RESEARCH ARTICLE
doi: 10.2306/scienceasia1513-1874.2013.39.423

ScienceAsia 39 (2013): **423–435**

Grid-line watermarking: A novel method for creating a high-performance text-image watermark

Wiyada Yawai*, Nualsawat Hiransakolwong

Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Road, Ladkrabang, Bangkok 10520 Thailand

*Corresponding author, e-mail: wyd.yawai@gmail.com

Received 4 Jun 2012
Accepted 18 Jan 2013

ABSTRACT: This study aims to discover an effective method to watermark any text image in any language. The ultimate goal of this work is to generate invisible and more robust watermarks, increase the hiding capacity, and identify any change in the original text. Generally these are the limitations of most text-image watermarking methods. Using a grid of horizontal and vertical lines is one of the most effective methods to overcome these limitations. These grid lines run across text-image character-skeleton lines to finely mark and detect watermarks on these line-character intersection points; there is one intersection for one specific zero watermark pixel. Each intersection point is defined by one hiding bit of the watermark such that the more reference horizontal and vertical lines of grid pattern are plotted, the more points of intersection are obtained. This approach increases the hiding-bit capacity to watermark embedding, which is a disadvantage shared by many text-image watermarking methods. In addition, if all positions of all intersection points are collected, these points will act as the identity reference points of all original characters to verify their integrities after they are modified.

KEYWORDS: digital watermarking, text image, integrity, cross ratio, collinear points, grid lines, zero watermark

INTRODUCTION

State of problem

Currently, creating and embedding watermarks in a text image faces many limitations, particularly reduced robustness against a variety of attacks and a reduced capacity to hide a large volume of watermark data due to the nature of the content, language and text, which is mostly written in black. This approach provides a small variation space to hide the large volume of invisible watermark data, hence making the data easy to observe and destroy by simple attacks. Many researchers have proposed algorithms to hide large amounts of watermark data in a text image. However, these algorithms only apply to certain languages, only achieve a small hiding-bit data capacity and are not strong enough to survive attacks. Thus increasing the capacity for watermark hiding and robustness are the main goals of watermarking applications for storing the copyrighted data of all digital media and text documents. Unfortunately, these features are the most difficult aspects to address for black and white text images. In fact, if watermarks are made to be absolutely invisible and impervious to any attack or tampering, these difficulties will be doubled. For the latter, even though we may be able to detect our own copyright by extracting information using intact redundant watermarks, we have not necessarily obtained our original text image. In contrast, a copyright on a modified text image is achieved instead. This issue is the greatest challenge for text-image watermarking research.

Existing research

According to the various types of host media, digital watermarking may be classified into the following four categories: image watermarking, video watermarking, audio watermarking and text watermarking. The principles of image watermarking, video watermarking and audio watermarking are similar in that they make use of the redundant information of their host media, but text watermarking techniques are different from those of non-text watermarking. Moreover, text watermarking algorithms cannot easily satisfy the requirements of transparency (i.e., invisibility or imperceptibility) and robustness [1].

Most watermarking studies have concluded that text watermarking is the most difficult type of steganography, largely due to the relative lack of redundant information in a text file when compared with a picture or a sound file [2]. One reason that text steganography is difficult is that text contains little redundancy compared to other media. Another reason is that humans are sensitive to abnormal text [3].

Text files present various challenges for copyright protection. Any text transformation should preserve the meaning, fluency, grammar, writing style and value of the text. Short documents have a low capacity for watermark embedding and are relatively difficult to protect. Text watermarking algorithms are also dependent on the text size, language, rules, grammar, conventions and writing style. In the past, text watermarking methods based on text images, synonyms, pre-supposition, syntactic-trees, nouns and verbs, word and sentences, acronyms and type errors[4].

The watermarking of text falls under two domains: (1) text-image watermarking and (2) natural language watermarking. The aim of both of these watermarking systems is the embedding of information by modifying the original data in a discrete manner such that the modifications are imperceptible and the embedded information is impervious to possible attacks. In image text watermarking, this goal is achieved by exploiting the redundancy in images and the limitations of the human visual system. However, language has a discrete and syntactical nature that makes such techniques more difficult to apply[5].

Our survey revealed that most of the existing studies are focused on watermarking an electronic text or document file, rather than a text image, in one specified language, as opposed to multiple languages, and stress the watermark embedding technique over watermark robustness, not the integrity verification. These existing text watermarking studies can be categorized into three methods as follows (see Table 1).

First method: Watermark embedding with text document physical layout/pattern/structure rearrangement. Specific examples of this method include the shifting of lines[4,6,7] and words, particularly the binding of word spacing, word shift coding or word classification[8,9]. This method has some disadvantages. For example, the line shifting technique of Low et al[6] would be weakly robust for a document passing through document processing.

Second method: Embedding text watermark by text character/letter feature modification. For example, Du et al[11] proposed a text digital watermarking algorithm based on a slight change in text colour; watermarks were embedded after changing the low 4 bits of the RGB colour components of the characters. However, this study tested only the robustness of this algorithm against word deletion and modification. Brassil, et al[10] used letter adjusting by reducing or increasing the length of letters, such as increasing the lengths of the letters *b*, d, or h. The limitation of this process is that the hidden data will be weakly robust for a document passing through document processing.

Third method: Watermarking with semantic schemes or word/vowel substitution and text structure. Topkara et al[13] developed a technique for embedding secret data without changing the meaning of the text after replacing words in the text with synonyms. This method reduces the quality of a document, and a large synonym dictionary is necessary. Samphaiboon et al[3] proposed a steganographic scheme for electronic Thai plain text documents that exploits redundancies for a particular vowel in the standard Thai character set. However, this method can be used with only Thai text documents, and these watermarks are easily destroyed after re-editing with a word processing program. Meanwhile, structural schemes of text watermarking use text structures to embed watermarks without modifying any original text, so called the zero watermarking[14,15]. For instance, the existence of double letter (aa-zz) in the text is used to watermark the copyright data. However, the structural algorithms are not applicable to all types of text documents and the algorithm use an alphabetical watermark[12].

Research objectives

This study aims to discover a novel method to solve the previously stated problems and to eliminate the weakest points of the currently used methods. In doing so, the cross-ratio theory will be applied with intersections of the virtual reference grid lines and character-skeleton lines to build an effective watermark performance including robustness against possible attacks, hiding-bit data capacity, imperceptivity, multi-language text image applying and text integrity verification. Thus the five objectives of this work are the following:

(1) To find a new method to effectively embed watermarks in any text image in any language.

(2) To make watermarks absolutely invisible or difficult to detect.

(3) To obtain a higher hiding-bit data capacity.

(4) To make the watermark more robust to possible attacks.

(5) To identify any change in an original text image and verify its integrity.

MATHEMATICAL METHODS FOR TEXT WATERMARKING

The two mathematical methods applied in this study are the cross-ratio theory to build marking robustness and detect the embedded zero watermark location and the matching percentage method to verify embedded zero watermarks and the integrity of the text.

The cross ratio is a basic invariance in projective geometry (i.e., all other projective invariances can be

Table 1 Performance comparison of the existing watermarking method.

Method	Technique	Applied to	Imperceptivity	Language	Approx. Capacity	Robustness
first	Text line shifting [4,6,7]	Digital text document, text image (300–400 dpi)	Visible	Multi-language	30 bits/A4 page	Weakly robust to document processing or OCR
	Inter-word space shifting (to left - right) [8,9]	Digital text document, text image	Visible	Multi-language	720 bits/A4 page	Weakly robust to document processing or OCR
second	Point shifting of letter 'i' & 'j' [5]	Text image	Visible	English	600 bits/A4 page	Weakly robust to brightness and noise signal adding
	Adjusting character size [1]	Digital text document, text image	Visible	Chinese	650 bits/A4 page	Weakly robust to document processing or OCR
	Character peak point distinction [2]	Text image	Visible	Persian and Arabic	900 bits/A4 page	Weakly robust to brightness and noise signal adding
	Reduce/increase length of letter for feature coding [10]	Text image	Visible	English	4000 bits /A4 page	Weakly robust to document processing or OCR
	Changing the low 4 bits of RGB colour of characters [11]	Digital plain text	Invisible	Chinese	40 000 bits /A4 page	Robust to deletion modification attack etc.
third	Existence of double letter (aa–zz) in the text [12]	Digital plain text	Invisible	English	50 bits/A4 page	Not applicable to all types of text document
	Particular vowel creating sequent changing [3]	Digital plain text	Invisible	Thai	58 bits/A4 page	Very weakly robust to document processing
	Substitute words in the text by synonyms [13]	Digital plain text	Invisible	English	870 bits/A4 page	Very weakly robust to document processing

derived from it). Let A, B, C and D be four collinear points (three or more points; A, B, C, are said to be collinear if they lie on a single straight line [16]). The cross ratio is the 'double ratio' which is given by $A, B; C, D = AC.BD/BC.AD$. Based on a fundamental theory, any homography preserves the cross ratio. Thus central projection, linear scaling, skewing, rotation, and translation preserve the cross ratio [17].

The matching percentage MPW is the percentage of the amount of the watermark embedded position, which exactly matches the watermark detected position, one-to-one, based on the total amount of the watermark embedded positions:

$$\mathrm{MPW} = \frac{\sum D_{\mathrm{wm}}(x_i, y_i)}{\sum P_{\mathrm{ip}}(x_i, y_i)} \times 100\%.$$

D_{wm} is the detected watermark at position (x_i, y_i) and P_{ip} is the prominent intersection points at position (x_i, y_i) which were selected to embed the watermark.

A NOVEL TECHNIQUE FOR DIGITALLY WATERMARKING TEXT IMAGES

Three steps for arranging the zero watermarks on a text image exist. The first step is to define the zero watermark embedding positions, for hiding secret data, and its hiding capacity, with certain intersection points where either horizontal or vertical grid lines, or both of grid lines run across the text-character skeleton lines. The second step is to match the cross-ratio reference points on specific line intersection points with zero watermarking positions to easily track the zero watermarking points after a text image has been attacked. The last step is to detect the specific zero

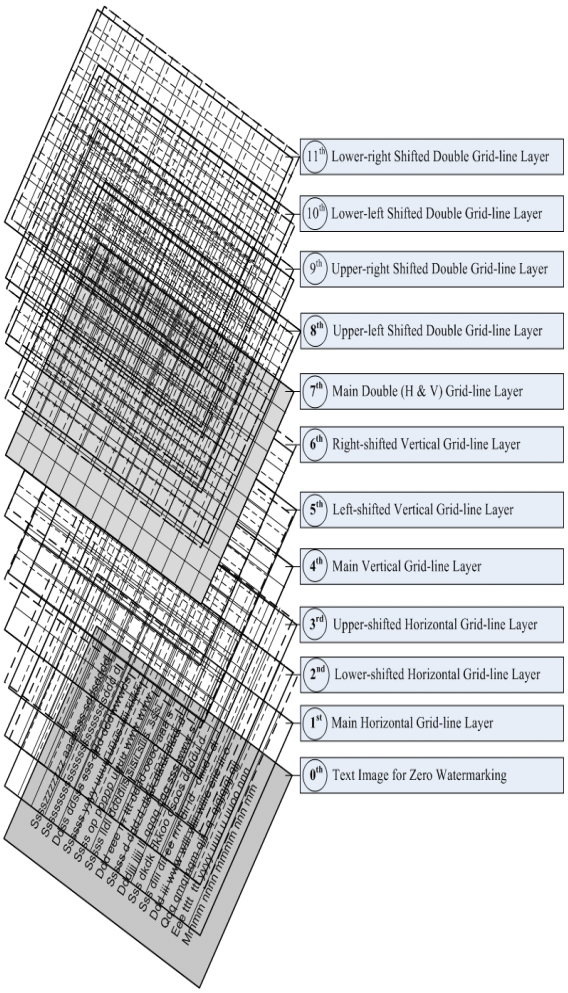

Fig. 1 Eleven grid-line intersection layers of main and 2-pixel-shifted horizontal (H), vertical (V) and double (H & V) lines for embedding a large volume of watermark secret data in each layer.

watermarked points and verify the integrity of the text with the matching percentage method.

Line C_aC_b runs from an origin point (C_a) to a destination point (C_b), where $a = 1, 2, 3, 4$ and $b = 1, 2, 3, 4$ and $Cr = ((CA/CD)/(BA/BD)) = (CA/CD)(BD/BA)$, $R = ((AC/CD)/Cr)$ and $BA = (AD \times R)/(1 + R)$. Q is the distance of $BA : AD$, which is equal to $BA : DA$.

Embedding Scheme: Create watermark embedding positions and hiding-bit capacity with multiple layers of grid-line intersections

The points for embedding zero watermarks and increasing the watermark hiding-bit capacity can be achieved by effectively created using multiple layers of grid-line intersection of the main and shifted horizontal (H), vertical (V) and double (H & V) lines that cross one another under a specific pattern, as in Fig. 1. Each layer is created by following the procedure below.

Step 1: Construct the first, second and third layers of main, lower-shifted and upper-shifted horizontal grid-line intersections by identifying and drawing the first reference horizontal lines for the top borders of the first text line on each text image page. Draw the second main, lower-shifted and upper-shifted horizontal lines next to the first main, lower-shifted and upper-shifted horizontal lines, the third and the rest until the last text line, at 3-pixel intervals, which is a suitable gap for a geometric variation buffer, as found in a previous experiment, on the first, second and third layers, respectively.

Step 2: Create the fourth, fifth and sixth layers of main, left-shifted and right-shifted vertical grid-line intersections by identifying the right and left borders of the text body on a text image, and draw the first main, left-shifted and right-shifted vertical lines along the left border of the fourth, fifth and sixth layers, respectively. Draw the second main, left-shifted and right-shifted vertical lines, the third and the rest until reaching the end of the right border, using a 3-pixel interval.

Step 3: Create the seventh, eighth, ninth, tenth and eleventh layers by drawing both the main, upper-left, upper-right, lower-left and lower-right shifted horizontal (H) and vertical (V) lines, respectively, e.g., the main double (H & V) lines, using a 3-pixel interval, on the same text-image page to create the seventh main double-line intersection layer, where both the main horizontal and vertical lines run across either text-character skeleton lines or blank areas and create intersection points for embedding watermark data bits.

Step 4: Select each prominent grid-line intersection position in each main or shifted line that runs across each text-character skeleton line and blank area, as shown in Fig. 2, for the embedding watermark data. If the tone of the grid-line intersection is less than the 245 greyscale level or is a gray/black tone, its bit value is specified as 1. If its greyscale value is higher than the 245 greyscale level or is a white tone, the bit value is specified as 0. These selected prominent white and black positions are specified as the marking points of the zero digital watermark bits for hiding secret data, and these points represent the main

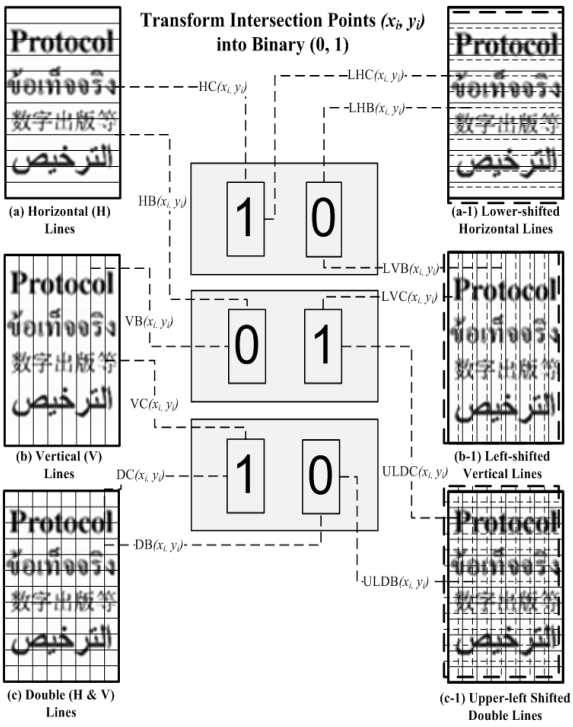

Fig. 2 The technique for transforming intersection points (x_i, y_i), running across text-character skeleton lines and blank areas with (a) horizontal (H), (b) vertical (V), (c) double (H & V), (a-1) lower-shifted horizontal, (b-1) left-shifted vertical and (c-1) upper-left shifted double H & V lines, into binary or bit, 0, 1, data to create the watermark-hiding bit capacity for each layer.

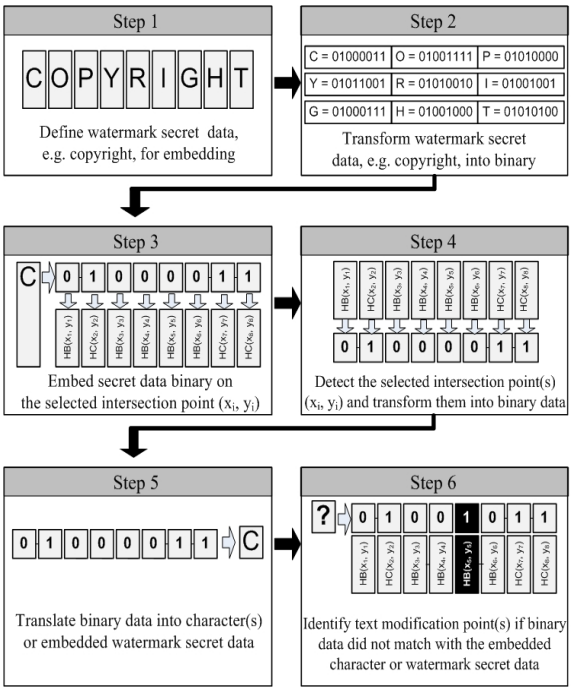

Fig. 3 The six steps for embedding and detecting watermark secret data, i.e., 'COPYRIGHT', which is transformed into binary data (0, 1) before being embedded into the selected reference line intersection points. The watermark is then again detected by checking whether the embedded intersection positions exist.

hiding-bit capacities of the first, second, third, fourth, fifth, sixth, seventh, eighth, ninth, tenth and eleventh grid-line layers.

Step 5: If more hiding-bit capacity is required in each layer, the horizontal, vertical and double (H & V) lines can be shifted to new positions, and new shifted horizontal, vertical and double (H & V) lines can be drawn to create new intersection points that run across the text-character skeleton lines and blank areas. These shifts create more intersection points at new positions to provide more hiding-bit capacity. Moreover, these new intersection points, surrounding their main original intersection points, can be used as crosscheck points to identify the exact point of origin and verify any changes to the integrity of the text image.

These 11 grid-line intersection layers are designed to generate relatively large watermark hiding-bit capacities for each text-image page. In our experiments, we tested 35 pages of Thai, English, Chinese and Arabic language scanned text images by using our MATLAB program to draw automatically grid lines and counting all grid-line intersection points in all layers, where one intersection point encodes one bit of watermark hiding data. This way, it can automatically watermark a lot of scanned text papers.

Transform the watermarked secret data into binary and embedding positions

To transform the watermarked secret data into binary and embedding positions can be done as described below.

Step 1: Define the secret data to be embedded as the watermark on each layer.

Step 2: Transform the watermarked secret data, i.e., the word COPYRIGHT, into binary, i.e., the bits 0 and 1, as illustrated in Figs. 3 and 4.

Step 3: Select the major intersection points or their locations, (x_i, y_i), corresponding to specifics bits (1 or 0) of the transformed secret data to assign the watermark embedding positions.

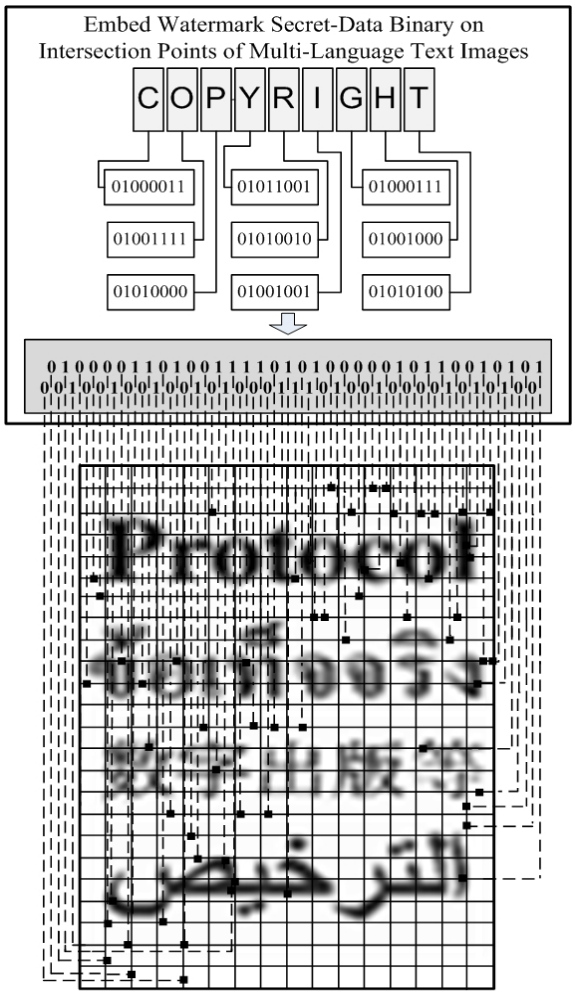

Fig. 4 The technique for embedding secret watermark data bits (0, 1) into the selected grid-line intersection positions.

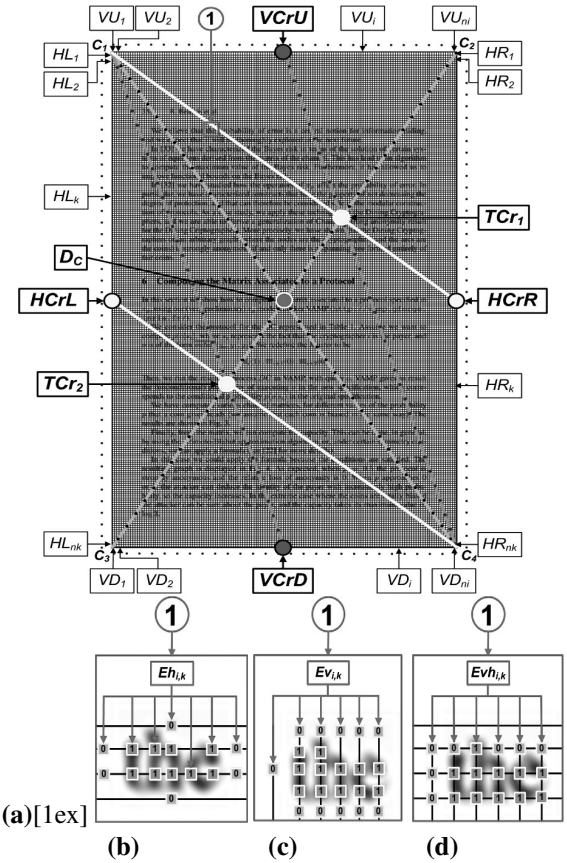

Fig. 5 (a) Notation of double reference grid-line intersection of vertical and horizontal lines on a text image. (b), (c) and (d) Notation of the horizontal, vertical and horizontal & vertical grid lines, in the upper parts, respectively, intersect text character skeleton lines and blank areas used to embed or mark zero watermark pattern bits in the text image.

Embed watermarks under cross-ratio directing

The text watermarking process shown in Figs. 3 (Step 3) and 4 is based on two subprocesses. The first one is to match the cross-ratio reference points with zero watermark marking positions to easily track the zero watermark marking points after a text image has been attacked. The second one is to detect the zero watermarks and verify the integrity of the text with the matching percentage method.

To apply the cross ratio to digital image watermarking, three reference points are required. In this section, a method for deriving such reference points is described. Let us start by considering the marking part. The method is described algorithmically, step by step, as follows.

Step 1: Predefine the set of cross-ratio values to be used in subsequent steps.

Step 2: Find the image centre using the line intersection formula[18] (two diagonal lines of the image) described by the following equations: $x_\mathrm{c} = x_\mathrm{t}/x_b$, $y_\mathrm{c} = y_\mathrm{t}/y_b$;

$$x_\mathrm{t} = \begin{vmatrix} a & x_1 - x_4 \\ b & x_3 - x_2 \end{vmatrix}, \quad x_b = \begin{vmatrix} x_1 - x_4 & y_1 - y_4 \\ x_3 - x_2 & y_3 - y_2 \end{vmatrix},$$

$$y_\mathrm{t} = \begin{vmatrix} a & y_1 - y_4 \\ b & y_3 - y_2 \end{vmatrix}, \quad y_b = \begin{vmatrix} x_1 - x_4 & y_1 - y_4 \\ x_3 - x_2 & y_3 - y_2 \end{vmatrix},$$

where

$$a = \begin{vmatrix} x_1 & y_1 \\ x_4 & y_4 \end{vmatrix}, \quad b = \begin{vmatrix} x_3 & y_3 \\ x_2 & y_2 \end{vmatrix}.$$

In addition, (x_i, y_i) is the coordinate of the point C_i, $i = 1, \ldots, 4$ (Fig. 5a). From the above equations, x_c is the x-axis value of the point D_c

of a two-line intersection; C_1C_4 intersects C_2C_3, and y_c is the y-axis value of the same point. $||$ denotes a determinant operator (Fig. 5a).

Step 3: Find each of the primary-level watermark marking points (TCr_1 and TCr_2) on the right diagonal line (Fig. 5a), using: $x_{TCr_i} = x_2 + Q(x_3 - x_2)$, $y_{TCr_i} = y_2 + Q(y_3 - y_2)$, where (x_{TCr_i}, y_{TCr_i}), $i = 1, 2$, $A = C_2$, $B = T_{Cr_i}$, $C = D_c$ and $D = C_3$. These points can be identified after using two corner points of the left diagonal line (C_2 and C_3), in combination with the image centre point D_c and the predefined cross-ratio values (C_r).

Step 4: For each pair of coordinates $((x_1, y_1)(x_2, y_2))$, find an intersection, (x_i, y_i), of the grid line of each level drawn across $(x_3, y_3)(x_4, y_4)$ by applying the following equations: $x_i = x_t/x_b$, $y_i = y_t/y_b$;

$$x_t = \begin{vmatrix} a & x_1 - x_2 \\ b & x_3 - x_4 \end{vmatrix}, \quad x_b = \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_3 - x_4 & y_3 - y_4 \end{vmatrix},$$

$$y_t = \begin{vmatrix} a & y_1 - y_2 \\ b & y_3 - y_4 \end{vmatrix}, \quad y_b = \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_3 - x_4 & y_3 - y_4 \end{vmatrix},$$

where

$$a = \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix}, \quad b = \begin{vmatrix} x_3 & y_3 \\ x_4 & y_4 \end{vmatrix}.$$

For the line C_4TCr_2, find a left-hand horizontal grid-line intersection, (x_{HCrL}, y_{HCrL}), of this line drawn across the line C_1C_3, where $(x_1, y_1) = (x_{TCr_2}, y_{TCr_2})$, $(x_2, y_2) = (x_{C_4}, y_{C_4})$, $(x_3, y_3) = (x_{C_3}, y_{C_3})$, and $(x_4, y_4) = (x_{C_1}, y_{C_1})$. For the line C_1TCr_1, find a right-hand horizontal grid-line intersection, (x_{HCrR}, y_{HCrR}), of this line drawn across the C_4C_2, where $(x_1, y_1) = (x_{C_1}, y_{C_1})$, $(x_2, y_2) = (x_{TCr_1}, y_{TCr_1})$, $(x_3, y_3) = (x_{C_4}, y_{C_4})$, $(x_4, y_4) = (x_{C_2}, y_{C_2})$.

Step 5: Find each of the secondary level watermark marking points (HL_k) on the left-hand horizontal grid line, as described by $x_{HL_k} = x_1 + Q(x_3 - x_1)$, $y_{HL_k} = y_1 + Q(y_3 - y_1)$, where $A = C_1$, $B = (x_{HL_k}, y_{HL_k})$, $C = HCrL$ and $D = C_3$. These points can be identified after using two corner points of the line C_1C_3, in combination with the image left-hand horizontal central point ($HCrL$) and the predefined cross-ratio values (C_r).

Step 6: Find each of the secondary level watermark marking points (HR_k) on the right-hand horizontal grid line, as described by $x_{HR_k} = x_2 + Q(x_4 - x_2)$, $y_{HR_k} = y_2 + Q(y_4 - y_2)$, where $A = C_2$, $B = (x_{HR_k}, y_{HR_k})$, $C = HCrR$ and $D = C_4$. These points can be identified after using two corner points of the line C_2C_4 in combination with the image right-hand horizontal central point ($HCrR$) and the predefined cross-ratio values (C_r).

Step 7: Repeat Steps 4–6 to find other secondary-level watermark marking points ($Eh_{i,k}$) at the prominent horizontal grid-line intersection positions where each horizontal grid line runs across a text-character line or blank area and its tone is lower or higher than the 245 greyscale level when more effective embedding and detection is required. These prominent positions are specified as the embedded bits, one or zero, of invisible zero watermarks ($Eh_{i,k}$) when their tones are lower than 245 or higher than 245, respectively, as indicated in Fig. 5b.

Step 8: Find each of the secondary level watermark marking points (VU_i) on the upper-part vertical grid line, as described by $x_{VU_i} = x_1 + Q(x_2 - x_1)$, $y_{VU_i} = y_1 + Q(y_2 - y_1)$, where $A = C_1$, $B = (x_{VU_i}, y_{VU_i})$, $C = VCrU$ and $D = C_2$. These points can be identified after using two corner points of the line C_1C_2 in combination with the image upper-part vertical central point ($VCrU$) and the predefined cross-ratio values (C_r).

Step 9: Find each of the secondary level watermark marking points (VD_i) on the lower-part vertical grid line, as described by $x_{VD_i} = x_3 + Q(x_4 - x_3)$, $y_{VD_i} = y_3 + Q(y_4 - y_3)$, where $A = C_3$, $B = (x_{VD_i}, y_{VD_i})$, $C = VCrD$ and $D = C_4$. These points can be identified after using two corner points of the line C_3C_4 in combination with the image lower-part vertical central point ($VCrD$) and the predefined cross-ratio values (C_r).

Step 10: For each vertical line, detect other prominent intersection positions of each vertical line that runs across each text-character skeleton line or blank area, where its tone is lower or higher than the 245 greyscale level when more effective embedding and detection is needed. These detected prominent positions are specified as the embedded bits, one or zero, of invisible zero watermarks ($Ev_{i,k}$) when their tones are lower or higher than 245, respectively, as indicated in Fig. 5c.

Step 11: For each pair (x_{VU_i}, y_{VU_i}) and (x_{VD_i}, y_{VD_i}), find an intersection, $(x_{Evh_{i,k}}, y_{Evh_{i,k}})$, of the grid line of each level drawn across (x_{HL_i}, y_{HL_i}) and (x_{HR_i}, y_{HR_i})

after applying the followings equations: $x_{Evh_{i,k}} = x_t/x_b$, $y_{Evh_{i,k}} = y_t/y_b$;

$$x_t = \begin{vmatrix} a & x_1 - x_2 \\ b & x_3 - x_4 \end{vmatrix}, \quad x_b = \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_3 - x_4 & y_3 - y_4 \end{vmatrix},$$

$$y_t = \begin{vmatrix} a & y_1 - y_2 \\ b & y_3 - y_4 \end{vmatrix}, \quad y_b = \begin{vmatrix} x_1 - x_2 & y_1 - y_2 \\ x_3 - x_4 & y_3 - y_4 \end{vmatrix},$$

where

$$a = \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix}, \quad b = \begin{vmatrix} x_3 & y_3 \\ x_4 & y_4 \end{vmatrix}.$$

For each pair (x_{VU_i}, y_{VU_i}) and (x_{VD_i}, y_{VD_i}), find an intersection, $(x_{Evh_{i,k}}, y_{Evh_{i,k}})$, of the grid line of each level drawn across (x_{HL_k}, y_{HL_k}) and (x_{HR_k}, y_{HR_k}), $(x_1, y_1) = (x_{VU_i}, y_{VU_i})$, $(x_2, y_2) = (x_{VD_i}, y_{VD_i})$, $(x_3, y_3) = (x_{HL_k}, y_{HL_k})$, $(x_4, y_4) = (x_{HR_k}, y_{HR_k})$.

Step 12: Determine the positions of each prominent intersection of the double (H & V) lines that run across each text-character line or blank area with tones lower or higher than the 245 greyscale level when more effective embedding and detection is needed. These prominent positions are specified as the embedded bits, one or zero, of invisible zero watermarks $(Evh_{i,k})$ when their tones are lower or higher than 245, respectively, as indicated in Fig. 5d.

DETECTION SCHEME

Detection of watermark bits at the embedded intersection points

To detect a watermark in a text image, first a watermarked paper has to be scanned and then its scanned text-image output would be passed through our MATLAB program to detect its four image corner points. This detection can be achieved using any existing corner detection algorithm. After detecting the four corner points, watermark marking points must be identified. Each selected intersection point used to embed a watermark data bit can be calculated using a method similar to that in the marking stage, as in Step 4 of Fig. 3. By extracting and transforming the pixel bit values corresponding to these watermark marking points, a watermark can be detected using any existing watermark detector. We adopt the basic matching percentage method described in "The matching percentage" to compare these detected watermark positions with their originally marked positions, one-by-one, until all specified positions are covered. We predefine the set of cross-ratio values to be used in subsequent steps.

Translation of watermark bits into the embedded secret data

After obtaining the extracted bit values of each layer, the next step is translating them into characters or embedded watermark secret data, as shown in Fig. 3 (Step 5), and comparing them with the original embedded secret data. These comparisons are made block-by-block or layer-by-layer by checking the matching percentage.

Verification of text image integrity and change detection

If the extracted data do not match the original embedded secret data, the unmatched bits lead us to their positions, (x_i, y_i), where we can check for modifications, i.e., new text additions, reordering or deletions, as in Fig. 3 (Step 6). This helps the copyright owners to trace any changes in their original text images. This technique can also prevent copyright owners from accepting ownership of counterfeit text images that embedding their verified secret watermark data. This situation may occur when the owners have applied a physical watermarking technique like inter-word or line shifting, which some hackers may observe and know how to manipulate by not altering the watermark embedding positions. The correctly watermarked data are still detected and extracted, although they have been modified. This false acceptance would rarely occur using this new technique because all text characters are finely marked, stored and detected, so a small bit change can eventually be identified.

EXPERIMENTS

Text images tested in multiple languages

We tested a set of 35 text images of greyscale multi-language text, including Thai, English, Chinese and Arabic, with a size of 1240×1754 pixels and a resolution of 150 dpi.

Testing the number of grid-line intersection layers for imperceptivity and hiding-bit capacity

To obtain a high hiding-bit capacity for watermarking, each page of these greyscale-text images, we first virtually created the three main grid-line intersection layers, containing the main horizontal (H), vertical (V) and double (H & V) grid-line layers, as in Fig. 1. Each layer was then derived into eight 2-pixel-shifted grid-line intersection layers, including two shifted horizontal grid-line intersection layers, two shifted vertical grid-line intersection layers and four shifted double (H & V) grid-line intersection layers. There

are 11 total grid-line intersection layers for testing and contributing the watermark hiding-bit capacity.

Watermark secret data embedding test

In this test, the prominent grid-line intersection points created on each grid-line intersection layer described above, i.e., the points where the main and shifted single horizontal, single vertical or double (H & V) lines run across the text-character skeleton lines and blank areas, were selected to embed the watermark secret data bits. One selected prominent intersection point represented one embedded data bit with cross-ratio controlling. Here, the word 'COPYRIGHT' was encoded in the secret data bits to be embedded in each grid-line intersection layer, as in Figs. 3 and 4. Here, each instance of 'COPYRIGHT' requires 72 bits: 010 000 110 100 111 101 010 000 010 110 010 101 001 001 001 001 010 001 110 100 100 001 010 100.

Testing for robustness against manipulation attacks on embedded watermark detection and integrity verification

The grid-line cross-ratio watermarking robustness against some possible attacks such as compression, sharpness, noise signal adding, shearing and rotating and three text-group manipulations, including ten text groups with addition, reordering and deletion manipulations, were used to test the embedded watermark detection and integrity verification scheme. The percentage of matching was used as the criterion for this testing.

RESULTS

The first result was from the embedding and detection testing of the controlled watermarked text images without manipulation attack. Comparing the plotted watermark positioning pattern and extracting the embedded secret data, this method obtained a 100% matching percentage and a zero error percentage. For each of the following tests, we briefly describe the result according to our objectives.

Language-independent testing

The results of this test clearly demonstrated that this multiple grid-line intersection layer technique can be applied to all Thai, English, Chinese and Arabic text images. This technique requires only the grid-line intersection points where the virtual reference lines run across text-character skeleton lines and blank areas to embed watermark data bits without physically modifying the original text images or using any special technique for any specific language. Each tested language affects only the number of intersection points created due to the different character skeleton line structures.

Imperceptivity testing

Virtual lines are drawn according to these grid lines only in the program process, and they are not drawn on each text image page. Thus it would not immediately be possible to observe the lines and watermarks after these text images were completely watermark embedded, even with many secret data bits. Conversely, the pages look exactly like their original text images before watermarking.

Hiding-bit capacity testing

The testing revealed that the 7th, 8th, 9th, 10th and 11th layers of the main and shifted double (H & V) line intersections generated a maximum hiding-bit capacity volume of 240 608 bits/layer/A4 page of text image (see Table 2), whereas the first, second, third, 4th, 5th and 6th layers of the main and shifted single horizontal and vertical line intersections generated a maximum hiding-bit capacity volume of only 77 334 bits/layer/A4 page of text image.

With respect to each language, this experiment demonstrated that Arabic text generates high hiding-bit capacities under both single horizontal and vertical grid-line intersection layers (see Table 2). The tested languages did not affect their hiding-bit capacities under the double (H & V) grid-line intersection layer, as all intersection points, whether running across text-character skeleton lines or blank areas, were counted.

The above results describe the hiding-bit capacities generated by eleven main and shifted grid-line intersection layers, as in Table 2. To count the total hiding-bit capacity per text image page using this technique, we would count all the hiding-bit capacities in all layers. Arabic script generates the maximum hiding-bit capacity per page: up to 1 545 738 bits/A4 page of text image can be hidden.

Robustness, copyright ownership and integrity verification testing

After the manipulation tests, with compression, sharpness, noise signal adding, shearing and rotating and ten text groups added, reordered and deleted, each layer could be used to detect and extract the embedded watermark and verify its integrity or modification, measured as an error percentage, as shown their higher performance results in Figs. 6, 7, 8, 9 and 10 and summarized in the robustness testing results below, comparing with the performance of the existing text-image watermarking methods summarized in Table 1.

Table 2 The hiding-bit capacity volume of each language and layer.

Layer No.	Type of grid-line intersection	Hiding-bit capacity (bits)			
		Thai	English	Chinese	Arabic
1	Main Horizontal (H)	64 278	65 125	61 614	75 329
2	Lower-shifted H.	67 244	64 327	61 541	75 802
3	Upper-shifted H.	61 829	64 457	61 893	77 334
4	Main Vertical (V)	32 234	32 449	30 829	38 053
5	Left-shifted V.	32 188	32 204	30 978	38 048
6	Right-shifted V.	32 253	32 302	30 717	38 132
7	Main Double (H & V)	240 608	240 608	240 608	240 608
8	Upper-left Shifted Double (H & V)	240 608	240 608	240 608	240 608
9	Upper-right Shifted Double (H & V)	240 608	240 608	240 608	240 608
10	Lower-left Shifted Double (H & V)	240 608	240 608	240 608	240 608
11	Lower-right Shifted Double (H & V)	240 608	240 608	240 608	240 608
	Total	1 431 237	1 493 904	1 480 612	1 545 738

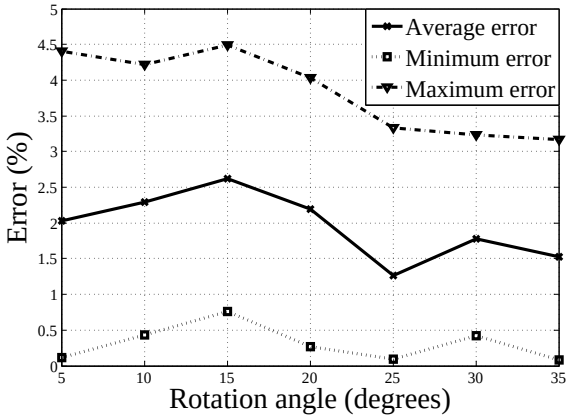

Fig. 6 Error level 4% of single horizontal reference grid-line zero watermark detection after rotation angles ranging from 5–35°.

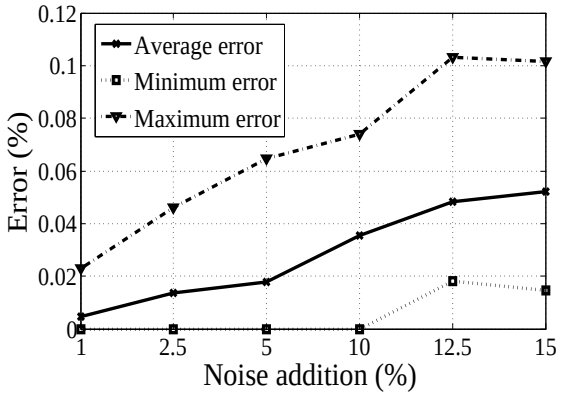

Fig. 7 Low error (%) level of single horizontal reference grid-line zero watermark detection after adding salt and pepper noise signals.

Figs. 6, 7 and the robustness testing results below show the remarkable performance of the cross-ratio theory in strengthening the robustness of text-image watermarking against some possible geometric distortion attacks, including shearing and rotating and manipulating attacks, such as noise signal adding, compression and sharpness. These attacks are the result of applying the single horizontal reference grid-line watermarking method while controlling the watermark data embedding and detecting using the cross ratio of four collinear points on each the horizontal reference grid lines that run across each text character skeleton line. The results show that these attacks did not significantly affect watermark detection. The rotating attack, for example, affected the embedded zero watermark by detecting only 0.15–4% of errors. This error indicates that the matching percentage of watermark detected is generally $> 96\%$.

The robustness testing results are as follows. Sharpness: 0–100%; low error (0.5–2%) - JPEG compression: 0–100%; low error (0–2%) - blur: 3x3 - 15x15 mask size; low error (0.083%) - contrasting: 10–50%; no error - noise signal adding:pepper; low error (0–0.11%) - shearing:x right shifted 0–5%; low error (0–2%) - rotation: 5–35°; error level: 0.15–4% - scaling: 10–130%; error level: 0–6% - text adding detection: by double (H & V) lines; error Level: 0.008–0.067% - text deleting detection: by horizontal (H) lines; error Level: 0.25–1% - text reordering detection:by horizontal (H) lines; error Level: 0.18–0.82%.

Figs. 8, 9, 10 and the robustness testing results above present the ability of horizontal and vertical grid line intersection with cross-ratio theory to strengthen

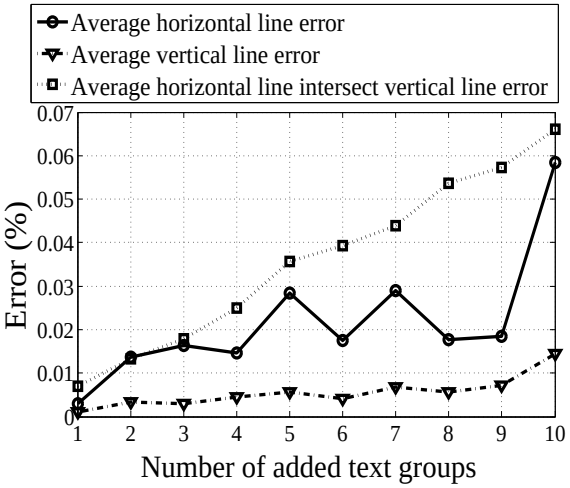

Fig. 8 The percent of errors, under the double reference grid-line intersection watermarking method and text adding manipulations attack, were more effectively detected than when using single reference grid-line watermarking.

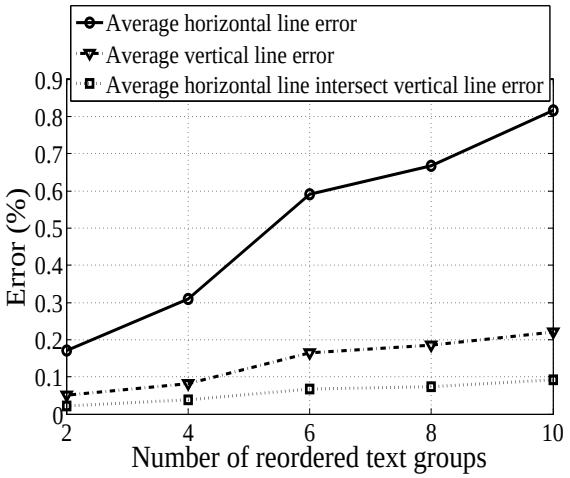

Fig. 9 The single reference grid-line intersection watermarking of the horizontal grid line was mostly suitable for detecting percent of errors after being attacked with text reordering.

the original text verification performance against three manipulations: text adding, reordering and deleting. The results indicate that the horizontal and vertical grid line intersection method with cross-ratio theory can still sensitively detect even a few changes, such as adding, reordering and deleting one to ten text groups, for which the observed maximum errors are 0.067%, 0.82% and 1% respectively.

These results show that the percent error of detection depends on the exact positions selected for zero

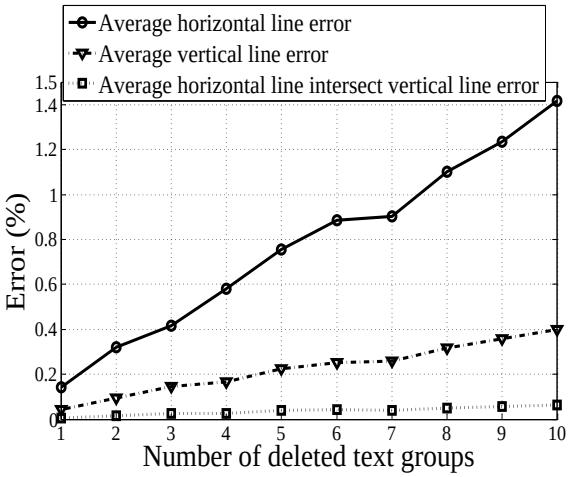

Fig. 10 The single reference grid-line intersection watermarking of the horizontal grid line was mostly suitable for detecting the percent of errors after being attacked with text deletion.

watermark embedding or data collecting. For example, the intersected positions of the double reference, vertical and horizontal grid lines, which run across or intersect one another on the blank area of a text image, could be used as blank reference points to trap any added text from the text adding attack. This is why the double reference grid lines are more effective than the single reference grid line, (Fig. 8). However, for text reordering and deleting, the horizontal single reference grid-line intersection watermarking method performs better than the other techniques (Figs. 9 and 10).

DISCUSSION

The results of all of the experiments mentioned above prove that the novel method, which applies the cross-ratio theory and grid-line intersection of the virtual horizontal and vertical grid lines that run across text-character skeleton lines and blank areas, could lead to a major breakthrough in overcoming all of the major limitations of text image watermarking (see Table 1 and the robustness testing results above). Thus all five objectives of this study have been achieved, and the following discussions have been drawn.

(1) This method can be applied to watermark any text image in any language with the grid-line intersection points of either horizontal or vertical reference grid-lines or both grid lines virtually intersected the text-character skeleton lines or blank areas, which are used as the embedding points of watermark data. This novel technique does not depend on the type of text

images used or language, even though these experiments tested for only four languages: Thai, English, Chinese and Arabic. Nevertheless, the method can be applied to other languages because this novel technique does not require any text alignment or syntactic or semantic structural modifications.

(2) This method can be applied to make watermarks absolutely invisible or unobservable because no actual colour-tone pixel of a watermark is embedded and no physical text structure is modified, as mentioned above. The intersection points used as reference embedding points of watermark data are virtual horizontal and vertical grid lines that run across all text-character skeleton lines and blank areas but cannot be seen. These grid-line intersection points can be used as the reference positions of data bits to combine them as the embedding patterns of secret watermark data, zero watermarks, and can also be used as reference points to later detect and interpret the combined secret data bits.

(3) This method is able to generate a high hiding-bit data capacity of up to 1 545 738 bits/A4 page of Arabic text image, which safely and effectively creates bit combination redundancy to embed a large amount of secret watermark data on one A4 page of the text image. This enormous hiding-bit capacity is obtained from the number of intersection points of virtual reference grid lines and text-character skeleton lines, on each layer, where one intersection point is defined as one bit of secret watermark data. Hence, an increase in the number of grid-line intersection points indicates an increase in the amount of hiding bits obtained. However, our results show that the most suitable interval between each line is 3 pixels because the method requires some clear space for buffering and tolerating variations under the attack of some geometric distortions. Nevertheless, we can still dramatically increase the hiding-bit capacity by adding more watermarking grid-line intersection plains such as 12 layers of diagonal grid-lines.

(4) This method can be applied to make watermarks more robust to or more likely to survive many possible attacks, especially geometric distortions such as scaling, shearing and rotating. Specifying the cross ratio of four collinear points for each reference horizontal and vertical grid line and each diagonal line is first set to control and direct the embedding position of each secret watermark data bit (Fig. 4), which makes it easy to refer to the embedded position, even though they may be attacked. Figs. 6, 7 and the robustness testing results above confirm that the cross ratio could generate a watermark that is reasonably robust to both geometric distortions, as mentioned earlier, and graphical manipulations, such as sharpness, compression, blur, contrast and the addition of noise signals. This robustness is due to the cross-ratio marking pattern being able to precisely lock all watermark embedding positions to easily and exactly detect and retrieve them after they have been attacked.

(5) This method can be applied to simultaneously identify any change in the original text image and verify the image's integrity during watermark detection. This performance is directly related to the increase in the hiding-bit capacity. This relationship indicates that the more hiding bits or intersection points we have, the more checkpoints for available for verifying the image's integrity. Using this technique, the horizontal and vertical grid line intersection points on the text-character skeleton lines can be used as check points for text deleting and reordering, while their grid-line intersection points on the blank area can be used as the checkpoints for text adding. From these experiments, horizontal and vertical grid-line intersections were shown to be very effective in verifying the addition of new text (Fig. 8). This result is because the intersected positions of these double reference grid lines, vertical and horizontal grid lines, run across or intersect one another in the blank area of a text image, which is used as a reference point and can trap any additional text introduced by a text adding attack and precisely pinpoint the exact positions of text additions. Meanwhile, for text reordering and deleting, a single horizontal reference grid line would perform better than a single vertical reference grid line and double reference grid lines (Figs. 9 and 10).

Thus the cross-ratio theory and grid-line intersection are able to effectively generate high text-image watermarking performance.

CONCLUSIONS

Although this study involved many experiments, many ways remain to apply and benefit from these grid-line intersections and cross-ratio theory, especially for obtaining greater hiding-bit capacity by applying further pattern grid-line intersections including diagonal, sine-curve and unique barcode lines. The real identity of the original text image is also proven by directing the watermark embedding points using the relative vector along the grid-line intersection points to more efficiently verify original text modifications and enhance the robustness of this scheme. Two double reference grid-line intersection points may be selected, and lines may be virtually drawn to connect them, storing their relative vector data to form a unique web or mesh network of virtual watermarking points, locking them using the cross-ratio format for

each text document image. This unique mesh network acts as a watermarking pattern fingerprint that can effectively be applied for both integrity verification and watermark retrieval after an attack. Our future work will focus on these advanced techniques.

Acknowledgements: This study was performed under the computer science doctoral study program of the Department of Computer Science, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, where all lecturers and officers supported us in completing this study.

REFERENCES

1. Zhang W, Zeng Z, Pu G, Zhu H (2006) Chinese Text Watermarking Based on Occlusive Components. In: *Proceedings of the 2nd Information and Communication Technologies*, pp 1850–4.
2. Shirali-Shahreza MH, Shirali-Shahreza M (2006) A New Approach to Persian/Arabic Text Steganography. In: *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science*, pp 310–5.
3. Samphaiboon N, Dailey MN (2008) Steganography in Thai text. In: *Proceedings of the 5th International Conference on Electrical Engineering/Electronics, Computer Telecommunications and Information Technology*, pp 133–6.
4. Jalil Z, Mirza AM (2009) A review of digital watermarking techniques for text documents. In: *Proceedings of the International Conference on Information and Multimedia Technology*, pp 230–4.
5. Ranganathan S, Ali AJ, Kathirvel K, Mohan KM (2010) Combined text watermarking. *Int J Comput Sci Inform Tech* **1**, 414–6.
6. Low SH, Maxemchuk NF, Brassil JT, O'Gorman L (1995) Document marking and identification using both line and word shifting. In: *Proceedings of the 14th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp 853–60.
7. Alattar AM, Alattar OM (2004) Watermarking electronic text documents containing justified paragraphs and irregular line spacing. In: *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents VI* vol 5306, pp 685–94.
8. Kim Y, Moon K, Oh I (2003) A text watermarking algorithm based on word classification and inter-word space statistics. In: *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR'03)*, pp 775–9.
9. Huang D, Yan H (2001) Interword distance changes represented by sine waves for watermarking text images. *IEEE Trans Circ Syst Video Tech* **11**, 1237–45.
10. Brassil JT, Low S, Maxemchuk NF, O'Gorman L (1995) Electronic marking and identification techniques to discourage document copying. *IEEE J Sel Area Comm* **13**, 1495–504.
11. Du M, Zhao Q (2011) Text watermarking algorithm based on human visual redundancy. *Adv Inform Sci Service Sci* **3**, 229–35.
12. Jalil Z, Mirza AM, Iqbal T (2010) A zero-watermarking algorithm for text documents using structural components. In: *Proceedings of the International Conference on Information and Emerging Technologies (ICIET 2010)*.
13. Topkara U, Topkara M, Atallah MJ (2006) The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In: *Proceedings of the 8th workshop on Multimedia and Security*, pp 164–74.
14. Jaseena KU, Anita J (2011) An invisible zero watermarking algorithm using combined image and text for protecting text documents. *Int J Comput Sci Eng* **3**, 2265–72.
15. Jalil Z, Farooq M, Zafar H, Sabir M, Ashraf E (2010) Improved zero text watermarking algorithm against meaning preservation attacks. In: *Proceedings of the World Academy of Science, Engineering and Technology*, pp 593–7.
16. Coxeter HSM, Greitzer SL (1967) Collinearity and concurrence. In: *Geometry Revisited*, Math. Assoc. Amer, pp 55–77.
17. Mohr R, Morin L (1991) Relative positioning from geometric invariants. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 139–44.
18. Antonio F (1999) Faster line segment intersection. In: *Graphics Gems III*, Academic Press.