PROPOSAL OF ASSOCIATIVE WATERMARKING METHOD

Ryoto Kanegae and Masaki Kawamura
Yamaguchi University, Yamaguchi, Japan
E-mail: kawamura@sci.yamaguchi-u.ac.jp Tel: +81-83-933-5701

***Abstract*—We propose a novel type of zero-watermarking method that incorporates associative memory models. Because the zero-watermarking method does not embed a watermark directly in an image, it avoids degrading the original image. However, as a watermark is associated with each image, it is difficult to manage the associations when dealing with a large number of images. Moreover, the conventional zero-watermarking method cannot correct the watermark when an image is degraded, and the watermark's length must be equal to that of the feature vector extracted from an image. Hence, we propose a novel management scheme that uses both a hetero-associative memory model and an auto-associative memory model. The proposed method can manage a large number of mappings between images and watermarks via the hetero-associative memory model, while also eliminating the watermark length restriction. Furthermore, even if an image is heavily degraded, the auto-associative memory model can correct watermark errors. The proposed method was evaluated in the case of JPEG compression, and we found that it can sufficiently reduce errors in watermarking.**

I. Introduction

In recent years, social networking services (SNSs) have rapidly spread in our daily lives. Images and videos are frequently uploaded by SNS users and can easily be downloaded. As many SNS users do not fully understand copyrights, they often accidentally violate other people's copyrights. The digital watermarking method is an effective solution to this problem.

Digital watermarking is a technology for embedding copyrights and IDs in digital content as watermarks. It is used to protect the copyrights of digital content and to inhibit tampering and unauthorized copying. Watermarks are often embedded in luminance values, discrete Fourier transforms (DFT), or discrete cosine transformed (DCT) domains [1]–[4]. Because these methods embed the watermark directly in the image, they cause a considerable amount of image distortion.

The zero-watermarking method [5]–[7] generates a secret key by extracting unique features from an image and multiplying them by a watermark. As this method does not directly embed the watermark in the image's pixels, it does not cause distortion. However, the zero-watermarking method has no error correction function; thus, if the original image is attacked by JPEG compression, noise addition, or filtering, watermark errors will occur. In addition, because a watermark is associated with each image, it is necessary to manage these associations separately for a large number of images, which requires more effort to manage the associations between images and secret keys.

In a hetero-associative memory model [8], [9], the associations between key patterns and associative patterns are stored, and when a key pattern $\boldsymbol{\eta}$ is given, the corresponding associative pattern $\boldsymbol{\xi}$ is recalled. Accordingly, a hetero-associative memory model can be adopted for the zero-watermarking scheme by using the features extracted from images as key patterns and the watermarks as associative patterns. In addition, because multiple associations can be stored in the weight matrix of an associative memory model, all keys can be managed together. Furthermore, even if there is an error in a feature, it can be corrected by the hetero-associative memory model.

On the other hand, an auto-associative memory model stores multiple associative patterns and recalls the closest stored pattern when a similar pattern is given [10]–[12]. In other words, this memory model has error correction capability because it can retrieve the original pattern from a pattern with errors. Through application of an auto-associative memory model to the watermarking method, the errors remaining in a watermark can be completely removed.

In this paper, we propose a novel type of zero-watermarking method that uses both hetero-associative and auto-associative memory models. We call the proposed method the associative watermarking method. Compared with the conventional zero-watermarking method, the proposed method can handle a large number of images and reduce errors in watermarks. Specifically, it only needs to store two memory matrices, regardless of the number of images. Moreover, even when images are degraded, errors are reduced by using both the hetero- and auto-associative memory models.

The rest of this paper is organized as follows. Section II explains the conventional zero-watermarking method, and section III explains the basic concepts of both the hetero- and auto-associative memory models. In section IV, we explain the proposed method. In section V, we describe a computer simulation to evaluate the proposed method's performance, and we conclude our study in section VI.

II. Zero-Watermarking Method

We first explain the zero-watermarking method using DCT coefficients [5], in which a K-bit watermark $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_K)^\top$ is mapped to the original image, where $\xi_i \in \{+1, -1\}$.

A. Watermark Mapping Procedure

The mapping from features to watermarks consists of the following two steps.

Step 1. Feature extraction from original image.

The original image is transformed into the frequency domain by a two-dimensional DCT. Then, K-bit coefficients are extracted from the low-frequency components, excluding the DC component, by using a zigzag scan. The coefficients $\boldsymbol{d} = (d_1, d_2, \cdots, d_K)^\top$ are binarized to obtain the feature $\boldsymbol{\eta} = (\eta_1, \eta_2, \cdots, \eta_K)^\top$. That is,

$$\eta_i = \mathrm{sgn}(d_i), i = 1, 2, \cdots, K, \tag{1}$$

where the function $\mathrm{sgn}(x)$ is defined by

$$\mathrm{sgn}(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases}. \tag{2}$$

Step 2. Secret key generation.
The secret key $\boldsymbol{W} = (W_1, W_2, \cdots, W_K)^\top$ is generated as the product of the feature $\boldsymbol{\eta}$ and the watermark $\boldsymbol{\xi}$. That is,

$$W_i = \eta_i \xi_i, \ i = 1, 2, \cdots, K. \tag{3}$$

Note that the bit lengths of $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ must be the same. Also, the associations between the generated secret key $\boldsymbol{W}$ and the original image must be preserved.

B. Watermark Extraction Procedure

The watermark extraction also consists of two steps.
Step 1. Feature extraction from original image.
The extraction of the feature $\boldsymbol{\eta}'$ is the same as Step 1 of the watermark mapping procedure above.
Step 2. Extraction of watermark $\boldsymbol{\xi}'$ from secret key $\boldsymbol{W}$.
By multiplying the feature $\boldsymbol{\eta}'$ by the stored secret key $\boldsymbol{W}$, the watermark $\boldsymbol{\xi}'$ can be retrieved as

$$\xi'_i = W_i \eta'_i, \ i = 1, 2, \cdots, K. \tag{4}$$

By generating $\boldsymbol{W}$ from a single image in this way, it is possible to map an image to a watermark $\boldsymbol{\xi}$ without embedding the watermark directly in the image's pixels. Moreover, if the image is degraded and its feature becomes inaccurate, the extracted watermark will also be inaccurate. In addition, as the number of images to be managed increases, the number of secret keys also increases, making it difficult to find the secret key corresponding to a given image.

III. Associative Memory Model

A. Hetero-Associative Memory Model

A hetero-associative memory model (HMM) is an associative memory model that retrieves associative patterns corresponding to key patterns [8], [9]. The associative and key patterns can respectively be regarded as the watermarks and features described in section II. Suppose that we have P key patterns $\boldsymbol{\eta}^\mu = (\eta_1^\mu, \eta_2^\mu, \cdots, \eta_K^\mu)^\top$ and P associative patterns $\boldsymbol{\xi}^\mu = (\xi_1^\mu, \xi_2^\mu, \cdots, \xi_N^\mu)^\top$, for $\mu = 1, 2, \cdots, P$, and that the μ-th key pattern maps to the μ-th associative pattern, where the length of the key pattern is K bits and that of the associative pattern is N bits. Each component of $\boldsymbol{\eta}^\mu$ and of $\boldsymbol{\xi}^\mu$ is assumed to be an independent random variable, which takes a value of either $+1$ or -1 according to the following probabilities:

$$\mathrm{Prob}\left[\eta_i^\mu = \pm 1\right] = \frac{1}{2}, \tag{5}$$

$$\mathrm{Prob}\left[\xi_i^\mu = \pm 1\right] = \frac{1}{2}. \tag{6}$$

The weight matrix $\boldsymbol{W}^h$ that recalls the associative pattern $\boldsymbol{\xi}^\mu$ from the key pattern $\boldsymbol{\eta}^\mu$ is given by

$$W_{ij}^h = \frac{1}{K} \sum_{\mu=1}^{P} \xi_i^\mu \eta_j^\mu. \tag{7}$$

When an input $\boldsymbol{y} = (y_1, y_2, \cdots, y_K)^\top$ is given to the HMM, the neuron's output, $\boldsymbol{x}^0 = \left(x_0^0, x_1^0, \cdots, x_N^0\right)^\top$, is given by

$$x_i^0 = \mathrm{sgn}\left(h_i\right), \tag{8}$$

where h_i is defined by

$$h_i = \sum_{j=1}^{K} W_{ij}^h y_j. \tag{9}$$

The overlap between the μ-th key pattern $\boldsymbol{\eta}^\mu$ and the input $\boldsymbol{y}$ is defined by

$$m_*^\mu = \frac{1}{K} \sum_{i=1}^{K} \eta_i^\mu y_i. \tag{10}$$

In the following, we assume that the input $\boldsymbol{y}$ is close to the ν-th key pattern $\boldsymbol{\eta}^\nu$. By substituting (7) into (9), we obtain

$$h_i = \alpha m_*^\nu \xi_i^\nu + z_i, \tag{11}$$

where $\alpha = \frac{K}{N}$, and z_i represents crosstalk noise given by

$$z_i = \frac{1}{N} \sum_{j=1}^{K} \sum_{\mu \neq \nu}^{P-1} \xi_i^\mu \eta_j^\mu y_j. \tag{12}$$

The mean of z_i is 0, and the variance $V[z_i]$ is

$$V[z_i] = \frac{P}{K}. \tag{13}$$

We define the loading rate as $\beta = \frac{P}{K}$. In the limit of infinite K, the crosstalk noise z_i can then be assumed to follow a Gaussian distribution with mean 0 and variance β. The overlap m_0^μ between the output $\boldsymbol{x}^0$ and the μ-th watermark $\boldsymbol{\xi}^\mu$ is defined by

$$m_0^\mu = \frac{1}{N} \sum_{i=1}^{N} \xi_i^\mu x_i^0. \tag{14}$$

Assuming that the crosstalk noise z_i follows a Gaussian distribution, and that the loading rate β is finite when $N \to \infty, K \to \infty$, and $P \to \infty$, the theoretical value of m_0^μ is given by

$$m_0^\mu = E\left[\frac{1}{N} \sum_{i=1}^{N} \xi_i^\mu \mathrm{sgn}(h_i)\right] \tag{15}$$

$$= \mathrm{erf}\left(\frac{m_*^\mu}{\sqrt{2\beta}}\right), \tag{16}$$

where $\mathrm{erf}(x)$ denotes the error function.

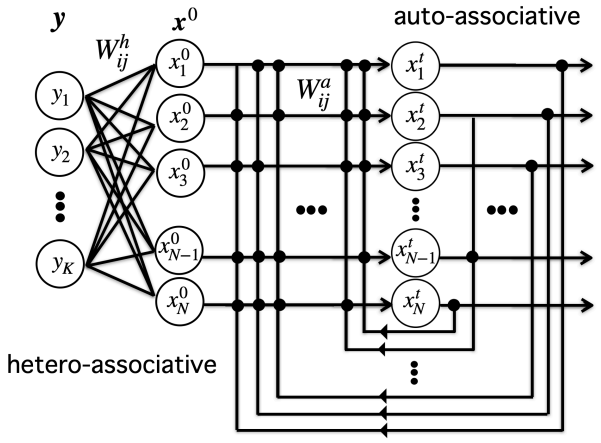


Fig. 1. Diagram of the proposed model.

B. Auto-Associative Memory Model

Next, an auto-associative memory model (AMM) is a model that recalls the closest associative pattern to a given pattern [10]–[12]. Suppose that P associative patterns $\boldsymbol{\xi}^\mu, \mu = 1, 2, \cdots, P$ are stored in an AMM. The length of an associative pattern is N bits, and the weight matrix W^a_{ij} is given by

$$W^a_{ij} = \frac{1}{N}\sum_{\mu=1}^{P}\xi^\mu_i\xi^\mu_j. \tag{17}$$

The state x_i^{t+1} of the i-th neuron at time $t+1$ is defined by

$$x_i^{t+1} = \mathrm{sgn}\left(h_i^t\right), \tag{18}$$

where h_i^t is given by

$$h_i^t = \sum_{j\neq i}^{N} W^a_{ij}x_j^t. \tag{19}$$

The overlap between the μ-th associative pattern $\boldsymbol{\xi}^\mu$ and the state $\boldsymbol{x}^t$ at time $t = 0, 1, 2, \cdots$ is defined by

$$m_t^\mu = \frac{1}{N}\sum_{i=1}^{N}\xi^\mu_i x_i^t. \tag{20}$$

The overlap at time $t = 0$ is called the initial overlap and is identical to (14).

IV. Proposed Method

As illustrated in Figure 1, in the proposed method, the associations between image features and watermarks are stored in an HMM. The HMM output is then given to an AMM, and watermark errors are corrected. This structure is the same as the HASP-type associative memory model [8]. We define $t = -1$ as the time when the key pattern is given to the HMM's input layer $\boldsymbol{y}$, and $t = 0$ as the time when the HMM output is given as the AMM's initial state $\boldsymbol{x}^0$.

A. Memory Matrix Generation

Suppose that we have P images. The μ-th image is mapped to a watermark $\boldsymbol{\xi}^\mu = (\xi^\mu_1, \xi^\mu_2, \cdots, \xi^\mu_N)^\top,\ \mu = 1, 2, \cdots, P$. As in the zero-watermarking method [5], K coefficients $\boldsymbol{d}$ of the image's low-frequency components are extracted as features. Then, by binarizing the coefficients $\boldsymbol{d}^\mu$ via (1), the feature $\boldsymbol{\eta}^\mu = (\eta^\mu_1, \eta^\mu_2, \cdots, \eta^\mu_K)^\top$ is obtained. Note that the feature's length K may be different from the watermark's bit length N.

Next, let the feature $\boldsymbol{\eta}^\mu$ be the key pattern and the watermark $\boldsymbol{\xi}^\mu$ be the associative pattern. Weight matrices $\boldsymbol{W}^h$ and $\boldsymbol{W}^a$ are then generated from (7) and (17). By storing $\boldsymbol{W}^h$, the associations between features and watermarks for P images can easily be managed. Moreover, storage of $\boldsymbol{W}^a$ enables watermark retrieval without errors.

B. Decoding Procedure

Suppose that the μ-th image is attacked and degraded. By using the features $\tilde{\boldsymbol{\eta}}^\mu = (\tilde{\eta}^\mu_1, \tilde{\eta}^\mu_2, \cdots, \tilde{\eta}^\mu_K)^\top$ extracted from this degraded image and the weight matrix $\boldsymbol{W}^h$, the μ-th watermark $\tilde{\boldsymbol{\xi}}^\mu = \left(\tilde{\xi}^\mu_1, \tilde{\xi}^\mu_2, \cdots, \tilde{\xi}^\mu_N\right)^\top$ is retrieved via (8) as the HMM's output $\boldsymbol{x}^0$. The output $\boldsymbol{x}^0$ is then given to the AMM as its initial state. The state at time $t+1$ is given by (18). After a sufficiently long time, if $m_t^\mu = 1$, then the watermark $\boldsymbol{\xi}^\mu$ has been successfully retrieved.

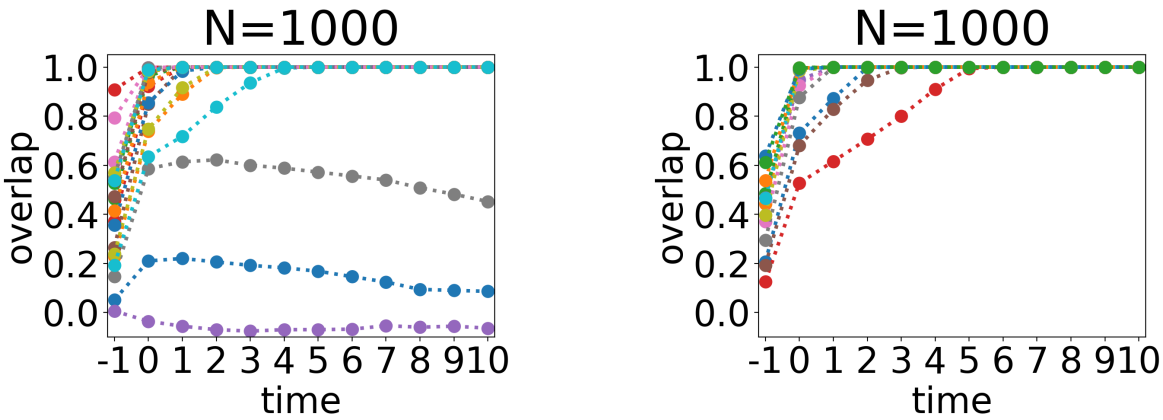
V. Computer Simulation

Through simulation results, we show that watermarks can be retrieved from an attacked image. The retrieval performance is evaluated in terms of the bit error rate (BER) of the watermark for a large number of images. The BER is computed from the overlap m and is given by

$$\mathrm{BER}(m) = \frac{1-m}{2}. \tag{21}$$

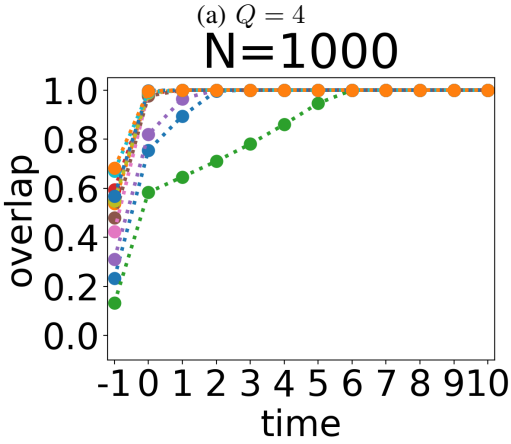
Specifically, the BERs for the feature $\boldsymbol{\eta}^\mu$ and the μ-th watermark are given by $\mathrm{BER}(m^\mu_*)$ and $\mathrm{BER}(m^\mu_t)$, respectively. A total of 38 images obtained from the USC-SIPI image database [13] were used as original images: 6 images of 108×108 pixels, 10 images of 256×256 pixels, 6 images of 512×512 pixels, 6 images of 600×800 pixels, and 7 images of other sizes. The feature bit lengths were $K = 500, 1000$, and the watermark bit length was $N = 1000$. We evaluated the proposed method's robustness against JPEG compression. As the watermark can be correctly recalled by the HMM for small attacks, we applied large attacks to the images.

First, we computed the BERs with bit lengths $K = N = 1000$. Figure 2 shows the time evolution of the overlap m^μ_t. The abscissa and ordinate represent the time and overlap, respectively. The images were JPEG-compressed with quality values of $Q = 4, 6, 8, 10$, and 20. Note that the overlap m^μ_* at time $t = -1$ is for the feature $\boldsymbol{\eta}^\mu$ given by (10), while the overlaps m^μ_t at time $t \geq 0$ are for the watermark given by (20). In the case of $m^\mu_0 = 1$, the watermark was fully recalled by the HMM. For clarity, we removed the lines for $m^\mu_0 = 1$ from Figure 2, because the majority of the results yielded $m^t_0 = 1$.

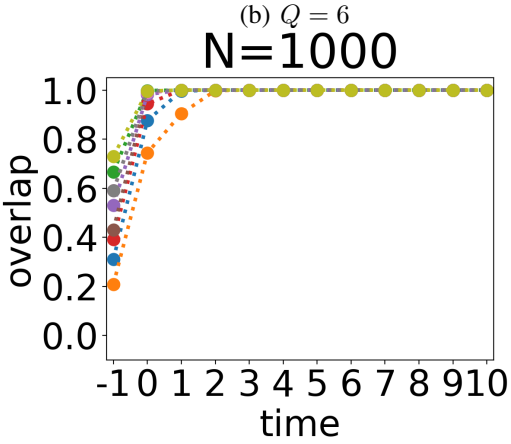


(a) $Q = 4$

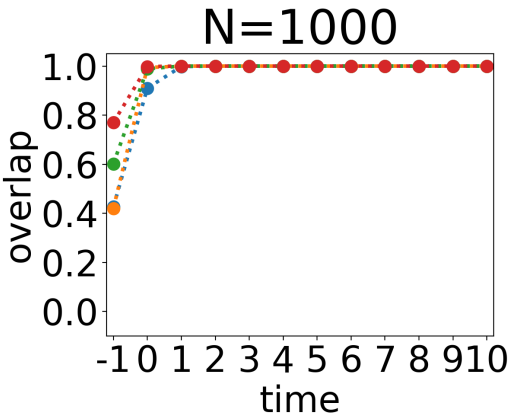
(b) $Q = 6$



(c) $Q = 8$



(d) $Q = 10$



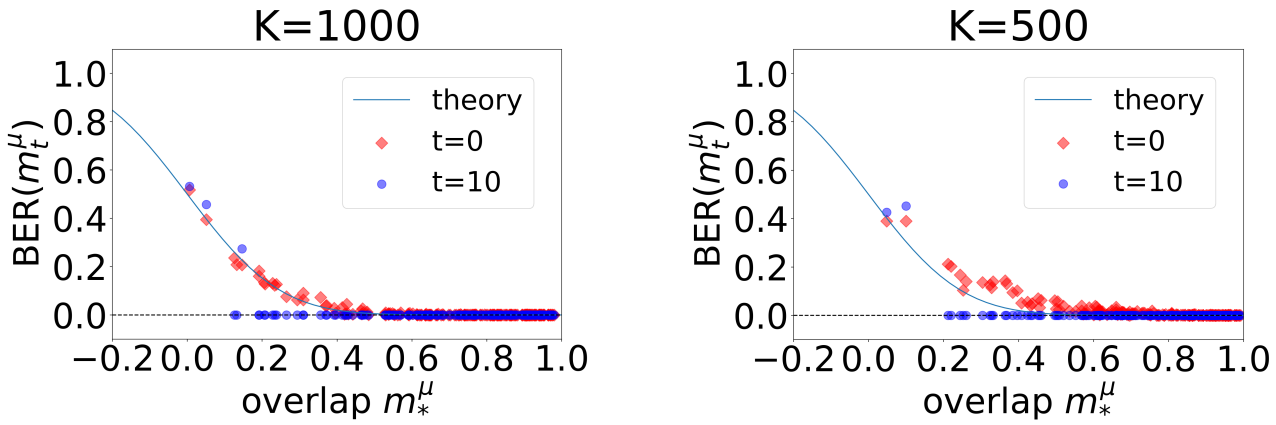
(e) $Q = 20$

Fig. 2. Time evolution of the overlap. The overlap m_*^μ at time $t = -1$ is given by (10), while the overlaps m_t^μ at time $t \geq 0$ are given by (20).

In other words, the figure only shows cases in which the image was compressed quite strongly ($Q \leq 20$). We found that as the Q-value decreased, the overlap m_*^μ at time $t = -1$ decreased, and that the overlaps at time $t = 0$ were all larger than those at time $t = -1$. This result implies that the HMM could correct certain errors. For strongly compressed cases, the overlap could be $m_0^\mu \neq 1$. However, errors that were not corrected by the HMM could be further corrected by the AMM at time $t \geq 1$. In some cases, recall could fail when the overlap m_*^μ was not above a certain value, but compression with $Q \leq 20$ is rare.

Figure 3 shows the $\mathrm{BER}(m_t^\mu)$ of the watermark versus the feature overlap m_*^μ. The curves represent the theoretical values according to (16). The red points represent $\mathrm{BER}(m_0^\mu)$ for the HMM's output, and the blue points represent $\mathrm{BER}(m_{10}^\mu)$ for the AMM's output at time $t = 10$. The number of blue points with $\mathrm{BER} = 0$ was larger than the number of red points, which demonstrates that the AMM could significantly reduce the BER.

Next, the BERs were calculated with bit lengths of $K = 500$ for the feature and $N = 1000$ for the watermark. Figure 4



(a) $K = 1000$ (b) $K = 500$

Fig. 3. BER with feature bit lengths of (a) $K = 1000$ and (b) $K = 500$.

shows the time evolution of the overlap m_t^μ. As with Figure 2, the lines for $m_0^\mu = 1$ were removed from Figure 4 for clarity. There are more lines in Figure 4 than in Figure 2, which indicates that more watermarks could not be correctly retrieved by the HMM because of the shorter feature length, $K = 500$. However, errors were corrected by the AMM at time $t \geq 1$. It was thus effective to use both the HMM and the AMM rather than only the HMM.

VI. Conclusions

In the zero-watermarking method, which does not degrade the original image, it is difficult to manage the associations between images and secret keys when there are many images. In addition, the bit lengths of the features and the watermark must be equal. Moreover, as that method does not have the ability to correct errors, it cannot retrieve the watermark when an image is degraded.

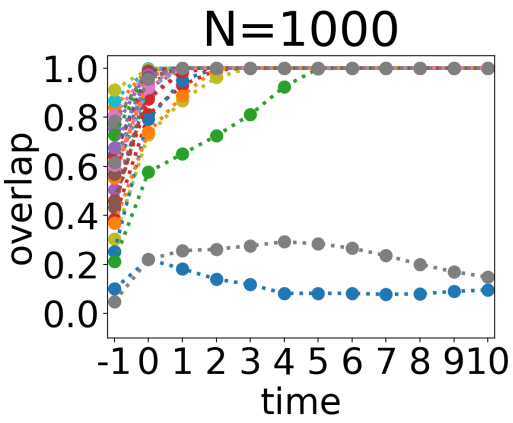
Accordingly, in this paper, we proposed a watermarking method using associative memory models, called the associative watermarking method. By introducing a hetero-associative memory model (HMM), we could solve the problem of managing the mapping between features and watermarks, as well as the bit-length restriction. Furthermore, the error correction capability could be improved by introducing an auto-associative memory model (AMM). In a computer simulation, a total of 38 features were mapped to watermarks, and the watermarks could be fully retrieved from the degraded features. Next, we plan to examine the case when an image that is similar to but different from stored images is represented by the hetero-associative memory model. Even though the probability of retrieval would be very small, it is possible that the stored watermark could be retrieved. For another future work, we will theoretically clarify the basin of attraction for watermarks in the proposed method.

Acknowledgments

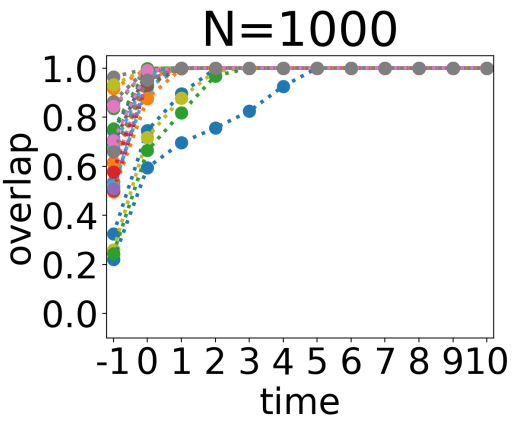
This work was supported by JSPS KAKENHI Grant Number JP20K11973. The computations were performed using the supercomputer facilities at the Research Institute for Information Technology, Kyushu University.

References

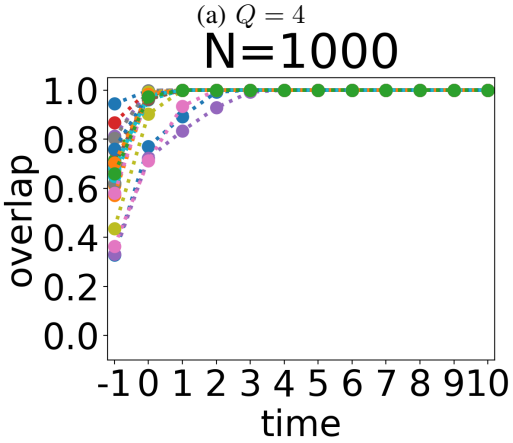
[1] J.A. Bloom J. Fridrich I.J. Cox, M.L. Miller and T. Kalker. Digital watermarking and steganography. 2nd edition. *Morgan Kaufmann*, 2007.
[2] Joseph JK Ò Ruanaidh and Thierry Pun. Rotation, scale and translation invariant spread spectrum digital image watermarking. *Signal processing*, 66(3):303–317, 1998.
[3] Ingemar J Cox, Joe Kilian, Tom Leighton, and Talal Shamoon. Secure spread spectrum watermarking for images, audio and video. In *Proceedings of 3rd IEEE international conference on image processing*, volume 3, pages 243–246. IEEE, 1996.
[4] Ingemar J Cox, Joe Kilian, F Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing*, 6(12):1673–1687, 1997.
[5] Chunhua Dong, Huaiqiang Zhang, Jingbing Li, and Yen-wei Chen. Robust zero-watermarking for medical image based on dct. In *2011 6th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, pages 900–904. IEEE, 2011.
[6] Asha Rani, Amandeep K Bhullar, Deepak Dangwal, and Sanjeev Kumar. A zero-watermarking scheme using discrete wavelet transform. *Procedia Computer Science*, 70:603–609, 2015.
[7] HyoungDo Kim. Crt-based color image zero-watermarking on the dct domain. *International Journal of Contents*, 11(3):39–46, 2015.
[8] Masaki Kawamura, Masato Okada, and Yuzo Hirai. Dynamics of selective recall in an associative memory model with one-to-many associations. *IEEE transactions on neural networks*, 10(3):704–713, 1999.
[9] Masaki Kawamura and Masato Okada. Transient dynamics for sequence processing neural networks. *Journal of Physics A: Mathematical and General*, 35(2):253, 2002.
[10] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.
[11] Shun-Ichi Amari and Kenjiro Maginu. Statistical neurodynamics of associative memory. *Neural Networks*, 1(1):63–73, 1988.
[12] Masato Okada. A hierarchy of macrodynamical equations for associative memory. *Neural Networks*, 8(6):833–838, 1995.
[13] University of Southern California. The usc-sipi image database, 1977. https://sipi.usc.edu/database/database.php, (Accessed on 09/10/2021).



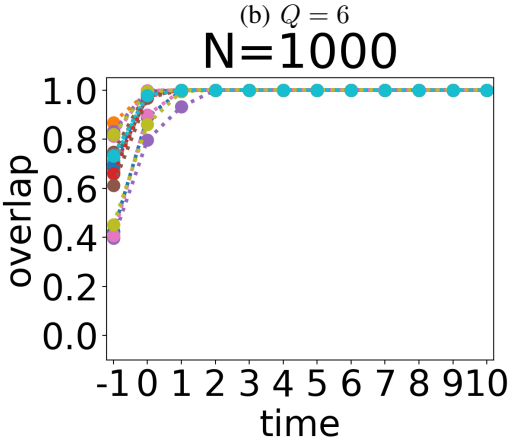
(a) $Q = 4$



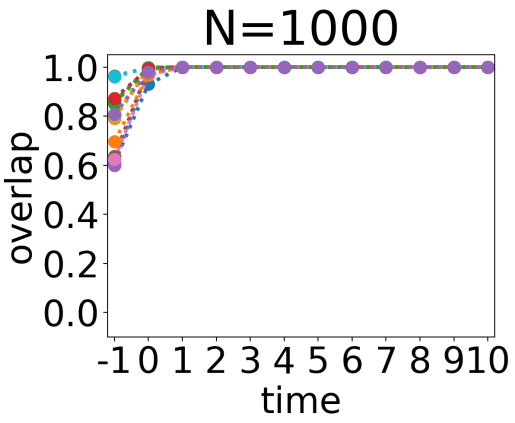
(b) $Q = 6$



(c) $Q = 8$



(d) $Q = 10$



(e) $Q = 20$

Fig. 4. Time evolution of the overlap. The overlap m_*^μ at time $t = -1$ is given by (10), while the overlaps m_t^μ at time $t \geq 0$ are given by (20).