

Digital Watermarking for Vector Drawings based on Discrete Haar Wavelet Transform

Yuichi Nakai

National Institute of Technology, Akashi College

Kuya Kohara

National Institute of Technology, Akashi College

ABSTRACT

Recently, a massive amount of cartoons or “manga” comics are distributed in digital format. Although raster format is a current mainstream format, it will be likely to use vector format from the viewpoint of data capacity and image quality. In such a situation, copyright protection technology for vector format is strongly required. In this paper, we propose a novel digital watermarking scheme for vector drawings based on discrete Haar wavelet transform. By utilizing Haar wavelet transform, the proposed method has robustness against geometric transformations such as translation, rotation, and scaling. Experimental results show that the proposed method can successfully embed watermarks without causing significant degradation to the original drawings and it also has robustness against geometric transformations.

Keywords

Digital watermarking, vector drawing, discrete Haar wavelet transform

1. INTRODUCTION

In recent years, the digital distribution of cartoons or manga comics has become more popular because of the development of electronic books and portable devices [1, 2]. On the other hand, many cartoons are plagued by the emergence of their pirated editions. Currently, most of such materials are distributed in raster formats such as JPEG or PNG. However, from the viewpoint of data capacity and image quality, the adoption of vector format is likely to become more popular. Vector drawings are substantially different from raster images in that they retain shapes like lines and points instead of a large set of pixels. Therefore, to protect the copyright, the need for a digital watermarking algorithm focused on vector drawings has recently been highlighted.

Some types of watermarking schemes for digital vector map have been proposed recently. In [3, 4], the watermarking schemes which embed watermarks into the spatial domain coefficients are proposed. There are other types of watermarking algorithms which embed watermarks in the frequency domain coefficients [5, 6]. Recently, the watermarking method for vector maps which utilize neural network is proposed in [7]. These schemes can successfully embed watermarks into vector maps. Although the algorithm proposed in [7] can tolerate geometric transformations to some extent, most of the previously proposed methods cannot resist geometric transformations, such as translation, rotation, and scaling.

As for cartoons, several watermarking algorithms have been proposed in [8, 9, 10]. However, these schemes treat cartoons in raster format data. In [11], watermarks are embedded into the bezier curves which are derived from original raster format drawings. Within the scope of our survey, there is not a watermarking method which specializes in cartoons in vector format.

We have already proposed a scheme which can directly embed watermarks into vector format cartoons in [12] as an improvement version of Kaneshiro and Inaba’s method[13]. These schemes are robust against isotropic scaling and translation since they are based on P-type Fourier descriptor[14] that is invariant against these transformations. However, there are some problems with the schemes. Namely, shapes are possible to deviate from the original by embedding, and the watermark capacity is not enough, and they are fragile against rotation.

In this paper, a new scheme of digital watermarking for vector drawings based on discrete Haar wavelet transform is proposed. The proposed method is supposed to suppress the changes of a whole shape of drawing and increase capacity and robustness against geometric transformations.

The rest of this paper is organized as follows. Discrete Haar wavelet transform for vector drawings is introduced in Section 2. In Section 3, the proposed watermarking algorithm is described in details. Experimental results and analysis are provided in Section 4. Finally, conclusions are drawn in Section 5.

2. HAAR WAVELET TRANSFORM FOR VECTOR DRAWINGS

2.1 The representation of vector drawings

In the proposed method, watermarks are embedded into vector drawings using discrete Haar wavelet transform (HWT). When applying HWT to vector drawings, it is assumed that the original curve consists of a set of N lines. Fig. 1 illustrates the approximation used in this scheme. C denotes the original curve, δ is a length of each line. Each node of lines is presented by z_m . By using this conversion, the approximation form of C is obtained as $G = \{z_m \mid m = 0, 1, \dots, N-1\}$. The parameter δ determines the accuracy of G . As a result, more precise G is available with smaller δ . However, a smaller δ results in the increase in N and this leads to an increase in file size. So the value of δ must be determined depending on the demand for the vector drawings. In the following discussion, we define z_m as

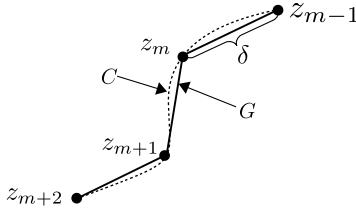


Fig. 1: Segmentation of a curve

$$\mathbf{z}_m = x_m \mathbf{i} + y_m \mathbf{j} \quad (m = 0, 1, \dots, N-1). \quad (1)$$

where $\mathbf{i} = (1, 0)$, $\mathbf{j} = (0, 1)$.

2.2 Definition

HWT for the sequence of vector \mathbf{z}_m is defined as (2) and (3).

$$\mathbf{a}_n = \frac{\mathbf{z}_{2n} + \mathbf{z}_{2n+1}}{2} \quad (2)$$

$$\mathbf{r}_n = \frac{\mathbf{z}_{2n} - \mathbf{z}_{2n+1}}{2}, \quad (3)$$

where $n = 0, 1, \dots, N/2 - 1$. \mathbf{a}_n and \mathbf{r}_n are called approximation coefficient vectors and detailed coefficient vectors respectively. Following equations show the inverse HWT for the two-dimensional vector sequence.

$$\mathbf{z}_{2n} = \mathbf{a}_n + \mathbf{r}_n \quad (4)$$

$$\mathbf{z}_{2n+1} = \mathbf{a}_n - \mathbf{r}_n. \quad (5)$$

The above equations reproduce the original vectors.

2.3 Invariance for geometric transformations

In this section, the invariant characteristics of HWT coefficients will be shown. For each vector $\mathbf{z}_m = (x_m, y_m)$ in G , apply to scale s_x, s_y , rotation around the origin θ and translation t_x, t_y . The result vector \mathbf{z}'_m is represented in (6).

$$\mathbf{z}'_m = \begin{bmatrix} s_x \cos \theta & -s_y \sin \theta \\ s_x \sin \theta & s_y \cos \theta \end{bmatrix} \begin{bmatrix} x_m \\ y_m \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \quad (6)$$

Applying HWT to \mathbf{z}'_m , these coefficients are available.

$$\mathbf{a}'_n = \frac{1}{2} \begin{bmatrix} s_x \cos \theta & -s_y \sin \theta \\ s_x \sin \theta & s_y \cos \theta \end{bmatrix} \times \begin{bmatrix} x_{2n} + x_{2n+1} \\ y_{2n} + y_{2n+1} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (7)$$

$$\mathbf{r}'_n = \frac{1}{2} \begin{bmatrix} s_x \cos \theta & -s_y \sin \theta \\ s_x \sin \theta & s_y \cos \theta \end{bmatrix} \times \begin{bmatrix} x_{2n} - x_{2n+1} \\ y_{2n} - y_{2n+1} \end{bmatrix}. \quad (8)$$

In (8), there is no t_x and t_y , which means detailed coefficient vectors are invariant against translations.

Now we introduce the different value α_n as follows.

$$\begin{aligned} \alpha_n &= \mathbf{a}'_{n+1} - \mathbf{a}'_n \\ &= \frac{1}{2} \begin{bmatrix} \alpha_{n,x} s_x \cos \theta - \alpha_{n,y} s_y \sin \theta \\ \alpha_{n,x} s_x \sin \theta + \alpha_{n,y} s_y \cos \theta \end{bmatrix}. \end{aligned} \quad (9)$$

$$\alpha_{n,x} = x_{2n+2} + x_{2n+3} - x_{2n} - x_{2n+1}. \quad (10)$$

$$\alpha_{n,y} = y_{2n+2} + y_{2n+3} - y_{2n} - y_{2n+1}. \quad (11)$$

Considering inner product $\zeta_n = \alpha_n \cdot (-\mathbf{r}'_n)$ and cross product $\eta_n = \alpha_n \times (-\mathbf{r}'_n)$,

$$\begin{aligned} \zeta_n &= \frac{1}{4} \alpha_{n,x} (x_{2n+1} - x_{2n}) s_x^2 \\ &\quad + \frac{1}{4} \alpha_{n,y} (y_{2n+1} - y_{2n}) s_y^2 \end{aligned} \quad (12)$$

$$\begin{aligned} \eta_n &= \frac{1}{4} \alpha_{n,y} (x_{2n} - x_{2n+1}) s_x s_y \\ &\quad - \frac{1}{4} \alpha_{n,x} (y_{2n} - y_{2n+1}) s_x s_y. \end{aligned} \quad (13)$$

In (12) and (13), θ is not included, so this means ζ_n and η_n are invariant against rotation. However, each ζ_n and η_n contain s_x and s_y , they are affected by scaling. Now let assume $s_x = s_y = s$. In this case, ζ_n, η_n are calculated as follows.

$$\begin{aligned} \zeta_n &= \frac{1}{4} \alpha_{n,x} (x_{2n+1} - x_{2n}) s^2 \\ &\quad + \frac{1}{4} \alpha_{n,y} (y_{2n+1} - y_{2n}) s^2 \end{aligned} \quad (14)$$

$$\begin{aligned} \eta_n &= \frac{1}{4} \alpha_{n,y} (x_{2n} - x_{2n+1}) s^2 \\ &\quad - \frac{1}{4} \alpha_{n,x} (y_{2n} - y_{2n+1}) s^2. \end{aligned} \quad (15)$$

Considering the ratio of ζ_n and η_n .

$$\frac{\eta_n}{\zeta_n} = \frac{\alpha_{n,x} (x_{2n+1} - x_{2n}) + \alpha_{n,y} (y_{2n+1} - y_{2n})}{\alpha_{n,x} (x_{2n} - x_{2n+1}) + \alpha_{n,y} (y_{2n} - y_{2n+1})}. \quad (16)$$

This does not contain s . Let $\varphi_n \in (-\pi/2, \pi/2)$ be the angle between α_n and $-\mathbf{r}'_n$, following relation is derived.

$$\tau_n = \frac{\eta_n}{\zeta_n} = \frac{\sin \varphi_n}{\cos \varphi_n} = \tan \varphi_n. \quad (17)$$

(17) shows that φ_n is invariant against uniform isotropic scaling. As a result, it is clear that φ_n is invariant against translation, rotation, and uniform isotropic scaling.

3. PROPOSED METHOD

From the discussion in 2.3, it is proven that watermarks which are robust against geometric transformation can be embedded in G by controlling φ_n . To control φ_n , either α_n or $-\mathbf{r}_n$ is a candidate for controlling. Because α_n hold important information about the figure of curves, it is preferable to change \mathbf{r}_n for watermark embeddings.

3.1 Preliminaries

Define watermark data D as follows.

$$D = (d_k), \quad d_k \in \{0, 1\} \quad (18)$$

where $k = 0, 1, \dots, N/2 - 2$. For embedding and extracting watermark, quantizer $Q_{\mu, \lambda}$ is used.

$$Q_{\mu, \lambda}(m) = \left\lfloor \text{clamp} \left[\frac{m - (\mu - \lambda)}{2\lambda} \right] \times 7 \right\rfloor. \quad (19)$$

where clamp is a function which is defined as follows.

$$\text{clamp}(x) = \begin{cases} 1 & \text{if } x > 1 \\ 0 & \text{if } x < 0 \\ x & \text{otherwise.} \end{cases} \quad (20)$$

Inverse quantizer $Q_{\mu, \lambda}^{-1}$ is also defined as

$$Q_{\mu, \lambda}^{-1}(t) = \mu - \lambda + \frac{t + 0.5}{7} \times 2\lambda. \quad (21)$$

For embedding and extracting watermark, a random number sequence E is introduced. Each element of E is unsigned 8 bit integer and $e_{k, b} \in \{0, 1\}$ denotes b -th bit of e_k .

$$E = (e_k), e_k = \sum_{b=0}^7 2^b e_{k, b}. \quad (22)$$

If watermark bit is d , embedding is executed by replacing b such as $e_{k, b} = d$. When extracting watermark, we can recover $e_{k, b'}$ from quantized value b' where e_k must meet following condition.

$$\forall k \{ \exists b_0 [e_{k, b_0} = 0] \text{ and } \exists b_1 [e_{k, b_1} = 1] \} \quad (23)$$

3.2 Watermark Embedding

Watermarks are embedded by executing following procedures for $k = 0, 1, \dots, N/2 - 2$.

- (1) $\alpha_k = \mathbf{a}_{k+1} - \mathbf{a}_k$.
- (2) calculate τ_k using $\alpha_k, -\mathbf{r}_k$.

$$\tau_k \leftarrow \frac{\alpha_k \times (-\mathbf{r}_k)}{\alpha_k \cdot (-\mathbf{r}_k)}. \quad (24)$$

- (3) $t_k \leftarrow Q_{\mu, \lambda}(\tau_k)$
- (4) For $e_{k, c}$ and d_k , choose a minimum integer $c \in \{0, 1, \dots, 7\}$ that satisfy $e_{k, c} = d_k$ and (25) and $c_k \leftarrow c$.

$$f(c) = |c_{k-1} - c| + 2|t_k - c|, \quad (25)$$

where $c_{(-1)} = 3$.

- (5) $\hat{r}_k \leftarrow Q_{\mu, \lambda}^{-1}(c_k)$
- (6) Generate vector ρ_k from \hat{r}_k and replace \mathbf{r}_k by \mathbf{r}'_k .

$$\mathbf{r}'_k \leftarrow -\rho_k \quad (26)$$

$$\rho_k = \frac{\alpha_k}{|\alpha_k|} * \left\{ \frac{|\mathbf{r}_k|}{\sqrt{1 + \hat{r}_k^2}} \begin{bmatrix} 1 \\ \hat{r}_k \end{bmatrix} \right\}. \quad (27)$$

For vector \mathbf{u} and \mathbf{v} , an operator $*$ defined as follows.

$$\mathbf{u} * \mathbf{v} = (u_1 v_1 - u_2 v_2) \mathbf{i} + (u_1 v_2 + u_2 v_1) \mathbf{j}. \quad (28)$$

where $\mathbf{u} = u_1 \mathbf{i} + u_2 \mathbf{j}$, $\mathbf{v} = v_1 \mathbf{i} + v_2 \mathbf{j}$.

- (7) calculate modified vector \mathbf{z}_k using (4), (5).

The parameters of quantizer μ, λ are determined as $\mu = \mu_\tau$ and $\lambda = 3\sigma_\tau$, where μ_τ is the average of τ_k and σ_τ is the standard deviation of τ_k .

3.3 Watermarking Extraction

Embedded watermarks are extracted by following procedures for $k = 0, 1, \dots, N/2 - 2$.

- (1) $\alpha'_k = \mathbf{a}'_{k+1} - \mathbf{a}'_k$.
- (2) calculate τ'_k using α'_k and $-\mathbf{r}'_k$ by (24).
- (3) $t'_k \leftarrow Q_{\mu, \lambda}(\tau'_k)$.
- (4) $d'_k \leftarrow e_{k, t'_k}$.

4. COMPUTER SIMULATION

In this section, some experimental results of the proposed scheme for sample vector drawings are demonstrated. In the computer simulation, three SVG drawings shown in Fig. 2(0??) to 2(0??) are used. Specifications of each drawing are shown in Table 1. Computer simulations were performed for $\delta = 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5$.

Table 1. : Specifications of each drawings

	size (pixels)	paths
english	541.621 × 125.355	1
hiragana	561.825 × 73.778	15
girl	338.029 × 407.150	64

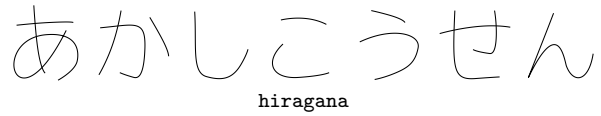
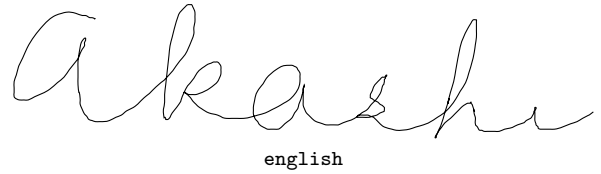


Fig. 2: Original SVG drawings

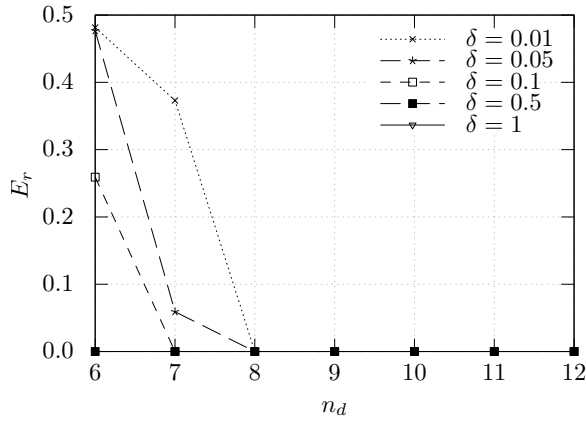


Fig. 3: The relation between n_d and E_r : english

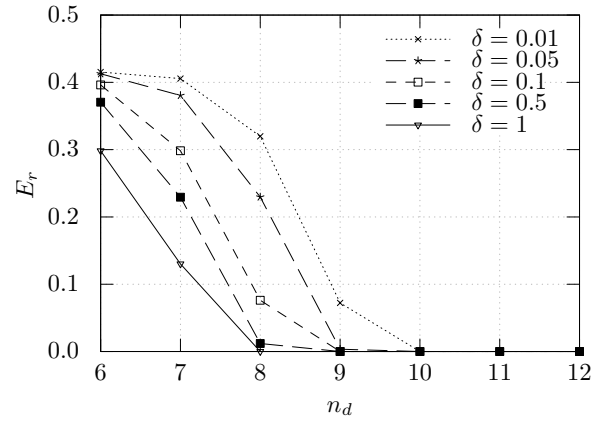


Fig. 5: The relation between n_d and E_r : girl1

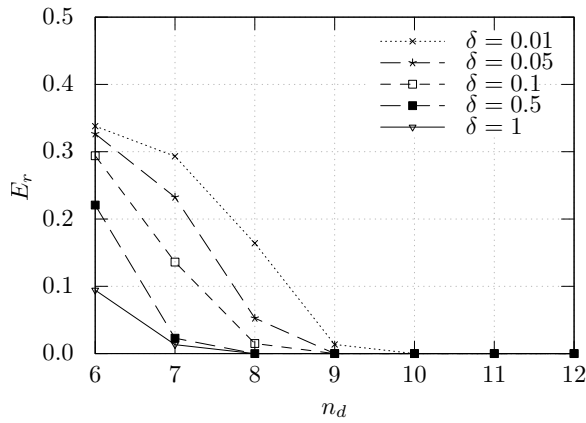


Fig. 4: The relation between n_d and E_r : hiragana

4.1 Embedding Capacity

The relation between the number of nodes N in drawings and watermark embedding capacity C_{emb} [bits] is shown in Table 2. It is theoretically guaranteed that $C_{emb} \leq N/2$ in the proposed method. From Table 2, it is obvious that C_{emb} approaches to $N/2$ when smaller δ is used.

4.2 Output Precision

In the proposed algorithm, modified values x'_m, y'_m are recorded as SVG format. It is assumed that the floating point values x'_m and y'_m rounded to n_d digits after the decimal point are written to the output file. In this case, n_d affects the extraction error rate of watermarks. The extraction error rate E_r is defined as

$$E_r = \frac{\text{the total number of extracted erroneous watermarks}}{\text{the total number of extracted watermarks}}. \quad (29)$$

The relations between the precision n_d and E_r for $n_d = 6, 7, \dots, 12$ are investigated. Some of the results are shown in Fig. 3 to Fig. 5. From these results, it is clear that the less the value of δ is, the larger n_d is needed. The required minimum values of δ and n_d vary depending on the attribute of an original drawing.

4.3 The quality of embedded drawings

Some of the embedded drawings are shown in Fig. 6 to Fig. 8. As for the value n_d , the minimum value is adopted among those in which $E_r = 0$. For those parameters, no noticeable visual degradation can be observed. From the discussion in 2.1, smaller δ is preferable for better image quality. However, smaller δ causes an increase in the file size and n_d . Considering the above results, $\delta = 1$ is an appropriate value in most cases.

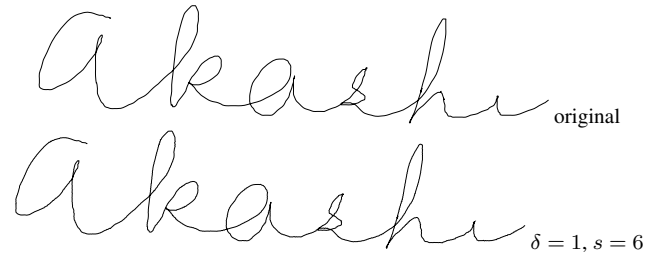


Fig. 6: The quality of embedded drawing : english

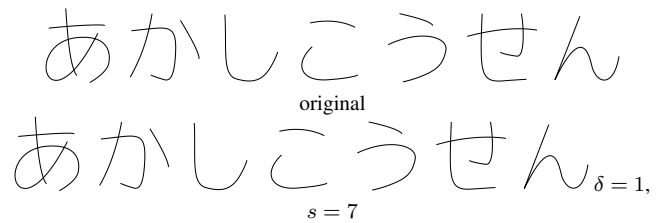


Fig. 7: The quality of embedded drawing : hiragana

4.4 Robustness against geometric transformations

The proposed method is robust to geometric transformations as discussed in 2.3. However, in real applications, the effect of n_d must be considered. As for the translation, it is evident that the proposed method is robust, so the simulations are performed only for rotation and uniform isotropic scaling. In Fig. 9,10, 11, the results for rotation $\theta = \pi/2$ are shown. The results for uniform

Table 2. : The number of nodes and embedding capacity

δ		0.01	0.02	0.05	0.1	0.2	0.5	1	2	5
english	N	174973	87486	34991	17493	8744	3490	1739	860	338
	C_{emb}	87486	43742	17495	8746	4371	1744	869	429	168
hiragana	N	137145	68574	27433	13720	6865	2749	1377	690	281
	C_{emb}	68560	34275	13705	6850	3421	1363	676	333	129
girl	N	594780	297404	118978	59508	29765	11927	5976	3001	1209
	C_{emb}	294111	147042	58800	29380	14675	5852	2909	1438	548

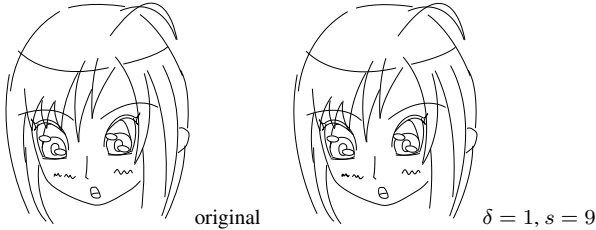


Fig. 8: The quality of embedded drawing : girl

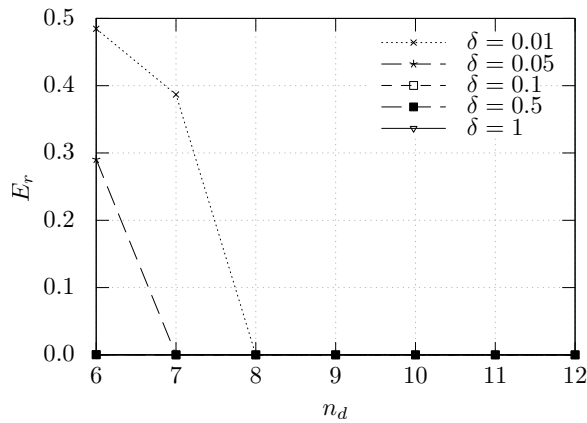


Fig. 9: Robustness against rotation : english

isotropic scaling are shown in Fig. 12, 13, 14. From these results, there are some differences in the value of E_r , the minimum n_d which give $E_r = 0$ are the same values for non-transformed drawings and rotated or scaled versions. Therefore, no extra digits are needed to guarantee the proposed method being robust against geometric transformations.

5. CONCLUSION

In this paper, a novel watermark embedding method for vector drawings is proposed. The proposed method utilizes a Haar wavelet transform. By exploiting the nature of HWT, the proposed method is robust against geometric transformations such as translation, rotation, and uniform isotropic scaling. Experimental results show the embedding capacity, the excellent image quality of the proposed scheme, robustness against geometric transformations. Although some parameters have to be determined by experiments, our scheme has enough watermark capacity and can embed watermarks into vector drawings without significant degradation. Also, it is robust against geometric transformations by setting appropriate

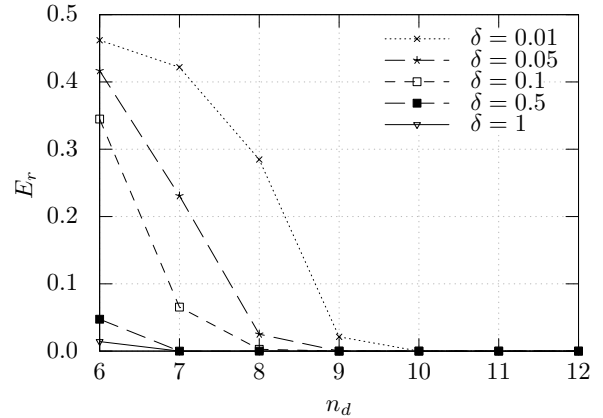


Fig. 10: Robustness against rotation : hiragana

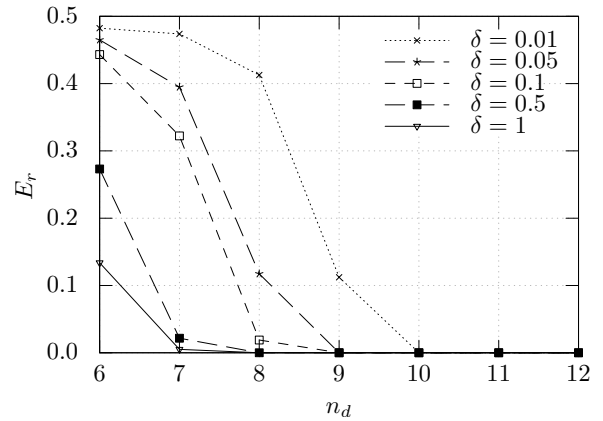


Fig. 11: Robustness against rotation : girl

parameters. Determining adequate parameters in our method is one of the future works.

6. REFERENCES

- [1] "Manga One", <http://manga-one.com/>, accessed 3 April 2019.
- [2] "Shonen Jump Plus", <https://shonenjumpplus.com/>, accessed 3 April 2019.
- [3] Yu-Chi Pu, Wei-Chang Du, and I-Chang Jou. Toward blind robust watermarking of vector maps. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 930–933. IEEE, 2006.

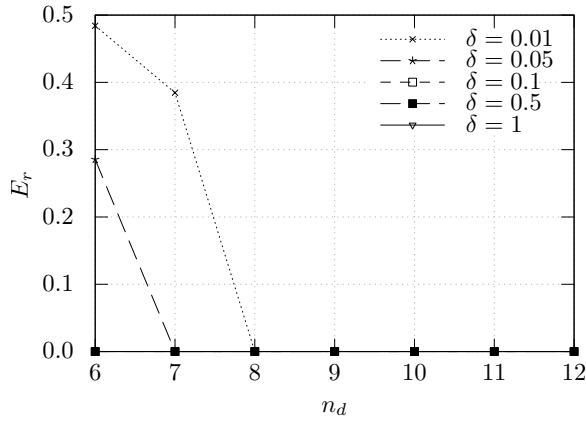


Fig. 12: Robustness against scaling : english

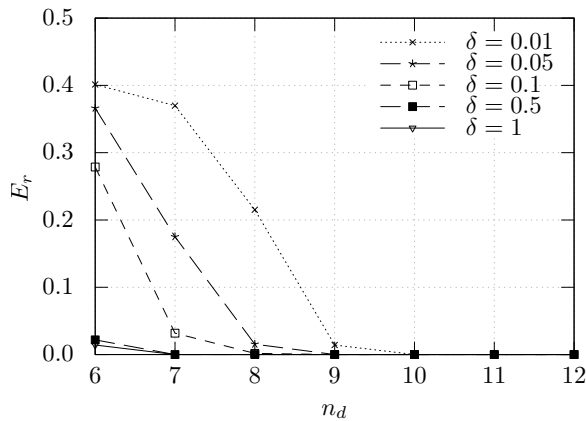


Fig. 13: Robustness against scaling : hiragana

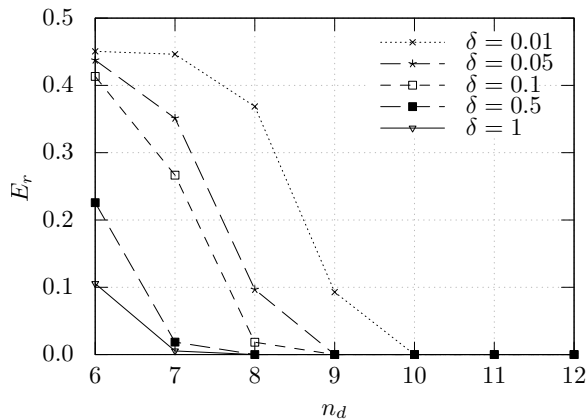


Fig. 14: Robustness against scaling : girl

[4] Baiyan Wu, Wei Wang, and Dandan Miao. 2d vector map watermarking based on spatial relations. In *International Conference on Earth Observation Data Processing and Analysis (ICEODPA)*, volume 7285, page 728532. International Society for Optics and Photonics, 2008.

[5] Itaru Kitamura, Satoshi Kanai, and Takeshi Kishinami. Copyright protection of vector map using digital watermarking method based on discrete fourier transform. In *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No. 01CH37217)*, volume 3, pages 1191–1193. IEEE, 2001.

[6] Chang-Qing Zhu, Cheng-Song Yang, and Qi-Sheng Wang. A watermarking algorithm for vector geo-spatial data based on integer wavelet transform. *Int Arch Photogramm Remote Sens Spat Inf Sci*, 37(B4):15–18, 2008.

[7] Jian-Guo Sun, Guo-Yin Zhang, Ai-Hong Yao, and Jun-Peng Wu. A reversible digital watermarking algorithm for vector maps. *coordinates*, 3(9):16–18, 2014.

[8] Koichi Kise Weihan Sun. Partial copy detection for copyright protection of line drawings. *The IEICE Transactions on Information and Systems (Japanese Edition) D*, 93(6):909–919, 2010.

[9] Taegeun Oh, Nakyeon Choi, Doyoung Kim, and Sanghoon Lee. Low-complexity and robust comic fingerprint method for comic identification. *Signal Processing: Image Communication*, 39:1–16, 2015.

[10] Yu Chao Sun et al. Robust cartoon zero-watermark algorithm based on nsst. In *Advances in Computer Science and Ubiquitous Computing*, pages 53–59. Springer, 2017.

[11] Takafumi Kudo, Junko Miyata, Tetsuya Morizumi, and Hiroto Kinoshita. A watermarking method for the bezier curve. *IEICE Technical Report*, 112(489):243–248, 2013.

[12] Yuichi Nakai Kuya Kohara. Geometrical transformation resistant digital watermarking for line drawing. In *Proc. Society of Information and System Conference 2012*, page 116. IEICE, 2012.

[13] Hiroyuki Inaba Kazushi Kaneshiro. Geometrical transformation resistant digital watermarking for line drawing. *Journal of Information Processing Society of Japan*, 46(8):1834–1842, 2005.

[14] Yoshinori Uesaka. A new fourier descriptor applicable to open curves. *The Transactions of the Institute of Electronics and Communication Engineers of Japan*, J67-A, 67(3):166–173, 1984.