A Single Chip Color Processor for Device Independent Color Reproduction

H. Kotera, K. Kanamori, T. Fumoto, O. Yamada, and H. Motomura
Matsushita Research Institute Tokyo, Inc., Japan

M. Inoue
Matsushita Electronics Company, Japan

Introduction

A common requirement in the different color systems is "how to interchange a device independent color" The major objective of PRISM processor is to answer this requirement. In order to exchange the colors between different systems, a device dependent signal to/from the system must be converted into a device independent standard signal. The novel algorithm, "PRISM (Programmable Interpolation by Small Memory)" has been proposed.[1] It has the following features to be applied for

- Device independent color interchange.
- Simple structure for LSI chip design.
- Reduced computations for real time operation.
- User definable applications.

Concept of Device Independent Color System

In general, color image input/output devices are designed to send/receive their own color signals to reproduce source colors. Input devices A, B, and C such as TV camera, scanner, and facsimile transmitter will send different color signals (*a,b,c*) for the same source color *S*. Output devices D, E, and F such as TV monitor, printer, and copier will need to receive different color signals (*d,e,f*) to reproduce the same destination color *S*. These signals (*a,b,c*) and (*d,e,f*) are device dependent. Then, if the signals (*a,b,c*) are converted into a standard signal *s*, all the input devices A, B, and C look like the same. Here, a standard signal *s* is desired to carry a correct tristimulus value for a source color *S*.

Next, if the standard signal *s* is converted into the signals (*d,e,f*) and fed to the devices, the same destination color *S* will be gotten from the output devices D, E, and F [Fig. 1].

PRISM color processor P is placed in back of the input devices or in front of the output devices. It works to convert the device dependent signals to/from a standard signal. As a result, all the input/output devices could be connected as if they were imaginary the same devices.

In the ideal color system, a tristimulus signal from source color *S* is to be transparently transmitted to the destination and correctly reproduced on the output devices.

A tristimulus signals from a source color *S* will take a variety of linear or nonlinear distortions when passing through the input/output devices. In the model, these distortions are eliminated by the color processor P which compensates them inversely [Fig. 2].

For instance, an output color printer should be driven by the pre-distorted signal *y* to reproduce the correct destination color *S* as same as the source, when the relation *y* vs. *s* is given as $s = \Phi(y)$ by the transfer function Φ. Hence, the color processor P, is placed in front of the printer so that it can convert the input color signal *s* to the pre-distorted signal *y* through the inverse transformation $y = \Phi^{-1}(s)$.

General Purpose Color Transform By LUT

Generally, color conversions are defined by a function of relating the input color (R,G,B) to the output color (R',G',B') as

$$\begin{aligned} R' &= r\,(R,G,B) \\ G' &= g\,(R,G,B) \\ B' &= b\,(R,G,B) \end{aligned} \tag{1}$$

Any function is realized by LUT (Look-Up-Table), where all the input/output relations can be stored whatever they are complicated. However, LUT method takes too much memories, for example, about 50 mega bytes for 8 bits × 3 color inputs and 3 giga bytes for 10 bits × 3 color high precision systems. Moreover, it takes too much time to make the LUT, which can not be changed quickly in response to different applications.

"PRISM" Architecture

PRISM color processor has been developed to solve these problems in full-size LUT.

A novel color processor is composed of two fundamental parts, "small LUT memory" and "PRISM interpolator" [Fig. 3]. Here, LUT stores the output data at coarse lattice points of the input color space and the PRISM interpolator completes the precision output data for intermediate inputs between lattice points.

User can define any kind of conversion by just making the small LUT and change it quickly according to the applications.

The small LUT is looked up by the upper bits of input data, then the precision output is calculated from the coarse lattice data by 3-D interpolator using the remaining lower bits. In the case of 8 bits × 3 color system, for example, each pixel data of the input color image are divided into 3 upper bits and 5 lower bits. Here, the input

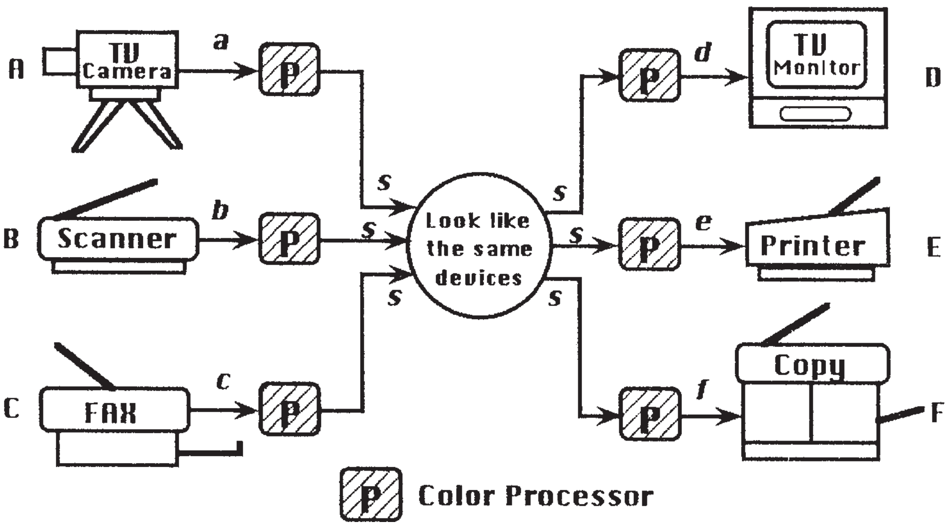

Figure 1. Concept of Device Independent Color System

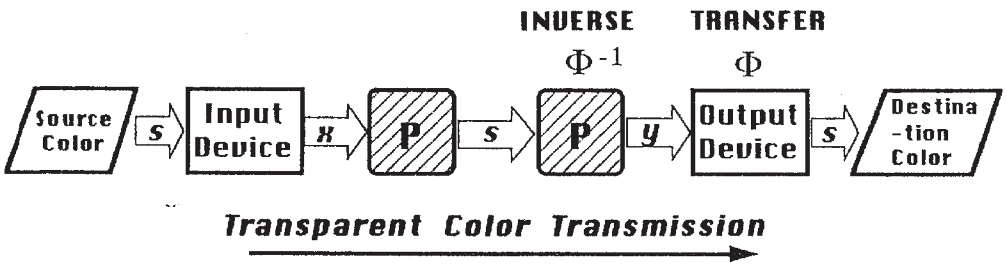

Figure 2. Conceptual Model of Transparent Color Reproduction

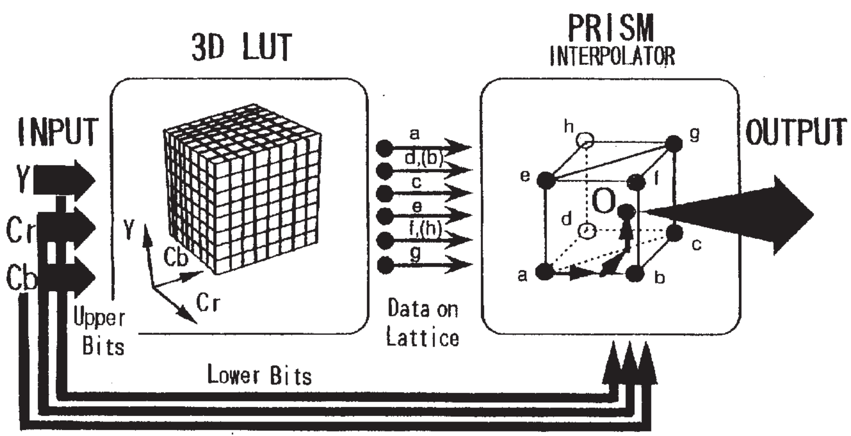

Figure 3. Structure of Color Processor using 3D LUT and Interpolation

color space is divided into 512 (8×8×8) unit cubes which include 729 (9×9×9) lattice points. A unit cube is divided into two PRISMs.

PRISM Interpolation

Conventional interpolators, "CUBE"[2], "TETRAHEDRON"[3,5], and "PYRAMID"[6] are defined in color mixture spaces such as RGB or CMY. CUBE is most popular and considered to be accurate because of using 8 lattice points at once. TETRAHEDRON using 4 points is most simple, while PYRAMID is composed of two TETRAHEDRONs and uses 5 points [Fig. 4].

A PRISM element is basically defined in "luminance-chrominance" spaces, such as YC_rC_b, LHS, or CIELAB [Fig. 5]. PRISM structure is more fittable to the perceptual color managements than CUBE or TETRAHEDRON. PRISM has a luminance axis perpendicular to a triangle plane and the triangle makes the minimum chrominance plane in 2D space. Hence, PRISM is a minimum 3-D unit element to formulate "luminance-chrominance" color space.

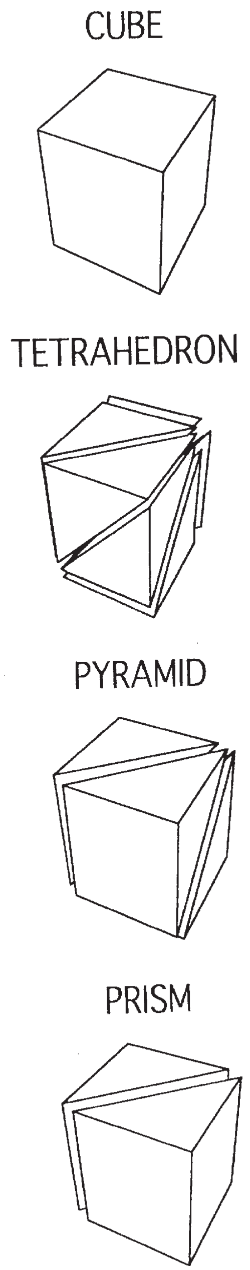

Figure 4. Interpolation Methods

In PRISM interpolator, for example, (Y, C_r, C_b) unit cube is divided into two elements, PRISM (0) and PRISM (1). Here, C_r, C_b are assigned to a triangle base plane and Y to the main axis perpendicular to the triangle. (Y, C_r, C_b) denote the displacements from a base point a.

The output value $(\tilde{0})$ at an input point 0 inside a PRISM (0), is interpolated by the weighting sum of output data (a) ~ (g) on six lattice points a - g for $C_r > C_b$ as follows.

$$(m) = (a) + [(b) - (a)]\, C_r + [(c) - (b)]\, C_b$$
$$(n) = (e) + [(f) - (e)]\, C_r + [(g) - (f)]\, C_b \qquad (2)$$
$$(\tilde{0}) = (m) + [(n) - (m)]\, Y$$

It is noted that the weighting coefficients include 2nd order non-linear cross terms such as YC_r and YC_b.

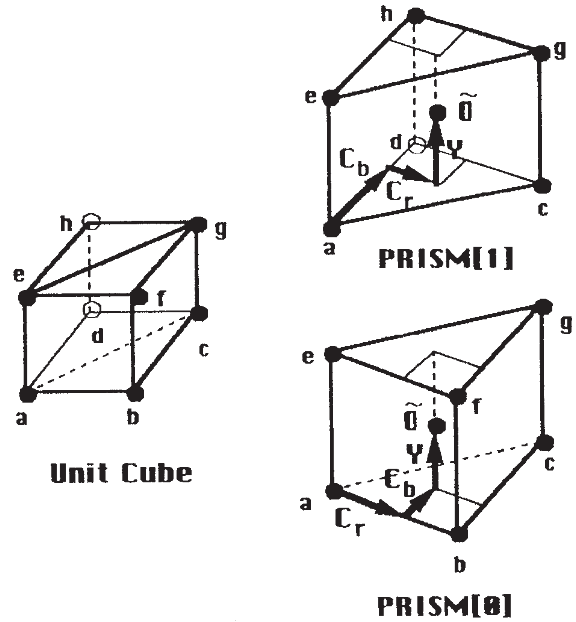

Figure 5. PRISM Interpolation

Estimation of Interpolation Error

The interpolation errors were evaluated for typical color space conversion between NTSC-RGB and CIELAB. First, the errors were numerically estimated for gray scale inputs.[1] In the forward conversion from RGB to LAB, PRISM showed the intermediate characteristics between TETRAHEDRON and CUBE and in the reverse from LAB to RGB, it gave the smooth gray scales better than PYRAMID and TETRAHEDRON, rather close to CUBE. PYRAMID showed similar results to TETRAHEDRON but no merits because of its computational complexities.

Next, the conversion test has been done for full color natural images. Here, the interpolation errors are shown for a typical standard image with 24 bits × 512 × 512 pixels.

The conversion errors are compared with each other in both ways of RGB to/from LAB by root mean square color difference ΔE_{rms} and maximum error ΔE_{max} in CIELAB space. The LUT size was changed from (5×5×5) to (33×33×33). Then, the each color pixel data are divided into upper (2+2+2) ~ (5+5+5) bits and lower (6+6+6) ~ (3+3+3) bits for (R+G+B). The upper bits are assigned to input address for 3D-LUT and each address corresponds to 125(5×5×5) ~ 35937 (33×33×33) lattice points in the given input color space.

The results are summarized in Fig. 6. CALCULATION means the accurate calculation using 8 bit integer data. As for the small LUT size of 5×5×5 and 9×9×9, CUBE is the best and PRISM comes next. ΔE_{rms} in PRISM for 729(9×9×9) lattice points was 5.17. This is more accurate than TETRAHEDRON and PYRAMID and close to CUBE. This value is not always sufficient for higher precision color reproduction such as photogravure printing but is considered to be practical for color copier or printer in office use. In the larger LUT size than 17×17×l7, all the methods are much the same.

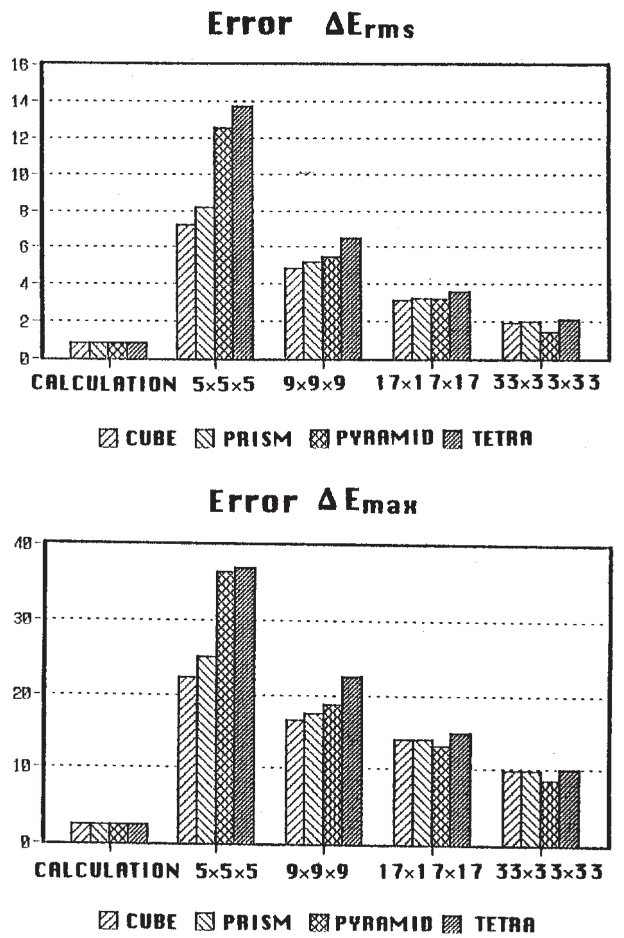

Figure 6. Interpolation Errors

Of course, the larger the LUT size is, the smaller the interpolation error, but the LUT larger than 33×33×33 will lose the cost merit of interpolation.

LSI Processor MN5511

PRISM algorithm has been first implemented on a single chip LSI processor, MN5511 [Fig. 7, Table 1].

LSI is designed to operate at video rate of 34.9ns/pixel.

Table l. Specification of MN5511

Max. speed	34.9 (ns/pixel)
Clock rate	28.7 (MHz)
Input data	8 (bit)/10 (bit)
Output data	8 (bit)/10 (bit)
1/0 bit width	8 (bit)/16 (bit)
Interface	CMOS/TTL
Pipe-line step	10 clocks
Power	1000 (mW)
Supply	5 (Volt)
Package	148 QFP
Option	Area Disposal/ LUT No. Setting

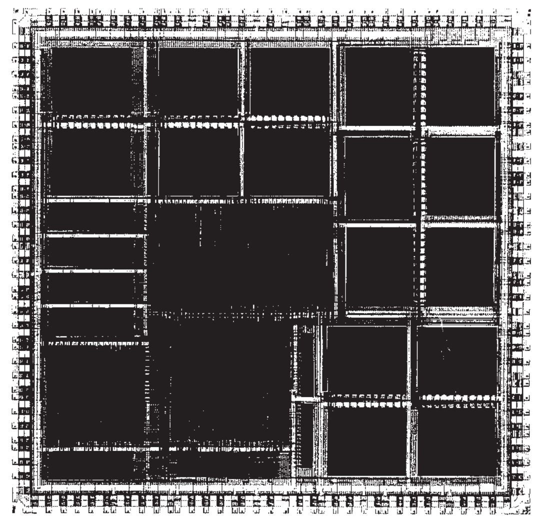

Figure 7. Color Processor MN5511

MN5511 has plural sets of on-chip RAM LUTs accessible in parallel and PRISM interpolator composed of plural multipliers and adders [Fig. 8].

The parallel pipelined processing is done by these hardwares according to the following equations using Table 2.

$$\begin{aligned} (t1) &= Cr\times(SUB1) + Cb\times(SUB2) \\ (t2) &= Cr\times(SUB4) + Cb\times(SUB5) \\ (\tilde{0}) &= (CRAM0) + (tl) + Y\times[(SUB3) \\ &\quad + (t2) - (t1)] \end{aligned} \qquad (3)$$

Applications

MN5511 is functionally equivalent to the full memory LUT. Then it can be used to any kinds of 3-D conversions from 3-color inputs to 3-color outputs. MN5511 have been tested in successful for the following applications.

l. Color Correction

Non-linear color correction such as quadratic masking has been done combined with UCR (Under Color Removal), getting 4-color prints in a single process through MN5511.

Table 2. Data for processor

RAM	PRISM(0)	PRISM(l)
(CRAM0)	(a)	(a)
(SUB1)	(a)	(a)
(SUB2)	(b) - (a)	(c) - (d)
(SUB3)	(e) - (a)	(e)- (a)
(SUB4)	(f) - (e)	(g) - (h)
(SUB5)	(g) - (f)	(g) - (e)

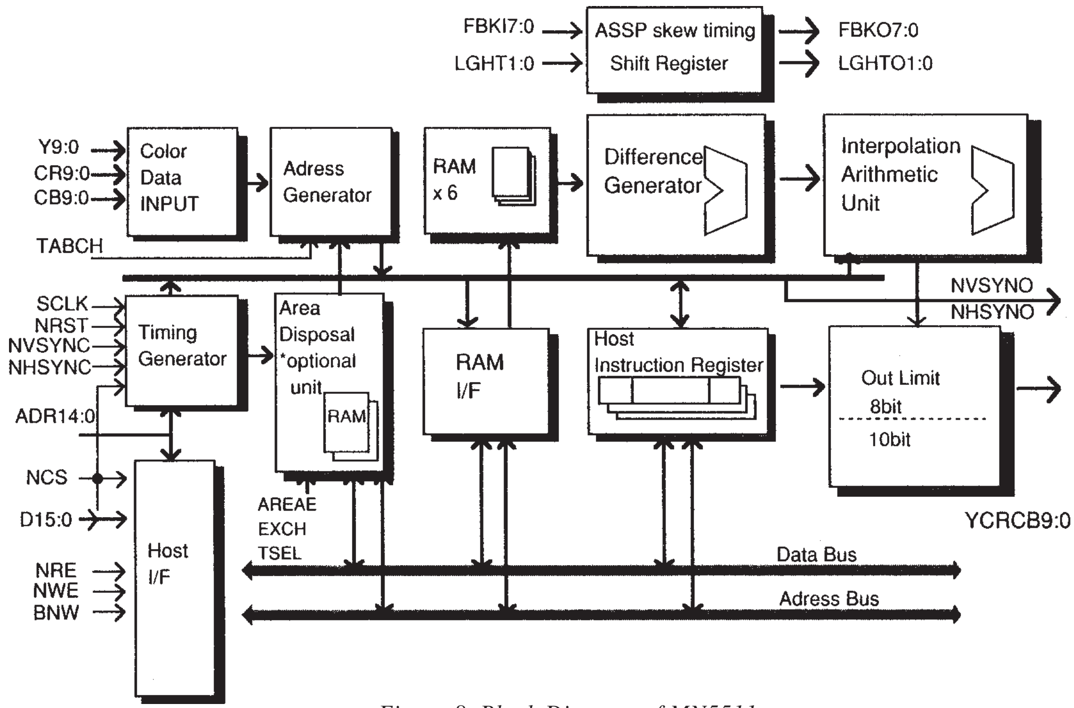

Figure 8. Block Diagram of MN5511

2. Selective Color Adjustments

Local color areas can be selectively changed by LUT on MN5511. A flesh tint area has been extracted and delicately adjusted in CIELAB cylindrical space defined by L* (metric lightness) - H°_{ab} (metric hue) - C^*_{ab} (metric chroma).

3. Chroma Key

Chroma key technique is often used to compose a video image with another different scene by separating the background color. A specified background color can be also separated from others by just setting null data to the local areas of LUT on MN5511 and replaced by other image.

4. Color Recognition

Colors can be used to recognize a specified object more effectively than monochrome. MN5511 has been tested to discriminate some colored parts for PC board inspection.

Conclusions

PRISM brought a single chip LSI flexibly transforms colors to/from the devices. LUT size of (9×9×9) was practical in its accuracy for office color equipments. The high precision colors will be obtained by applying (17×17×17) LUT.

The first engineering model MN5511 is expected to open the forthcoming office color age in its portability and high speed. The next high grade version is now under planning.

References

1. K. Kanamori, T. Fumoto, and H. Kotera, “A Color Transformation Algorithm Using “PRISM” Interpolation”, *IS&T’s 8th Int’l. Cong. on Adv. in NlP.*, pp. 477-482 (1992).
2. Po-Chien Hung, “Color Reproduction Using 3-D Interpolation”, *Imaging Applications in the Work World, SPIE* vol. **900** (1988).
3. K. Kanamori and H. Kotera, “Color Correction Technique for Hardcopies by 4-Neighbors Interpolation Method”, *J. IS&T.*, **36**,1, pp. 73-80 (1992).
4. K. Kanamori and H. Kotera, “A Method for Selective Color Control in Perceptual Color Space”, *J. IS&T*, **35**, 5, pp. 307-316(1991).
5. T . Sakamoto, Japan Patent, PO. 53-1232201 (1978)
6. Flanklin, U.S. Patent, 4334240.
7. T. Yamasaki, “Optimum Color Space for Color Data Exchange and Its Mutual Transformation to Other Color Spaces”, *IS&T’s 6th Int’l. Cong. on Adv. in NIP.*, pp. 837-843 (1990).