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

Karen M. Braun, Xerox Corporation, and Ricardo L. de Queiroz, Universidade de Brasilia

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 sub-band (wavelet) transform and replacing band-pass sub-bands by the chrominance signals. The low-pass sub-band 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. Registration problems are discussed and examples are presented.

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.

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 and similar luminance component are assigned the same output luminance level, and therefore look the same.

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. The method works as follows:

1) The color image, assumed to be in some RGB color space, is transformed into Y,Cb,Cr planes using a common RGB-YCbCr linear color transformation, such as those in CCIR 601, JPEG, JPEG 2000, etc.[3-5] Figure 1 shows a color graphic and its corresponding YCbCr planes. A color space like CIELab[6] would work equally well.

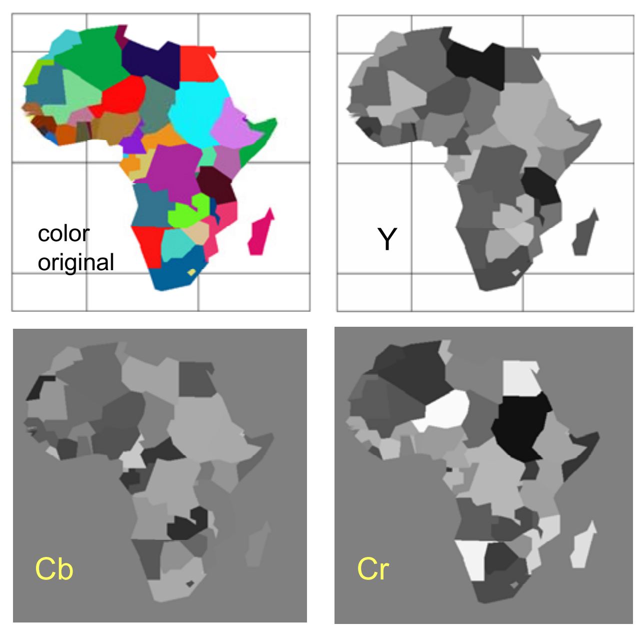


Figure 1. A color image and its Y, Cb , Cr components.

2) Using one level of the wavelet transform,[7] the luminance image Y is divided into 4 sub-bands: $Y \rightarrow (S_l, S_h, S_v, S_d)$, corresponding to the low-pass, vertical, horizontal and diagonal (high-pass in both directions) sub-bands, respectively. Using decimated filter banks, the dimensions of S_l, S_h, S_v, S_d 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) S_h is replaced by Cb and S_v is replaced by Cr.

5) An inverse wavelet (sub-band) transform [7] is carried to recompose the monochrome image as $(S_l, Cb, Cr, S_d) \rightarrow Y'$

6) Image Y' is the resulting gray image and may be printed, which often includes scaling and halftoning.

The process is illustrated in Fig. 2. 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 by "fooling" the wavelet transform to believe that chrominance signals are high-frequency sub-band samples. Both the high-pass bands and chrominance signals adapt well to scene object contours, making the texture pattern changes appear to be natural.

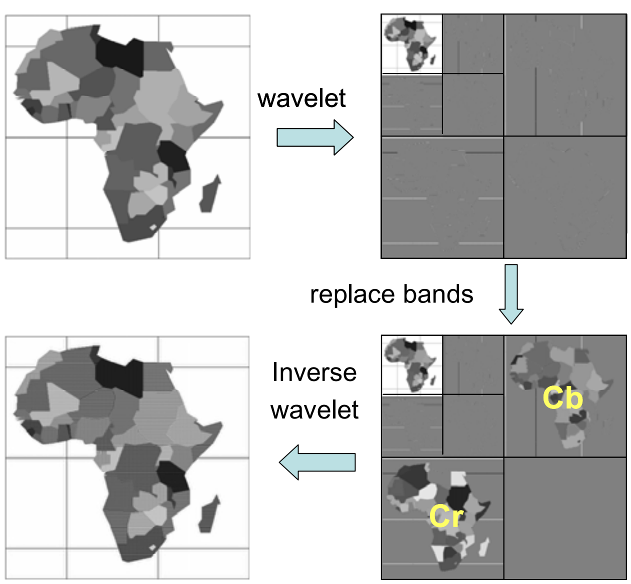


Figure 2. The mapping from color to monochrome (gray) images. It is possible that the reader might be viewing some large bands in the reconstructed image, which is caused by moiré, i.e. interference between the texture and viewer resolutions. The embedded texture is at a much higher frequency.

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

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 reverse the steps in the color-to-gray mapping as depicted in Fig. 3, i.e.:

1) The gray textured image is read or scanned.

2) A wavelet transform converts the gray image into sub-bands.

3) The high-pass horizontal and vertical sub-bands are interpreted as the Cb and Cr components, which may be interpolated back to the original image size.

4) The high-pass sub-bands 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.

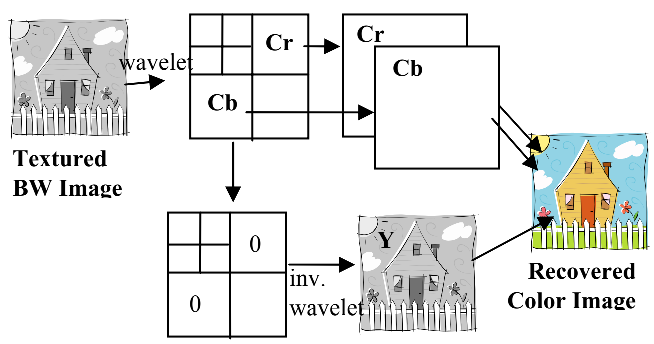


Figure 3. Retrieving color from gray image. A textured image is read or scanned. After a wavelet transform, the chrominance bands are extracted from the high-pass sub-bands. These sub-bands are set to zero. An inverse transform is applied to obtain the Y component, which is then merged with the chrominance ones to recompose the color image.

Improvements to the 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. Figure 4 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 sub-bands. An example is shown in Fig. 4.

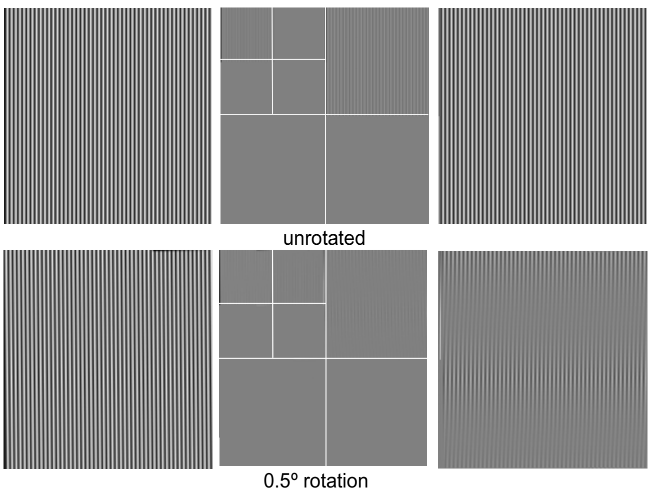


Figure 4. The effect of even the slightest rotation. Vertical texture (left) produces virtually no horizontal patterns in wavelet domain (center). A zoom of the horizontal sub-band is shown (right). After 0.5o rotation, low-frequency vertical patterns appear. Similar results are repeated for the rotated texture in the bottom row.

- Registration. The image needs to have perfect scanning registration. Any shift in the texture image may cause major color shifts. See, for example, Fig. 5. 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:

- 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 the image by a factor of 4 before halftoning. An image with 512 x 512 pixels would be printed in less than a 2x2 in^2 area on a 1200 dpi printer.

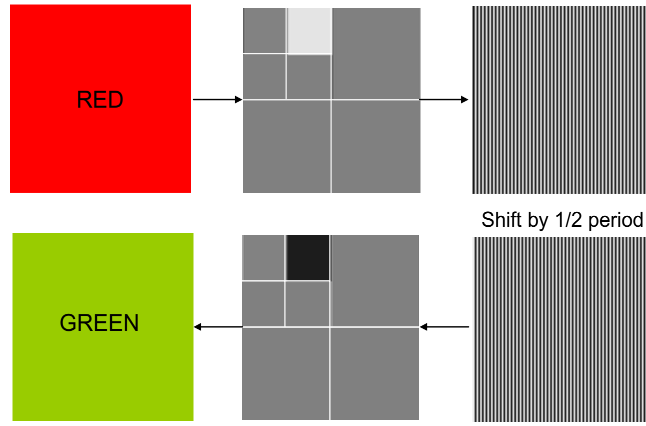


Figure 5. Opposite colors produce inverted textures. Small shifts in the image can cause the textures to be inverted and lead to large color recovery errors.

- Pre-warp 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 sub-bands. 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 4 planes. Cb is divided into two planes: Cb+ and Cb-. In Cb+ we reproduce the pixels of Cb which are greater than 0, i.e. Cb+ = (Cb > 0). The remaining pixels are set to zero. Same thing happens for Cb-. In Cb- we reproduce the pixels of Cb which less than 0, i.e. Cb- = (Cb<0). The remaining pixels are set to zero. The same arrangement is made for Cr. Note that Cb = (Cb+) + (Cb-) and Cr = (Cr+) + (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 sub-band 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 sub-bands and recombine them into Cb and Cr as in Eq. 1.

$$Cb = |Cb+| - |Cb-| \quad \text{and} \quad Cr = |Cr+| - |Cr-| \qquad (1)$$

As a result, we have 4 images to embed: Cb+, Cb-, Cr+, and Cr-. If we do a 2-level wavelet transform, the image plane Y is transformed into Y→ (S_l, S_{h1}, S_{v1}, S_{d1}, S_{h2}, S_{v2}, S_{d2}), where the level-2 sub-bands are the higher-frequency bands. Then, band replacement occurs as in Eq. 2.

$$S_{d1} \leftarrow Cb- \;;\; S_{h2} \leftarrow Cr+ \;;\; S_{v2} \leftarrow Cb+ \;;\; S_{d2} \leftarrow Cr- \qquad (2)$$

Note that S_{d1} 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 Cb-, further, i.e. to ¼ 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:

Color embedding

1) Convert image from RGB into Y,Cb,Cr (or CIELab).

2) Use a two-level wavelet transform on Y , so that Y is divided into 7 sub-bands: Y→ (S_l, S_{h1}, S_{v1}, S_{d1}, S_{h2}, S_{v2}, S_{d2}).

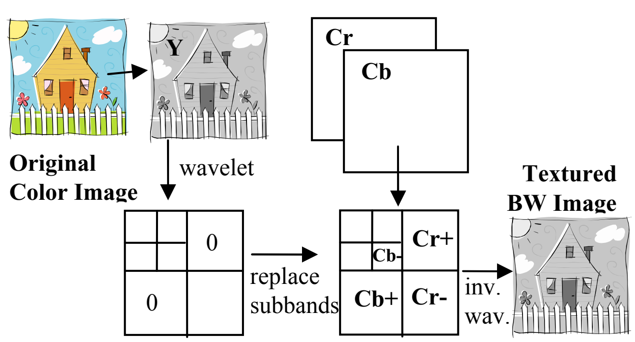
3) Reduce Cb and Cr by ½, construct Cb+, Cb-, Cr+, Cr-, and reduce Cb- further to ¼ of its original size.

4) Replace sub-bands:

$$S_{d1} \leftarrow Cb- \;;\; S_{h2} \leftarrow Cr+ \;;\; S_{v2} \leftarrow Cb+ \;;\; S_{d2} \leftarrow Cr- \qquad (3)$$

5) Take inverse wavelet transform to obtain the textured gray image, i.e. (S_l, S_{h1}, S_{v1}, Cb-, Cr+, Cb+, Cr-) → 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.

5) Use a wavelet transform to convert the gray image into sub-bands Y'→ (Sl, Sh1, Sv1, Sd1, Sh2, Sv2, Sd2).

6) Interpolate Sd1, doubling its resolution.

7) Make Cb = $|S_{v2}|$ - $|S_{d1}|$, and Cr = $|S_{h2}|$ - $|S_{d2}|$.

8) Interpolate Cb and Cr, doubling their resolutions.

9) Remove the embedded sub-bands, i.e. set $S_{d1} = S_{h2} = S_{v2} = S_{d2} = 0$, and take the inverse wavelet transform to find Y as (S_l, S_{h1}, S_{v1}, 0, 0, 0, 0) → Y.

10) Convert the Y,Cb,Cr planes back to RGB.

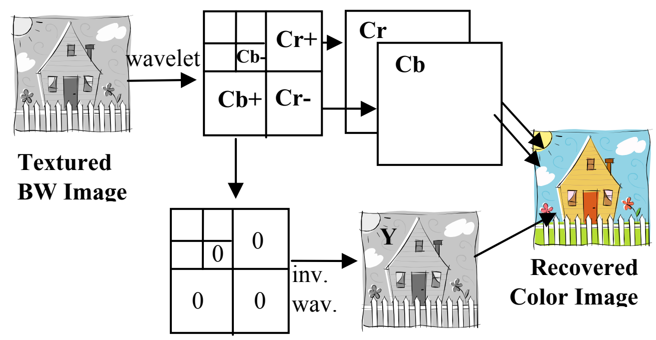


Figure 7. Recovering color from a textured gray image. The embedded sub-bands are recovered to form the chrominance planes and zeroed before inverse transform. The YCbCr data is then converted back into RGB.

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 non-uniform 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. 8 shows a typical color image. The textured images is shown in Fig. 9. 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 image is halftoned using standard error diffusion or any other method, then reduced by averaging KxK binary pixels to recompose a gray image. 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 a particular image (different than shown in Fig. 8). The results are presented in Table I. Figure 10 shows the reconstructed image "wine" using a scaling factor K=4.

Table I. Peak SNR (dB) 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 same image 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. 11. 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. 12. 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 non-uniform distortions to the image. Taking into account all non-linear and unpredictable variables derived from the physical processes involved, we consider images like those in Fig. 12 to be excellent results. Furthermore, one can process the recovered image. Since we lost color saturation and details (we removed some high-frequency sub-bands in the process), Figure 13 shows the image in Fig. 12 after color enhancement and overall sharpening. This result in Fig. 13 is still a very good reproduction, considering we started from the image in Fig. 11.

Conclusions

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 sub-bands 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," SPIE Vol. 5293, pp. 196-202, 2004.
2. Y. Bai, S.J. Harrington, J. Taber, "Improved algorithmic mapping of color to texture," Proc. SPIE Vol. 4300, Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts VI, pp. 444-451, San Jose, US, 2000.
3. W. B. Pennebaker and J. L Mitchell., JPEG: Still Image Compression Standard, Van Nostrand Reinhold, NY, US, 1993.
4. D. S. Taubman, M. W. Marcellin, JPEG2000, Image Compression Fundamentals, Standards and Practice, Norwell, MA, US: Kluwer Academic, 2002.
5. R. L. de Queiroz, Compression of Color Images , in The Handbook on Transforms and Data Compression, edited by G. Sharma, CRC Press, 2002.
6. R. W. G. Hunt, The Reproduction of Color, Fountain Press, Toolworth, England, 2000.
7. G. Strang and T. Nguyen, Wavelets and Filter Banks, Wellesley-Cambridge, Welesley, MA, US, 1996.

Author Biographies

Dr. Karen Braun received her B.S. degree in physics from Canisius College in 1991 and her Ph.D. in imaging science from Rochester Institute of Technology in 1996. Since then, she has worked in the Xerox Innovation Group at Xerox Corporation, focusing on color reproduction, gamut mapping, color perception, and color workflows. Dr. Braun has been an active member of IS&T and ISCC for ten years.

Dr. Ricardo L. de Queiroz received his Engineer degree from Universidade de Brasilia , Brazil, in 1987, his M.Sc. degree from Universidade Estadual de Campinas, Brazil, in 1990, and his Ph.D. degree from University of Texas at Arlington , in 1994, all in Electrical Engineering. He was a researcher at Xerox Corporation from 1994- 2002, and an adjunct faculty member at the Rochester Institute of Technology from 2000-2001. He is now with the Electrical Engineering Department at Universidade de Brasilia. Dr. de Queiroz is a Senior Member of IEEE and a member of IS&T.