

# High Capacity Data Hiding in Binary Document Images

N.B. Puhan<sup>1</sup>, A.T.S. Ho<sup>2</sup>, and F. Sattar<sup>1</sup>

<sup>1</sup> School of Electrical and Electronic Engineering  
Nanyang Technological University, Singapore, 639798

<sup>2</sup> Department of Computing

Faculty of Engineering and Physical Sciences  
University of Surrey, Guildford, Surrey, UK

puhan@ntu.edu.sg, a.ho@surrey.ac.uk, efsattar@ntu.edu.sg

**Abstract.** In this paper, we propose a high capacity data hiding method in binary document images towards semi-fragile authentication. Achieving high capacity in binary images with strict imperceptibility criterion is found to be a difficult task. In this method, noise type pixels are selected for pixel-wise data embedding using a secret key. The data hiding process through pixel flipping introduces some background noise in watermarked images and could preserve relevant information. The reversible nature of noise pixel patterns used in flipping process enables blind detection and provides high watermark capacity illustrated in different test images. After extraction process, the background noise is removed to generate the noise-free version of the watermarked image.

## 1 Introduction

Data hiding could address important applications of multimedia security by embedding a proprietary mark which may be easily retrieved to verify about ownership and authenticity [1]. There has been a growing interest in the authentication of binary document images such as text, circuit diagrams, signature, financial and legal documents. For such images in which the pixels take on only a limited number of values, hiding significant amount of data for authentication purpose with strict imperceptibility criterion becomes more difficult.

Low *et al* [2, 3, 4] introduced robust watermarking methods for formatted document images based on imperceptible line and word shifting. The methods were applied to embed information in document images for bulk electronic publications. The line shifting method was found to have low capacity but the embedded data was robust to photocopying, scanning and printing process. The word shifting method could offer higher capacity than the line shifting method but the robustness was reduced to printing, photocopying and scanning. Brassil and O’Gorman proposed a method in [5], where the height of the bounding box enclosing a group of words could be used as a feature for embedding. This method has a better data hiding capacity than the line and word shifting methods. It was also robust to distortions caused by photocopying.

Wu and Liu hide authentication data in a binary image using a hierarchical model in which human perception was taken into consideration [6]. Distortion that occurred due to flipping of a pixel was measured by considering the change in smoothness and connectivity of a 3×3 window centered at the pixel. In a block, the total number of

black pixels is modified to be either odd or even for embedding the data bits. Shuffling was used to equalize the uneven embedding capacity over the image. Koch and Zhao [7] proposed a data hiding algorithm in which a data bit '1' is embedded if the percentage of white pixels was greater than a given threshold, and a data bit '0' is embedded if the percentage of white pixels was less than another given threshold. This algorithm was not robust to attacks and the hiding capacity was low. Mei et al modified an eight-connected boundary of a connected component for data hiding [8]. A fixed set of pairs of five-pixel long boundary patterns have been identified for embedding data. A unique property of the method is that the two patterns in each pair are dual of each other. This property allowed for blind detection of the watermark in a document image.

Amamo and Misaki proposed a feature calibration method in which text areas in an image were identified and the geometry of the bounding box of each text line was calculated in [9]. Each bounding box was divided into four partitions and grouped into two sets. The average width of the horizontal strokes of characters was modified as a feature. In [10], a new perceptual measure based on curvature-weighted distance measure was proposed towards perceptual watermarking of binary document images. Puhan and Ho [11] proposed an exact authentication algorithm using the reversible property of the perceptual measure so that the possibility of any undetected content modification is removed. The method embeds an authentication signature computed from the original image into itself after identifying an ordered set of low-distortion pixels. The parity attack found in the block-wise data hiding methods becomes infeasible due to pixel-wise embedding of the authentication signature. Fragile authentication methods for tamper localization and restoration using imperceptible watermarks have been proposed in [12, 13, 14].

The above described methods could effectively address the issue of authentication and annotation using a fragile and imperceptible watermark. However, the hiding capacity achieved using the methods are not sufficient for semi-fragile authentication, where a certain level of robustness against non-intentional signal processing is required. Due to simple pixel statistics in binary document images, it is found to be difficult for a high capacity watermark embedded with strict imperceptibility criterion. In this paper, we describe an effective method for achieving high watermark capacity with relaxed imperceptibility criterion. The paper is organized as follows: in section 2 the proposed method is discussed. In section 3, the experimental results showing high capacity in several test binary images are presented. Finally, conclusions are drawn in section 4.

## 2 Proposed Data Hiding Method

In document images, we obtain relevant information by recognizing various patterns such as symbols, lines and curves etc. These patterns are represented by connected foreground (black) pixels against a white background and they are the source of perceived information. The existing embedding methods [6, 10, 13] use perceptual models to select a subset of foreground pixels along with certain white contour pixels so that the imperceptibility criterion can be maintained. If other foreground pixels are embedded to achieve higher capacity, there will be annoying distortion in the watermarked image. Along with, the user may face difficulty in correct interpretation of the document. In the proposed method, we select two types of pixel patterns shown in

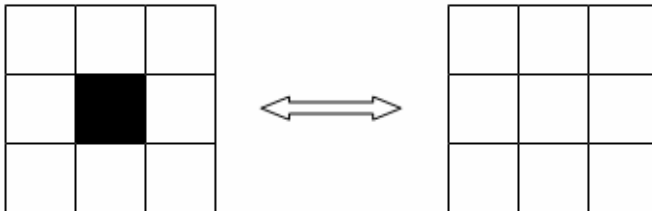
Fig. 1. The center pixels can be black or white, while other eight pixels are white. The center pixels in these patterns do not convey important information for document images. If these pixels are altered, a background noise will be formed in the image which is similar to the salt-and-pepper noise found in case of natural images. It is known that human vision has remarkable ability to recognize different structures/patterns in an image even in the presence of noise. So after embedding a watermark in these pixels, the user can still obtain relevant information about the document. Flipping of the center pixel in one pattern creates another and vice-versa; so blind detection of the embedded pixels is possible. We shall outline the proposed data hiding method in the following steps.

### Embedding

1. The original image is divided into non-overlapping blocks of  $3 \times 3$  pixels. Each such block is assigned a block index ( $I_b$ ) in a sequential order starting from left to right and top to bottom of the image.
2. If a block matches with one of the noise pixel patterns in Fig. 1, it is considered for embedding. The center pixel of such blocks is defined as noise pixel.
3. Let the set of block indices corresponding to the noise pixels be denoted as  $N$ . All block indices in  $N$  are randomly permuted using the secret key  $K$ . Let the set containing such permuted block indices be denoted as  $N_p$ .
4. A binary watermark ( $W$ ) of length  $L$  is used in embedding. The noise pixels corresponding to the first  $L$  block indices in  $N_p$  are embedded. A noise pixel is set to black if the watermark bit is 0; otherwise it is set to white.

### Detection

5. To extract the embedded binary watermark sequence from an image, steps 1 and 2 are performed at the blind detector.
6. Similar to embedding, the set containing the permuted block indices ( $N_p^d$ ) is detected using the secret key  $K$ .
7. The noise pixels corresponding to the first  $L$  block indices in  $N_p^d$  is extracted. The watermark bit is extracted as 0 if the noise pixel is black. Otherwise it is detected as 1.
8. All noise pixels are set to white for generating the noise-free version of the watermarked image for further use and analysis.



**Fig. 1.** Noise pixel patterns used in the proposed method; flipping of the center pixel in one pattern creates another

### 3 Results and Discussion

In this section, we present simulation results by embedding a binary watermark using the proposed data hiding method. The original image is shown in Fig. 2. The total number of noise pixels (i.e. the maximum capacity) in the original image is found to be 19366. The watermarked images after embedding with watermarks of different length  $L$  are shown in Fig. 3. From the figures, it is evident that a large number of watermark bits can be embedded without destroying the document information. In each case, the watermark is extracted correctly at the blind detector. The embedded data is secure and can not be extracted correctly by an adversary without using the secret key. When all noise pixels are embedded, the visual quality could get significantly affected, as shown in Fig. 3(i). Thus, a fraction of maximum capacity should be considered during embedding as a trade-off between capacity and visual quality.

To demonstrate the efficacy of the proposed method, we find data hiding capacity in several test document images (Fig. 4). The results are presented in Table 1. For each test image, it is found that a large number of bits can be embedded using the noise pixels. Both during embedding and detection, each  $3 \times 3$  pixel pattern need to be matched with two noise pixel patterns. The computational complexity of the proposed method is lower than the perceptual based methods. Since simple noise patterns are employed instead of perceptual modeling, the computational complexity of the proposed method is significantly low. We have implemented the proposed method in Matlab 7.1 software and executed them on a 2.66 GHz PC running Windows XP and

#### 2. Introduction

Image authentication using steganography is quite different from authentication using cryptography. In cryptographic authentication, the intention is to protect the communication channel and make sure that the message received is authentic. It is typically done by appending the image hash (image digest) to the image and encrypting the result. Once the image is decrypted and stored on the hard disk, its integrity is not protected anymore. Steganography offers an interesting alternative to image integrity and authenticity problem. Because the image data is typically very redundant, it is possible to slightly modify the image so that we can later check with the right key if the image has been modified and identify the modified portions. The integrity verification data is embedded in the image rather than appended to it. If the image is tampered with, the embedded information will be modified thus enabling us to identify the modifications.

**Fig. 2.** Original image of  $463 \times 535$  pixels

## 2. Introduction

Image authentication using steganography is quite different from authentication using cryptography. In cryptographic authentication, the intention is to protect the communication channel and make sure that the message received is authentic. It is typically done by appending the image hash (image digest) to the image and encrypting the result. Once the image is decrypted and stored on the hard disk, its integrity is not protected anymore. Steganography offers an interesting alternative to image integrity and authenticity problem. Because the image data is typically very redundant, it is possible to slightly modify the image so that we can later check with the right key if the image has been modified and identify the modified portions. The integrity verification data is embedded in the image rather than appended to it. If the image is tampered with, the embedded information will be modified thus enabling us to identify the modifications.

(a)

## 2. Introduction

Image authentication using steganography is quite different from authentication using cryptography. In cryptographic authentication, the intention is to protect the communication channel and make sure that the message received is authentic. It is typically done by appending the image hash (image digest) to the image and encrypting the result. Once the image is decrypted and stored on the hard disk, its integrity is not protected anymore. Steganography offers an interesting alternative to image integrity and authenticity problem. Because the image data is typically very redundant, it is possible to slightly modify the image so that we can later check with the right key if the image has been modified and identify the modified portions. The integrity verification data is embedded in the image rather than appended to it. If the image is tampered with, the embedded information will be modified thus enabling us to identify the modifications.

(b)

**Fig. 3.** The watermarked images after embedding with watermarks of different  $L$ ; (a)  $L=1000$ , (b)  $L=2000$ , (c)  $L=3000$ , (d)  $L=4000$ , (e)  $L=5000$ , (f)  $L=6000$ , (g)  $L=8000$ , (h)  $L=10000$ , (i)  $L=19366$

## 2. Introduction

Image authentication using steganography is quite different from authentication using cryptography. In cryptographic authentication, the intention is to protect the communication channel and make sure that the message received is authentic. It is typically done by appending the image hash (image digest) to the image and encrypting the result. Once the image is decrypted and stored on the hard disk, its integrity is not protected anymore. Steganography offers an interesting alternative to image integrity and authenticity problem. Because the image data is typically very redundant, it is possible to slightly modify the image so that we can later check with the right key if the image has been modified and identify the modified portions. The integrity verification data is embedded in the image rather than appended to it. If the image is tampered with, the embedded information will be modified thus enabling us to identify the modifications.

(c)

## 2. Introduction

Image authentication using steganography is quite different from authentication using cryptography. In cryptographic authentication, the intention is to protect the communication channel and make sure that the message received is authentic. It is typically done by appending the image hash (image digest) to the image and encrypting the result. Once the image is decrypted and stored on the hard disk, its integrity is not protected anymore. Steganography offers an interesting alternative to image integrity and authenticity problem. Because the image data is typically very redundant, it is possible to slightly modify the image so that we can later check with the right key if the image has been modified and identify the modified portions. The integrity verification data is embedded in the image rather than appended to it. If the image is tampered with, the embedded information will be modified thus enabling us to identify the modifications.

(d)

**Fig. 3.** (continued)

## 2. Introduction

Image authentication using steganography is quite different from authentication using cryptography. In cryptographic authentication, the intention is to protect the communication channel and make sure that the message received is authentic. It is typically done by appending the image hash (image digest) to the image and encrypting the result. Once the image is decrypted and stored on the hard disk, its integrity is not protected anymore. Steganography offers an interesting alternative to image integrity and authenticity problem. Because the image data is typically very redundant, it is possible to slightly modify the image so that we can later check with the right key if the image has been modified and identify the modified portions. The integrity verification data is embedded in the image rather than appended to it. If the image is tampered with, the embedded information will be modified thus enabling us to identify the modifications.

(e)

## 2. Introduction

Image authentication using steganography is quite different from authentication using cryptography. In cryptographic authentication, the intention is to protect the communication channel and make sure that the message received is authentic. It is typically done by appending the image hash (image digest) to the image and encrypting the result. Once the image is decrypted and stored on the hard disk, its integrity is not protected anymore. Steganography offers an interesting alternative to image integrity and authenticity problem. Because the image data is typically very redundant, it is possible to slightly modify the image so that we can later check with the right key if the image has been modified and identify the modified portions. The integrity verification data is embedded in the image rather than appended to it. If the image is tampered with, the embedded information will be modified thus enabling us to identify the modifications.

(f)

Fig. 3. (continued)

## 2. Introduction

Image authentication using steganography is quite different from authentication using cryptography. In cryptographic authentication, the intention is to protect the communication channel and make sure that the message received is authentic. It is typically done by appending the image hash (image digest) to the image and encrypting the result. Once the image is decrypted and stored on the hard disk, its integrity is not protected anymore. Steganography offers an interesting alternative to image integrity and authenticity problem. Because the image data is typically very redundant, it is possible to slightly modify the image so that we can later check with the right key if the image has been modified and identify the modified portions. The integrity verification data is embedded in the image rather than appended to it. If the image is tampered with, the embedded information will be modified thus enabling us to identify the modifications.

(g)

## 2. Introduction

Image authentication using steganography is quite different from authentication using cryptography. In cryptographic authentication, the intention is to protect the communication channel and make sure that the message received is authentic. It is typically done by appending the image hash (image digest) to the image and encrypting the result. Once the image is decrypted and stored on the hard disk, its integrity is not protected anymore. Steganography offers an interesting alternative to image integrity and authenticity problem. Because the image data is typically very redundant, it is possible to slightly modify the image so that we can later check with the right key if the image has been modified and identify the modified portions. The integrity verification data is embedded in the image rather than appended to it. If the image is tampered with, the embedded information will be modified thus enabling us to identify the modifications.

(h)

**Fig. 3.** (continued)

## 2. Introduction

Image authentication using steganography is quite different from authentication using cryptography. In cryptographic authentication, the intention is to protect the communication channel and make sure that the message received is authentic. It is typically done by appending the image hash (image digest) to the image and encrypting the result. Once the image is decrypted and stored on the hard disk, its integrity is not protected anymore. Steganography offers an interesting alternative to image integrity and authenticity problem. Because the image data is typically very redundant, it is possible to slightly modify the image so that we can later check with the right key if the image has been modified and identify the modified portions. The integrity verification data is embedded in the image rather than appended to it. If the image is tampered with, the embedded information will be modified thus enabling us to identify the modifications.

(i)

Fig. 3. (continued)

with Pentium 4 processor and 2 GB RAM. It is found that the proposed method requires 2s approximately, for both embedding and detection. The availability of a large number of noise pixels will enable to design an effective semi-fragile authentication technique. We are currently designing such a new technique exploiting the large data hiding capacity offered by the proposed method.

any change made to any bit plane will be detected. The localization properties of this simple scheme can be improved if it is applied to image blocks rather than the whole image.

One of the first fragile watermarking techniques proposed for detection of image tampering was based on inserting check-sums of gray levels determined from the seven most significant bits into the least significant bits (LSB) of pseudo-randomly selected pixels [1]. In this paper, we are going to describe one possible implementation of this idea. First, we choose a large number  $N$  that will be used for calculating the check sums. Its size directly influences the probability of making a change that might go undetected. The image is then divided into  $8 \times 8$  blocks, and in each block, a different pseudo-random walk through all 64 pixels is generated. Let us denote the pixels as  $p_1, p_2, \dots, p_{64}$ . We also generate 64 integers  $a_1, a_2, \dots, a_{64}$  comparable in size to  $N$ . The check sum  $S$  is calculated as

There are some obvious advantages of this approach. First, the logo itself can carry some useful visual information about the image or its creator. It can also represent a particular authentication device or software. Second, by comparing the original logo with the recovered one, one can visually inspect the integrity of the image. Third, the authentication watermark is embedded not only in the LSBs of the image but somewhat deeper ( $\pm 5$  gray scales). This makes it more secure and harder to remove. Fourth, the method is fast, simple, and amenable to fast and cheap hardware implementation. This makes it very appealing for still image authentication in digital cameras.

This method, however, has a serious security gap if the same logo and key are reused for multiple images. Given two images  $I_1$  and  $I_2$  with gray levels  $g^{(1)}$  and  $g^{(2)}$  watermarked with the same key and logo  $L$ , we have

$$f_g^{(1)}(i, j) = L(i, j) = f_g^{(2)}(i, j) \text{ for all } (i, j).$$

Fig. 4. Test document images

The authors apply this technique to small 8×8 pixel blocks. The block is DCT transformed, and the frequency masking values  $M(i,j)$  for each frequency bin  $P(i,j)$  are calculated using a frequency masking model. The values  $M(i,j)$  are the maximal changes that do not introduce perceptible distortions. The DCT coefficients are modified to  $P'_s(i,j)$  according to the following expression

$$P'_s(i,j) = M(i,j) \{ \lfloor P(i,j) / M(i,j) \rfloor + r(i,j) \text{sign}(P(i,j)) \},$$

where  $r(i,j)$  is a key-dependent noise signal in the interval (0,1), and  $\lfloor x \rfloor$  rounds  $x$  towards zero. Since  $|P(i,j) - P'_s(i,j)| \leq M(i,j)$ , the modifications to DCT coefficients are imperceptible.

For a test image block with DCT coefficients  $P_s(i,j)$ , the masking values  $M(i,j)$  are calculated. The error at  $(i,j)$  is estimated by the following equation

$$e' = P'_s - M' \{ r \text{sign}(P'_s) + \lfloor P'_s / M' - (r-1/2) \text{sign}(P'_s) \rfloor \},$$

$$\mathcal{Q}_{\Delta}(f) = 0 \text{ if } \lfloor f/(\Delta\sigma^2) \rfloor \text{ is even,}$$

$$\mathcal{Q}_{\Delta}(f) = 1 \text{ if } \lfloor f/(\Delta\sigma^2) \rfloor \text{ is odd}$$

at the quantization level  $l$ . If a wavelet coefficient  $f_{\lambda_l}(m,n)$  is chosen for watermark embedding, it is modified so that

$$\mathcal{Q}_{\Delta}(f_{\lambda_l}(m,n)) = w(i) \text{ XOR } qkey(m,n),$$

where  $w(i)$  is the  $i$ -th watermark bit and  $qkey$  is a bit generated from the image and a secret key. The construction of the quantization function  $\mathcal{Q}_{\Delta}$  guarantees that one will never have to modify the coefficient at the level  $l$  by more than  $\pm\Delta\sigma^2$ . The watermark is extracted by evaluating the expression

$$w(i) = \mathcal{Q}_{\Delta}(f'_{\lambda_l}(m,n)) \text{ XOR } qkey(m,n),$$

where  $f'$  is the wavelet coefficient of the potentially tampered image. The extent of tampering is evaluated using the number of correctly recovered watermark bits  $w(i)$ . The

$$V_{\text{cov}} \left[ \frac{m_{kx}}{N} \right] = \frac{1}{N} \cdot \frac{\binom{r}{n-r} \binom{S-r}{n}}{\binom{S}{n}} + \left( 1 - \frac{1}{N} \right) \frac{\binom{r}{n} \binom{S-2r}{n-2r}}{\binom{S}{n}} - \left[ \frac{\binom{r}{n} \binom{S-r}{n}}{\binom{S}{n}} \right]^2. \quad (4)$$

For the signature image of fig. 5, we have

block size	$q = 16 \times 16$ .
image size	$S = 288 \times 48$ .
block number	$N = \frac{S}{q} = 18 \times 3$ .
flippable percentage	$p = 5.15\%$ .

The analytic results are shown in Fig. 8, along with the simulation results from 1000 random shuffles. The statistics of blocks with no or few flippables are also shown in Table I. The analysis and simulation are seen to agree well, and the percentage of blocks with no or few flippables is extremely low. Error correction coding can be used to handle a very small number of blocks that have no flippable pixels. As illustrated by the block

**Step-2) Compute Flippability Score**

The smoothness and connectivity measures are passed into a decision module to produce a flippability score. Main considerations when designing this module are 1) whether the original pattern is smooth, 2) whether flipping will increase nonsmoothness by a large amount, and 3) whether flipping will cause any change in connectivity. The changes on these patterns are generally more noticeable. Listed here are the rules used by our decision module.

- 1) The lowest score (i.e., not flippable) is assigned to uniformly white or black regions as well as to the isolated single white or black pixels. These trivial cases are handled first.
- 2) If the number of transitions along horizontal or vertical direction is zero (i.e., the pattern is smooth and regularly structured), assign zero as a final score for the current

REFERENCES

- [1] M. Wu, E. Tang, and B. Liu, "Data hiding in digital binary images," in *IEEE Int. Conf. Multimedia & Expo (ICME'00)*, New York, 2000.
- [2] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—a survey," *Proc. IEEE*, vol. 87, pp. 1062-1078, July 1999.
- [3] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. IEEE*, vol. 87, pp. 1079-1107, July 1999.
- [4] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. San Mateo, CA: Morgan Kaufmann, 2001.
- [5] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, pp. 1673-1687, Dec. 1997.
- [6] C. Podilchuk and W. Zeng, "Image adaptive watermarking using visual models," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 525-538, May 1998.
- [7] M. Wu, H. Yu, and B. Liu, "Data hiding in images and videos: Part II—Designs and applications," *IEEE Trans. Image Processing*, vol. 12, pp. 696-705, June 2003.

The use of a suitable perceptual model is necessary to minimize the visual distortion in the marked images, because minor modification to the pixels can be perceptible since the pixels are either black or white. In this paper, a new perceptual model is proposed for binary images that is useful for data hiding applications. In our model, the distortion that occurs after flipping a pixel is estimated on the novel curvature-weighted distance difference (CWDD) measure between two contour segments. Through subjective tests the perceptual measure is validated and highly correlated with human perception.

Fig. 4. (continued)

proposed robust data hiding methods in formatted document images based on imperceptible line and word shifting. Their methods were applied to embed information in text images for bulk electronic publications. The line shifting method has low data hiding capacity as compared to the word shifting method but the embedded data is more robust to photocopying, scanning and printing process. Koch and Zhao [3] proposed a data hiding algorithm in which a data bit '1' is embedded if the percentage of white pixels was greater than a given threshold, and a data bit '0' is embedded if the percentage of white pixels was less than another given threshold. In [4], the proposed algorithm slightly modified interword spaces so that different lines across a text act as sampling points of a sine wave. After the modification, the

distortion measure (DRDM) for binary document images [6] that could be used for performance comparison in data hiding applications. Traditional objective distortion measures like mean square error (MSE), signal-to-noise ratio (SNR), and peak signal-to-noise ratio (PSNR) are not well correlated with human perception for binary images. All three measures only take the number of flipped pixels into account and the distortions in the binary images can be different even if the number of flipped pixels is the same.

The paper is organized as follows: Section 2 presents a contour- based metric for the proposed perceptual model. The

The purpose of the subjective test is to validate the high correlation between the model and the subjective ratings for different values of  $CWDD$  measure. Using the Adobe Photoshop software, four characters 'A', 'B', 'E', 'S' are converted to binary images of size 128x128 pixels. These binary images, as shown in Fig. 1 are used as the original image set from which all test images are produced for the subjective experiments. It is difficult to produce a test image for each value of the  $CWDD$  measure due to an insufficient number of flipping pixels of one particular value. To overcome this difficulty, we use the technique called *binning* to produce the test images. We divide the  $CWDD$  range from [0, 8] into 9 bins. With the exception of the first one, each bin is of unity length in terms of

Here we show high correlation between the perceptual model described in Section 2 with the subjective test values obtained in Section 3. The subjective mean opinion score (MOS) was computed for each test image from the data obtained in the subjective experiments. In the plot of subjective MOS versus  $CWDD_{min}$  shown in Fig. 3, 26 points out of 34 are within the 95% confidence interval. The performance attributes Spearman rank-order correlation coefficients and correlation coefficients [8] are computed between subjective MOS and  $CWDD_{min}$  for all the test cases.



## 6. REFERENCES

- [1] S. H. Low, N. F. Maxemchuk, and A. M. Lapone, "Document identification for copyright protection using centroid detection," *IEEE Trans. on Communication*, vol. 46, no. 3, March 1998, pp. 372-383.
- [2] S. H. Low, and N. F. Maxemchuk, "Performance comparison of two text marking methods," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, May 1998.
- [3] E. Koch, J. Zhao, "Embedding robust labels into images for copyright protection," *Proc. International Congress on Intellectual Property Rights for Specialized Information, Knowledge &*

Digital watermarking is the art of protecting the multimedia data by inserting the proprietary mark which may be easily retrieved by the owner of the data to verify about its ownership or authenticity. A variety of digital watermarking methods have been developed for such purposes [1, 2]. For certain applications, watermarks for checking the authenticity of the multimedia data should be fragile because any corruption to watermarked data easily destroy the watermark and so the detection algorithm will be able to verify the integrity of the data being tested. Provable security of digital media can be guaranteed through the use of cryptographic signatures as the fragile watermark. Cryptographic signature has been well studied in cryptography and algorithms such as DSA, RSA, and MD5 are extensively used in various authentication applications [3]. In authentication watermarking, the advantage of having the cryptographic signature hidden inside the digital data rather than appended to it is obvious. Lossless format conversion of the watermarked data does not render it inauthentic though the representation of the data is changed. Another advantage is that if the authentication information is localized, it is then possible to

Fig. 4. (continued)

**Table 1.** Data hiding capacity in test document images

Image number	Image size	Maximum capacity
1	438×519	17313
2	444×510	17289
3	462×510	19346
4	513×543	25038
5	495×549	24388
6	426×534	17852
7	549×798	37811
8	456×459	16099
9	579×474	20596
10	480×462	17001
11	561×462	19610
12	609×480	24297
13	603×495	25072
14	369×690	18720

## 4 Conclusion

In this paper, we proposed a data hiding method that could identify a large number of noise pixels in binary document images with blind detection. The proposed method creates watermarked images with some background noise and the noise can be erased after extraction process. The extracted image differs from the original image in positions where an original noise pixel was black. In fact, such black noise pixels occur rarely in a document image and converting them to white does not impact much on the information content. The proposed method is of low computational complexity and its large data hiding capacity will be useful for designing an effective and practical semi-fragile authentication method.

## References

1. Cox, I.J., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann Publishers Inc., San Francisco (2001)
2. Low, S.H., Maxemchuk, N.F., Lapone, A.M.: Document Identification for Copyright Protection Using Centroid Detection. *IEEE Trans. on Communication* 46(3), 372–383 (1998)
3. Low, S.H., Maxemchuk, N.F.: Performance Comparison of Two Text Marking Methods. *IEEE Journal on Selected Areas in Communications* 16(4), 561–572 (1998)
4. Brassil, J.T., Low, S., Maxemchuk, N.F.: Copyright Protection for the Electronic Distribution of Text Documents. *Proc. of the IEEE* 87(7), 1181–1196 (1999)
5. Brassil, J., O’Gorman, L.: Watermarking Document Images with Bounding Box Expansion. In: Anderson, R. (ed.) *IH 1996. LNCS*, vol. 1174, pp. 227–235. Springer, Heidelberg (1996)

6. Wu, M., Liu, B.: Data hiding in binary image for authentication and annotation. *IEEE Transactions on Multimedia* 6(4), 528–538 (2004)
7. Koch, E., Zhao, J.: Embedding Robust Labels into Images for Copyright Protection. In: *Proc. International Congress on Intellectual Property Rights for Specialized Information, Knowledge & New Technologies*, Vienna (1995)
8. Mei, Q., Wong, E.K., Memon, N.: Data Hiding in Binary Text Documents. In: *Proc. SPIE Security and Watermarking of Multimedia Contents III*, San Jose (2001)
9. Amamo, T., Misaki, D.: Feature Calibration Method for Watermarking of Document Images. In: *Proc. 5th Int'l Conf on Document Analysis and Recognition*, Bangalore, India, pp. 91–94 (1999)
10. Ho, A.T.S., Puhan, N.B., Marziliano, P., Makur, A., Guan, Y.L.: Perception Based Binary Image Watermarking. In: *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, Vancouver, Canada, May 2004, vol. 2, pp. 37–40 (2004)
11. Puhan, N.B., Ho, A.T.S.: Secure Exact Authentication in Binary Document Images. In: *Proc. IET Intl. Conference on Visual Information Engineering (VIE)*, Bangalore, India, September 2006, pp. 29–34 (2006)
12. Kim, H.Y., de Queiroz, R.L.: Alteration-Locating Authentication Watermarking for Binary Images. In: Kalker, T., Cox, I., Ro, Y.M. (eds.) *IWDW 2003*. LNCS, vol. 2939. Springer, Heidelberg (2004)
13. Yang, H., Alex, C.K.: Binary Image Authentication with Tampering Localization By Embedding Cryptographic Signature and Block Identifier. *IEEE Signal Processing Letters* 13(12), 741–744 (2006)
14. Makur, A.: Self-embedding and Restoration Algorithms for Document Watermark. In: *Proc. IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 2, pp. 1133–1136 (2005)