COMPUTER-AIDED DESIGN & APPLICATIONS, 2016
VOL. 13, NO. 6, 845–854
http://dx.doi.org/10.1080/16864360.2016.1168232

Watermarking scheme for geometric data protection and detection on 3D CAD assembly model

K. M. Yu[a], K. M. Au[b] and W. K. Chiu[a]

[a]The Hong Kong Polytechnic University; [b]HKU SPACE Community College

ABSTRACT

Nowadays, different contemporary products are modeled in a CAD environment. These involve a number of components and assembled together to form an assembly model. Such related components of a product are assembled by different mating conditions in a product design and development stage. Normally, the assembly model would be distributed to different staffs or personnel involved in the process. To ensure that the assembly model received is an original one and is not altered by any unauthorized persons during transmission and exchange of files on the internet, watermarks should be embedded into the model nowadays. In this paper, a watermarking scheme is proposed specifically for watermarking the assembly model structure and the positions and orientations of the components. In this proposed scheme, unique component ID number is assigned to each component. For each component, a hash value is calculated based on the ID numbers of the components related to it in the assembly model tree structure. By detecting the hash value, any change or update in the tree structure can be detected. To watermark a component's position and orientation, some points on a component are first selected. While keeping the selected points on the corresponding component, watermarks and hash values are embedded into the coordinates of the selected points using specified keys by changing the values of the coordinates slightly. The coordinates of the selected points are then stored in the attribute properties of the corresponding component. To detect whether a component's position or orientation are altered, the selected points must be checked and verified whether they are still on the component. In case some of the points are not on the component, the position or orientation of the component is considered as altered. If all the points are still on the components, the watermarks and hash values must be extracted in order to detect whether the watermarks are attacked. Based on the proposed scheme, any unauthorized changes or intended attacks on the watermarked information of an assembly model can be detected. The geometric data and information of the components for the original 3D assembly model can be protected.

KEYWORDS
CAD; Assembly model; watermarking; hash value

1. Introduction

Product design processes are essential activities in a product-based corporation and the processes must be properly controlled so that the success of the products in the market and the growth of the corporation can be ensured. Normally, a product consists of a number of components and a 3D CAD assembly model is used to represent the product. Nowadays, different personnel involved in a design team of a product design process may disperse in different geographical sites. In most of the cases, the CAD files (components or assembly) would be shared and changed by them. As a result, the assembly model of a product should thus be frequently transferred between different personnel through internet. With the explosive growth of the information technology and designing tools, numerous software tools, such as CAD systems [23], product data management (PDM) systems [18], distributed product data management (DPDM) systems, etc. are available and they can be used for supporting the product design and development activities and handling the product data. Normally, numerous modifications on the CAD models of the components and the assembly model of the product being designed will be made. In this case, different versions of the product data will be created and stored in a DPDM or PDM system. One of the main concerns of these systems is the data transferred between different sites must be precise and up-to-date so that different personnel would work on the same set of CAD models. In case different versions of data are distributed to different personnel, the work done based on wrong version of data would be useless and wasted. Extra time may be

CONTACT K. M. Yu km.yu@polyu.edu.hk

required for identifying these mistakes and correcting the newly introduced data. To prevent the presence of this issue, in most of the commercial PDM and DPDM systems, specified functions are introduced for checking the version of the product data. Functions are also available for the users to mark any changes in a CAD model so that other personnel who view the model can be aware of the changes. However, in case unattended changes or unauthorized modifications are accidently made without notifying others, problem will arise and additional cost will be induced for identifying this unauthorized modification. In this case, the requirement of maintaining correct data in the system would become an important issue. In this case, other methodologies should be provided for easily identifying any unauthorized changes on a CAD assembly model.

Nowadays, to protect the copyright of an object or verify whether an object in the electronic formats is altered, digital watermarking technologies are developed. Different watermarking schemes have been developed and are widely used for authenticating of different types of electronic entities, such as computer images, video streams and document etc. [1], [16], [21]. Apart from these formats, 3D geometrical models are required to be watermarked in their applications. Numerous watermarking schemes for watermarking 3D geometrical models can be found in different literature and they can be specifically applied on 3D polygonal models or CSG models. Even many research works have been done on the topic of watermarking 3D models, watermarking schemes for 3D models represented in other schemes such as boundary representation (B-rep) model, feature-based model and 3D CAD assembly model, are not common. To cope with this problem, a new watermarking scheme for 3D assembly models should be introduced. In most of the watermarking schemes for 3D polygonal models, the coordinates of the model's vertices are changed by a small amount so that watermarks are embedded into the model. Proper watermark extraction process is adopted in order to extract watermarks from the model. The extracted watermarks are checked and it can be determined whether any changes are made on the watermarked model. One of the drawbacks of these watermark schemes is the geometrical data of a polygonal model to be watermarked are needed to be slighted modified in order to embed watermarks. This may not be an issue for 3D polygonal models. However, for an engineering model such as a CAD assembly model, any changes on its geometrical data are generally not allowed. In this case, watermarks should be added into the model in another way so that the original geometrical data of the assembly model would not be altered by the introduction of the watermarks.

In this paper, a watermarking scheme for an assembly model is introduced. In this proposed scheme, to watermark the positions and orientations of all the components of the model, proper points on a component are selected and modified and the coordinates of these watermarked points are added to the attributes of the corresponding component. By extracting the watermarks using the keys of the watermarking scheme, any changes in the components' positions or orientations can be detected.

2. Literature reviews

In recent years, different watermarking schemes have been proposed for watermarking 3D geometrical models. Most of the schemes focus on watermarking 3D polygonal models and only some focus on solid geometrical model such as CSG or boundary representation. In [3], a public watermarking scheme was proposed for authenticating CSG models. In this scheme, a new kind of node for the CGS model is introduced and it contains the watermark information. This new node is linked to the original CSG tree by a union operator. To achieve an invisible watermarking, null volume objects are employed in this new node kind. Using this proposed scheme, a CSG model can be watermarked. Watermarking schemes have also been proposed for parametric surfaces. For instances, Sun et al. [19] developed a scheme for non-uniform B-spline surface (NUBS). In their scheme, some points on a NUBS surface are sampled and the watermark is embedded into the DCT coefficients of the sample points. A watermarked surface is then reconstructed from the watermarked sample points. Ohbuchi et al. [11,12] discussed the problem of watermarking non-uniform rational B-spline (NURBS) surfaces. In one approach, a rational linear re-parameterization method is employed while in another approach, knots are inserted into the original surfaces for embedding watermark. In these approaches, the shape of the NURBS surfaces can be preserved. Other schemes for watermarking parametric surfaces can also be found in [8–10]. On the other hand, a multi-resolution system was proposed by Kim et al. [6] and applied in similarity comparisons for geometry-based or feature-based modes to determine the comparison result in the overall shape and to solve security problems in collaborative design conceptually. However, the watermarking issue about assembly models is rarely discussed.

In [13], the concept of embedding data into 3D polygonal models was discussed. Ohbuchi et al. proposed that the geometry information and the topology information can be used for watermark embedding and they argued that geometry is the most suitable data type in a

3D geometrical model for embedding. Numerous watermarking schemes that use geometry data type for watermark embedding were introduced. Chou et al [1] focused on the issue of watermarking 3D polygonal models. In their proposed scheme, some vertices are marked and their positions are slightly perturbed in order to embed watermarks into these marked vertices. Normally, the three coordinates of a vertex are used to store different information: the first coordinate is used to indicate whether a vertex is marked, the second coordinate is used for embedding watermark and the third coordinate is used for embedding a hash value and this hash value is used for checking whether the model is modified. To account for the convergence and causality problems, a multi-function vertex embedding method and an adjusting-vertex method are introduced in their scheme. Wang et al. [20] proposed a watermarking scheme for authenticating 3D polygonal model based on integral invariants. In their scheme, integral invariants of the vertices to be watermarked are first calculated. These values are changed slightly for embedding watermark and so do the positions of the vertices and their neighbours. The watermark embedded in this way can survive under rigid transforms and certain noise attacks. Apart from the vertices coordinates, other geometrical data can also be employed for watermark embedding. In [7], the vertices are clustered to a number of patches and the normal vector distribution of each patch is calculated. The watermarks are then embedded into selected normal vectors. In this watermarking scheme, in order to change normal vectors, the positions of vertices would also be altered.

Some watermarking schemes embed watermark into a 3D model depending on some form of transformation. For instances, Kanai et al. [4] proposed a transformed-domain watermarking approach on 3D polygonal model. In their approach, a model is decomposed using lazy wavelets induced on the polygonal meshes. The wavelet coefficients are modified in order to embed a watermark. In [2], an informed-detection, robust mesh-watermarking algorithm that works in a transformed domain was introduced. In this scheme, to embed information in the "low frequency" component of the shape of the mesh, the shape is modified by using a spatial kernel. Other robust mesh-watermarking algorithms can be found in [5], [14], [17,18], [22].

From the mentioned literatures, different watermarking schemes have been proposed. However, the watermarking issue about assembly models is seldom discussed. Even the watermarking technique for CSG models may be applicable for assembly models with some modifications; the details have not been discussed. To cope with this problem, a corresponding watermarking scheme for assembly models is discussed in this paper.

3. Watermarking scheme for assembly model

To watermark an assembly model, watermark information should be added. As mentioned in [3], the watermark information may be embedded into the original geometrical data of the model or added into the attributes of the components. As the original data of an assembly model cannot be altered, in the proposed watermarking scheme, the latter approach is employed. The detailed procedures for generating watermark information are described below.

3.1. Assembly model data

An assembly model should consist of a number of sub-assemblies and components. Each sub-assembly would in turn consist of a number of components. Apart from the structural relationship between the assembly model, its sub-assemblies and components, other relationships (such as kinematic relationship) may also exist between components. Other data, such as the mating conditions must be set and other attribute properties of the components may be added. In the proposed watermarking scheme, the following information are proposed to be watermarked.

3.1.1. Assembly model structure

Normally a tree structure is employed for representing an assembly hierarchy so that the structure of the assembly model and the relationship between the sub-assemblies and the components can be illustrated. To identify a component, a unique component ID number is assigned to each component. An assembly model structure is altered if new components are added; some components are removed or replaced by other components. To check whether these changes happen, watermark information for each component should be generated based on the assembly model structure.

3.1.2. Position and orientation of each component

To anchor the positions and orientations of the components, proper mating conditions between components should be set. When a mating condition is altered, the positions or orientations of the related components may change. Generally, there are different types of mating conditions and these mating conditions are usually described qualitatively (for example, a surface of a component is coincidence to a surface of another component, etc.). To generate watermark for the position and orientation of each component, instead of the mating condition descriptions, some geometrical data related to a component are extracted and a watermark is embedded into these data.

3.2. Tree of assembly model tree structure

An assembly model structure is represented by a tree and an assembly model tree structure consists of N node. In this tree structure, there are three types of nodes (N_i) (Fig. 1):

(1) Root node – consists of the assembly model. It is a parent node and has a number of child nodes.
(2) Interior nodes – consist of sub-assemblies. An interior node has a number of child nodes and it is also a child node of other interior node (or root node).
(3) Leaf nodes – consist of components. A leaf node has no child nodes and is a child node of an interior node or the root node.

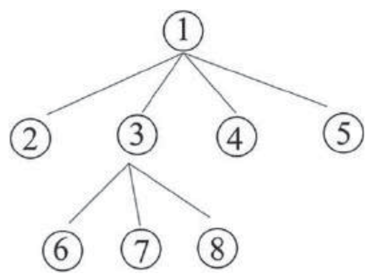

Figure 1. An assembly model tree structure. Node 1 is the root node, node 3 is an interior node and other nodes are the leaf nodes.

The nodes with the same parent are arranged in an ascending order according to their ID numbers. For a root node or an interior node with ID number ID_i, it consists of Nn_i child nodes. A check vector V_{cni} is then estimated for these types of node such that, e.g. Eqn. (3.1).

$$V_{cni} = (Nn_i, ID_p, \ ID_q) \tag{3.1}$$

where ID_p and ID_q are the ID numbers of the node just smaller and greater than ID_i.

In case node i is the first (or last) node in the group of node with the same parent, ID_p would equal to ID number of the last (or first) node in the same group. If there is only one element in the group, the two elements of the check vector of element i are the ID number of node i. For example, the check vector of node 3 is, e.g. Eqn. (3.2).

$$V_{cn3} = (3, \ ID_2, \ ID_4) \tag{3.2}$$

For a leaf node with ID number ID_k, a neighbour-nodes vector V_{nnk} is assigned, e.g. Eqn. (3.3).

$$V_{nnk} = (ID_r, \ ID_s) \tag{3.3}$$

where ID_r and ID_s are the ID numbers just smaller and greater than ID_k. The similar criteria mentioned above will be adopted if node k is the first or last node in the group of node with the same parent. As an example, the neighbour-nodes vector for node 8 is, e.g. Eqn. (3.4).

$$V_{nn8} = (ID_6, ID_7) \tag{3.4}$$

These check vector and neighbour-nodes vectors of all the nodes of the assembly model structure tree will then be used for generating watermark information of the tree structure.

3.3. Geometric data for position and orientation modification checking

The position or orientation of a component would be altered when the mating condition of the component changes. To achieve a new mating condition, a set of transformation operations (translation or rotation) would be performed for transforming the component to a new position and orientation. To watermark the position and orientation of a component, proper geometric data related to the component must be extracted and watermarked so that any change in the position and orientation of the component can be detected by extracting the watermark from these geometric data.

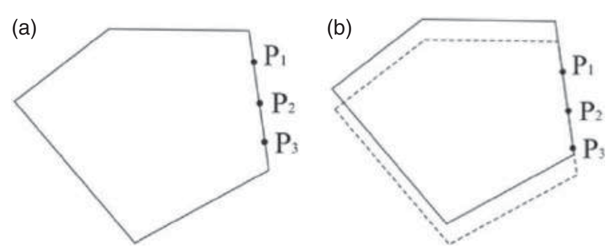

Figure 2. Points on the component: (a) Three points which are collinear are sampled, and (b) The points are still on the component if the component is translated by a proper distance in the direction along the line defined by these sampled points.

For an arbitrary point P, its containment property related to a component – inside, outside or on a component, can be determined. When the component is transformed, the containment property of a point relative to this component may change in some cases. If more than three points on a component are properly sampled (e.g. the points are not collinear) (see Fig. 2), the possibility for the containment properties of all the sampled points remaining unchanged after the corresponding component is transformed would be low, i.e. at least one of these points will be inside or outside the component after the transformation is performed in most of the cases (see Fig. 3). It can be seen that these sampled points can be used for checking whether a component is transformed. In the proposed scheme, for each component, a set of

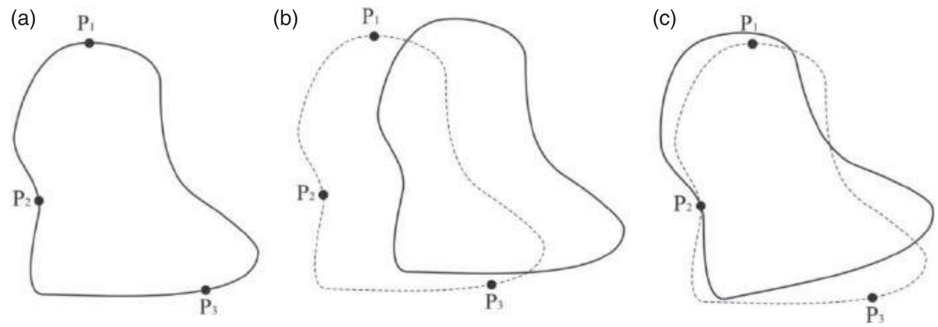

Figure 3. Points on, insides, or outside the boundary: (a) Three points are sampled on the boundary of a component, (b) P1, P2 and P3 are outside after the component is translated, and (c) P2 is still on the component but P1 and P3 are inside and outside after the component is rotated about P2.

properly sampled points on it will be used as the geometric data for embedding data and these watermarked points will be added into the comment of the component and treated as the watermark.

4. Detailed procedures of the proposed scheme

4.1. Embedding watermark into the sample points

Normally, several sampled points on a component can be used to check whether the corresponding component is transformed or not after the points are sampled. However, if the point data are added to the comment of a component without embedding watermarks, these selected points can be easily replaced by other points that are on the component after any malicious change on the assembly model is made. To protect these data, the coordinates of these sampled points should be watermarked while the containment properties of these points are unchanged after watermark is embedded. In the proposed scheme, a hash function $h(w_j)$ and a point containment function $g_i(P_j)$ are established such that, e.g. Eqn. (4.1),

$$h_j = h(w_j) \tag{4.1}$$

where w_j is a watermark and h_j is the hash value for P_j, and Eqn. (4.2) is.

$$g_i(p_j) = \begin{cases} 0 & if \quad P_j \text{ is inside component } i \\ 1 & if \quad P_j \text{ is on component } i \\ 2 & if \quad P_j \text{ is out of component } i \end{cases} \tag{4.2}$$

The approach proposed in [1] for embedding information into point coordinates is adopted here with some modifications. For a component, n_i points on it are sampled and each point P_j is denoted as (x_{j1}, x_{j2}, x_{j3}) (the coordinates are relative to the working coordinate system of the assembly model). A watermark w_j would first be embedded into x_{jk} (k = 1, 2 or 3) so that x_{jk} is modified as X'_{jk}. The hash value h_j is then embedded into another coordinate (x_{jl}) of P_j such that l is not equal to k. A new point P'_j is then resulted. After the watermark and its hash value are embedded into the two coordinates of the selected point, the containment property of the new point is checked, and

1. If $g_i(P'_j) \neq 1$, the remaining coordinate is modified so that the new point P'_j is moved onto the component (see Fig. 4).
2. If $g_j(P'_j) = 1$, the new point P'_j is considered as the watermarked point.

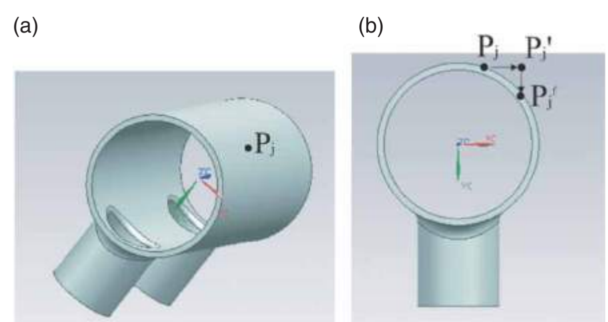

Figure 4. (a) A point P_j on the surface of a component, and (b) The z coordinate of the point is first altered and then the x coordinate is changed. The new point P_j' is not on the surface of the component. By altering the y coordinates of the point, the point can possibly be moved back onto the surface.

As a result, all the watermarked points would fulfil the following criteria:

1. Watermark and the corresponding hash value are embedded to the two coordinates of the watermarked point.
2. The watermarked point is on the corresponding component.

The detailed procedures for embedding the watermark and hash value are as follow:

4.1.1. Select the pair of coordinates of a point to be embedded

For each selected point P_j, the pair of coordinates to be modified by the watermark embedding process is first identified. Normally, for most types of surfaces, the pair would be arbitrary sampled. However, in some cases, if the pair of coordinates is not properly selected, a point P_j after embedding watermark and hash value could not be moved back onto the surface again, such as that shown in Fig. 5. To select a proper pair of coordinates, the normal of the sampled point P_j is checked by following criteria:

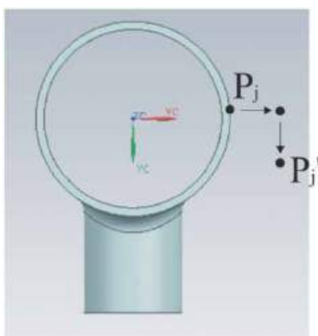

Figure 5. To embed watermark into a point P_j, the x coordinate of P_j is first altered. The y coordinate of P_j is also altered and the point becomes P_j'. It can be seen that P_j' cannot be moved back onto the surface of the component by altering its z coordinate.

1. If the normal of the sampled point is not perpendicular to any one of the principal axes, x_{j1} and x_{j2} are selected.
2. If the normal of the sampled point is perpendicular to one of the principal axes a_r ($r = 1, 2$ or 3), x_{jr} and x_{jl} are selected where l equals to 1, 2 or 3 and l does not equal to r (Fig. 6(a)).
3. If the normal is parallel to one of the principal axes a_r ($r = 1, 2$ or 3), x_{js} and x_{jt} are selected ($s, t \neq r$) (Fig. 6(b)).

4.1.2. Modify the selected coordinates

Selected coordinates x_{jk} of P_j are slighted changed by a value α_k, where $k = 1, 2$ or $1, 3$ or $2, 3$, such that the new coordinates are divisible by a preset quantization number m_k, e.g. Eqns. (4.3) and (4.4).

$$x'_{jk} = x_{jk} + \alpha_k \tag{4.3}$$

and,

$$x'_{jk} \equiv 0 \pmod{m_k} \tag{4.4}$$

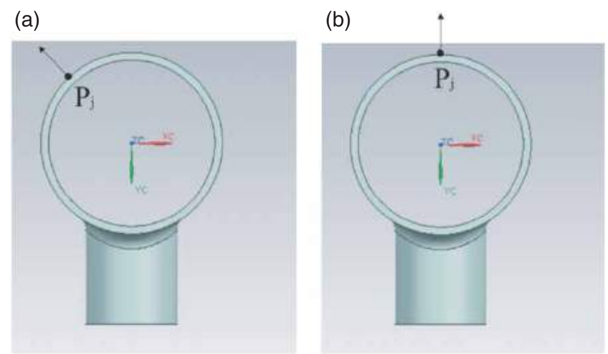

Figure 6. (a) The normal of the point P_j is perpendicular to z axis and so the z coordinate must be selected to be watermarked, and (b) The normal of the point P_j is parallel to y axis and so the x and z coordinates of the point are altered.

4.1.3. Embed watermark and hash value to the pair of selected coordinates

A watermark w_j for P_j and the hash value h_j of this watermark are then embedded into the selected coordinates x'_{jk} ($k = l$ or q), while m_l^d is another preset number ("diameter") for the smallest perturbation sphere enclosing x_{jk} and x'_{jk}, such that Eqns. (4.5) and (4.6).

$$x_{jk}{}^{f} = x_{jk}{}' + (m_l * w_j/m_l{}^d) \tag{4.5}$$

$$x_{jk}{}^{f} = x'_{jk} + (m_q * h_j/m_q^d) \tag{4.6}$$

where $(l,q) = (1, 2)$ or $(1, 3)$ or $(2, 3)$, $q > l$, and x_{jk}^f is the final coordinates.

4.1.4. Displace the point onto the component

After the watermark and hash value are embedded, the sampled point P_j is moved to a new position P'_j. The remaining coordinate x_{jr} is displaced by a small displacement δ_r, $r = 1, 2$, or 3 and $r \neq l, q$, such that the resultant point P_j^f will be on the component, i.e., Eqns. (4.7) and (4.8).

$$x_{jr}^f = x_{jr} + \delta_r \tag{4.7}$$

and,

$$g_i(P_j^f) = 1 \tag{4.8}$$

In case the resultant point cannot move back onto the component, another pair of coordinates is selected and the Sections 4.1.1 to 4.1.4 are repeated for embedding the watermark into the selected point P_j. The parameters m_k and m_k^d ($k = 1, 2$ and 3), the watermark w_j and the hash function h are used as the keys in the watermark extraction process.

4.2. Watermarking the assembly model tree structure

To generate watermark for the assembly model tree structure, another hash function is established. For a component at a leaf node N_k, a neighbour-nodes vector V_{nnk} is obtained. This vector is set as the input to the hash function and a hash value is calculated, e.g. Eqn. (4.9).

$$h_{vk} = h_v(V_{nnk}) \tag{4.9}$$

For a root or an interior node, the check vector V_{cni} is set as the input to the hash function and a hash value is also obtained, e.g. Eqn. (4.10).

$$h_{vi} = h_v(V_{cni}) \tag{4.10}$$

The hash value is then stored in the attributes of the corresponding component. Normally, each component (or sub-assembly) should hold a hash value and this hash value can be used to check whether the components (or sub-assembly) at its neighbour nodes are altered. Based on this criterion, any change in the tree structure of a watermarked assembly model can be identified.

4.3. Verifying the tree structure

To verify an assembly model, the same hash function (h_v) used to watermark the tree structure of the model will be employed. The tree structure of the model will first be established using the approach mentioned in Section 3 above. The hash value at each node is then calculated by the hash function and its neighbour-nodes vector (or check vector). If the assembly model tree structure has not been changed, the hash values calculated in the verification process would be the same as the hash values stored in the attribute of the corresponding node. If one or some hash values are different, the assembly model at the corresponding nodes can be considered as altered. For the root or interior nodes, it simply means that some of their children components are removed or some components are added. For a leaf node, the components next to it in the assembly tree structure may be altered.

4.4. Verifying the positions and orientations of the components

To verify whether the position or orientation of a component is altered, the watermarked points stored in the attributes of each component are first extracted. If these watermarked points are missed, this component can be considered as being attacked or is a newly added component after the assembly model is watermarked. If the watermarked points exist, the watermark and the hash value are extracted from the point coordinates using the keys employed in the watermark embedding process. The procedures are as follow:

4.4.1. Checking the watermarked points

The containment property of each watermarked point relative to the corresponding component is checked. If one of the watermarked points is not on the component, the position or orientation of the component is considered as altered. In this case, the watermark and hash value embedded onto the pair or coordinates of the watermark are not needed to check.

4.4.2. Extracting the watermark and hash value

The watermark and the corresponding hash value can also be extracted by using Eqns. (4.11) and (4.12).

$$w_j = (x_{jl}^f \bmod m_l) * \frac{m_l^d}{m_l} \tag{4.11}$$

$$h_j = (x_{jq}^f \bmod m_q) * \frac{m_q^d}{m_q} \tag{4.12}$$

where $(l, q) = (1, 2)$ or $(1, 3)$ or $(2, 3)$.

For each watermarked point P_j, if it has not be altered accidently, the extracted watermark and the hash value would satisfy the relationship, $h(w_j) = h_j$. Normally, a point on a component can be easily obtained. However, it is almost impossible for a forger to select a point and maintaining the relationship without the keys used in the watermark embedding process. So if this relationship does not hold, it simply implies that the watermarks have been attacked.

In the watermark embedding process, watermark and hash value are embedded into a pair of coordinates of a marked point. However, the pair of coordinates is not specified. In the extraction process, three possible pairs must be checked. If the relationship holds for any one pair, the watermarked point is classified as not attacked; otherwise it is classified as being attacked.

Based on the proposed watermarking methodology, if the position or orientation of a component are altered due to any unexpected change of the mating conditions related to it, these changes can be detected by checking the watermarked points stored at the attribute of the component.

5. Implementation

To verify the process methodology, an example assembly model is tested (see Fig. 7). In this assembly model, there are 11 components. For each component, a hash function of 128 bits is used for watermarking each node of the assembly model structure. As there are 11 leaf nodes

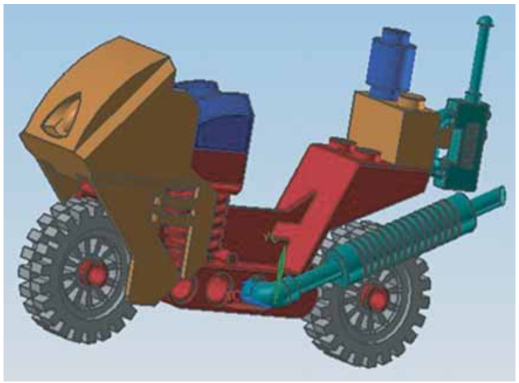

Figure 7. An example assembly mode.

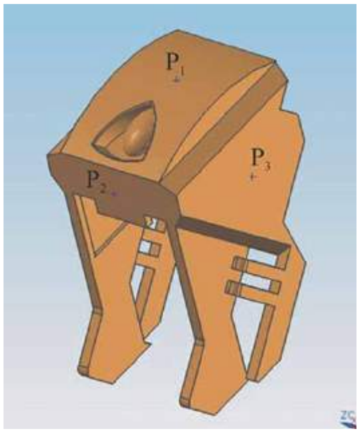

Figure 8. The points on the surfaces of one of the components for embedding the watermark.

Table 1. Coordinates of the points for watermarking.

Point	x_{j1}	x_{j2}	x_{j3}
P_1	31.4943	33.8035	21.4936
P_2	36.9442	18.7554	22.3712
P_3	24.2010	25.3267	11.5729

Table 2. Parameters of the watermarking process.

m_1	0.01
m_2	0.1
m_3	0.001
$m_1{}^d$	25
$m_2{}^d$	40
$m_3{}^d$	40
w	2
h	$h = 2 \times w$

Table 3. The coordinates of the example points in Tab. 1 after the watermark is embedded.

Point	x_{j1}	x_{j2}	x_{j3}
P_1	31.5008	33.7978	21.4941
P_2	36.9508	18.7619	22.3712
P_3	24.2108	25.4100	11.5729

and 1 root node in the assembly model structure, 11×128 bits $= 176$ bytes are used for watermarking the structure of the model.

For each component, 3 points are selected for anchoring the position of the component. In Fig. 8, selected points for one of the components are shown and their coordinates are listed in Tab. 1. The parameters, the watermark and the hash function of the watermarking process are shown in Tab. 2. As the normal at P_1 and P_3 are perpendicular to the z-axis, x_{j1} and x_{j3} of P_1 and P_3 are watermarked ($j = 1$ and 3). The points are then moved back onto the surface by altering the coordinate x_{j2}. As the normal at P_2 is parallel to the z-axis, the coordinates x_{21} and x_{22} are watermarked. The new coordinates of these watermarked points are shown in Tab. 3 and they are still on the surface of the component. Watermarks and hash values are also embedded to the watermarked points of other components and their coordinates are stored in the attributes of the components. Each watermarked point is represented by three double values (3×8 bytes) and so 72 bytes are required for watermarking the position and orientation of a component. For the example assembly model, 11×72 bytes $= 792$ bytes are used in the watermarking process. The overall size of the watermark amounts to 968 bytes. It can be seen that the memory consumed is linear with the number of components in the assembly model. However, it is independent of the file sizes of the assembly model and the components. When comparing with the original file sizes of the assembly model before watermarking (around 2.5MB), the amount of memory required for storing the watermark of the model is small and negligible.

6. Discussion

In this paper, a watermarking scheme for watermarking an assembly model is proposed. In this methodology, some points on each component of the assembly model are extracted. The coordinates of these points are altered in order to add some watermarks onto them while the points are still kept on the corresponding component. These containment properties of the watermarked points (i.e. on the component) are then used as one of the hash functions for checking whether the component's position and orientation are altered. As a result, any unauthorized changes on the components' positions and orientations on the assembly model can be detected by extracting the watermarks of the assembly model.

However, if the position or orientation of the whole assembly model relative to the working coordinate system of the assembly model is altered while the mating conditions of all components are not violated, the model would still be considered as attacked by some unauthorized changes. The proposed method is needed to be modified in order to handle this case.

Apart from altering the positions or orientations of the components, the geometric models of some components may also be attacked. In the proposed method, these cases are not handled. To protect the geometric models of the components, other watermarking methods which can be found in the literature reviews may be adopted for watermarking each component so that any changes in the geometries of the components can be detected. By incorporating with the proposed scheme, an assembly model and the geometries of its components are possibly protected.

In this proposed method, only the positions and orientations of the components and the structure of an assembly model are watermarked. In an assembly model, other information is also contained. The proposed method must be further developed in order to watermark other assembly information. One future work will be to develop the theory for scale-able and mega assemblies.

7. Conclusions

In the proposed method, the watermarks do not alter the geometries of the components and can be applied in 3D assembly models. The watermark information can be embedded into a selected vertex with flexibility. Besides, the original assembly model with watermark information can be accurately and quickly detected and verified when compared with other existing CSG models. The memory sizes of these watermarks are negligible when compared with the file sizes of the assembly model so that the resultant file size of the assembly model would not be largely altered by the proposed method. As a result, an assembly model would now possibly be watermarked by the proposed method without violating the geometric data. Any unauthorized changes or intended attacks on the watermarked information of an assembly model can be detected.

Acknowledgments

The work presented in this paper was supported by a grant from the Hong Kong Polytechnic University (Project No.: G-YH61). Their financial supports have made this research to complete successfully.

ORCID

K. M. Yu http://orcid.org/[0000-0002-5187-9006]
K. M. Au http://orcid.org/[0000-0002-9653-2997]
W. K. Chiu http://orcid.org/[0000-0001-6715-3755]

References

[1] Chou, C.-M.; Tseng, D.-C.: A public fragile watermarking scheme for 3D model authentication, Computer-Aided Design, 38, 2006, 1154–1165. http://dx.doi.org/10.1016/j.cad.2006.06.009

[2] Cotting, D.; Weyrich, T.; Pauly, M.; Gross, M.: Robust watermarking of point-sampled geometry, Proceeding of the International Conference on Shape and Modeling and Application 2004, 233–242, 2004. http://dx.doi.org/10.1109/SMI.2004.1314510

[3] Fornaro, C.; Sanna, A.: Public key watermarking for authentication of CSG models, Computer-Aided Design, 32, 2000, 727–735. http://dx.doi.org/10.1016/S0010-4485(00)00048-8

[4] Kanai S.; Date, H.; Kishinami, T.: Digital watermarking for 3D polygons using multiresolution wavelet decomposition, Proceedings of the Sixth IFIP WG 5.2 International Workshop on Geometric Modeling: Fundamentals and Applications (GEO-6), 296–307, Tokyo, Japan, 1998. http://citeseerx.ist.psu.edu/viewdoc/summary?doi = 10.1.1.16.3250

[5] Kim, M.-S.; Cho, J.-W.; Jung, H.-Y.; Prost, R., A robust blind water marking for 3d meshes using distribution of scale coefficients in irregular wavelet analysis, Proceedings of Acoustics, Speech and Signal Processing, ICASSP 2006, 5, 14–19 May 2006. http://dx.doi.org/10.1109/ICASSP.2006.1661316

[6] Kim, S.; Lee, K.; Hong, T.; Jung, M.: Modeling in multiresolution and its applications, Journal of Computer Science and Technology, 21(2), 272–278. http://link.springer.com/article/10.1007/s11390-006-0272-9

[7] Kwon, K.-R.; Kwon, S.-G.; Lee, S.-H.; Kim, T.-S.; Lee, K.-I.: Watermarking for 3D polygonal meshes using normal vector distributions of each patch, Proceedings of 2003 International Conference on Image Processing, 3, 2003, 499–502. http://dx.doi.org/10.1109/ICIP.2003.1246726

[8] Lee, J.; Cho, N.; Lee, S.: Watermarking algorithms for 3D NURBS graphic data, EURASIP Journal on Applied Signal Processing, 14, 2004, 2142–2152. http://dx.doi.org/10.1155/S1110865704407148

[9] Lee, J.; Cho, N.; Nam, J.: Watermarking for 3D NURBS graphic data, IEEE International Workshop on MMSP December 2002, 304–307, 2002. http://dx.doi.org/10.1109/MMSP.2002.1203306

[10] Mitrea, F. P. M.; Zaharia, T.: Spread spectrum robust watermarking for NURBS surfaces, WSEAS Transactions on Communications, 3(2), 2004, 734–710. http://www-artemis.it-sudparis.eu/Publications/library/Mitrea-WSEAS-TransComm2004.pdf

[11] Ohbuchi, R.; Masuda, H.; Aono, M.: A shape-preserving data embedding algorithm for NURBS curves and surfaces, Computer Graphics International 1999 (CGI 99), Canmore, Canada, 1999. http://citeseerx.ist.psu.edu/viewdoc/summary?doi = 10.1.1.41.1797

[12] Ohbuchi, R.; Masuda, H.: Managing CAD data as a multimedia data type using digital watermarking, Proceedings of the Fourth International Workshop on Knowledge Intensive CAD, May 2000, Parma, Italy, 103–115, 2000. http://dx.doi.org/10.1007/978-0-387-35494-1_8

[13] Ohbuchi, R.; Takahashi, S.; Miyazawa, T.; Mukaiyama, A.: Watermarking 3D polygonal meshes in the mesh spectral

domain, Proceedings of the Graphic Interface, 2001. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.8498
[14] Peng, F.; Liu, Y.; Long, M.: Reversible watermarking for 2D CAD engineering graphics based on improved histogram shifting, Computer-Aided Design, 49, 2014, 42–50. http://dx.doi.org/10.1016/j.cad.2013.12.006
[15] Philpotts, M.: An introduction to the concepts, benefits and terminology of product data management, Industrial Management and Data Systems, 96(4), 2006, 11–17. http://dx.doi.org/10.1108/02635579610117467
[16] Pitas, I.; Voyatzis, G.: Protecting digital-image copyrights: a framework, IEEE Computer Graphics and Applications, 19, 1, 1999, 18–24. http://dx.doi.org/10.1109/38.736465
[17] Qiu, J.; Ya, M.-J.-H.; Sheng, Q.: Watermarking on 3D mesh based on spherical wavelet transform, Journal of Zhejiang University - Science A, 5(3), 2004, 251–258. http://www.zju.edu.cn/jzus/2004/0403/040301.pdf
[18] Su, Z.-Y.; Li, W.-Q.; Kong, J.-S.; Dai, Y.-W.; Tang, W.-Q.: Watermarking 3D CAPD models for topology verification, Computer-Aided Design, 45, 2013, 1042–1052. http://dx.doi.org/10.1016/j.cad.2013.04.001
[19] Sun, S.; Pan, Z.; Kim, T.-W.: Blind watermarking of non-uniform B-spline surfaces, International Journal of Image and Graphics, 8(3), 2008, 439–454. http://dx.doi.org/10.1142/S0219467808003179
[20] Wang, Y.-P.; Hu, S. M.: A new watermarking method for 3D model based on integral invariant, IEEE Transactions on Visualization and Computer Graphics, 15(2), 2009, 285–294. http://dx.doi.org/10.1109/TVCG.2008.101
[21] Xie, L.; Arce, G. R.: A class of authentication digital watermarks for secure multimedia communication, IEEE transactions on Image Processing, 10(11), 2001, 1754–1764. http://dx.doi.org/10.1109/83.967402
[22] Yin, K.-K.; Pan, Z.-G.; Shi, J.-Y.: Robust mesh watermarking based on multiresolution processing, Computers and Graphics, 25(3), 2001, 409–420. http://dx.doi.org/10.1016/S0097-8493(01)00065-6
[23] Zeid, I.: CAD/CAM theory and practice, McGraw-Hill, UK, 1991.