Device-Independent Rendering in Display Color Space

Konstantin V. Kolchin and Andrei B. Khodulev

Keldysh Institute of Applied Mathematics RAS
Moscow, Russia

Abstract

A scheme for device-independent calculation of the color of light reflected by, or transmitted through, an object is presented. This scheme involves expressing object and light colors in display color space, so it is suitable for hardware implementation. Surface color is considered to be a linear operator acting on a 3-vector representing light, so the color of reflected (transmitted) light is the product of a matrix and a vector. Because contemporary hardware does not support matrix representation of color, we suggest a simplified version of the scheme, which uses only 3-vector representation of color.

Keywords: *Color Matching , Device-independent Rendering.*

1. INTRODUCTION

Device-independent color reproduction in image rendering is essential for CAD applications of computer graphics [1, 2]. The color of light reflected by, or transmitted through, an object is found most accurately by spectral rendering, but rendering in a tristimulus color space is most frequently used in practice. Color reproduction in such rendering is attempted to be ensured by two major types of methods. In the first, rendering is performed in a device-independent color space (spectrum, the CIE XYZ space or some standard RGB space), and colors are mapped from this space to the color space of a display device. The second approach exploits conversion of object and light colors into device color space and rendering in this space.

Both approaches have their advantages and disadvantages. The first one ensures the device independence of rendered colors (at least, those which are within device gamuts), but the mapping from the rendering space to a device space imposes an additional computational burden and impedes the use of 3D graphics hardware. The second approach can employ hardware-based rendering, but the result is device-dependent because simulated color of light reflected by, or transmitted through, an object is calculated as the component-by-component product of object and light RGB triplets, and this product is different in different RGB spaces. The latter difference increases when interreflections are taken into account.

We propose a modified version of the second approach, in which object color is interpreted as a linear operator, so that the color of reflected or transmitted light is the product of a matrix and a vector representing incident-light color in the display device space.

Our method can be regarded as a variant of device-directed rendering, proposed in [3]. The difference is that our primary goal is not mapping out-of-gamut colors, but the device independence of rendered images.

In addition, we ignore issues related to gamma correction.

2. ALGORITHM

Consider two color spaces - the reference one, which may be an RGB space or the CIE XYZ or the spectral space, and the display one -- the tristimulus RGB color space of a display device. We make the following assumptions:

1. Conversion from the reference space to the device space is linear and is described by a matrix, which we denote *A*, or A_{ik} in components, where i = R, G, B; k = R, G, B, if the reference space is an RGB one, and k = X, Y, Z if the reference space is the CIE XYZ space, and k = index of wavelength band if the reference space is the spectral one. The conversion can include the relative colorimetric gamut mapping [4] or any linear chromatic-adaptation transform, like that used in CIECAM97s [5]. But in the latter case, the parameter *p* of this model should be set to unity (in fact, it is close to unity for most standard illuminants) for the conversion to be linear.

2. Light is a 3-vector, so light color in the display space L is expressed in terms of light color in the reference space L^{ref} as

$$L_i = \sum_k A_{ik} L_k^{\text{ref}}; \qquad (1)$$

3. Color of object is the linear operator acting on a 3-vector representing light, whose matrix in the reference space is diagonal, i.e.

$$C_{ik}^{\text{ref}} = \sum_k c_i \delta_{ik};$$

To calculate the matrix representing object color in the display space C, we should require that the color of reflected (transmitted) light in the display space is expressed in terms of that in the reference space in accordance with assumption 2:

$$C\mathbf{L} = AC^{\mathrm{ref}}\mathbf{L}^{\mathrm{ref}}. \qquad (2)$$

Substituting (1) into (2) we find:

$$CA\mathbf{L}^{\mathrm{ref}} = AC^{\mathrm{ref}}\mathbf{L}^{\mathrm{ref}}. \qquad (3)$$

If we want this equation to hold regardless of color of illuminating light $\mathbf{L}^{\mathrm{ref}}$, we get the following equation for the unknown object color matrix C:

$$CA = AC^{\mathrm{ref}}. \qquad (4)$$

If the reference space is 3-dimensional, this equation allow the solution:

$$C = AC^{\mathrm{ref}}A^{-1}, \qquad (5)$$

and C is not necessarily diagonal. For the case of spectral reference space, an exact solution of equation (4) is impossible in general; only approximate solutions can be found [2].

Because the conversion from the reference space to the device space is linear, conversions (1) and (5) can be made before rendering. In other words, colors of objects and light sources are calculated at the stage of the processing of scene specification in accordance with equations (1) and (5). Whatever local illumination model is used to describe light interaction with objects in rendering, it should be extended in such a way that object color is the 3×3 matrix computed by equation (5), and light after interaction is the product of this matrix and a 3-vector representing light.

3. SIMPLIFIED VERSIONS

For hardware implementation, when interreflections are ignored, and under the assumption that the reference space is a tristimulus space, the above scheme can be simplified by replacing assumption 2 with $\mathbf{L} = \mathbf{L}^{\mathrm{ref}}$. From (2), we then find the matrix of object color to be

$$C = AC^{\mathrm{ref}}.$$

However, contemporary 3D graphics hardware and standards for it, such as OpenGL, do not allow the matrix description of object colors.

Because of this, we suggest one more simplified version of rendering in display color space. Namely, we retain assumptions 1 and 2 and require additionally that object color in the display color space be a diagonal matrix. In such a case, equation (3) does not allow a unique solution suitable for all light colors $\mathbf{L}^{\mathrm{ref}}$. To find a reasonable approximation, we choose one representative light color $\mathbf{W}$ and require equation (3) to hold for this color. This gives us

$$C = AC^{\mathrm{ref}}\mathbf{W}^{ref} \;/\; A\mathbf{W}^{ref} \qquad (6)$$

where "/" means the component-by-component division. In particular, if the reference space is an RGB space, and the representative color is the white represented in the reference space by the RGB triplet (1, 1, 1), equation (6) reduces to

$$C_i = \sum_k A_{ik} c_k \;/\; \sum_j A_{ij}.$$

where the denominator is the 3-vector for the white light in the display space.

4. ACKNOWLEDGEMENTS

The work was in part supported by Russian Basic Research Foundation, grant no. 98-01-00547.

We are grateful to D. D. Zhdanov and Prof. Yu. M. Bayakovskii for useful discussions.

5. REFERENCES

[1] T. Oshima, et. al. *Computer Graphics Forum (Proc. EUROGRAPHICS '92)*, vol. 11, no. 3, p. C-381 (1992).

[2] R. Schettini, *Computer Graphics Forum*, vol. 13, no. 4, p. 211 (1994).

[3] A.S. Glassner, K.P. Fishkin, D.H. Marimont, and M.C. Stone, *ACM Transactions on Graphics*, vol. 14, no. 1, p. 58 (1995).

[4] R.D. Hersch and V. Ostromoukhov, *Proceedings of GraphiCon'98*, p. 11.

[5] The CIE 1997 Interim Colour Appearance Model (Simple Version), CIECAM97s, CIE TC1-34. (it can be freely downloaded from http://www.cis.rit.edu/people/faculty/fairchild/CAM.html).

Authors:

Konstantin V. Kolchin, researcher of Scientific and Technological Computational Center of Keldysh Institute of Applied Mathematics.
Address: 4 Miusskaya square,
Moscow 125047, Russia
E-mail: okk@gin.keldysh.ru

Andrei G. Khodulev, senior researcher of Keldysh Institute of Applied Mathematics.
Address: 4 Miusskaya square,
Moscow 125047, Russia.
E-mail: abkhod@gin.keldysh.ru

Независящий от устройства рендеринг в цветовом пространстве дисплея

К. В. Колчин, А. Б. Ходулев