vrml.org/>
15. *The International Color Consortium*. <http://www.color.org/>

★ This paper was previously published in *IS&T's 5th Color Imaging Conference Proc.*, p. 27 (1997).