Robust Watermarking in Slides of Presentations by Blank Space Coloring: A New Approach

Tsung-Yuan Liu[1] and Wen-Hsiang Tsai[1,2]

[1] National Chiao Tung University, Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan
gis91811@cis.nctu.edu.tw
[2] Asia University, Department of Computer Science and Information Engineering, Asia University, Taichung, 413, Taiwan
whtsai@cis.nctu.edu.tw

Abstract. A new robust method to embed imperceptibly a watermark image into the slides of a presentation is proposed. The watermark is partitioned into blocks and embedded into the space characters existing in the slides in a repeating pseudo-random sequence. The embedding is achieved by changing the colors of the space characters into new ones which are results of encoding the contents and indices of the blocks. The embedded watermark is resilient against many common modifications on slides, including copying and pasting of slides; insertion, deletion and reordering of slides; slide design changes; and file format conversions. A security key is used during embedding and extraction of a watermark, such that if an offending presentation contains slides taken from presentations watermarked with different security keys, each watermark can be extracted reliably in turn with the respective key using a weighted voting scheme also proposed in this study. Experiments conducted in Microsoft PowerPoint confirm the feasibility of the proposed method. On average, a recognizable watermark of size 64×64 can be extracted from a presentation containing five watermarked slides. The proposed method is useful for various applications of information hiding involving slides, including slide copyright protection, slide authentication, covert communication through slides, etc.

Keywords: information hiding, slide presentation, watermarking, Microsoft PowerPoint, OpenOffice Impress.

1 Introduction

The rapid development of digital technologies in the past decade has made the reproduction and transmission of digital contents simpler and cheaper as ever. However, digital infringement also arises, and anything of value becomes a possible target for illegal duplications and misuses. Digital watermarking researches [1,2,3] offer copyright protection mechanisms to counter offending uses by embedding watermarks into media to prove their ownerships and reveal their distributions. Such watermarks should be imperceptibly hidden in the host media,

Y.Q. Shi (Ed.): Transactions on DHMS IV, LNCS 5510, pp. 49–64, 2009.

and must be resilient against probable modifications by offenders. Watermarking in images, for example, should be robust against scaling, cropping, and format conversion attacks. Lin et al. [4] and O' Ruanaidh and Pun [5] have proposed rotation, scaling and translation resilient watermarking methods for images.

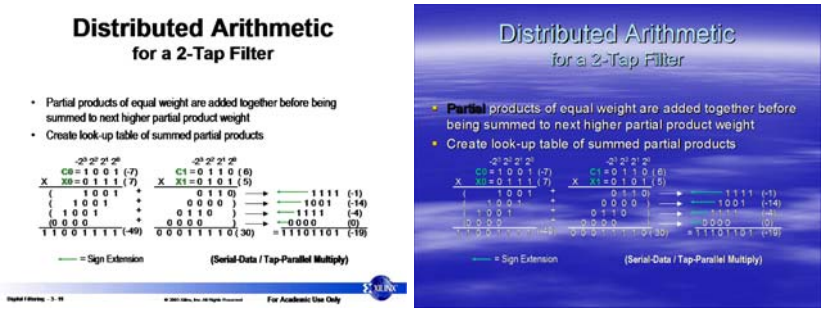
Slide presentation is an increasingly popular way of communication, thanks to cheap projectors and a widespread deployment of them in institutions and businesses. Slide presentations are used for numerous purposes, including lecturing, training, idea presentation, and sales reporting. The slides of a presentation usually are crafted carefully and include texts, images, animations, audios, videos, etc., in order to present valuable contents concisely and lively. It is usually desired to protect the copyright of slides. One way for this purpose is to embed digital watermarks into the slides. To the best knowledge of the authors, digital watermarking of slides has not been investigated before. The traditional means to achieve copyright protection of slides is to place an annoying *visible* logo in the slide background.

One scenario of typical attacking on slides is a person composing a presentation simply by stealing slides out of others' presentations. It is desirable that such malpractice be identifiable automatically. Another different application scenario is where there are confidential internal slides and public marketing slides in a company, and while it is perfectly fine to mix those slides in an internal talk, it is undesirable for the confidential slides to be carelessly shown in external presentations. It will be convenient if there is an automatic means to detect whether a set of slides contain any of the confidential slides.

In this paper, a robust watermarking method for slide copyright protection is proposed, which embeds an *invisible* watermark image imperceptibly into the slides of a presentation. The embedded watermark survives common operations performed on the slides, such as copying and pasting of slides, addition of new slides, removal of slides, reordering of slides, editing of slide contents, and modification to the slide design. The last operation is often applied by a presentation designer to quickly change the style of slides for a desired appearance. The fonts, styles, and colors of texts in the slides, among others, are automatically modified according to a slide design template, as seen in the example shown in Fig. 1.

There are two different ways of setting the colors of texts in presentations. The first, more common approach is to select a color from a color palette, and the selected *index* in the color palette is actually stored. The second approach is to directly set the color of the text usually in the RGB color space, which is a triplet specifying the relative intensities of the red, green, and blue components of the color, usually each in the range of 0 to 255, such as (*red*: 0, *green*: 255, *blue*: 255) for the color yellow. A slide editing software application usually allows any of the two approaches to be used for any text in a slide, and it is possible, for example, to have a whole sentence colored using the color palette approach except the first word which is highlighted using a special RGB color.

For automatic modifications of text colors to work when applying different slide design templates, the first approach of text coloring is used. In more details, different slide design templates have different color palettes, and the colors



(a) (b)

Fig. 1. Illustration of slide designs. (a) A slide from a tutorial from Xilinx, Inc. with black texts on white background. (b) The slide in (a) with a slide design template of bluish background applied.

of the titles of a slide, the texts and the hyperlinks in a slide, etc. are set to specific colors from the color palette. The color palette in a design template supplied by a slide editing software application is designed to ensure high contrast texts and an overall appealing appearance. For example, slide templates with a white background will have matching black texts, and templates with a dark background will have texts in light colors. By adhering to the color palettes when editing slides, one can ensure that the colors will be modified appropriately and automatically when selecting a different slide design.

The automatic modifications made during slide design changes pose great challenges to watermarking in slides, and make many previously proposed watermarking techniques ineffective. Text watermarking by making use of the text layout [6] or the LSBs of the text colors, for example, are either removed after the process of applying a slide design, or revealed by the modification. In more details, the method of manipulating the LSBs of the text colors for embedding a watermark does not work because such modifications mean that the colors will no longer be palette entries but specified RGB colors, and thus not altered automatically when a different design template is applied. As an example, the LSB replacement technique has been applied to the word "Partial" in the Fig. 1(a) by changing its color from completely black to dark gray. Although the modification is imperceptible in the original slide, the embedding becomes eye-catching after a slide design modification. The resultant slide after applying a design template that has white text on blue background is shown in Fig. 1(b). The text with the unchanged dark gray color now stands out from the white texts around it. We also note that the visible logo and the copyright information in the slide background have been *removed automatically* after the application of the slide design.

In this paper, we propose to use the *colors of space characters* (called simply as *spaces* hereafter) to embed watermarks, that is, to embed watermark data by altering the color of a space between two words. The colors of the spaces in a slide can be changed without affecting the visual appearance of the slide, and these colors are *unchanged* during application of other slide designs, changing

of slide layouts, reordering of slides, reordering of texts in slides, and conversion of file formats, as found in this study. As the spaces are *transparent*, we can manipulate the colors freely for the purpose of information embedding, using either the color palette or the RGB coloring approach, without any visual side-effects. The latter is chosen due to its greater embedding capacity. Specifically, we divide a watermark image into blocks and encode the index and data of each block into a RGB color value which is then taken to replace the original color of a space, accomplishing the embedding of a watermark block's information into a space character. For security, the watermark blocks are embedded into the slides in a random sequence created by a pseudo-number generator with a user-specified key. To extract the embedded watermark, a weighted voting scheme is designed to handle the problem of watermark recovery from presentations that contain watermarked slides and non-watermarked slides. The average number of watermarked slides required to achieve a specified percentage of coverage of an extracted watermark image is also analyzed. Experimental results showing feasibility of the proposed approach is also included.

In the remainder of this paper, the details of the watermark embedding and extraction processes follow in Section 2. Section 3 presents some of our experimental results, and in Section 4 we conclude with some suggestions for future works.

2 Robust Watermarking in Slides of Presentations

In the proposed method, the spaces among the texts in the slides of a presentation are used to embed a watermark, which is assumed to be an $N \times N$ black-and-white image, such as a logo of a company. The image is divided into M blocks, each containing L pixels, where $L = N^2/M$. The L pixel values of each block are concatenated in a raster scan order into a string, which we call a *block data string* in the sequel. The basic idea of watermarking in the proposed approach is to *encode* the data string and the index of each block into an RGB color, with which the color of a text space in a slide is replaced. That is, watermarking here consists of the two steps of *watermark block encoding* and *space color replacement*.

Since a copyright violator might copy only some of the watermarked slides of a presentation, we choose to embed the watermark *repeatedly* throughout the slides. The embedding of the block index along with the block data string means that the embedded data are *invariant* against insertion and reordering of slides or slide contents.

2.1 Watermark Embedding

More specifically, during watermark embedding, the spaces are taken for data embedding *in the reading/presentation order*, that is, the spaces in the first slide are used first in a top-to-bottom and left-to-right order, followed by the spaces in the second slide, and so on. While the blocks of the watermark are embedded into this normal sequence of spaces, the indices of the embedded blocks instead

follow a pseudo-random sequence controlled by a key to increase security of data protection. The algorithm below describes the proposed process of watermark embedding.

Algorithm 1: Embedding a watermark image into slides of a presentation.

Input: A set P of slides of a presentation; a watermark image I to be embedded, which is partitioned into M block data strings $B_1, B_2, \ldots, B_M$; and a user-specified key K.

Output: Watermarked slides of P with I embedded by coloring the spaces in P appropriately.

Steps:

1. Generate a random integer sequence $E = i_1, i_2, \ldots, i_M$ in the range of $1, 2, \ldots, M$ without repetitive values, using K and a pseudo-random number generator f.
2. Find all spaces $s_1, s_2, \ldots, s_Q$ in P in the reading/presentation order, and repeat the sequence E for $\lceil Q/M \rceil$ times to arrive at another sequence $E' = j_1, j_2, \ldots, j_R$, where $R = M \times \lceil Q/M \rceil \geq Q$.
3. For each space s_k in P, $1 \leq k \leq Q$, pick out the index j_k in E' and the corresponding block data string B_{j_k}, and encode the pair (j_k, B_{j_k}) into a color C to replace that of s_k in the following way:
   (a) combine j_k and B_{j_k} into an integer $A = j_k \times 2^L + B_{j_k}$, regarding B_{j_k} as an L-bit number;
   (b) compute color $C = (R, G, B)$ by taking the three components respectively to be $B = A \bmod 2^l$, $G = \lfloor A/2^l \rfloor \bmod 2^l$, and $R = \lfloor A/2^{2l} \rfloor$, where each component is assumed to have l bits;
   (c) replace the color of space s_k with C.

The embedding of the blocks of a watermark image in a predefined random sequence as described in the above algorithm has several benefits, as described in the following.

1. A recognizable partial watermark can be extracted if an offender copies only a portion of the watermarked material. Fig. 2 shows two series of watermark images with different percentages of blocks successfully reconstructed. The watermark can be recognized already when only half of the blocks (i.e., $M/2$ blocks) are present.
2. If an offender puts some of the watermarked slides together with other non-watermarked ones, the watermarked slides can still be correctly identified using a weighted voting scheme proposed in this study (described later), which gives more weights to extracted block data strings with indices in right orders defined by the random sequence.
3. Furthermore, if an offender copies watermarked slides from multiple sources with different user keys and watermark images, the individual watermark images can be extracted correctly in turn by using the respective user keys, as confirmed in the experiment.

Fig. 2. Two series of watermark logos with different block coverages

2.2 Weighted Voting of Partial Sequences for Watermark Extraction

During watermark extraction, it might happen that a given set of slides includes both watermarked slides and non-watermarked ones, as mentioned previously. Since a space with no embedded data in a non-watermarked slide also has a color value which may be as well decoded into a block data string and a block index, a weighted voting scheme is proposed in this study to identify the spaces that really contain watermark data, so that a correct watermark can be reconstructed. We assume that the block indices extracted from non-watermarked slides to be uniformly distributed in the range of $\{1, 2, \ldots, M\}$.

More specifically, since an offender usually copies a complete slide or an entire sentence in a slide at a time, the order of the spaces in the copied contents are preserved. The basic idea of the proposed weighted voting technique is to analyze the sequence of block indices extracted from the spaces of the slides of a suspect presentation, and check whether the extracted sequence follows an expected sequence. Blocks that follow the expected sequence are more likely to contain watermark data than those that do not, and thus are given larger weights in using them for reconstructing the watermark image.

We denote the sequence of pairs of block indices and block data strings that have been extracted from the spaces of a suspect presentation as $S = \{(j_1, B_1), (j_2, B_2), \ldots, (j_Q, B_Q)\}$, where j_k is the index of block data string B_k. The expected sequence $E = \{i_1, i_2, \ldots, i_M\}$, as generated by Algorithm 1, is a random integer sequence in the range of $\{1, 2, \ldots, M\}$ without repetitive values. As the blocks of the watermark are embedded repeatedly according to the sequence E, we regard the sequence to be cyclic and use the new notation E^+ to specify a sequence of arbitrary length formed by concatenating sequence E repeatedly. Also, we use the notation $\{j_k, j_{k+1}, \ldots, j_l\} \subset E^+$ to mean that the sequence $\{j_k, j_{k+1}, \ldots, j_l\}$ is a subsequence of E^+.

Now consider a text space s_k which does not contain previously embedded watermark data, and from which the pair (j_k, B_k) in S is extracted. There are three cases here.

1. For $k = 1$ (corresponding to the case that s_k is the first space of the suspect presentation), the probability that the index sequence $\{j_k, j_{k+1}\} \subset E^+$ is $1/2^M$, irrespective of whether the block with index j_{k+1} contains embedded watermark data or not. The probability that $\{j_k, j_{k+1}\} \not\subset E^+$ is $1 - 1/2^M$.

2. For $k = Q$ (corresponding to the case that s_k is the last space of the suspect presentation), the probability that $\{j_{k-1}, j_k\} \subset E^+$ is $1/2^M$, irrespective of whether the block with index j_{k-1} contains embedded watermark data or not. The probability that $\{j_{k-1}, j_k\} \not\subset E^+$ is $1 - 1/2^M$.
3. For $1 < k < Q$, the probability that $\{j_{k-1}, j_k, j_{k+1}\} \subset E^+$ is $1/2^M \times 1/2^M = 1/2^{2M}$, irrespective of whether the blocks with block indices j_{k-1} and j_{k+1} contain embedded watermark data or not. The probability that $\{j_{k-1}, j_k, j_{k+1}\} \not\subset E^+$ but $\{j_{k-1}, j_k\} \subset E^+$ is $1/2^M \times (1-1/2^M)$, and so is the probability that $\{j_{k-1}, j_k, j_{k+1}\} \not\subset E^+$ but $\{j_k, j_{k+1}\} \subset E^+$. The probability that $\{j_{k-1}, j_k\} \not\subset E^+$ and $\{j_k, j_{k+1}\} \not\subset E^+$ is $(1 - 1/2^M) \times (1 - 1/2^M)$.

We propose to weigh the block data strings in S according to the minus base-2 logarithm values of the above probabilities in using the data string values in the process of watermark extraction. That is, for $k = 1$, block data string B_k is given a weight of $-\log_2(1/2^M)$, which we denote as W_A, if $\{j_k, j_{k+1}\} \subset E^+$; and if $\{j_k, j_{k+1}\} \not\subset E^+$, then block data string B_k is given a weight of $-\log_2(1-1/2^M)$, which we denote as W_B. Similarly, if $k = Q$, block data string B_k receives a weight of W_A if $\{j_{k-1}, j_k\} \subset E^+$, and a weight of W_B if $\{j_{k-1}, j_k\} \not\subset E^+$. For $1 < k < Q$, the block data string B_k is given a weight of $-\log_2(1/2^{2M}) = 2W_A$ if $\{j_{k-1}, j_k, j_{k+1}\} \subset E^+$, a weight of $-\log_2[1/2^M \times (1 - 1/2^M)] = W_A + W_B$ if either $\{j_{k-1}, j_k\} \subset E^+$ or $\{j_k, j_{k+1}\} \subset E^+$, and a weight of $-log2[(1 - 1/2^M) \times (1 - 1/2^M)] = 2W_B$ if none of the above are true. The block data strings with the largest weights are then chosen for watermark reconstruction, as described in the following algorithm.

Algorithm 2: Weighted voting of partial index sequences for watermark extraction.

Input: An expected sequence $E = \{i_1, i_2, \ldots, i_M\}$; an input sequence of extracted block indices and data strings $S = \{(j_1, B'_1), (j_2, B'_2), \ldots, (j_Q, B'_Q)\}$; and a threshold T.

Output: Block data strings $B_1, B_2, \ldots, B_M$ of the M blocks that comprise a watermark image I.

Steps:

1. Initialize $W_1, W_2, \ldots, W_M$ to be M empty sequences of 2-tuples.
2. For each pair (j_k, B'_k) in S with block index j_k and block data string B'_k, $1 \le k \le Q$, add the pair (B'_k, W) to the sequence W_{j_k} where:
   (a) for $k = 1$, $W = W_A$ if $\{j_k, j_{k+1}\} \subset E^+$, and $W = W_B$ if $\{j_k, j_{k+1}\} \not\subset E^+$;
   (b) for $k = Q$, $W = W_A$ if $\{j_{k-1}, j_k\} \subset E^+$, and $W = W_B$ if $\{j_{k-1}, j_k\} \not\subset E^+$;
   (c) for $1 < k < Q$, $W = 2W_A$ if $\{j_{k-1}, j_k, j_{k+1}\} \subset E^+$; else $W = W_A + W_B$ if either $\{j_{k-1}, j_k\} \subset E^+$ or $\{j_k, j_{k+1}\} \subset E^+$; or else $W = 2W_B$.
3. Derive the data string B_l of the lth block of I as follows where $1 \le l \le M$:
   (a) sum up the weights with identical data strings in W_l;
   (b) select the data string $B'_{\max}$ in W_l with the maximum weight $W_{\max}$;
   (c) if there are no such $B'_{\max}$'s, or if there were multiple such $B'_{\max}$'s, or if $W_{\max}$ is smaller than the threshold T, then regard B_l as missing, and

represent the lth block as a gray-colored block; else set the data string B_l to be $B'_{\max}$.

As an example, let $M = 4$, $E = \{3, 1, 4, 2\}$, $S = \{(2, \text{A}), (3, \text{B}), (1, \text{C}), (4, \text{D}), (1, \text{E}), (4, \text{F}), (2, \text{A})\}$, and $T = 0$. Then $E^+ = \{3, 1, 4, 2, 3, 1, 4, 2, 3, \ldots\}$. After Step 2 of the above algorithm, we have $W_1 = \{(\text{C}, 2W_A), (\text{E}, W_A + W_B)\}$ because the partial data set $\{(3, \text{B}), (1, \text{C}), (4, \text{D})\}$ in S forms an index sequence of $\{3, 1, 4\}$ which fits well with the first three indices of E^+, yielding the result of the pair $(\text{C}, 2W_A)$ in W_1; and the partial data set $\{(1, \text{E}), (4, \text{F})\}$ forms an index sequence of $\{1, 4\}$ found also in E^+, yielding $(\text{E}, W_A + W_B)$ in W_1. In similar ways, we can compute $W_2 = \{(\text{A}, W_A), (\text{A}, W_A)\}$, $W_3 = \{(\text{B}, 2W_A)\}$, and $W_4 = \{(\text{D}, W_A + W_B), (\text{F}, W_A)\}$. Accordingly, in Step 3 the block data strings P_1, P_2, P_3 and P_4 are set to C, A, B and D, respectively, with the weighting for data string A summed to be $2W_A$.

The above method ensures that sequences of blocks that contain watermark data dominate during the watermark image reconstruction in Step 3. However, if only a small portion of the watermarked contents are copied, then some of the blocks of the reconstructed watermark image may be missing. The threshold T is useful in this case, where setting T to a large value causes noise from non-watermarked blocks to be ignored. A value of at least W_A is recommended, since block weights from any partial sequences of the watermark contents are at least of the value of W_A.

In Step 3c of the algorithm, instead of ignoring all of the multiple candidate block data strings with the same weights, we could use a voting algorithm to restore the correct watermark pixel values amid noise, as described in the following.

Algorithm 3: Intra-block voting for pixel value reconstruction.

Input: A set S_C of V block data strings $B_1, B_2, \ldots, B_V$; and an adjustable threshold H, where $0.5 \leq H < 1$.

Output: Colors (black or white) $p_1, p_2, \ldots, p_L$ of the pixels comprising a block P of a watermark image.

Steps:

1. Set the color of each pixel p_j, $1 \leq j \leq L$, as follows:
   (a) count the number of blocks in S_C, whose corresponding pixel value is *black* (i.e., with bit value 1), and denote the number as C_B;
   (b) count the number of blocks in S_C, whose corresponding pixel value is *white* (i.e., with bit value 0), and denote the number as C_W;
   (c) set the color of p_j to be black if $C_B/V > H$; else set the color of the pixel to be white if $C_W/V > H$; or else set the color of the pixel to be gray, meaning the pixel color was indeterminate.

The basic idea of the above algorithm is that the block data strings decoded from text spaces that do not contain watermark data can be considered to contain random values. For each pixel, the number of blocks that have the corresponding pixel value of black is approximately the same as that having a pixel

value of white. The assumption of inclusion of the correct block data string causes the scale to tip towards the correct side. To handle the case where no correct blocks is available, as the case may be when only a few watermarked slides are taken, the value of H can be increased to reduce the resulting noise in the extracted watermark image.

2.3 Watermark Extraction

In the proposed watermark extraction process, the spaces in the slides of a suspect presentation are analyzed to extract a sequence of block indices and block data strings. Algorithm 2 is then used to analyze the extracted block indices and data strings to reconstruct the previously embedded watermark image. The algorithm below describes the details.

Algorithm 4: Extracting a watermark image from the slides of a suspect presentation.

Input: A set P of slides of a suspect presentation and a key K.

Output: A watermark image in P comprised by M block data strings B_1, $B_2, \ldots, B_M$.

Steps:

1. Generate the random integer sequence $E = \{i_1, i_2, \ldots, i_M\}$ in the range of $\{1, 2, \ldots, M\}$ without repetitive values, using K and the same pseudo-random number generator f used during watermark embedding.
2. Initialize S to be an empty sequence of pairs of block indices and block data strings.
3. Find all spaces $s_1, s_2, \ldots, s_Q$ in P in the same order as that of embedding, and for each space s_k, $1 \leq k \leq Q$, decode the color $C = (R, G, B)$ of s_k into a pair (j, D) of a block index j and a block data string D and put it into S in the following way:
   (a) compute an integer $A = R \times 2^{2l} + G \times 2^l + B$, assuming that the RGB color space has l bits per channel;
   (b) compute j and D as $j = \lfloor A/2^L \rfloor$ and $D = A \bmod 2^L$, respectively (because presumably $A = D \times 2^L + j$ according to Step 3a of Algorithm 1);
   (c) add (j, D) to S.
4. Reconstruct $B_1, B_2, \ldots, B_M$ using Algorithm 2 with E and S as inputs.

2.4 Embedding Capacity and Expected Reconstruction Coverage

When embedding the blocks of a watermark image into the spaces of slides, popular slide presentation formats like Microsoft PowerPoint and OpenOffice Impress can be used. Eight bits per color channel and hence 24 bits can be embedded into each text space in slides of such formats. This embedding capacity allows us to embed a black-and-white watermark image as large as 64×64 into the slides of a presentation of normal sizes. When embedding a watermark image of such a size, we first divide it into $M = 256$ blocks with each block

containing $L = 16$ pixels. Each space then is used to store an 8-bit block index and 16-bits of pixel values. For a presentation we use in this study that contains slides with 40 spaces per slide on average, only seven slides is required to embed a complete watermark, and four slides may be sufficient to extract a recognizable watermark. The watermark image is embedded repeatedly into the slides as mentioned previously. Fig. 2 shows a series of 64×64 logos with different coverages that have been divided into 256 blocks of 4×4 pixels. If a smaller watermark image was used during watermark embedding, the number of spaces required to extract a recognizable watermark is reduced.

When an offender takes slides selectively, instead of consecutively, from a watermarked presentation, the block data contained in these slides may overlap with each other, meaning that a higher number of spaces are required to reconstruct a recognizable watermark. We estimate the number of watermarked spaces required to achieve the desired watermark image coverage by assuming that the offender draws R spaces from a watermarked presentation randomly, and that the block index in each of the drawn spaces is uniformly distributed in the sample space $\{1, 2, \ldots, M\}$. We denote the R random block indices as $i_1, i_2, \ldots, i_R$, and G the number of *distinct values* in $\{i_1, i_2, \ldots, i_R\}$ over the value M. In other words, G is the percentage of the blocks of the watermark that is contained in the R randomly chosen spaces, and the *expected coverage* $E(G)$ is the expected percentage of the watermark image that can be reconstructed. To derive $E(G)$, we first introduce random variables $I_1, I_2, \ldots, I_M$, where $I_j = 1$ if none of the values of $i_1, i_2, \ldots, i_R$ is equal to j, and $I_j = 0$ if at least one of the values is equal to j. The probability that $I_j = 1$ is $(1 - 1/M)^R$, and the expected value of I_j is thus $E(I_j) = (1 - 1/M)^R$. Since $G = [\sum(1 - I_j)]/M$, the expected coverage is

$$\begin{aligned} E(G) &= E\left(\frac{\sum(1 - I_j)}{M}\right) \\ &= \frac{M - \sum E(I_j)}{M} \\ &= 1 - (1 - 1/M)^R. \end{aligned} \tag{1}$$

From (1), the number of spaces R required to result in a desired expected coverage $E(G)$ is

$$R = \frac{\log(1 - E(G))}{log(1 - 1/M)}. \tag{2}$$

Using the above equation, we can estimate the number of spaces that are required to achieve a recognizable watermark image. A recognizable watermark should have at least 50% coverage, as seen in Fig. 2 and Fig. 3. With $E(G) = 0.5$ and $M = 256$, approximately $R = 178$ spaces, or about 5 slides, for a presentation containing on average 40 spaces per slide, are required according to Equation (2); and for a good quality watermark image with 80% coverage, approximately 412 spaces, or 11 slides, are required. In practice, images and drawings in slides should also be utilized during watermark embedding by using information hiding techniques for these media [1,7,8] to reduce the amount of slides required for a desired reconstructed watermark.

(a) (b)

Fig. 3. Illustration of watermark reconstruction coverage. (a) Three watermarks each with coverage of 50%; (b) the three watermarks with 80% coverage.

2.5 Robustness of Proposed Method against Common Operations

The watermark embedded into the slides of a presentation using the proposed method is resilient against many common operations performed on slides. In particular, the embedded watermark is robust against changes to the slide design template, as described in the introduction, whereas traditional visible logos are removed automatically in the process. Also, the automatic changes to the text and background colors lure a thieve to believe that no color information could have survived the transformation.

The watermark is also robust against copying and pasting of watermarked slides, as the colors and orderings of the spaces in the slides are unaltered during these types of processes. If we assume reasonably that there are at least two spaces in a slide, the block data strings embedded in the spaces of the slide will receive a sufficiently large weight using the proposed scheme for correct reconstruction of the embedded watermark image. Reordering of the slides in a presentation is a similar operation to copying and pasting of slides, and has little impact on correct watermark extraction. Reordering of slide contents is often conducted by moving pictures and text blocks around or by exchanging the order of the sentences in a slide. The former operation does not have any effect, while the latter is the same as reordering of slides if there are at least two spaces in a sentence.

Insertion of new non-watermarked slides or watermarked slides created with a different key into a slide set does not affect the watermarked contents, but only increases the amount of noise during reconstruction. Algorithms 2 and 3 are capable of selecting out the correct watermark data amid noise, as is verified in the experiments conducted in this study, where correct watermark images were reconstructed from watermarked slides that have been reordered and put together with slides that have been watermarked with different keys.

The proposed method is also resilient against removals of slides or slide contents, as long as sufficient watermarked contents remain. Experiments have shown that a recognizable watermark of size 64×64 can be reconstructed from approximately five watermarked slides.

Lastly, the watermark embedded using the proposed method has been proven to be robust against file format conversion attacks. Specifically, a presentation with slides watermarked using the proposed method was first saved in Microsoft PowerPoint in its PPT format. The file was then opened by another presentation software, OpenOffice Impress, and saved in the OpenDocument ODP format.

The ODP format file was then reopened by Impress, and finally saved back into the PPT format by Impress. Fig. 4(a) shows the first two slides of a test presentation before file format conversion, and Fig. 4(b) shows the same two slides after the above described format conversions from PPT to ODP and back to PPT. We note that the font type (changed from *Arial* to *Times New Roman*) and the font size (changed from 42pt to 44pt) of the title on the first slide, and the drawing on the second slide, among others, were changed during the file format conversions. The embedded watermark image, however, was untouched during the process and can be perfectly reconstructed.

3 Experimental Results

We implemented the proposed watermark embedding and extraction methods in C#.NET, and conducted a series of experiments using the popular presentation editing application Microsoft PowerPoint 2003. We use the *Automation* technique provided by Microsoft [9] to process PowerPoint presentation files. We have collected slides from the presentations of some projects we are currently investigating, as well as some presentations that are available from the web [10,11,12]. The average number of spaces per slide in these samples ranges from 35 to 60. Three slides with the characteristics listed in Table 1 are chosen for the experiments.

In the experiments, three logo images, each of size 64×64, are used as watermarks. Each of them was divided into 256 blocks of 4×4 pixels each, and individually embedded into the slides of the three presentations with different security keys using Algorithm 1. A presentation was then constructed by drawing slides randomly from the watermarked presentations. Specifically, N slides were drew randomly from each of the three presentations and then combined to form an experimental presentation that contains $3N$ slides. The three watermark logos were then extracted from the experimental presentation with the three respective keys in turn using Algorithm 4. The number of pixels that were correctly reconstructed in each of the three extracted watermark logos was counted, and the fractions of correct pixel extraction for the three images were recorded for each trial. This process was repeated 10,000 times for each value of N ranging from 1 to 19, and the average correct coverages of the three extracted watermarks are plotted in Fig. 5. To reconstruct a recognizable extracted watermark with

Table 1. Characteristics of presentations used in the experiments

	A	B	C
Number of slides	35	28	51
Total number of spaces	2086	1029	2605
Average number of spaces per slide	59.6	36.8	51.1
Maximum number of spaces in a slide	352	159	339
Minimum number of spaces in a slide	0	1	1
Standard deviation of spaces per slide	73.0	37.4	53.7

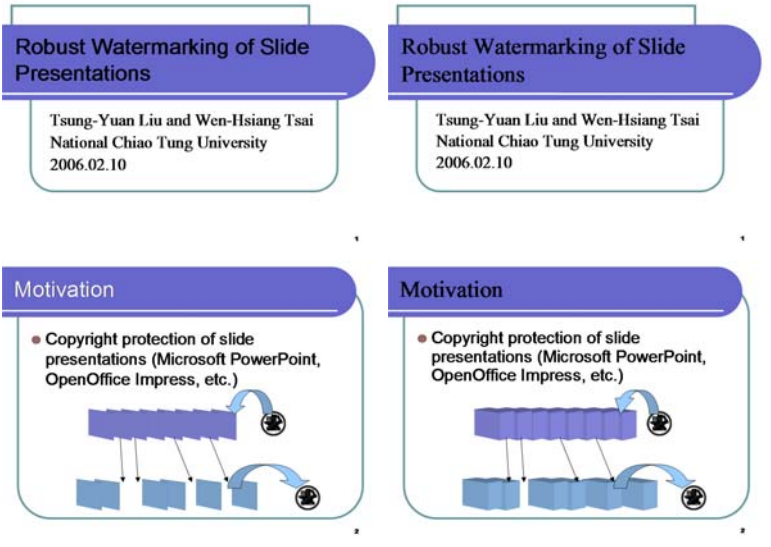


Fig. 4. An experimental result of file format conversion. (a) Two slides in Microsoft PowerPoint. (b) The two slides after file format conversion from PPT to ODP and back.

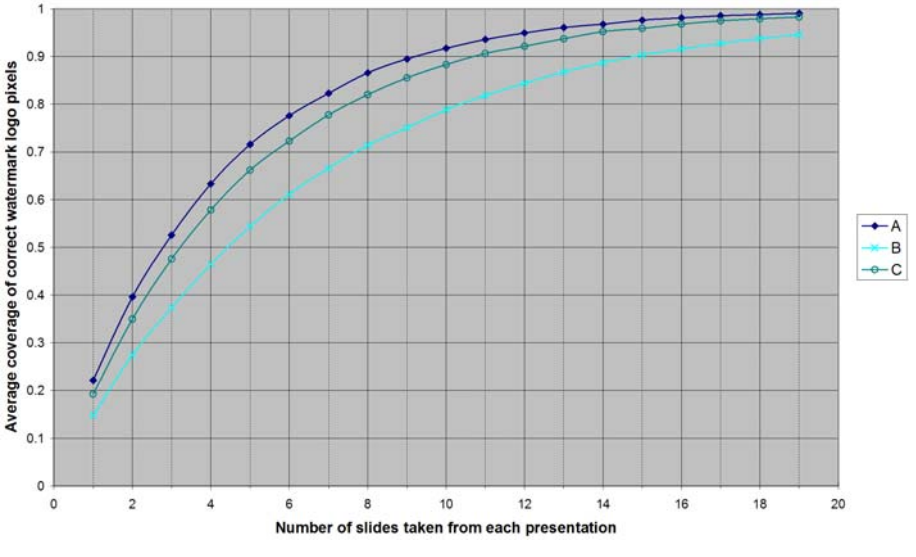


Fig. 5. Plot of average correct watermark pixel extractions from presentations constructed from randomly drawn slides

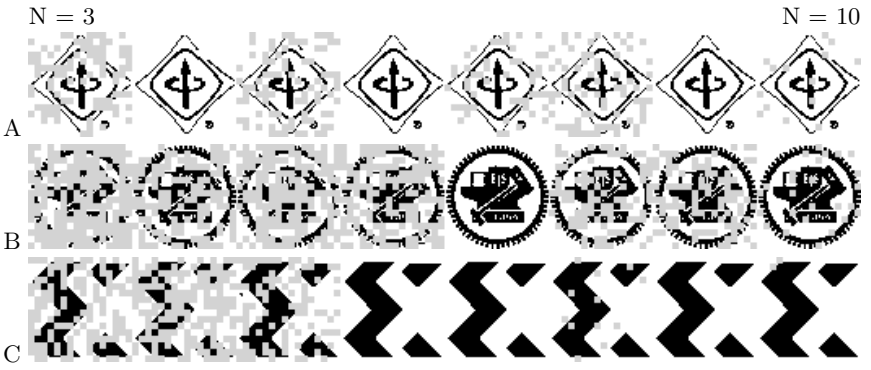


Fig. 6. An experimental result of the three extracted watermarks with N ranging from 3 to 10

at least 50% correct coverage, 3, 5, and 4 slides are required from presentations A, B, and C, respectively; and to reconstruct one with 80% correct coverage, 7, 11, and 8 slides are required from A, B, and C, respectively.

Fig. 6 shows one result of the three extracted watermarks for N ranging from 3 to 10. The result is imperfect because of the large variations in the number of spaces in each slide. Specifically, some slides in the presentations used in the experiments contain more than one hundred spaces. The selection of just a few of these slides will result in perfect reconstruction of the watermark image. For example, the presence of a slide from presentation C that contains 339 spaces (shown in Fig. 1) alone in an offending presentation allows the perfect reconstruction of the embedded watermark. On the contrary, if slides that contain less than ten spaces were picked during a trial of the experiment, then many blocks of the reconstructed watermark image will be missing. We note that the randomly constructed presentation contains slides watermarked with a key different to the one used for extraction, and can thus include incorrect data for these missing blocks. However, since such erroneous blocks do not follow the expected sequence specific for the extraction key, they are effectively filtered out by the proposed weighted voting scheme, as observed in the figure. The average percentages of the watermark that was incorrectly recovered fall in the range of $0.02\% \sim 0.24\%$ in the experiments.

The plot in Fig. 5 was normalized by multiplying the number of slides taken from each of the presentation by the average number of spaces per slide. The normalized plot, with the average number of spaces taken from each presentation as the x-axis, is shown in Fig. 7. It is clear that the performance of the proposed method is relatively insensitive to the properties of the slides in a presentation. The estimated coverage of the extracted watermark derived as Equation (1) is also plotted in the figure for comparison.

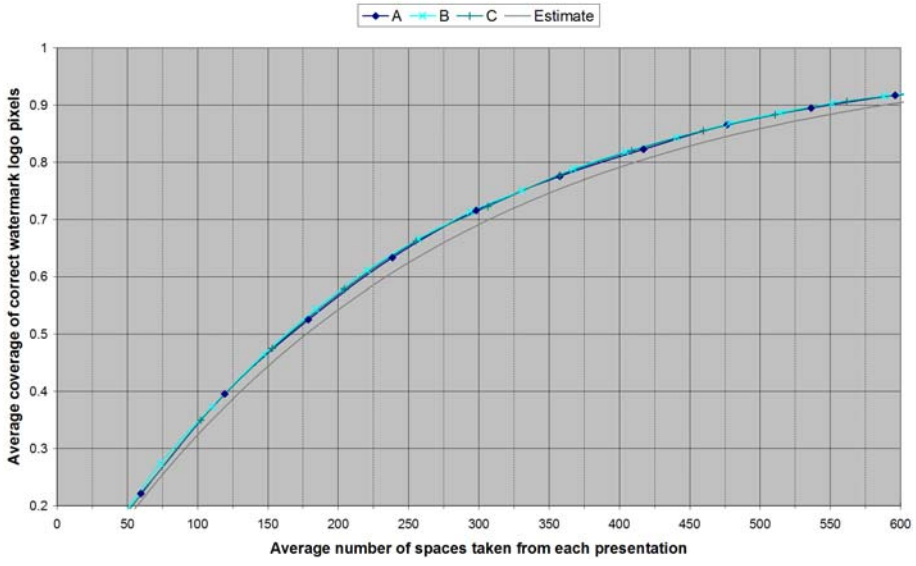


Fig. 7. Normalized plot of average correct watermark pixel extractions from presentations constructed from randomly drawn slides

To achieve a correct coverage of 50% and 80% respectively, approximately 165 and 385 spaces are required respectively according to the figure, which are close to the theoretical estimates of 178 and 412. The actual experimental results are slightly better than the theoretical estimates because we assumed that all spaces were drawn randomly for the estimate, whereas in the experiments a whole slide that contains spaces in sequence was taken at a time. As observed in the experimental results, the interferences between the different sets of watermarked slides have been eliminated by the application of the proposed weighted voting scheme.

4 Conclusions and Future Works

In this paper we described a novel method for embedding watermark images into the slides of presentations. The watermarked presentation is visually indistinguishable from the original version, and is robust against most common editing operations. Specifically, the watermarked presentation is resilient against insertions, removals, and reordering of slides, copying and pasting of slides, changes to the slide design templates, and file format conversions. Furthermore, if slides taken from multiple presentations that have been watermarked using different keys are combined into a single presentation, each of the previously embedded watermark images can be individually extracted correctly with the respective keys using the proposed method. For example, a lecturer may assemble a presentation by taking slides from presentations supplied by a book publisher and

presentations taken from online course pages. If watermarks have been embedded in the slides of the source presentations, we can extract the watermarks from the assembled slide presentation using the proposed method.

The watermark embedding and extraction methods have been tested using the popular presentation software Microsoft PowerPoint, and good experimental results demonstrate the feasibility and resilience of the proposed method. On average, five slides taken from a watermarked presentation using a certain security key is sufficient to extract a recognizable watermark of size 64×64.

Due to the robustness of the proposed method, adaptability of the proposed method for the applications of covert communication, slide authentication, or text annotation is certainly plausible, and can be pursued in future works. Other data hiding techniques appropriate for slides of presentations and other popular file formats are also good future research topics.

References

1. Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.: Information hiding – a survey. Proceedings of IEEE 87(7), 1062–1078 (1999)
2. Johnson, N.F., Duric, Z., Jajodia, S.: Information hiding: steganography and watermarking: attacks and countermeasures. Advances in information security, vol. 1, p. 137. Kluwer Academic Publishers, Dordrecht (2001)
3. Cox, I.J., Miller, M.L.: The first 50 years of electronic watermarking. EURASIP Journal on Applied Signal Processing 2002(2), 126–132 (2002)
4. Lin, C.Y., Wu, M., Bloom, J.A., Cox, I.J., Miller, M.L., Lui, Y.M.: Rotation, scale, and translation resilient watermarking for images. IEEE Trans. Image Processing 10(5), 767–782 (2001)
5. Ó Ruanaidh, J.J.K., Pun, T.: Rotation, translation and scale invariant digital image watermarking. In: IEEE Signal Processing Society 1997 International Conference on Image Processing (ICIP 1997), Santa Barbara, California (1997)
6. Brassil, J.T., Low, S., Maxemchuk, N.F.: Copyright protection for the electronic distribution of text documents. Proceedings of IEEE 87(7), 1181–1196 (1999)
7. Wu, D.C., Tsai, W.H.: A steganographic method for images by pixel-value differencing. Pattern Recognition Letters 24(9-10), 1613–1626 (2003)
8. Solachidis, V., Pitas, I.: Watermarking polygonal lines using fourier descriptors. IEEE Computer Graphics and Applications 24(3), 44–51 (2004)
9. Microsoft: Microsoft Office 1997 Visual Basic: programmer's guide. Microsoft, Redmond (1997)
10. Muterspaugh, M., Liu, H., Gao, W.: Thomson proposal outline for WRAN, proposal for IEEE P802.22 Wireless RANs, doc.: IEEE802.22-05/0096r1 (November 2005)
11. Tseng, S.S., Tsai, W.H.: High confidence information systems mid-term report, NSC advanced technologies and applications for next generation information networks (October 2002)
12. Xilinx: Digital filtering, DSP design flow (2003), `http://users.ece.gatech.edu/~hamblen/4006/xup/dsp_flow/slides/`