A Semi-fragile Watermarking Algorithm for Authenticating 2D Engineering Graphics Based on Log-Polar Transformation

Fei Peng[1], Re-Si Guo[1], Chang-Tsun Li[2], and Min Long[3]

[1] College of Computer and Communication, Hunan University, Changsha, PRC, 410082
[2] Department of Computer Science, University of Warwick, Coventry, UK, CV47 AL
[3] College of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, PRC, 410076
pengfei@hnu.cn, hnu_grs@163.com, c-t.li@warwick.ac.uk, longm@tom.com

Abstract. A semi-fragile watermarking algorithm for authenticating 2D engineering graphics based on log-polar coordinates mapping is proposed. Firstly, the coordinates of vertices of entities are mapped to the log-polar coordinate system to obtain the invariance of translation, scaling and rotation. Then the watermark is embedded in the mantissa of the real-valued log-polar coordinates via bit substitution. Theoretical analysis and experimental results show that the proposed algorithm is not only robust against incidental operations such as rotation, translation and scaling, but can also detect and locate malicious attacks such as entity modification, entity addition/deletion.

Keywords: semi-fragile watermarking, integrity verification, data hiding, content authentication, engineering graphics.

1 Introduction

Nowadays, copyright violation and illegal manipulation of multimedia are becoming serious problems in the cyberspace. Traditional cryptographic systems can protect the information to a certain degree, but it still has some drawbacks [1]. On the other hand, watermarking has been viewed as an efficient solution to protect the copyright and the integrity of digital data, thus has been paid significant attention in recent years [2, 3].

The functions of semi-fragile watermarking are similar to those of the digital signatures used in cryptosystems. However, for many multimedia applications, semi-fragile watermarking is superior to digital signature [4].

Since the publication of the first paper on watermarking for 3D models by Ohbuchi [5], many watermarking schemes [6-8] have been proposed to serve the similar purpose. However, little work has been done in watermarking for 2D vector graphics[9-11], which are commonly used in geographical information systems (GIS) and computer aided design (CAD). At the same time, current semi-fragile watermarking researches on vector graphics are mainly focused on 3D models. Therefore, it is our intention in this work to propose a semi-fragile watermarking scheme for verifying the integrity of 2D vector graphics.

D. Weerasinghe (Ed.): ISDF 2009, LNICST 41, pp. 11–18, 2010.

The remainder of the paper is organized as follows. 2D engineering graphics and log-polar coordinate transformation are introduced in Section 2. In Section 3, the watermarking algorithm is described in detail. The analyses of experimental results and performance are conducted in Section 4. Finally, Section 5 concludes the work.

2 2D Engineering Graphics and Log-Polar Transformation

2.1 2D Engineering Graphics

A 2D engineering graphics is a vector graphics composed of entities such as points, lines, arcs, polygons, circles, elliptic circles, etc, and vertex is a basic element of the entities. Every entity also has its own properties such as handle value, color, line type and etc. The characteristics of 2D engineering graphics are as following [12]:

- The data of 2D engineering graphics include geometry and topology information.
- The redundancy of 2D engineering graphics is less than that of raster image.
- There is no concept of sampling rate in 2D engineering graphics, and mathematical operations such as DFT, DCT and DWT cannot be applied to its data directly.

Considering the afore-mentioned characteristics, approaches to watermarking for authenticating 2D engineering graphics is different from those to the watermarking of raster images, audio and video.

2.2 Log-Polar Transformation

Given a point (x_0, y_0) in the Cartesian coordinate system, if it is the origin of the polar coordinate system after transformation, any Cartesian coordinates (x, y) can be transformed into polar coordinates (r, θ) according to Eq. (1) [13]

$$\begin{cases} r = \sqrt{(x - x_o)^2 + (y - y_0)^2} \\ \theta = \tan^{-1} \dfrac{y - y_0}{x - x_0} \end{cases} , \tag{1}$$

where r represents the radial distance from the origin and θ represents the angle. It can be described as a complex number z,

$$z = x + iy = r(\cos\theta + i\sin\theta) = re^{i\theta} . \tag{2}$$

Applying log operation to Eq. (2), i.e.,

$$\ln z = \ln r + i\theta = u(r, \theta) + iv(r, \theta) .$$

We get the transformation from the polar coordinate system to the log-polar coordinate system as formulated in Eq. (3).

$$\begin{cases} u(r, \theta) = \ln r \\ v(r, \theta) = \theta \end{cases} . \tag{3}$$

If a pair of Cartesian or polar coordinates is scaled and rotated by a factor of r_0 and θ_0, respectively, then new polar coordinates would become $(r_0 r, \theta + \theta_0)$. Applying logarithm operation to $(r_0 r, \theta + \theta_0)$, we get

$$\ln z = \ln(r_0 r e^{i(\theta+\theta_0)}) = \ln r + \ln r_0 + i(\theta + \theta_0) = u(r,\theta) + iv(r,\theta), \tag{4}$$

where

$$\begin{cases} u(r,\theta) = \ln r + \ln r_0 \\ v(r,\theta) = \theta + \theta_0 \end{cases}. \tag{5}$$

This indicates that scaling and rotation in the Cartesian coordinate system are equivalent to translation in the radial and angular axes in the log-polar coordinate system, respectively. Specially, if the centre of the log-polar system corresponds to the origin of the Cartesian coordinate system, r is invariant to rotation, θ is invariant to scaling, and both are invariant to translation in the log-polar system.

3 Semi-fragile Watermarking for 2D Engineering Graphics

Given a 2D engineering graphics G, where V represent the vertex set in G. In the proposed watermarking scheme, the vertices in V are first divided into n groups and then a vertex V_W is selected under the control of a secret key as watermarkable while the rest of the group are seen as non-watermarkable $\overline{V_w}$. Finally a content-dependent watermark w_i is embedded into the watermarkable vertex V_W of the corresponding group i. In a 2D engineering graphics, usually the coordinates of the vertices are represented in IEEE-754 double-precision format, as illustrated in Fig. 1.

	Sign	Exponent	Mantissa
bits	63	62 ~ 52	51 ~ 0

Fig. 1. IEEE-754 double-precision floating-point format

3.1 Generation of Watermark

Because each entity in a 2D engineering graphics has a unique handle value, and it is not changed even the entity is modified, so we use the handle values to construct the watermark. In order to improve the accuracy of tamper localization, topological information is considered in the generation of watermark information. Assuming the entity groups is a circular linked list, the generation of a b-bit watermark to be embedded in the i^{th} group of entity E_i is described in Eq. (6),

$$w_i = Intcp((hash(H(V_{W_{i-1}})) \text{ XOR } hash(H(V_{W_i}))), b, K) \tag{6}$$

where $H(\cdot)$ is the method for acquiring the handle value , *hash*( ) represents a hash operation (MD5 is used in our algorithm) and *Intcp*(*A*, *b*, *K*) is a function that returns *b*(*b*>0, *b* is an even integer) bits from a string *A* in a random fashion under the control of a secret key *K*.

3.2 Watermark Embedding

The watermark embedding algorithm is presented as follows.

Step 1. Represent each entity of an *N*-entity 2D engineering graphics with the centroid of its vertices and store all centroids in an *N*-element array *E*. For example, given an entity E_i with *k* vertices $V_{E_i} = \{V_{E_i}^1, V_{E_i}^2, ..., V_{E_i}^k\}, k \geq 1$, the entity is represented by the centroid $(\frac{1}{k}\sum_{j=1}^{k}\left(V_{E_i}^j\right)_x, \frac{1}{k}\sum_{j=1}^{k}\left(V_{E_i}^j\right)_y)$, where $\left(V_{E_i}^j\right)_x$ and $\left(V_{E_i}^j\right)_y$ are the *x* and *y* coordinates of vertex $V_{E_i}^j$.

Step 2. Divide *E* into groups, each composed of *n* entities, and identify one vertex from each group as watermarkable vertex V_W and denote the non-watermarkable vertices as $\overline{V}_M$. The entity grouping and watermarkable entity identification are both under the control of the secret key *K*.

Step 3. Generate watermark w_i according to method described in Section 3.1.

Step 4. For each entity group, apply log-polar transformation to the watermarkable vertex V_W such that

$$\begin{cases} r_{V_M} = \sqrt{((E_V)_x - x_c)^2 + ((E_V)_y - y_c)^2} \\ \theta_{V_M} = \tan^{-1}\dfrac{(V_M)_y - y_c}{(V_M)_x - x_c} \end{cases}, \tag{7}$$

where $x_c = \frac{1}{n-1}\sum_{j\in \overline{V}_M}(V_j)_x$ and $y_c = \frac{1}{n-1}\sum_{j\in \overline{V}_M}(V_j)_y$ .From Eq. (7), we can see that the origin of the log-polar coordinate systems corresponds to the geometrical centre (x_c, y_c) of the *n*-1 non-watermarkable vertices in the Cartesian coordinate system.

Step 5. For each entity group, select *b* bits each, at random according to the secret key *K*, from the mantissa parts of r_{V_M} and θ_{V_M} , and substitute them with the watermark bit sequences w_i. Denote the watermarked vertex as V'_W.. For the convenience of verification, a new line type is used to mark the embedded entity that the watermarked vertex belongs to.

Step 6. For each entity group, perform the reverse log-polar transformation to the watermarked vertex E'_W.

Step 7. For each entity group, select one vertex at random according to the secret key *K* and adjust its coordinates so that the change to the centroid (See Step 1) due to watermark embedding is compensated for. This step is to ensure that the same centroid of each group can be obtained at the watermark verification stage.

3.3 Watermarks Verification

Given a 2D engineering graphics G' as input, the following algorithm is performed to extract and verify the embedded watermark based on the same secret key K used by the Watermark Embedding algorithm.

Step 1. Repeat Step 1 of Watermark Embedding algorithm to form a centroid array E'.

Step 2. According to the line type of the entities, divide E' into several groups, and identify one vertex from each group as watermarked vertex V'_W and denote the non-watermarkable vertices as $\overline{V_M}$. The entity grouping and watermarkable vertex identification are both under the control of the secret key K and a pre-defined line type.

Step 3. Generate watermark w_i according to method described in Section 3.1.

Step 4. For each entity group, apply log-polar transformation to the watermarked vertex V'_W according to Eq. (7)

Step 5. For each entity group, take b bits each, at random according to the secret key K, from the mantissa parts of r'_{V_M} and θ'_{V_M} as the watermark bit sequences w'_{i1} and w'_{i2}. If $w'_{i1} = w_i$ or $w'_{i2} = w_i$ the i^{th} entity group is deemed authentic. Otherwise the group seen as tampered.

4 Experimental Results and Discussion

4.1 Experimental Results

The experiments are carried out on a PC with CPU P4 2.2G, RAM 1G, WinXP Professional, AutoCAD2006,VC++6.0 and DWGDirect 4.2. The results show the watermarking algorithm has a good performance in imperceptibility, robustness and ability of tampering location.

4.2 Discussion of the Robustness and Ability of Tamper Location

A. Analysis of robustness against translation, rotation and scaling

According to the properties of Log-polar transformation described in 2.2, it is easy to find the watermarking algorithm is robust against translation, rotation and scaling.

B. Analysis of ability of tamper localization

1). *Modification.* Generally, modification to entities can be classified into two categories: one is modification of vertices in E_W (watermarked), the other is modification of vertices in $\overline{E_W}$ (non-watermarked). As for the former, it may lead to modification of (r_{E_M}, θ_{E_M}) directly, while the later may lead to modification of coordinates (x_c, y_c), which will also change the value of (r_{E_M}, θ_{E_M}). As a result, the extracted watermark will be different from the original, thus the modification is detected.

2). *Entity addition.* For the entities in a 2D engineering graphics is traversed in temporal order, the newly added entities will be traversed last, and classified into the last

group. During the watermarks verification, the extracted watermark will be different from the generated watermark, thus the added entities are detected.
3). *Entity Deletion.* Entity deletion changes the topology of the 2D engineering graphics. Generally, entity deletion can be classified into three categories. The first category is that all deleted entities are only in $\overline{V_W}$ (non-watermarked). The second category is that some deleted entities are in V_W while some in $\overline{V_W}$. The third category is that the deleted entities form one or more complete embedding group. According to the grouping method described in Section 3.2, the first category only destroys the topology of the group and there is no influence to the relation between groups, which is similar to entity modification. The second category changes the grouping at Step 2 of the verification algorithm and as a result the extracted watermark is different from the original. As for the third category, the extracted watermark in the adjacent groups will be different from the original ones. For all situations will lead to incorrect verification, thus entity deletion is detected.

4.3 Performance Discussion

A. The relationship between embedding distortion and n
We use *root mean square* (*RMS*), as formulated in Eq. (8), to represent distortion inflicted on the graphics by the watermarking algorithm

$$RMS = \frac{1}{N}\left\|V - V'\right\| \tag{8}$$

where V' and V are the sets of vertices in the original 2D engineering graphics and its watermarked counterpart, respectively, and N is the number of vertices. The relationship of *RMS* and n is shown in Fig.2. With the increase of n, *RMS* decrease greatly, which means *RMS* is inversely proportional to n.

B. The relation between RMS and b and the relation between RMS and embedding positions
The bit positions where the watermarks are embedded and the length b of the watermark embedded in each group are another two factors affecting the embedding distortion. A small value of b leads to lower distortion. Theoretically, the embedding distortion would be less perceptible if the watermark is embedded in the least significant bits of the mantissa, but it may also reduce the scheme's authentication power in detecting malicious attacks because a floating number is not an accurate number, especially the value of the end part of the mantissa. On the other hand, embedding watermark in the most significant bits of the mantissa may cause visible distortion to 2D engineering graphics, so, the middle bits of the mantissa is a reasonable compromise between false-negative rate and distortion.

Given fixed watermark length b, the relation between *RMS* and embedding positions is shown in Fig. 3, where *SIM* represents the start embedding position. With the same *SIM*, *RMS* is proportional to b, which demonstrates our analysis above. The *RMS-SIM* relationships when the proposed algorithm is applied to 4 different graphics with b = 8 and n = 5 are shown in Fig. 4. After many experiments, we recommend that *SIM* should take value between 41 and 26, and b between 5 and 8 to ensure a good performance of the algorithm.

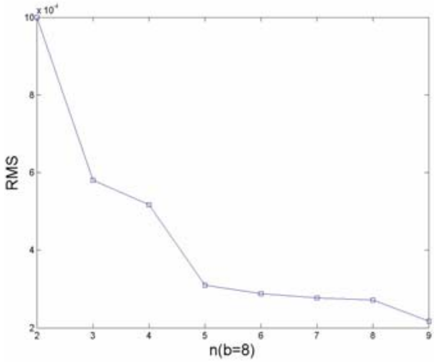

Fig. 2. Relationship between *RMS* and *n*

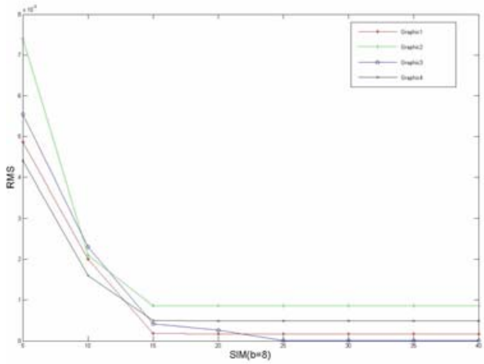

Fig. 4. Different *RMS* – *SIM* relationships of 4 different graphics given $b = 8$ and $n = 5$

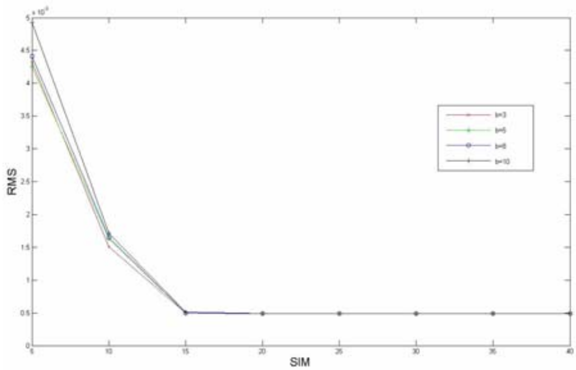

Fig. 3. Relationship between *RMS*- and *b*

5 Conclusions

A semi-fragile watermarking algorithm for verifying the content integrity of 2D engineering graphics is proposed in this work. To strike a good balance between the contradicting requirements of high authentication power, low embedding distortion and high accuracy of tamper localization, the watermark is generated by hashing the content of the graphics and embedded in the middle bits of the mantissa parts of the coordinates of graphical entities. Because the entity coordinates are transformed from the Cartesian system into the log-polar system before the watermark embedding takes place, the algorithm is tolerant to some incidental operations such as rotation, translation and scaling, but sensitive to malicious attacks.

Acknowledgement

This work is supported by Hunan Provincial Natural Science Foundation of China (09JJ4032, 08JJ4020).

References

1. Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital Watermarking and Steganography. Morgan Kaufmann, San Francisco (2008)
2. Yang, Y., Sun, X., Yang, H., Li, C.T., Xiao, R.: A Contrast-Sensitive Reversible Visible Image Watermarking Technique. IEEE Transactions on Circuits and Systems for Video Technology (accepted, 2009)
3. Gupta, G., Pieprzyk, J.: Reversible and Blind Database Watermarking Using Difference Expansion. International Journal of Digital Crime and Forensics 1(2), 42–54 (2009)
4. Li, C.T., Si, H.: Wavelet-based Fragile Watermarking Scheme for Image Authentication. Journal of Electronic Imaging 16(1), 013009-1–013009-9 (2007)
5. Ohbuchi, R., Masuda, H., Aono, M.: Watermarking Three-Dimensional Polygonal Models. In: Proceeding of the ACM Multimedia 1997, pp. 261–272 (1997)
6. Chou, C.M., Tseng, D.C.: A Public Fragile Watermarking Scheme for 3D Model Authentication. Computer-Aided Design 38(11), 1154–1165 (2005)
7. Wang, W.B., Zheng, G.Q.: A Numerically Stable Fragile Watermarking Scheme for Authenticating 3D Models. Computer-Aided Design 40, 634–645 (2008)
8. Wu, H.T., Cheung, Y.-M.: A fragile watermarking scheme for 3D meshes. In: Proceedings of the 7th Workshop on Multimedia and Security, New York, pp. 117–124 (2005)
9. Solachidis, V., Pitas, I.: Watermarking Polygonal Lines Using Fourier Descriptors. IEEE Computer Graphics and Applications 3(24), 44–51 (2004)
10. Gou, H., Wu, M.: Data Hiding in Curves with Application to Fingerprinting Maps. IEEE Trans. Signal Processing 10(53), 3988–4005 (2005)
11. Huang, X.F., Peng, F., Deng, T.: A Capacity Variable Watermarking Algorithm for 2D Engineering Graphic Based on Complex Number System. In: Proceedings of International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 339–342. IEEE Press, Harbin (2008)
12. Huang, X.S., Gu, J.W.: Survey of Watermarking Techniques for CAD Graphic Data. Journal of Engineering Graphic 6, 140–146 (2005) (in chinese)
13. Zhang, W., Ren, X.Y., Zhang, G.L.: An Robust Object Locating Method Based On Log-polar Transform and Affine Transform. Journal of Image and Graphics 11(9), 1255–1259 (2006) (in Chinese)