A Reversible Watermarking for 2D Engineering Graphics Based on Difference Expansion With Adaptive Interval Partitioning

Fei Peng, *Member, IEEE*, Wenyan Jiang, Min Long, and Keqin Li, *Fellow, IEEE*

Abstract—To resolve the distortion resulting from the mismatch between the interval partition and the watermark distribution in the reversible watermarking based on difference expansion, an investigation is made to difference expansion with adaptive interval partitioning. It is found that the conventional floating number-based methods with uniform interval partitioning are not fully considering different watermark distributions, and the embedding distortion can be optimized. Thus, a reversible watermarking based on difference expansion with adaptive interval partitioning is put forward for 2D engineering graphics. Each interval is adaptively partitioned into 2^s sub-intervals, and the watermark is embedded via transforming the positions of the vertices to its corresponding sub-intervals. The watermark extraction is accomplished by acquiring the indices of the sub-intervals of the watermarked vertices, and the graphics recovery is achieved by moving the watermarked vertices to the reverse direction. Moreover, a new polar coordinate system is built to be invariant to overall geometric transformation and vertices rearrangement. Experimental results and analysis indicate that it maintains good authentication ability and semi-fragility. Furthermore, its imperceptibility is significantly improved compared with the existing watermarking schemes under the same experimental conditions.

Index Terms—Adaptive partition, difference expansion, reversible watermarking, 2D engineering graphics

1 Introduction

As an important manufacturing data, 2D engineering graphics are widely used in military, construction, machinery, clothing, medical diagnosis, judicial expertise and other fields. Meanwhile, the fast development of the network manufacturing promoted the collaborative design, which has become one of the most popular and convenient engineering design modes. During collaborative design, engineering graphics are frequently transferred between collaborative units. Aiming to guarantee the security of engineering graphics in the transfer process, the integrity authentication of them is extremely important [1]. Currently, as a content integrity protection means for 2D engineering graphics, reversible watermarking [2], [3], [4] can restore the host graphics after authentication, which is especially suitable for the application scenarios with strict data precision demands.

At present, the typical reversible watermarking schemes are primarily based on lossless compression (LC) [5], [6], [7], histogram shifting (HS) [8], [9], integer transform (IT) [10], [11], [12], difference expansion (DE) [13], [14], [15], [16], [17], prediction error expansion (PEE) [18], [19] and their variants [20], [21], etc. Among them, difference expansion based reversible watermarking algorithms are extensively investigated for raster images. Since the target hosts are pixel values represented by integers and the capacity relies on the correlation of the host, its application in hosts with low correlation such as 2D engineering graphics is limited.

Recently, difference expansion based reversible watermarking schemes were proposed for 2D engineering graphics, and they can be classified into integer-based schemes [22], [23], [24], [25] and floating number-based schemes [26], [27], [28], [29], [30], [31], [32]. For the former, most of them are transplanted by the reversible watermarking schemes for raster images. They do not fully utilize the nature of 2D engineering graphics represented by floating numbers, which leads to poor imperceptibility and limited capacity. While for the later, floating number-based schemes can achieve larger capacity and better imperceptibility. After analysis, it is found that even if the watermark information is processed using traditional encryption or compression techniques, its uniformity cannot be completely guaranteed. However, the interval partition of the existing floating number-based schemes is essentially in uniform, which cannot adapt to different watermark distributions, and it hampers the further reduction of the embedding distortion. Large geometry distortion may provide clues for

- *Fei Peng is with the Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou, Guangdong 510000, China. E-mail: eepengf@gmail.com.*
- *Wenyan Jiang is with the School of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410082, China. E-mail: jwy1996@hnu.edu.cn.*
- *Min Long is with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, Hunan 410114, China. E-mail: caslongm@aliyun.com.*
- *Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA. E-mail: lik@newpaltz.edu.*

Manuscript received 9 June 2020; revised 21 Mar. 2022; accepted 4 Apr. 2022. Date of publication 8 Apr. 2022; date of current version 13 May 2023.
This work was supported in part by Project supported by National Natural Science Foundation of China under Grants 92067104, 62072055, and U1936115.
(Corresponding author: Min Long.)
Digital Object Identifier no. 10.1109/TDSC.2022.3166134

the adversary and increase the risk of watermark damage. Consequently, to optimize the performance, a difference expansion based reversible watermarking with adaptive interval partitioning is put forward in this paper. In contrast to the fixed uniform interval partition, it adaptively partitions the intervals according to the distribution of the watermark information, which is helpful for improving the imperceptibility.

The main contributions of the paper are summarized as follows.

1) An investigation is made to the existing floating number based reversible watermarking schemes for 2D engineering graphics, and the relationship between the interval partition modes, distribution of the watermark information and the imperceptibility is analyzed.
2) An adaptive interval partition method is proposed to adaptively match the distribution of watermark information. Analysis shows that it can reduce the distortion of the watermarked host to a certain extent. As far as we know, it is the first research on reducing embedding distortion by using the adaptive interval partition.
3) Based on difference expansion, a reversible watermarking scheme with adaptive interval partitioning for 2D engineering graphics is put forward. Experimental results and analysis indicate that it can achieve good invisibility, and maintain a good balance among authentication ability, imperceptibility, and semi-fragility.

The remaining contents of the paper are organized as follows. The review of the related work is presented in Section 2. Analysis is performed to the improved difference expansion (IDE) and then the implementation of the difference expansion with adaptive interval partitioning is elaborated in Section 3. In Section 4, the proposed semi-fragile reversible watermarking scheme is depicted. Experiment results and analyses are given in Section 5. Some conclusions are made in Section 6.

2 Related Work

2.1 DE Based Reversible Watermarking for Images

A DE based reversible watermarking scheme was first proposed for raster images by Tian *et al.* [13]. It divides an image into several pixel pairs, selects expandable pixel pairs for difference expansion and watermark embedding, and uses the redundancy of digital content to achieve reversibility. Nevertheless, it leads to large distortion, and the location map limits its capacity. A DE based reversible watermarking scheme with high data-hiding capacity for color images was developed by Alattar *et al.* [14]. Several bits are hidden into the difference expansion of the vectors of the adjacent pixels. With the requirements of avoiding underflow and overflow, the general reversible integer transformation is implemented for each vector. In addition, data embedding is recursively performed to the color components to maximize the capacity. Nevertheless, its capacity highly relies on the nature of the image. After that, a DE based reversible watermarking using integer Haar wavelet transform was created by Hu *et al.* [15]. The horizontal and vertical difference images are used for data hiding, and an embedding mechanism is introduced to reduce the difference of two difference images. A two-dimensional DE based reversible data hiding scheme for images was put forward by Osamah *et al.* [16]. It first partitions the image into blocks with the same size. For each expandable block, 16 bits are embedded into it. By choosing some specific blocks for embedding, the visual effect of the stego image is maintained. Recently, a novel DE based reversible watermarking was constructed by Gujjunoori *et al.* [17]. It reduces the auxiliary information by rearranging the pixel pair indices of the changeable differences. Furthermore, the capacity and visual quality are both improved by embedding watermark information twice.

2.2 Reversible Watermarking for 2D Vector Graphics

2.2.1 Reversible Watermarking in Transform Domain

A reversible watermarking for 2D vector graphics in transform domain was first investigated by Voigt *et al.* [33]. The coordinates of the graphics are first grouped with a size of 8 coordinates. For each group, DCT transform is conducted to it to obtain its corresponding coefficients. After that, the high frequency coefficients are modified to embed watermark. However, the capacity is limited and the visual quality of the watermarked graphics is poor. Thereafter, a lossless watermarking using wavelet transform for vector graphics was presented in [34]. The high frequency wavelet coefficients are used for embedding, and the spatial data can be recovered without affecting the visual effect of the vector graphics during the watermark extraction. A reversible watermarking using discrete Fourier transform was designed by Shen *et al.* [35]. It first groups the vertices, and then the amplitudes and phases of each group are acquired by using discrete Fourier transform. By using the quantization index modulation, the watermark is embedded into the phase. It can resist against rotation, translation, and data format conversion.

2.2.2 Reversible Watermarking in Spatial Domain

Since the above transform domain-based algorithms have high computation complexity and limited capacity, much efforts are put on the research in the spatial domain. Currently, the existing schemes can be classified into integer-based methods [22], [23], [24], [25] and floating number-based methods [26], [27], [28], [29], [30], [31], [32].

A reversible data hiding algorithm for 2D vector graphics was produced by Shao *et al.* [22]. The difference of two adjacent vertices is expanded and the watermark is embedded by modifying its least significant bit. Since the marked information requires to be compressed before the embedding, the payload depends on the self-correlation of the graphics, and the capacity is generally low. Based on DE, Wang *et al.* suggested to modify the difference of adjacent coordinates or the Manhattan distance between adjacent vertices for watermark embedding [23]. However, extra room needs to be reserved for the location map. After that, a reversible watermarking scheme based on invariant sum value was devised by Geng *et al.* [24]. It modifies the integer part of a pair of coordinates to embed watermark and meanwhile their sum value is kept unchanged. However, as the distortion needs to be controlled within the error tolerance, the capacity is limited. Subsequently, a reversible watermarking scheme based on RCM and triple differences expansion was demonstrated in [25]. However, its capacity and imperceptibility still depend on the correlation of the data. The above integer-based schemes are essentially based on difference expansion and its variants. Since these schemes do not make full use of the characteristics that the engineering graphics are represented by floating numbers, their performance is limited. To this end, some floating number based reversible watermarking schemes were successively emerged.

A reversible watermarking based on IDE was invented by Peng *et al.* [26]. It embeds watermark in the ratio coefficient of the vertex relative position, which can realize good performance in imperceptibility and semi-fragility. Subsequently, they contributed a reversible watermarking scheme based on the improved quantization index modulation (IQIM) [27]. It modifies the corresponding relative distances of vertices in the specific quantization interval to embed watermark. However, the vertices located within the boundaries of the quantization intervals cannot be embedded. Therefore, a large capacity reversible watermarking algorithm by combining IDE and IQIM was reported by Xiao *et al.* [28]. It guarantees that all vertices can be embedded, and the capacity is significantly improved. Subsequently, a reversible watermarking algorithm based on virtual coordinates (VC) was introduced by Wang *et al.* [29]. It first calculates two virtual coordinates for each embedded vertex, and the watermark is embedded by modifying the state value of the located interval that each vertex is in. Although it can realize a relatively high capacity, it does not possess the semi-fragility of rotation, scaling and translation (RST). Afterwards, a low-distortion reversible watermarking based on region nesting (RN) for 2D engineering graphics was studied by Peng *et al.* [30]. The 2D watermark region is partitioned into several sub-regions with the same size in a nested regulation, and the watermark embedding is accomplished by moving vertices from their original sub-regions to the corresponding watermark sub-regions according to the embedding function. Recently, a reversible watermarking based on error correction codes (ECC) was implemented by Qiu *et al.* [31]. The watermark is embedded into the polar coordinates to enhance the robustness. After that, a rich information (RI) reversible watermarking was proposed for vector map in [32]. Compression coding is utilized to shorten the size of the watermark data, and the host is grouped and transformed to polar coordinates to embed watermark. A relative good robustness is achieved.

Based on the above analysis, the interval partition of floating number based reversible watermarking schemes are essentially in uniform. With different embedding strength, when the watermark information is unevenly distributed, the distortion of the graphics cannot be further reduced. To this end, the relation among interval partition modes, distribution of the watermark information and the imperceptibility is investigated, and a reversible watermarking for 2D engineering graphics based on difference expansion with adaptive interval partitioning is proposed.

3 Reversible Watermarking Based on DE With Adaptive Interval Partitioning

3.1 Introduction to Typical Notations

For the readability of this paper, some commonly used notations are listed in Table 1, and they are arranged according to its appearance order.

3.2 Analysis of IDE

3.2.1 Watermarking Embedding

For a given host data D represented by floating number, embedding position p $(p \geq 0)$, embedding strength s, and watermark w $(w \in \{0, 1, 2, \ldots, 2^s - 1\})$, the process of watermark embedding is depicted as follows.

Step 1. Adjust D by using p and s according to

$$D_r = (D \times 10^p)/2^s. \tag{1}$$

The integer part I and the decimal part F of D_r are obtained as

$$I = \mathrm{ITrim}(D_r), F = \mathrm{DTrim}(D_r), \tag{2}$$

where ITrim(·) and DTrim(·) are used to get the integer part and decimal part of D_r, respectively.

Step 2. Embed watermark into the integer part I according to

$$I^w = I \times 2^s + w \times \mathrm{sign}(D). \tag{3}$$

Step 3. Compute the host data D^w after watermark embedding as

$$D^w = (I^w + F) \times 10^{-p}. \tag{4}$$

According to Eqs. (1), (2), (3), and (4), the general function of the watermark embedding can be summarized as

$$D^w = \lfloor D/R \rfloor \times R + \frac{w \times R}{2^s} + \frac{D - \lfloor D/R \rfloor \times R}{2^s}, \tag{5}$$

where $R = 2^s \times 10^{-p}$ represents the length of interval. Just the same as IQIM [27], R also denotes the quantization interval, and it is partitioned into 2^s sub-intervals, which are also called quantization cells.

3.2.2 Watermarking Extraction and Host Recovery

Step 1. According to R calculated by p and s, the watermark is extracted by

$$w' = \lfloor D^w/R \times 2^s \rfloor - \lfloor D^w/R \rfloor \times 2^s. \tag{6}$$

Step 2. The host data is recovered as

$$D' = \lfloor D^w/R \rfloor \times R - w' \times R + (D^w - \lfloor D^w/R \rfloor \times R) \times 2^s. \tag{7}$$

3.2.3 Analysis of Distortion

The interval partition examples for $s = 1$ and $s = 2$ are illustrated in Figs. 1a and 1b, respectively. By using the uniform interval partition, the length of each sub-interval is equal and the length is fixed as $R/2^s$.

According to Eq. (5), it embeds the watermark by moving D to the corresponding watermark sub-interval. For any vertex $v(x, y)$, it can be embedded to obtain a watermarked

TABLE 1
Explanation of Notations

Notation	Explanation
D	the host data represented in type double
s	the embedding strength
p	the embedding position
D^w	the watermarked data
x^l	the left margin of the interval that x is in
x^r	the right margin of the interval that x is in
$d(\cdot,\cdot,\cdot,\cdot)$	the average distortion of each vertex in a single region
$d'(\cdot,\cdot,\cdot,\cdot)$	the normalized average distortion of each vertex in a single region
$\bar{d}(\cdot,\cdot)$	the normalized average distortion of all vertices in a single region
H	a proportion sequence represents the distribution of watermark
H^A	the ascending sequence of H
u	the random floating number between [a,b] (a=0, b=0.1)
τ	the error tolerance 10^{-6}
α	the random floating number between $[\mathrm{k}_1, \mathrm{k}_2]$ $(k_1 = 0.2, k_2 = 0.4)$
V	the original vertex sequence in Cartesian system
$\hat{V}$	the vertex sequence in polar system
$\hat{V}^a$	the vertex sequence in polar system with specific order
N_G	the number of vertices that can be embedded
$\widehat{V}^{aw}$	the watermarked vertex sequence in polar system
V^w	the watermarked vertex sequence in Cartesian system
V'^w	the received watermarked vertex sequence
$\widehat{V}''^w$	the watermarked vertex sequence after recovering
$\widehat{V}''^{aw}$	the watermarked vertex sequence with a specific order
$\widehat{V}''^a$	the recovered vertex sequence of $\widehat{V}''^{aw}$ in the polar system
V''	the recovered vertex sequence in the Cartesian system

vertex by using $g(\cdot)$, which is formulated as

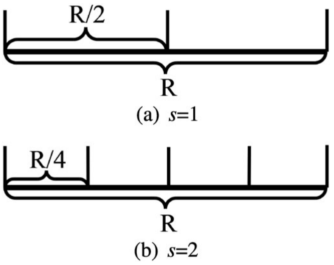
$$\begin{aligned}(x^w, y^w) &= g((x,y), R, s, w)\\ &= \left(wR/2^s + \frac{x - x^l}{2^s} + x^l, y\right),\end{aligned} \tag{8}$$

where x^l represents the left point of the interval that x is in, and $x^l = \lfloor x/R \rfloor \times R$. Meanwhile, set x^r as the right point of the interval and $x^r = (\lfloor x/R \rfloor + 1) \times R$, where $x \in [x^l, x^r)$. To guarantee the maximum distortion is controlled within R, x^w should meet the condition $x^w \in [x^l, x^r)$.

The maximum movement of the vertex in a single watermark region is

$$\begin{aligned}\underset{\mathrm{IDE}}{\mathrm{Maxd}}(R,s) &= \|(x', y) - \mathrm{g}(x', y, R, s, 2^s - 1)\|\\ &= R\left(1 - \frac{1}{2^s}\right)\\ &= (2^s - 1) \times 10^{-p},\end{aligned} \tag{9}$$

where $\|\cdot\|$ is used to calculate the euclidean distance between two different vertices.



(a) s=1

(b) s=2

Fig. 1. Examples of uniform and fixed interval partition.

Under the premise that the interval is evenly partitioned and the watermark information is evenly distributed, the average distortion of each vertex in a single watermark region can be represented as

$$d_{IDE}(x_i, y_i, R, s) = \frac{1}{2^s} \sum_{w=0}^{2^s-1} \|(x_i, y_i) - g((x_i, y_i), R, s, w)\|. \tag{10}$$

However, in the case that the watermark information is not completely evenly distributed, a proportion sequence $H(h(0), \ldots, h(2^s - 1))$ which is used to represent the distribution of watermark can be obtained, where $h(i)$ represents the proportion of the corresponding watermark value i. The average distortion of each vertex in a single watermark region is

$$d_{\mathrm{IDE}}(x_i, y_i, R, s) = \sum_{w=0}^{2^s-1} h(w) \|(x_i, y_i) - g((x_i, y_i), R, s, w)\|. \tag{11}$$

To fairly compare the distortion, the average distortion of each vertex in a single watermark region is normalized by

$$d'(x_i, y_i, 1, s) = \frac{d(x_i, y_i, R, s)}{\mathrm{Maxd}(R, s)} = \frac{d(x_i, y_i, 1, s)}{\mathrm{Maxd}(1, s)}. \tag{12}$$

The normalized average distortion of all vertices in a single watermark region is

$$\begin{aligned}\bar{d}(1,s) &=_{D=1\times 1} d'(m, n, 1, s) dmdn\\ &= \frac{1}{\mathrm{Maxd}(1,s)}_{D=1\times 1} d(m, n, 1, s) dmdn.\end{aligned} \tag{13}$$

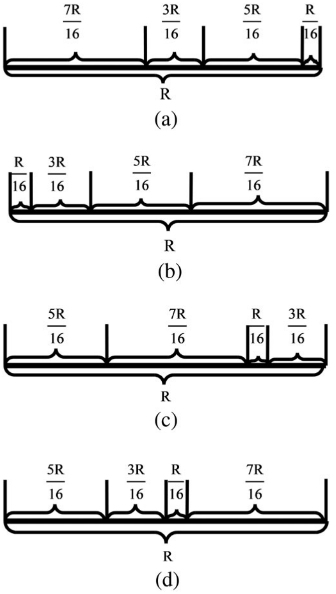


Fig. 2. Examples of adaptive interval partition.

According to Eq. (13), the imperceptibility is depended on the maximum distortion and d_{IDE}. Here, a theorem can be obtained.

Theorem 1. *Under the premise that the interval length* R *is fixed and the sub-intervals within each interval is in an ascending order, the theoretical maximum movement of a uniform interval partition based method is always less than or equal to that of an adaptive interval partition based method.*

Proof. proof Since the interval partition of the IDE-based method is essentially uniform, the minimum sub-interval of IDE is $R/2^s$, such that the theoretical maximum movement is $R(1-1/2^s)$ according to Eq. (9). Once the interval is adaptively partitioned, it can be easily obtained that the minimum sub-interval of which is always less than or equal to $R/2^s$, and the corresponding theoretical maximum movement is always larger than or equal to $R(1-1/2^s)$. □

Since the theoretical maximum movement of IDE is fixed, it is hard to further reduce the normalized distortion of the watermarked graphics by using IDE according to Theorem 1 and Eq. (13). Thus, an adaptive interval partition mode is proposed to match different watermark distributions in this paper.

3.3 The Proposed DE With Adaptively Interval Partitioning

For a given embedding strength s, supposing that the embedded watermark is not completely evenly distributed, 2^s sub-intervals are adaptively distributed in each interval, and the proportion of the sub-intervals are represented by a sequence $H(h(0),\ldots,h(2^s-1))$, where $0,1,2,\ldots,2^s-1$ are the indices of the sequence H. After that, H is sorted in an ascending order to obtain a sequence $H^A(h(\sigma(0)),\ldots,h(\sigma(2^s-1)))$, where σ:$\{0,1,2,\ldots,2^s-1\}\rightarrow\{0,1,\ 2,\ldots,2^s-1\}$ is a one-to-one mapping. The sum of the proportions of the sub-intervals in each interval satisfies

$$\sum_{i=0}^{2^s-1} h(i)=1. \tag{14}$$

Taking the embedding strength $s=2$ as an example, if the length of each sub-interval is not equal, there are $A_4^4=4\times3\times2\times1=24$ different permutations. Four of them are illustrated in Fig. 2. From Figs. 2a and 2b, it can be concluded that the theoretical maximum movement is $15R/16$ when the smallest sub-interval is located on the left or the right side of the interval. However, when the smallest sub-interval is not on both sides of the interval, as shown in Figs. 2c and 2d, the theoretical maximum movements are $13R/16$ and $11R/16$, respectively, which are smaller than the theoretical maximum $15R/16$. According to Eq. (13), when the average distortion of the vertices in the unit region is equal, the larger the theoretical maximum movement is, the better the imperceptibility is. Hence, as seen in Fig. 2b, the sub-intervals are sorted in an ascending order to ensure that the theoretical maximum distortion can be always obtained.

3.3.1 Watermarking Embedding

For a given data D represented by floating number and a specific embedding position p, the proportion sequence $H(h(0),\ldots,h(2^s-1))$ can be derived according to the watermark sequence W and the embedding strength s. The watermark embedding procedure is depicted as follows.

Step 1. With the mapping function σ and the watermark w_i, the index j of $h(w_i)$ in the sequence H^A can be calculated as

$$j=\sigma^{-1}(w_i), \tag{15}$$

where $w_i\in\{0,1,2,\ldots,2^s-1\}$ and $\sigma^{-1}(\cdot)$ represents the inverse mapping function of $\sigma(\cdot)$.

Step 2. Move the data D to the j^{th} sub-interval to embed watermark, and then the watermarked data D^w can be calculated as

$$\begin{aligned} D^w=&\lfloor D/R\rfloor\times R+\left(\sum_{i=0}^{j-1}h(\sigma(i))\right)\times R\\ &+h(\sigma(0))\times(D-\lfloor D/R\rfloor\times R), \end{aligned} \tag{16}$$

where $R=2^s\times10^{-p}$ represents the length of the interval. The relative distance of D in the interval is scaled with the minimum proportion value $h(\sigma(0))$ to ensure that the watermarked data D^w will not out of the range of the watermark sub-interval, and the watermark can be correctly extracted.

3.3.2 Watermarking Extraction and Host Recovery

Step 1. With a watermarked data D^w and R calculated by p and s, its relative distance r^w in the interval is calculated by

$$r^w=D^w-\lfloor D^w/R\rfloor\times R. \tag{17}$$

Step 2. Sort H in an ascending order to obtain its ascending sequence H^A, and calculate the sub-interval index j^A of r^w in H^A as

$$j^A=\mathrm{Index}\big(r^w,R,H^A\big), \tag{18}$$

where $\mathrm{Index}(\cdot,\cdot,\cdot)$ is a function to calculate the sub-interval index of r^w in the sequence H^A. The calculation of $\mathrm{Index}(\cdot,\cdot,\cdot)$ is described as follows: the value of k is first set

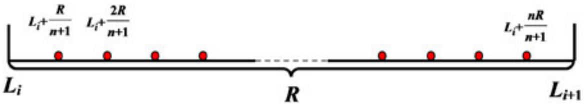


Fig. 3. Vertex distribution in a single region.

as 0, and then calculate the value of $R \times \sum_{i=0}^{k} h(\sigma(i))$. If $R \times \sum_{i=0}^{k} h(\sigma(i)) \leq r^w$, set $k = k+1$ until R satisfies the condition $R \times \sum_{i=0}^{k} h(\sigma(i)) > r^w$. Finally, the result of $\mathrm{Index}(\cdot,\cdot,\cdot)$ is k.

Step 3. According to j^A and the mapping function σ, the extracted watermark can be calculated by

$$w_i' = \sigma\big(j^A\big). \tag{19}$$

Step 4. The host data can be recovered as

$$\begin{aligned} D' =& \lfloor D^w/R \rfloor \times R \\ &+ \left(r^w - \left(\sum_{i=0}^{j^A-1} h(\sigma(i)) \times R \right) \right) / h(\sigma(0)). \end{aligned} \tag{20}$$

3.3.3 Analysis of Distortion

According to Eq. (16), given a vertex $v(x,y)$, it can be embedded to obtain the watermarked vertex $v^w(x^w, y^w)$, which is formulated as

$$\begin{aligned} (x^w, y^w) &= \mathrm{g}_{\mathrm{DEA}}((x,y), R, s, w) \\ &= (R \sum_{i=0}^{\sigma^{-1}(w)-1} h(\sigma(i)) \\ &\quad + (x - x')h(\sigma(0)) + x', y). \end{aligned} \tag{21}$$

According to the analysis in Section 3.2.3, the maximum movement of a single vertex is

$$\underset{\mathrm{DEA}}{\mathrm{Maxd}}(R, s) = R(1 - h(\sigma(0))). \tag{22}$$

When the interval is adaptively partitioned, the average distortion of each vertex in a single watermark region can be calculated by

$$d_{\mathrm{DEA}}((x,y), R, s) = \sum_{w=0}^{2^s-1} h(w) \|(x,y) - \mathrm{g}_{\mathrm{DEA}}((x,y), R, s, w). \tag{23}$$

As shown in Fig. 3, there are n vertices evenly distributed in each interval, the gap between adjacent vertices is $R/(n+1)$ according to the interval length R. If the left border of the i^{th} $(i \geq 1)$ interval is $L_i = (i-1) \times R$, the vertex set in each interval is $V_i = \{L_i + \frac{R}{n+1}, L_i + \frac{2R}{n+1}, \ldots, L_i + \frac{nR}{n+1}\}$.

According to Eq. (13), it can be found that $\bar{d}(1,s)$ is positive proportional to the average movement and negative proportional to the maximum movement of all vertices in a single watermark region. Here, theoretical analysis is made to the relationship between $\bar{d}_{IDE}(1,s)$ and $\bar{d}_{DEA}(1,s)$. An unevenly distributed watermark is represented as $c(0) : c(1) : c(2) : \ldots : c(2^s-1) = p_0 : p_1 : p_2 : \ldots : p_{2^s-1}$, where $c(\cdot)$ represents a function to record the number of the corresponding watermark values and $p_0 < p_1 < \ldots < p_{2^s-1}$ $(p_i > 0, 0 \leq 2^s - 1)$.

The sum of the average distortion of all vertices in a single watermark region for IDE is

$$\begin{aligned} \underset{\mathrm{IDE}}{\mathrm{SumD}} = \sum_{j=1}^{n} \Bigg(& h(0) \times \left| -\frac{2^s-1}{2^s} \times \frac{jR}{n+1} \right| \\ & + h(1) \times \left| \frac{R}{2^s} - \frac{2^s-1}{2^s} \times \frac{jR}{n+1} \right| \\ & + \cdots + h(2^s-1) \\ & \times \left| \frac{(2^s-1)R}{2^s} - \frac{2^s-1}{2^s} \times \frac{jR}{n+1} \right| \Bigg). \end{aligned} \tag{24}$$

where $h(i) = p_i / \sum_{j=1}^{2^s-1} p_j$ and $i \in [0, 2^s-1]$.

As for the proposed method, it can be calculated as

$$\begin{aligned} \underset{\mathrm{DEA}}{\mathrm{SumD}} = \sum_{j=1}^{n} \Bigg(& h(0) \times \left| -(1-h(0)) \times \frac{jR}{(n+1)} \right| + h(1) \\ & \times \left| p_0 R / \sum_{i=0}^{2^s-1} p_i - (1-h(0)) \times \frac{jR}{(n+1)} \right| + \cdots + h(2^s-1) \\ & \times \left| \left(1 - p_{2^s-1} / \sum_{i=0}^{2^s-1} p_i \right) \times R - (1-h(0)) \times \frac{jR}{(n+1))} \right| \Bigg). \end{aligned} \tag{25}$$

Specifically, when the embedding strength is set as $s = 1$ and the embedded watermark is unevenly distributed, set $c(0) : c(1) = p_0 : p_1 (p_0 < p_1)$. According to Eq. (11), the sum of the average distortion of all vertices in a single watermark region of IDE is

$$\begin{aligned} \underset{\mathrm{IDE}}{\mathrm{SumD}} = \sum_{j=1}^{n} \Bigg(& \frac{p_0}{p_0+p_1} \times \frac{jR}{2(n+1)} \\ & + \frac{p_1}{p_0+p_1} \times \frac{(n+1-j)R}{2(n+1)} \Bigg). \end{aligned} \tag{26}$$

Similarly, according to Eq. (23), the average distortion of each vertex in a single watermark region for the proposed method is

$$\begin{aligned} \underset{\mathrm{DEA}}{\mathrm{SumD}} = \sum_{j=1}^{n} \Bigg(& \frac{p_0}{p_0+p_1} \times \frac{p_1 jR}{(p_0+p_1)(n+1)} \\ & + \frac{p_1}{p_0+p_1} \times \frac{(p_0(n+1) - p_1 j) \times R}{(p_0+p_1)(n+1)} \Bigg). \end{aligned} \tag{27}$$

From the above analysis, we can obtain the following theorem.

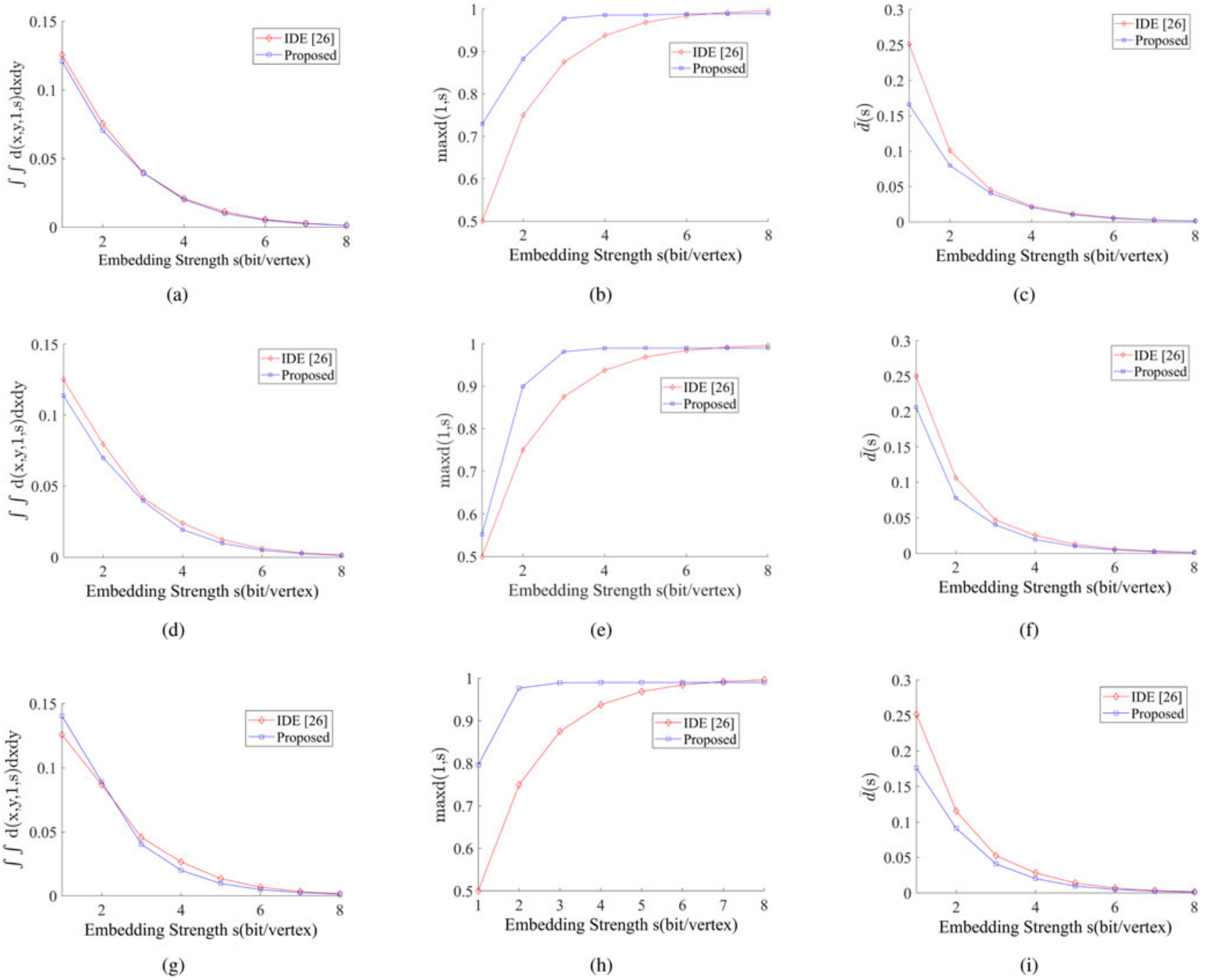
Theorem 2. *When the embedding strength is set as $s = 1$ and the watermark information is not completely evenly distributed, the average distortion of each vertex in a single region for the proposed method is smaller than that of IDE.*

Proof. proof According to Eqs. (26) and (27), the difference between $\mathrm{SumD}_{\mathrm{DEA}}$ and $\mathrm{SumD}_{\mathrm{IDE}}$ is

$$\begin{aligned} \underset{\mathrm{IDE}}{\mathrm{SumD}} - \underset{\mathrm{DEA}}{\mathrm{SumD}} = & \left(\frac{1}{2} - \frac{p_0}{p_0+p_1} \right) \\ & \times \frac{Rp_1}{(n+1) \times (p_0+p_1)}. \end{aligned} \tag{28}$$

□

Since $p_0 < p_1$, it can be easily concluded that $\frac{p_0}{p_0+p_1} < \frac{1}{2}$, thus the difference between them is always greater than 0. That is, $\mathrm{SumD}_{\mathrm{DEA}} < \mathrm{SumD}_{\mathrm{IDE}}$ always holds. □



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 4. Analysis of distortion: (a)-(c): distortion analysis when $\gamma = 2$; (d)-(f): distortion analysis when $\gamma = 4$; (g)-(i): distortion analysis when $\gamma = 8$.

Since it is difficult to directly compare the difference between Eqs. (24) and (25) for different s, a simulation is performed on Matlab 2014a with a step of 0.01. For a given s, $2^s - 1$ sampling points are randomly generated based on the power-law function and the random function, and the gamma differences between the adjacent sampling points are regarded as the proportion data of the watermark information. The proposed method is compared with IDE with different γ, where different γ represent different gamma curves, that is, different watermark information distributions. The experimental results are shown in Fig. 4. According to the results, the theoretical maximum distortion of the proposed algorithm is always larger than that of IDE under the same condition. Although the average distortion of the proposed algorithm is not necessarily less than that of IDE, the normalized average distortion of the proposed method is always lower than that of IDE, which indicates the superior imperceptibility of the proposed method.

To better verify the relationship between $\bar{d}_{\text{DEA}}(1, s)$ and $\bar{d}_{\text{IDE}}(1, s)$ when $s = 1$, simulation is also conducted in the MATLAB R2014a. Here, the proportion of the watermark value 0 is in the interval $[0, 0.5]$ with a step of 0.0001. Three indicators average distortion, maximum distortion and normalized average distortion of all vertices in a single watermark region are compared, and the experimental results are illustrated in Fig. 5. It can be found that a higher maximum distortion and a lower normalized average distortion can be achieved for the proposed method compared with those of IDE, which illustrates the effectiveness of the adaptive interval partition in reducing distortion.

4 The Proposed Scheme

The flowchart of the proposed scheme is illustrated in Fig. 6. In the watermark embedding, for a given 2D engineering graphics G, it is first tranformed from rectangle coordinate system to a polar coordinate system. After that, the number of eligible vertices N_G is obtained, and the watermark length L is calculated based on N_G and an embedding strength s. The watermark sequence W is generated by AES-CTR with a key K_H. Finally, W is embedded into the eligible vertices by using the proposed method, and the watermarked graphics is transformed to the original coordinate system and get G^w. During the authentication and recovery, for the received watermarked graphics G^w, it is first transformed to polor coordinate system, and then the watermark

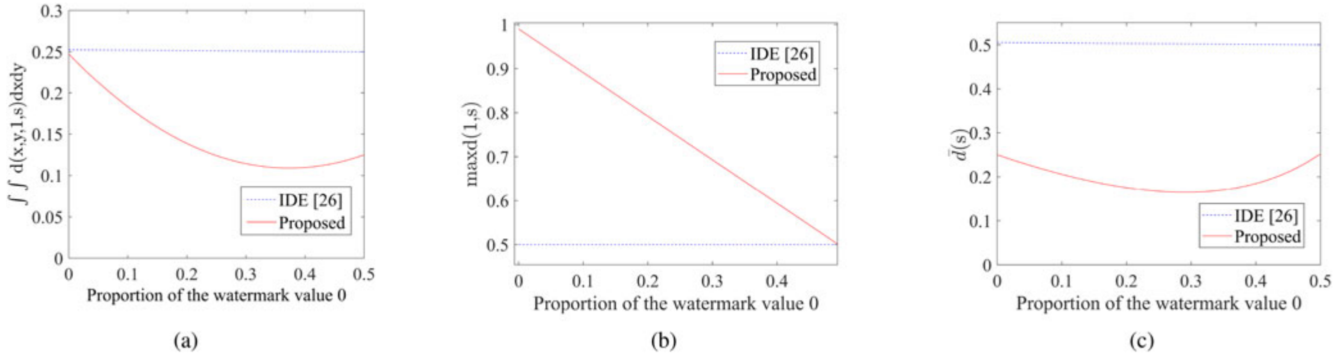


Fig. 5. Simulation results. (a) the average movement, (b) the maximum movement, and (c) the normalized average movement of all vertices in a unit region of different methods.

W' is extracted from it. Just the same as watermark embedding, the watermark sequence W is generated by AES-CTR with the key K_H. If W and W' are mismatched, the graphics is failed to pass the authentication; otherwise, it means that the graphics pass the authentication, and the original host data can be recovered by the proposed method. After transforming the recovered host data to the original coordinate system, a recovered graphics G' can be obtained. In the following, watermark generation, watermark embedding, watermark extraction, graphics authentication and recovery are depicted in details.

4.1 Watermarking Generation

For simplicity, the details of the coordinate system transformation and the determination of the eligible vertices are described in watermarking embedding.

Step 1. For the given 2D engineering graphics G, it is first transformed from rectangle coordinate system to a polar coordinate system with two reference vertices.

Step 2. Traverse the vertices of the graphics and get the number of the vertices N, and obtain the number of the eligible vertices N_G.

Step 3. With a watermark strength s, the length of the watermark is $L = N_G \times s$.

Step 4. With a preset initial vector IV and a key K_H with a size of 128 bits, a random information sequence B is generated by AES-CTR, and the length of B is $(\lfloor L/128 \rfloor + 1) \times 128$ bits.

Step 5. The watermark sequence is obtained as

$$W = LTrim(B, L), \tag{29}$$

where $LTrim(s, a)$ is a function to trim a bits from the sequence s from the left.

4.2 Watermarking Embedding

Step 1. For the given 2D engineering graphics G, the vertices are first extracted and a vertex sequence V=$\{v_i | v_i = (x_i, y_i), i \in \{0, 1, \ldots, N-1\}\}$ is obtained. After that, choose a vertex pair $(v_{m1}(x_{m1}, y_{m1}), v_{m2}(x_{m2}, y_{m2}))$ that has the farthest distance, and calculate its mid-point $O(x_o, y_o)$ as

$$\begin{cases} x_o = (x_{m1} + x_{m2})/2 \\ y_o = (y_{m1} + y_{m2})/2. \end{cases} \tag{30}$$

Move v_{m1} outward with a length $u\tau$ along the direction of $\overrightarrow{ov_{m1}}$, and obtain $v'_{m1}(x'_{m1}, y'_{m1})$ according to

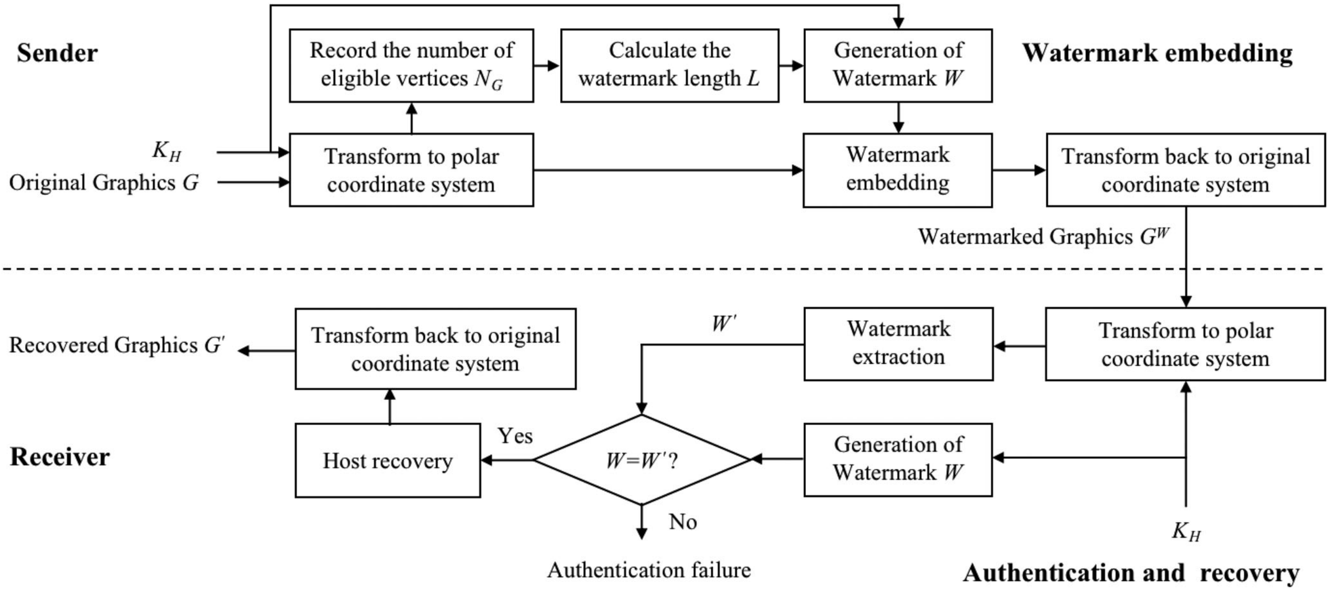

Sender
Record the number of eligible vertices N_G
Calculate the watermark length L
Generation of Watermark W
Watermark embedding
K_H
Original Graphics G
Transform to polar coordinate system
Watermark embedding
Transform back to original coordinate system
Watermarked Graphics G^W
Recovered Graphics G'
Transform back to original coordinate system
W'
Watermark extraction
Transform to polar coordinate system
Receiver
Host recovery
Yes
W=W'?
Generation of Watermark W
No
Authentication failure
K_H
Authentication and recovery

Fig. 6. Flowchart of the proposed scheme.

$$\begin{cases} x'_{m1} = \frac{\|ov_{m1}\|+u\tau}{\|ov_{m1}\|}(x_{m1} - x_o) + x_o \\ y'_{m1} = \frac{\|ov_{m1}\|+u\tau}{\|ov_{m1}\|}(y_{m1} - y_o) + y_o, \end{cases} \tag{31}$$

where u is a random floating number between $[a, b]$ $(a \leq b)$, and it is generated by a random function (for example, function $rand()$ in C++) with a seed of K_H. In the same way, v_{m2} is adjusted to obtain $v'_{m2}(x'_{m2}, y'_{m2})$. After adjustment, the farthest vertex pair is unique, so the search of the farthest vertices pair can get rid of the vertex traversal order. Thus, it can achieve semi-fragility to vertices rearrangement.

Step 2. The distance between v'_{m1} and v'_{m2} is calculated, and it is regarded as the reference distance d. A random floating number α between $[k_1, k_2]$ is first generated by a random function with the seed of K_H. After that, the number of vertices N_{m1} and N_{m2} around v'_{m1} and v'_{m2} is calculated in the distance range of $(0, \alpha d]$, respectively. Finally, the vertex that has more surrounding vertices is selected as the reference vertex v_{r1}, while the other is the reference vertex v_{r2}.

Step 3. The original point and the polar axis are set as o and ov_{r1}, respectively. For each vertex v_i in V, it is transformed to a polar coordinate system according to

$$\begin{cases} \hat{\rho}_i = \|ov_i\| \\ \hat{\theta}_i = \mathrm{sign}(\overrightarrow{ov_{r1}} \times \overrightarrow{ov_i} \cdot \overrightarrow{e_z}) \\ \qquad \times \arccos\left(\frac{\|ov_i\|^2+\|ov_{r1}\|^2-\|v_{r1}v_i\|^2}{2\|ov_i\|\times\|ov_{r1}\|}\right) \mathrm{if} \|\overrightarrow{ov_{r1}} \times \overrightarrow{ov_i}\| \neq 0 \\ \hat{\theta}_i = \arccos\left(\frac{\|ov_i\|^2+\|ov_{r1}\|^2-\|v_{r1}v_i\|^2}{2\|ov_i\|\times\|ov_{r1}\|}\right) \mathrm{if} \|\overrightarrow{ov_{r1}} \times \overrightarrow{ov_i}\| = 0, \end{cases} \tag{32}$$

where $\overrightarrow{e_z}$ is the unit vector of Z-axis. In this way, a vertex sequence $\widehat{V} = \{\widehat{v}_i | \widehat{v}_i = (\widehat{\rho}_i, \widehat{\theta}_i), i \in \{0, 1, \ldots, N-1\}\}$ in the polar coordinate system can be obtained.

Step 4. All vertices in $\widehat{V}$ are sorted based on the angles in an ascending order. If the angles are equal, they are sorted via the polar diameters in an ascending order. In this way, a vertex sequence $\widehat{V}^a = \{\widehat{v}_i^a | \widehat{v}_i^a = (\widehat{\rho}_i^a, \widehat{\theta}_i^a), i \in \{0, 1, \ldots, N-1\}\}$ in a specific order can be obtained.

Step 5. For simplicity, the length of the interval is set as $R = d/\lceil d/\tau \rceil$ and the initial number of eligible vertices that can be embedded is $N_G = N - 2$ (excluding two reference vertices). To ensure the correctness of the watermark extraction, all vertices in $\widehat{V}^a$ are traversed in order. Watermark cannot be embedded to the vertices satisfying the following situations.

Situation 1. If there exists a vertex $\widehat{V}_i^a$ in the same polar diameter interval with the reference vertex $\widehat{V}_{r1}^a$, namely $\lfloor \widehat{\rho}_i^a / R \rfloor = \lfloor d/(2R) \rfloor$, it is a non-embeddable vertex, and the number of eligible vertices is updated as $N_G = N_G - 1$.

Situation 2. If two different vertices $\widehat{V}_i^a$ and $\widehat{V}_j^a$ with the same angle value are located in the same polar diameter interval, and it satisfies $\widehat{\theta}_i^a = \widehat{\theta}_j^a$, $\lfloor \widehat{\rho}_i^a / R \rfloor = \lfloor \widehat{\rho}_j^a / R \rfloor$ and $\lfloor \widehat{\rho}_i^a / R \rfloor \ / = \lfloor d/(2R) \rfloor$, all vertices in the polar diameter interval are set to be non-embeddable. If there are c_i vertices in the polar diameter interval, the number of eligible vertices is updated as $N_G = N_G - c_i$.

The purpose of setting the vertices in the same interval to be non-embeddable is to prevent the change of the specific ordering in Step 4, which may lead to errors in the watermark extraction.

Step 6. Generate a watermark sequence W according to Section 4.1, and obtain its proportion sequence H. According to Section 3.3.1), W is embedded into the polar diameters of the eligible vertices in $\widehat{V}^a$ to obtain the sequence $\widehat{V}^{aw} = \{\widehat{v}_i^{aw} | \tilde{v}_i^{aw} = (\widehat{\rho}_i^{aw}, \widehat{\theta}_i^{avv}), i \in \{0, 1, 2, \ldots, N-1\}\}$.

Step 7. All coordinates in $\widehat{V}^{aw}$ are transformed to the original coordinate system, and then the watermarked vertex sequence $\widehat{V}^w$ in the Cartesian system is obtained by

$$\begin{cases} x_i = \widehat{\rho}_i^{aw} \cos\left(\hat{\theta}_i^{aw}\right) + x_o \\ y_i = \widehat{\rho}_i^{aw} \sin\left(\widehat{\theta}_i^{aw}\right) + y_0. \end{cases} \tag{33}$$

Step 8. $\widehat{V}^w$ is updated to the graphics and obtain the watermarked graphics $\widehat{G}^w$. Through a secure channel, the key K_H, sequence H, reference distance d, and embedding strength s can be transmitted to valid users.

4.3 Watermarking Extraction, Graphics Authentication and its Analysis

Step 1. For the watermarked graphics $\widehat{G}'^w$, the vertex sequence $\widehat{V}'^w$ is first obtained. After that, the farthest vertex pair $(v'_{m1}(x'_{m1}, y'_{m1}), v'_{m2}(x'_{m2}, y'_{m2}))$ can be quickly located and the mid-point $O'(x'_o, y'_o)$ is calculated, and then the reference distance d' can be acquired. After that, a random floating number α is generated by the random function with the seed of K_H. Calculate the reference vertex v'_{r1} and construct the polar axis according to Section 4.2. By using Eq. (32), all vertices are transformed to the polar coordinate system. After that, all polar diameters are restored to obtain $\widehat{V}''^w$ according to

$$\widehat{\rho}_i''^w = \widehat{\rho}_i'^w d/d'. \tag{34}$$

Step 2. All vertices in $\widehat{V}''^w$ are sorted according to Section 4.2. After that, the vertex sequence $\widehat{V}''^{aw}$ with a specific order can be obtained.

Step 3. According to Section 3.3.2), the watermark is extracted from the eligible vertices in the vertex sequence $\widehat{V}''^{aw}$, and the watermark segment $\widehat{w}'_i$ is obtained. Subsequently, the polar diameters of the eligible vertices are restored by Eq. (20), and the vertex sequence $\widehat{V}''^a$ is obtained.

Step 4. After all the watermark bits are extracted, an extracted watermark sequence W' is obtained. With the key K_H, the original watermark sequence W can be re-generated by AES-CTR according to Section 4.1.

Step 5. If the extracted watermark sequence W' and the original watermark sequence W are mismatched, the authentication is failed; otherwise, it passes the authentication and goes to Step 6 to recover the original graphics.

Step 6. All vertices in $\widehat{V}''^a$ are transformed to the original coordinate system according to Eq. (33), and the recovered vertex sequence $\widehat{V}''$ is obtained.

Step 7. A random floating number u' is generated, and the reference vertex v'_{r1} is moved according to

$$\begin{cases} x''_{r1} = \frac{\|o'v'_{r1}\|-u'\tau}{\|o'v'_{r1}\|}\left(x'_{r1} - o'_x\right) + o'_x \\ y''_{r1} = \frac{\|o'v'_{r1}\|-u'\tau}{\|o'v'_{r1}\|}\left(y'_{r1} - o'_y\right) + o'_y. \end{cases} \tag{35}$$

TABLE 2
The Basic Information of Experimental Material

Graphics	Vertices number	Feature number	Cover size(KB)	Watermark size(bit)
G1	219	172	53	430
G2	1281	871	44	2554
G3	3925	2357	150	7838
G4	5522	3385	382	11028
G5	9483	5527	479	18942
G23	135	108	46	262
G52	31385	10515	1306	62598
Average of 300 graphics	3080.27	1595.21	148.78	3079.27

In this way, the recovered reference vertex v''_{r1} can be obtained. Similarly, another reference vertex v'_{r2} can be moved to obtain the recovered reference vertex v''_{r2}.

Step 8. All recovered vertices are updated to the graphics, and the recovered 2D engineering graphics G'' is obtained.

5 Experimental Results and Analysis

5.1 Experimental Results

The configuration of PC in the experiments are: CPU i5-7500 3.40GHz, RAM 8G, Windows 10 64bit, Teigha-vc14 Libraries, Visual Studio 2015, AutoCAD 2013. Here, the embedding position *p* is not fixed, and the value of *R* is directly set. The experimental parameters are: error tolerance $\tau = 10^{-6}$, embedding strength $s = 2$, adjustment parameter $a = 0$, $b = 0.1$, and the reference vertices selection parameter $k_1 = 0.2$, $k_2 = 0.4$. Since there does not exist publicly avalaible dataset on 2D engineering graphics, 300 2D engineering graphics are utilized for experiments, and they are obtained from multiple websites. Among them, *G*23 and *G*25 are the graphics with the minimum vertices and the maximum vertices, respectively. The details of these graphics are listed in Table 2. Five sample graphics are illustrated in Figs. 7a, 7b, 7c, 7d, and 7e, the corresponding watermarked graphics are illustrated in Figs. 7f, 7g, 7h, 7i, and 7j, and Figs. 7k, 7l, 7m, 7n, and 7o are the recovered graphics.

For evaluating the distortion between the original graphics and the processed graphics, the average distortion *AvgD* and the maximum distortion *MaxD* are utilized as indices to measure its difference. Furthermore, the correlation coefficient (*NC*) is used to measure the similarity between the original watermark and the extracted watermark. They are formulated as

$$\mathrm{AvgD}(P,Q) = \frac{1}{N}\sum_{i=0}^{N-1}\|p_i - q_i\|, \tag{36}$$

$$\mathrm{MaxD}(P,Q) = \max(\|p_i - q_i\|), i \in \{0,1,\ldots,N-1\}, \tag{37}$$

$$\mathrm{NC}(A,B) = \frac{\sum_{i=0}^{L-1}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=0}^{L-1}(a_i - \bar{a})^2}\sqrt{\sum_{i=0}^{L-1}(b_i - \bar{b})^2}}, \tag{38}$$

where *P* and *Q* represent the vertex sequences of the graphics before and after watermark embedding, respectively, *N* denotes the total number of vertices, and *L* represents the watermark length.

For all 300 graphics, with three different watermark distributions, an average $AvgD \approx 2.8657 \times 10^{-7}$ and an average $MaxD \approx 8.0957 \times 10^{-7}$ are obtained between the original graphics and the watermarked ones. Moreover, an average $AvgD \approx 8.6572 \times 10^{-14}$ and an average $MaxD \approx 78.3786 \times 10^{-14}$ can be obtained between the original graphics and the recovered ones. The average *NC* between the original watermark and the extracted watermark is 1. The results indicate the effectiveness of the proposed scheme.

5.2 Performance Analysis

5.2.1 Analysis of Reversibility

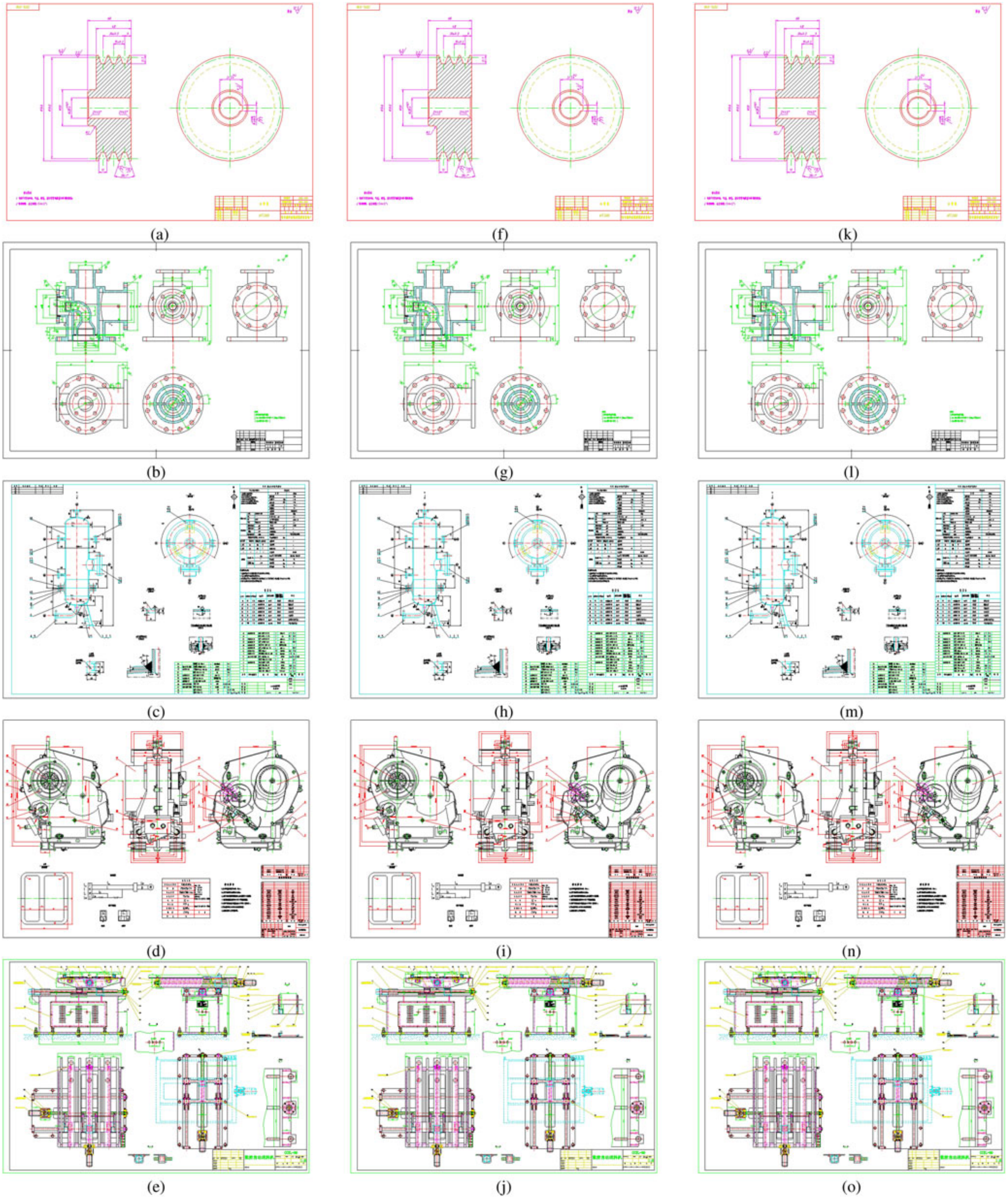
Based on the principle, all vertices are partitioned into non-reference vertices and reference vertices v_{r1}, v_{r2}. For two reference vertices, they are moved outward with the length of $u\tau$ based on its distance to the midpoint *O*. Since two adjusted reference vertices must be the unique farthest vertices pair, it can be found out again during the watermark extraction and authentication process, and the euclidean distance from the two reference vertices to the midpoint can be calculated again. By moving two reference vertices in the opposite directions, their original positions can be recovered.

For any non-reference vertex v_i, its new coordinate $\widehat{V_i}$ can be calculated by Eq. (32), and the watermark $\widehat{w_i}$ is embedded to obtain $\widehat{V_i^w}$ by Eq. (16). By using Eq. (33), $\widehat{V_i^w}$ is transformed to the original coordinate system and obtain V_i^w. During the extraction and authentication process, V_i^w is transformed to get $\widehat{V_i'^w}$ in the polar coordinate system. As Eqs. (32) and (33) are inverse, $\widehat{V_i^w}$=$\widehat{V_i'^w}$ can always be obtained. Afterwards, $\widehat{V_i'^w}$ is restored to $\widehat{V_i'}$ by using Eq. (20). By combining Eqs. (16) and (20), it can be concluded that $\widehat{V_i'}$=$\widehat{V_i}$ holds. At last, $\widehat{V_i'}$ is transformed to the Cartesian coordinate system by Eq. (33). In this way, v_i' is obtained. Based on the above analysis, $v_i' = v_i$ always maintains.

To further validate the reversibility of the proposed scheme, the difference between the original graphics and the recovered graphics is calculated, and the experimental results are listed in Table 3. According to the results, the average *AvgD* between the original graphics and the recovered graphics is smaller than 10^{-13}, and the average *MaxD* is smaller than 10^{-12}. As indicated in [26], if the difference between two different graphics is less than 10^{-8}, they are completely identical. The above results illustrate the good reversibility of the proposed scheme.

5.2.2 Analysis of Imperceptibility

Here, experiments are made to some existing methods [27], [29], [30], [31], [32] and the proposed scheme to compare the imperceptibility. For a fair comparison, the parameters of IQIM-A [27] are set as $\Delta = 10^{-6}/(2^2 - 1)$, $b = 2$; the parameters of VC [29] are set as $\tau = 10^{-6}$, $c_1 = c_2 = 1$; the parameters of RN [30] are set as $\tau = 10^{-6}$, $n = 2$, $r = 0.999$; the parameter of ECC [31] and RI [32] are set as $p = 6$, $s = 2$, where *p* is embedding position and *s* is embedding strength. Experiments with different watermark distributions are done to the graphics dataset, and the ratios between the number of bit '0' and bit '1' are 0.5, 1, and 1.5, respectively. The experimental results are listed in Tables 4, 5, and 6. When the ratio is 0.5, the average distortion of the proposed



(a) (f) (k)

(b) (g) (l)

(c) (h) (m)

(d) (i) (n)

(e) (j) (o)

Fig. 7. Experiment results of some sample 2D engineering graphics:(a)-(e): Original graphics; (f)-(j): Watermarked graphics; (k)-(o): Recovered graphics.

sheme is improved by 28.97%, 33.94% and 8.52%, and the maximum distortion is improved by 13.38%, 11.84%, and 7.80% compared with IQIM-A, VC and RN, respectively. When the ratio is 1, the average distortion is improved by 2.219%, 32.89%, and 5.629% and the maximum distortion is improved by 0.565%, 23.96%, and 20.87%, compared with IQIM-A, VC and RN, respectively. When the ratio is 1.5, the average distortion is improved by 15.66%, 30.89%, and 13.42%, and the maximum distortion is improved by 8.116%, 16.86%, and 10.40% compared with IQIM-A, VC and RN, respectively. According to the results, as the distortion of ECC [31] and RI [32] are not controlled by error

TABLE 3
Analysis of the Difference Between the Original Graphics and the Recovered Graphics ($\times 10^{-14}$)

Graphics	*AvgD*	*MaxD*
G1	5.6154	17.0678
G2	9.9457	45.8286
G3	11.9574	98.1317
G4	7.2559	23.3657
G5	3.3549	37.4994
G23	1.6483	5.2253
G52	13.0185	38.4692
Average of 300 graphics	8.6572	78.3786

tolerance τ, their distortions are much larger than those of the proposed sheme. While for the proposed sheme, it can achieve better imperceptibility compared with the state-of-the-art schemes in [27], [29], [30], [31], [32], which indicates its good imperceptibility.

5.2.3 Analysis of Semi-Fragility

Under the premise that the farthest vertices pair of the graphics is not tampered, as the relative position of each vertex in the graphics does not change, the embedded watermark will not change as well. The normal operations for 2D engineering graphics contain RST and vertices rearrangement. Since global scaling operation simply affects the reference distance without influencing the relation of vertices in the graphics, the original reference distance can be restored by Eq. (34), and then the watermark can be normally extracted. Similarly, it can also resist global translation and rotation. In addition, since it first sorts the polar angles and uses the polar diameters of the vertices to conduct watermark embedding and extraction, the proposed sheme does not depend on the vertex traversal order. Thus, it can resist the vertices/entities reordering attack.

To test the semi-fragility of the proposed sheme, experiments are done to compare the semi-fragility of different methods. Watermark is first embedded into the 2D engineering graphics in the dataset. After the watermarked graphics are processed by normal operations, the watermark is extracted from them. The normal operations include: rotation (rotation angle $\rho \in \{1^\circ, 5^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ\}$), translation (translation vector $\vec{v} \in \{(-6.3, -3.9), (-3.7, 5.6), (9.2, 4.3), (7.8, -8.6), .(0, 12.3), (16.8, 0)\}$), scaling (scaling factor $\zeta \in \{0.1, 0.35, 0.5, 1.25, 2.0, 2.8\}$), and inverse vertex order. *NC* coefficient is utilized here to measure the similarity between the extracted watermark and the embed-

TABLE 4
Distortion Comparisons Between Different Methods When the Ratio is 0.5 ($\times 10^{-7}$)

Graphics	IQIM-A[27]		VC[29]		RN[30]		ECC[31]		RI[32]		Proposed	
	AvgD	*MaxD*	*AvgD*	*MaxD*	*AvgD*	*MaxD*	*AvgD*	*MaxD*	*AvgD*	*MaxD*	*AvgD*	*MaxD*
G1	3.7225	9.9999	4.8845	9.9202	3.0223	9.4892	42.734	63.253	40.425	90.233	2.6679	8.5791
G2	3.9398	9.9985	4.2039	9.7416	3.2194	9.6959	52.993	61.274	74.642	86.757	2.7059	8.5860
G3	3.9686	9.9939	4.5289	9.8882	3.0191	9.5007	77.596	79.634	63.534	77.534	2.8934	8.8323
G4	4.0479	10.000	4.4060	9.9536	3.0269	9.4916	47.934	72.835	43.535	57.725	2.8619	8.8523
G5	4.1269	9.9999	4.4463	9.9234	3.0961	9.2923	55.266	67.237	86.334	96.734	2.8485	8.8354
Average of 300 graphics	3.9451	9.9769	4.3741	9.8524	3.0633	9.3734	78.667	81.099	75.239	81.462	2.8023	8.6420

TABLE 5
Distortion Comparisons Between Different Methods When the Ratio is 1 ($\times 10^{-7}$)

Graphics	IQIM-A[27]		VC[29]		RN[30]		ECC[31]		RI[32]		Proposed	
	AvgD	*MaxD*	*AvgD*	*MaxD*	*AvgD*	*MaxD*	*AvgD*	*MaxD*	*AvgD*	*MaxD*	*AvgD*	*MaxD*
G1	2.8673	7.3785	4.3433	9.7364	2.8501	8.2452	63.532	69.233	47.239	66.293	2.7352	7.2454
G2	2.9907	7.4817	4.2354	9.8740	3.0809	9.6343	54.922	61.337	51.805	63.644	2.8683	7.4877
G3	2.9343	7.4991	4.2668	9.8485	3.0882	9.7202	45.295	73.244	41.665	69.251	2.9062	7.5770
G4	2.9363	7.5000	4.1697	9.8730	3.0464	9.7809	67.385	81.392	67.359	74.297	2.9194	7.5986
G5	2.9146	7.8314	4.4309	9.9727	3.1064	9.9797	72.868	76.364	57.331	83.266	2.8886	7.5687
Average of 300 graphics	2.9286	7.5381	4.2674	9.8577	3.0344	9.4720	65.395	75.357	57.341	64.347	2.8636	7.4955

TABLE 6
Distortion Comparisons Between Different Methods When the Ratio is 1.5 ($\times 10^{-7}$)

Graphics	IQIM-A[27]		VC[29]		RN[30]		ECC[31]		RI[32]		Proposed	
	AvgD	*MaxD*	*AvgD*	*MaxD*	*AvgD*	*MaxD*	*AvgD*	*MaxD*	*AvgD*	*MaxD*	*AvgD*	*MaxD*
G1	3.2886	7.4997	4.1743	9.6584	3.4537	7.7125	76.558	79.353	58.345	75.341	2.8598	8.3883
G2	3.1141	8.4985	4.2998	9.7133	3.3573	8.7472	47.935	74.387	39.346	58.375	2.9311	8.3687
G3	3.1322	8.4991	4.4403	9.8994	3.3824	9.4587	63.383	66.367	53.634	68.744	2.9605	8.2971
G4	3.1131	7.4971	4.4373	9.9343	3.3539	9.5805	58.449	65.548	49.394	75.395	2.9381	8.3392
G5	3.7377	8.9348	4.4082	9.9492	3.3795	9.9797	74.388	87.338	63.865	77.398	2.9662	8.3548
Average of 300 graphics	3.4754	8.8695	4.2419	9.8023	3.3854	9.0957	67.472	78.428	59.385	75.348	2.9312	8.1496

TABLE 7
Analysis of the Semi-Fragility of Different Methods on an Average of 300 Graphics (*NC*)

operations	IQIM-A[27]	VC[29]	RN[30]	ECC[31]	RI[32]	Proposed
rotation($\rho = 1°$)	1	-0.0322	1	1	1	1
rotation($\rho = 5°$)	1	0.0026	1	1	1	1
rotation($\rho = 45°$)	1	0.0031	1	1	1	1
rotation($\rho = 90°$)	1	-0.0037	1	1	1	1
rotation($\rho = 135°$)	1	-0.0173	1	1	1	1
rotation($\rho = 180°$)	1	0.0114	1	1	1	1
translation (-6.3,-3.9)	1	-0.0109	1	1	1	1
translation (-3.7,5.6)	1	0.0506	1	1	1	1
translation (9.2,4.3)	1	0.0166	1	1	1	1
translation (7.8,-8.6)	1	-0.0389	1	1	1	1
translation (0,12.3)	1	-0.0327	1	1	1	1
translation (16.8,0)	1	0.0072	1	1	1	1
scaling (ζ = 0.1)	1	-0.0063	1	-0.0064	0.0058	1
scaling (ζ = 0.35)	1	0.0487	1	0.0052	0.0067	1
scaling (ζ = 0.5)	1	0.0032	1	-0.0026	0.0215	1
scaling (ζ = 1.25)	1	0.0725	1	0.0074	-0.0104	1
scaling (ζ = 2.0)	1	-0.0266	1	-0.0274	0.0447	1
scaling (ζ = 2.80)	1	-0.0624	1	-0.0352	-0.0062	1
inverse vertex order	-0.0326	1	-0.1097	-0.0048	0.0265	1

ded watermark. According to Eq. (38), the closer the *NC* value to 1 is, the more similar the two watermarks are. On the contrary, the closer the NC value to 0 is, the more different the two watermarks are. The experimental results are shown in Table 7. According to the results, it can be found that IQIM-A[27] and RN[30] all can resist against RST operations but cannot resist against inverse vertex order operation. VC[29] can resist against reordering operation but cannot resist against RST operations. ECC[31] and RI[32] can resist against rotation and translation operation. From Table 7, the proposed scheme can resist against both RST and inverse vertex order operations, which indicates the good semi-fragility of the proposed scheme.

5.2.4 Analysis of Security

Currently, most existing reversible watermark schemes based on uniform interval partition cannot get rid of the dependence on the vertex traversal order. If the interval length *R* is intercepted by a malicious attacker, he/she can exhaustively enumerate the embedding strength *s*. As the length of each sub-interval is $R/2^s$, the embedded information is confronted with the risk of leakage. In the proposed scheme, adaptively partitioning intervals can ensure the security of the embedded watermark.

It is assumed that the proposed scheme and parameters are public. If an attacker wants to perform a successful tampering, he should guarantee that the extracted watermark can pass the authentication. If the attacker does not know the distribution of the watermark information when the watermark is embedded into all the eligible vertices, the probability of correct watermark extraction is about $(1/2^s)^{n-2}$. If the embedding strength *s* is set to 2, the probability of correct watermark extraction is close to 0 when the number of the vertices n is large enough. Moreover, since the length of the key K_H is 128 bits, it is difficult for the attacker to forge a watermark. According to the above analysis, the security can be ensured.

5.2.5 Analysis of Authentication Ability

To analyze the authentication ability of the proposed scheme, several different degrees of modifications are made to the watermarked graphics, and the watermark is extracted from the modified watermarked graphics. When the *NC* value between the extracted watermark and the original watermark does not equal to 1, the graphics is identified as tampering. The experimental results are listed in Table 8. According to the results, the modification to the watermarked graphics can be detected by the proposed scheme even the modification percentage is 1%, which demonstrates the good authentication ability of the proposed scheme.

To further evaluate the authentication ability when few vertices of the watermarked graphics are modified, some attacks are made to 300 watermarked graphics, and the

TABLE 8
The NC Between the Extracted Watermark and the Original Watermark

Graphics	Vertices modification percentage				
	1%	2%	4%	8%	10%
G1	0.9952	0.9906	0.9798	0.9639	0.9539
G2	0.9940	0.9887	0.9800	0.9598	0.9468
G3	0.9953	0.9892	0.9813	0.9602	0.9485
G4	0.9955	0.9903	0.9796	0.9591	0.9506
G5	0.9952	0.9901	0.9796	0.9596	0.9490
Average of 300 graphics	0.9947	0.9896	0.9801	0.9593	0.9499

TABLE 9
The Probability of Passing Authentication When Attacks are Conducted to Few Vertices of Graphics

Operation	Passing authentication rate	
	RN[30]	Proposed
A	0%	0%
B	0%	0%
C	1.32%	0.99%
A+B	0%	0%
A+C	0%	0%
B+C	0%	0%
A+B+C	0%	0%

statistical authentication results are calculated. At the same time, the authentication performance of the proposed scheme is compared with the method in [30], and the results are listed in Table 9, where A represents randomly addition of a vertex, B represents randomly deletion of a vertex, and C represents randomly modification of the coordinates of a vertex. According to the results, both methods can achieve good performance. Furthermore, for the situation C, the passing authentication rate of the proposed sheme is smaller than that of the method in [30], which indicates its better authentication ability in vertex modification.

6 Conclusion

Based on difference expansion with adaptive interval partitioning, a novel semi-fragile reversible watermarking is proposed for 2D engineering graphics. As far as we know, it is the first research on reducing embedding distortion by using the adaptive interval partition. Experimental results and theoretical analysis also indicate that it can achieve better imperceptibility compared with the existing schemes, and it has good semi-fragility in terms of rotation, translation, scaling and vertices rearrangement. However, the tampering of the farthest vertices pair will compromise the authentication. Our future works will be concentrated on choosing new geometrically invariant features in graphics and studying the related theories and methods of reversible watermarking with low distortion.

Acknowledgments

The authors would like to thank the anonymous reviewers for their kind comments and suggestions for improving the quality of this article.

References

[1] A. K. Singh and C. Kumar, "Encryption-then-compression-based copyright protection scheme for E-governance," *IT Professional*, vol. 22, no. 2, pp. 45–52, 2020.

[2] X. Niu, C. Shao, and X. Wang, "A survey of digital vector map watermarking," *Int. J. Innov. Comput., Informat. Control*, vol. 2, no. 6, pp. 1301–1316, 2006.

[3] J. Wu, F. Yang, and C. Wu, "Review of digital watermarking for 2D-vector map," in *Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Social Comput.*, 2013, pp. 2098–2101.

[4] A. K. Singh, "Data hiding: Current trends, innovation and potential challenges," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 16, no. 3s, pp. 1–16, 2020.

[5] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding-new paradigm in digital watermarking," *EURASIP J. Adv. Signal Process.*, vol. 2002, no. 2, 2002, Art. no. 986842.

[6] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Reversible data hiding," in *Proc. Int. Conf. Image Process.*, 2002, p. II.

[7] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.

[8] A. Van Leest, M. van der Veen, and F. Bruekers, "Reversible image watermarking," in *Proc. Int. Conf. Image Process.*, 2003, pp. II–731.

[9] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Oct. 2006.

[10] D. Coltuc and J.-M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Process. Lett.*, vol. 14, no. 4, pp. 255–258, Apr. 2007.

[11] X. Wang, X. Li, B. Yang, and Z. Guo, "Efficient generalized integer transform for reversible watermarking," *IEEE Signal Process. Lett.*, vol. 17, no. 6, pp. 567–570, Jun. 2010.

[12] Y. Qiu, Z. Qian, and L. Yu, "Adaptive reversible data hiding by extending the generalized integer transformation," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 130–134, Jan. 2016.

[13] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[14] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1147–1156, Aug. 2004.

[15] Y. Hu, H.-K. Lee, K. Chen, and J. Li, "Difference expansion based reversible data hiding using two embedding directions," *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 1500–1512, Dec. 2008.

[16] O. M. Al-Qershi and B. E. Khoo, "Two-dimensional difference expansion (2D-DE) scheme with a characteristics-based threshold," *Signal Process.*, vol. 93, no. 1, pp. 154–162, 2013.

[17] S. Gujjunoori and M. Oruganti, "Difference expansion based reversible data embedding and edge detection," *Multimedia Tools Appl.*, vol. 78, pp. 25 889–25 917, 2019.

[18] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3524–3533, Dec. 2011.

[19] B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5010–5021, Dec. 2013.

[20] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 1, pp. 187–193, Mar. 2010.

[21] I.-C. Dragoi and D. Coltuc, "Local-prediction-based difference expansion reversible watermarking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1779–1790, Apr. 2014.

[22] C. Shao, X. Wang, X. Xu, and X. Niu, "Study on lossless data hiding algorithm for digital vector maps," *J. Image Graph.*, vol. 12, no. 2, pp. 206–211, 2007.

[23] X. Wang, C. Shao, X. Xu, and X. Niu, "Reversible data-hiding scheme for 2-D vector maps based on difference expansion," *IEEE Trans. Inf. Forensics Secur.*, vol. 2, no. 3, pp. 311–320, Sep. 2007.

[24] M. Geng, P. Yu, H. Han, Z. Teng, J. Hu, and Y. Gao, "Reversible watermarking based on invariant sum value for 2D vector maps," in *Proc. 3rd IEEE Int. Conf. Netw. InfraStruct. Digit. Content*, 2012, pp. 521–525.

[25] F. Peng, Z.-J. Yan, and M. Long, "A reversible watermarking for 2D vector map based on triple differences expansion and reversible contrast mapping," in *Proc. Int. Conf. Secur. Privacy Anonymity Comput. Commun. Storage*, 2017, pp. 147–158.

[26] F. Peng, Y.-Z. Lei, M. Long, and X.-M. Sun, "A reversible watermarking scheme for two-dimensional cad engineering graphics based on improved difference expansion," *Comput.-Aided Des.*, vol. 43, no. 8, pp. 1018–1024, 2011.

[27] F. Peng and Y.-Z. Lei, "An effective reversible watermarking for 2D CAD engineering graphics based on improved QIM," *Int. J. Digit. Crime Forensics*, vol. 3, no. 1, pp. 53–69, 2011.

[28] D. Xiao, S. Hu, and H. Zheng, "A high capacity combined reversible watermarking scheme for 2-D CAD engineering graphics," *Multimedia Tools Appl.*, vol. 74, no. 6, pp. 2109–2126, 2015.

[29] N. Wang, H. Zhang, and C. Men, "A high capacity reversible data hiding method for 2D vector maps based on virtual coordinates," *Comput.-Aided Des.*, vol. 47, pp. 108–117, 2014.

[30] Z.-X. Lin, F. Peng, and M. Long, "A low-distortion reversible watermarking for 2D engineering graphics based on region nesting," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 9, pp. 2372–2382, Sep. 2018.

[31] Y. Qiu, H. Gu, and J. Sun, "Reversible watermarking algorithm of vector maps based on ECC," *Multimedia Tools Appl.*, vol. 77, no. 18, pp. 23 651–23 672, 2018.

[32] Y. Qiu, H. Duan, J. Sun, and H. Gu, "Rich-information reversible watermarking scheme of vector maps," *Multimedia Tools Appl.*, vol. 78, no. 17, pp. 24 955–24 977, 2019.

[33] M. Voigt, B. Yang, and C. Busch, "Reversible watermarking of 2D-vector data," in *Proc. Workshop Multimedia Secur.*, 2004, pp. 160–165.

[34] L. Deng and H. Xiao, "A lossless watermarking algorithm for vector graphics based on wavelet transform," *Comput. Knowl. Technol.*, vol. 6, no. 4, pp. 889–898, 2010.

[35] S. Tao, X. Dehe, L. Chengming, and S. Jianguo, "Watermarking GIS data for digital map copyright protection," in *Proc. 24th Int. Cartographic Conf.*, 2009, pp. 1–9.

Fei Peng (Member, IEEE) received the PhD degree in circuits and systems from the South China University of Science and Technology, Guangzhou, China, in 2006. He was a visiting fellow with the Department of Computer Science, The University of Warwick, U.K., from 2009 to 2010. He was a visiting professor with the SeSaMe Centre, School of Computing, National University of Singapore, in 2016. He is currently a professor with the Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou. His areas of interest include digital watermarking and digital forensics.

Wenyan Jiang Wen-Yan Jiang received the BS degree in the Internet of Things from Jiangxi Normal University, Nanchang, Jiangxi, China, in 2018. She is currently working toward the master's degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha. Her areas of interests include multimedia security and digital watermark.

Min Long received the PhD degree in circuits and systems from the South China University of Science and Technology, Guangzhou, China, in 2006. She was a visiting fellow with the Department of Computer Science, The University of Warwick, U.K., from 2009 to 2010. She is currently a professor with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China. Her areas of interest include digital watermarking and digital forensics.

Keqin Li (Fellow, IEEE) is currently a SUNY distinguished professor of computer science with the State University of New York. He is also a national distinguished professor with Hunan University, China. His research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyber-physical systems, heterogeneous computing systems, Big Data computing, high-performance computing, CPU-GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, intelligent and soft computing. He has authored or coauthored more than 820 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He holds more than 60 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the worlds top 10 most influential scientists in parallel and distributed computing based on a composite indicator of Scopus citation database. He has chaired many international con- ferences. He is currently an associate editor for *ACM Computing Surveys* and the *CCF Transactions on High Performance Computing*. He has served on the editorial boards for *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Services Computing*, and *IEEE Transactions on Sustainable Computing*.