Mitsubishi Electric Research Laboratory
MERL - Cambridge Research

Technical Report 2001-TR2001-09 March 27, 2001

A New Representation for Device Color Gamuts

Ronald N. Perry
Sarah F. Frisken

Abstract

An important part of color reproduction is the rendering of colors onto display devices, where the colors may be spot colors (e.g., tints) or the colors of a digitized photograph. Display devices, such as monitors and printers, are capable of showing only a subset of all perceivable colors; this subset is called the device color gamut.

Colors subjected to a reproduction system may not fall within the gamut of the target display device. In these cases, the colors cannot be exactly reproduced. When this occurs, some action is required, either notifying the user to modify the colors to fit within the gamut, or employing an algorithm to automatically compress the colors according to some predetermined strategy. In either case, a reliable technique is needed to determine whether or not a color lies inside a device gamut. When considering applications involving high resolution digitized images containing millions of colors, this gamut test must be very fast.

In this report, we identify a new variation of adaptively sampled distance fields (ADFs) for representing device color gamuts that overcome limitations of previous work. Associated algorithms for gamut testing and gamut construction are discussed. We then describe how the new representation fits nicely into the computational framework used in the ISO ICC specification, the international standard for color reproduction adopted by the industry.

Submitted to SIGGRAPH 2001 Conference Abstracts and Applications.

201 Broadway, Cambridge, Massachusetts 02139

A New Representation for Device Color Gamuts

Ronald N. Perry and Sarah F. Frisken
Mitsubishi Electric Research Laboratory

Introduction

An important part of color reproduction is the rendering of colors onto display devices [1], where the colors may be spot colors (e.g., tints) or the colors of a digitized photograph. Display devices, such as monitors and printers, are capable of showing only a subset of all perceivable colors; this subset is called the device color gamut.

Colors subjected to a reproduction system may not fall within the gamut of the target display device. In these cases, the colors cannot be exactly reproduced. If no corrective action is taken, the reproduction of graphics having out of gamut colors results in undesirable distortions of the tonal and color range of the graphic. When this occurs, some action is required, either notifying the user to modify the colors to fit within the gamut, or employing an algorithm to automatically compress the colors according to some predetermined strategy to fit within the gamut. In either case, a reliable technique is needed to determine whether or not a color lies inside a device gamut. When considering applications involving high resolution digitized images containing millions of colors, this gamut test must be very fast.

Prior Art

Current practice [2] tests colors by using a three-dimensional look up table (LUT). Each dimension, or axis of this LUT represents a channel of a tristimulus color. Each axis is uniformly sampled at discrete values. The LUT cell addressed by a set of channel samples defining a sample color contains information indicating whether or not that sample color lies inside the gamut. When a color is tested using this LUT, the sample intervals along each axis are found which contain the value of each color channel. A color being tested whose channel values are not exactly represented by the samples in the table are estimated by some interpolation scheme based on the relative locations of each color channel within their respective bounding sample intervals.

[2] uses the LUT cells to contain Boolean information (i.e. inside or outside) to indicate the state of the color addressing the cell. This technique is subject to two major flaws. First, there is no information available to indicate how far out of gamut the color is. For colors which are just barely out of gamut, a user may choose to ignore the minor adjustments needed to bring them into gamut. However, if the colors are very far out of gamut, they will require attention. The Boolean LUT result cannot distinguish between these cases.

The second, and more serious flaw is that since the actual function representing the result of a color test is discontinuous at the gamut boundary (a Boolean inside or outside), a LUT containing samples of this function may give incorrect results in the interval containing the actual gamut surface. The incorrect results in this interval are due to the use of a continuous reconstruction function (e.g., trilinear interpolation) to obtain an estimate of a discontinuous, sampled function. Because of this, inaccurate color estimates and substitutions are made, random errors are introduced, and colors which are out of gamut may not be recognized as such while colors which are in gamut may be erroneously identified as being out. The net result of these shortcomings is the expensive rework of a production job, often requiring considerable operator intervention and successive trial and error attempts to obtain output that is acceptable for the intended use.

Although numerous research papers [1] have presented various other methods for the representation of device gamuts, these methods have been essentially ignored by industry because of their inefficiencies and/or their limited ability to represent complex gamut topologies.

Adaptively Sampled Distance Fields

As suggested in [3], adaptively sampled distance fields (ADFs) can be used to represent a device gamut. ADFs adaptively sample the signed distance field of an object (i.e. gamut) and store the sample values in a spatial hierarchy (e.g., octree) for efficient processing. An ADF gamut representation overcomes the two identified flaws with Boolean LUTs and provide other advantages. Specifically, the reconstructed distance at a query color point provides an indication of how far out of gamut that point is. Because ADFs contain samples of a continuous function (rather than a discontinuous function), and use a continuous reconstruction function (trilinear interpolation), the resulting reconstructed samples are very accurate.

In this sketch, we identify some of the obstacles with the approach suggested in [3] and outline improvements. We then describe how these improvements fit nicely into the computational framework used in the ISO ICC specification [2], the international standard for color reproduction adopted by the industry (e.g., Apple, Microsoft, and Sun).

Bounded-Surface Generation

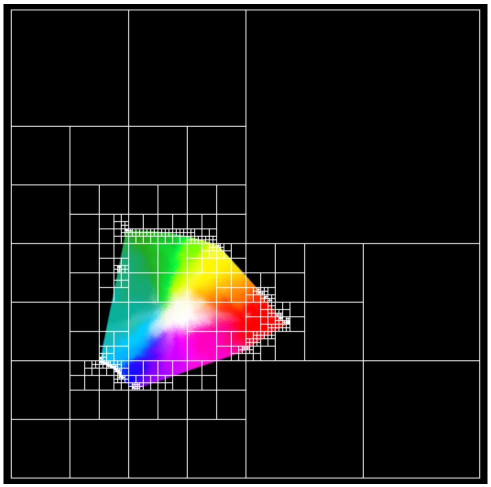
The ADF (i.e. gamut) generation algorithm presented in [3] does not subdivide interior or exterior cells. This surface-limited generation reduces memory requirements while assuring accurate representation of the distance field in boundary cells, which is sufficient for many types of processing (e.g., rendering). However, exterior and interior cells very close to the surface may not accurately reproduce the distance field in those cells, thus leading to color errors similar to those described for Boolean LUTs. To overcome these problems, the generation algorithm is modified so that exterior and/or interior cells are not coalesced unless their parent cells satisfy a maximum error constraint whenever those parent cells are within a bounded region defined by a specified minimum distance from the surface [4].

Contiguous ADFs

The octree ADFs (and algorithms) suggested in [3] are not optimized for gamut testing; the ADFs consume too much memory, lack memory coherency, take too long to generate, and are limited (in practice) to a maximum tree depth of 9. To overcome these problems, we construct a contiguous ADF representing the gamut using the tiled bounded surface generation algorithm outlined in [4]. Representing the ADF in a contiguous block is important for conforming to the ICC framework (see below). We have also trimmed the cell data structure to contain only the necessary elements for reconstruction (the primary operation in gamut testing) and have found that 8 or 16 bit distance values (rather than 32) are sufficient. This new ADF representation is compact and amenable to a hardware implementation.

Integration into the ICC Framework

At the heart of the ICC framework [2] is a data structure which describes how colors are transformed. This structure converts an input color into an output color using tables with 8 or 16 bit precision and contains four processing elements: a 3 by 3 matrix, a set of 1D input LUTs, a single multi-dimensional LUT, and a set of 1D output LUTs. Data is processed using these elements via the following sequence: [matrix] → [1D input LUTs] → [multi-dimensional LUT] → [1D output LUTs]. This same mechanism is used for gamut testing, with the requirement that there is a single output interpreted as either "in-gamut" if zero or "out-of-gamut" if non-zero. The multi-dimensional LUT uses trilinear interpolation. Custom ASICs have been developed by Kodak and LinoColor to implement this processing model. Because the new contiguous ADFs use trilinear interpolation for reconstructing distances, the proposal in this sketch could be easily adopted by the ICC standard by storing the ADF in the memory typically used to store the multi-dimensional LUT and by inserting an ADF cell location step (requiring at most O(log n) operations, where n is the maximum number of cells on a side of the ADF) into the processing model that identifies the ADF cell to interpolate.



The CIE chromaticity diagram showing an ADF gamut of an offset printing press.

References

[1] M. Bourgoin, D. LeHoty, C. Poynton, A. Ramamurthy, M. Stone. Color Science and Color Management for CGI and Film. In *SIGGRAPH 2000 Course Notes*.
[2] ISO Specification ICC.1:1998-09, File Format for Color Profiles.
[3] S. Frisken, R. Perry, A. Rockwood, and T. Jones. Adaptively Sampled Distance Fields: A General Representation of Shape for Computer Graphics. In *SIGGRAPH 2000 Conference Proceedings*.
[4] R. Perry and S. Frisken. Kizamu: A system for sculpting digital characters. In *SIGGRAPH 2001 Conference Proceedings*.