A Low-Distortion Reversible Watermarking for 2D Engineering Graphics Based on Region Nesting

Zi-Xing Lin, Fei Peng, *Member, IEEE*, and Min Long

***Abstract*— With the aim of reducing the distortion of traditional partitions, this paper investigates a novel method that partitions every square region into 2^n nesting sub-regions. Based on the partition, a low-distortion reversible watermarking for 2D engineering graphics is proposed. The watermark is embedded by mapping the vertices in the original region to its corresponding sub-regions, and it is extracted according to the locations of the mapped vertices. Meanwhile, the original locations of the vertices can be restored by inverse mapping. Furthermore, by constructing a new coordinate system, the rotation, scaling, and translation semi-fragility of the watermarking can be achieved. Experimental results and analysis show that the proposed watermarking achieves good performance in terms of the capacity and semi-fragility, while the imperceptibility of the watermarking is significantly improved in comparison with the existing algorithms under the same conditions.**

***Index Terms*— Reversible watermarking, low distortion watermarking, region nesting, 2D engineering graphics.**

I. Introduction

As a protection technology for 2D engineering graphics, reversible watermarking has been widely studied in the past decade. Since the watermarked 2D engineering graphics can be losslessly recovered after watermark extraction, it is suitable for the occasions with high data precision requirement. However, for the watermarked 2D engineering graphics, distortion is unavoidable. Large distortion may provide clue for adversary in detecting watermark, which will increase the risk of watermark damage. Thus, it is important to improve the imperceptibility for the reversible watermarking of 2D engineering graphics.

Currently, the existing reversible watermarking for 2D engineering graphics can be classified into two categories, and they are integer based methods (e. g., lossless compression [1], difference expansion (*DE*) [2] etc.) and floating number based methods (e. g., improved difference expansion (*IDE*) [3], improved quantization index modulation (*IQIM*) [4], virtual coordinates (*VC*) [5] etc.). For the integer based methods, the first several digits of the floating number of the graphics are treated as an integer, and the watermark is embedded by the integer transformation. As the data of 2D engineering graphics are mainly represented in float, it cannot make full use of the features of floating numbers, which results in large distortion. In comparison, for the floating number based methods, the watermark embedding is mainly involved in the fraction part of the floating numbers. Although it has a certain improvement in imperceptibility and capacity compared to the integer based methods, there still has room for improving its imperceptibility because the features of 2D plane are not fully implemented.

In order to improve the imperceptibility of the reversible watermarking for 2D engineering graphics, a low distortion reversible watermarking based on region nesting is proposed in this paper. Since the bar pattern partition has the defect that the distortion only exists in one direction, a new region partition method is designed. The new partition can distribute the distortion in all directions, which is helpful for the improvement of imperceptibility. The watermark is embedded by mapping the vertices in the original region to the corresponding sub-regions. The extraction of the watermark can be done by finding out the sub-regions where the vertices are. The reversibility is accomplished by the reverse mapping. The contributions of this paper include:

- We investigate the relation between the imperceptibility and the embedding space partition means for the floating number based reversible watermarking of 2D engineering graphics. It is found that with the same maximum distortion and capacity, bar partition can achieve better imperceptibility. However, the distortion is only distributed in one dimension.
- We design a region nesting partition method which partitions the sub-space of the watermark in two-dimensional space with nesting, and it distributes the distortions in all directions. To the best of our knowledge, it is the first work to improve the imperceptibility in this way.
- We propose a reversible watermarking based on region nesting for 2D engineering graphics. Experimental results show that the imperceptibility of the proposed watermarking is significantly improved compared to the existing algorithms. Meanwhile the proposed watermarking maintains high capacity and good semi-fragility.

The rest of the paper is organized as follows. The related work is briefly introduced in Section II. The analysis of existing region partition methods and the improvement are respectively made in Section III. In Section IV, the proposed

Manuscript received November 22, 2017; revised February 26, 2018 and March 10, 2018; accepted March 15, 2018. Date of publication March 23, 2018; date of current version May 1, 2018. This work was supported in part by a project supported by the National Natural Science Foundation of China under Grant 61572182 and Grant 61370225, and in part by a project supported by the Hunan Provincial Natural Science Foundation of China under Grant 15JJ2007. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dinu Coltuc. *(Corresponding author: Fei Peng.)*

The authors are with the School of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: b1410z0225@hnu.edu.cn; eepengf@gmail.com; caslongm@aliyun.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2018.2819122

1556-6013 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

reversible watermarking scheme is described in details. Experimental results and analysis are provided in Section V. Finally, conclusions are drawn in Section VI.

II. Related Work

Nowadays, the research of reversible watermarking mainly focuses on the raster image. Lossless compression [6], [7], histogram shifting [8]–[12], integer transform [13], [14], difference expansion [15], [16], code division multiplexing [17], prediction error expansion [18]–[23] and their variants [24]–[27] have been widely used for constructing reversible watermarking. As these algorithms were initially designed for raster images, the watermark is embedded and extracted from the data represented in integer data.

In the early stages, the reversible watermarking for 2D vector graphics were proposed by referencing the algorithms for the raster image. Voigt *et al.* first proposed a reversible watermarking for 2D-vector data [28]. It modifies the relation between the high frequency coefficients and the low frequency coefficients in DCT (Discrete Cosine Transform) domain to embed watermark. However, the modification to the coefficients significantly reduce the imperceptibility. After that, Shao *et al.* proposed a lossless data hiding algorithm for digital vector maps [1]. It vacates room by lossless compressing the LSB plane of the data, and utilizes the vacated space to embed data. Wang *et al.* proposed two reversible watermarking schemes for 2D vector maps based on difference expansion [2]. One scheme embeds watermark by modifying the difference of the coordinates between two adjacent vertices, while the other scheme modifies the second order difference among three adjacent vertices. Since both of the schemes depend on the lossless compression technique, their capacities are not stable and the robustness is limited. Zhong *et al.* proposed a reversible watermarking algorithm for vector maps using the difference expansion method with a composite integer transform [29]. It uses a composite difference expansion integer transform to embed the watermark among several integers. It achieves larger capacity by sacrificing the imperceptibility. Cao *et al.* proposed a nonlinear scrambling-based reversible watermarking for 2D-vector maps [30]. After scrambling the feature points, it embeds the watermark and the location map of the feature points of every polyline into themselves. However, it introduces large distortion to the maps.

The other reversible watermarking schemes for 2D-vector graphics focus on the floating point numbers. Peng *et al.* proposed a reversible watermarking scheme for 2D engineering graphics based on improved difference expansion (*IDE*) [3]. It embeds watermark into the relative coordinates by using *IDE*. As it embeds *n* bits into every non-reference vertex, it can achieve high capacity. However, as the reference vertices are randomly selected, the maximum distortion is hard to control. After that, Peng and Lei proposed a reversible watermarking based on improved quantized index modulation (*IQIM*) [4]. They improved *QIM* scheme by making it reversible. *IQIM* is applied to either the relative amplitudes or the relative phases of vertices to construct reversible watermark scheme for 2D engineering graphics. Based on the above two methods, Xiao *et al.* proposed a combined reversible watermarking for 2D CAD (computer aided design) engineering graphics [31]. It not only embeds the watermark into the relative coordinates by *IQIM*, but also embeds the watermark into the original distance that should be saved by *IDE* at the same time. Comparing with the above methods in [3] and [4], the capacity is improved. Wang and Wang proposed a reversible data hiding method for point-sampled geometry [32]. It groups every three adjacent sorted coordinates, and embeds watermark by modifying the middle coordinate of every group. By replacing the unused coordinates with virtual coordinates, Wang *et al.* proposed a reversible data hiding scheme (*VC*) [5], and its capacity is three times higher than that of [32]. After that, Wang proposed a normalized vertices-based reversible watermarking (*NV*) based on Xiao's combined method [33]. It embeds watermark into coordinates by using Xiao's method, and enhances the RST semi-fragility by constructing a new coordinate system. Cao *et al.* proposed a high-capacity reversible watermarking scheme for 2D vector data [34]. It embeds watermark by modifying the middle value of the coordinates group. The capacity is improved by iteratively embedding, but it significantly depends on the data correlation. Based on the ideas of iteratively embedding and virtual coordinates, Peng *et al.* proposed an improved reversible watermarking scheme for 2D engineering graphics [35]. It creates two virtual coordinates for every real coordinate, treats them as a group, and embeds watermark by modifying the real coordinate with Cao's method. It solves the problem of low data correlation, but the robustness is limited. Recently, Lin *et al.* proposed a reversible watermarking for authenticating 2D vector graphics based on bionic spider web [36]. By adding vertices and substituting entities, it can detect tampering according to the changes of intersection state. It solves the problem of low data correlation, but it still has some limitation in extra side information for tampering localization.

According to the above analysis, the integer based reversible watermarking methods rely on the data correlation. As the data correlation of 2D engineering graphics is generally low, this type of methods usually result in poor imperceptibility. As the modification of the fraction of the floating-point number has small impact on the graphics content, the floating point number based methods have advantages in the imperceptibility. Furthermore, the existing embedding space partitions still have limitation in the distribution of distortion. The geometric characteristics of 2D plane are not fully implemented. In this paper, we design a novel embedding space partition method and propose a reversible watermarking for 2D engineering graphics with low distortion.

III. Analysis of the Imperceptibility of Different Region Partitions

In essence, the existing floating point number based reversible watermarking is based on the partition of the 2D plane. In the representative reversible algorithms, *IDE* and *IQIM* partition 2D plane in bar pattern, while virtual coordinates based reversible algorithms partitions plane in

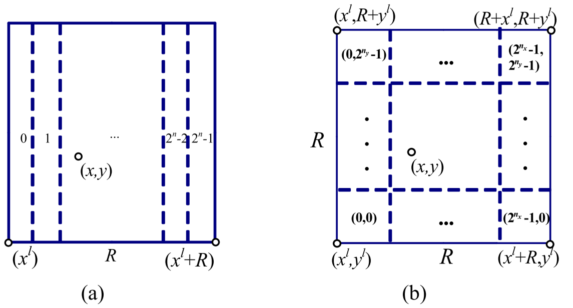


Fig. 1. The existing region partition methods. (a) Bar pattern partition. (b) Grid pattern partition.

grid pattern. In this section, two types of partition methods are investigated.

A. Bar Pattern and Grid Pattern Partitions

For *IDE* based reversible watermarking algorithm [3], it firstly partitions the x-axis into several intervals with a length of R, and then partitions each of them into 2^n sub-intervals, as shown in Fig. 1(a), where n is the embedding strength. After that, it embeds watermark by mapping the vertices to the corresponding sub-intervals of the watermark using uniform scaling. Let x, y and x^l represent the horizontal, vertical coordinate of the vertex, and the horizontal coordinate of the left endpoint of the interval where the vertex is, respectively. For a vertex (x, y), it can be mapped to (x', y') by using the mapping function of the bar pattern partition, which can be formulated as

$$(x', y') = f_B((x, y), R, n, w) = (w\frac{R}{2^n} + \frac{x - x^l}{2^n} + x^l, y),\quad w \in \{0, \ldots, 2^n - 1\}, \tag{1}$$

where w represents the watermark, and x^l is calculated by

$$x^l = \lfloor x/R \rfloor \times R. \tag{2}$$

The maximum movement of the coordinates is:

$$\begin{aligned} Maxd_B(R, n) &= \left\| (x^l, y) - f_B((x^l, y), R, n, 2^n - 1) \right\| \\ &= R(1 - \frac{1}{2^n}). \end{aligned} \tag{3}$$

For virtual coordinates based reversible watermarking algorithm [5], it applies the bar pattern partition to both x and y-axis, and the square region of side length R is partitioned into $2^{n_x} \times 2^{n_y}$ sub-regions, where $n_x = \lfloor n/2 \rfloor$ and $n_y = \lceil n/2 \rceil$, which is shown in Fig. 1(b). Let (x^l, y^l) represent the coordinates of the bottom-left endpoint of the sub-region where the vertex is located. For a vertex (x, y), it can be mapped to (x', y') by using the mapping function of the grid pattern partition, which can be formulated as

$$\begin{aligned} (x', y') &= f_G((x, y), R, n, w) \\ &= (w_x\frac{R}{2^{n_x}} + \frac{x - x^l}{2^{n_x}} + x^l, w_y\frac{R}{2^{n_y}} + \frac{y - y^l}{2^{n_y}} + y^l), \end{aligned} \tag{4}$$

where the watermark w is split into two parts w_x and w_y. They satisfy the relation $w = w_x \times 2^{n_y} + w_y$, $w_x \in \{0, \ldots, 2^{n_x} - 1\}$, $w_y \in \{0, \ldots, 2^{n_y} - 1\}$, and y^l can be calculated by replacing x with y in (2).

The maximum movement of the coordinates is

$$\begin{aligned} Maxd_G(R, n) &= \left\| (x^l, y^l) - f_G((x^l, y^l), R, n, 2^n - 1) \right\| \\ &= R\sqrt{(1 - 1/2^{n_x})^2 + (1 - 1/2^{n_y})^2}. \end{aligned} \tag{5}$$

In order to compare the distortion of the vertices of these two methods, the average movement of all vertices of a single region is calculated, and it is used to evaluate the imperceptibility when the maximum movement *Maxd* and embedding strength n are the same. The average movement of a specific vertex is

$$d((x, y), R, n) = \frac{1}{2^n}\sum_{w=0}^{2^n-1} \|(x, y) - f((x, y), R, n, w)\|. \tag{6}$$

As $d((x,\ y),\ R,\ n)$ and $Maxd(R,\ n)$ of different partitions are both the linear functions of R, $d'((x, y), 1, n)$ can be normalized as

$$d'((x, y), 1, n) = \frac{d((x, y), R, n)}{Maxd(R, n)} = \frac{d((x, y), 1, n)}{Maxd(1, n)}, \tag{7}$$

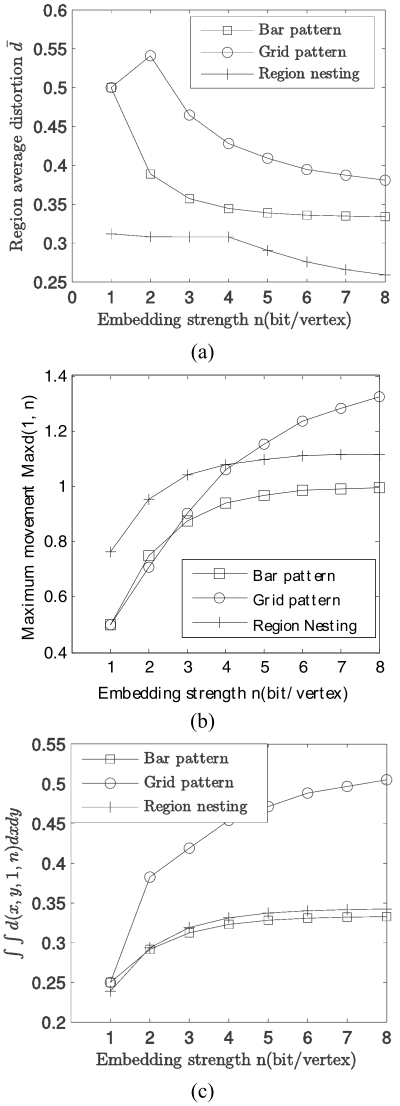
where $Maxd(R,\ n)$ represents $Maxd_G(R,\ n)$ or $Maxd_B(R,\ n)$ according to different partitions. Under this situation, the average movement of the vertices in a single region with side length of 1 (unit region) can be calculated as

$$\begin{aligned} \overline{d}(1, n) &= \iint\limits_{D=1\times 1} d'((x, y), 1, n)dxdy \\ &= \frac{1}{Maxd(1, n)} \iint\limits_{D=1\times 1} d((x, y), 1, n)dxdy. \end{aligned} \tag{8}$$

Based on (8), the imperceptibility of a special partition is determined by the maximum movement and the average movement of all vertices in a unit region. Since there is no analytic solution for $\overline{d}_B(1, n)$ or $\overline{d}_G(1, n)$, the simulation with a step of 0.01 was performed with MATLAB R2014a, and the results are shown in Fig. 2. As shown in Fig 2(a) and (b), even though $Maxd_G(1, n) > Maxd_B(1, n)$ $(n \geq 3)$, the average regional movement of the bar pattern partition is still smaller than that of the grid pattern partition with the same capacity, except the case that two partitions are identical when $n = 1$. As seen in Fig. 2(c), the average movement of all vertices in a unit region are much smaller than that of the grid pattern partition, because the original and the mapped coordinates of every vertex are in the same line for the bar pattern partition.

After the comparison, two principles for improving imperceptibility are identified. First, the maximum movement in a unit region should be increased. Second, the distance between the original and the mapped coordinates should be decreased by applying both the partition and the mapping methods.

However, even though the bar pattern partition can achieve better performance in imperceptibility, it does not fully utilize the features of 2D plane. The distortion is only distributed in a single direction. Thus there is room to improve the imperceptibility with other partition means.



(a)

(b)

(c)

Fig. 2. The simulation results. (a) The average regional movement of different methods; (b) the maximum distortion *Maxd*(1, *n*) of different methods; (c) the sum of the average movement of all vertices in a unit region of different methods.

B. Region Nesting Partition

1) Partition Principle: With the analysis of the grid pattern partition, it can be found that the mapped coordinates of a specific vertex spread all over the region, no matter what the mapping method is. Besides, in the bar pattern partition, the original and the mapped coordinates of a specific vertex are in the same horizontal line. The distortion is distributed in a single direction, meanwhile the *Maxd*(1, *n*) is also limited. Based on these characteristics, we propose a novel partition named as nesting partition, which can guarantee that the original and the mapped coordinates of different vertices are gathered in the lines with different slopes, as illustrated in Fig. 3. It reshapes the watermarking sub-regions in a nesting way. In order to keep equal areas of all sub-regions, the partition parameters are firstly analyzed. For facilitating analysis, we mark the extensions of the partition lines in dotted lines, and they are represented by $L_0, L_1, \ldots, L_{2^n}$ in order from the left to the right and from the bottom to the top, respectively. In the same way, mark the horizontal or vertical gap between two adjacent extension lines by $0, 1, \ldots, 2^n - 1$ in the same order, and mark the watermarking sub-regions by $0, 1, l, 2^n - 1$ from inside to outside.

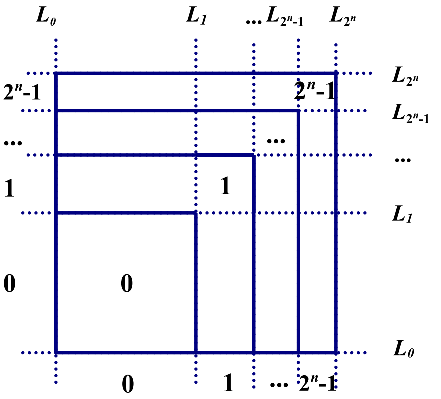


Fig. 3. Region nesting partition.

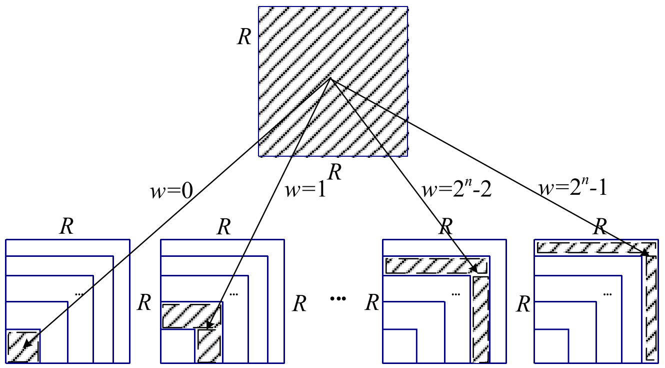


Fig. 4. The schematic for watermark embedding.

For the innermost 0th sub-region, it degenerates into a square. Its side length is:

$$R_0 = \sqrt{R^2/2^n} = R/2^{n/2}. \tag{9}$$

For the w^{th} sub-region, it forms a square with the first w sub-regions, and the side length of this square is

$$R_w = \sqrt{(w+1)R^2/2^n} = R\sqrt{w+1}/2^{n/2}, \tag{10}$$

where $w = 0, 1, \ldots, 2^n - 1$.

2) Mapping Method: The mapping method is presented in Fig.4 where the vertex is mapped from the original region to the sub-region according to the watermark. Due to the shape difference between the original region and the sub-regions, the block division and uniform scaling are applied. For the watermark w, it firstly divides the original region into two blocks $Z_{0,w}$ and $Z_{1,w}$ with a parameter T_w that relates to the watermark w before the mapping. It is shown in Fig. 5(a) and (b). Meanwhile, the target sub-region corresponding to the watermark is partitioned into blocks $Z'_{0,w}$ and $Z'_{1,w}$. After constructing the mapping relation between $Z_{0,w}$ and $Z'_{0,w}$ and the relation between $Z_{1,w}$ and $Z'_{1,w}$, the vertex is mapped according to its location. As for the vertex on the boundary of blocks/region/sub-region, a convention is made. If the vertex is on the left or the

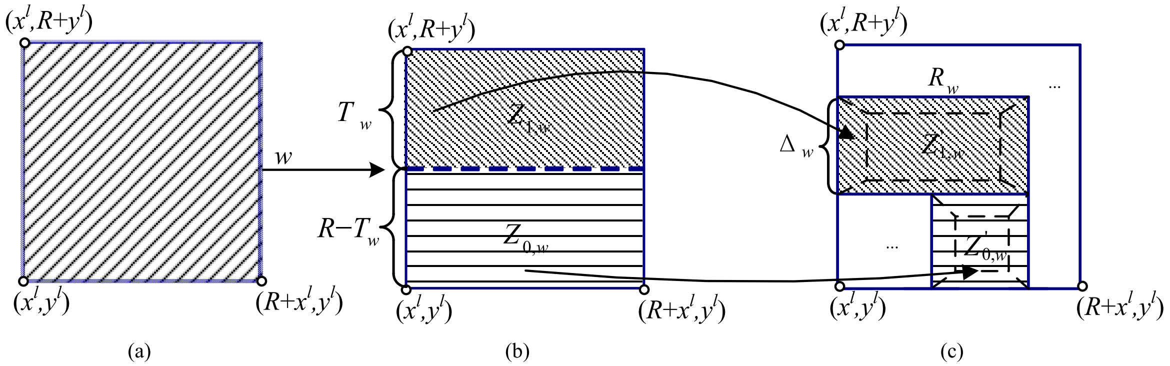


Fig. 5. Illustration of the mapping method. (a) Division of the original region. (b) Division of the sub-region. (c) Construction of the sub-region.

bottom boundary of a block/region/sub-region, it belongs to this block/region/sub-region; otherwise, it does not.

Assuming that the ratio of the area of $Z_{1,w}$ to that of the original region is equal to that of $Z'_{1,w}$ to that of the sub-region, the relation between T_w and w can be calculated as

$$\frac{T_w R}{R^2} = \frac{\Delta_w R_w}{R^2/2^n} \Rightarrow T_w = R\sqrt{w+1}(\sqrt{w+1}-\sqrt{w}), \quad (11)$$

where Δ_w represents the difference between the length of the w^{th} square, and that of the $(w-1)^{\text{th}}$ square is

$$\begin{aligned}\Delta_w &= R_w - R_{w-1} \\ &= R(\sqrt{w+1}-\sqrt{w})/2^{n/2}, w = 1, \ldots, 2^n - 1. \quad (12)\end{aligned}$$

Note that errors may occur if the peripheral vertices stride over the sub-region boundary after mapping, which will cause watermark extraction failure. To reduce this probability, the actual dividing blocks are constricted to its center with a constrictive factor $r(0 < r \leq 1)$, and it is shown in Fig. 5(c). After constriction, the ratio of the long side of $Z'_{1,w}$ to R_w equals r, and the distances between the left boundary of $Z'_{1,w}$ and the left boundary of original region is $(1-r)R_w/2$. Thus, the mapped coordinates of the vertex can be calculated by (13), as shown at the bottom of this page.

We use the following example for illustration. Given $n = 4$, $R = 10^{-6}$, $w = (0010)_2$ and a vertex $(x, y) = (10^{-7}, 2 \times 10^{-7})$, it will be mapped to the 2nd watermarking sub-region according to the watermark information, as seen in Fig. 4. Before mapping, it is needed to calculate the sub-block where the mapped vertex is in. According to (3) and (11), the coordinates of the bottom-left vertex of the region is $(x^l, y^l) = (0, 0)$, and $T_2 = \sqrt{3}(\sqrt{3}-\sqrt{2})R \approx 5.505\times 10^{-7}$. According to Fig.5 (a) and (b), as the distance between (x, y) and the bottom boundary of the region is 2×10^{-7} which is less than $R - T_2 \approx 4.495 \times 10^{-7}$, (x, y) belongs to $Z_{0,2}$, thus the vertex will be mapped to the sub-block $Z'_{0,2}$. According to (10) and (12), it can be also obtained that $R_1 = R\sqrt{2}/4$ and $\Delta_2 = R(\sqrt{3}-\sqrt{2})/4$. By using uniform scaling, the distance between the mapped vertex and the left boundary of $Z'_{0,2}$ is $x_i \times \Delta_2/R$, and the distance between the mapped vertex and the bottom boundary of $Z'_{0,2}$ is $y_i \times R_1/(R - T_2)$. After constricting the mapped vertex toward the center of $Z'_{0,2}$ with $r = 0.999$ (as seen in the region with dotted lines in Fig. 5(c)), two distances are changed to $x_i \times \Delta_2/R\times r+(1-r)/2\times\Delta_2$ and $y_i\times R_1/(R-T_2)\times r+(1-r)/2\times R_1$, respectively. Finally, we can get the mapped vertex $(x^l + R_1 + x_i \times \Delta_2/R \times r + (1-r)/2 \times \Delta_2, y^l+y_i\times R_1/(R-T_2)\times r+(1-r)/2\times R_1) \approx (3.615\times 10^{-7}, 1.573 \times 10^{-7})$.

3) Analysis of Distortion: In fact, two potential maximum movements may occur when the watermark $w = 0$ is embedded into the vertex in the region's top-right position, and when the watermark $w = 2^n - 1$ is embedded into the vertex $(x^l, y^l + R - T_{2^n-1})$. They are shown in Fig. 6. As r generally approaches 1 in real application, it is assumed to be 1 for simplicity.

For the first case, the movement is

$$Md_{RN}(R, n)_0 = R\sqrt{2}(1 - \frac{1}{\sqrt{2^n}}). \quad (14)$$

While for the second one, the movement is

$$\begin{aligned} Md(R,n)_{2^n-1} &= \sqrt{(R-\Delta_{2^n-1})^2+((R-T_{2^n-1})-(R-\Delta_{2^n-1}))^2} \\ &= R\sqrt{\frac{1}{c^2}(1+\frac{1}{(1+c)^2})}, \quad (15)\end{aligned}$$

where $c = \sqrt{1 + (2^n - 1)^{-1}}$.

$$\begin{aligned}(x'_i, y'_i) &= f_{RN}(x, y, R, n, w) \\ &= \begin{cases} \left(R_w(\frac{x_i - x_i^l}{R} \times r + \frac{1-r}{2}) + x_i^l, \Delta_w(\frac{y_i - y_i^l - R + T_w}{T_w} \times r + \frac{1-r}{2}) + y_i^l + R_{w-1}\right), & \textit{if } y_i - y_i^l \geq R - T_w. \\ \left(\Delta_w(\frac{x_i - x_i^l}{R} \times r + \frac{1-r}{2}) + x_i^l + R_{w-1}, R_{w-1}(\frac{y_i - y_i^l}{R - T_w} \times r + \frac{1-r}{2}) + y_i^l\right), & \textit{otherwise}. \end{cases} \quad (13)\end{aligned}$$

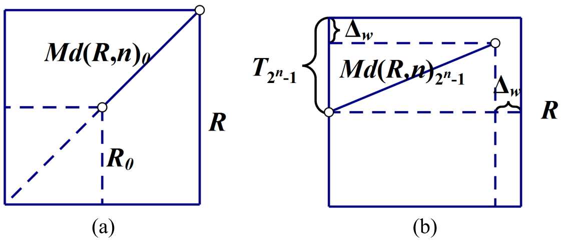


Fig. 6. Two potential maximum movement: (a) $w = 0$, (b) $w = 2^n 1$

For a specific embedding strength n, the maximum movement is the larger one between them, and that is

$$Maxd_{RN}(R, n) = \max(Md(R, n)_0, Md(R, n)_{2^n-1}). \quad (16)$$

It can be found that $Maxd_{RN}(R, n)$ is also a linear function of R. According to (14), (15) and (16), it can be found

$$Maxd_{RN}\,(1, n) = \begin{cases} Md_{RN}(1, n)_{2^n-1} & \textit{if } n \le 4 \\ Md_{RN}(1, n)_0 & \textit{otherwise.} \end{cases} \quad (17)$$

As shown in Fig. 2(b), $Maxd_{RN}(1, n) > Maxd_B(1, n)$ always holds. Meanwhile, by applying (13) to the vertices, it can be found that most of the original and the mapped coordinates of a special vertex are still in a same line. It means that the performance of the proposed method in the average movement of all vertices in a unit region is similar to that of the bar pattern partition. The results are shown in Fig. 2(c). From the average movement $\overline{d}_{RN}(1, n)$ shown in Fig. 2(a), it can be found that the average movement of the proposed method is much less than that of the traditional methods, which indicates better imperceptibility can be achieved.

IV. The Proposed Reversible Watermarking

Given a secret watermark information $S = \{s_i | s_i \in \{0, 1\}, i \in \{0, 1, \ldots, L - 1\}\}$, a 2D engineering graphics G with N vertices and a precision tolerance τ, a new coordinate system is constructed to guarantee that the reversible watermarking can resist against RST operations. During the new coordinate construction, two reference vertices are randomly selected. For every vertex except two reference vertices, n bits of watermark information are embedded into it. Here, $L = (N - 2) \times n$, and n represents embedding strength.

In order to guarantee that the maximum movement is less than the precision tolerance τ, it is needed to select an eligible R. For the precision tolerance τ, R needs to satisfy

$$R \le \tau / Maxd_{RN}\,(1, n)\,. \quad (18)$$

In addition, to further improve the robustness of the watermark, the distance between two reference vertices should be divided by R with no remainder [33]. Assume the distance is divided into D parts, the relation between D and R should satisfy

$$R = \frac{\|v_{r1} - v_{r2}\|}{D}, D \in N^+. \quad (19)$$

According to (18) and (19), the partition coefficient D can be calculated by

$$D = ceil(\|v_{r1} - v_{r2}\| \times Maxd_{RN}\,(1, n) / \tau). \quad (20)$$

A. Watermark Embedding

Step 1: Segment S into groups with n bits, and then the watermark sequence $W = \{w_i | w_i \in \{0, 1, \ldots, 2^n - 1\}, i \in \{0, 1, \ldots, N - 3\}\}$ is obtained.

Step 2: Obtain the vertex set $V = \{v_i | v_i = (x_i,\ y_i), i \in \{0, 1, \ldots, N - 1\}\}$ of the 2D engineering graphics G, and choose two references vertices $v_{r1}(x_{r1},\ y_{r1})$ and $v_{r2}(x_{r2},\ y_{r2})(v_{r1} \neq v_{r2})$ with a key K.

Step 3: Calculate the partition coefficients D and R by (19) and (20), respectively.

Step 4: Transform the vertex set from the original coordinate system to the new coordinate system constructed by two reference vertices, and then the new vertex set $\widehat{V} = \left\{\widehat{v}_i | \widehat{v}_i = (\widehat{x}_i, \widehat{y}_i), i \in \{0, 1, \ldots, N - 1\}\right\}$ is obtained. The transform function is

$$\begin{pmatrix} \widehat{x}_i \\ \widehat{y}_i \end{pmatrix}^T = \begin{pmatrix} x_i - \dfrac{x_{r1} + x_{r2}}{2} \\ y_i - \dfrac{y_{r1} + y_{r2}}{2} \end{pmatrix}^T \begin{pmatrix} \dfrac{x_{r2} - x_{r1}}{\|v_{r1} - v_{r2}\|} & -\dfrac{y_{r2} - y_{r1}}{\|v_{r1} - v_{r2}\|} \\ \dfrac{y_{r2} - y_{r1}}{\|v_{r1} - v_{r2}\|} & \dfrac{x_{r2} - x_{r1}}{\|v_{r1} - v_{r2}\|} \end{pmatrix}. \quad (21)$$

Step 5: Embed the watermark W into the non-reference vertices in $\widehat{V}$according to (1), (13) and R, and the watermarked vertex set $\widehat{V}' = \left\{\widehat{v}'_i | \widehat{v}'_i = (x'_i, y'_i), i \in \{0, 1, \ldots, N - 1\}\right\}$ is obtained.

Step 6: Inverse transform is operated on the vertex set$\widehat{V}'$, and the watermarked vertex set in the original coordinate system V' is obtain by

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix}^T = \begin{pmatrix} \widehat{x}'_i \\ \widehat{y}'_i \end{pmatrix}^T \begin{pmatrix} \dfrac{x_{r2} - x_{r1}}{\|v_{r1} - v_{r2}\|} & \dfrac{y_{r2} - y_{r1}}{\|v_{r1} - v_{r2}\|} \\ -\dfrac{y_{r2} - y_{r1}}{\|v_{r1} - v_{r2}\|} & \dfrac{x_{r2} - x_{r1}}{\|v_{r1} - v_{r2}\|} \end{pmatrix} + \begin{pmatrix} \dfrac{x_{r1} + x_{r2}}{2} \\ \dfrac{y_{r1} + y_{r2}}{2} \end{pmatrix}^T. \quad (22)$$

Step 7: Restore the vertices of V' and get the watermarked graphics G'. D and G' can be published, and the key K and the secret watermark S can be sent to the receiver with a secure channel.

B. Watermark Extraction, Graphics Authentication and Recovery

Step 1: Obtain the vertex set $V' = \{v'_i | v'_i = (x'_i, y'_i), i \in \{0, 1, \ldots, N - 1\}\}$ of the watermarked graphics G', and get the two reference vertices $v_{r1}(x_{r1}, y_{r1})$ and $v_{r2}(x_{r2}, y_{r2})$ with the key K.

Step 2: Calculate R with the published D by (19).

Step 3: Transform the watermarked vertex set to the new coordinate system by (21), and obtain the vertex set $\widehat{V}$.

Step 4. Extract the watermark from the non-reference vertices of $\widehat{V}$. The watermark extraction is equivalent to the process of calculating the sequence number of the sub-region where the vertex (x'_i, y'_i) is. It is calculated by

$$w = \max(layer(x'_i), layer(y'_i)), \quad (23)$$

where $layer(x'_i)$ and $layer(y'_i)$ are the functions to calculate the serial number of the horizontal and vertical gaps of the

TABLE I
THE DETAIL OF THE TEST GRAPHICS

Graphics	Vertices number	Feature number
$G1$	4166	2398
$G2$	5522	3385
$G3$	1818	1150
$G4$	1460	931
$G5$	3605	2238
Average of 50 graphics	3673.88	2076.96

non-reference vertex. For example, $layer(x_i')$ can be calculated by

$$layer(x_i') = \arg_t(R_{t-1} \leq x_i' - x_i^l < R_t), \tag{24}$$

where $\arg_t(\bullet)$ calculates the value t that satisfies the inside condition.

After processing all non-reference vertices, the watermark $W' = \{w_i' | w_i' \in \{0, 1, \ldots, 2^n - 1\}, i \in \{0, 1, \ldots, N-3\}\}$ is obtained.

Step 5: Concatenate the watermark segments, and then the watermark S' is obtained.

Step 6: If $S = S'$, the 2D engineering graphics pass the authentication, and go to step 7; otherwise, it indicates that the graphics has been modified.

Step 7: Recover the non-reference vertices (x_i', y_i') of $\widehat{V}$ by (25), as shown at the bottom of this page.

Step 8: Inverse transform is made to the vertex set $\widehat{V}$ by (22), and the original vertex set in the original coordinate system V'' is obtained.

Step 9: Restore the vertices of V'' and the original graphics G'' can be obtained.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Results

Experiments are performed on a PC with the following configuration: CPU i5-4460S 2.90GHz, RAM 16 GB, Windows 7 64bit, DWGdirect C++ Libraries and Visual C++6.0. Fifty 2D engineering graphics are used as the experimental data, and five of them are shown in Fig. 7 (a) - (e).

The details are listed in Table I. The parameters of the experiments are set as follows: embedding strength $n = 4$, precision tolerance $\tau = 10^{-6}$, $r = 0.999$, and the watermark information is randomly generated.

The watermarked graphics and the recovered graphics are shown in Fig. 7(f) – (j) and (k) – (o), respectively. It can be found that the watermarks extracted from the watermarked graphics are the same as the original ones.

TABLE II
THE DIFFERENCE BETWEEN ORIGINAL GRAPHICS AND THE RECOVERED ONE

Graphics	***AvgD***($\times 10^{-12}$)	***Maxd***($\times 10^{-12}$)
G1	0.8537	1.7053
G2	1.1261	3.4106
G3	0.4971	1.3642
G4	0.8975	3.1832
G5	0.4966	0.9095
Average of 50 graphics	1.1838	2.9164

The imperceptibility of the watermarked graphics is evaluated by the average distortion *AvgD* and the maximum movement *MaxD* [5], and they are defined as

$$AvgD(V, V') = \frac{1}{N}\sum_{i=0}^{N-1} \left\| v_i - v_i' \right\|, \tag{26}$$

$$MaxD(V, V') = \max(\left\| v_i - v_i' \right\|), i \in \{0, 1, \ldots, N-1\}, \tag{27}$$

where V is the vertex set of the original graphics, V' is the vertex set of the watermarked graphics, $\left\| v_i - v_i' \right\|$ is the distance between the i^{th} vertex of the original graphics and that of the watermark graphics, and N is the number of vertices.

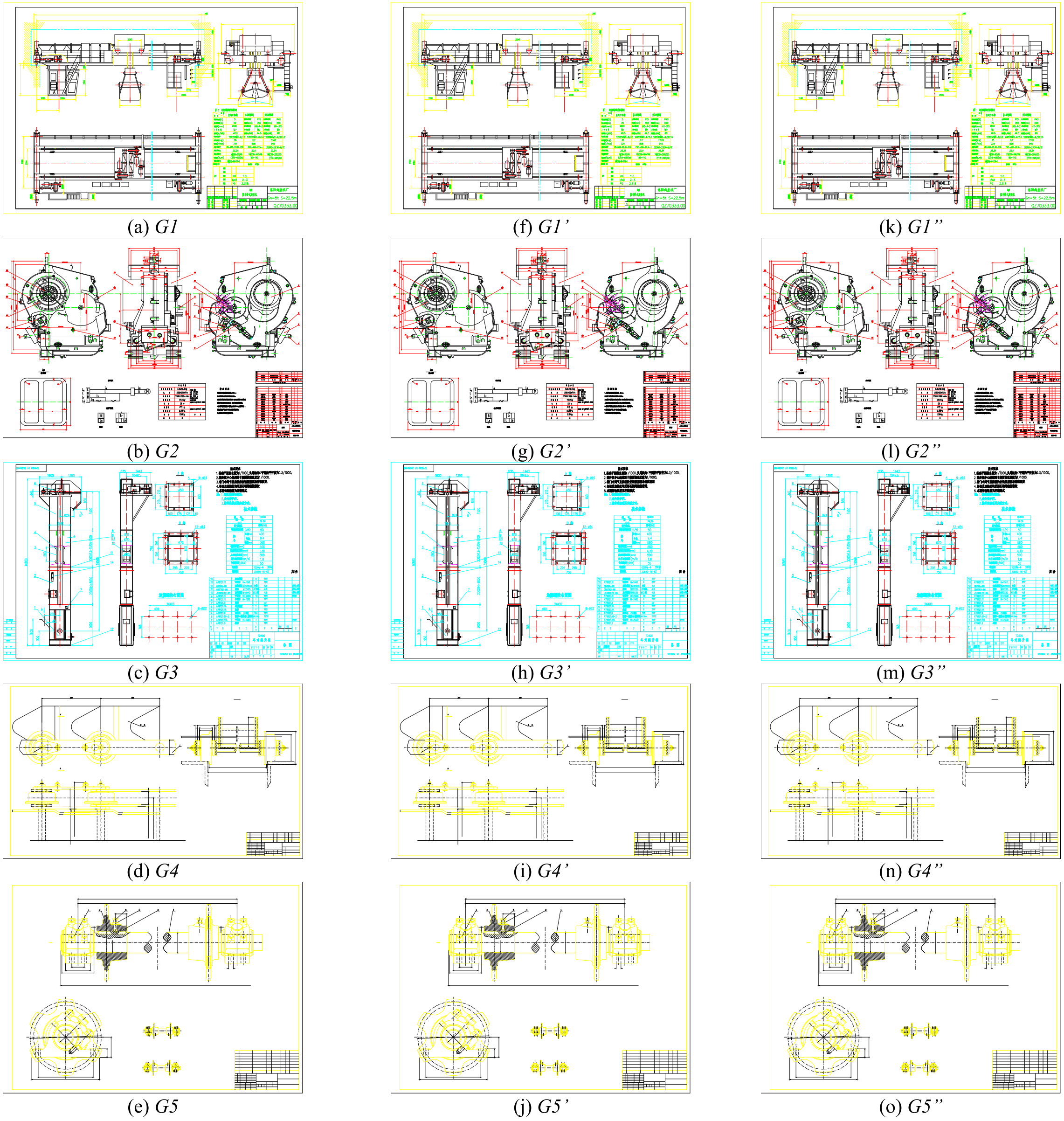
In the experiments, the average *AvgD* between the watermarked graphics and the original ones is 3.1072×10^{-7}; and the average *MaxD* is 9.5426×10^{-7}. The average *AvgD* and *MaxD* between the original graphics and the recovered ones are 1.1838×10^{-13} and 6.2265×10^{-12}, respectively.

B. Performance Analysis

1) Analysis of Reversibility: For a vertex v in the spatial domain, its coordinate $\widehat{v}$ in the new coordinate system can be calculated by (21). By using (13), the watermark is embedded into $\widehat{v}$ and the watermarked vertex $\widehat{v}'$ is obtained. Then, $\widehat{v}'$ is transformed to v' in the original coordinate system by (22). During the extraction and recovery stage, the watermarked vertex v' can be transformed to $\widehat{v}''$ in the new coordinate system by (21). As the transform matrixes in (21) and (22) are inverse to each other, $\widehat{v}'' = \widehat{v}'$ always holds. After that, $\widehat{v}''$ is recovered to $\widehat{v}'''$ by (25). According to (13) and (25), it can be found that $\widehat{v}'''$. Finally, $\widehat{v}'''$ can be restored to v''' by (22). From the aforementioned analysis, $v = v'''$ always holds.

To verify the reversibility of the scheme, the difference between the original graphics and the recovered ones is calculated. The results are listed in Table II. As seen in Table II, *AvgD* and the *MaxD* are both smaller than 10^{-11}. For 2D engineering graphics, if the difference between two graphics

$$\begin{aligned}(x_i'', y_i'') &= f_{RN}^{-1}(x_i', y_i', R, n, w)\\ &= \begin{cases} (\frac{R}{r}(\frac{x_i' - x_i^l}{R_w} - \frac{1-r}{2}) + x_i^l, \frac{T_w}{r}(\frac{y_i' - y_i^l - R_{w-1}}{\Delta_w} - \frac{1-r}{2}) + R - T_w + y_i^l), & \textit{if layer}(y') = w \\ (\frac{R}{r}(\frac{x_i' - x_i^l - R_{w-1}}{\Delta_w} - \frac{1-r}{2}) + x_i^l, \frac{R - T_w}{r}(\frac{y_i' - y_i^l}{R_{w-1}} - \frac{1-r}{2}) + y_i^l), & \textit{otherwise} \end{cases}\end{aligned} \tag{25}$$


(a) *G1* (f) *G1'* (k) *G1"*

(b) *G2* (g) *G2'* (l) *G2"*

(c) *G3* (h) *G3'* (m) *G3"*

(d) *G4* (i) *G4'* (n) *G4"*

(e) *G5* (j) *G5'* (o) *G5"*

Fig. 7. Some samples of 2D engineering graphics. (a) - (e) Original graphics; (f) - (j) Watermarked graphics; (k) – (o) Recovered graphics.

is smaller than 10^{-8}, these two graphics can be regarded identical [3]. Thus, the results illustrate the good reversibility of the proposed scheme.

2) Analysis of Imperceptibility:

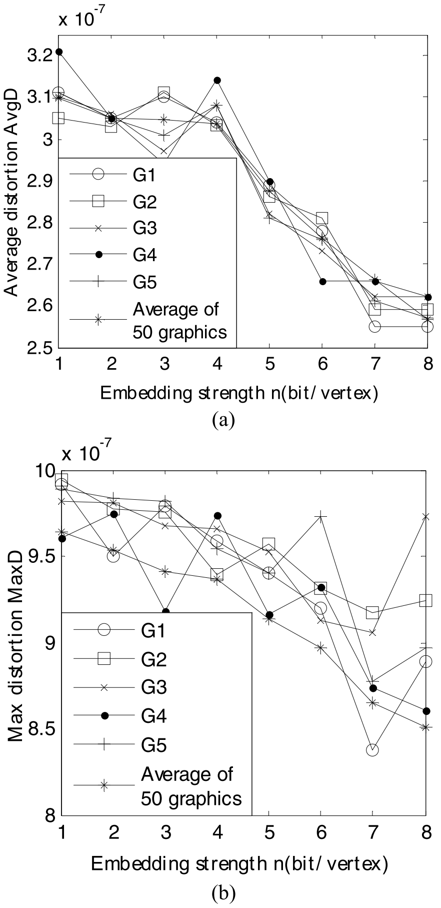
a) Analysis of the relation between the imperceptibility and n: It can be found that the embedding strength n has strong relation with imperceptibility. According to $f_{RN}((x, y), R, n, w)$, the movement of a vertex is limited by the parameter R. With a constant precision tolerance, R will be decreased with the increase of n, which means that the average movement will be decreased in a whole. This situation can be found in Fig. 2. Besides, with the increase of n, the probability that the watermark w equals 2^n-1 or 0 will decrease. Therefore, the movement of the vertices $\left(x^l, y^l + R - T_{2^n-1}\right)$ or $(x^l + R,\ y^l + R)$ are harder to achieve the theoretic maximum value. It indicates that the actual maximum movement will be decreased as n is increased.

In order to verify the aforementioned analysis, experiments are conducted on fifty graphics to calculate the *AvgD* and *MaxD* with different n ($\tau = 10^{-6}$). The results are shown in Fig. 8. As seen in Fig. 8, both *AvgD* and *MaxD* are decreased as n is increased. The experimental results are consistent with the above analysis. According to the results and the analysis, the imperceptibility of the proposed scheme is decreased as n is increased with a given precision tolerance τ.

b) Comparison of imperceptibility of different methods: Here, experiments are done to compare the imperceptibility of different methods. The parameters for *IDE* based scheme [3] are $p = -6$ and $s = 4$, the parameters for *IQIM-A* [4] based scheme are $\Delta = 10^{-6}/(2^4 - 1)$, $b = 4$, the parameters for *VC* based scheme [5] are $\tau = 10^{-6}$, $c_1 = c_2 = 2$, the parameters for *NV* based scheme [33] are $\tau = 10^{-6}$, $b = s = 2$; and the parameters for the proposed scheme are $\tau = 10^{-6}$,

TABLE III
IMPERCEPTIBILITY OF DIFFERENT METHODS ($\times 10^{-7}$)

Graphics	***IDE*[3]**		***IQIM-A*[4]**		***VC*[5]**		***NV*[33]**		**Proposed**	
	AvgD	***Maxd***	***AvgD***	***Maxd***	***AvgD***	***Maxd***	***AvgD***	***Maxd***	***AvgD***	***Maxd***
*G*1	1611.7	4581.6	3.3860	9.9742	4.2570	9.7096	5.4093	9.8965	3.0284	9.3653
*G*2	2295.5	6614.4	3.4778	10.000	4.2833	9.7149	5.3647	9.9881	3.0546	9.5781
*G*3	137.37	407.88	3.3468	9.9556	4.2880	9.7195	5.4733	9.9660	3.1258	9.5958
*G*4	1203.6	3629.8	3.4841	9.9726	4.2587	9.7132	5.4641	9.7714	2.9569	9.5552
G5	282.87	824.44	3.4314	9.9425	4.2416	9.6759	5.4094	9.8381	3.0753	9.6546
Average of 50 graphics	1225.0	3520.9	3.4284	9.8878	4.2275	9.5384	5.3666	9.8116	3.0605	9.3849



(a)

(b)

Fig. 8. The relation between the imperceptibility and $n(\tau = 10^{-6})$. (a) The relation between *AvgD* and n; (b) The relation between *MaxD* and n.

$n = 4$. The results are listed in Table III. As seen in Table III, although *IDE* scheme [3] is based on bar pattern partition, its distortions are very large. The reason is that it embeds watermark by modifying the ratio of the distance between the vertex and a reference vertex to the reference distance. As the reference distance is uncontrollable, the modification of the ratio leads to uncontrollable distortion. Besides, it can be found that the average *AvgD* of *IQIM-A* scheme [4] matches with the simulation result $\overline{d}_B(1,4) = 0.3445$. It not only verifies the previous simulation result, but also indicates that the distortion are strongly related with the embedding position. By comparing it with the proposed scheme, the imperceptibility of proposed scheme is averagely improved by more than (3.4284−3.0605)/3.4284 ≈ 10.73%, which is consistent with the results illustrated in Fig. 2, i.e., $(\overline{d}_B(1,4) - \overline{d}_{RN}(1,4))/\overline{d}_B(1,4) = (0.3445{-}0.3079)/0.3445 \approx 10.62\%$.

TABLE IV
COMPARISON OF THE AVERAGE CAPACITY OF DIFFERENT METHODS

Graphics	***IDE*[3]**	***IQIM-A*[4]**	***VC*[5]**	***NV*[33]**	***Proposed***
*G*1	3.9981	3.9990	3.9846	3.9981	3.9981
*G*2	3.9986	3.9993	3.9971	3.9986	3.9986
*G*3	3.9934	3.9956	3.9670	3.9956	3.9956
*G*4	3.9918	3.9945	3.9849	3.9945	3.9945
*G*5	3.9967	3.9978	3.9922	3.9978	3.9978
Average of 50 graphics	3.9941	3.9952	3.9941	3.9978	3.9978

Meanwhile, by comparing the grid pattern partition based *VC* scheme [5] and the proposed scheme, the imperceptibility of proposed scheme is averagely improved by more than (4.2275−3.0605) /4.2275 ≈ 27.61%, which is consistent with the results in Fig. 2, i.e., $(\overline{d}_G(1,4) - \overline{d}_R(1,4))/\overline{d}_G(1,4) = (0.4282 - 0.3079)/0.4282 \approx 28.09\%$.

From the above, the rationality of the aforementioned analysis in Section 3 is verified, which indicates the good imperceptibility of the proposed scheme.

3) Analysis of Capacity: Similar to the normalized vertex based scheme [33], the proposed scheme embeds n bits of watermark into every non-reference vertex. If the number of vertices of a certain graphics is N, $(N-2) \times n$ *bits* can be embedded into the graphics, so the average capacity is $((N-2) \times n)/N$ *bit*/*vertex*(*bpv*).

In order to verify the aforementioned analysis, the capacity with different embedding strength n is calculated, and the results are shown in Fig. 9.

As shown in Fig.9, the capacity increases linearly with n, which verifies the above analysis. Meanwhile, the comparison of the average capacity of *IDE* [3], *IQIM-A* [4], *VC* [5], *NV* [33] and the proposed scheme are listed in Table IV. The parameters of four schemes are the same as those defined in Section 5.2.2(2). As seen in Table IV, the capacity of these schemes is nearly the same. It indicates that it still can maintain large capacity when the imperceptibility is significantly improved.

4) Analysis of the Robustness: We firstly prove that the proposed scheme is robust against RST operations in theory. Assume the coordinates of the watermarked vertex in the

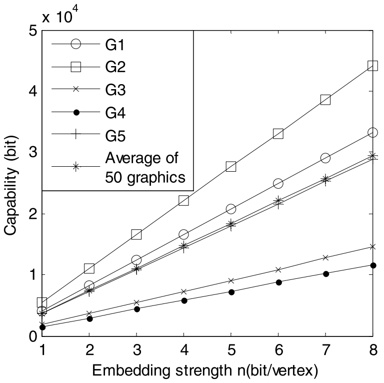


Fig. 9. The relation between the capacity and the embedding strength n.

new coordinate system is $\overset{\frown}{v}_i'\left(\overset{\frown}{x}_i', \overset{\frown}{y}_i'\right)$, and that after RST operations is $\overset{\frown}{\tilde{v}}_i'\left(\overset{\frown}{\tilde{x}}_i', \overset{\frown}{\tilde{y}}_i'\right)$. It has been proved in [18] that

$$\left(\overset{\frown}{\tilde{x}}_i', \overset{\frown}{\tilde{y}}_i'\right) = \left(L\overset{\frown}{x}_i', L\overset{\frown}{y}_i'\right), \tag{28}$$

$$\tilde{R} = L \times R, \tag{29}$$

where L represents the scaling factor.

The relative position between the vertex and its bottom-left vertex is:

$$\begin{aligned}&\left(\overset{\frown}{\tilde{x}}_i - \overset{\frown}{\tilde{x}}_i^{l}, \overset{\frown}{\tilde{y}}_i - \overset{\frown}{\tilde{y}}_i^{l}\right)\\ &= \left(\overset{\frown}{\tilde{x}}_i - \left\lfloor \overset{\frown}{\tilde{x}}_i/\tilde{R} \right\rfloor \times \tilde{R}, \overset{\frown}{\tilde{y}}_i - \left\lfloor \overset{\frown}{\tilde{y}}_i/R \right\rfloor \times \tilde{R}\right)\\ &= \left(L(\overset{\frown}{x}_i - \left\lfloor \overset{\frown}{x}_i/R \right\rfloor \times R), L(\overset{\frown}{y}_i - \left\lfloor \overset{\frown}{y}_i/R \right\rfloor \times R)\right). \end{aligned} \tag{30}$$

Substitute $\overset{\frown}{\tilde{x}}_i'$ into (25), and get

$$\begin{aligned} layer(\overset{\frown}{\tilde{x}}_i') &= \arg_t(\tilde{R}_{t-1} \leq \overset{\frown}{\tilde{x}}_i' - \overset{\frown}{\tilde{x}}_i'^{l} < \tilde{R}_t)\\ &= \arg_t(\frac{\tilde{R}_{t-1}}{\tilde{R}} \leq \frac{\overset{\frown}{\tilde{x}}_i' - \overset{\frown}{\tilde{x}}_i'^{l}}{\tilde{R}} < \frac{\tilde{R}_t}{\tilde{R}})\\ &= \arg_t(\frac{\sqrt{w+1}}{2^{(t-1)/2}} \leq \frac{\overset{\frown}{x}_i' - \overset{\frown}{x}_i^{l}}{R} < \frac{\sqrt{w+1}}{2^{t/2}})\\ &= layer(\overset{\frown}{x}_i'). \end{aligned} \tag{31}$$

Similarly, $layer(\overset{\frown}{\tilde{y}}_i') = layer(\overset{\frown}{y}_i')$ can be also obtained.

According to (25), the watermark extracted from the graphics after RST operations equals the original one. Thus, the proposed method is robust against RST operations.

In order to test the robustness against RST operations, experiments are done to calculate the *NC* (Normalized correlation) between the original watermark and the watermark extracted from the graphics after operations. The *NC* is defined as

$$NC(S, S') = \frac{\sum_{i=0}^{L-1} s_i s_i'}{\sqrt{\sum_{i=0}^{L-1} s_i^2}\sqrt{\sum_{i=0}^{L-1} s_i'^2}}, \tag{32}$$

where S is the original watermark and the S' is the watermark extracted from the graphics after RST operations.

In the experiments, the watermarked graphics are rotated by different angles (rotation angle $\rho\epsilon\{30°, 150°, 180°, 240°\}$), scaled with different factors (scaling factor $\varsigma\epsilon\{0.25, 1.73, 2.24, 3.15\}$), and translated with different translation vectors (translation vector $\vec{v}\epsilon\{(-1.3, -2.2), (-5.4, 4.9), (3.7, -6.8), (0.8, 0.9)\}$), respectively. The results show that the *NC* value of different situations are all 1.0. Thus the proposed scheme is robust against RST operations.

VI. Conclusion

A novel region nesting partition method is designed in this paper. Based on this partition, a reversible watermarking with low distortion is proposed. The region nesting partition optimizes the region partition and significantly improves the imperceptibility of the watermarked graphics compared to the exiting schemes. To the best of our knowledge, our work is the first work on the optimization of the region partition for the reversible watermarking of 2D engineering graphics. Experimental results also show the good imperceptibility of the proposed scheme, good capacity and robustness against RST operations. It is promising in the application of integrity authentication for 2D engineering graphics. However, the upper bound of the imperceptibility of floating point number based reversible watermarking still needs further research. Our future work will concentrate on the theory of imperceptibility and its applications in reversible watermarking for 2D engineering graphics.

Acknowledgment

The authors thank the anonymous reviewers for their kind suggestions for improving this paper, and also thank Dr. Yan Li, who is with Advanced Digital Sciences Center, University of Illinois at Urbana–Champaign for his kind proofreading of this paper.

References

[1] 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.

[2] 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 Security*, vol. 2, no. 3, pp. 311–320, Sep. 2007.

[3] 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 Design*, vol. 43, no. 8, pp. 1018–1024, 2011.

[4] 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.

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

[6] 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.

[7] B. Yang, M. Schmucker, W. Funk, C. Busch, and S. Sun, “Integer DCT-based reversible watermarking for images using companding technique,” *Proc. SPIE*, vol. 5306, pp. 405–416, Jun. 2004.

[8] G. Xuan *et al.*, “Lossless data hiding using histogram shifting method based on integer wavelets,” in *Proc. Int. Workshop Digital Watermarking*, 2006, pp. 323–332.

[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, Mar. 2006.

[10] X. Wu, "Reversible semi-fragile watermarking based on histogram shifting of integer wavelet coefficients," in *Proc. Inaugural IEEE-IES IEEE Digit. EcoSyst. Technol. Conf. (DEST)*, Feb. 2007, pp. 501–505.

[11] C.-C. Lin, W.-L. Tai, and C.-C. Chang, "Multilevel reversible data hiding based on histogram modification of difference images," *Pattern Recognit.*, vol. 41, no. 12, pp. 3582–3591, 2008.

[12] X. Li, W. Zhang, X. Gui, and B. Yang, "A novel reversible data hiding scheme based on two-dimensional difference-histogram modification," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 7, pp. 1091–1100, Jul. 2013.

[13] 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.

[14] D. Coltuc and J.-M. Chassery, "Simple reversible watermarking schemes: Further results," *Proc. SPIE*, vol. 6072, p. 60721X, Feb. 2006.

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

[16] S. Weng, Y. Zhao, J. S. Pan, and R. Ni, "Reversible watermarking based on invariability and adjustment on pixel pairs," *IEEE Signal Process. Lett.*, vol. 15, pp. 721–724, 2008.

[17] B. Ma and Y. Q. Shi, "A reversible data hiding scheme based on code division multiplexing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 1914–1927, Sep. 2016.

[18] X. Li, J. Li, B. Li, and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion," *Signal Process.*, vol. 93, no. 1, pp. 198–205, 2013.

[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] B. Ou, X. Li, Y. Zhao, and R. Ni, "Reversible data hiding based on PDE predictor," *J. Syst. Softw.*, vol. 86, no. 10, pp. 2700–2709, 2013.

[21] 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.

[22] D. Coltuc, "Low distortion transform for reversible watermarking," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 412–417, Jan. 2012.

[23] D. Coltuc, "Improved embedding for prediction-based reversible watermark- ing," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 873–882, Mar. 2011.

[24] 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.

[25] I. C. Dragoi and D. Coltuc, "On local prediction based reversible watermarking," *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1244–1246, Apr. 2015.

[26] I.-C. Dragoi and D. Coltuc, "Adaptive pairing reversible watermarking," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2420–2422, May 2016.

[27] L. An, X. Gao, X. Li, D. Tao, C. Deng, and J. Li, "Robust reversible watermarking via clustering and enhanced pixel-wise masking," *IEEE Trans. Image Process.*, vol. 21, no. 8, pp. 3598–3611, Aug. 2012.

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

[29] S. Zhong, Z. Liu, and Q. Chen, "Reversible watermarking algorithm for vector maps using the difference expansion method of a composite integer transform," *J. Comput. Aided Design Comput. Graph.*, vol. 21, no. 12, pp. 1840–1849, 2009.

[30] L. Cao, C. Men, and R. Ji, "Nonlinear scrambling-based reversible watermarking for 2d-vector maps," *Vis. Comput.*, vol. 29, no. 3, pp. 231–237, 2013.

[31] 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.

[32] P.-C. Wang and C.-M. Wang, "Reversible data hiding for point-sampled geometry," *J. Inf. Sci. Eng.*, vol. 23, no. 6, pp. 1889–1900, 2007.

[33] N. Wang, "Reversible watermarking for 2D vector maps based on normalized vertices," *Multimedia Tools Appl.*, vol. 76, no. 20, pp. 20935–20953, 2017.

[34] L. Cao, C. Men, and R. Ji, "High-capacity reversible watermarking scheme of 2D-vector data," *Signal, Image Video Process.*, vol. 9, no. 6, pp. 1387–1394, 2015.

[35] F. Peng, Q. Long, Z.-X. Lin, and M. Long, "A reversible watermarking for authenticating 2D CAD engineering graphics based on iterative embedding and virtual coordinates," *Multimedia Tools Appl.*, pp. 1–21, 2017.

[36] Z.-X. Lin, F. Peng, and M. Long, "A reversible watermarking for authenticating 2D vector graphics based on bionic spider Web" *Signal Process., Image Commun.*, vol. 57, pp. 134–146, Sep. 2017.

Zi-Xing Lin received the B.S. degree in information security from Hunan University, Changsha, Hunan, China, in 2014, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Electronic Engineering. His areas of interest include multimedia security and digital watermark.

Fei Peng received the Ph.D. 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, 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 School of Computer Science and Electronic Engineering, Hunan University, Changsha. He is also the Director of the Department of Cyber Security. His areas of interest include digital watermarking and digital forensics.

Min Long received the Ph.D. 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, University of Warwick, U.K., from 2009 to 2010. She is currently a Professor with the College of Computer and Communication. Her areas of interest include digital watermarking and chaos-based secure communication.