# Detecting spatially varying gray component replacement with application in watermarking printed images

**Ricardo L. de Queiroz**
Universidade de Brasília
Departamento de Engenharia Elétrica
CP 4386, Brasília DF 70919-970
Brazil
E-mail: queiroz@ieee.org

**Karen M. Braun**
**Robert P. Loce**
Xerox Corporation
800 Phillips Rd. M/S 128-27E
Webster, New York 14580

**Abstract.** *We present a method to include watermarks in printed images. With accurate printer calibration, in theory, the same color under different gray component replacement (GCR) strategies should look the same, under specific viewing conditions. We spatially vary the GCR along the image in a manner that is not perceptible, and we employ an estimation method to detect such changes. The choice of GCR for a given pixel (or region) comprises an additional information channel that embeds a watermark or hidden information. The challenge is how to detect which GCR was used and that is our focus. For that, we estimate the RGB value of each pixel and the CMYK values intended to be put onto the paper by scanning the printed page. With that information, we can estimate which GCR strategy was used in a given region and retrieve the watermark message. Instead of focusing on a particular watermarking scheme, we are concerned only with the practical aspects of producing a spatially varying GCR and of robustly estimating which GCR strategy was used at a region. Promising GCR detection results are shown to illustrate the method's potential to watermark printed images. © 2005 SPIE and IS&T.* [DOI: 10.1117/1.2001671]

## 1 Introduction

It is desirable to enable data hiding within a digital image, for example, for image security or authentication, to covert communication, to render instructions, and to provide additional useful information. Some existing digital watermarking methods do not survive the printing process. In fact, most known methods are designed for continuous-tone images and are too fragile to be encoded into a printed page and remain retrievable and invisible. Glyphs[1] and other low-frequency methods that can be used in a print setting often introduce undesirable textures or lower the spatial resolution of the image. If one has full control of the halftone, watermarks can sometimes be embedded into the halftone design itself.[2–5] Other methods applicable to printed pages such as image multiplexing[6] and Xerox Glossmarks® (Ref. 7) can also be used. In general, automated watermark retrieval is difficult and the watermarks are not always invisible.

To address the seemingly conflicting requirements of invisibility and retrievability of the watermark, one must take into account physical properties of the printing process. For example, we know that very likely the printing will occur at a high resolution, i.e., 600 pixels per inch (ppi) and beyond. Also, the printed images will be viewed at a comfortable distance from the page. Hence, the density of pixels per degree of subtended viewing angle is actually very high and the visual sensitivity will be very low to pixel-level details.

Another example of an important physical characteristic is the absorption spectra of the colorants. The three fundamental subtractive primaries, cyan (C), magenta (M), and yellow (Y), are typically used as colorants in printing devices.[8–11] All hues can be reproduced with these colorants, however, black colorant is usually also employed for reasons such as extending the dark portion of the color gamut, improving the rendering of neutrals, and reproducing colors with less CMYK colorant to save money and reduce pile height. Hence, a transformation is required to convert from the set of fundamental primaries CMY to the larger set CMYK. The inclusion of the neutral colorant K enables us to substitute in some amount of K for an equivalent darkness neutral mixture of CMY. Thus, for a given color, some K may be added and some CMY subtracted to produce the same perceived color. This colorant substitution method is known as gray component replacement[9] (GCR). Because color is inherently a 3-D entity, this addition of a fourth colorant K allows for some redundancy. As a limiting case, consider that for much of the color gamut, a given color can be created with either a combination of CMY or with K plus two of those colorants, which is known as a 100% GCR strategy. Lesser amounts of gray component may be substituted, creating the potential for multiple GCR strategies. By switching in space between

Paper 04030 received Apr. 6, 2004; revised manuscript received Aug. 31, 2004; accepted for publication Mar. 15, 2005; published online Aug. 1, 2005.
1017-9909/2005/14(3)/033016/9/$22.00 © 2005 SPIE and IS&T.

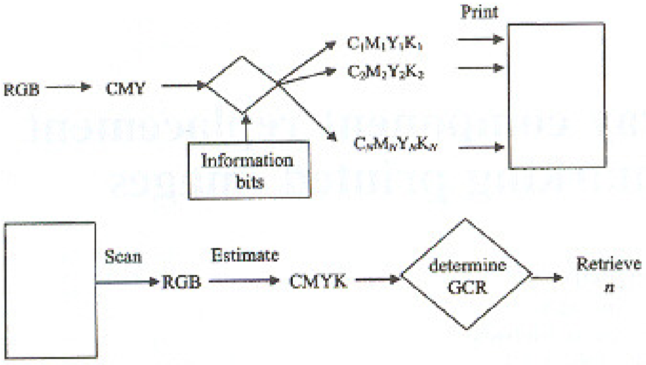


**Fig. 1** Top, inserting the watermark; bottom, extracting the watermark.

multiple GCR strategies, we can produce the same color in a region, while causing a subtle change in the high-resolution dot pattern that produces the color.

Our method embeds hidden data in a manner that is designed for the printing process and does not degrade image texture or resolution. It exploits the properties of a properly designed GCR strategy and applies spatially varying GCR. In Sec. 2, we describe the watermarking scheme. In Sec. 3, we explain the enabling technique of detecting the GCR employed for a given pixel or region. Section 4 describes experiments that validate the method and indicate that the method holds promise for embedding a significant amount of data. Finally, Sec. 5 presents the conclusions of this work.

## 2 Proposed Scheme: Data Hiding via Spatially Varying GCR

We can view the present watermarking problem as the one of embedding hidden information into the printed image. Without loss of generality, assume we have a message composed of $N$-ary symbols to embed and that we have at our disposal $N$ GCR schemes. The pixels are received in a color space such as RGB, but note that other color spaces are also valid input. To enable printing, the pixel representation is converted to a CMY space. As in conventional GCR, the minimum value of the CMY colorant set for a pixel is used to determine K via a GCR curve that relates $\min(C,M,Y)$ to $K$. We have $N$ curves to be chosen for each pixel: $\mathrm{GCR}_1$, or $\mathrm{GCR}_2$, ... , up to $\mathrm{GCR}_N$. Hence, from the same RGB data, two or more possible CMYK sets can be generated, depending on the chosen GCR scheme. If the printing path has been properly calibrated, printed CMYK for each of the GCR curves should look the same, at least under the illuminant for which the calibration was performed. Note that under perfect circumstances, different GCRs produce images that look nearly identical and that appearance is presumably robust against small change of illuminants.

We refer to the method to embed the bits into a printed page as the "encoder" and we refer to the method for retrieving the message out of the printed paper as the "decoder." The encoder algorithm is depicted in the top half of Fig. 1. It works as follows:

1. The input image is composed of pixels in a color space, such as RGB. The pixels are converted to CMY color space.
2. The image is divided into regions, each region composed of a number of pixels.
3. According to the information bits, in each region, we embed one $N$-ary symbol. This is done by associating each state of the $N$-ary symbol to one of the $N$ available GCR schemes.
4. Each pixel is converted to CMYK representation via the chosen GCR for the given region.
5. The image is printed.

The decoder algorithm is depicted in the bottom half of Fig. 1 and can be outlined as follows:

1. The image is scanned and aligned such that the decoder analysis can identify and operate on the encoded regions of the image. The scanning produces RGB values at pixel locations.
2. The image is divided into the same regions as employed by the encoder.
3. For each region, it estimates the GCR by analyzing the scanned RGB data and estimating the CMYK values that occur on the paper. It then recovers the $N$-ary symbol in that region using the estimate of CMYK.
4. By recovering each symbol, the hidden message is retrieved.

This outline of the GCR encoding/decoding method gives rise to several key questions:

1. How can we estimate CMYK values from scanned RGB values? This is the most significant challenge of the proposed watermarking and retrieval method and is the core of this paper. Estimating the CMYK values enables detecting the local GCR curve. We discuss this in Sec. 3.
2. How can we vary the GCR and not cause visible artifacts? Calibration is never perfect and there might be visible differences for some colors, under typical illuminants, when we switch from one GCR to another. We try to calibrate the printing path as well as possible and we suggest that the embedding be applied in a spatially diffuse manner. There are two methods for employing spatial diffusion: make the regions diffuse or make the information diffuse. For example, one can blur the transitions of the embedding message and apply error diffusion to avoid a sharp transition between two different GCRs that might cause perceptible artifacts. This crucial issue must be thoroughly tested.
3. What happens at regions with little or no GCR ($C=0$ or $M=0$ or $Y=0$) or at very dark regions? Dark regions in the image are not useful for embedding information using this technique, since the excessive dot overlap makes it very difficult to detect black pixels. Also, in light regions, we lose the necessary redundancy since no GCR is needed. These regions can be either skipped, or one can just absorb the error into the error correction (EC) mechanisms.
4. What EC mechanism should be employed? Any efficient correction can be used, including block, convo-

lutional, and turbo codes.[12,13] Error correction is further discussed in Sec. 4.

5. How can we perfectly register the image? Registration is crucial to enable reading embedded data at any moderate resolution. For example, we target data embedding at 100 or 200 dpi (before error correction). If we do not accurately register the image, dots will be observed out of position and bit error rates will be large, which would force us to considerably drop the resolution. This is another topic that must be explored in a future study.

Undoubtedly there are many issues to be resolved before implementing a working watermarking scheme. The most crucial one, retrieving CMYK from RGB is discussed in the next section. The other issues will be addressed in future studies.

## 3 GCR Detection

There are important practical obstacles for estimating the GCR strategy, particularly because one must retrieve the CMYK values from scanned RGB data. In general, exact retrieval of CMYK from RGB is not feasible because it is an ill-posed problem. As a result, if one wants to estimate the printed CMYK from scanned RGB, some information in addition to the color values is required. We exploit the nonoverlap property of rotated halftone screens to provide the additional information.

Let the printer-side mapping be: $RGB\text{-}C_nM_nY_nK_n$, where we have the choice of GCR strategies, hence, of CMYK quadruples to pick for a given RGB triple. On the scanning (decoding) side, a scan of the page would produce some colorimetric $R'G'B'$ values. The estimation problem reduces to estimating the value of $n$ for a particular region. In other words, estimate which GCR was used in that region. If the color correction process used by the printer and its printer characterization are sufficiently accurate, we expect the colorimetric RGB values of the printed page, as produced by a low-resolution scanner, to resemble the input $RGB$ values, i.e., $RGB=R'G'B'$. Therefore if we can estimate the actual CMYK amounts put onto the paper we would have the RGB and CMYK values from which to estimate the GCR strategy.

We derive estimates of the actual CMYK data from a high-resolution scan, while the scanner RGB values are found from a low-resolution scan. The nontrivial part is to estimate $K$ from the scan. To simplify, we use RGB values scanned at a resolution higher than the print resolution, where we might be able to discern ink dots and estimate $K$. The process is illustrated in Fig. 2.

There are three different resolutions involved: the printing, scanning and watermarking resolutions. For example, printing might take place at 600 ppi, scanning may be at 1200 ppi, and the watermarking may be embedded and detected at a rate of 120 ppi, i.e., transmission rate is 120 bits per inch (bpi).

### 3.1 *Estimating K from a High-Resolution Scan*

If we assume halftoning with rotated screens, it is likely that all the four CMYK dots do not overlap completely. Actually, the amount of overlap of CMY dots covers a small area percentage at midtones. Assume the printed image has perfect dots and that the scanner can resolve the dots and their overlaps. In this hypothetical case, the scanned RGB values assume only a small number of combinations that depend on the geometry of the dots against the scanner resolution. If each of the scanned high-resolution RGB values for a given pixel is very low, then that pixel possesses a black color. A black-colored pixel is indicative of a $K$ dot. This $K$ assumption is a good approximation because for low values of RGB, there must be a $K$ dot or the overlap of CMY dots. Since the overlap area of CMY dots is small due to the rotated screens, it is more likely the black pixel indicates the presence of a $K$ dot.

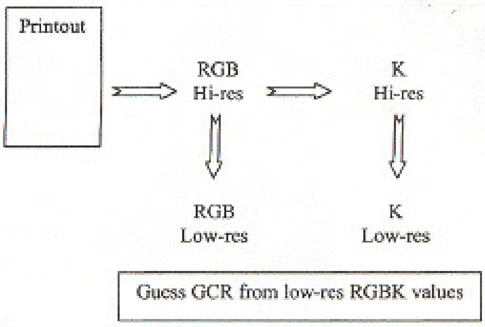


**Fig. 2** GCR estimation steps diagram.

Thus, we estimate the $K$ signal by the following procedure. The scanned RGB values are inverted to obtain CMY estimates, then $K$ is estimated as $K=\min(C,M,Y)=1-\max(R,G,B)$. Note that, except for when there is full CMY overlap or a $K$ dot, at least one of the RGB values will be near maximum (paper) so that the estimate yields $K$. Let $K_h$ be an estimate of the $K$ toner image derived from a high-resolution scan. We assume that information is embedded in the image at a lower spatial resolution than the scanned image. Let $S$ be a suitable operator to reduce the image from the scanner resolution to the watermark resolution. For example, $S$ can be a sequence of blurring filters followed by subsampling. Then $K_h$ can be written as

$$K_h=S[\min(C,M,Y)]=S[1-\max(R,G,B)]. \qquad (1)$$

Even though the formula for $K_h$ resembles a formula for generating $K$ in a 100% GCR strategy, its present use is distinct from that relationship. In fact, its use is limited to high-resolution scanned data where the printer dots can be discerned. We call this method the deterministic approach.

A difficulty with the deterministic approach is that the luminance of a black dot depends on the luminance of the patch it is printed on, as well as the size of the dot itself. A small black dot on a light-colored area has a higher luminance than a large black dot on a darker background. Thus, a more empirical approach is employed, whereby $K$ is estimated by adaptive thresholding. Scanned pixels below a particular luminance level are assumed to be black. The idea is to determine how many pixels in each region correspond to black dots. Thus, we varied the threshold level based on the average luminance of the region. For each average gray value it is desirable to apply a threshold that maximizes the distinction between two GCRs.

The threshold versus luminance relation was obtained by training on a number of scanned patches. Each patch was

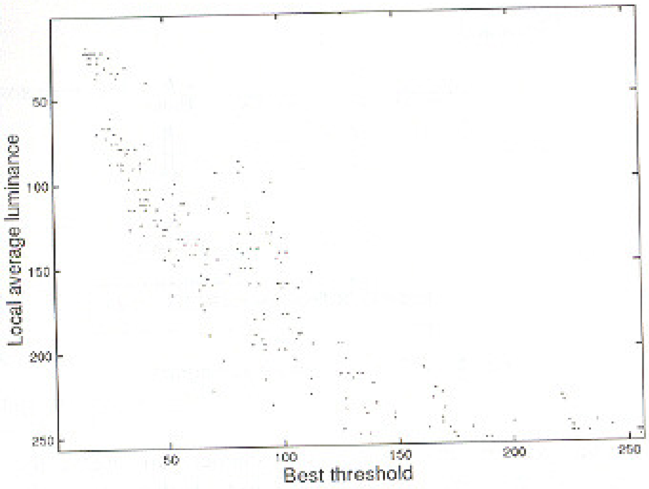


**Fig. 3** Average local luminance value against best threshold.

printed using both GCR$_1$ or GCR$_2$ strategies, by which we denote as $P_n^1$ and $P_n^2$, where $n$ is the patch index or counter. For the $n$'th pair of patches, the average luminance is taken as the average, i.e., $L_n = \text{mean}[(P_n^1 + P_n^2)/2]$. Ideally, the luminance of both patches should be equal. However, practical limitations in calibration and printing, along with the fact that the scanner is not the calibrated observer, might make the luminance of the patches slightly different. We apply a threshold to estimate whether or not the pixel is a $K$ pixel. We test each threshold $\tau$ $(0<\tau<255)$ to evaluate the number of pixels in patches 1 or 2 above the threshold. We want to find the threshold that maximizes the ratio between the estimated $K$ values for each GCR. The reasoning is that the given threshold would make the $K$ values for each GCR to be farther apart, thus facilitating some separation later on. To do that, one can find the threshold $\tau_n$ for the $n$'th patch, such that

$$\tau_n = \max_k[\text{Sum}\,(P_n^1 > k)/\text{Sum}\,(P_n^2 > k), \text{Sum}\,(P_n^2 > k)/ \text{Sum}\,(P_n^1 > k)]. \tag{2}$$

Then, every patch has an $(L_n, \tau_n)$ pair. Figure 3 shows a typical relation between thresholds and average luminance values. We approximated this behavior by a piecewise-linear curve $\tau(L)$ that passes through the following $[L, \tau(L)]$ points: (0,0), (25,20), (100,40), (150,80), (210,125), (255,255). (This is a printer dependent curve.) Using this curve, $K_h$ is estimated as

$$K_h = S\{T[\tau(L), L]\}, \tag{3}$$

where $T(\tau, x)$ is a threshold function such that $T(\tau, x) = 255$ if $x > \tau$ and $T(\tau, x) = 0$ otherwise. In other words, $K_h$ is estimated as

1. Convert the RGB high-resolution data into gray $L$, for example using $L = 0.253R + 0.684G + 0.063B$.
2. Blur the image using a large filter to get the local average luminance values $L'$.
3. For each pixel, given $L'$, look up the associated threshold $\tau$ from the preceding $\tau(L)$ curve. Then, threshold $L$ to obtain $K$ as a binary dot.
4. Reduce the image to any suitable lower resolution. This is the same resolution in which RGB will be analyzed.

### 3.2 *Maximum Likelihood Estimation of the GCR*

Different GCR strategies generate images that may look different except under the illuminant for which the color correction was derived. This is known as illuminant metamerism.[8–11] This effect may be exacerbated by metamerism between humans and typical scanners. That is, two colors (made up of two different CMYK combinations) that match for humans may not match for the scanner because its spectral sensitivity is different from the human visual system. Whereas $K_h$ exploited high resolution and halftone overlap geometry to extract an amount of $K$ toner, we can define $K_l$ as a low-resolution measure of the darkness of a local area and provide a measure of the capacity of that area to possess $K$ toner. In this case, the $K_l$ values are related to the RGB values in an average colorimetric way and purposely do not comprehend microscopic halftone dot geometry. If we assume there is only a high-resolution scanner, the $K_l$ data can be reduced (using $S$) so that

$$K_l = 1 - \max[S(R), S(G), S(B)]. \tag{4}$$

Note the important difference between Eq. (1) and Eq. (4).

One can derive the luminance value from the low-resolution scans as $L = 0.253S(R) + 0.684S(G) + 0.063S(B)$ and we now have two other quantities to normalize the value of $K_h$ against some reference point so that we can estimate the GCR used (hence, the embedded bit data) by analyzing the triplet $\{K_l, K_h, L\}$ somehow. Then, the GCR detection problem becomes similar to the one in classical estimation theory.[14] The $N$-ary symbol is transmitted over a noisy channel. Given the reception of some $K_h$ value, one must estimate which symbol was actually used. Let $P(A|B)$ denote the conditional probability of an event $A$ given $B$ and $E(x|B)$ denote the conditional expectation of a random variable. It is known that for equiprobable sources, the maximum likelihood (ML) estimation is the symbol that maximizes

$$\max_i P(K_h - q_i | K_l, L), \tag{5}$$

where $q_i$ are the values of $K_h$ one would produce in a noiseless environment, i.e., the values obtained if there was perfect estimation of $K_h$. In other words, if $GCR = n$, then $\eta = K_h - q_n$ is the noise input the system. Note that in this simplified model we have not incorporated noise from measuring either $K_l$ or $L$. For noise densities with a strictly decaying spectrum, one wants to choose the closest $q_i$ to the given $K_h$.

Let us define

$$\beta(n, K_l, L) \equiv E(K_h | K_l, L, \text{GCR} = n), \tag{6}$$

then one can show that for a given well-behaved noisy framework, the $q_i$ values are well estimated by

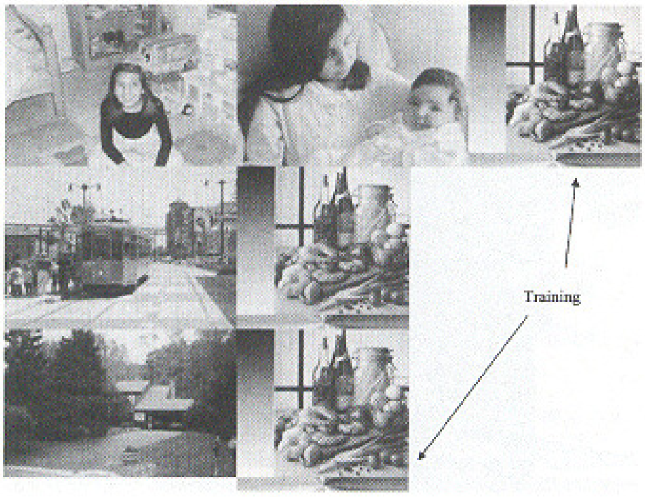


**Fig. 4** Scanned image reduced to 200 dpi.

**Fig. 5** Luminance ($L$) corresponding to Fig. 4.

$$\min_i [K_h - \beta(i,K_l,L)]. \tag{7}$$

Summarizing, the steps for detecting one out of $N$ GCRs are

1. In the training session, find one image or region that was processed with each GCR. For each region or image compute $\beta(n,K_l,L)=E(K_h|K_l,L,\text{GCR}=n)$, i.e., the average value of $K_h$ for each $(K_l,L)$ pair.
2. In the online detection phase, select $\beta$ that is the closest to the received $K_h$.

Note that in the binary case, there are only two values to compute, $\beta(1,K_l,L)$ and $\beta(2,K_l,L)$, so that the algorithm is equivalent to setting up a threshold:

$$\tau(K_l,L)=[\beta(1,K_l,L)+\beta(2,K_l,L)]/2, \tag{8}$$

so that $K_h$ is simply compared to a threshold $\tau(K_l,L)$. If $K_h > \tau$, then we pick GCR$_2$, else we pick GCR$_1$. It is very simple and the array of thresholds for each $K_l$ and $L$, typically of size 256 × 256, can be set up beforehand. Hence, detection can be made with one look up and one comparison.

### 3.3 *Look-up-table-based estimation of the GCR*

To train the system, we create an image (preferably of a test target of patches) with the GCR strategies of interest. For every pixel, a quintuple is organized: $\{R,G,B,K,Q\}$, where $Q$ is a discrete number telling us which GCR was actually used to print that given scanned pixel. We are now left with the task of mapping the RGBK hypercube into one number $Q$.

The training algorithm works by simple majority.

1. Break the RGBK hypercube into $N$ cells.
2. For every pixel in the image,
   Find its RGBK cell.
   Fill in its $Q$ value.
3. At the end, for each cell,
   Compute the histogram of $Q$ values.
   Associate the cell with the most "popular" $Q$ value for that cell.
4. Make a look-up table (LUT) with the RGBK values as input and $Q$ as output

Each cell maps the RGBK value to its most likely GCR strategy. To ensure that no cells are empty, we start by running the preceding training algorithm with a partition of RGBK into 2 × 2 × 2 × 2 cells. If, at the end of training, any of the cells is still empty, assign a random $Q$ value to it. Then we repeat the process for a 4 × 4 × 4 × 4 cell partition. If at the end of the process we obtain any empty cell, make it inherit the $Q$ value from the corresponding position in the 2 × 2 × 2 × 2 partition stage. For nonempty cells, we use the 4 × 4 × 4 × 4 value. We repeat the process for 8 × 8 × 8 × 8. Any empty cells inherit the corresponding $Q$ value from the 4 × 4 × 4 × 4 partition stage. The process is repeated yet one more time to obtain 16 × 16 × 16 × 16 cells, inheriting $Q$ values from the 8 × 8 × 8 × 8 partition stage. We judged the 16 × 16 × 16 × 16 partition might be accurate enough to map RGBK values to a few GCRs, for example, 2, i.e., a $Q=0$ or $Q=1$ decision.

The run-time detection algorithm works as follows:

1. In the high-resolution image, estimate $K$.
2. Reduce the $K$ image to low resolution.
3. Reduce RGB scanner data to low resolution.
4. For every pixel in the low-resolution image,
   Compute the RGBK quadruple.
   Feed RGBK into the cell LUT, and retrieve the $Q$ that is the GCR estimation.

## 4 Evaluating Embedding Potential

We have not yet implemented a real-data embedding system. Future work will deal with registration and with diffuse embedding patterns. We did not switch GCR on a pixel-by-pixel basis but we set up whole images or large regions with each of the GCRs and measured detection er-

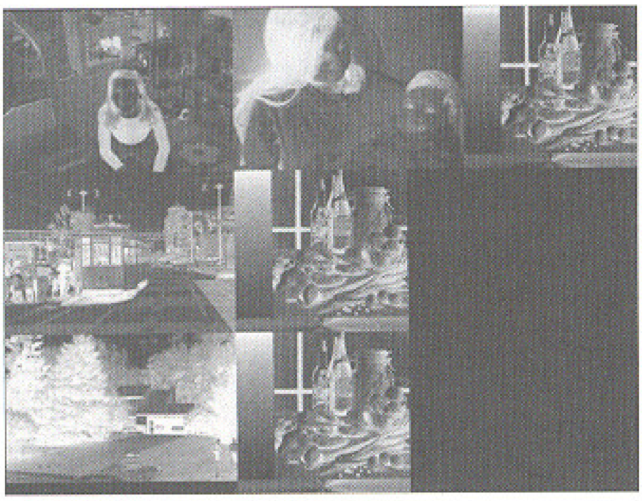


Fig. 6 Value of colorimetric $K$ ($K_l$) obtained from the image in Fig. 4.

ror rates in each region. This gives a very good measure of the method's embedding potential, apart from any registration errors.

### 4.1 Method I—Deterministic Estimation of K Plus ML Estimation of GCR

In an example, we processed several images using two different GCRs. The GCR strategies were devised using color calibration tools tuned for a Xerox DocuColor 2060 printer, which uses a 600-ppi rotated-line screen. Printer calibration was a bit off, so the variation of GCR caused a very small but sometimes noticeable change in appearance. The images were printed at the DC2060 and scanned using a Scitex scanner at true 2000 ppi. The images were reduced ($S$) using running averaging filters and straight subsampling, until reaching a factor of 10:1 in each direction, i.e., 200 ppi. This is the watermark resolution.

The left side of each image was processed using GCR$_1$, while the right side was processed using GCR$_2$, except for the two images that were used for training. One training image was processed using GCR$_1$ in its entirety, while the other was only processed using GCR$_2$. All images (including the training images) were printed on the same page to simplify the experiment, but this certainly was not necessary. The recovered 200-ppi RGB composite image is shown in Fig. 4. The images that were used for training are indicated.

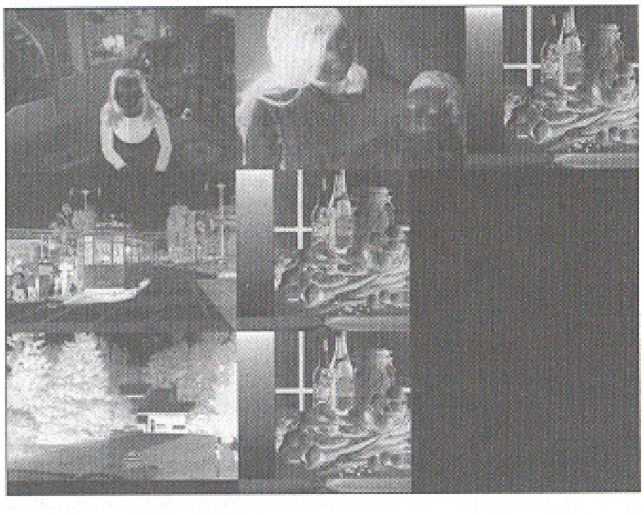


Fig. 7 Estimated $K$ value ($K_h$) from the high-resolution (2000-dpi) scan. Image size reduction to 200 dpi occurs after $K$ is estimated.

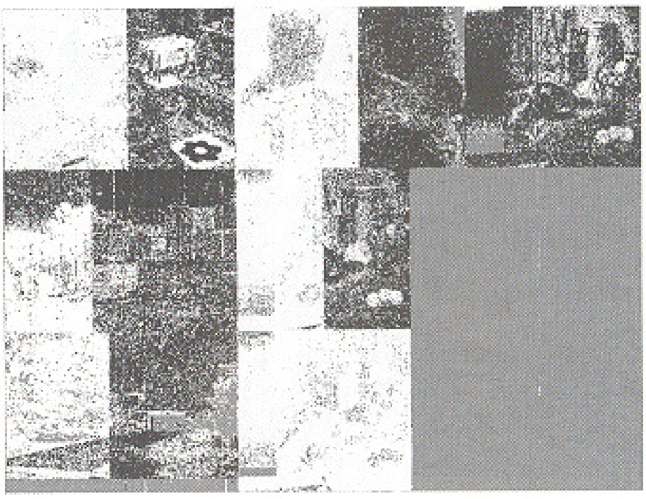


Fig. 8 Detected information. Gray values indicate excluded areas where $L$ was out of bounds 15 to 240. The original information consists of different GCR for the left and right sides of each image. The two training images were fully processed using one GCR, a different GCR each, of course.

Figure 5 shows the corresponding luminance $L$, Fig. 6 shows the $K_l$ values, and Fig. 7 shows $K_h$ for the same image. The detected image is shown in Fig. 8. Since data cannot be effectively embedded in very light or dark areas because they have mostly uniform ink coverage, we decided to exclude these areas from processing. In regions where $L$ was larger than 240 or less than 15, the pixels in Fig. 6 were marked as gray, i.e., not embedded. The bit error rates (BERs) for these example images are shown at Table 1.

### 4.2 Method II—Empirical Estimation of K Plus LUT-Based Estimation of GCR

In this set of tests, we used scans made on a desktop 1200-ppi scanner of prints made on a Xerox Phaser 7700. After reduction of 10:1, the watermark detection (RGBK data) resolution is 120 bpi. Figure 9 depicts a comparison of the amount of $K$ used during printing and the amount of $K$ detected from the scanned data. The same image was printed under two different GCRs and each image was subject to our GCR detection method. Figure 10 shows the

Table 1 BERs for several images and for both GCRs as references, where watermark resolution is 200 bpi.

| Image | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| GCR$_1$ | 0.0300 | 0.2753 | 0.1546 | 0.0867 | 0.0321 |
| GCR$_2$ | 0.3506 | 0.2384 | 0.3212 | 0.1562 | 0.3191 |

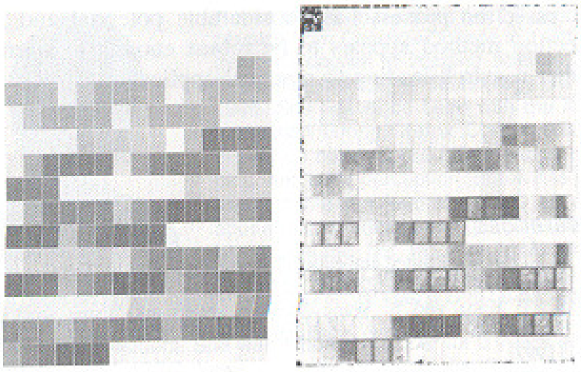


**Fig. 9** Original $K$ values before printing (left) and estimated $K$ values after printing and scanning (right).

result of method I, while Fig. 11 shows the result for method II. The left half of each figure was made with $GCR_1$ and the right half with $GCR_2$. White pixels indicate that $GCR_1$ was detected and black pixels that $GCR_2$ was detected. Thus, a perfect result would be an image with the left half white and the right half black. Comparing Figs. 10 and 11, we can see that the second technique is much less noisy for this class of printers. Method I led to too many noisy patches. A comparison of detection efficiency is shown in Table 2. We then computed the percentage of times there was a correct estimation for each method and GCR. These results do not include the detection errors from the borders between the patches. However, they do include patches where it is impossible to embed information, namely, those where one or more of the pre-GCR $C$, $M$, or $Y$ values is zero.

### 4.3 Channel Rate and Distortion Considerations

The image embedding method can be viewed as a communications channel so that we can borrow information theory results for it. The problem is that we have not yet fully characterized the channel. However, based on the results for both methods we have not seen an average BER (1 to 0 and vice versa) of more than 30%. Please note that these tests were carried for a low-resolution RGBK set at 200 and 120 dpi. The channel is not symmetric, as the results show. It is not stationary either, since contiguous regions typically have similar colors. We can homogenize the error by logically scrambling the pixel locations before embedding, and equalizing (biasing) the information so that the channel should appear to the encoder as a binary symmetric channel (BSC). The BSC is well known in the literature and let us assume the error probability $P_e$ to be 1/3 (1/3 of the bits are wrong in average). It is known that the equiprobable BSC capacity for $P_e=1/3$ is[13] about 0.0817 bits/symbol. At 200 ppi, there are 40,000 pels/in.$^2$, hence the channel capacity would be 3268 bits/in.$^2$. At 100 ppi, assuming the same error rate, the capacity drops to 817 bits/in.$^2$ or near 100 bytes/in.$^2$.

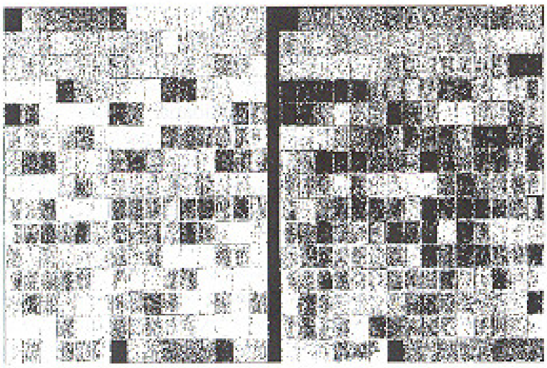


**Fig. 10** Result for patches image using method I. Left image was printed under $GCR_1$ while the one on the right was printed under $GCR_2$. The part on the left should be white and the one on the right should be all black.

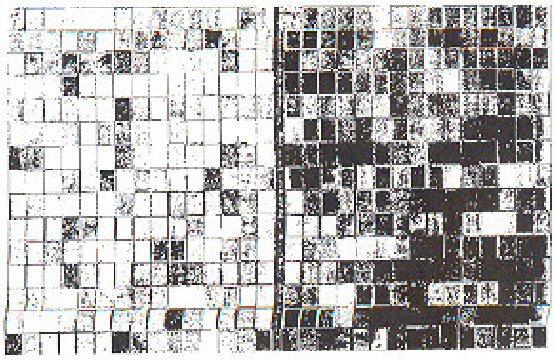


**Fig. 11** Result for patches image using method II; left image printed under $GCR_1$, while the one on the right was printed under $GCR_2$. The part on the left should be white and the one on the right should be all black.

If we construct one of the simplest error correction mechanisms there is, the recursive use of block codes, e.g., the (7,4) Hamming code,[13] we can protect the channel until any prescribed BER. At each step, the embedding rate falls by 4/7, but BER drops by 0.627. The BER versus bit rate curves for both 100 and 200 ppi watermarking RGBK data, assuming the equiprobable BSC with $P_e=1/3$, are shown in Fig. 12. Figure 12 is just an example. Other channel coding schemes will greatly improve the curves in Fig. 12.

The nature of the GCR strategies that are used will increase or decrease the probability of correct detection, depending on how much they differ. For example, excellent detection would be attainable from using 100% GCR and no GCR. However, this would result in more visibility of the embedded signal as a result of metamerism than two GCR strategies that are more similar. The watermark visibility is not computed in any trade-off, so that the best strategy is to make the two GCR strategies not too dissimilar. The inclusion of visibility constraints in the rate-distortion trade-off is left to a future work.

**Table 2** BERs for both methods using patches images from Figs. 9–11, where watermarking detection resolution is 120 bpi.

| Method | $GCR_1$ | $GCR_2$ |
|---|---|---|
| I | 0.28 | 0.47 |
| II | 0.18 | 0.29 |

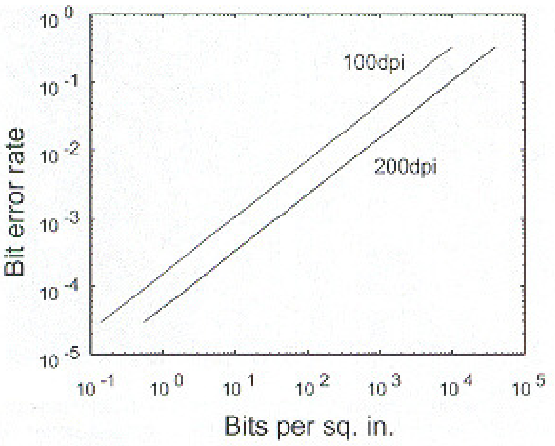


**Fig. 12** Representative rate versus distortion characteristics of the system. The above curves relate the BER and the payload capacity, assuming both 200 and 100 bpi watermarking resolution. We assumed an equiprobable symmetric channel with 1/3 probability of error and channel coding via recursive application of Hamming block codes.

## 5 Final Remarks

In many applications where a watermark is embedded, all the embedded bits are taken together to embed just 1 bit that is used to determine whether or not the watermark is there. The rate distortion analysis of the method assuming a BSC with 1/3 error probability indicates that a reasonable number of bits can be embedded with a reasonable error probability. It is surely sufficient to simply recognize a watermark (the 1-bit embedding) or to carry small payloads. In other words, the method has large bit rate and large BER, so that good error correction capabilities are essential.

Certain colors in an image are not useful for embedding information by varying GCR, namely, those that have no $K$ replacement. For example, CMY values of $(c, m, 0)$ will be converted to $(c, m, 0, 0)$ using either GCR technique. Ideally, these would be eliminated from the encoding/decoding process in some way, or perhaps information could be redundantly embedded such that information erased in one region of the image could be found in another region. At this point, no such scheme has been devised.

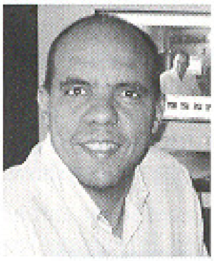
We used training images, but the technique would also work if the nature of the GCR algorithms was known and a fully color-characterized system was used. The training images eliminate the need for such careful calibration.

The detection might be improved by using the spatial orientation of the dots on the page. Black dots are expected to appear at a 45-deg angle, which would differentiate them from other dots. Attempts at using this information have not yet been successful.

Inspection of the results shows the potential of GCR detection in watermarking. A working system is far from being developed. There is much analysis and fine-tuning that can be done. Future work is planned on (1) improving print path calibrations and extending the family of test printers and technologies; (2) including error correcting capabilities and analyzing the channel capacity, i.e., how much payload can we embed after error recovery; (3) optimizing embedding patterns; and (4) reliable registration. In any case, the process has considerable potential and the detection method appears to be robust enough to warrant further research leading to practical applications.

### References

1. D. L. Hecht, "Printed embedded data graphical user interfaces," *IEEE Comput.*, 47–55 (Mar. 2001).
2. K. T. Knox and S. Wang, "Digital watermarks using stochastic screens—a halftoning watermark," *Proc. SPIE* **3022**, 310–316 (1997).
3. S. Wang and K. T. Knox, " in Embedding digital watermarks in halftone screens,"*Security and Watermarking of Multimedia Contents II*, *Proc. SPIE* **3971**, 218–227 (2000).
4. M. S. Fu and O. C. Au, "Data hiding in halftone image by pixel toggling," in*Security and Watermarking of Multimedia Contents II*, *Proc. SPIE* **3971**, 228–236 (2000).
5. Z. Baharav and D. Shaked, "Watermarking of digital halftones," *Electron. Imaging* **3657**, 307–313 (Jan. 1999).
6. G. Sharma, R. P. Loce, S. J. Harrington, and Y. Zhang, "Illuminant multiplexed imaging: GCR and special effects," in *Proc. IS&T/SID 11th Color Imaging Conf.: Color Science, Systems and Applications*, pp. 266–271, Scottsdale, AZ (2003).
7. C. Liu, S. Wang, and B. Xu, "Authenticate your digital prints with glossmark images," in *Proc. NIP20: The 20th Int. Congr. on Digital Printing Technologies*, Salt Lake City, UT (2004).
8. R. W. G. Hunt, *The Reproduction of Color*, Fountain Press, Toolworth, England (2000).
9. R. Bala, "Device characterization," Chap. 5 in *Digital Color Imaging Handbook*, G. Sharma, Ed., CRC Press, Boca Raton, FL (2003).
10. G. Sharma, "Digital color imaging," *IEEE Trans. Image Process.*, **6**, 901–932 (1997).
11. E. Giorgianni and T. Madden, *Digital Color Management*, Addison-Wesley, Reading, MA (1997).
12. J. B. Anderson and S. Mohan, *Source and Channel Coding*, Kluwer Academic Press, Norwell, MA (1991).
13. C. Schlegel and L. Perez, *Trellis and Turbo Coding*, Wiley-IEEE Press, New York (2003).
14. J. M. Wozencraft and I. M. Jacobs, *Principles of Communications Engineering*, Wiley, New York (1965).



**Ricardo L. de Queiroz** received his engineering degree from the Universidade de Brasilia in 1987, his MSc degree from the Universidade Estadual de Campinas, Brazil, in 1990, and his PhD degree from the University of Texas at Arlington in 1994, all in electrical engineering. In 1990 and 1991 he was a research associate with the DSP research group at Universidade de Brasilia. He joined Xerox Corporation in 1994, where he was a member of the research staff until 2002. In 2000 and 2001 he was also an adjunct faculty at the Rochester Institute of Technology. He is currently with the Electrical Engineering Department, Universidade de Brasilia. Dr. de Queiroz has published extensively in journals and conference proceedings and contributed chapters to books. He also holds 35 issued patents, and has many others that are are still pending. He received several scholarships and grants from the Brazilian government and universities. He is an associate editor for the *IEEE Signal Processing Letters* and has been actively involved with the Rochester chapter of the IEEE Signal Processing Society, which he chaired, and he organized the Western New York Image processing Workshop since its inception until 2001. He was also part of the organizing committee of ICIP'2002. His research interests include multirate signal processing, image and signal compression, and color imaging. Dr. de Queiroz is a senior member of IEEE and a member of IS&T.

**Karen M. Braun** received her BS degree in physics from Canisius College in 1991 and her PhD degree in imaging science from Rochester Institute of Technology (RIT) in 1996. She has since been with the Xerox Innovation Group, Xerox Corporation, focusing on color reproduction, gamut mapping, and color perception. Braun has published numerous journal articles on color appearance modeling and gamut mapping, has presented her work at conferences, and has coauthored *Recent Progress in Color Science* in 1997. She has been an active member of ISCC for 10 years. She helped create the first student chapter of the ISCC at the RIT in March 1993, and was its first president. She was a board member for 3 years, associate editor of *ISCC News*, a voting delegate for 3 years, and chaired the Individual Member Group (IMG). Braun is an active member of IS&T. She has served on the Technical Committee of the IS&T/SID Color Imaging Conference for 6 years and was poster chair for 3 years.

**Robert P. Loce** is a principal scientist in the Wilson Center for Research and Technology, Xerox Corporation. He joined Xerox in 1981. While working in optical and imaging technology and research departments at Xerox, he received his BS degree in photographic science from the Rochester Institute of Technology (RIT) in 1985, his MS degree in optical engineering from the University of Rochester (UR) in 1987, and his PhD degree in imaging science from RIT in 1993. His current work involves development of image processing methods for color electronic printing. He is currently an associate editor for *Journal of Electronic Imaging, Real-Time Imaging,* and *IEEE Transactions on Image Processing.* He is a member of SPIE and IEEE.