数理解析研究所講究録
990 巻 1997 年 145-154

Mathematical Techniques for Image Marking

小林　メイ (Mei Kobayashi), IBM Tokyo Research Laboratory
& Dept. of Mathematical Sciences, Univ. of Tokyo
e-mail: mei@trlvm.vnet.ibm.com, http://lettuce.ms.u-tokyo.ac.jp

1. Introduction

Marking of documents using wooden, stone, and ivory stamps has been used for authentication throughout the ages [12], and its use remains prevalent in many Asian countries. A modern adaptation of the practice is the embedding of official seals into the structure of a document (e.g., stationery, certificates, and currencies) as a watermark. More recently, with the advent of the Internet and the World Wide Web, another new twist is being added to the repertoire, the development of analogous methods to label, authenticate, and protect digitized information. Currently, most digital watermarking work focuses on the four media: image, audio, video and text. Since the nature of the media are very different, few of the techniques can be successfully applied to more than one of the data types. In this paper, we will examine marking of digitized, still image data. In the next section we discuss some examples of visible watermarking. Next, some very primitive and fragile transparent methods are described, as well as more sophisticated approaches. In the final section, we present quantitative methods for their benchmarking and evaluation.

2. Visible watermarking

An example of very simple and crude visible marking is the labeling of digitized photo images in a fixed location with the date. More sophisticated, attractive, and robust visible marking methods for enhancing digital documents have been developed by Braudway, Magerlein, and Mintzer. Their method for altering pixel values in a still image was used to mark digitized pages of manuscripts from the Vatican's archive with a logo, in part for use in authenticating the images, and in part for deterring any parties seeking to "purloin or misappropriate" the documents [6].

To be attractive and effective when applied to digitized still image data representing works with artistic merit, according to Braudway et al., a visible watermark must: be obvious to any person with normal or corrected vision, including the color blind, be flexible enough that it can be made as obtrusive or unobtrusive as desired, have bold features that, by themselves, form a recognizable image, allow all features of the unmarked image to appear in the marked image, and be very difficult, if not impossible, to remove.

The method designed by Braudway et al. to fulfill these criteria begins with the construction of a mask corresponding to the watermark. The mask determines which pixels in an image will remain unchanged and which will have their brightness altered. The mask is then re-sized, if necessary, to dimensions appropriate for the image size and marking purpose, and the location at which the watermark will be placed is chosen. Finally, the brightness in the pixels specified by the mask is altered. The scientists used a mathematical model of the brightness in an image:

$$\tilde{Y}_{m,n} = Y_{m,n} + C \times \Delta L^*,$$

where $Y_{m,n}$ and $\tilde{Y}_{m,n}$ represent the brightness of the $(m,n)^{\text{th}}$ pixel in the original and marked images, respectively, the constant C is a function that reflects various properties of the specific image and watermark mask, and L^* is the brightness, i.e., the amount of light received by the eye, regardless of color [26]. The appearance or obtrusiveness of the watermark is controlled by varying the brightness L^*. If the same value of ΔL^* were used to alter all the pixels that fall under the mask, then the watermark could be easily removed by a hostile party. To render robustness to the mark, randomness is introduced by using $2R_{m,n}\Delta L^*$ in place of ΔL^*, where $R_{m,n} \in [0,1]$ is a discrete random variable that, if truly randomly distributed, satisfies:

$$\lim_{M\to\infty} \lim_{N\to\infty} \frac{2}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} R_{m,n}\, \Delta L^* = \Delta L^* \ .$$

A watermark needs to have bold features because the introduction of the random variable $R_{m,n}$, depending on its values, can make fine details of the mark less discernible. As an addendum to their method, Braudway et al. remark that additional robustness can be achieved by introducing small random variations in the size as well as in the horizontal and vertical placement of the watermark, as suggested by Pickerell and Child [21].

3. Transparent watermarking

Visible and transparent marking techniques for digital still images are, for the most part, distant cousins. Although some ideas can be used in both, they are by and large answers to different technical problems and are intended for scenarios with different sets of requirements. Fragile and easily detectable transparent, or steganographic, methods can be used to label digitized images for book keeping purposes and more robust methods to protect an embedded message.

A simple example of the former application is least significant bit (LSB) coding. The message embedding process begins with all of the LSBs of the host image set to 0 (or all to 1). 1's (or 0's) are then used to embed information in the LSB plane, e.g., a pattern or image in which 0 represents black and 1 represents white, or words coded in binary form. An analogous procedure can be used for color images, which are represented by three matrices for the intensities of the colors (e.g., red, green, and blue) in the image.

Because each matrix can be used for coding, three times as much space is available for embedding information.

An attractive feature of LSB coding is its introduction of noise of at most one unit. In practice, this noise is imperceptible, so long as the host signal is not extremely low or weak. The creators of *Stego* [16], LSB-based steganographic freeware for image marking, note that, unless led to believe otherwise, most people viewing digital still image data would not be inclined to check for information embedded in the LSB plane; LSB coding, then, unlike visible stamping methods, does offer some degree of privacy for a use life unofficial office memos.

Another use of LSB coding is the placement of markers to detect enlargements or reductions of an image that may have taken place during photo editing and to recover the associated dilation factor. Transparent cross marks are embedded in the LSB plane at fixed intervals in both the horizontal and vertical directions prior to editing. Changes in the dimensions made during editing can be detected and quantitatively measured by comparing the distances between the cross marks before and after the edit. If cropping of an image is also expected, horizontal and vertical line numbers can be embedded at fixed intervals in the LSB plane to keep track of the pixel indices from which a crop is made. The pixel index information will remain with the cropped image and can be recovered without a copy of the original, full-size image. Alternative LSB marking methods for embedding pixel index information include the use of curvature information from two sets of concentric circles embedded in the LSB plane [20].

These examples illustrate how a simple tool, LSB coding, primitive as it is, can and should be used in contexts that do not require more sophisticated approaches. The advantages are, however, counterbalanced by shortcomings of different degrees of severity. The simplicity of the basic idea and the ease of its implementation render the method more susceptible to detection than more sophisticated methods. Of greater concern is the quality of the transmission lines or the possibility of any kind of contamination with low-level noise: The LSB-coded information is highly sensitive to any signal modification, and anything short of perfect integrity of the data will lead to imperfect recovery of the embedded information, or, worse yet, misleading information.

To alert the user to contamination or tampering of LSB-coded data, Walton suggests using check sums [31], e.g., the parity checkbit x_{n+1}, to insure the integrity of a string $x_1\ x_2\ x_3 \ldots x_n$, defined as: $x_{n+1} \equiv x_1 + x_2 + x_3 + \ldots + x_n \pmod 2$, and the ISBN check digit for a book number represented by the string $y_1\ y_2\ y_3 \ldots y_{10}$, defined as: ISBN check digit $\equiv \sum_{i=1}^{9} i\ y_i \equiv 0$ in base 11, with X representing the integer 10. These and further examples, along with more advanced references, are given in [22].

Walton also suggests that a pseudorandom number generator be used to generate a random walk on the image pixel plane to be used for selecting embedding locations. After a user-specified number of steps, say N, a check digit for the pixel values at the N preceding positions is embedded in the $(N+1)^{\text{st}}$ pixel along the random walk. This procedure is repeated many times. The path of the random walk should not cross over itself during the embedding of the checksums, since it could lead to false alarms of tampering. If the possible discovery of the pseudorandom sequence generation mechanism by a hostile party is a consideration, variations that disguise the locations of the checksums have been developed to prevent tampering with the checksums themselves. For color images, the basic check sum scheme can be straightforwardly applied three times to the three color planes. More interesting variations that take advantage of the three dimensions from the three color planes can be developed. The basis set for representing the images, for example, can be changed from RGB (red-green-blue) to HLS (hue-lightness-saturation); the checksum is then calculated in the new coordinate system, and the check sum digit is encoded in the original coordinate system. Details on standard bases for color image representation and conversion factors are given in [23].

Among the early modern works on digital steganography, similar in spirit to LSB coding is a series of papers by Matsui and his colleagues which are surveyed in [17]. The scientists suggest that imperfections of the human visual system (HVS) be exploited to transparently mark images. Most data we perceive is contaminated with some degree of noise, i.e.,

$$\text{data} = \text{structure} + \text{noise},$$

where *structure* represents meaningful information, such as an image or text on paper. But the HVS helps us process data so that we do not notice and, consequently, are not distracted by, the noise. Image marking can be made virtually transparent by disguising messages as minimally distracting noise. Matsui et al. proposed an embedding scheme for each of the following media: gray scale, dithered binary, facsimile and color still images and video. The schemes are not robust enough for general distribution, e.g., use on the World Wide Web, because they embed binary sequences in a manner which requires perfect preservation of the signal for successful extraction of the hidden message; noisy transmission, filtering, cropping, color space conversion, or re-sampling would destroy the message. Nevertheless, they may be used in limited contexts. As such, we briefly describe the first three below.

The first embedding scheme is for digitized gray-scale image data which consists of a set of integers between 0 and 255, representing the gray levels of an image at sampled points. The digitized image data $\{x_i\}$; $i \in N$ is converted to a sequence in which the first element is x_1, and subsequent elements are the differences between successive points, i.e., $\Delta_i = x_i - x_{i-1}$. Next, the person(s) embedding and extracting the message agree on the use of a particular cipher key table which assigns a value c_i, either 0 or 1, to each Δ_i

(see Table 1). To embed a binary sequence $B = \{b_i : b_i = 0 \text{ or } 1\}$; $i \in N$, look up the value of c_i corresponding to Δ_i in the table. If $c_i = b_i$, then keep Δ_i as is. If $c_i \neq b_i$, go to the nearest Δ_j such that $c_j = b_i$ and substitute Δ_j in place of Δ_i. The error introduced into the image data during the i^{th} step is $\text{error}_i = \Delta_j - \Delta_i$, which is usually on the same order as noise, i.e., negligible. The hidden message can be retrieved by looking up the value for c_i corresponding to Δ_i.

Table 1: Example of Cipher Key Table

Δ_i	...	-4	-3	-2	-1	0	1	2	3	4	...
c_i	...	0	1	1	0	1	0	0	1	0	...

A second message hiding scheme is for images produced by a process known as digital halftoning or spatial dithering, which uses changes in the relative spacings between black marks on paper to create the illusion of continuous gray scale tones [2],[29]. Dithering a monotone image amounts to deciding whether to turn on or off a black mark at each pixel, according to a user specified threshold T, for the brightness level. Matsui and Tanaka's scheme uses ordered dithering. An image is divided into 4-by-4 pixel blocks and brightness thresholds $\{x_i\}$; $i = 0, 1, \ldots, 15$ for each of the sixteen pixels in the block, are assigned from top to bottom, left to right. Next, define sets

$$S_k = \{(x_i, x_j)_k \ : \ x_j - x_i = k\} \ ; \ i, j = 0, 1, ..., 15; \ i \neq j.$$

Let $(y_i, y_j)_k$ be a pair of output signals which pass through $(x_i, x_j)_k$. Then $(y_i, y_j)_k$ is either $(0,0)$, $(1,0)$, $(0,1)$, or $(1,1)$, where 0 indicates that the the pixel will be turned "off" and 1 for "on". Only the pairs $(1,0)$ or $(0,1)$ will be used to embed a sequence of bits $B = \{b_n : b_n = 0 \text{ or } 1\}$; $n \in N$. To embed $b_n = 0$, set $(y_i, y_j)_k = (0,1)$. To embed $b_n = 1$, set $(y_i, y_j)_k = (1,0)$. To decode, disregard the $(0,0)$ and $(1,1)$ outputs and simply reverse the procedure described above.

Facsimile document signals serve as the host medium for a third message hiding scheme. Documents are digitized following the international standard facsimile scanning rate of 8.23 pixels/mm in the horizontal direction [7],[11]. The scanned data indicates whether a pixel is black or white, the two options. The message embedding scheme is based on the fact that the data will be compressed using *run length coding* (RLC) and modified *Huffman* coding schemes. Huffman coding is one of several encoding schemes which are based on probability of occurrence; characters which occur frequently are assigned short codes, and those which occur infrequently, long codes. RLC reduces data by replacing repeated characters with three characters: a *flag* character to signal compression follows, the repeated character, and the number of repetitions. A binary message $B = \{b_n : b_n = 0 \text{ or } 1\}$; $n \in Z$ is embedded by shortening or lengthening runs by one pixel at the boundaries of the runs. Matsui and Tanaka suggest a simple illustrative in which runs are set to be even number length when $b_i = 0$ (by leaving it as is if it is already of even length and lengthening it by one if it is odd) and to an odd length when $b_i = 1$ (by leaving it as is if it is already of odd length and by shortening it by one if it is even). Runs used for coding must have length greater than two.

All three of Matsui and Tanaka's schemes described above are easy to defeat, as is, because recovery of the embedded message depends on perfect preservation of the data. A more serious problem than erasure of marks is the possibility of extraction of a binary message by hostile parties. To circumvent the latter problem, embedding locations can be controlled through the use of keys, a very primitive example of which is a seed $s(i)$, for generating a pseudorandom sequence, e.g.,

$$\begin{aligned} s(i+1) &= (16807.0 \times s(i)) \bmod (2^{31}-1) \\ a(i+1) &= s(i+1)/(2^{31}-1) \end{aligned}$$

[10]. More advanced and thorough discussions of keys and pseudorandom number generators can be found in [13],[14],[25],[27]. Other marking schemes which extend and enhance Matsui and Tanaka's ideas have been developed. van Schnydel, Tirkel, and Osborne use M-sequences to encode marks in images [30], and their scheme is potentially compatible with JPEG compression. Their work has been further extended and enhanced by Wolfgang and Delp [32] to detect locations where tampering (by even one pixel) has occurred. Aura [3] introduces the term *cover bit* for those bits which will be used for marking and notes that LSBs are one of the simplest and most commonly used examples. To minimize the danger of algorithm or code breaking by hostile parties, Aura recommends the use of a private key to generate pseudorandom permutations of the cover bits which will determine the order in which message bits will be encoded. In particular, a highly secure pseudorandom permutation generator of Luby and Rackoff [15] is recommended, and re-use of a cover and cover sequence is not.

In a related approach, Sanford II, Bradley, and Handel use least significant differences between data values (rather than LSBs) of a palette colored image for marking [24]. Great care has been taken to ensure the security of the key and hidden information, and entropy and statistical properties of the data are little affected so that hostile parties would have difficulty detecting that embedding has taken place.

4. Benchmarking

Recently, as the use of the Internet and the World Wide Web has soared, many businessmen and scientists are beginning to recognize the potentially enormous impact of a robust marking technology suitable for open networks and the yet untapped market.

The scientists appear to be taking one of two divergent paths: some post their marking tools, manuals and methods on their web pages and publish in scientific journals, while most have chosen a more potentially lucrative approach, i.e., to patent ideas and to protect future licensing and product development opportunities, so that secrecy and ambiguity surround the details of their scientific work. To learn about or to test the wares up for licensing, interested parties must either take technological claims on faith, sign a non-disclosure agreement, or obtain reliable benchmarking data.

The development and evaluation of watermarking methods, as with most valuable technologies, is highly dependent on an associated application scenario. Requirements which should be considered to some extent for evaluating image marking techniques include: (1) the appearance of the watermark: imperceptible, almost imperceptible or acceptable with slight degradation in image (e.g., changes in hue, sharpness, edge and feature degradation), and visible; (2) probability of detection by statistical and other analytical means; (3) integrity and robustness of watermark with respect to cropping, rotations, re-sampling, filtering, noise, digital-analog (D/A) followed by analog-digital (A/D) conversion, decoding, and general tampering ;removal, destruction); (4) data type to be embedded (e.g., text, binary code, picture, logo) and volume; (5) speed of encoding/decoding; (6) costs associated with embedding (e.g., human, computer, and other operational costs); (7) multiple watermark options, and robustness to marking history (i.e., the ability to determine order in which watermarks were embedded); (8) type of network environment: open or closed; (9) additional service requirements (e.g., registration or payment center).

Tewfik, Swanson, et al. [18] note that combinations of these requirements (e.g., robustness with respect to cropping, multiple watermarking, noise) are not independent when pirates attempt to defeat a watermark. For instance, pirates may manipulate a signal to render a watermark undetectable or show that the marking scheme is unreliable because it is liable to set off false alarms, i.e. detects a mark when there is none. They propose two criteria for quantitatively evaluating the robustness of a watermark: measure the associated percentages of false alarms, correct detections, and retrieval of embedded messages [5],[28]. To design a transparent watermark which meets the above criteria, the scientists took advantage of perceptual masking, i.e., hiding of the embedded data by perceptually more prominent signals. Results from evaluation tests to show robustness to lossy JPEG coding, noise, A/D-D/A conversions, signal re-sampling, and filtering are excellent. The underlying ideas from perceptually masked image marking can be extended to audio, video, and other marking environments. Cox et al. have also taken a perceptual masking approach [8], however, the details of their work differ from those of Tewfik et al., such as the volume of the embedded data and the frequencies used for embedding.

Bender notes that most of the so-called invisible watermarking technologies for images fall under the category of *imperceptible, added noise* and that other approaches to embedding should be considered since the *added-noise* approach inevitably must compete with compression schemes; good compression schemes are based on the finding and ridding of pockets of imperceptible noise. He suggests two other promising approaches: (1) enhancement of fundamental and important features, i.e. features which good compression algorithms would not alter, and (2) use of slight changes in object placement or object characteristics [4].

As an example of the first approach, Bender cites work with his colleagues on digital watermarking of audio signals [19]. Dubbed, "*echo hiding*", the technique enhances natural echoes which are produced in vocal cords during the course of human speech, and are indiscernible to the human ear. The temporal delay of an artificially added echo can be used as a parameter for encoding a message. For example, we can label a speech signal by considering segments which correspond to tenths or hundredths of a second. To encode the coded message: 2, 7, 6, echoes with a 2, 7 and 6 temporal unit delay can be added into the first through third segments According to Bender, it is not clear whether a suitable counterpart to echo hiding exists in the still image or video domain.

Bender uses an example from video coding to illustrate the second approach. Movies consist of scenes which have many objects. A valuable character might be watermarked by introducing subtle alterations in visible, but not particularly notable objects, such as a button or shoelace. The movements or changes of the objects in the temporal domain can be used as a coding scheme. Since a considerable amount of video piracy is carried out through the use of hidden, portable video cameras in movie theaters, signal analysis-based watermarking approaches may be too subtle to be useful; problems such as cropping, changes in hue, sharpness, and slight tilting of image are likely to be introduced during illicit filming. In particular, robustness with respect to very slight tilting or shaking of the camera is a very difficult problem to solve using signal analysis-based techniques.

Some researchers suggest that marking of digital signals alone is not enough to deter piracy. They suggest a concurrent registering of the marked images at a legally acknowledged copyright database center. Zhao and his colleagues [33] propose a copyrighting scheme [35] and framework [34] for implementing this idea. Adobe has included a watermarking feature in its latest version of an image processing toolkit [1] with the option of registering a watermarked image in a database [9].

Acknowledgements The author gratefully acknowledges helpful conversations with Walter Bender, Ahmed Tewfik, the editors fo SIAM News, and the members of IBM's Digital Library Team

References

[1] Adobe, Photoshop 4.0, (http://www.adobe.com).

[2] D. Anastassiou, K. Pennington, "Digital Halftoning of Images", *IBM J. Res. Dev.*, vol. 26, no. 6, pp. 687–697, Nov. 1982.

[3] T. Aura, "Invisible communication", preprint, Helsinki Univ. of Technology, Nov. 6, 1995.

[4] W. Bender, *private communication*, Tokyo, Nov. 1996.

[5] L. Booney , A. Tewfik, K. Hamdy, "Digital Watermarks for Audio Signals", *Proc. of Multimedia '96*, IEEE Press, Piscataway, N.J., pp. 473–480.

[6] G. Braudway, K. Magerlein, F. Mintzer, "Protecting publically-available images with a visible image watermark", *IBM Research Report* TC-20336(89918), Jan 15, 1996, (http://www.software.ibm.com/is/dig-lib/vatican.html).

[7] CCITT Recommendation T.6, "Facsimilie coding schemes and coding control functions for group 4 facsimilie apparatus for document transmission", 1984.

[8] I. Cox, J. Kilian, T. Leighton, T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia", *NEC Research Technical Report*, 95-10, 1995, (http://www.neci.nj.nec.com).

[9] Digimarc, PictureMarc in Photoshop 4.0, (http://www.digimarc.com).

[10] IBM publications *XL Fortran Compiler/6000, Language Reference*, ver. 3, release 1, Random Number (Harvest), p. 423, Dec. 1993.

[11] Infomedia, Inc., Scanning FAQ, (http://www.infomedia.net/scan/).

[12] D. Kahn, *The Codebreakers*, MacMillan, NY, 1967.

[13] D. Knuth, *The Art of Computer Programming, vol. 2*, second ed. Addison-Wesley, Menlo Park, CA, 1981.

[14] M. Luby, *Pseudorandomness and Cryptographic Applications*, Princeton Univ. Press, Princeton, NJ, 1996.

[15] M. Luby, C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions", *SIAM Journal of Computing*, vol. 17, no. 2, pp. 373–326, Apr. 1988.

[16] R. Machado, *Stego*, (http://www.fqa.com/romana/romanasoft/stego.html).

[17] K. Matsui, K Tanaka, "Video-Steganography: how to embed a signature in a picture", *IMA Intellectual Property Proceedings*, vol. 1, issue 1, pp. 187–206, Jan. 1994.

[18] Multiscale Signal Processing Group, University of Minnesota, Dept. of Electrical Engineering, (http://ee.ee.umn.edu/groups/msp/).

[19] News in the Future Group, MIT Media Lab, (http://nif.www.media.mit.edu/DataHiding/index.html).

[20] M. Numao, S. Shimizu, N. Morimoto, M. Kobayashi, "Data hiding methods and data extraction methods", *IBM Japan Patent JA8-96-0107*, June 20, 1996, (in Japanese).

[21] J. Pickerell, A. Child, "Marketing Photography in the Digital Enviroment", *DiSC* (100 Frederick Ave., Rockville, MD 20850 USA), 1994.

[22] K. Rosen, *Elementary Number Theory and its Applications*, 3rd edition, Addison Wesley, Tokyo, 1992.

[23] J. Russ, *The Image Processing Handbook*, second ed., CRC Press, Tokyo, 1995.

[24] M. Sanford II, J. Bradley, T. Handel, "The data embedding method", *Los Alamos National Laboratory Report* 9LA-95-2246UR, Sept. 25, 1995,
(http://www.lanl.gov/users/u078743/embed1.htm).

[25] B. Schneier, *Applied Cryptography*, second ed., John Wiley & Sons, NY, 1996.

[26] M. Sid-Ahmed, *Image Processing*, McGraw Hill, Tokyo, 1995.

[27] D. Stinson, *Cryptography: theory and practice*, CRC Press, Tokyo, 1995.

[28] M. Swanson, B. Zhu, A. Tewfik, "Transparent Robust Image Watermarking", *Proc. of the 1996 IEEE Int'l. Conf. on Image Processing* , Lausanne, Switzerland, Sept. 1996, IEEE Press, Piscataway, N.J..

[29] R. Ulichney, *Digital Halftoning*, MIT Press, Cambridge, MA, 1987.

[30] R. van Schnydel, A. Tirkel, C. Osborne, "A digital watermark", pp.86 -90, *Proc. of ICAASP*, vol. 2, IEEE Press, Piscataway, NJ, 1994.

[31] S. Walton, "Image authentication for a slippery new age", *Dr. Dobb's Journal*, pp. 18–26, 82–87, Apr. 1995.

[32] R. Wolfgang, E. Delp, "A watermark for digital images", *Proc. of ICIP*, IEEE Press, Piscataway, NJ, Sept. 1996.

[33] J. Zhao, Fraunhofer Institute for Computer Graphics,
(http://www.igd.fhg.de/ zhao/zhao.html).

[34] J. Zhao, "A WWW Service to Embed and Prove Digital Copyright Watermarks", *Proc. European Conf. on Multimedia Applications, Services and Techniques*, Louvin La-Neuve, Belgium, May 1996.

[35] J. Zhao, E. Koch "Embedding Robust Labels into Images for Copyright Protection", *Proc. Int'l. Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*, Vienna, Aug. 1995.