Robust Hashing for Joint Topology and Geometry Authentication via Covariance-Based Descriptors

Zhiyong Su, Ying Ye, Qi Zhang, and Yuewei Dai

***Abstract*—This paper investigates both the topology and geometry authentication problem of 2D engineering CAD (computer-aided design) graphics, which focus on topological modeling more than geometric modeling of objects. A robust hashing scheme is proposed for joint topology and geometry authentication. The covariance matrices of descriptors are explored to fuse and encode both topology and geometry features with different types into a compact representation. First, a normalized binary shape texture is rendered for each geometric object through the Render-To-Texture technique. Then, for each geometric object, geometry features are computed based on statistical features which are extracted from image rings. And, topology features are generated according to the topology relation among joint objects. To generate hash codes of the graphic, all geometric objects are first grouped according to their geometry features. Then, for each group, the covariance matrices of descriptors are applied to fuse both topology and geometry features of all objects, followed by computing intermediate hash codes of each group based on the covariance matrices. The final hash sequence is formed by concatenating the intermediate hash codes which correspond to each group. Secret keys are introduced both in feature extraction and hash construction. While being robust against topology-preserving graphic manipulations, the hash is sensitive to malicious attacks. By decomposing the hashes, the location of tampered objects can be determined. Experimental results are presented to evaluate the performance and show effectiveness of the method.**

***Index Terms*—Covariance descriptor, authentication, topology authentication, geometry authentication, hash.**

I. Introduction

ENGINEERING CAD (computer-aided design) graphics are very important industrial graphic documentation and are extensively used in Architecture, Engineering and Construction (AEC), as one branch of CAD. With intensive global competition and increasing product complexity in AEC industry, companies are increasingly focusing on collaborative design technologies in which a company concentrates only on its core activity and collaborates with other companies for other activities. These technologies provide a consistent set of solutions to support the collaborative creation, management, dissemination, and use of design documentation through the entire product and project lifecycle [1]. Therefore, integrity and security of engineering CAD graphics sharing among all collaborative participants are essential to successful Product Lifecycle Management (PLM) applications.

Z. Su, Y. Ye, Q. Zhang and W. Dai are with the School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: suzhiyong@njust.edu.cn; 41245246@qq.com; 13905161271@163.com; daiywei@163.com).

Manuscript received April 19, 2005; revised August 26, 2015. This work was supported by the National Natural Science Foundation of China under Grant 61300160.

Digital contents of engineering CAD graphics typically consist of geometry, engineering, and topology information. Geometry information refers to the shape, dimension and position of objects. Geometric shape of objects can be designed by using basic geometric entities, such as LINE, POLYLINE, ARC, CIRCLE and 3DFACE. Engineering information depicts design constraints, engineering disciplines, etc. Topology information describes complex topological relation among various joint objects. The design of engineering CAD graphics focuses on topological modeling more than geometric modeling of objects. The objective of topological modeling is to determine the most economical spatial arrangement of various objects that satisfy construction, operation, maintenance, and safety requirements [2], [3]. This is significantly different from traditional mechanical CAD, as another branch of CAD, which concentrates on geometric modeling. Hence, both topology and geometry information should be taken into account in content authentication.

Content authentication and identification technique can be classified into two main categories from the technological perspective: watermarking and hashing. In terms of watermarking based techniques, watermarks associated with authentication information are embedded into specific area of the content and then are extracted to judge if there are malicious manipulations on the received content. Therefore, in this way, the precision of host content can be inevitably changed slightly by watermarking [4], [5]. This is an important problem in highly detailed digital design graphics in CAD applications. Different from watermarking based techniques, the hashing based schemes require no embedding process. Hash codes are generated based on well designed features extracted from the host content that are in accordance with certain characteristics. Content authentication is performed via comparing the hash codes of the host content with the hash codes of the received content [6]–[8]. Therefore, hashing based techniques do not introduce any distortion to the host content and are generally more suitable for CAD applications.

To the best of our knowledge, there is no related work that provides a detailed analysis of authenticating both topology and geometry information for 2D engineering CAD graphics in the literature. In case of geometry authentication, a large number of digital watermarking schemes have been recently proposed for mechanical CAD graphics [4], [9]–[12]. And, few hashing based authentication schemes have been proposed for vector data models [13], [14]. In terms of topology authentication, by comparison, few related works have been reported. The topology authentication problem of piping isometric drawings, as a kind of 2D engineering CAD graphics, was first

investigated by Su et al. [15] and a watermarking based scheme was proposed to verify just the topology integrity. Therefore, the problem of joint topology and geometry authentication for 2D engineering CAD graphics has not been reported and addressed yet.

A. Contributions

In this paper, we aim to tackle the problem of joint topology and geometry information authentication for 2D engineering CAD graphics. The contributions of this paper can be summarized as follows.

(1) A novel framework for jointly authenticating topology and geometry information of 2D engineering CAD graphics is proposed in this paper for the first time. The framework decomposes the authentication task into three stages: topology and geometry features extraction, topology and geometry features fusion, as well as joint topology and geometry hashing.

(2) Geometry features of geometric objects are extracted in the image space rather than in the geometric space through ring partition [8], [16]. The proposed descriptor is robust to a wide range of nonmalicious manipulations such as global and local RST transformations (rotation, uniform scaling and translation) by incorporating the shape texture rendering method for geometric objects.

(3) Covariance matrices are proposed as a new descriptor for fusion of topology and geometry features. While similar descriptors have been proposed for object tracking and texture analysis in 2D images, it is the first time that covariance-based analysis is explored for content authentication of CAD graphics in the literature. The advantage of using covariance matrices compared with geometric descriptors is that they enable the fusion of multiple and heterogeneous features without the need for normalization [17], [18].

(4) A hashing based scheme is proposed to authenticate topology and geometry information of 2D engineering CAD graphics. The proposed method is robust to a wide range of nonmalicious manipulations such as global and local RST transformations while it is also sensitive to topology and geometry changes caused by malicious attacks. Furthermore, it can detect and locate tampered objects.

The rest of this paper is organized as follows. Section II reviews the related work. Section III introduces the preliminaries used in this paper. Section IV overviews the framework of the proposed scheme. Details of the proposed hashing scheme are described in Section V, Section VI, and Section VII, respectively. Section VIII presents the performance analysis and experimental results. This work is concluded in Section IX.

II. Related work

This section reviews some related works with respect to geometry and topology authentication for CAD models .

A. Geometry authentication

Existing works regarding geometry authentication for CAD models in the literature can be divided into two main categories: watermarking based methods and hashing based methods.

Watermarking based methods: There are many watermarking methods for geometry authentication of CAD models reported in the past years [19], [20]. Fornaro et al. [21] proposed a distributed watermarking scheme for verifying CSG (Constructive Solid Geometry) models. Watermarks were computed from selected attributes of the model and then were stored in control nodes or in comments of the model. Peng et al. [12] presented two reversible watermarking schemes, which can be applied for content authentication, for 2D CAD engineering graphics based on histogram shifting. Both schemes exploited the correlation of adjacent coordinates or relative phases. Watermarks were embedded by shifting and modifying the difference histogram of coordinates or phase. Xiao et al. [4] introduced a combined reversible watermarking scheme for 2D CAD engineering graphics. Watermarks were embedded into the distance ratios of vertices through improved quantization index modulation and improved difference expansion.

Hashing based methods: A information-theoretic hashing of a 3D mesh using spectral graph theory and entropic spanning trees was presented by K. Tarmissia [22]. The scheme applied Eigen-decomposition to the Laplace-Beltrami matrix of each sub-mesh then generates the hash value based on the spectral coefficients and the Tsallis entropy estimate. Lee et al. [14] proposed a vector data hashing method for authentication and copyright protection of CAD design graphics. Feature values were extracted by projecting the polyline curvatures, which are obtained from groups of vector data using GMM (Gaussian mixture model) clustering, onto random values. The final hash values were generated based on the binarization of the feature values.

B. Topology authentication

The problem of topology authentication for engineering CAD graphics in the AEC industries is relatively new compared with existing image, video, 3D model and vector data hashing and has not been researched as widely compared with geometry authentication. Su et al. [15] first investigated the topology integrity authentication problem for piping isometric drawings, as a kind of 2D engineering CAD graphics. A semi-fragile watermarking scheme was proposed to address the referred interesting issue. Topological relation among joint components was encoded into singular watermarks. Authentication was achieved by embedding topology sensitive watermarks into geometrical invariants of selected objects via quantization index modulation.

All in all, although great progress has been made in geometry authentication for CAD models, there still are very few methods that focus on topology authentication. Furthermore, the problem of joint topology and geometry authentication for 2D engineering CAD graphics has not been well investigated and addressed yet in the literature. Therefore, this paper aims at developing hashing based methods to jointly authenticate topology and geometry information for 2D engineering CAD graphics.

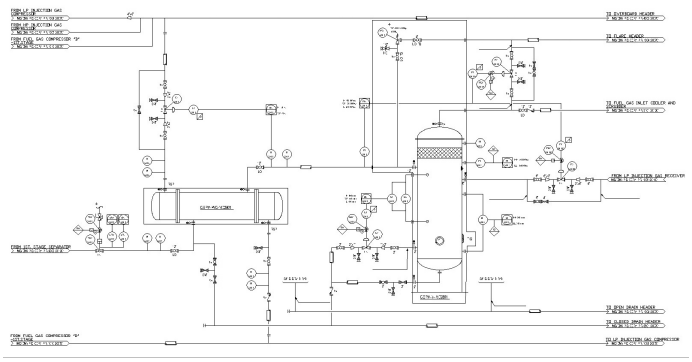


Fig. 1. Part of a typical 2D engineering CAD graphic.

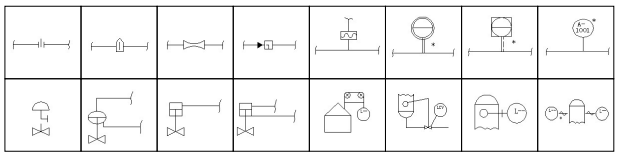


Fig. 2. Some geometric objects used in 2D engineering CAD graphics.

III. Preliminaries

A. 2D Engineering CAD Graphics

2D engineering CAD graphics consist of a number of geometric objects. Fig. 1 shows part of a typical 2D engineering CAD graphic. And Fig. 2 gives some geometric objects used in 2D engineering CAD graphics. Geometric objects are composed of various basic geometric entities such as LINE, POLYLINE, CIRCLE, ARC and POLYGON. These objects often have complex external and internal shape, as illustrated in Fig. 2. In terms of geometry and topology information, without loss of generality, a 2D engineering CAD graphic $\mathbf{G}$ may be defined as a undirected graph $\mathbb{G} = (\mathbb{O}, \mathbb{E})$, where $\mathbb{O} = \{\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_m\}$ is the set of nodes, $\mathbb{E} = \{\mathbf{e}_{ij}\}$ is the set of edges. Each node $\mathbf{o}_i$ corresponds to a geometric object. Each edge $\mathbf{e}_{ij} = [\mathbf{o}_i, \mathbf{o}_j]$ indicates that $\mathbf{o}_i$ connects with $\mathbf{o}_j$.

2D engineering CAD graphics can be easily edited through various geometry and topology operations provided by CAD tools. These operations can be classified into nonmalicious operations and malicious operations. Hash codes are expected to be able to survive nonmalicious operations and reject malicious tampering within an acceptable extent. Nonmalicious operations cover global and local RST transformations. Global RST transformations are performed on the whole graphic to have a better view. While local RST transformations are often applied to certain individual objects to achieve a satisfactory appearance and fit. These geometry operations are applied to create a cleaner and more legible graphics and further facilitate the annotation for various objects. And, they affect the position, dimension and orientation of objects based on the precondition of keeping the topology relation unchanged. Malicious operations cover inserting objects, deleting objects, and changing topology relations logically. Inserting and deleting objects, which can be defined as malicious geometry attacks, always involve topology modification. It should be pointed out that all the above operations are performed on objects rather than their geometric entities.

B. Vector Quantization

The Vector Quantization (VQ) technique is used to make clusters for all geometric objects in this paper. It was formerly introduced as image compression technique and proved to be efficient [23].

VQ can be simply regarded as a mapping function which maps the m-dimensional space R^m into a finite subset $Y = \{Y_0, Y_1, \cdots, Y_{k-1}\}$, where Y is called codebook with k codewords, $Y_j = \{Y_j^0, Y_j^1, \cdots, Y_j^{m-1}\}$ is the j-th codeword in the codebook Y. Codebook training is performed in advance through the Linde-Buzo-Gray (LBG) algorithm [24] in this paper. The details of the LBG algorithm are given as follows:

Step1 : Generate an initial codebook Y^0 of size k. Set the iteration counter $i = 0$ and the initial average distortion $D_{-1} = \infty$. Set the maximum iteration counter as I and the distortion threshold as ε.

Step2 : For each training vector x, find its best match codeword with the least distortion in the current codebook Y^i through calculating the Euclidean distance between each codeword and the input vector x.

Step3 : Assign the training vectors into k cells and update the centroid of each cell to obtain a new codebook Y^{i+1}.

Step4 : Calculate the current average distortion D_i for all training vectors at the i-th iteration.

Step5 : If $(D_{i-1} - D_i)/D_i \leq \varepsilon$ or $i = I$, set the ultimate codebook $Y = Y^{i+1}$ and the LBG algorithm is completed. Otherwise, let $i = i+1$, return to Step 2.

C. Covariance Descriptor

The covariance descriptor, which was first introduced by Tuzel et al. [17] for object detection and texture classification, is employed to fuse and represent topology and geometry features of 2D engineering CAD graphics in this paper.

From a statistics point of view, covariance can be understood as a measure of how several variables change together. Within the context of the descriptor definition, the set of random variables must correspond to a set of observable features that are correlated to each other [18], [25]. Given an image $I \in R^{W \times H}$, let $F(x, y)$ be the $W \times H \times d$ dimensional feature image extracted from I,

$$F(x, y) = \phi(I, x, y) \tag{1}$$

where the function ϕ can be any pixel-wise mapping such as intensity, color, gradients, filter response, as well as higher-order derivatives, etc. For a given rectangular region $R \in F$, let $\{z_i\}_{i=1}^n$ be the d-dimensional feature points inside R, then the region R can be described using a $d \times d$ covariance matrix of their points [17],

$$C_R = \frac{1}{n-1} \sum_{i=1}^{n} (z_i - \mu)(z_i - \mu)^T \tag{2}$$

where μ is the mean of the feature vectors of all points in the region.

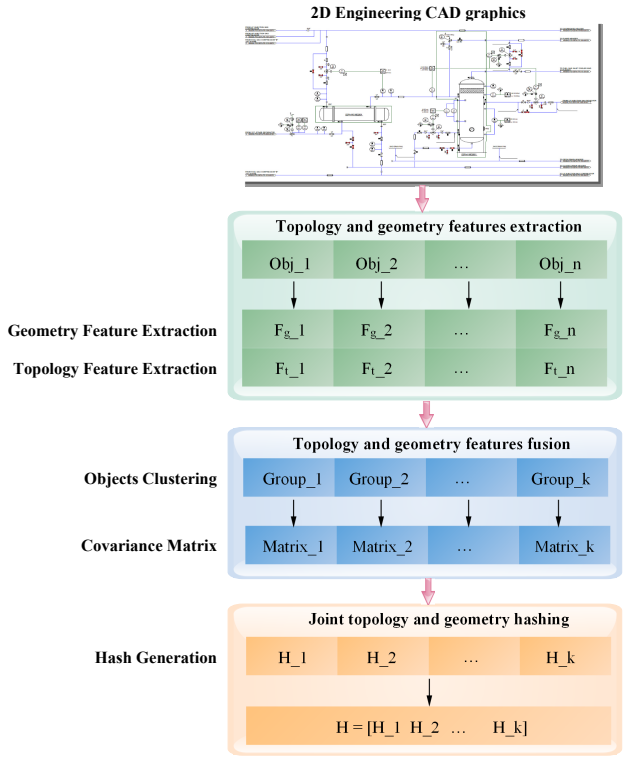


Fig. 3. Overview of the proposed framework.

IV. Overview of the Framework

The framework of the proposed hashing scheme consists of three major parts: topology and geometry features extraction, topology and geometry features fusion, as well as joint topology and geometry hashing. The flow chart of the authentication framework is shown in Fig. 3.

In the topology and geometry features extraction part, for each geometric object in the graphic, the binary shape texture is first rendered and then the geometry feature is computed based on the ring partition. Topology feature is extracted according to its topology relation. In the topology and geometry features fusion part, all objects are clustered into k groups with different number of objects on the basis of their geometry features. Then, for each group, a covariance matrix that encodes the topology and geometry features of objects in the group is computed. In the joint topology and geometry hashing part, a feature vector for each group is constructed according to its covariance matrix. To reduce hash length and improve convenience for storage, a gaussian random matrix is used to compress the feature vector to get an intermediate hash, which is then pseudo-randomly scrambled based on a secret key. Encryption and randomization are utilized to reduce hash collisions to improve the security of the algorithm. The final hash sequence is generated by concatenating the intermediate hash which corresponds to each group.

V. Topology and Geometry Features Extraction

A. Geometry Feature Extraction

For each geometric object $\mathbf{o}_i$, its geometry feature $\mathbf{v}_i^g$ is computed in the image space rather than in the geometric space as illustrated in Fig. 4, because of their complex contours and internal structures. A normalized binary texture is firstly generated through projecting $\mathbf{o}_i$ onto a fixed size texture orthogonally. Then, the rendered texture is divided into different rings. Finally, its geometry feature $\mathbf{v}_i^g$ is computed through statistical features extracted from each ring.

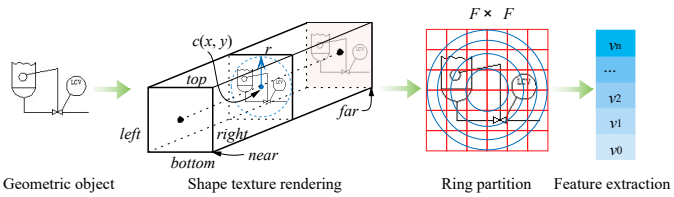


Fig. 4. Illustration of geometry feature extraction.

1) Shape texture rendering: A normalized $F \times F$ binary texture T is rendered for each geometric object $\mathbf{o}_i$ through the Render-To-Texture technique [26] as illustrated in Fig.4. First, an empty texture T in which we're going to render is created. Then, the smallest enclosing circle with its center $c(x, y)$ and radius r of $\mathbf{o}_i$ is computed. These parameters are further utilized to define the six parameters ($left$, $right$, top, $bottom$, $near$, far) of the projection matrix as illustrated in Fig.4. Finally, the object $\mathbf{o}_i$ is rendered to the texture T in terms of orthographic projection. It is obvious that the rendered normalized texture is invariant to object translation and uniform scaling.

2) Ring Partition: Ring partition [8], [16] is employed to extract geometry features which are resilient to object rotation. The rendered normalized texture is divided into a set of rings with equal area as illustrated in Fig.4. It is theoretically proved that the region in the inscribed circle of an image is still the same after rotation [8], [16]. This provides us an opportunity to extract image features resilient to rotation.

Given a normalized $F \times F$ texture T, let n be the ring number, r_m be the m-th radius ($m = 0, 2, \cdots, n-1$) arranged in ascending order, and $\mathbf{R}_m$ be the set of those pixel values of the m-th ring. Clearly, $r_{n-1} = \lfloor F/2 \rfloor$ for the texture T. And r_m can be determined by iteratively calculating the following equation:

$$r_m = \sqrt{\frac{\bar{S} + \pi r_{m-1}^2}{\pi}} \tag{3}$$

where

$$r_0 = \sqrt{\frac{\bar{S}}{\pi}} \tag{4}$$

and $\bar{S}$ is the average area of each ring

$$\bar{S} = \lfloor S/n \rfloor \tag{5}$$

in which S is the area of the inscribed circle

$$S = \pi r_{n-1}^2 \tag{6}$$

Thus, image pixels $p(x, y)$ $(0 \le x \le F-1, 0 \le y \le F-1)$ can be classified into different sets by comparing their distances to the image center with these radii

$$\mathbf{R}_0 = \{p(x, y) | d_{x,y} \le r_0\} \tag{7}$$

$$\mathbf{R}_m = \{p(x,y)|r_{m-1} \le d_{x,y} \le r_m\}(m = 1, 2, \cdots, n-1) \tag{8}$$

where $d_{x,y}$ is the Euclidean distance from $p(x,y)$ to the image center (x_c, y_c) which is defined as:

$$d_{x,y} = \sqrt{(x-x_c)^2 + (y-y_c)^2} \tag{9}$$

where $x_c = y_c = F/2+0.5$ if F is an even number. Otherwise, $x_c = y_c = (F+1)/2$.

3) Feature extraction: Four statistics are chosen to efficiently capture visual content of each ring $\mathbf{R}_m$ $(m = 0, 1, \cdots, n-1)$, i.e., mean (μ_m), variance (δ_m), skewness (s_m), and kurtosis (w_m), which are defined as follows:

$$\mu_m = \frac{1}{N_m} \sum_{i=0}^{N_m-1} R_m(i) \tag{10}$$

$$\delta_m = \frac{1}{N_m - 1} \sum_{i=0}^{N_m-1} (R_m(i) - \mu_m)^2 \tag{11}$$

$$s_m = \frac{\frac{1}{N_m}\sum_{i=0}^{N_m-1}(R_m(i)-\mu_m)^3}{\left(\sqrt{\frac{1}{N_m}\sum_{i=0}^{N_m-1}(R_m(i)-\mu_m)^2}\right)^3} \tag{12}$$

$$w_m = \frac{\frac{1}{N_m}\sum_{i=0}^{N_m-1}(R_m(i)-\mu_m)^4}{\left(\sqrt{\frac{1}{N_m}\sum_{i=0}^{N_m-1}(R_m(i)-\mu_m)^2}\right)^2} \tag{13}$$

where $N_m = card(\mathbf{R}_m)$ is the total number of elements in $\mathbf{R}_m$, and $R_m(i)$ is the i-th element of $\mathbf{R}_m$ $(0 \le i \le N_m - 1)$. Consequently, these statistics of each ring are exploited to form the geometry feature vector $\mathbf{v}_i^g$ which contains $(4 \times n)$ elements.

$$\mathbf{v}_i^g = [\mu_0, \delta_0, s_0, w_0, \cdots, \mu_{n-1}, \delta_{n-1}, s_{n-1}, w_{n-1}] \tag{14}$$

B. Topology Feature Extraction

For each object $\mathbf{o}_i$, a fixed dimensional topology feature vector $\mathbf{v}_i^t$ is formed according to its topology relation

$$\mathbf{v}_i^t = [n_{max}, \mathbf{v}_1^g, \cdots, \mathbf{v}_{n_{max}}^g] \tag{15}$$

where $\mathbf{v}_j^g (0 \le j \le n_{max} - 1)$ is the j-th joint object of $\mathbf{o}_i$, n_{max} is the maximum number of joint objects. Elements of geometry feature vectors of the missing ones are set to zero, if the object $\mathbf{o}_i$ has less than n_{max} joint objects. The number n_{max} and geometry feature vectors of all joint objects together form a topology feature vector $\mathbf{v}_i^t$, which contains $(1 + n_{max} \times 4 \times n)$ elements.

VI. Topology and Geometry Features Fusion

A. Objects Clustering

To facilitate tampering localization as well as ensure that the generated hash has a fixed length and the same computation complexity, for a given 2D engineering CAD graphic $\mathbf{G}$, all objects are clustered into k groups $\{\mathbb{G}_j,\ 0 \le j \le k-1\}$ according to their geometry features, using the vector quantization technique [23]. Thus, those objects with similar shape are clustered into the same group.

A large number of geometric objects of 2D engineering graphics are collected and chosen to train a codebook Y through LBG [24]. Codebook size is predefined as k and the influence of different codebook sizes is analyzed further in Section VIII-C. With VQ, for the geometry feature vector $\mathbf{v}_i^g$ of each object $\mathbf{o}_i$, we find the best match codeword Y_j and its index j, then we assign the object $\mathbf{o}_i$ to the j-th group $\mathbb{G}_j$.

B. Covariance Matrix for Fusing of Geometry and Topology Features

For each group $\mathbb{G}_j$ with n_j objects, a covariance matrix is built for fusing of topology and geometry features of all objects in $\mathbb{G}_j$.

There are several good reasons for using a covariance matrix to characterize the group $\mathbb{G}_j$. First, it provides an elegant mechanism for fusing heterogeneous features of arbitrary dimension and scale. It captures not only the geometry but also the topology features of objects in each group, thus characterizing the graphic. Second, it has a fixed dimension independently of the size of the group. Third, it is compact and easy to compute. Owing to the symmetry, a covariance matrix has only different elements which is small compared with many other region descriptors.

First of all, a feature selection function $\Phi(\mathbb{G}_j)$ is defined for a given group $\mathbb{G}_j$:

$$\Phi(\mathbb{G}_j) = \{\mathbf{v}_i,\ \forall \mathbf{o}_i \text{ s.t. } \mathbf{o}_i \in \mathbb{G}_j,\ 0 \le i \le n_j - 1]\} \tag{16}$$

where $\mathbf{v}_i$ is the feature vector that encodes topology and geometry properties of each object $\mathbf{o}_i$, and is defined as:

$$\mathbf{v}_i = [\mathbf{v}_i^g, \mathbf{v}_i^t] \tag{17}$$

where $\mathbf{v}_i^g$ is the geometry feature vector, and $\mathbf{v}_i^t$ is the topology feature vector.

Then, a $d \times d$ Symmetric Positive Definite (SPD) covariance matrix $\mathbf{M}_{\mathbb{G}_j}$ is defined to represent the given group $\mathbb{G}_j$:

$$\begin{aligned}\mathbf{M}_{\mathbb{G}_j} &= \frac{1}{n_j - 1} \sum_{i=0}^{n_j-1} (\mathbf{v}_i - \mu)(\mathbf{v}_i - \mu)^T \\ &= \begin{bmatrix} m(1,1) & m(1,2) & \dots & m(1,d) \\ \dots & \dots & \dots & \dots \\ m(d,1) & m(d,2) & \dots & m(d,d) \end{bmatrix}\end{aligned} \tag{18}$$

where μ is the mean of the set of feature vectors $\{\mathbf{v}_i\}$ computed in the group $\mathbb{G}_j$, $d = 1 + (n_{max} + 1) \times 4 \times n)$. The diagonal elements of the covariance matrix represent the variance of each one of the feature distributions, and the non-diagonal elements will represent their pairwise correlations.

VII. Joint Topology and Geometry Hashing

A. Hash Generation

For each group $\mathbb{G}_j$, we zigzag the upper triangular elements of $\mathbf{M}_{\mathbb{G}_j}$, which is a symmetric positive definite (SPD) matrix, to obtain the following vector:

$$\begin{aligned}\mathbf{v}_j^m = [&m(1,1), \cdots, m(1,d), \\ &m(2,2), \cdots, m(2,d), \\ &\cdots, \\ &m(d,d)]\end{aligned} \tag{19}$$

1) Compression and Projection: A gaussian random matrix $\mathbf{M}^g$ derived from the compressive sensing model is generated and employed to reduce the dimensionality of the vector $\mathbf{v}_j^m$. To obtain a compressed vector $\mathbf{v}_j^{mc}$, the equation (20) is used to achieve compression and projection:

$$\mathbf{v}_j^{mc} = \mathbf{M}^g \cdot (\mathbf{v}_j^m)^T \tag{20}$$

where $\mathbf{M}^g$ is a $s \times (d(d+1)/2)$ matrix, $s = \lfloor d(d+1)/2 \times p \rfloor$, p is the projection rate and is selected via the experiments. Finally, a compressed s-dimensional vector $\mathbf{v}_j^{mc}$ is generated.

2) Encryption and Randomization: To increase the security of the proposed hashing algorithm, a deterministic chaotic map is employed to generate a chaotic sequence which is extremely sensitive to initial condition [7]. The function used in this paper is the logistic difference equation:

$$y_{n+1} = a y_n (1 - y_n) \tag{21}$$

where a is the function seed, y_n is a number between 0 and 1 and it is the current value of the mapping in time with an initial value y_0. The sequence iterated with the initial value is chaotic when $a > 3.5699456$. Let $y = (y_0, y_1, \cdots, y_{s-1})$ be the generated chaotic sequence. The compressed vector $\mathbf{v}_j^{mc}$ can be randomized by

$$\begin{aligned} \tilde{\mathbf{v}}_j^{mc} &= (\tilde{\mathbf{v}}_{j,0}^{mc}, \tilde{\mathbf{v}}_{j,1}^{mc}, \cdots, \tilde{\mathbf{v}}_{j,s-1}^{mc}) \\ &= (\mathbf{v}_{j,0}^{mc} \times y_0, \mathbf{v}_{j,1}^{mc} \times y_1, \cdots, \mathbf{v}_{j,s-1}^{mc} \times y_{s-1}) \end{aligned} \tag{22}$$

Then, a intermediate binary hash $\mathbf{h}_j$ for each group $\mathbb{G}_j$ is generated through thresholding

$$\mathbf{h}_j = [h(0), \cdots, h(s-1)] \tag{23}$$

where

$$h(i) = \begin{cases} 1, & \tilde{\mathbf{v}}_{j,i}^{mc} > T_j \\ 0, & \tilde{\mathbf{v}}_{j,i}^{mc} \le T_j \end{cases}, \quad 0 \le i \le s-1 \tag{24}$$

$$T_j = \frac{1}{s} \sum_{i=0}^{s-1} \tilde{\mathbf{v}}_{j,i}^{mc} \tag{25}$$

3) Hash Construction: The intermediate hash $\mathbf{h}_j$ of each group $\mathbb{G}_j$ is concatenated to form the final hash sequence, namely $\mathbf{h}$.

$$\mathbf{h} = [\mathbf{h}_1, \cdots, \mathbf{h}_k] \tag{26}$$

It is clear that the length of our hash $\mathbf{h}$ is $(k \times s)$ bits. To guarantee the uniqueness of the final hash and then facilitate the authentication stage, the k groups should be arranged in advance. This can be achieved through sorting the codewords in Y according to their vector component values in sequence. By doing this, a sorted codebook Y is achieved. And then, k groups and their hash codes are arranged consequently.

B. Group-level Tampering Detection and Localization

The proposed hashing scheme is designed to yield group-level tampering detection and localization ability through comparing a distance metric to measure the similarity between hash values of each group. Regarding malicious geometry and topology modifications, it is difficult to locate the tampered objects accurately because of the trade-off between compactness of hash codes and sensitivity to malicious tampering. Provided that the hash $\mathbf{h}$ of a trusted engineering CAD graphic $\mathbf{G}$ is available and called the reference hash. The hash of a received engineering CAD graphic $\mathbf{G}^{'}$ to be tested, $\mathbf{h}^{'}$, is extracted using the above method. An object group can be considered as tampered if it contains maliciously modified objects, and the change of objects can be measured via distances between hash values of the trusted graphic and the tested graphic in the corresponding group. Here, two graphics having the same contents do not need to have identical geometry information, except topology information, since objects may be modified by topology preserving operations such as rotating, uniform scaling and translation, as discussed in Section III-A.

The graphic authentication process is performed in the following way.

Step 1: For the received reference hash $\mathbf{h}$, decompose it into k groups $\{\mathbf{h}_j\}(j = 0, \ldots, k-1)$ according to the pre-trained codebook Y. Each group has s bits.

Step 2: For the received graphic $\mathbf{G}^{'}$, extract the geometry feature $\mathbf{v}_i^g$ and then the topology feature $\mathbf{v}_i^t$ of each object.

Step 3: Cluster all the objects into k groups according to their geometry features with the given codebook Y.

Step 4: Compute the covariance matrix $\mathbf{M}_{\mathbb{G}_j^{'}}$ for fusing of topology and geometry features of objects in each group $\mathbb{G}_j^{'}$.

Step 5: Generate the intermediate hash code $\mathbf{h}_j^{'}$ of each group $\mathbb{G}_j^{'}$, and then form the final hash sequence $\mathbf{h}^{'}$.

Step 6: To measure the similarity between group $\mathbb{G}_j$ and group $\mathbb{G}_j^{'}$, the normalized Hamming distance d_{group} is exploited as a metric:

$$d_{group}(j) = \frac{1}{s} \sum_{m=0}^{s-1} |\mathbf{h}_j^{'}(m) - \mathbf{h}_j(m)|^2 \tag{27}$$

where $\mathbf{h}_j^{'}(m)$ and $\mathbf{h}_j(m)$ are the m-th elements of $\mathbf{h}_j^{'}$ and $\mathbf{h}_j$, $0 \le j \le k-1$, respectively. Thus, the normalized Hamming distance $D_{graphic}$ for graphic similarity measurement is defined as:

$$D_{graphic} = max(d_{group}(0), d_{group}(1), \cdots, d_{group}(k-1)) \tag{28}$$

Step 7: $\mathbb{G}_j$ and $\mathbb{G}_j^{'}$ are said to be functionally identical if $d_{group}(j) < T$, where T is a threshold. Else, the group $\mathbb{G}_j^{'}$ is a tampered version of $\mathbb{G}_j$ or is different from $\mathbb{G}_j$. Furthermore, $\mathbf{G}$ and $\mathbf{G}^{'}$ should be considered functionally identical if $D_{graphic} < T$. Otherwise, they are different graphics or one is a tampered version of the other.

VIII. Performance Analysis and Experimental Results

In this section, various experiments are carried out to evaluate the performance of the proposed hashing scheme for 2D engineering CAD graphics with respect to robustness, sensitivity, discriminative capability and security.

A. Graphic Data Sets

Taking the process plant in AEC industry for example, 40 different 2D engineering CAD graphics with various number

TABLE I
NONMALICIOUS OPERATIONS AND PARAMETER VALUES

Operations	Parameters	Number of graphics
Global rotation	90, 180	2
Global scaling	0.5, 2	2
Global translation	100(x), 100(y)	2
Local rotation (20% objects)	90, 180	2
Local scaling (20% objects)	0.5, 2	2
Local translation (20% objects)	1.5(x), -2(y)	2
Total		12

TABLE II
MALICIOUS OPERATIONS AND PARAMETER VALUES

Operations	Parameters	Number of graphics
Inserting objects	5, 25	2
Deleting objects	5, 10	2
Changing topology logically	10	1
Total		5

of objects (including 10 graphics with about 50 objects, 10 graphics with about 100 objects, 10 graphics with about 300 objects, and 10 graphics with about 500 objects) are tested. To train the codebook Y, an object database containing about 106 different kinds of objects collecting from a large number of 2D engineering CAD graphics of process plants is also constructed.

Detailed parameter settings of nonmalicious and malicious operations are presented in Table I and II respectively. It can be seen that each test graphic has 12 nonmalicious attacked versions and 5 malicious attacked versions with different tampering ratios. Therefore, $40 \times 12 = 480$ pairs of identical graphics are used for robustness validation, $40 \times 5 = 200$ pairs of similar graphics are used for sensitivity validation, and $40 \times (40-1)/2 = 780$ pairs of different graphics are used for discrimination test.

B. Performance Criteria

To discuss the performance in detail, true positive rate (TPR) P_{TPR} and false positive rate (FPR) P_{FPR} are defined:

$$P_{\text{TPR}} = \frac{N_{\text{similar}}}{N_{\text{identical}}} \tag{29}$$

$$P_{\text{FPR}} = \frac{N_{\text{distinct}}}{N_{\text{different}}} \tag{30}$$

where N_{similar} is the number of pairs of functionally identical graphics correctly identified as same graphics, $N_{\text{identical}}$ is the total pairs of functionally identical graphics, N_{distinct} is the number of pairs of distinct graphics mistakenly considered as same graphics, and $N_{\text{different}}$ is the total pairs of different graphics.

C. Parameter Setting

To achieve satisfactory performance, parameters used in the proposed hashing scheme are estimated via experiments. In the experiments, the used parameters are as follows. The rendered binary shape texture size is 100×100 ($F = 100$). Small F leads to loss of fine details, while large F results in high computation complexity. We choose $F = 100$ as an appropriate trade-off. The number n_{max} in equation (15) is determined in accordance with the specific application areas. For example, in our case, n_{max} is set to 4 since the maximum number of joint objects will not exceed 4 in the process industry in general. The logistic function in equation (21) is seeded with the value $a = 4$ and $y_0 = 0.20160614$ for 2000 iterations. The group number k is equal to the size of the codebook Y. The normalized Hamming distance in Equation (27) is used to measure the hash distances between corresponding groups. Considering that the proposed method presents a satisfactory performance for all tested operations when $T \geq 0.2$, we set $T = 0.2$.

1) Group Number & Codebook Size k: In order to facilitate detecting and locating tampered objects, all objects are clustered into k groups according to the codebook Y with k codewords in the preprocessing step for each graphic. The proposed scheme is designed to yield the group-level tamper detection and localization capability. Therefore, large k will certainly result in fewer objects in each group and then give rise to high tamper detection and localization capability. However, total hash length which depends on k and s will also increase with the increasement of k. Thus, it is a trade-off between total hash length and tamper localization capability. Meanwhile, it is also a trade-off among total hash length, sensitivity, and discriminative capability. Generally, compact hash codes include less graphic information, which will contribute to stronger robustness. However, discriminative capability and sensitivity will be weaker. In contrast, hash values of longer length will include abundant graphic information and hence will contribute to ideal tampering localization functionality. Thus, discriminative capability as well as sensitivity will be stronger, and robustness will be weaker. In this paper, from the practical application point of view, we set k to 25 as an appropriate trade-off among tamper localization capability, discriminative capability, sensitivity, and total hash length.

2) Ring number n: Geometry features which are resilient to object translation, scaling, and especially rotation are extracted through dividing the rendered normalized binary texture into n rings. Theoretically, large n will no doubt bring about better object discrimination performance. However, this will lead to greater geometry feature vector dimension and further higher computational complexity. To select a proper n for ring partition, experiments are conducted on the constructed object database with 106 different kinds of objects. The geometry feature vector $\mathbf{v}_i^g$ in equation (14) is first formed for each object with 3 different numbers ($n = 2, 4, 6$). Then, for each number, euclidean distances between each pair of feature vectors are calculated, and $106 \times (106-1)/2 = 5565$ results are finally obtained. Distribution of these euclidean distances is illustrated as shown in Fig.5, where the x-axis is the number of different object pairs and the y-axis is the value of euclidean distance. Statistics of euclidean distances under different ring numbers are also computed and given in Table III. It is observed that the proposed scheme achieves better object discrimination power when $n \geq 4$. Therefore, we choose $n = 4$ as an appropriate trade-off between discrimination performance and computational complexity.

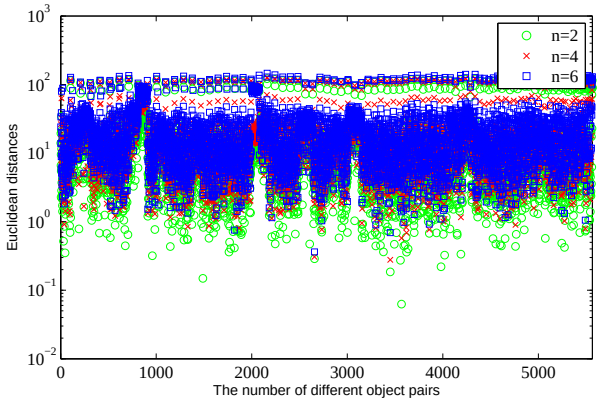


Fig. 5. Euclidean distances between each pair of feature vectors of 106 different kinds of objects.

TABLE III
STATISTICS OF EUCLIDEAN DISTANCES BASED ON 106 DIFFERENT OBJECTS

Number of rings	Min	Max	Mean	Std Dev
n=2	0.0627	121.5968	12.5148	18.4457
n=4	**0.2771**	126.2303	17.9501	21.5741
n=6	0.3615	146.1591	22.7178	25.0350

3) Projection rate p: A $s \times (d(d+1)/2)$ gaussian random matrix $\mathbf{M}^g$ is employed to reduce the dimensionality of the vector $\mathbf{v}_j^m$ derived from the covariance matrix $\mathbf{M}_{\mathbb{G}_j}$ in Eq.(20), where $s = \lfloor d(d+1)/2 \times p \rfloor$, $d = 1 + (n_{max}+1) \times 4 \times n$. Therefore, the projection rate p determines the dimension of the compressed vector $\mathbf{v}_j^{mc}$ and further the hash length. To view effect of projection rate on hash performances and choose an appropriate value for p, receiver operating characteristics (ROC) graph is employed to make visual classification comparisons with respect to robustness and discrimination among different projection rates under the condition $k = 25$. In ROC graph, the x-axis is P_{FPR}, the y-axis is P_{TPR}. And, the ROC curve is formed by a set of points with coordinates (P_{FPR}, P_{TPR}). It is clear that P_{FPR} and P_{TPR} are indicators of robustness and discrimination capability, respectively. For two ROC curves, the curve close to the top-left corner has better classification performances than that far away from the top-left corner.

In the experiment, 40 test engineering CAD graphics described in Section VIII-A are used for testing, i.e., $40 \times 12 = 480$ pairs of identical graphics for robustness validation, and $40 \times (40-1)/2 = 780$ pairs of different graphics for the discrimination test. Table IV presents those projection rates p and thresholds T used for calculating ROC curves. For each pair of graphics, the hash code $\mathbf{h}_j^{'}$ of each group $\mathbb{G}_j^{'}$ of the test graphic $\mathbf{G}^{'}$ is first extracted. Then, group distance $d_{group}(j)$ between $\mathbb{G}_j^{'}$ and $\mathbb{G}_j$ is calculated. Finally, the graphic distance $D_{graphic}$ between $\mathbb{G}^{'}$ and its trusted graphic $\mathbf{G}$ is generated. Different thresholds T are used to find their P_{TPR} and P_{FPR}, and finally obtained the ROC curve for each projection rate p. Fig.6 illustrates the ROC curve comparisons among different projection rates. It is observed that all ROC curves are very

TABLE IV
DIFFERENT PROJECTION RATES AND AUTHENTICATION THRESHOLDS FOR ROC CURVES

Items	Values
Projection rates p	3%, 5%, 8%, 10%, 12%
Authentication thresholds T	$m/30(m = 1, 2, 3, \cdots, 30)$

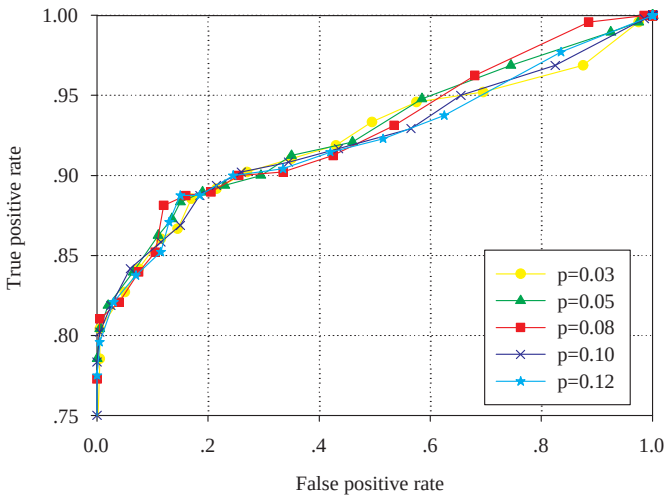
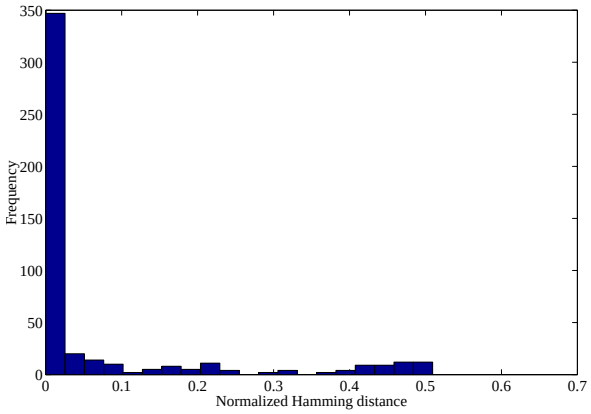


Fig. 6. ROC curve comparisons among different projection rates.

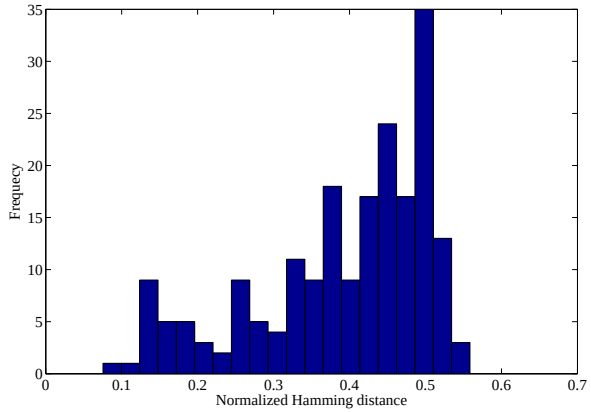
close to the top-left corner. This means that the proposed hashing scheme has satisfactory classification performances with respect to robustness and discrimination. And, it is found that the ROC curve of $p = 0.08$ is little closer to the top-left corner than those of other p values. Therefore, a moderate projection rate, e.g., $p = 0.08$, is a good choice for keeping a desirable trade-off between robustness and discrimination.

4) Authentication threshold T: The threshold T is utilized to measure the similarity between group and graphic pairs respectively. It is clear that the smaller the T value, the better the discriminative capability. However, robustness performance will be hurt as T decreases. Therefore, the threshold T should be chosen in terms of specific applications to give a satisfactory balance between discrimination and robustness.

To determine the threshold T for differentiating two groups, $40 \times 12 = 480$ pairs of identical graphics and $40 \times 5 = 200$ pairs of similar graphics are used. For each pair of graphics, the hash sequence $\mathbf{h}^{'}$ of each test graphic $\mathbf{G}^{'}$ is first extracted. Then, the graphic distance $D_{graphic}$ between $\mathbb{G}^{'}$ and its trusted graphic $\mathbf{G}$ is calculated. Figs. 7(a) and .7(b) show respectively the normalized Hamming distance distributions for hashes of identical graphics and for hashes of similar graphics, respectively. Table V also illustrates the statistics of normalized Hamming distance of identical and similar graphic pairs. It can be observed that the mean distance of identical graphic pairs is only 0.0691 and all maximum distances are less than 0.2, except some rotated graphics. Therefore, 88.13% identical graphics (including some rotated versions) can be correctly detected. And further, all maximum distances will be less than 0.2 if there is no rotated identical graphic. It is



(a) Identical graphic pairs



(b) Similar graphic pairs

Fig. 7. Distribution of normalized Hamming distances of identical and similar graphic pairs

.

TABLE V
STATISTICS OF NORMALIZED HAMMING DISTANCE BASED ON GRAPHIC PAIRS

Graphic pairs	Min	Max	Mean	Std Dev
Identical graphic pairs	0.000	0.509	**0.0691**	0.142
Similar graphic pairs	0.0755	0.558	0.390	0.115
Different graphic pairs	**0.411**	0.585	**0.517**	0.0218

also observed that the mean distance of similar graphic pairs is 0.390 and all minimum distances of similar graphic pairs are larger than 0.2, except some tampered complex graphics with more than 300 objects. And, tampering rates of those graphics just range from about 1% to 5%. Therefore, when $T = 0.2$, 88.13% identical graphics can be correctly detected, and 12.00% similar graphics with low tampering rates are detected by mistake. This is why the authentication threshold we use in the sequel is set to $T = 0.2$.

D. Robustness Analysis

The proposed hashing scheme is designed to be robust to nonmalicious operations including global and local RST transformations. On the premise of keeping the topology among objects unchanged, these manipulations are performed on graphic objects to have a better view or to achieve a satisfactory appearance and fit, as discussed in Section III-A. Therefore, these operations only affect the geometric shape and position of objects.

Test graphics used in Section VIII-C4 are taken and all operations listed in Table I are exploited to attack these graphics. Therefore, each test graphic has 12 functionally consistent graphics and the total number of pairs of identical graphics is $40 \times 12 = 480$. Hash values of the original and the attacked graphics are calculated and then normalized Hamming distance is exploited to evaluate their distance. Fig. 7(a) shows the perceptual robustness of the proposed method considering the above parameterizations. It is observed that 88.13% identical graphics (including some rotated versions) can be correctly detected when $T = 0.2$. And, mean distance of identical graphic pairs is only 0.0691 and all maximum distances are less than 0.2, except some rotated graphics. This means that our hashing scheme can achieve satisfactory robustness performance when $T = 0.2$.

E. Sensitivity Analysis

The sensitivity requires that the proposed hashing scheme is sensitive to malicious operations, including inserting objects, deleting objects, and changing topology relations logically. In terms of objects addition, the added objects should be connected with existing objects. This kind of attack changes the topology of modified objects. In the case of objects removing, it first disconnects the target objects from its joint objects and then deletes them from the graphic. Thus, the topology relation of the involved objects is modified. Modifying local topological relation of objects involves various operations, such as disconnecting two joint objects logically, and connecting two disconnected objects logically. As a result, all of the above operations inevitably bring about the alteration of geometry or topology information of referred objects. Further, they lead to the modification of the covariance matrix of the corresponding group and finally the generated hash codes.

To further validate the sensitivity of the proposed hashing scheme, malicious operations listed in Table II are taken to conduct attacks on each original graphic. Thus, each test graphic has 5 malicious attacked versions and $40 \times 5 = 200$ pairs of similar graphics are used in total. Finally, 200 normalized Hamming distances are calculated as shown in Fig. 7(b) . It is observed that almost all distances are greater than 0.2 except for some graphics where the tampering ratios are less than 5%. This confirms that the proposed method is sensitive to malicious operations.

F. Visual Effect of Tampering Localization

For a content authentication scheme, the tampering localization functionality is of crucial importance. This functionality refers to the capability to identify tampered graphic objects. The proposed hashing scheme is designed to achieve group-level tampering localization capability which can be improved by increasing the group number k as discussed in Section VIII-C1. A graphic is selected and taken to demonstrate the functionality via visual effect. For space limitation, Fig .8(a)

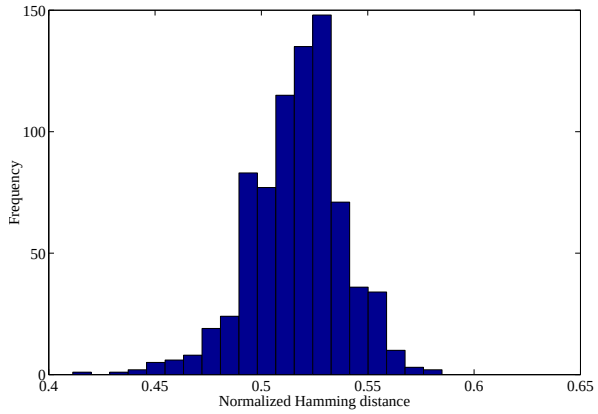


Fig. 9. Distribution of normalized Hamming distances of different graphic pairs.

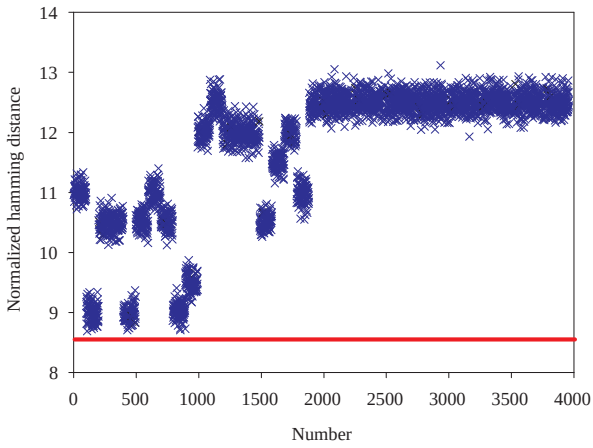


Fig. 10. Distribution of normalized Hamming distances with different keys.

shows part of the test graphic. Three malicious operations discussed above are used to alter graphic objects, respectively. The proposed hashing scheme is applied to the test graphic and it is observed that all normalized Hamming distances are greater than 0.2. All the object groups the tampered objects belong to are correctly identified first. Then, objects in the identified groups are marked as suspicious objects. Figs. 8(b), (c) and (d) show the visual detection results of the proposed method for different attack types. The detected results are highlighted by red color squares, respectively. For example, in Fig. 8(b), a new object A with the same kind of B_1 is added and connected with B_1 and C_1. Thus, the geometry information of the graphic and the topology relation of B_1 and C_1 are modified. Groups those attacked objects belong to are correctly detected by the proposed scheme. And, all the objects (including A, B_1, B_2, B_3, C_1, and C_2) in the detected groups are labelled and highlighted by red color squares.

G. Discriminative Capability Analysis

Discriminative capability means that two different graphics have a very low probability of generating similar hash. If the normalized Hamming distance of two different graphics is less than the threshold T, then the collision occurs.

To evaluate the discriminative capability of the proposed scheme, $40 \times (40-1)/2 = 780$ pairs of different graphics are employed. The proposed hashing scheme is used to extract hashes of 40 different graphics firstly. Then, normalized Hamming distance $D_{graphic}$ between each pair of different graphics is calculated, and $40 \times (40-1)/2 = 780$ results are finally obtained. The statistics of normalized Hamming distance of different graphic pairs are listed in Table .V. Fig. 9 gives the normalized Hamming distance distributions for hashes of different graphics. It is observed from the results that the minimum and mean distance are 0.411 and 0.517 respectively. Clearly, all distances are much bigger than the above mentioned threshold $T = 0.2$, indicating that the proposed hashing scheme has a good discrimination.

TABLE VI
STATISTICS OF NORMALIZED HAMMING DISTANCE USING DIFFERENT KEYS

Min	Max	Mean	Std Dev
8.688	13.117	11.714	1.128

H. Security Analysis

Security depends on the unpredictability of hash codes. This implies that it should be very difficult to decode a hash without knowledge of the key. The security of the proposed hashing scheme can be guaranteed by applying key-dependent encryption in the process of feature vector compression, and randomization. The gaussian random matrix $\mathbf{M}^g$, which is employed to reduce the dimensionality of feature vectors, can be kept as a security key. And, the function seed a and initial value y_0 in the logistic mapping equation (21), which is utilized to encrypt the compressed feature vectors, can also be served as security keys.

To validate the security performance of the proposed hashing scheme, in our experiments, 40 test graphics are adopted. Different keys are exploited to extract hashes, and distances between these key-based hashes are calculated. Only secret keys are varied and other parameters remain unchanged. For each graphic, firstly, a gaussian random matrix $\mathbf{M}^g$, a function seed a and an initial value y_0 are used to extract graphic hash. Then, 99 different gaussian random matrices, 99 different function seeds, and 99 different initial values are employed to generate 99 different graphic hashes, respectively. Finally, normalized Hamming distances between the first hash and other 99 hashes are computed, respectively. Fig. 10 shows the obtained results of all test graphics, where the x-axis is the index of the key and the y-axis is the normalized Hamming distance. Table VI also gives the statics of the calculated distances. It is observed that the minimum distance is much bigger than $T = 0.2$. These results empirically validate that our graphic hashing scheme is key-dependent and meets the security requirements.

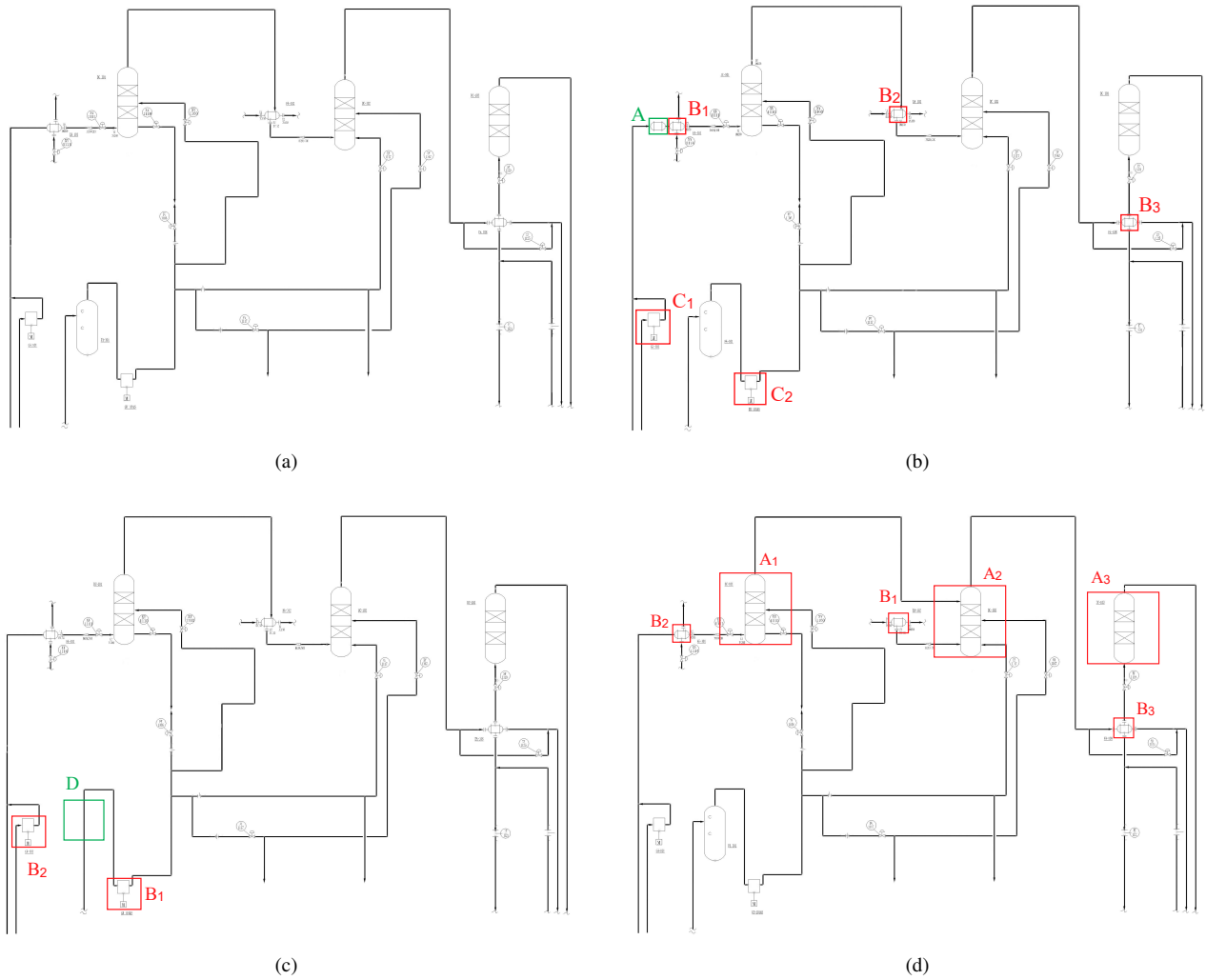


Fig. 8. Tampered graphics and localized objects. (a) Part of the test graphic. (b) An object A is added and connected with B_1 and C_1. (c) An object D is deleted from the graphic. (d) Topology relation among A_1, B_1, and A_2 is modified logically.

IX. Conclusions

In this paper, a novel robust hashing scheme is proposed for jointly authenticating topology and geometry information of 2D engineering CAD graphics . A new covariance-based descriptor is introduced to fuse multiple heterogeneous topology and geometry features. Hashes produced with the proposed method are robust to nonmalicious operations and are sensitive to changes caused by malicious attacks. The hashing scheme described in this paper yields group-level tampering detection and localization ability. The hash can be used to differentiate similar and different graphics. At the same time, it can also identify and locate object groups containing maliciously attacked objects. The proposed scheme achieves a trade-off among robustness, sensitivity, discriminative capability, and tampering localization. The experimental results show the effectiveness and availability of the proposed hashing algorithm.

Further study is desired to find geometry features that better represent various kinds of geometric objects so as to enhance the hash's robustness against the rotation operation. Another important aspect to consider is that achieving a more precise tampering localization accuracy while maintaining short hash length and good sensitivity to malicious attacks.

Acknowledgment

The authors would like to acknowledge the helpful comments and kindly suggestions provided by anonymous referees.

References

[1] W. Shen, Q. Hao, H. Mak, J. Neelamkavil, H. Xie, J. Dickinson, R. Thomas, A. Pardasani, and H. Xue, "Systems integration and collaboration in architecture, engineering, construction, and facilities management: A review," *Advanced Engineering Informatics*, vol. 24, no. 2, pp. 196–207, 2010.

[2] A. Burdorf, B. Kampczyk, M. Lederhose, and H. S. Traub, "CAPD-computer-aided plant design," *Computers & Chemical Engineering*, vol. 28, no. 1-2, pp. 73–81, 2004.

[3] R. Guirardello and R. Swaney, "Optimization of process plant layout with pipe routing," *Computers & Chemical Engineering*, vol. 30, no. 1, pp. 99–114, 2005.

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

[5] C. P. Yan, C. M. Pun, and X. C. Yuan, "Multi-scale image hashing using adaptive local feature extraction for robust tampering detection," *Signal Processing*, vol. 121, pp. 1–16, 2016.
[6] Y. Zhao, S. Wang, X. Zhang, and H. Yao, "Robust hashing for image authentication using zernike moments and local features," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 55–63, 2013.
[7] X. Wang, K. Pang, X. Zhou, Y. Zhou, L. Li, and J. Xue, "A visual model-based perceptual image hash for content authentication," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1336–1349, 2015.
[8] Z. Tang, X. Zhang, X. Li, and S. Zhang, "Robust image hashing with ring partition and invariant vector distance," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 200–214, 2016.
[9] R. Ohbuchi and H. Masuda, "Managing CAD data as a multimedia data type using digital watermarking," in *Proceedings of the IFIP TC5 WG5.2 Fourth Workshop on Knowledge Intensive CAD to Knowledge Intensive Engineering*, Deventer, The Netherlands, The Netherlands, 2001, pp. 103–116.
[10] S. Lee and K. Kwon, "CAD drawing watermarking scheme," *Digital Signal Processing*, vol. 20, no. 5, pp. 1379–1399, 2010.
[11] L. Cao, C. Men, and R. Ji, "Nonlinear scrambling-based reversible watermarking for 2d-vector maps," *The Visual Computer*, vol. 29, no. 3, pp. 231–237, 2013.
[12] F. Peng, Y. Liu, and M. Long, "Reversible watermarking for 2D CAD engineering graphics based on improved histogram shifting," *Computer-Aided Design*, vol. 49, no. 4, pp. 42–50, 2014.
[13] S. H. Lee, S. G. KWON, and K. R. Kwon, "Robust hashing of vector data using generalized curvatures of polyline," *IEICE Transactions on Information and Systems*, vol. E96.D, no. 5, pp. 1105–1114, 2013.
[14] S. H. Lee, W. J. Hwang, and K. R. Kwon, "Polyline curvatures based robust vector data hashing," *Multimedia Tools and Applications*, vol. 73, no. 3, pp. 1913–1942, 2014.
[15] Z. Su, X. Yang, G. Liu, W. Li, and W. Tang, "Topology authentication for piping isometric drawing," *Computer-Aided Design*, vol. 66, no. 9, pp. 33–44, 2015.
[16] Z. Tang, X. Zhang, L. Huang, and Y. Dai, "Robust image hashing using ring-based entropies," *Signal Processing*, vol. 93, no. 7, pp. 2061–2069, 2013.
[17] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Proc. European Conference on Computer Vision, Part II)*, Graz, Austria, May 2006, pp. 589–600.
[18] H. Tabia and H. Laga, "Covariance-based descriptors for efficient 3d shape matching, retrieval, and classification," *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1591–1603, 2015.
[19] K. Wang, G. Lavoue, F. Denis, and A. Baskurt, "A comprehensive survey on three-dimensional mesh watermarking," *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1513–1527, 2008.
[20] A. Khan, A. Siddiqa, S. Munib, and S. A. Malik, "A recent survey of reversible watermarking techniques," *Information Sciences*, vol. 279, no. 20, pp. 251–272, 2014.
[21] C. Fornaro and A. Sanna, "Public key watermarking for authentication of csg models," *Computer-Aided Design*, vol. 32, no. 12, pp. 727–735, 2000.
[22] K. Tarmissi and A. B. Hamza, "Information-theoretic hashing of 3d objects using spectral graph theory," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9409–9414, 2009.
[23] R. M. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, 1984.
[24] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transaction on Communication*, vol. 28, no. 1, pp. 84–95, 1980.
[25] P. Cirujeda, Y. D. Cid, X. Mateo, and X. Binefa, "A 3d scene registration method via covariance descriptors and an evolutionary stable strategy game theory solver," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 306–329, 2015.
[26] D. Shreiner, G. Sellers, J. M. Kessenich, and B. Licea-Kane, *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, 8th ed., D. Shreiner, Ed. Addison-Wesley Professional, 2013.