PAPER

Device Dependent Information Hiding for Images

Hiroshi ITO[†a)] ***and*** **Tadashi KASEZAWA**[††], ***Members***

SUMMARY A new method for hiding information in digital images is proposed. Our method differs from existing techniques in that the information is hidden in a mixture of colors carefully tuned on a specific device according to the device's signal-to-luminance (gamma) characteristics. Because these reproduction characteristics differ in general from device to device and even from model to model, the hidden information appears when the cover image is viewed on a different device, and hence the hiding property is device-dependent. To realize this, we modulated a cover image using two identically-looking checkerboard patterns and switched them locally depending on the hidden information. Reproducing these two patterns equally on a different device is difficult. A possible application of our method would be secure printing where an image is allowed to be viewed only on a screen but a warning message appears when it is printed.

key words: *digital watermarking, covert communication, printing deterrent, information hiding, color reproduction*

1. Introduction

Secrecy has been an important part of human history especially in the premodern age of the 19th and previous centuries when the phrase "knowledge is power" probably had more meaning than it has today [1]. Even in the modern era, a variety of techniques, such as cryptography, secret sharing, and covert channels have been investigated and put into practice to handle many aspects of hiding and sharing secrets [2]–[5]. Information hiding, in a broad sense, is the art of transmitting a message without it being noticed by parties other than the sender and receiver. Invisible ink is a typical example of such communication [6]. It is well known that letters written by lemon juice are invisible until heated allowing it to be used for secret writing. In [7], [8], more functional versions of ink made of specific chemicals were proposed for possible use in secure printing.

As digital content proliferates, hiding information in multimedia data has attracted researchers both in industry and academia in the past few decades. This has resulted in many methods of steganography and digital watermarking. These two categories differ in applications that have different requirements. Steganography [9] is more like invisible ink and closely related to secret communication. The utmost requirement of steganography is that the fact that the content has any secret message must be hidden [10]. Due to this rather simple objective, the secureness of steganography is easier to define, leading to methods with a theoretical basis for their security [11], [12]. Recently many techniques using artificial intelligence have been proposed both in steganography and steganalysis [13]–[15]. In contrast, the primary use of digital watermarking is to trace or verify the origin of information such as copyright holders and distribution routes. Although the watermark must be invisible, the fact that a message has been embedded can be made public in some cases. In general, watermarking has much more variety of applications so its secureness in a strict sense is difficult to define. Hence recent research efforts have been diverted into reversibility [16], [17], encrypted domain embedding [18], and additional functionalities [19], [20].

In this paper, a method for information hiding that is device-dependent is presented. The information is hidden in color mixtures within a cover image with careful tuning of the gamma correction, that is, the reproduction characteristics of the signal. Because the gamma properties are different from device to device, fine-tuned hidden information becomes visible when viewed in a different device. We show that an invisible pattern on a screen becomes visible when printed or even displayed on the screen of devices produced by different manufactures. Possible applications would include deterrence of digital image printings, where viewing an image on screen is allowed but printing is prohibited.

2. Related Work

Steganography modifies a small portion of data to embed information [9]. The most simple method, called LSB embedding, completely replaces the least significant bit plane with the secret message. However, this changes the histogram of the data and is subject to easy attack. A more secure method is to modify each datum by ±1 at most so that an odd or even value carries one bit of information of 0 or 1 accordingly. This basic idea has been the norm in steganography for more than a decade. All efforts have been devoted solely to the selection of information-carrying pixels, that is, which pixels should be modified to make the stego image look as innocuous as possible with other pixels untouched.

Digital watermarking, closely related to steganography, also embeds information by subtly changing the signal level [21]. An early example in [22] discloses a method where two small areas in an image are chosen and biased by a small amount in opposite directions so that one area is

Manuscript received June 21, 2022.
Manuscript revised August 31, 2022.
Manuscript publicized November 8, 2022.

[†]The author is with College of Industrial Technology, Nihon University, Narashino-shi, 275–8575 Japan.
[††]The author is with College of Engineering, Nihon University, Koriyama-shi, 963–8642 Japan.
a) E-mail: ito.hiroshi@nihon-u.ac.jp

DOI: 10.1587/transinf.2022EDP7109

brightened while the other is darkened. The bias is too small to be noticed by human eyes but the direction can be retrieved by inspecting the average signal level in the selected areas. This method is generalized in the currently prevalent spread-spectrum watermarking proposed in [23], where a minute frequency-spread signal is added to the cover signal. As these examples indicate, often in watermarking, a bit of information is carried by a large number of pixels to allow the information to survive various signal processing methods. Usually the invisible signal change is retrieved by correlation using a matched filter.

Some anti-copy techniques [24]–[28] have tried to solve a problem similar to ours. Their primary objective was to discourage unwanted copying of important documents, such as contracts, certificates and licenses. This is accomplished by printing a message or an image on a document with two different patterns. These patterns look the same to the human eye but have properties that are reproduced differently by scanning and printing processes. Hence the invisible message appears on a copied document. The patterns consist of elements such as dots, lines, characters, symbols and many other shapes that are small enough to give a uniform appearance of tones to the eye. Camouflage patterns can be superimposed for further invisibility of the hidden message [25]. Glyphs can be used as elements to carry machine-readable information for improved security [27].

In [29], one of the authors of the current paper proposed a method to hide information in a mixture of colors for four-color displays, which have additional luminance dots of yellow in addition to the primary RGB dots. This was an early attempt to use color combinations for information hiding, where an additional yellow channel was used for carrying augmented information. The signal in the yellow channel can be diffused into the red and green channels to remain invisible to human vision. The level of diffusion is modulated by the hidden information. Because the spectra of light emitted from the yellow, red and green dots are different, the secret information is revealed using a color filter that extracts the yellow channel exclusively.

In general, information hiding is a two-step technology. First we need to construct a method to hide information within a cover signal. Second we need to construct a method to retrieve the information from the stego signal. Naturally, these two steps must be designed in tandem for the specific application in mind because each application has different requirements. In the case of steganography and watermarking, the information is hidden in a small change in the signal and is retrieved by specific signal processing methods. In anti-copy techniques, switching two different patterns embeds information that is retrieved by the process in the copier. In [29], information is embedded in the yellow channel and retrieved by an optical filter. This paper proposes a scheme with two steps that, to our knowledge, have not been previously combined in the literature, and they form a new method for information hiding.

3. Method

3.1 Principle

Here we employ the basic idea from [29] with a new perspective. The essence is that a binary watermark is embedded into a cover image by switching two types of checkerboard patterns depending on whether the pixel is black or white on the watermark image. We assume that the watermark and the cover image are the same size. An elementary unit of the checkerboard patterns is a 2×2 block with alternating color pixels.

A diagram of the watermarking process is depicted in Fig. 1. First, in the encoding step, all the pixels in the cover image are replaced by a pattern that keeps the original shades of gray unchanged. Next, in the embedding step, some pixels are replaced depending on the watermark image, with another pattern that look the same perceptually. When output, a stego image looks like the host image but is watermarked with two different types of patterns. Each step is described in more detail in the following sections.

Figure 2 shows the two blocks of checkerboard patterns, composed of a diagonally aligned pair of colors, (a, b) and (c, d). They are chosen so as to be dissolved as the same color when they are mixed, and hence they are exchangeable. Their signal levels are also modulated by the local brightness and colors of the cover image to maintain its original appearance. Note that the colors in the blocks must be carefully chosen so that they are completely identical. Otherwise the watermark will not be hidden in the stego image. Because the signal values can be handled directly, we need precise knowledge on the relationship between the signal level and the brightness for the intended device to achieve this.

3.2 Encoding

We first describe our method for grayscale images and then extend it to color images later in Sect. 3.5. Let $x(i, j) \in [0 : 255]$ represent the signal values for a cover image at position (i, j) ($i \in [0 : M-1]$, $j \in [0 : N-1]$), quantized with 8 bits. The constants M and N are the image width and height in pixels respectively. In the encoding step, we map $x(i, j)$ to $u(i, j)$ using the (a, b) block shown in Fig. 2(a), by

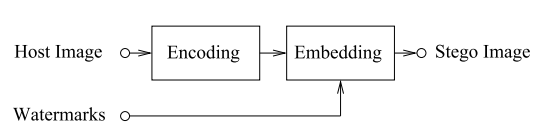


Fig. 1 Diagram of proposed method

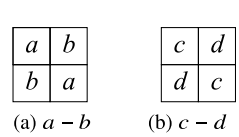


Fig. 2 Block pairs used for watermarking

$$u(i,j) = \begin{cases} a & \text{if } i+j \text{ is even} \\ b & \text{otherwise} \end{cases} \tag{1}$$

The values of a and b are determined so that the brightness of $x(i, j)$ is maintained when they are mixed. This is achieved in the following way. Consider a uniform gray image with a signal value of x. Let $f(x) \in [0 : 255]$ be a function to convert x to its brightness level on the device with $f(0) = 0$ and $f(255) = 255$. When $f(x(i, j)) < 128$, we set

$$a = 0 \tag{2}$$
$$b = f^{-1}\left(2f(x(i,j))\right), \tag{3}$$

where $f^{-1}(\cdot)$ is the inverse function of $f(\cdot)$. Figure 3(a) shows how this works. In the figure, the original signal x is at point P, and its brightness is doubled at point B, which represents the brighter dot of the block. The darker dot of the block is at the origin, marked with A. Note that the middle point Q of segment AB has the same brightness as P indicating that the brightness of the block is the same as the original gray signal. The signal values at point A and B are given by Eqs. (2) and (3).

Similarly for $f(x(i, j)) \geq 128$, as shown in Fig. 3(b), we set

$$a = f^{-1}(2f(x(i,j)) - 255) \tag{4}$$
$$b = 255 \tag{5}$$

to make the brightness of the block equal to that of the uniform gray image again.

3.3 Embedding

Next we introduce another block shown in Fig. 2(b). This is derived from the (a, b) block by increasing the brightness of the darker dots by dy and decreasing the brighter dots by the same amount. These brightness changes should be cancelled out when they are mixed and hence the two blocks are indistinguishable. To do this we set the signal levels, when $f(x) < 128$, as

$$c = f^{-1}(dy) \tag{6}$$
$$d = f^{-1}(2f(x(i,j)) - dy), \tag{7}$$

and when $f(x) \geq 128$, as

$$c = f^{-1}(2f(x(i,j)) - 255 + dy) \tag{8}$$
$$d = f^{-1}(255 - dy). \tag{9}$$

To see how this works, consult Fig. 3 again. In both Figs. 3(a) and (b), the signal levels (c, d) correspond to points C and D respectively. The brightness of the middle point R is the same as that of Q, which means that the human eyes can not discern the two blocks when the pixel luminance is mixed.

We modulate the encoded image with a watermark by switching the two above blocks on a pixel basis. Let $y(i, j)$ represent the grayscale stego image and let $w(i, j) \in \{0, 1\}$ be the watermark image. If $w(i, j) = 1$, we set

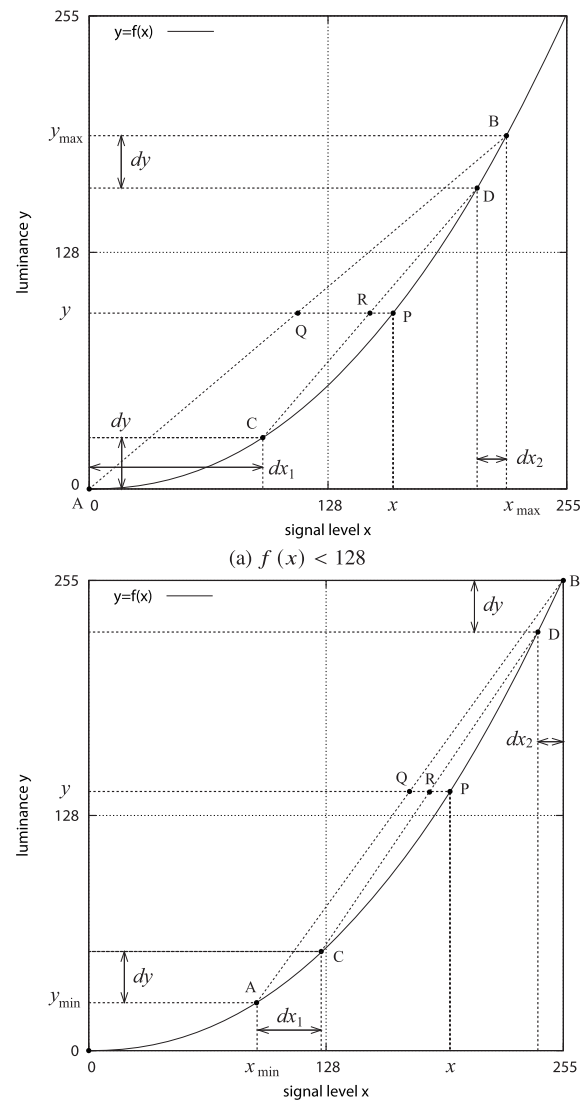


Fig. 3 Setting pixel values for blocks (solid line is a reproduction with $\gamma = 2.2$)

$$y(i,j) = \begin{cases} c & \text{if } i+j \text{ is even} \\ d & \text{otherwise} \end{cases} \tag{10}$$

In other cases we leave it unchanged after encoding, that is,

$$y(i,j) = u(i,j). \tag{11}$$

Note that in the encoding and embedding, we have replaced the pixels in the cover image with two patterns and switched them depending on the watermark while keeping the original gray shade untouched.

The value of dy determines the depth of modulation. We introduce the watermark strength $r \in [0 : 1]$, defined by

$$r = \frac{dy}{f(b) - f(a)}. \tag{12}$$

as the ratio of dy to the maximum possible depth, $f(b) - f(a)$. When $r = 1$, the dark and bright dots are completely exchanged, namely, $c = b$ and $d = a$. We obtain $c = d$ and

therefore a uniform grayscale block for $r = 0.5$.

3.4 Recovery

A mismatch in the gamma characteristics makes the watermark appear. This is because the average brightness is different for (a, b) and (c, d) in general if $f(x)$ is replaced with another function. Assume that an image has been watermarked using $f(x)$ as a gamma function but is displayed on a device with a different function $g(x)$. Consider the case where $f(x) < 128$. From Eqs. (6) and (7), we see that the brightness of the (a, b) block is given by

$$y_{ab} = \frac{g \circ f^{-1}(2f(x))}{2}, \tag{13}$$

where x is the host image's signal value. Here we used $g(0) = 0$. For the (c, d) block we have

$$y_{cd} = \frac{g \circ f^{-1}(dy) + g \circ f^{-1}(2f(x) - dy)}{2}. \tag{14}$$

If the function $g \circ f^{-1}(x)$ is linear, we obtain

$$\begin{aligned} & g \circ f^{-1}(dy) + g \circ f^{-1}(2f(x) - dy) \\ &= g \circ f^{-1}(dy + 2f(x) - dy) \\ &= g \circ f^{-1}(2f(x)), \end{aligned} \tag{15}$$

and hence $y_{ab} = y_{cd}$. However, $g \circ f^{-1}(x)$ is non-linear in general if $g(x)$ and $f(x)$ are different. For example, when $f(x)$ and $g(x)$ are given in the form

$$f(x) = 255\left(\frac{x}{255}\right)^{\gamma} \tag{16}$$

with $\gamma = \gamma_1$ and γ_2 respectively, we obtain

$$g \circ f^{-1}(x) = 255\left(\frac{x}{255}\right)^{\gamma_2/\gamma_1}, \tag{17}$$

which is non-linear unless $\gamma_1 = \gamma_2$. This causes $y_{ab} \neq y_{cd}$ in general. The same discussion follows for the case that $f(x) \geq 128$.

3.5 Extension to Color Images

Extension of our method to color images is straightforward. We first consider a case where the stego image is in color, but the cover image remains in grayscale, and then a case where both the stego and the cover images are in color.

First, when the stego image has three color channels, we can modulate one or more of them with the watermark image by switching two patterns. Let $\mathbf{y}(i, j) = (y_0(i, j), y_1(i, j), y_2(i, j))$ be the stego image composed of RGB components. In the encoding, we simply replace all the channels $k = 0, 1, 2$ in the same way using

$$u_k(i, j) = \begin{cases} a & \text{if } i + j \text{ is even} \\ b & \text{otherwise,} \end{cases} \tag{18}$$

where a and b are determined as described in Sect. 3.2. Assuming that the k-th channel is to be modulated, if $w(i, j) = 1$, we set

$$y_k(i, j) = \begin{cases} c & \text{if } i + j \text{ is even} \\ d & \text{otherwise.} \end{cases} \tag{19}$$

For all other cases, we leave it unchanged after encoding, that is,

$$y_k(i, j) = u_k(i, j). \tag{20}$$

The (c, d) block is used only for the k-th channel where $w(i, j) = 1$.

Second, when the cover image is in color, we can just apply the encoding process for each channel using Eq. (18) with independent values of (a, b). This conserves the color and luminance because the brightness of all channels remains unchanged. Embedding can be done in the same way as described above.

We show some examples of blocks in Fig. 4. As the number of channels grows, a variety of block combinations become possible.

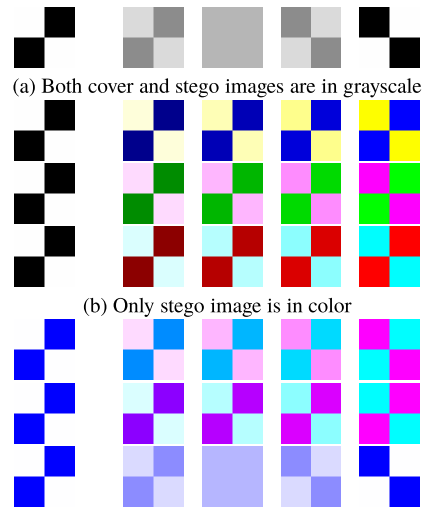

(a) Both cover and stego images are in grayscale
(b) Only stego image is in color
(c) Both cover and stego images are in color

Fig. 4 Examples of blocks. Left most blocks are (a, b) blocks. The subsequent blocks from left to right are (c, d) blocks with $r = 0.25, 0.50, 0.75$, and 1.00 respectively. In (b), the watermark is embedded in the B, G, and R channels from top to bottom. In (c), the watermark is embedded in the G, R, and yellow (G+R) channels from top to bottom for light blue cover.

4. Generating Gamma Tables

Although Eq. (16) is widely accepted for general monitors with a constant γ of around 2.2, this approximation is too coarse to hide watermarks. In this section we present a method of more precise estimation of the signal–brightness relationship for a given monitor.

Assume that two signal values x_1 and x_2 are given and that their corresponding brightness values $f(x_1)$ and $f(x_2)$ are known. Then, a block with $a = x_1$ and $b = x_2$ would have its brightness of $(f(x_1) + f(x_2))/2$. Next, we generate a uniform gray window on screen and adjust the signal level so that the brightness looks the same as that of the block with (x_1, x_2). If x_3 is the signal level of the gray-scale window at

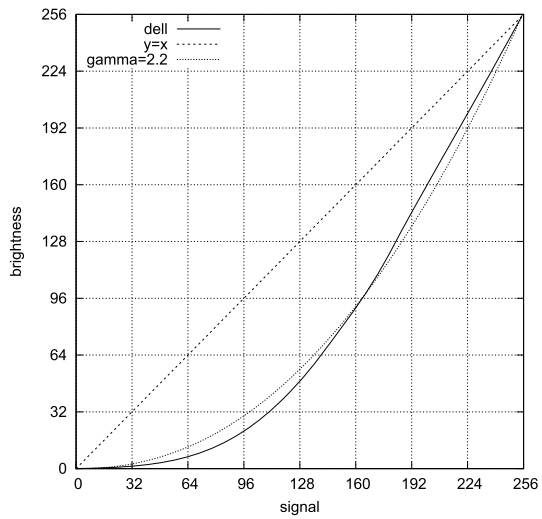


Fig. 5 Gamma characteristics of a Dell U2412M monitor

this time, then it follows that

$$f(x_3) = \frac{f(x_1) + f(x_2)}{2} \tag{21}$$

Hence, we can fix a point on $f(x)$ for $x = x_3$ at $(x_3, f(x_3))$ because the right–hand side of Eq. (21) is known. Then, we can use this point to find a new one by generating a block (x_1, x_3) or (x_3, x_2) and continue to search for a signal level that produces the same brightness. Starting with the two fixed points at $(0, 0)$ and $(255, 255)$ and repeating the above process, we can obtain $f(x)$ for every value of x. We summarize the above procedure as follows:

1. Let $F_0 \leftarrow \{0\}$, $F_{255} \leftarrow \{255\}$, and $F_i = \phi$ for $i = 1, 2, \cdots, 254$. Let $S \leftarrow \{0, 255\}$.
2. Pick $i, j \in S$ and compute the average brightness as
$$y_l = \frac{1}{|F_l|} \sum_{y \in F_l} y \tag{22}$$
for $l = i, j$.
3. Generate block (i, j) on the screen. Obtain a uniform gray level k that looks comparable in brightness to the pattern.
4. Update $F_k \leftarrow F_k \cup \{(y_i + y_j)/2\}$ and $S \leftarrow S \cup \{k\}$.
5. If we have enough samples, stop. Otherwise return to step 2.

We obtain the final estimate of $f(x)$ first by taking an average using $f(x) = \frac{1}{|F_x|} \sum_{y \in F_x} y$, and then by smoothing it with a moving average low–pass filter. Figure 5 shows the function for a Dell U2412M monitor obtained like this. We see that $f(x)$ cannot be precisely represented by Eq. (16) for any value of γ although $\gamma = 2.2$ produces a good approximation.

5. Consideration of Improvements

5.1 Border Visibility

So far, the cover image has been replaced either by the (a, b) or (c, d) block completely according to the binary watermark $w(i, j)$. However, this sometimes causes the edges of the watermark to be visible in the stego image especially when the watermark is embedded in the green channel.

Fig. 6 Effect of border smoothing: Left, No Gaussian filter, Right, Blurred with $\sigma = 3$, embedded in RGB channels with maximum strength and screen of Dell U2412M monitor taken by the Nikon D60 camera

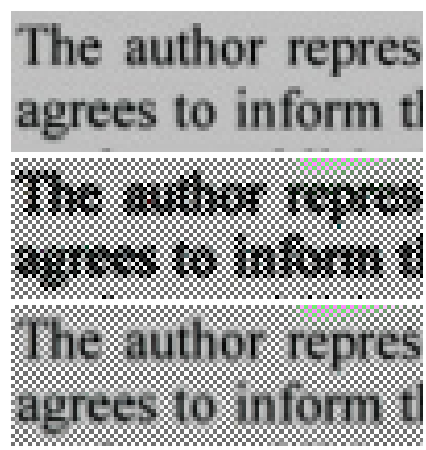


Fig. 7 Effect of region restriction: Top, original, Middle, stego image without restriction, Bottom, stego image with borders untouched (embedded in G channel with maximum strength and directly imported bitmap images.)

To address this problem, we relaxed the restriction on block substitution using a signal $\hat{w}(i, j)$ that takes arbitrary values in $[0 : 1]$. We first convert $w(i, j)$ to $\hat{w}(i, j)$, a blurred version of $w(i, j)$, using a Gaussian filter and mix the two blocks in proportion to the values of $\hat{w}(i, j)$; that is, instead of using Eqs. (1), (10) and (11), we compute

$$u(i, j) = a\hat{w}(i, j) + c(1 - \hat{w}(i, j)) \tag{23}$$

if $i + j$ is even, and

$$u(i, j) = b\hat{w}(i, j) + d(1 - \hat{w}(i, j)) \tag{24}$$

otherwise. Figure 6 shows the effect of this gradual transition. Embedded letters are far less visible in the right image because some coherent patterns at the watermark border in the left image have been suppressed. This improvement was applied to all experiments described in section 6.

5.2 Jagged Edges

Because the cover image is replaced with checkered blocks

in the encoding process, the sharpness of the original image is unavoidably compromised. This is most conspicuous in document images, where the edges of the text take on a jagged appearance.

To improve the text visibility, we can refrain from encoding around the border of the text. More specifically, we separate the text from the background in the cover image, dilate the text regions and paste the original image into these regions, while normal encoding is performed only outside them. Note that the text itself is not subject to encoding if it is drawn in black, so this process extends non-encoding regions slightly beyond the border of the text. An example is shown in Fig. 7. We see that the jagged edges of the letters are smoothed out when we prohibit encoding around the borders. This was applied to document images in experiments.

6. Experiments

In this section, we show the simulation results for ramp images, natural images, and document images. In all our experiments, unless otherwise stated, we fine-tuned the gamma table using the method in Sect. 4 so that it fits Dell's U2412M monitor and used the second monitor and the two printers listed in Table 1 for testing. In all results reported in this section, images on the monitor were taken by a Nikon D60 digital camera from distance that diffuses pixels enough to avoid moiré patterns. Printed images are imported digitally by scanning the sheet using a Canon LiDE210 or Epson GT-D1000 scanner. Note that the images might look different depending on the viewing acuity of readers, although we did our best to avoid this in preparing them.

6.1 Ramp Images

Because the possible depth of modulation depends on the cover signal luminance level, we first experimented with a ramp image.

Figure 8 is a screen shot of images on the Dell U2412M monitor. The watermark at the top is vertically split embedding (a, b) patterns in the bottom and (c, d) patterns in the top halves. Following are the cover and stego images respectively. The watermark has been embedded in the G channel with $r = 1.0$. We see that the shades of gray are well conserved and that the watermark is almost invisible.

Figure 9 shows stego images printed on paper using the Epson LP-S6160 printer. The watermark has been embedded in the B, G, and R channels separately and all three channels together (Y), with $r = 1.0$. It is seen that the printer is not good enough at producing gradation of (a, b) patterns for which the brightness level changes more abruptly at an intermediate level. Because of this, the difference in appearance is clear between the upper and lower halves in panels (a)–(c). This is most dominant in the middle range of gradation, while it is diminished towards both ends of the ramp signal. This means that the watermark embedded in the middle region would be easier to see when printed.

We see in Fig. 9(d) that embedding in the Y channel only makes the border line visible with no apparent difference in the upper and lower regions. This is because the two blocks are formed by simply exchanging dark and bright pixels for $r = 1.0$, as shown in Fig. 4. Reducing the value of r can produce the difference as in Fig. 10, which shows the results for smaller values of r.

Table 1 Equipment used in experiments

item	make	model
1st monitor	Dell	U2412M
2nd monitor	Mitsubishi	RDT233WLM
1st printer	Epson	LP-S6160
2nd printer	Canon	LBP9200C

6.2 Natural Images

We picked up two frames from SMPTE's StEM movie [30], both having large smooth areas with medium intensity levels. We downsampled each of them to 840×352 pixels and embedded a text pattern with repeated words in their color channels (B, G, R and Y) with varying strengths of $r = 0.25, 0.50, 0.75$, and 1.00.

Watermarks were barely visible on the Dell U2412M monitor in all cases. Examples of stego images displayed on this target monitor are shown in Fig. 11. It should be

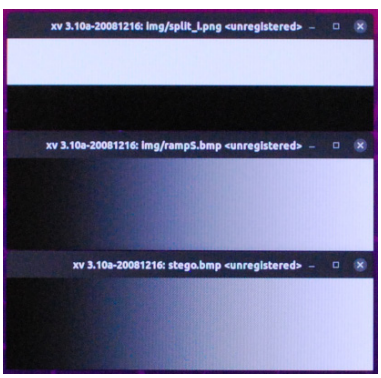
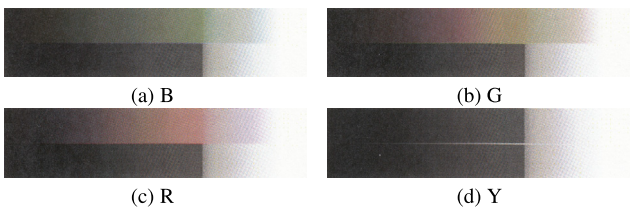


Fig. 8 Watermark hidden in ramp image on the Dell U2412M monitor (watermark, original image, and stego image from top to bottom)



(a) B (b) G

(c) R (d) Y

Fig. 9 Printed ramp images watermarked in different channels (Epson LP-S6160 printer)

(a) $r = 0.25$ (b) $r = 0.50$

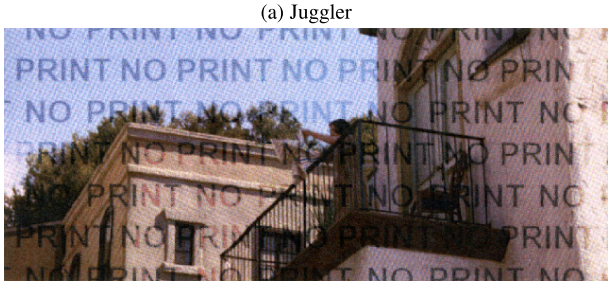
Fig. 10 Printed ramp images watermarked in the Y channel with lesser strength (Epson LP-S6160 printer)

(a) Juggler

(b) Balcony

Fig. 11 Stego images displayed on the Dell U2412M monitor (watermarked in the Y channel with $r = 0.25$)

(a) Juggler



(b) Balcony

Fig. 12 Stego images in Fig. 11 produced by the Epson LP-S6160 printer

noted that in the "Balcony" image, a closer look with trained eyes may reveal a slight change in uniform areas such as skies and building walls if the G channel is used. Figure 12 shows the same images when they are printed by the Epson LP-S6160 printer. The embedded "NO PRINT" message is clearly visible. Figure 13 shows (a) the result of printing by the Canon LBP9200C printer and (b) a screen shot of an image displayed on the Mitsubishi RDT233WLM monitor. The watermark on the monitor is obscured but still recognizable.

Table 2 summarizes how the hidden message appeared for various parameters when the watermarked "Juggler" image was viewed on devices other than the Dell U2412M monitor. We admit that this is not a result of objective testing but only to reflect the authors' evaluation. We note the

(a) Printed by the Canon LBP9200C printer

(b) Displayed on the Mitsubishi RDT233WLM monitor

Fig. 13 Stego images in Fig. 11(a) rendered by other devices

Table 2 Watermark visibility on Juggler image (+, weakly visible; ++, visible; +++, strongly visible. Note that the Mitsubishi RDT233WLM monitor has a viewing angle dependency. The data shown are estimates when the screen is viewed from a slightly elevated position.)

channel	r	S6160	9200C	RDT233
B	0.25	++	++	+
	0.50	++	++	+
	0.75	++	++	+
	1.00	++	++	+
G	0.25	+++	+++	++
	0.50	+++	+++	++
	0.75	+++	+++	++
	1.00	++	+++	++
R	0.25	++	++	++
	0.50	++	++	++
	0.75	++	++	++
	1.00	++	++	+
Y	0.25	+++	+++	++
	0.50	+++	+++	++
	0.75	+++	+++	++
	1.00	++	+++	++

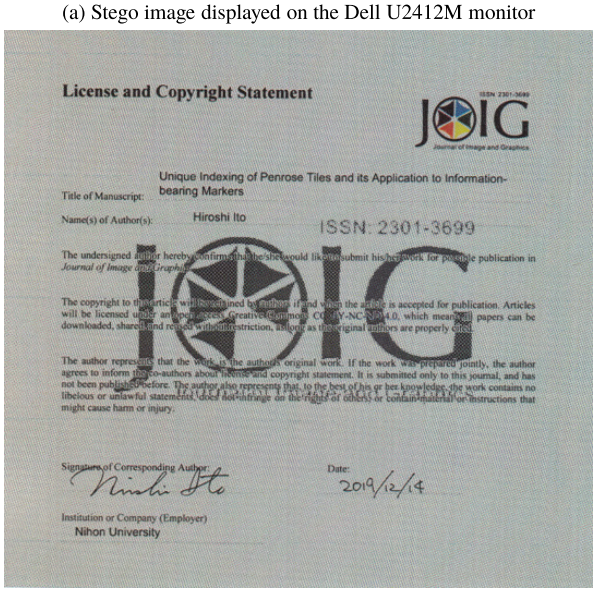
following points:

1. The G channel and combinations including G, such as Y, tend to produce maximum visibility.
2. The embedding strength is not necessarily in proportion to the visibility. In many cases, $r = 0.25$ is comparable to $r = 0.75$.
3. Visibility can be diminished for $r = 1.0$. This is because two blocks can become flipped in relation to each other depending on the colors.
4. As expected, we achieve stronger visibility between different devices (printers and monitors) than different models of the same device.

6.3 Document Images

Document images that are allowed to be seen only on displays would be a good application of our watermarking

(a) Stego image displayed on the Dell U2412M monitor



(b) Same image printed by the Epson LP-S6160 printer

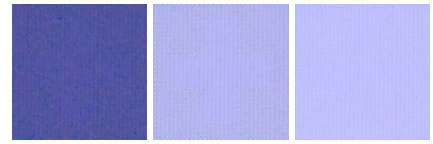
Fig. 14 Document image watermarked in the Y channel with $r = 0.25$

method. Here, we assume that documents have letters written in black against white backgrounds. Unfortunately, a 100% white background cannot be watermarked by our method, so we darkened the background to around 70% white. We used the method in Sect. 5.2 to reduce the degradation of text visibility.

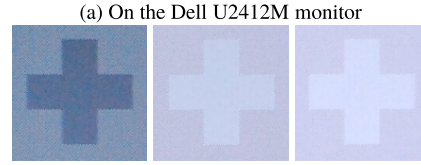
An example of stego images is shown in Fig. 14. We see that the embedded logo clearly appears in (b) when the image is printed, while it is well hidden in (a). Similar results were obtained for various values of embedding channels and strength.

6.4 Tuning for Different Monitors

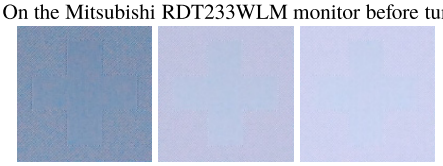
So far we have used a gamma table exclusively tuned for the Dell U2412M monitor. Since the hidden nature of our method varies from device to device, we experimented with invisibility under a different monitor. Because producing a complete gamma table is time-consuming, we just show a few examples of (x_1, x_2) and x_3 that are comparable to the eye, where we try to equalize the brightness of a uniform gray signal of x_3 and a block with $a = x_1$ and $b = x_2$ for the Mitsubishi RDT233WLM monitor.



(a) On the Dell U2412M monitor



(b) On the Mitsubishi RDT233WLM monitor before tuning



(c) On the Mitsubishi RDT233WLM monitor after tuning

Fig. 15 Effect of tuning (In each row, $(x_1, x_2) = (0, 128), (0, 255)$ and $(128, 255)$ respectively from left to right. In (a), x_3 was tuned for the Dell U2412M monitor such that $x_3 = 101, 180, 195$ and viewed on the same monitor. In (b), the same values of x_3 were used but viewed on the Mitsubishi RDT233WLM monitor. In (c), x_3 was tuned for RDT233WLM such that $x_3 = 118, 167, 176$ respectively.)

Figure 15 shows stego images with a cross mark embedded in various brightness levels of backgrounds. In (a), the x_3 was adjusted to hide the mark for the Dell U2412M monitor. The same values were used for the Mitsubishi RDT233WLM monitor in (b). Due to mismatch in gamma properties the hidden cross mark began to appear. Then we adjusted the value of x_3 to fit to the Mitsubishi RDT233WLM monitor in (c). Detailed parameters are listed in the caption.

7. Conclusion

A novel information hiding method that exploits color combinations has been proposed. It is device-dependent because the information is hidden from the eye after careful matching between the signal levels and the color reproduction properties, which usually differ for various equipment and models. We have shown that the watermark on monitors becomes visible when printed.

Limitations are 1) reduced resolution due to encoding, 2) inability to embed information into saturated images, and 3) weak security standard, that is, no protection of secrets by keys. Some partial solutions to 1) and 2) are discussed in the text. In addition to these, we might be able to restrict the encoding and embedding to slowly changing areas of the cover image. If the watermark is well hidden, these rather uniform areas are more suitable for embedding because the

watermark stands out more clearly when it appears in uniform areas than when it appears in areas with complex textures.

We need to further investigate the recovery process. The effect of changing parameters such as brightness and contrast on the same monitor requires more study. As shown partially in the experiment, the proposed method seems to be unable to avoid screen capture by digital cameras. Making recovery possible in this process would produce many useful applications. Finally, we found that resizing the stego image is another way of recovery and we often used this method to check whether the information is actually hidden in preparing the materials for this paper. How this can be used in a real application is another topic of investigation.

References

[1] D. Jütte, The age of secrecy – Jews, Christians, and the economy of secrets, 1400–1800, Yale University Press, London, 2015.
[2] P. Wayner, Disappearing cryptography – Information hiding: steganography and watermarking, Morgan Kaufmann, Burlington, 2009.
[3] M. Guri, B. Zadov, and Y. Elovici, "ODINI: Escaping sensitive data from faraday-caged, air-gapped computers via magnetic fields," IEEE Trans. Inf. Forensics Security, vol.15, pp.1190–1203, 2020.
[4] J.D. Park and J.F. Doherty, "A steganographic approach for covert waveform design," Proc. IEEE Military Communications Conference, Nov. 2016.
[5] L. Caviglione, M. Gaggero, J-F. Lalande, W. Mazurczyk, and M. Urbański, "Seeing the unseen: Revealing mobile malware hidden communications via energy consumption and artificial intelligence," IEEE Trans. Inf. Forensics Security, vol.11, no.4, pp.799–811, April 2016.
[6] K. Macrakis, Prisoners, lovers, and spies: The story of invisible ink from Herodotus to al-Qaeda, Yale University Press, London, 2014.
[7] P. She, Y. Ma, Y. Qin, M. Xie, F. Li, S. Liu, W. Huang, and Q. Zhao, "Dynamic luminescence manipulation for rewritable and multi-level security printing," Matter, vol.1, pp.1644–1655, no.6, Dec. 2019.
[8] H. Zhao, X. Qin, L. Zhao, S. Dong, L. Gu, W. Sun, D. Wang, and Y. Zheng, "Invisible inks for secrecy and anticounterfeiting: From single to double-encryption by hydrochromic molecules," Applied Materials and Interfaces, vol.12, no.7, pp.8952–8960, Jan. 2020.
[9] J. Fridrich, Steganography in digital media, Cambridge University Press, Cambridge, 2010.
[10] G.J. Simmons, "The prisoners' problem and the subliminal channel," Advances in Cryptography, pp.51–67, Springer, Boston, 1984.
[11] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," Proc. Security, Steganography, and Watermarking of Multimedia Contents IX, Feb. 2007.
[12] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," Lect. Notes Comput. Sci., vol.6387, pp.161–177, Springer, Berlin, 2010.
[13] G. Xu, H-Z. Wu, and Y-Q. Shi, "Structural design of convolutional neural networks for steganalysis," IEEE Signal Process. Lett., vol.23, no.5, pp.708–712, May 2016.
[14] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," IEEE Trans. Inf. Forensics Security, vol.14, no.5, pp.1181–1193, May 2019.
[15] S. Benard, P. Bas, J. Klein, and T. Pevný, "Explicit optimization of min max steganographic game," IEEE Trans. Inf. Forensics Security, vol.16, pp.812–823, 2021.
[16] B. Ma and Y.Q. Shi, "A reversible data hiding scheme based on code division multiplexing," IEEE Trans. Inf. Forensics Security, vol.11, no.9, pp.1914–1927, Sept. 2016.
[17] F. Balado, "Optimum reversible data hiding and permutation coding" Proc. IEEE International Workshop on Information Forensics and Security, Nov. 2015.
[18] Y-C. Chen, T-H. Hung, S-H. Hsieh, and C-W. Shiu, "A new reversible data hiding in encrypted image based on multi-secret sharing and lightweight cryptographic algorithms," IEEE Trans. Inf. Forensics Security, vol.14, no.12, pp.3332–3343, Dec. 2019.
[19] S. Kim, R. Lussi, X. Qu, and H.J. Kim, "Automatic contrast enhancement using reversible data hiding" Proc. IEEE International Workshop on Information Forensics and Security, Nov. 2015.
[20] P. Korus, J. Bialas, and A. Dziech, "Iterative filtering for semi-fragile self-recovery," Proc. IEEE International Workshop on Information Forensics and Security, Dec. 2014.
[21] I.J. Cox, M.L. Miller, J.A. Bloom, J. Fridrich, and T. Kalker, Digital watermarking and steganography, 2nd Edition, Morgan Kaufmann, Burlington, 2008.
[22] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," IBM Systems Journal, vol.35, Nos. 3 & 4, pp.313–333, 1996.
[23] I.J. Cox, J. Killan, F.T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," IEEE Trans. Image Process., vol.6, no.12, pp.1673–1687, Dec. 1997.
[24] GuardSoft, https:www.guard-soft.com
[25] W.H. Mowry, Jr., M.J. McElligott, V.J. Tkalenko, Jr., and J. Baran, "Protected document," US Patent, 4227720, Oct. 1980.
[26] D.C. Gordon, "Copy indicator for a document," US Patent, 4891666, Jan. 1990.
[27] W.H. Mowry, Jr., "Security document containing encoded data block," International Patent, WO98/56589, Dec. 1998.
[28] J.W. Wu, "Digitally printed anti-copy document and processes and products therefore," US Patent, 9738106, Aug. 2017.
[29] H. Ito, "Embedding overlay information in images on four color displays," Proc. IEEE Global Conf. Consum. Electron., Oct. 2012.
[30] StEM materials, https://members.smpte.org/store/stem- materials

Hiroshi Ito received B.S and M.S degrees in Electrical Engineering from Kyoto University, Japan in 1981 and 1983 respectively. In 1983, he joined the Imaging Systems Laboratory, Mitsubishi Electric Corporation, Japan. From 1994 to 1995, he was a visiting research assistant at the University of Maryland, MD, USA. Since 2008, he has been with the College of Industrial Technology, Nihon University. His research interests include digital signal processing and information security.

Tadashi Kasezawa received a B.S. degree in communication engineering and a Ph.D. degree in information science from Tohoku University, Japan in 1983 and 1998, respectively. He is currently a professor in the Department of Computer Science, College of Engineering, Nihon University. His research interests include signal processing, machine learning and computer vision.