Color Image Resolution Conversion

Michael Vrhel, *Senior Member, IEEE*

***Abstract*—In this paper, we look at the problem of spatially scaling color images. We focus on an approach that takes advantage of the human visual system's color spatial frequency sensitivity. The algorithm performs an efficient least-squares (LS) resolution conversion for the luminance channel and a low-complexity pixel replication/reduction in the chrominance channels. The performance of the algorithm is compared to a LS method in sRGB and CIELAB color spaces, as well as standard bilinear interpolation in sRGB space. The comparisons are made in terms of computational cost and color error in sCIELAB.**

I. Introduction

RESOLUTION conversion is a common operation in the pipeline from image capture to image reproduction. The resolution of the digital camera or digital scanner often does not match that of the output device, making it necessary to perform a scaling of the image spatial coordinates. In other cases, it is desired to enlarge or reduce the captured image for aesthetic reasons.

Devices such as multifunction peripherals (MFPs), which FAX, copy, and scan with user-adjusted enlargement/reduction settings, have very demanding resolution conversion requirements. In addition, the embedded systems in these devices require computationally efficient algorithms.

Resolution conversion of monochrome images has been heavily researched [1]–[18]. A naive approach to resolution conversion in colorimetric images would be to apply the monochrome algorithms to the three RGB channels that make up the recorded image data. Such an approach would likely be suboptimal in terms of quality and/or computational efficiency.

Many algorithms developed for color images take advantage of the human visual system's reduced sensitivity to chrominance spatial variations compared to the sensitivity to luminance spatial variations. Examples include [25] for temporal noise reduction, [26] for computational improvements on color correction, [27] for color halftoning and of course color image and video compression methods, like JPEG and MPEG [23], [24]. Missing from the above list is a study of how the visual system's properties can be used in an improved resolution conversion method, which is the focus of this work. Some work has been done on reducing the artifacts introduced due to subsampling of chrominance channels in image compression methods [21], [22], but little or no research has been published on color image resolution conversion.

Manuscript received April 19, 2004; revised April 22, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhigang (Zeke) Fan.

The author is with ViewAhead Technology, Redmond, WA 98052 USA (e-mail: michael.vrhel@viewahead.com).

Digital Object Identifier 10.1109/TIP.2004.841194

In embedded systems, quality and computational efficiency are of a great concern and, typically, tradeoffs are required between them. In the case of image resolution conversions, the algorithms range from a low-quality, low computational-cost algorithm, like pixel replication/reduction to a high-quality, high-computational cost algorithm, like cubic spline least-squares (LS) resolution conversion. In this paper, we propose an algorithm for colorimetric image resolution conversion, which falls between these two extremes. The algorithm performs a high-quality LS conversion [1] for the luminance channel and a computationally low-cost pixel replication/reduction in the chrominance channels. The performance of the algorithm is assessed and compared with other approaches in terms of computational efficiency and colorimetric error in sCIELAB [20].

The paper is organized as follows. Section II provides the details of the proposed algorithm. Section III provides quantitative examples of the method's performance in terms of sCIELAB and computational efficiency.

II. Efficient Colorimetric Resolution Conversion

In studying colorimetric color conversion, there are essentially two variables to consider. One is the color space (e.g., sRGB) in which the conversion will occur; the other is the method (e.g., bilinear interpolation) used to scale the data. Here, we limit the color spaces to those that are commonly used as well as those that we suspect will provide the best performance. In terms of interpolation methods, we rely upon research that has been done on monochrome resolution conversion.

It is well known that the human visual system is much less sensitive to high-frequency chrominance errors compared to luminance errors. This characteristic can be used to guide us toward an algorithm that reduces computational cost without introducing large colorimetric errors. This suggests an algorithm that performs its processing in a color space that divides the luminance and chrominance channels. In this case, a high-quality interpolation algorithm should be used in the luminance channel, where high-frequency errors are more noticeable, while a low-quality, low computational-cost algorithm could be used in the chrominance channels, where high-frequency errors can be tolerated.

A block diagram of such an algorithm is shown in Fig. 1. The diagram shows a transformation to YCC color space [32]; however, other luminance-chrominance color spaces, such as CIELAB, could also be used where L^*, a^*, and b^*, replace Y, Cr, and Cb, respectively. The primary factors in this decision are the computational cost and the encodability of the processing color space. Let us now focus on the details of the algorithm shown in Fig. 1.

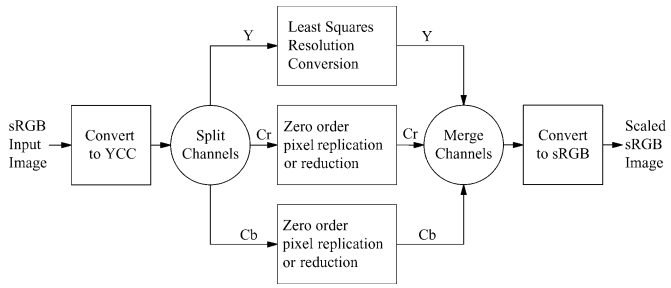

Fig. 1. Block diagram of proposed color image resizing method.

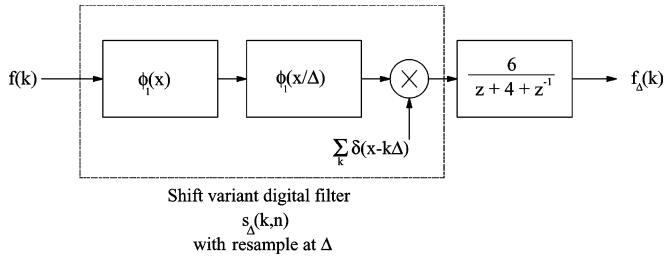

Fig. 2. LS resolution conversion for linear b-spline approximation function at a signal scaling of $1/\Delta$.

A. Luminance Channel

Unser *et al.* [1] introduced a monochrome resolution conversion process that was optimal in the LS sense. The algorithm is dimensionally separable, which is desirable for most embedded systems since the processing is usually more efficient. The derivation details of the algorithm are contained in [1].

Due to their favorable approximation and computational properties [29], we will use a b-spline representation for the input signal. In particular, we will use a first-order (linear) spline model allowing a fair comparison with the commonly used bilinear interpolation method [3, Sec. 4.3.2]. The linear b-spline function is given by

$$\phi_1(x) = \begin{cases} 1-|x| & |x| \le 1 \\ 0, & \text{else} \end{cases}. \tag{1}$$

The LS resolution conversion algorithm can be implemented as a series of digital filters as shown in Fig. 2. The last filter in Fig. 2 can be implemented as a series of recursive IIR filters as described in [2].

The first filter, which is represented by the dashed box and denoted by $s_\Delta(k,n)$ in Fig. 2, appears as a shift variant FIR filter (hence, the variables k and n) with a dependency upon the scale factor Δ. The filter $s_\Delta(k,n)$ is created from the convolution of two scaling functions, $\phi_1(x)$ and $\phi_1(x/\Delta)$ (see [1] for details). The filter's shift variance occurs due to the possible noninteger sampling rate change. Figs. 3 and 4 display a graphical example, where the filter $s_\Delta(k,n)$ is convolved with the input samples $f(k)$ when the signal is to be reduced by $1/\sqrt{2}$ (i.e., $\Delta = \sqrt{2}$). Note how the coefficients of $s_\Delta(k,n)$ change depending upon the spatial location of the data. In implementation, the filter coefficients can be computed from the continuous convolution

$$s_\Delta(x) = \frac{1}{\Delta}\phi_1(x) * \phi_1\left(\frac{x}{\Delta}\right). \tag{2}$$

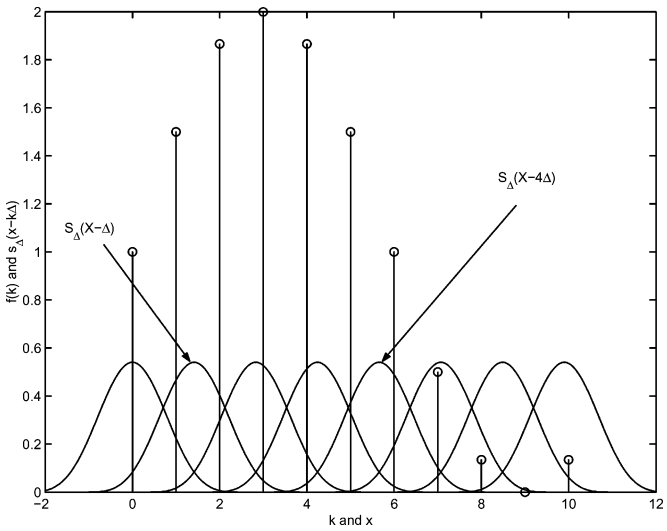

Fig. 3. Input signal and sample function $s_\Delta(x)$ for $\Delta = \sqrt{2}$.

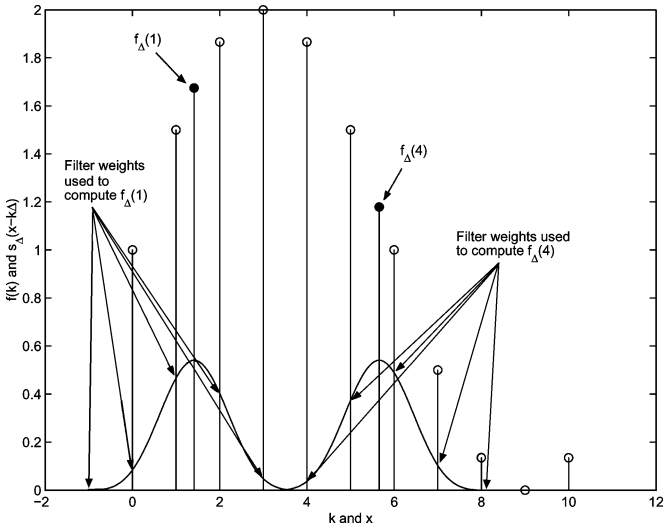

Fig. 4. Graphical illustration of convolution of $f(k)$ and $s_\Delta(k,n)$ for $\Delta = \sqrt{2}$. Note in digital filtering, the filter coefficients will appear to be shift variant.

If computational resources are limited, the filter coefficients of $s_\Delta(k,n)$ can be placed in a look-up-table (LUT) of size $S_\Delta \mathrm{x} N_\Delta$ where

$$N_\Delta = \text{floor}\left(\frac{N-1}{\Delta}\right) + 1 \tag{3}$$

is the size of the output signal, N is the size of the input signal and

$$S_\Delta = \text{ceil}\left(2 * (\Delta + 1)\right) \tag{4}$$

is the number of filter taps in $s_\Delta(k,n)$.

B. Chrominance Channels

For the chrominance channels, a zero-order b-spline approximation of the input signal $f(x)$ is used where

$$f(x) = \sum_{k=-\infty}^{\infty} f(k)\phi_0(x-k) \tag{5}$$

Fig. 5. Text input image to scaling algorithms.

Fig. 6. Child input image to scaling algorithms.

and

$$\phi_0(x) = \begin{cases} 1 & -\frac{1}{2} \le x < \frac{1}{2} \\ 0, & \text{else} \end{cases}. \tag{6}$$

Sampling the signal at a rate of Δ, to scale it by a factor of $1/\Delta$, results in

$$f_\Delta(n) = f(x)|_{x=n\Delta} = \sum_{k=-\infty}^{\infty} f(k)\phi_0(n\Delta - k). \tag{7}$$

From (6) and (7), it can be shown that $f_\Delta(n)$ is given by

$$f_\Delta(n) = f(k)|_{k=\text{floor}\left(n\Delta+\frac{1}{2}\right)} \tag{8}$$

which is one of the current image pixels. In this approach, no arithmetic operations are performed on the chrominance channel pixels. The existing signal samples are either replicated or removed. For implementation, a one-dimensional (1-D) LUT the size of the output signal dimension can be computed which indexes into the current signal.

III. Performance and Examples

The traditional colorimetric error quantities such as ΔE_{ab} and ΔE_{94} were developed for the comparison of solid patches. Since we will be performing the operations on images that contain spatial variation, it is necessary to consider an error measure that accounts for the human visual system's (HVS's) color spatial frequency sensitivity. The spatial CIELAB (sCIELAB) visual model provides one such measure [20] and is popular for such assessments [31, Section 1.10].

Using this measure, the algorithm (**LS-Y**) in Fig. 1, using the definition of YCC color space as defined in ITU-R BT.601 [32], was compared to the following methods:

1) **LS-sRGB**: an sRGB LS approach, which applied the algorithm in [1] to the bands of a sRGB [28] image;
2) **LS-LAB**: a CIELAB LS approach, which applied the algorithm in [1] to L*, a*, and b* image bands;
3) **Bi-sRGB**: a standard separable bilinear method applied to each of the bands of the sRGB image;
4) **LS-Y Bi-C**: a YCC approach, in which the algorithm in [1] was applied to the Y band and standard separable bilinear interpolation was applied to the Cr and Cb bands.

TABLE I
Average ΔE_s Values for Images

Method	Text		Child	
	ΔE_s CRT	ΔE_s 300dpi Print	ΔE_s CRT	ΔE_s 300dpi Print
LS-Y	1.58	0.83	0.72	0.37
LS-sRGB	0.67	0.14	0.27	0.042
LS-LAB	0.67	0.12	0.27	0.043
Bi-sRGB	1.63	0.92	0.81	0.44
LS-Y Bi-C	1.54	0.87	0.65	0.32

For each method, the input image was reduced and then enlarged by $\sqrt{2}$. The sCIELAB ΔE_s [20] between the input and output was computed, for two viewing conditions. The first was a CRT condition with a viewing distance of 18 in (45.72 cm) and a resolution of 72 dpi. The second was a print condition, with a viewing distance of 12 in (30.48 cm) and a resolution of 300 dpi. Two input images were tested and are shown in Figs. 5 and 6. The ΔE_s values (which were the average error across the image pixels) are given in Table I. The distribution of the errors for the child image using **LS-LAB**, **LS-Y**, and **Bi-sRGB** are shown in Fig. 7 for the 72-dpi viewing condition.

Computationally, at a scaling value of $\Delta = \sqrt{2}$ the LS algorithm requires five multiplies and four adds for each output of $s_\Delta(k, n)$. This is assuming the filter values are precomputed and stored in a LUT. The cost of this precomputation would be amortized across the pixels of the image and, hence, be a relatively small cost. At a scaling value of $\Delta = 1/\sqrt{2}$, t he LS

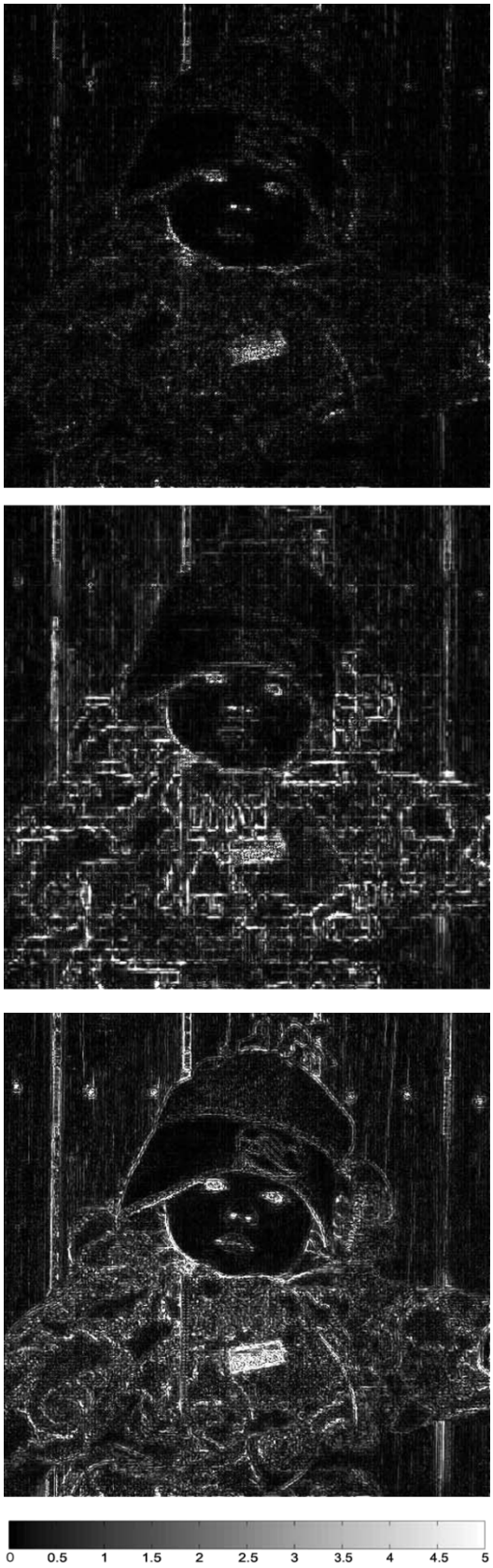

Fig. 7. ΔE_s error images for (from top to bottom) **LS-LAB**, **LS-Y**, and **Bi-sRGB**.

algorithm requires four multiplies and three adds for each output of $s_\Delta(k,n)$. Regardless of the scaling size, the IIR filter

TABLE II
COMPUTATIONAL COST OF METHODS

Method	Multiplies	Adds	Color Transformation	
			Multiplies	Adds
LS-Y	16	12	13	10
LS-sRGB	48	36	0	0
LS-LAB	48	36	24	18
Bi-sRGB	12	6	0	0
LS-Y Bi-C	24	16	13	10

Fig. 8. Interpolation comparison. From top to bottom, the image section is input image, **LS-Y**, **LS-sRGB**, **LS-LAB**, **Bi-sRGB**, and **LS-Y Bi-C**. Note the pixel errors in **LS-Y** versus the blurring errors in **Bi-sRGB**.

$6/(z+4+z^{-1})$ can be implemented with two additions and three multiplications for each output point [19]. Since the operations must be performed along each row and column, these values must be multiplied by two to obtain the cost per output pixel per color plane.

Implemented as a separable algorithm, the filter used for bilinear interpolation requires two multiplications and one addition for each output point. Again, these values must be multiplied by two to account for the processing along each row and column. The chrominance channel algorithm in Section II-B requires no multiplies or additions, assuming the indexing is precomputed and, hence, amortized across the pixels of the image.

Table II displays the number of multiplications and additions required for each method. The computational cost of the color transformations includes converting to and from the color space.

The cost of the CIELAB operations was based on the implementation found in [30] where a 1-D LUT was used for the cube root nonlinearity and for the sRGB nonlinearity.[1]

A. Discussion of Results

The smallest ΔE_s errors were achieved by the **LS-LAB** and **LS-sRGB** approaches. These methods were also computationally the most expensive. The commonly used **Bi-sRGB** algorithm created the largest ΔE_s errors but had the lowest computational complexity. Fig. 8 displays a portion of the text image of each approach for a closer comparison. In Fig. 8, the loss of detail (blurring) for the **Bi-sRGB** method is clearly visible.

The methods **LS-Y** and **LS-Y Bi-C** provide a solution between the low complexity of the **Bi-sRGB** and the detail preserving methods of **LS-LAB** and **LS-sRGB**. Fig. 7 displays how the errors are distributed for the two extremes and the **LS-Y** image. The high-frequency nature of the errors is readily apparent.

The optimal scaling algorithm for a particular application depends upon a number of factors. One needs to consider the available computational units (especially in an embedded system), the amount of data that needs to be transformed, and, in the case of an MFP, the page/minute that the device must operate. As shown from the ΔE_s values (where the 300-dpi errors are smaller than the 72-dpi errors), the display and viewing conditions will also play a significant role in the visibility of errors.

IV. Conclusion

In many applications, it is necessary to consider the tradeoffs between accuracy and computational cost. In this paper, we introduced an algorithm for colorimetric image scaling that provides a balance between the low-cost, low-quality, commonly used bilinear interpolation method and the high-cost high-quality LS-algorithms discussed in [1]. The exact algorithm that should be used for a particular application depends greatly upon the system resources, display method, and user expectation, to name a few.

References

[1] M. Unser, A. Aldroubi, and M. Eden, "Enlargement or reduction of digital images with minimum loss of information," *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 247–258, Mar. 1995.

[2] ——, "Fast b-spline transforms for continuous image representation and interpolation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, pp. 277–285, Mar. 1991.

[3] W. K. Pratt, *Digital Image Processing*, 2nd ed. New York: Wiley, 1991.

[4] J. A. Parker, R. V. Kenyon, and D. E. Troxel, "Comparison of interpolation methods for image resampling," *IEEE Trans. Med. Imag.*, vol. MI-2, no. 1, pp. 31–39, Jan. 1983.

[5] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981.

[6] H. S. Hou and H. C. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 6, pp. 508–517, Dec. 1978.

[7] P. W. Wong and C. Herley, "Area-based interpolation for scaling of images from a CCD," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Oct. 1997, pp. 905–908.

[8] T. Blu, P. Thevenaz, and M. Unser, "How a simple shift can significantly improve the performance of linear interpolation," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Jun. 2002, pp. 377–380.

[9] P. Thevenaz, T. Blu, and M. Unser, "Interpolation revisited (medical images application)," *IEEE Trans. Med. Imag.*, vol. 19, no. 7, pp. 739–758, Jul. 2000.

[10] T. Blu, P. Thevenaz, and M. Unser, "Generalized interpolation: higher quality at no additional cost," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Oct. 1999, pp. 667–671.

[11] E. Maeland, "On the comparison of interpolation methods," *IEEE Trans. Med. Imag.*, vol. 7, no. MI-3, pp. 213–217, Sep. 1988.

[12] M. R. Smith and S. T. Nichols, "Efficient algorithms for generating interpolated (zoomed) MR images," *Magn. Reson. Med.*, vol. 7, no. 2, pp. 156–171, Jun. 1988.

[13] N. A. Dodgson, "Quadratic interpolation for image resampling," *IEEE Trans. Image Process.*, vol. 6, no. 9, pp. 1322–1326, Sep. 1997.

[14] A. Schaum, "Theory and design of local interpolators," *CVGIP: Graph. Models Image Process.*, vol. 55, no. 6, pp. 464–481, Nov. 1993.

[15] C. Lee, M. Eden, and M. Unser, "High-quality image resizing using oblique projection operators," *IEEE Trans. Image Process.*, vol. 7, no. 5, pp. 679–692, May 1998.

[16] E. Meijering, "A chronology of interpolation: from ancient astronomy to modern signal and image processing," *Proc. IEEE*, vol. 90, no. 3, pp. 319–342, Mar. 2002.

[17] C. H. Kim, S. M. Seong, J. A. Lee, and L. S. Kim, "Winscale: an image-scaling algorithm using an area pixel model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 6, pp. 549–553, Jun. 2003.

[18] R. R. Schultz and R. L. Stevenson, "A Bayesian approach to image expansion for improved definition," *IEEE Trans. Image Process.*, vol. 3, no. 3, pp. 233–242, Mar. 1994.

[19] M. Unser, "Splines. A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, no. 11, pp. 22–38, Nov. 1999.

[20] X. Zhang and B. A. Wandell, "A spatial extension of CIELAB for digital color image reproduction," in *Proc. SID Symp.*, 1996, pp. 731–734.

[21] G. Qiu and G. Schaefer, "High quality enhancement of low resolution color images," in *Proc. 7th IEE Int. Conf. Image Processing and Its Applications*, Manchester, U.K., 1999, pp. 358–362.

[22] B. E. Schmitz and R. L. Stevenson, "The enhancement of images containing subsampled chrominance information," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 1052–1056, Jul. 1997.

[23] W. B. Pennebacker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand-Reinhold, 1993.

[24] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. Legall, *MPEG Video Compression Standard*. New York: Chapman & Hall, 1997.

[25] L. Lucchese and S. K. Mitra, "A new method for denoising color images," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Sep. 2002, pp. 373–376.

[26] R. Balasubramanian, "Reducing the cost of lookup table based color transformations," in *Proc. IS&T/SID 7th Color Imaging Conf.: Color Science, Systems, and Applications*, Nov. 1999, pp. 65–68.

[27] T. J. Flohr, B. W. Kolpatzik, R. Balasubramanian, D. A. Carrara, C. A. Bouman, and J. P. Allebach, "Model-based color image quantization," in *Proc. SPIE Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, 1993, pp. 270–281.

[28] M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes, "Proposal for a standard default color space for the internet—sRGB," in *Proc. IS&T/SID 4th Color Imaging Conf.: Color Science, Systems, and Applications*, Nov. 1996, pp. 238–246.

[29] M. Unser, "Ten good reasons for using spline wavelets," in *Proc. SPIE, Wavelet Applications in Signal and Image Processing V*, vol. 3169, Jul. 1997, pp. 422–431.

[30] C. Connolly and T. Fliess, "A study of efficiency and accuracy in the transformation from RGB to CIELAB color space," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 1046–1048, Jul. 1997.

[31] G. Sharma, *Digital Color Imaging Handbook*. Boca Raton, FL: CRC, 2003.

[32] ITU-R Recomendation BT.601, Encoding Parameters of Digital Television for Studios.

[1] Another approach would be to use trilinear interpolation and a three-dimensional LUT.

Michael Vrhel (SM'03) received the B.S. degree in electrical engineering (*summa cum laude*) from the Michigan Technological University, Houghton, and the M.S. and Ph.D. degrees in electrical engineering from North Carolina State University, Raleigh, in 1987, 1989, and 1993, respectively. During his Ph.D. studies, he was an Eastman Kodak Fellow.

He is currently a Distinguished Engineer at Conexant Systems, Redmond, WA. From 1993 to 1996, he was a National Research Council Research Associate at the National Institutes of Health (NIH), Bethesda, MD, where he researched biomedical image and signal processing problems. In 1996, he was a Senior Staff Fellow with the Biomedical Engineering and Instrumentation Program, NIH. From 1997 to 2002, he was the Senior Scientist at Color Savvy Systems Limited, Springboro, OH, where he developed color device characterization software and low-cost color measuring instrumentation. From 2002 to 2004, he was the Senior Scientist at ViewAhead Technology, Redmond. He holds two patents at Color Savvy Systems and has several pending. He has published over 40 refereed journal and conference papers.

Dr. Vrhel is a member of SPIE.