World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:11, 2007

Entropy Based Data Hiding for Document Images

Swetha Kurup, Sridhar G., and Sridhar V.

Open Science Index, Computer and Information Engineering Vol:1, No:11, 2007 publications.waset.org/5449/pdf

Abstract**—**In this paper we present a novel technique for data hiding in binary document images. We use the concept of entropy in order to identify document specific least distortive areas throughout the binary document image. The document image is treated as any other image and the proposed method utilizes the standard document characteristics for the embedding process. Proposed method minimizes perceptual distortion due to embedding and allows watermark extraction without the requirement of any side information at the decoder end.

Keywords**—**Entropy, Steganography, Watermarking.

I. Introduction

DOCUMENT watermarking has been used extensively for copyright protection applications. Watermarks in both visible and invisible forms provide protection to document images by facilitating document usage tracking. Several approaches were proposed for inserting watermarks in document images and these approaches in general address the issue of robustness and aim to enhance the embedding capacity. Din Huang et al in [5] modified the interword spaces appearing throughout the document to represent a sine wave that is used to carry encoding information. Another method suggested by Young-Won et al in [4] groups adjacent words into segments and further groups segments into classes. Statistical distributions are then computed for the segments belonging to the same class and these distributions are in turn modified for the purpose of encoding information.

Huijuan Yang et al in [2] proposed a method wherein consecutive characters along a line are grouped into components of three characters each. The embedding location is identified with the help of a component window which is an overlapping window that includes one character from each neighboring component so that two components share a character that will not be used for shifting purposes. If shifting the center character in each component does not change the number of characters as well as the number of components, then that component is used for embedding. The Embedding process further includes shifting the characters to the left or right based on embedding rules applied to each of the embeddable components to indicate either a '0' bit or a '1' bit respectively. These and other methods [6], [7] are mainly an extension of the Line-Shift, Word-Shift methods. The method suggested later by Huijan Yang et al in [3] was however a deviation from the traditional techniques that used the space characteristics in a document. In this method, the entire image is divided into blocks of equal size and the flippability of each block is determined based on certain denoising patterns. The flippability checking is done on a 3x3 neighborhood centered on every pixel within the block excluding the boundary ones. This way the flipped pixels in every block contribute to the overall embedding capacity of the technique. In this paper we propose a novel method for watermarking binary document images which aims to reduce the distortion introduced by the insertion of watermark. We use entropy based technique [8] for detecting the suitable areas in the document image where data can be embedded with minimum distortion. We used Distance Reciprocal Distortion Ratio [1], as a measure for the distortion in binary documents. The proposed method falls under blind watermarking category and the original watermark is not required for detection of the watermark.

Manuscript received March 31, 2005.

Swetha Kurup is with the Applied Research Group, Satyam Computer Services Limited, SID Block, IISC Campus, Bangalore, India 560 012. (e-mail: Swetha_Kurup@satyam.com).

Sridhar G. is with the Applied Research Group, Satyam Computer Services Limited, SID Block, IISC Campus, Bangalore, India 560 012. (Phone: +91 80 2360 6830 Fax: +91 80 2360 6833-1016; e-mail: Sridhar_Gangadharpalli@satyam.com).

Sridhar V. is with the Applied Research Group, Satyam Computer Services Limited, Block, IISC Campus, Bangalore, India 560 012. (e-mail: Sridhar@satyam.com).

II. Entropy Based Data Hiding

Any document in general comprise of characters, words of varying font sizes along with some additional Fig.s. It can also be observed that a document has a well organized structure consisting of title, headings and sub headings and the body. Normally the titles and headings are of larger size than the body and constitute a minority of the document surface. The text body which constitutes the majority of the document surface is normally of a smaller font size. Any distortion occurring in more prominent regions of the document like the Titles or Sub-titles are more visible than distortion that may occur by data hiding in smaller font regions of the document.

Proposed watermarking method focuses on identifying the groups of characters and other regions in the image where the data can be hidden with minimum perceptible distortion to the document image. This is done as follows:

1. Identify the most frequently occurring smallest font in the document and select the regions, this corresponds to the text body and images if any as part of the body text.
2. Within these selected regions further identify specific regions at word level where the embedding causes minimum perceptual distortion.

For both the above steps we use entropy variations for detecting the start and end of a character and also the suitable

Open Science Index, Computer and Information Engineering Vol:1, No:11, 2007 publications.waset.org/5449/pdf

locations in the word for embedding the data. Entropy of an area in a binary image is given by the following:

$$H_k(a) = \sum pdf_k(i) \log(1/pdf_k(i)) \quad (1)$$

where $H_k(a)$ is the Entropy of the kth block under consideration and a is the area covered in that image block which can consist of pixels of maximum two intensity values i=0 for black pixels and i=1 for white pixels. The pdf refers to the probability density function of the white or black pixels in the kth block. We begin by considering an mxm square block from the top left corner of the image and compute the entropy for the square region covered. The sides of the square block are increased by a fixed size until we detect an entropy change in the most recently covered area. To better understand the idea of using entropy in the proposed method, consider the entropy change that occurs across a word that occurs in a line of a document image as shown in Fig. 1. The graph shows the entropy variations corresponding to the variations in pixel intensities that occur across a word. For example, in the region covering a start of a character there is generally high entropy noticed while in the region covering the inter-character space comprising more of white pixels ie where the occurrence of white pixels can be predicted, the entropy is observed to be low.

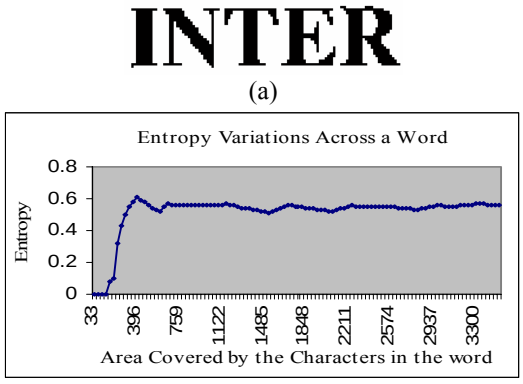


Fig. 1: Entropy variations across a word. (a)The image of a word in a document. (b) The corresponding entropy variations over the area covered by each character.

In the word considered in Fig. 1, we now consider only the first character 'I'. Observe how the entropy clearly varies across a character from Fig. 2 (a) through 2 (h) indicating the most likely start and end portions of a character and hence helping in identifying the probable font size for that character. We use these entropy variations to detect various character sizes within a given binary document image. Fig.-3 illustrates the algorithmic steps for this procedure. Entropy based detection method will not only detect comparatively bigger font sized characters throughout the document but will also identify broader and longer regions even among the smaller fonts. For example longer characters like 'l' or 'h' or broader characters like 'm' or 'w' can be identified from comparatively smaller characters like 'i' or 'o'.

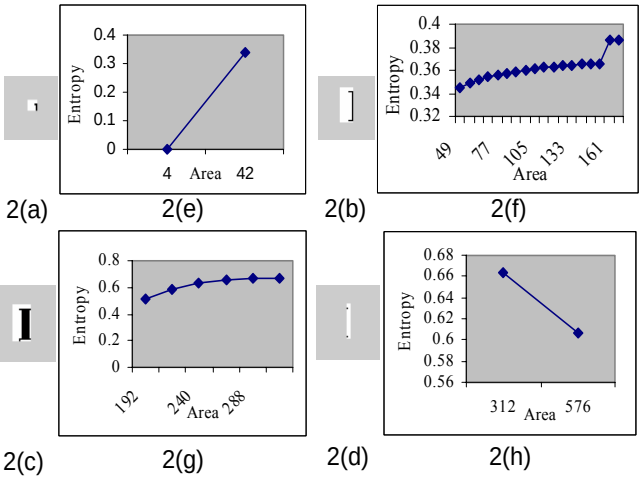


Fig. 2: Fig. (a)-(d) show regions covering parts of the character. Fig. (e)-(h) show the respective entropy variations over that area.

1. Consider a square block of size mxm starting from the top-left corner of the image.
2. Compute the entropy E_i of this square region.
3. Performing a raster scan on the image block-wise from left to right, consider the next adjacent square block of same size mxm and compute the new entropy of this region E_{i+1}
4. Continue to calculate entropy of adjacent mxm blocks until $E_{i+1} > E_i$.
5. When $E_{i+1} > E_i$
   a. Increase the area of the region by moving one pixel down to get a new region $m'_x xm$ where $m'_x = m + \Delta$ (x-direction)
   b. Compute the entropy E_{xi} of this region.
   c. Further increase the area covered in the vertical direction and compute new entropy E_{xi+1}.
   d. If $E_{xi+1} < E_{xi}$ then
      i. Increase the area of the region by moving one pixel towards the right to get a new region $m'_x xm'_y$ where $m'_y = m + \Delta$ (y-direction)
      ii. Continue to increase m'_y by Δ in a similar manner until $E_{yi+1} < E_{yi}$.
      iii. The bounds of the region covered by the above process gives the size of the character.

Fig. 3: Algorithm for detecting character sizes

The interesting feature of this type of size detection is that it is done treating the document as an image, treating every character and its glyphs as simply different figures occurring in an image contrary to the previously known methods which treat the document image Line-wise or modifying some inter-

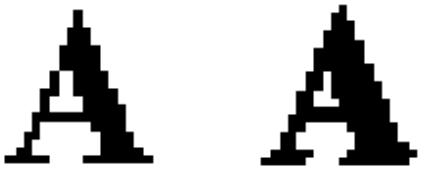
Open Science Index, Computer and Information Engineering Vol:1, No:11, 2007 publications.waset.org/5449/pdf

character space characteristics. Proposed method can be applied to images of documents that not necessarily have uniformly formatted and aligned text data.

Based on the detection results we further derive the statistics of the different font sizes that occur throughout the document. For increased data hiding capacity we further choose the most frequently occurring font size over the entire document.

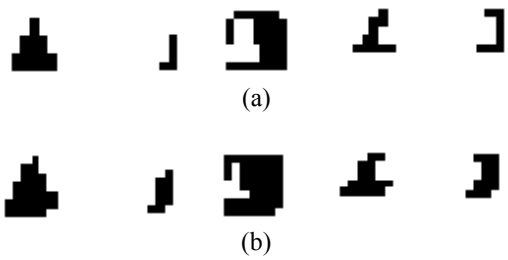
A. Watermark Embedding

In the embedding process we treat the entire document block-wise from the top-left corner to the bottom right end of the image. We simply do a raster scanning on the image block-wise from left to right. We use controlled dilation on every selected block for embedding the data. Standard dilation techniques use a certain Structuring Element that will dilate the entire image. In our method we use it on individual blocks. The overall affect is that performing it on the entire image is more visible than doing selectively on individual blocks of the image. This can be understood by the figures provided below. Consider the binary image of a character 'A' as in Fig. 3(a). Individual blocks of this image appear as in Fig. 4(a) and the result of dilating each block individually, using the usual dilation method, is as seen in Fig. 4(b). From the figures below it can be observed that the effect of dilating the entire image of the character is much more visible than the overall effect of dilating individual blocks of the same image. Dilating the entire image has lesser control than dilating blocks of the image as is evident from the Fig. below.



(a) (b)

Fig. 4: (a) Original Image. (b) Dilated Image.



(a)

(b)

Fig. 5: (a) Blocks obtained by dividing the above original image into blocks. (b) Result of dilating each block individually.

Thus, instead of performing dilation on the entire image or segments of the document image which will still be larger than the image area covered by a small block we perform dilation on each selected block on the basis of block-specific characteristics. This type of controlled dilation in turn reduces the overall distortion caused to the entire document. Again, we used Distance Reciprocal Distortion measure to compute the distortion. Algorithm for the embedding process is illustrated in Fig. 6.

Step 1: Start by considering an mxm block from the top-left corner of the document image. The block size is chosen based on the result of the pre-processing step and by choosing a size equivalent to the most commonly occurring small font size.
Step 2: Compute the entropy of this region.
Step 3: Consider the next mxm block & compute its entropy.
Step 4: Continue this process until entropy change is detected.
Step 5: Once an entropy change is detected increase the block size vertically till there is drop in entropy. This indicates the white pixels (inter-line spacing) occurring below a character.
Step 6: Once entropy drops vertically, increase the block size horizontally until entropy drops again. This indicates the white pixels to the right of the character or the inter-character or inter-word spacing.
Step 7: If in the process the size of the block becomes too large compared to the minimum size m of the block, then it indicates that the block lies in the region of larger font characters hence the block can be discarded.
Step 8: Move to the next mxm block from the previous mxm block.
Step 9: Select only those blocks that do not extend to sizes much larger than the minimum size of the block.

Fig. 6: Algorithm for detecting character sizes

This way only those blocks that lie in the region of small font characters will be chosen for embedding purposes. Selection of blocks is then followed data hiding process where each selected block is encoded to represent the embedded information.

B. Encoding

Encoding the block requires only a few computations on the pixels present within that block. We modify the selected block in such a way that the modified block does not look perceptually different from the original block. In the experiments we conducted we used block parity for encoding purposes. We considered a small text data as the watermark data and embedded it into the document image in the form of stream of bits obtained from the binary representation of the ASCII values of each of the characters present in the text data. As mentioned earlier in order to achieve the required parity for a block under consideration, we perform controlled dilation of the block, pixel-by-pixel. The amount of dilation depends upon the number of pixels of both intensities that are already present, and the amount that is required to be present in the block, for the block to be able to give precise information at the decoding end.

C. Decoding

The decoding process is similar to the data encoding process. The proposed method needs no side information in order to locate the blocks into which data has been embedded.

Open Science Index, Computer and Information Engineering Vol:1, No:11, 2007 publications.waset.org/5449/pdf

Embedded blocks are identified by processing the received document image at the decoding end to identify the document specific most commonly occurring small font sizes and then selecting the blocks which lie in the region of this relatively small font and hence identifying blocks among them into which data has been embedded. The extraction procedure involves checking the blocks that are chosen as embeddable at the encoding end and computing the block parity to identify the bit embedded into that block.

III. Experimental Results

The proposed algorithm was implemented on a document image which was obtained by scanning a text document. The document selected was such that it consisted of characters of varying font sizes like those appearing in the Titles, sub-titles, paragraphs etc.

For every block chosen for the dilation process, the structuring element is applied in such a way that it takes care of the smoothness and neighborhood properties that are characteristic of that particular block, thus avoiding the chances of embedding any stray pixels into the block which will simply add to the distortion. We first implemented the algorithm to determine the various font sizes that were present across the chosen document. The result of this gave us a set of various character sizes ranging from 6 to 24. From this we identified the most commonly occurring small font size and selected a block size of 7 for the following watermark embedding process. For embedding purpose, we performed dilation on every block using a 3x3 Structuring Element (zeros matrix) to embed more pixels into that block. The result of the embedding process is as shown in Fig. 7(a) and (b). Only a portion of the watermarked document has been shown here. The Distance Reciprocal Distortion obtained by the implementation of proposed method was very low (0.0028). We encoded a total of 88 bits by encoding the selected blocks as per their pixel intensity values.

IV. Conclusions

In this paper we propose a new approach to watermarking of document images. Proposed watermarking method focuses on identifying the groups of characters and other regions in the image where the data can be hidden with minimum perceptible distortion to the document image. This is done by :

- Identifying the most frequently occurring smallest font in the document and selecting suitable regions based on entropy variations.
- And within these selected regions further identifying specific location at character level where the embedding causes minimum perceptual distortion.

Results of applying the proposed technique on sample document images are illustrated.

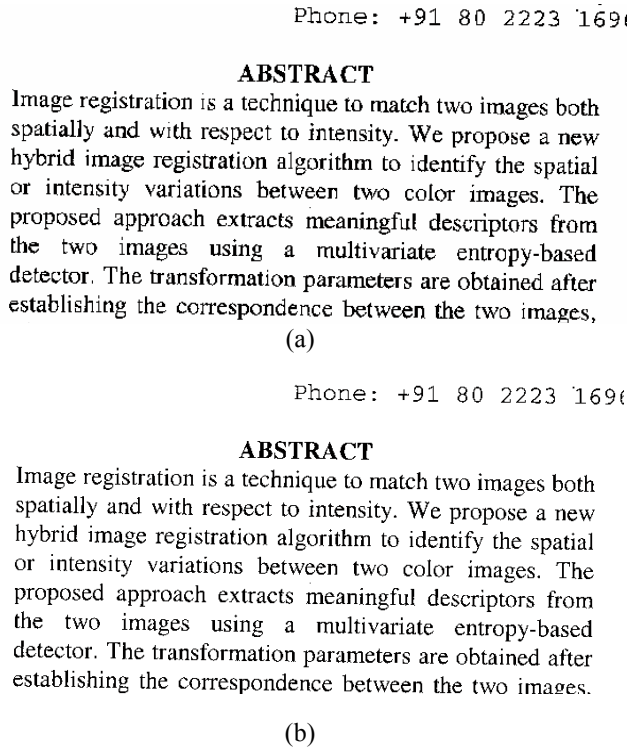


Fig. 7: Result of the proposed watermarking process. (a) Original Binary Document Image. (b) Watermarked Document Image.

References

[1] L. Haipang , Alex C. Kot, and Yun Q. Shi , "Distance-Reciprocal Distortion Measure for Binary Document Images," in *IEEE Signal Processing Letters*, Vol. 11,No. 2,February 2004.

[2] Y. Huijuan and Alex C. Kot, "Text Document Authentication By Integrating Inter Character And Word Spaces Watermarking" ", *The 2004 IEEE* International Conference on Multimedia and Expo. (ICME'2004), June 26-30, 2004.

[3] Y. Huijuan and Alex C. Kot, "Data hiding for Bi-level Documents Using Smoothing Techniques", *The 2004 IEEE* International Symposium on Circuits and Systems(ISCAS'2004), Vol. V, pp. 692-695, May 23-26, 2004.

[4] Young-Won Kim, Kyung-Ae Moon, and Il-Seok Oh, "A Text watermarking Algorithm based on Word Classification and Inter-word Space Statistics", *IEEE, Seventh International Conference on Document Analysis and Recognition Volume II,* August 03-06,2003, Edinburgh, Scotland.

[5] H. Ding and Hong Yan "Interword Distance Changes Represented by Sine Waves for Watermarking Text Images", *CirSysVideo*, Vol 11, No. 12, December 2001.

[6] K. Jonathan Su, Frank Hartung, and Bernd Girod ,"Digital Watermarking of Text, Images and Video Documents", Computer Graphics International 98, Hannover, Germany, June 1998.

[7] C. Nopporn, "Document Image Data Hiding Using Character Spacing Width Sequence Coding", in ICIP'99, pg II:250-254.

[8] S. Pavan, Sridhar Gangadharpalli, Sridhar V., "Multivariate Entropy Detector Based Hybrid Image Registration Algorithm", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 18-23, 2005 (ICASSP).