Nonlinear Color Transformations in Real Time Using a Video Supercomputer

Edward F. Kelley, Bruce F. Field and Charles Fenimore
National Institute of Standards and Technology, Gaithersburg, Maryland*

Abstract

Investigations of the effects of color transformations used for video displays are often hampered by the inability to see the effects of these changes in real time for a variety of input signals. Using a video supercomputer, the Princeton Engine, the effects of a parametric nonlinear color transformation can be shown in real time.

Introduction

With the advent of high-definition television (HDTV) and a variety of new technologies for display devices, the concern for accurate color rendering is growing; indeed, accuracy in color rendering may become an important measure of display quality.[1] Should manufacturers be interested in how a finished display will appear with the use of certain color filters, the employment of new phosphors, or with the use of new linear or nonlinear luminance or color transformations, they have little recourse but to build a prototype and view the resulting display. This paper reports the use of a video supercomputer to perform any general luminance or chrominance transformation on input video and display the effects of those changes in real-time output video. The transformation functions and any matrixing are parameterized, and these parameters can be changed "on-the-fly" while viewing the results. To illustrate a parameterized nonlinear transformation, we simulate an electroluminescent (EL) display using a cathode-ray tube (CRT) as much as the CRT color gamut will permit. Our purpose in doing this is to demonstrate the versatility of the video supercomputer as a research tool for simulation. Another use for such nonlinear transfor-mations would be to investigate the conversion of wider-gamut color source signals as they are adapted to a smaller-gamut display.[1]

The advantage the supercomputer offers is that it permits viewing of the resulting modified video in real time. This real-time video processing is useful in that it allows the operator to spend much more time investigat- ing new ideas on how to make changes in the video, rather than generating the methods to make the changes or waiting for processing to see the changes obtained. Up until the advent of parallel-processing supercomputers, researchers have been limited to static images, frame-by-frame analysis of a video sequence that is then recorded frame-by-frame to be played back in real time, or they have designed matrixing hardware to accomplish the color adjustments.[2-5] As high-speed computers become more available, we will be seeing more research and simulation being performed in real time because of the advantages offered by being able to make complicated adjustments while viewing a wide variety of full motion video.

Video Supercomputer

Our research uses the Princeton Engine, a massively-parallel video supercomputer designed for the processing of video-rate signals in real time. The machine was developed at the David Sarnoff Research Center, Princeton, NJ, expressly for the purpose of simulating complex analog and digital video processing systems.[6]

The architecture of the Princeton Engine is optimized for raster scanned data; a linear array of processors (1024 in the NIST machine) processes an entire video scan line simultaneously. Analog-to-digital converters and clock synchronizing circuits provide sampling of six analog video inputs at up to 80 megasamples/second. The array of 16-bit processors operate on the acquired video scan line executing instructions from a common instruction store (a Single Instruction Multiple Data, SIMD architecture). Real-time processing of NTSC (National Television System Committee) video can occur if the total instruction limit for each video scan line does not exceed 63.5 μs. After processing, the pixel data are output through digital-to-analog converters to component (RGB: red, green, blue) CRT monitors for display. Six outputs are available and may be configured to drive six monochrome or NTSC monitors, or two component CRT monitors. A unique feature of the architecture is the ability to arbitrarily map several outputs from any processor to different pixel positions on the screen. This provides the ability, for example, to place "processed" and original video images side-by-side on the same screen for comparative viewing. Figure 1 provides an example of the flexibility of the output: The source video (a) can be displayed, the modified video (b), sections of both source and modified displayed side by side (c), or both subsampled to display entire screens side by side (d). Up to four video images can be displayed on one screen if desired.

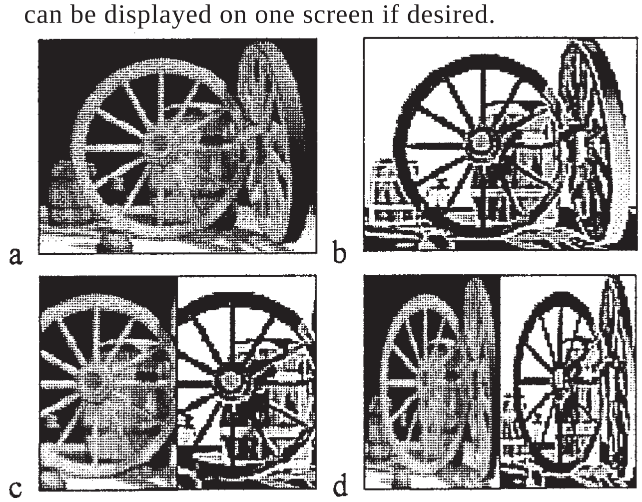

Figure 1. Typical Princeton Engine display options

Programming is done using a specialized C language compiler that has been extended for this parallel architecture. While editing and compilation are done on workstations networked to the Princeton Engine, user parameters (variables) may be defined within the program that are adjustable in real time while the program is executing on the Princeton Engine.

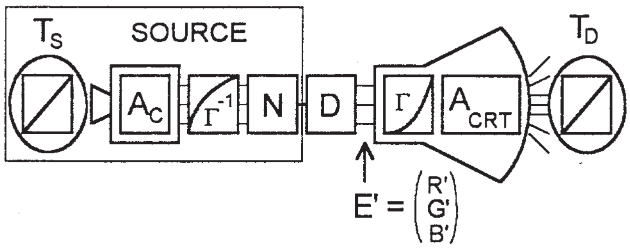

Figure 2. NTSC television system schematic

Nonlinear Model

General Model

Figure 2 shows a diagram of the present normal NTSC television system. The entire system can be represented mathematically as a series of operators and linear matrices. The operators are diagrammed as graphs of functions in boxes to indicate the nature of the nonlinearity involved, circled boxes show the linearity or nonlinearity of the grayscale obtained starting from a linear gray scale at the camera source, and the matrices are represented as letters in boxes. We define E = (R, G, B) to be a column vector of the component signals and T = (X, Y, Z) to be a column vector of the tristimulus values. The operators are functions that do not mix the components but operate on each component separately. The vectors will be in italics, and the operators will be in bold face to distinguish them from the matrices. With this notation, the normal NTSC television system can be written as

$$T_D = \mathrm{A}_{\mathrm{CRT}}\, \Gamma\, \mathrm{D}\, \mathrm{N}\, \Gamma^{-1}\, \mathrm{A}_C T_S. \quad (1)$$

Here the original scene color T_S is transformed into electrical component signals according to the camera primaries represented by the matrix $\mathrm{A_C}$. The signals are then "gamma corrected" by Γ^{-1} to compensate for the anticipated CRT display at the receiving end. These signals are encoded by the matrix N, transmitted, and decoded by the matrix D. The signals then pass into the CRT with its inherent space-charge induced nonlinearity represented by Γ, the resulting electron beam then encounters the tube phosphors that transforms them to a color T_D according to the phosphor matrix $\mathrm{A_{CRT}}$. If the system were perfect, T_D would be the same as T_S. This is only approximately true in the NTSC system and we will not be concerned with its deficiencies or complexities. For our purposes, we will consider only the signal coming out of the decoder

$$E' = D N \Gamma^{-1} A_C T_S, \quad (2)$$

where the prime denotes the so-called "gamma" correction and indicates the application of an opto-electronic transfer function at the source. The Γ operator and its inverse Γ^{-1} are specified by the SMPTE 170M standard,[7]

$$\Gamma^{-1} E = \begin{cases} (1+a)\, E_i^{\,1/\gamma} - a, & \text{for } c \le E_i \le 1 \\ (10/\gamma)\, E_i, & \text{for } E_i \le c, \end{cases} \quad (3)$$

where a = 0.99, γ = 2.2222..., c = 0.018, and where the E_i represent the component RGB signals according to the opto-electronic transfer characteristics of the reference camera. In the Princeton Engine program this function is generated by a nine-parameter piecewise linear approximation, see Fig. 3. Although each component can have a different function in this program implementation, for the purposes of this paper each RGB component is modified by the same function specified in Eq. (3).

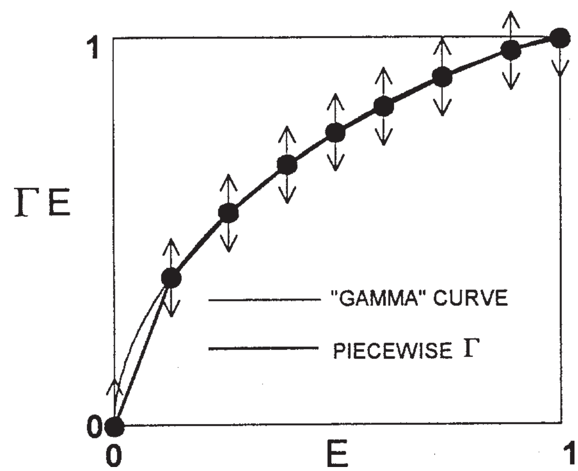

Figure 3. Piecewise linear function approximation

New Phosphor Model

The above formulation can be applied to considering the effects of a change in phophors. Assume that the CRT employed has SMPTE phosphors[7] and that we wish to simulate the appearance of an EL display using different phosphors and having a linear signal-luminance response,[8] see Table 1. In doing so, we start with a casual approach and directly apply the composite RGB signals we get from a video decoder to the new phosphors without performing any matrixing to adjust the colors and compensate for the new phosphors. We may be interested in seeing how satisfactorily we can correct the resulting picture by using any existing contrast, brightness, saturation, and tint controls.

TABLE 1.
Phosphor chromaticity coordinates (CIE 1931[9])

	SMPTE-C Phosphors[7]		EL Phosphors[8]	
	x	y	x	y
R	0.630	0.340	0.65	0.34
G	0.310	0.595	0.31	0.60
B	0.155	0.070	0.15	0.19

Figure 4a shows the connection of a linear EL display to the component signal that would normally be sent to a CRT. The EL panel displays a tristimulus color T_{EL} from the E' signals according to the phosphor matrix A_{EL}:

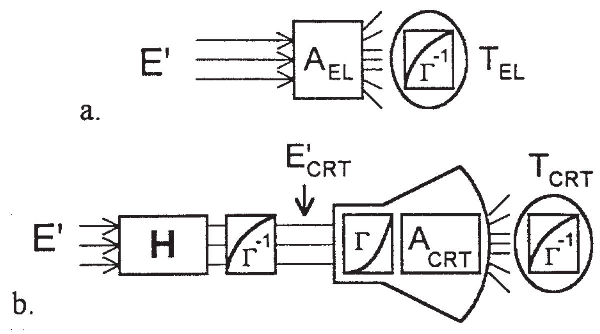

Figure 4. Simulation of EL display with CRT

$$T_{EL} = A_{EL}\, E'. \qquad (4)$$

The coefficients of the A_{EL} matrix are determined by the chromaticity coordinates of the RGB phosphors used in the EL display and the selected white point.[10,11] In this simulation, the D_{65} white point is chosen where x = 0.3127, y = 0.3290. Simulating the EL display with a CRT amounts to requiring (within the CRT gamut) that the color displayed by the CRT T_{CRT} matches the color observed through use of the EL display, that is,

$$T_{CRT} = T_{EL}\, . \qquad (5)$$

Referring to Fig. 4b, we need to find the component signals E'_{CRT} supplied to the CRT that will produce this color. The CRT color is given in terms of the CRT phosphor matrix and the intrinsic gamma response of the electron guns

$$T_{CRT} = A_{CRT}\, \Gamma\, E'_{CRT.} \qquad (6)$$

Here, the phosphor matrix coefficients are given in terms of the CRT phosphor chromaticity coordinates and the D65 white point.[10,11] The requirement of color matching, Eq. (5), permits us to relate the requisite CRT signal E'_{CRT} to the signal given the EL display E' via Eq. (4) to obtain

$$E'_{CRT} = \Gamma^{-1}\, A_{CRT}^{-1}\, A_{EL}\, E'. \qquad (7)$$

The product of the inverse CRT matrix and the EL matrix is the matrix H in Fig. (4). We find

$$H = A_{CRT}^{-1}\, A_{EL} = \begin{pmatrix} 1.256 & -0.005 & -0.251 \\ -0.016 & 0.650 & 0.366 \\ -0.015 & -0.005 & 1.020 \end{pmatrix}. \qquad (8)$$

The elements of H then become complicated functions of the chromaticity coordinates of the EL phosphors, the CRT phosphors, and the D_{65} white point. Thus, the simulation amounts to converting the E' signal to the E'_{CRT} signal:

$$\boldsymbol{E'_{CRT}} = \Gamma^{-1}\, \mathrm{H}\, \boldsymbol{E'}, \qquad (9)$$

where the inverse gamma operator and H matrix are implemented in the Princeton Engine.

There is a remaining difficulty that needs to be addressed, viz., how are out-of-gamut colors handled? Figure 5 shows the two gamuts involved. The CRT cannot produce colors outside its RGB gamut (solid lines). Consider an out-of-gamut color opposite the red CRT phosphor, point p, yet within the EL gamut. The application of the H matrix yields a CRT red value that is negative for the out-of-gamut point p. The line from the red CRT phosphor through point p represents constant green and blue amounts as red varies. In this simulation, the CRT out-of-gamut negative values are handled by simply limiting the value to zero. The color then rendered for point p is p' that occurs at the intersection of the red-p line with the green-blue gamut limit line.

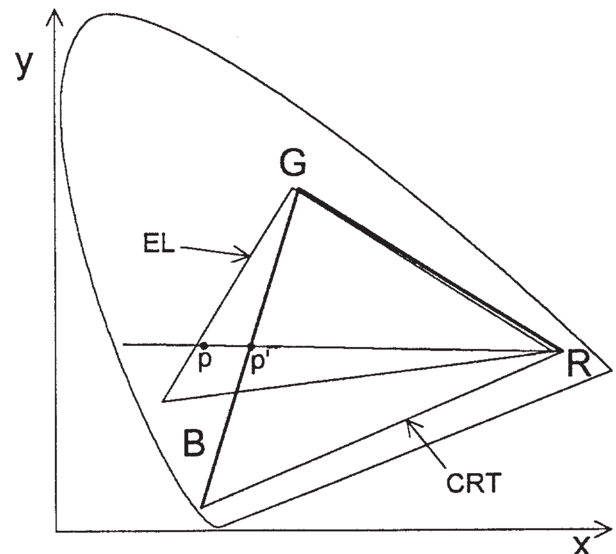

Figure 5. Phosphor gamuts (not to scale)

An alternative way to handle out-of-gamut values would be to move them to the CRT gamut border along the line joining the out-of-gamut color and the reference white. Since only the most saturated colors would be affected by our selection of out-of-gamut handling methods, we selected the simplest method to avoid further computational complexity.

The color transformations described have been implemented on the Princeton Engine with analog baseband composite-NTSC video signals input from either an over-the-air broadcast receiver, videotape player, or video laserdisk player. The NTSC signal is decoded digitally into red, green, and blue components. The color transforma-tion is defined by nine user-adjustable parameters and the electro-optical transfer function is characterized by an adjustable nine-point piecewise-linear function for each of the red, green, and blue channels. The calculations for each video frame are all done within one frame time so the modified output video sequence is displayed synchronously with the input. All the program parameters may be changed using a graphical display program and mouse to select the parameter and adjust its value. Update of the parameters within the program takes place every video frame, thus the output image reflects the change within 1/30 of a second.

Use of the mouse to adjust program parameters requires that the operator redirect attention from the video output CRT to the workstation screen to manipulate the controls. Therefore, a second method for adjusting parameters has been constructed. A rotary optical encoder has been interfaced with the Princeton Engine through an additional video port. Turning (or spinning) the knob sends a signal proportional to the rate of rotation that can be integrated by the transformation program to control any selected parameter (e.g., contrast, brightness, or tint), thus permitting the operator to adjust the image while maintaining eye contact.

Figure 6 is a still frame of a video sequence showing the split-screen capability with the image that would be produced on a CRT on the left, and the simulation of the EL panel on the right. The resulting EL image is very bright, consistent with the implications of an overall nonlinear system, and the colors tend to be more cyan, i.e., the sky and all blues are noticeably more cyan and the greens are warmer. Figure 7 demonstrates that an improved display can be obtained by adjusting the contrast, tint, and brightness to achieve acceptable skin tones, but at the expense of cooler blues and greens. (See next page for figure 6 and 7).

Conclusion

We have implemented a general nonlinear color transformation for video display simulation using a video supercomputer, the Princeton Engine. Such a device permits us to view the effects of our simulation in real time on full-motion video. As a demonstration, we simulated the connection of a linear electroluminescent display (employing currently available phosphors) to an NTSC video signal source designed for a CRT. The video signals were adjusted in real time using the supercomputer so that a CRT would simulate the appearance of the electroluminescent display.

* *Electricity Division, Electronics and Electrical Engineering Laboratory, Technology Administration, U.S. Department of Commerce. This is a contribution of the National Institute of Standards and Technology; not subject to copy right.*

References

1. L. DeMarsh, "Colorimetry for HDTV," *IEEE Trans. Consumer Electronics*, Vol. **37**, No. 1, pp. 1-6, Feb., 1991.
2. L. A. Harwood, E. J. Witmann, J. Hettiger, and R. L. Shanley, "Integrated NTSC Chrominance/Luminance Processor," *IEEE Trans. Consumer Electronics*, Vol. **CE-26**, pp. 693-706, Nov., 1980. See also the cited references to Harwood's previous publications.
3. N. W. Parker, "An Analysis of the Necessary Decoder Corrections for Color Receiver Operation with Non-Standard Receiver Primaries," *IEEE Trans. Broadcast and Television Receivers*, Vol. **BTR-12**, No. 1, pp. 23-32, 1966.
4. R. Sasaki, Y. Nagaoka, T. Tomimoto, "Non-Linear Chrominance Signal Decoding," *IEEE Trans. Consumer Electronics*, Vol. **CE-23**, No. 3, pp. 335-341, Aug. 1977.
5. C. B. Neal, "Computing Colorimetric Errors of a Color Television Display System," *IEEE Trans. Consumer Electronics*, Vol. **21**, pp. 63-73, Feb. 1975.
6. D. Chin, J. Passe, F. Bernard, H. Taylor, and S. Knight, "The Princeton Engine: A Real-Time Video System Simulator," *IEEE Trans. Consumer Electronics*, Vol. **34**, No. 2, pp. 285-297, May 1988.
7. Society for Motion Picture and Television Engineers, Proposed SMPTE Standard for Television: SMPTE 170M, "Composite Analog Video Signal - NTSC for Studio Applications," *SMPTE Journal,* Vol. **101**, No. 1, pp. 97-105, Jan., 1993.
8. W. A. Barrow, R. C. Coovert, E. Dickey, C. N. King, C. Laakso, S. S. Sun, R. T. Tuenge, R. Wentross, and J. Kane, "A New Class of Blue TFEL Phosphors with Application to a VGA Full-Color Display," *Soc. Information Display In ternational Symposium*, Seattle, Washington, pp. 761-764, May 18-20, 1993.
9. For a discussion on colorimetry and the Commission Internationale de l'Eclarirage (CIE) color spaces, see G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd Ed., Chap. 3, "Colorimetry" (John Wiley & Sons, New York, 1982).
10. K. B. Benson and J. C. Whitaker, *Television Engineering Handbook*, Revised Ed. (McGraw-Hill, New York, 1992) pp. 2.44-2.47.
11. T. Rzeszewski, Ed., and A. Troiano, Assoc. Ed., *Color Television* (IEEE Press, New York, 1983). For a review of matrixing and color errors, especially see the included article: C. Bailey Neal, "Television Colorimetry for Receiver Engineers," *IEEE Trans. Broadcast and Television Receivers*, Vol. **BTR-19**, pp. 149-162, Aug. 1973.

Figure 6. NTSC original video (left side) compared to EL model (right).

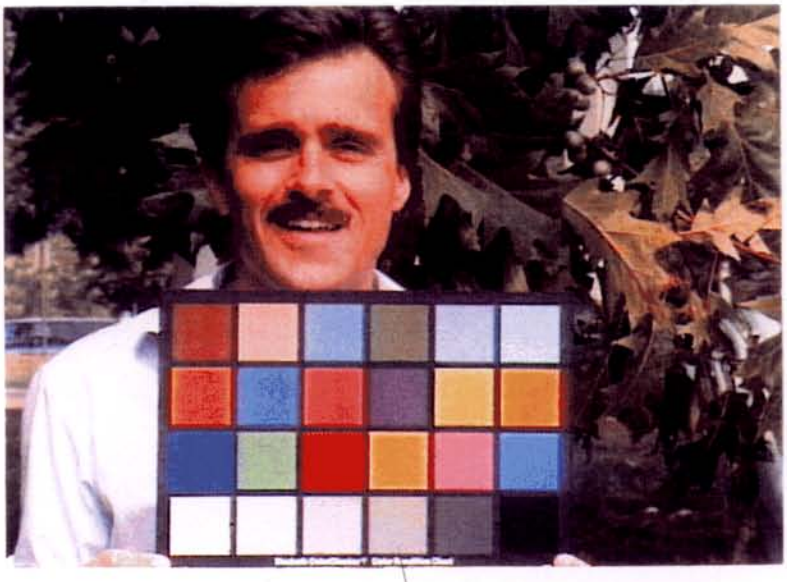

Figure 7. EL model adjusted with contrast, brighness and tint.