Color to Gray and Back: Color Embedding Into Textured Gray Images

Ricardo L. de Queiroz, *Senior Member, IEEE,* and Karen M. Braun

***Abstract*—We have developed a reversible method to convert color graphics and pictures to gray images. The method is based on mapping colors to low-visibility high-frequency textures that are applied onto the gray image. After receiving a monochrome textured image, the decoder can identify the textures and recover the color information. More specifically, the image is textured by carrying a subband (wavelet) transform and replacing bandpass subbands by the chrominance signals. The low-pass subband is the same as that of the luminance signal. The decoder performs a wavelet transform on the received gray image and recovers the chrominance channels. The intent is to print color images with black and white printers and to be able to recover the color information afterwards. Registration problems are discussed and examples are presented.**

***Index Terms*—Color embedding, color-to-gray conversion.**

I. Introduction

COLOR documents are commonplace in contemporary offices and appear in a variety of forms. Documents are frequently prepared, stored, and displayed electronically, but they are also commonly printed and distributed as hardcopies. From brochures to technical papers, printed paper is still an important component of an office.

We are concerned with color documents prepared digitally, which are to be printed using a black-and-white printer or transmitted using a conventional black-and-white fax machine. We, therefore, address the problem of representing color images in black-and-white, while trying to retain the information conveyed in charts and pictures. Graphics, like pie charts, were likely prepared using very contrasting colors to enhance visibility. Once the color graphic is converted to monochrome, sometimes the contrasting colors are mapped to the same gray level and their visual difference vanishes. So, the first problem is how to convert colors to black and white such that different colors would look different on paper too, even if they have the same luminance component.

Beyond the above problem, we devised a color-to-gray mapping that is reversible; that is, given the monochrome image, or black and white printed paper produced with our method, we can recover the original colors. The intent is to print color images using regular (high-volume) black-and-white devices and be able to recover the color image aftterward, if necessary. For that, we use techniques which can be used for watermarking electronic images. Nevertheless, the intent of the paper was to "watermark" printed images in the sense of leaving the color information embedded into the gray image. It is conceivable that someone might apply the ideas here to the electronic transmission of images and in effect the proposed method can be readily applied to embed color into black and white fax transmission. However, this paper is more concerned with imaging than with telecommunications.

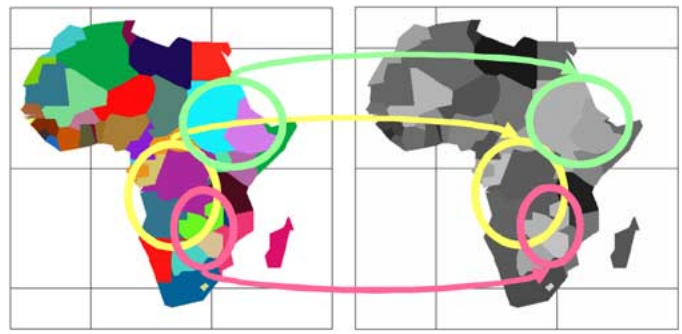

Fig. 1. Mapping from color to monochrome (gray) images.

Textured images in this paper may be displayed incorrectly and show artifacts caused by resampling. This might happen both in print or electronic (PDF) formats. Since these images are crucial to the paper, we suggest looking at the images in http://image.unb.br/queiroz/papers/textured-figures.doc or contacting the authors for access to the original images.

The paper is organized as follows. Section II describes the method to convert color-to-gray pictures by introducing textures. Section III presents the method to recover color information from the textured gray image. Section IV presents some experimental results, and, finally, Section V contains the conclusions of this paper.

II. From Color to Textured Gray

The most trivial way to convert a color image to grayscale for printing is to retain and use the luminance component of the color image. The problem with this approach is that regions that have contrasting colors with similar luminance components would be assigned the same output luminance level and would, therefore, look the same. Fig. 1 shows an example of a colorful map that might have different colors translated into similar shades of gray, thus obfuscating the borders between countries.

Manuscript received February 9, 2005; revised June 6, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Luca Lucchese.

R. L. de Queiroz was with Xerox Corporation, Webster, NY 14580 USA. He is now with the Electrical Engineering Department, Universidade de Brasilia, CP 04591, Brasilia, DF 70910-900, Brazil (e-mail: queiroz@ieee.org).

K. M. Braun is with Xerox Corporation, Webster, NY 14580 USA (e-mail: kbraun@crt.xerox.com).

Digital Object Identifier 10.1109/TIP.2006.871181

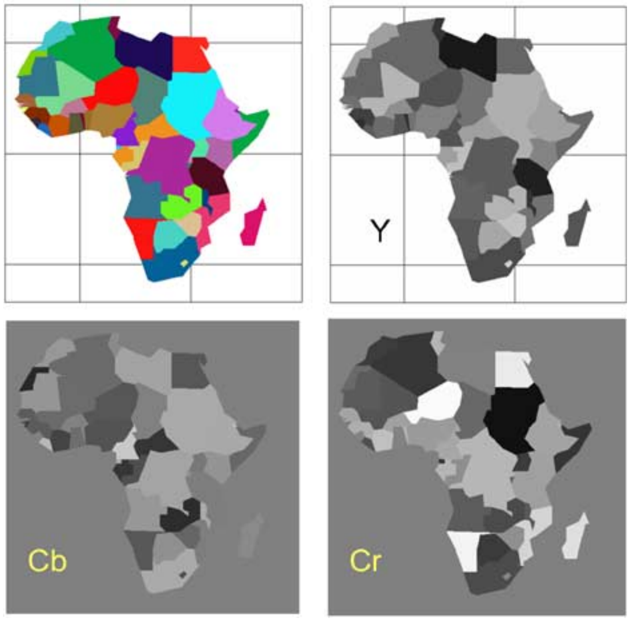

Fig. 2. Color image and its $\mathrm{Y}, \mathrm{Cb}, \mathrm{Cr}$ components.

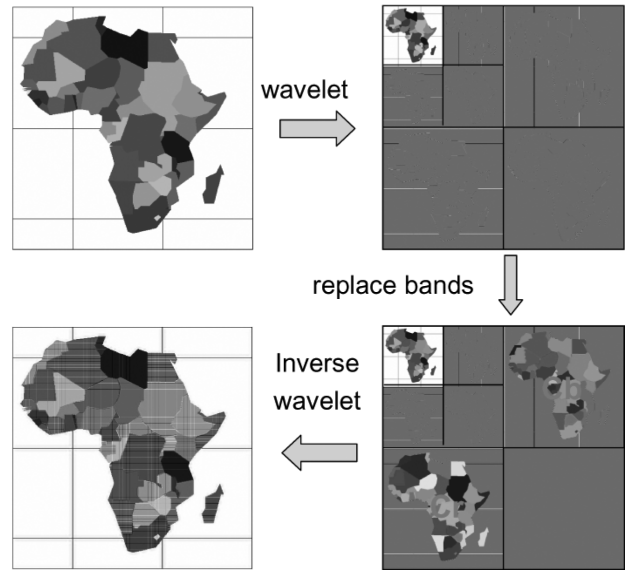

Fig. 3. Mapping from color to monochrome (gray) images. It is possible that the reader might be viewing some large bands in the reconstructed image, which is cause by Moiré, i.e., interference between the texture and viewer resolutions. The bands might change or disappear if the viewing resolution is changed.

An alternative is to compute the colors in the graphic (typically a small number of distinct colors) and to assign different levels of gray to all neighboring colors [1]. This approach may not work for complex graphics.

Another approach is to map colors to textures. One can control halftone dots or patterns as a function of the colors (e.g., as a function of hue and saturation). Hence, regions of different colors with similar luminance will look different after mapping because they would have different textures [2].

Our method maps colors to texture. However, instead of having a dictionary (or palette) of textures and colors, it produces a continuum of textures that naturally switch between patterns without causing visual artifacts. For that, we make use of the discrete wavelet transform [3], or DWT, which decomposes an image into several subbands [3], each representing different spatial frequency contents. Our wavelet-based mapping method works as follows.

1) The color image, assumed to be in RGB color space (the details of the particular RGB primaries are not very important), is transformed into $\mathrm{Y}, \mathrm{Cb}, \mathrm{Cr}$ planes using the popular $\mathrm{RGB}{-}\mathrm{YCbCr}$ linear color transformation, which is also used in CCIR 601, JPEG, JPEG 2000, etc. [4], [5]. Fig. 2 shows a color graphic and its corresponding $\mathrm{Y}, \mathrm{Cb}, \mathrm{Cr}$ planes. A color space like CIELab [6] would work equally well. These are only examples and any luminance-chrominance color space should yield comparable results.
2) Using one level of the DWT [3], the luminance image Y is divided into four subbands: $\mathrm{Y} \rightarrow (\mathrm{Sl}, \mathrm{Sh}, \mathrm{Sv}, \mathrm{Sd})$, corresponding to the low-pass, vertical, horizontal, and diagonal (high-pass in both directions) subbands, respectively. Using decimated filter banks, the dimensions of $\mathrm{Sl}, \mathrm{Sh}, \mathrm{Sv}, \mathrm{Sd}$ are half of those of Y in each direction. Oversampled filter banks and wavelets would yield bands of same dimension as Y and would also work in our context.
3) The planes Cb and Cr are spatially reduced by a factor of 2 in each direction (it has often been shown that somewhat decreasing the spatial detail in the chrominance band of an image does not severely affect its image quality, e.g., JPEG compression).
4) Sh is replaced by Cb and Sv is replaced by Cr.
5) An inverse DWT [3] is carried to recompose the monochrome image as $(\mathrm{Sl}, \mathrm{Cb}, \mathrm{Cr}, \mathrm{Sd}) \rightarrow \mathrm{Y}'$.
6) Image Y' is the resulting gray image and may be printed, which often includes scaling and halftoning.

The process is illustrated in Fig. 3. The idea is that different regions, even presenting the same luminance, might contain different textures and be perceived differently. At least there should be enough differentiation to allow distinction between neighboring regions. In effect, we create artificial high-frequency patterns when we used chrominance data as high-frequency subband samples. These chrominance values may be scaled (multiplied) by a value to increase or decrease the texture "intensity." In our tests, we used unscaled chrominance values and unit-gain DWT filters.

Both the high-pass bands and the chrominance signals adapt well to scene object contours. Hence, the texture pattern changes appear to be natural and are mostly invisible. For graphics, such as in Fig. 3, the best color mapping should produce very visible and distinct textures, which can be created artificially. However, such a method would not work for natural pictures, while the wavelet-based method just described would work for most types of images.

Apart from being based on wavelets, the novelty in this method lies on three other key aspects: 1) the texture differentiation is applied to the gray image and not directly to the halftone image; 2) its smooth and natural color blending is suitable to both graphics and pictures; and, most important, 3) it is reversible, enabling retrieval of the colors from the textured image.

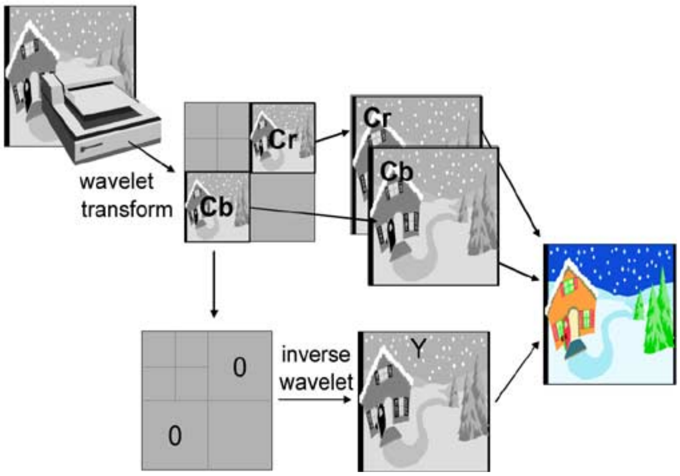

Fig. 4. Retrieving color from gray image. A textured image is read or scanned. After a DWT, the chrominance bands are extracted from the high-pass subbands. These subbands are set to zero. An inverse transform is applied to obtain the Y component, which is then merged with the chrominance ones to recompose the color image.

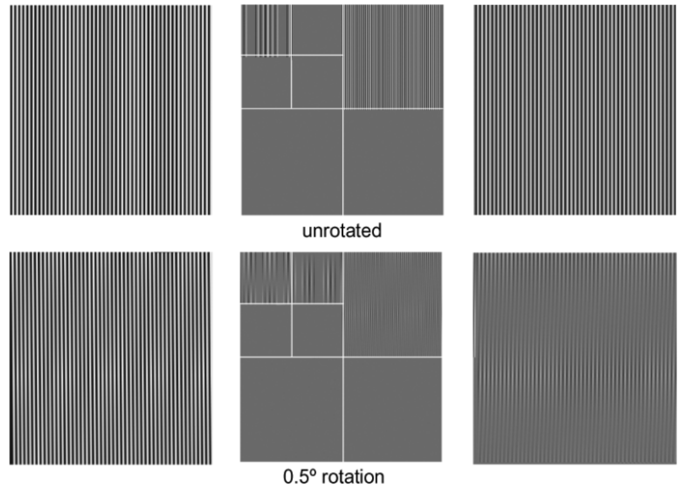

Fig. 5. Effect of even the slightest rotation. (Left) Vertical texture produces virtually no horizontal patterns in (center) wavelet domain. (Right) A zoom of the horizontal subband is shown. After 0.5° rotation, low-frequency vertical patterns appear. Similar results are repeated for the rotated texture in the bottom row.

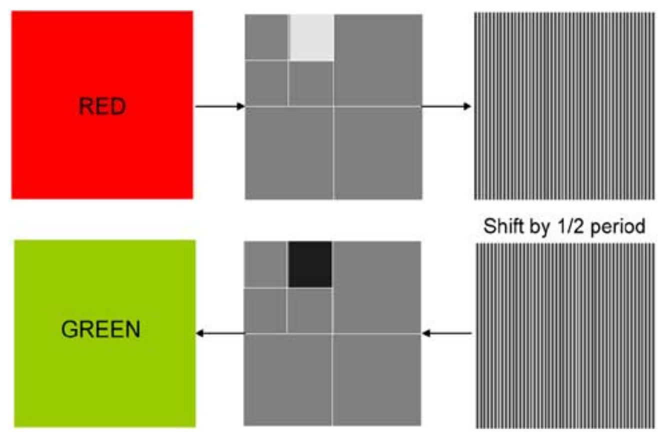

Fig. 6. Opposite colors produce inverted textures. Small shifts in the image can cause the textures to be inverted and lead to large color recovery errors.

III. Recovering Color

One nice feature of the proposed embedding method is the ability to recover the color from the gray (textured) image. For that, we just need to reverse all steps in the color-to-gray mapping as depicted in Fig. 4.

1) The gray textured image is read or scanned.
2) A DWT converts the gray image into subbands.
3) The high-pass horizontal and vertical subbands are interpreted as the Cb and Cr components, which may be interpolated back to the original image size.
4) The high-pass subbands are zeroed (the information originally contained in these bands was lost in the encoding process, which leads so some loss of detail in the output).
5) An inverse transform is applied, producing the Y (gray) component of the image without the texture.
6) The Y, Cb, and Cr components are gathered and used to reconstruct the RGB image using an inverse of the color transform used in the color-embedding process.

The proposed method for embedding and retrieving color into and from a gray image is theoretically sound but faces some practical obstacles. These obstacles arise when one halftones, prints, and scans the gray image.

- *De-screening*. After halftoning and, perhaps, printing and scanning back the image, the decoder (which maps the gray image back to color) needs to filter out the halftone (de-screen the image, or inverse halftone it) and to scale the image to its right size.
- *Warping*. Printing may warp the image, stretching the paper. Scanning may not be properly aligned, causing the recovered gray image to be a warped version of the one before printing. Results can be catastrophic. Fig. 5 depicts the situation where a vertical texture pattern (which should not produce any vertical high frequency coefficients) is rotated by as little as half a degree. Such a small rotation will cause low frequency vertical patterns and distort the horizontal subbands. An example is shown in Fig. 5. This effect is very interesting and some low-frequency vertical patterns occur in the horizontal high-frequency subband! This effect is caused by the wavelet filters whose phase coincidence with the vertical patterns will vary along the image. For example, we do not expect this to happen with shift-invariant wavelet transforms [3].
- *Registration*. The image needs to have perfect scanning registration. Any shift in the texture image may cause major color shifts. See for example Fig. 6. Red and green patches produce sinusoidal textures that are only half a cycle apart from each other. It is easy to mix opposite colors due to minimal registration gaps.
- *Blurring*. The sharp texture we apply is blurred as a result of the printing process which translates to desaturation of the output image. The output image saturation can be boosted to account for this.

To deal with these issues, we have to do the following.

- Scale the image before halftoning and printing enough to ensure the gray texture patterns will survive printing, scanning, and de-screening (inverse halftoning). In our simulations, we typically scale (interpolate) the image by a factor of 4 before halftoning. An image with 512×512 pixels would be printed in less than a 2×2 in^2 area on a 1200-dpi printer. In this paper, we scale up images by simple pixel replication.

- Prewarp and scale back the scanned image before processing. To do that, we detect corners of the scanned image and perform an affine transformation to make the scanned rectangle fit a specified image dimension. That might include scaling down the image, to compensate for the spatial scaling applied before halftoning. The inconvenience is that the decoder must know the image size. This can be solved by only allowing a small number of image dimensions and estimating which image size was used, by measuring the distances of the image corners of the scanned and warped image. Of course, we need to take into account the scaling.
- Most important of all, we changed the way we embed the color information into the subbands. This will be explained next.

In order to get more robust color embedding against decoding opposite colors caused by a small image shift, we divided the chrominance information into four planes. Cb is divided into two planes: $\mathrm{Cb}+$ and $\mathrm{Cb}-$. In $\mathrm{Cb}+$, we reproduce the pixels of Cb which are greater than 0, i.e., $\mathrm{Cb}+ = (\mathrm{Cb} > 0)$. The remaining pixels are set to zero. Same thing happens for $\mathrm{Cb}-$. In $\mathrm{Cb}-$, we reproduce the pixels of Cb which less than 0, i.e., $\mathrm{Cb}- = (\mathrm{Cb} < 0)$. The remaining pixels are set to zero. The same arrangement is made for Cr. Note that $\mathrm{Cb} = (\mathrm{Cb}+) + (\mathrm{Cb}-)$ and $\mathrm{Cr} = (\mathrm{Cr}+) + (\mathrm{Cr}-)$. The reason to create positive- and negative-valued chrominance planes is to avoid completely the color inversion problem depicted in Fig. 6. If a subband is supposed to have only positive values and we obtained negative ones, then it is a sign of texture inversion (in future work, this may help us determine the amount of shifting or misregistration that has occurred). Hence, one can take the absolute value of the subbands and recombine them into Cb and Cr as

$$\mathrm{Cb} = |\mathrm{Cb}+| - |\mathrm{Cb}-| \quad \text{and} \quad \mathrm{Cr} = |\mathrm{Cr}+| - |\mathrm{Cr}-|.$$

As a result, we have four images to embed: $\mathrm{Cb}+, \mathrm{Cb}-, \mathrm{Cr}+$, and $\mathrm{Cr}-$. If we do a two-level DWT [3], the image plane Y is transformed into $\mathrm{Y} \rightarrow (\mathrm{Sl}, \mathrm{Sh1}, \mathrm{Sv1}, \mathrm{Sd1}, \mathrm{Sh2}, \mathrm{Sv2}, \mathrm{Sd2})$, where the level-2 subbands are the higher frequency bands. Then, band replacement occurs as follows:

$$\mathrm{Sd1} \leftarrow \mathrm{Cb}-; \quad \mathrm{Sh2} \leftarrow \mathrm{Cr}+; \quad \mathrm{Sv2} \leftarrow \mathrm{Cb}+; \quad \mathrm{Sd2} \leftarrow \mathrm{Cr}-.$$

Note that $\mathrm{Sd1}$ may have lower resolution than the other three, if we use critically decimated filter banks and wavelets. Thus, one has to reduce one of the chrominance channels, say $\mathrm{Cb}-$, further, i.e., to $1/4$ of the resolution in each dimension, compared to the original Y plane.

The color embedding scheme is illustrated in Fig. 7 while the recovery process is illustrated in Fig. 8. Hence, the color embedding and recovery steps are as follows.

Color Embedding:

1) Convert image from RGB into $\mathrm{Y}, \mathrm{Cb}, \mathrm{Cr}$ (or CIELab).

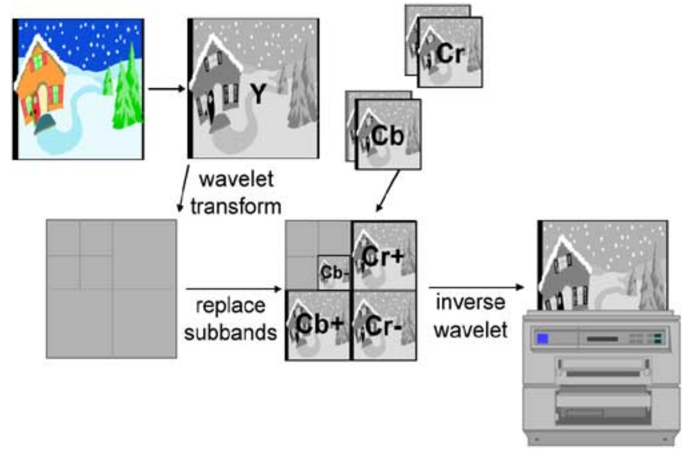

Fig. 7. Mapping from color to monochrome (gray) images through color embedding into textured gray images. We employ subband replacement by absolute-valued chrominance planes.

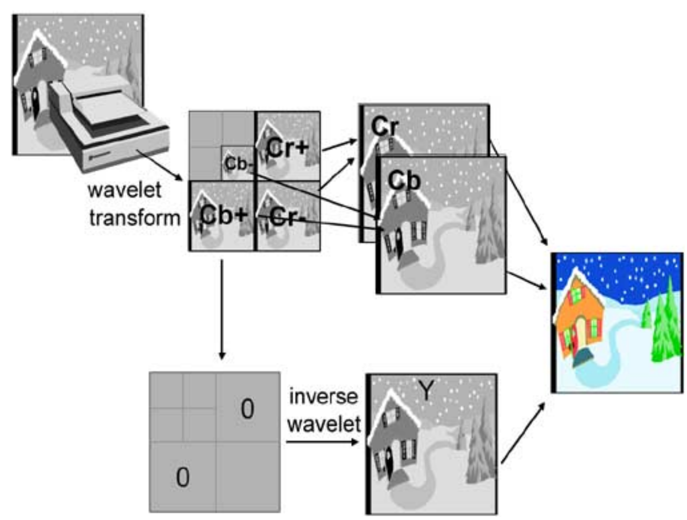

Fig. 8. Recovering color from a textured gray image. The embedded subbands are recovered to form the chrominance planes and zeroed before inverse transform. The YCbCr data is then converted back into RGB.

2) Use a two-level DWT on Y, so that Y is divided into seven subbands: $\mathrm{Y} \rightarrow (\mathrm{Sl}, \mathrm{Sh1}, \mathrm{Sv1}, \mathrm{Sd1}, \mathrm{Sh2}, \mathrm{Sv2}, \mathrm{Sd2})$.
3) Reduce Cb and Cr by $1/2$, construct $\mathrm{Cb}+, \mathrm{Cb}-, \mathrm{Cr}+, \mathrm{Cr}-$, and reduce $\mathrm{Cb}-$ further to $1/4$ of its original size.
4) Replace subbands

$$\mathrm{Sd1} \leftarrow \mathrm{Cb}-; \quad \mathrm{Sh2} \leftarrow \mathrm{Cr}+; \quad \mathrm{Sv2} \leftarrow \mathrm{Cb}+; \quad \mathrm{Sd2} \leftarrow \mathrm{Cr}-.$$

5) Take inverse DWT to obtain the textured gray image, i.e., $(\mathrm{Sl}, \mathrm{Sh}, \mathrm{Sv}, \mathrm{Cb}-, \mathrm{Cr}+, \mathrm{Cb}+, \mathrm{Cr}-) \rightarrow \mathrm{Y}'$.
6) Image Y' is the resulting gray image and may be printed, which often includes scaling and halftoning.

Color Recovery:

1) Read or scan the gray textured image.
2) Determine image dimensions.
3) If necessary, identify corners and carry an affine transform to de-warp the gray image.
4) Reduce image to the correct resolution.

Fig. 9. Two example color images: “wine” (764 × 832 pixels) and “kids” (488 × 340 pixels).

5) Use a DWT to convert the gray image into subbands $\mathrm{Y}' \rightarrow (\mathrm{Sl}, \mathrm{Sh1}, \mathrm{Sv1}, \mathrm{Sd1}, \mathrm{Sh2}, \mathrm{Sv2}, \mathrm{Sd2})$.
6) Interpolate $\mathrm{Sd1}$, doubling its resolution.
7) Make $\mathrm{Cb} = |\mathrm{Sv2}| - |\mathrm{Sd1}|$, and $\mathrm{Cr} = |\mathrm{Sh2}| - |\mathrm{Sd2}|$.
8) Interpolate Cb and Cr, doubling their resolutions.
9) Remove the embedded subbands, i.e., set $\mathrm{Sd1} = \mathrm{Sh2} = \mathrm{Sv2} = \mathrm{Sd2} = 0$, and take the inverse DWT transform to find Y as $(\mathrm{Sl}, \mathrm{Sh1}, \mathrm{Sv1}, 0, 0, 0, 0) \rightarrow \mathrm{Y}$.
10) Convert the $\mathrm{Y}, \mathrm{Cb}, \mathrm{Cr}$ planes back to RGB.

IV. Results

We have tested the algorithm with and without going through the printing and scanning cycle. The great difficulty with printing and scanning is the nonuniform stretching and rotation that might occur to the image after it is printed and then scanned. This is a hard registration problem that is common to many machine reading systems and is beyond the scope of this paper. In some cases, like when sending the halftoned image via standard black and white faxes, registration is not an issue.

Fig. 9 shows two typical color images, picked randomly among the many we tested. The textured images are shown in Fig. 10. The high-frequency textures have low visibility and blend well with the image.

The textured image is often spatially scaled up by an integer factor K (e.g., $K = 4$) in each direction, before printing, to ensure the texture will survive the printing and scanning process.

In the simulation mode, the scaled images are halftoned using standard error diffusion or any other method, then reduced by averaging $K \times K$ binary pixels to recompose a gray image. The recovered “kids” image for different scaling factors is shown in Fig. 11. Note how some colors are de-saturated. Nevertheless the colors seem to have the right hue and, on average, are approximately correct. The larger K, the better the reconstruction. In order to relate the quality of the reconstruction of the color image, we computed the peak signal-to-noise ratio (PSNR) between input and recovered RGB images, for many values of K for image “kids.” The results are presented in Table I. Fig. 12 shows the reconstructed image “wine” using a scaling factor $K = 4$.

The PSNR for various images using $K = 4$ are shown in Table II. For the tests in Table II, the images were picked from

Fig. 10. Textured versions of images “kids” and “wine” along with enlarged portions to better show the texture details.

Fig. 11. Image with color recovery after error diffusion halftoning and scaling. From left to right, $K = 1$, $K = 4$, and $K = 8$.

TABLE I
Peak SNR (in Decibels) Evolution for a Given Scaling Factor K for Image Kids

K	PSNR (dB)
1	13.7
2	22.6
3	25.6
4	27.2
5	27.8
8	28.6
10	28.7

the USC database except for “kids” and “wine” (shown here) and for a color-sweeps chart. The numbers in Table II range from 20 dB up to almost 28 dB. Note that the luminance component of the reconstructed images lost details because of the replacement of the high-frequency subbands. It is important to point that the PSNR numbers in Table II are comparable to the

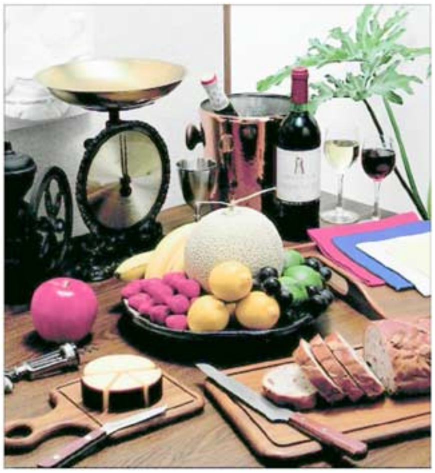

Fig. 12. Image "wine" recovered after haftoning and scaling with $K = 4$.

TABLE II
PEAK SNR (IN DECIBELS) FOR MANY IMAGES AND $K = 4$

Image	PSNR (dB)
Baboon	21.3
House	26.4
Airplane	27.5
Lena	26.7
Sailboat	25.2
Splash	25.1
Tiffany	25.1
Color sweeps	20.3
Kids	27.2
Wine	21.9

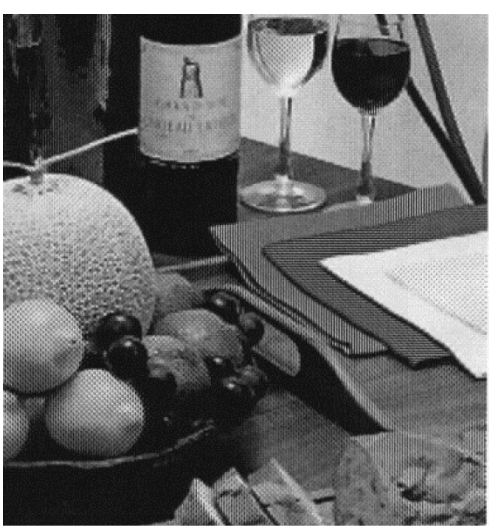

Fig. 13. Zoom of scanned image after color to texture mapping and printing. The image is slightly rotated and warped.

respective PSNR numbers we would obtain if we filter the images with a 3×3 averaging filter (e.g., 20.8 dB for "wine" and 29.4 dB for "kids").

The image wine was printed using a standard 600-dpi laser printer with $K = 4$ and scanned using a 1200-dpi scanner. A portion of this scanned image is shown in Fig. 13. After applying the affine transformations to de-warp and to de-rotate the gray textured image, the resulting image after color recovery is shown in Fig. 14. Note that the colors are a little de-saturated. However, most parts of the image were recovered correctly. This is a test involving real printing and scanning, which causes nonuniform distortions to the image. Taking into account all nonlinear and unpredictable variables derived from the physical processes involved, we consider images like those in Fig. 14 to be excellent results. Furthermore, one can process the recovered image. Since we lost color saturation and details (we removed some high-frequency subbands in the process), Fig. 15 shows the image in Fig. 14 after color enhancement and overall sharpening. This result in Fig. 15 is still a very good reproduction, considering we started from the image in Fig. 13.

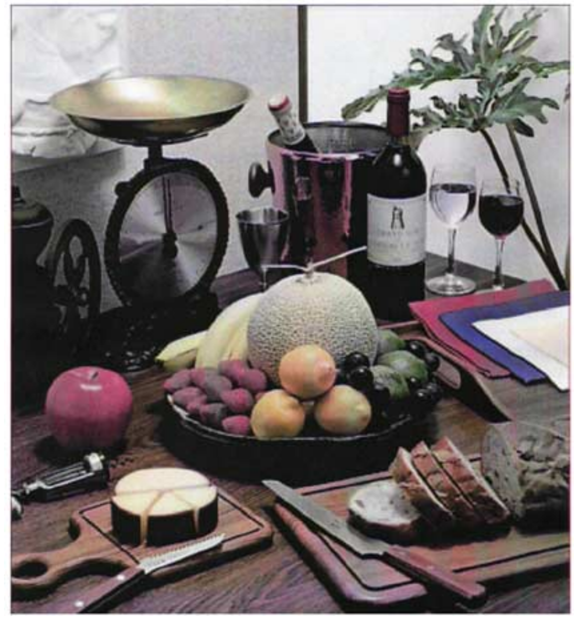

Fig. 14. Color image reconstructed from the textured printed image in Fig. 13.

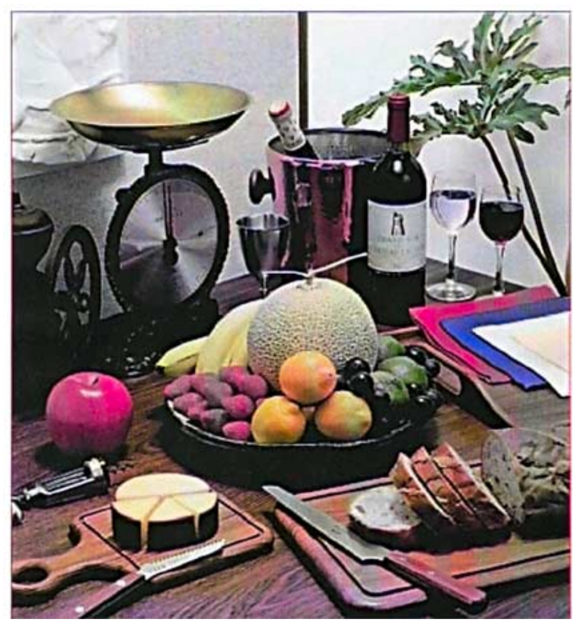

Fig. 15. Reconstructed color image, after sharpening, and color enhancement (saturation).

V. Conclusion

We have presented a method to convert color images to gray that is invertible and allows easy distinction of colors with similar luminance values. The highlight of this paper is the fact that the color-to-gray mapping is invertible by mapping color to textures which can be later decoded and converted back to color. We are unaware of any other attempt to do so. The method is based on wavelet transforms and on replacing subbands by chrominance planes.

The method allows one to send color images through regular black and white fax systems, by embedding the color in a gray image that is binarized. It is also useful for instances where a color printer is not available but the document needs to be printed anyway. One might want to recover the colors later on, from the printed black-and-white hardcopy.

Registration and geometric distortions are still problems. Our next research step is to try shift invariant, complex wavelets, to test whether they would be more robust against geometric image distortions caused by the printing and scanning processes.

References

[1] R. Bala and K. Braun, "Color-to-grayscale conversion to maintain discriminability," *Proc. SPIE*, vol. 5293, pp. 196–202, 2004.

[2] Y. Bai, S. J. Harrington, and J. Taber, "Improved algorithmic mapping of color to texture," *Proc. SPIE*, vol. 4300, pp. 444–451, 2000.

[3] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley, MA: Wellesley-Cambridge, 1996.

[4] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard*. New York: Van Nostrand Reinhold, 1993.

[5] R. L. de Queiroz, *Compression of Color Images, in the Handbook on Transforms and Data Compression*, G. Sharma, Ed. Boca Raton, FL: CRC, 2002.

[6] R. W. G. Hunt, *The Reproduction of Color*. Toolworth, U.K.: Fountain, 2000.

Ricardo L. de Queiroz (S'88–M'94–SM'99) received the Engineer degree from Universidade de Brasilia, Brasilia, Brazil, in 1987, the M.Sc. degree from the Universidade Estadual de Campinas, Campinas, Brazil, in 1990, and the Ph.D. degree from University of Texas at Arlington in 1994, all in electrical engineering.

From 1990 to 1991, he was with the DSP research group at Universidade de Brasilia, as a Research Associate. He joined Xerox Corporation in 1994, where he was a member of the research staff until 2002. From 2000 to 2001, he was also an Adjunct Faculty at the Rochester Institute of Technology, Rochester, NY. He is now with the Electrical Engineering Department, Universidade de Brasilia.

He has published extensively in journals and conferences and has contributed chapters to books, as well. He also holds 36 issued patents. His research interests include multirate signal processing, image and signal compression, and color imaging.

Dr. de Queiroz is a member of the IEEE Signal Processing Society's Image and Multidimensional Signal Processing (IMDSP) Technical Committee. He is an Associate Editor for the IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology, and a past Editor for IEEE Signal Processing Letters. He has been actively involved with the Rochester chapter of the IEEE Signal Processing Society, where he served as Chair and organized the Western New York Image Processing Workshop since its inception until 2001. He was also part of the organizing committee of ICIP 2002. He is a member of the Brazilian Telecommunications Society.

Karen M. Braun received the B.S. degree in physics from Canisius College, Buffalo, NY, in 1991 and the Ph.D. degree in imaging science from the Rochester Institute of Technology (RIT), Rochester, NY, in 1996.

She is with the Xerox Innovation Group at Xerox Corporation, Webster, NY, focusing on color reproduction, gamut mapping, and color perception. She has published numerous journal articles on color appearance modeling and gamut mapping, presented her work at conferences, and coauthored *Recent Progress in Color Science* (IS&T Press, 1997).

Dr. Braun is an active member of IS&T. She has served on the Technical Committee of the IS&T/SID Color Imaging Conference for the six years, and served as Poster Chair for three years. She has been an active member of the ISCC for ten years. She helped create the first student chapter of the ISCC at RIT in March 1993, and served as its first President. She was a Board Member for three years, Associate Editor of ISCC News, a voting delegate for three years, and chairman of the Individual Member Group (IMG).