

Managing CAD Data as a Multimedia Data Type Using Digital Watermarking

Ryutarou Ohbuchi¹, Hiroshi Masuda²

¹*Yamanashi University, Computer Science Department (ohbuchi@acm.org)*

²*Tokyo University, Artifacts Research Center (masuda@race.u-tokyo.ac.jp)*

Abstract: This paper advocates the use of *digital watermarking* as a part of a scheme to provide security to 3D geometric CAD (GCAD) systems. This paper discusses requirements for and opportunities available to watermark 3D GCAD models in general. The paper then presents approaches to watermark 3D GCAD models defined by using parametric curves and surfaces. In particular, a pair of watermarking algorithms targeting Non-Uniform B-Spline (NURBS) curves and surfaces, one based on reparameterization and the other based on knot insertion, are presented in detail. An important feature of both of the method is that they preserve the exact geometric shape of the target NURBS objects

Key words: Geometric modeling, information security, digital watermarking, parametric surfaces, Non-Uniform B-Spline (NURBS), reparameterization, knot insertion

1. INTRODUCTION

The membership of “multimedia” data type has long been limited to sound, text, still image, and movie data types. Recently, however, three-dimensional (3D) models such as VRML [VRML] and MPEG-4 SNHC [MPEG4] have increasingly been regarded as full-fledged multimedia data objects. A 3D geometric Computer-Aided Design (GCAD) model can also be viewed as a multimedia data object if one considers such scenarios as design documentation, design reuse, collaborative design, concurrent design

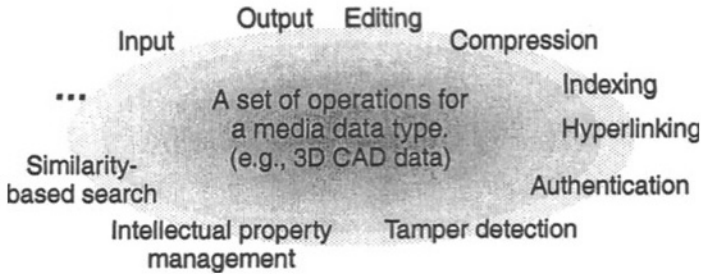


Figure 1. Examples of operations associated with a multimedia data type, including 3D CAD data.

and manufacturing, marketing, and maintenance. These scenarios, all of which may be prefixed with “network-based” or “Internet-based”, demand that we treat a 3D GCAD model as a multimedia data type.

As an abstract data type, each media data type is associated with its own set of operations (Figure 1). A minimal set of operations most likely includes *input*, *output*, and *editing*. In addition, when the data objects are accumulated, shared, distributed, or broadcast, e.g., via the Internet, such operations as *compression*, *indexing*, *hyperlinking*, *similarity-based search*, *authentication*, and *intellectual property (IP) management* gain importance.

In case of the text data type, technologies for most of these operations are fairly well understood and available. Social acceptance of and legal infrastructure necessary for some of these operations, e.g., IP protection, have long been established. For other data types, however, not all of the operations are well understood. For example, in case of still image, technologies for similarity-based search and copy protection are still in the research and development stage. Time-dependency that exists in such media data types as audio and movie adds a degree of difficulty. Editing, hyperlinking, and indexing time-varying data embody many issues to be solved. Simply hyperlinking “from this to that” can be quite difficult if “this” and “that” must be specified both in time and space.

In case of 3D model, even such basic operations as input and output are still the subject of active research and development. In addition, increased incidence of distribution and sharing of 3D models, especially through the Internet, created a whole new set of requirements for 3D models. Operations other than input, output, and editing have suddenly become important. Foremost in the list of newly important operation is the *compression* of 3D geometry and its attributes [Rossignac99]. Hyperlinking and indexing have found utilities in some CAD systems, e.g., to annotate and cross-reference designs and parts for documentation and collaboration. The other operations, for example, authentication, IP management, and similarity-based search have seen little work so far. Yet these operations are vital to net-based almost-everything.

Nowadays it is commonplace for several divisions within a company to collaborate via network over physical distance and time zones. Often, more than one company are involved in a design, exchanging design documents. In the future, a mechanical part may be purchased as a “virtual component”, i.e., as a design, not as a finished product. (In fact, trading a part as a design is already quite common in the world of VLSI chips. They call a design of a component “IP” for intellectual property.) Is the CAD data you just received from your colleagues in a branch office (or a vendor) via the Internet genuine, or is it subtly tampered with so that your project fails to meet the deadline? If you are a vendor of a part, how do you prevent a thief disguised as a customer copying your design you have provided for pre-production “evaluation”?

This paper advocates the use of digital watermarking for 3D GCAD models in order to realize various security related operations for the models. Following a review of previous work on watermarking 3D geometric models, we attempt to classify possible approaches for watermarking GCAD models based on redundancy employed by each class of methods. We then restrict our watermarking target to parametric curves and surfaces, which are the mainstay shape representation for 3D GCAD. We list a number of approaches for watermarking parametric curves and surfaces. In particular, details are described of a pair of algorithms for Non-Uniform B-Spline NURBS curves and surfaces that exactly preserves their geometric shapes, a property necessary for most of the 3D GCAD applications.

1.1 Digital Watermarking

Digital watermarking puts structures called *watermarks* into digital contents (e.g., images) in such a way that the structures do not interfere with the intended uses (e.g., viewing) of the contents, and that the structures are not easily separable from the contents. The watermarks encode information that can be used to manage the contents, in order, for example, to add annotations, to detect tampering, or to authenticate rightful purchasers. A book by Katzenbeisser and Petitcolas [Katzenbeisser00] provides a good starting point to study digital watermarking in general. We will review a few important terminologies and concepts in this section.

The act of adding a watermark is called (data) *embedding* or *watermarking*, and that of retrieving the information encoded in the watermark for perusal is called *extraction* (Figure 2). An object to be watermarked is called *cover- \langle datatype \rangle* , the object with watermark is called *stego- \langle datatype \rangle* . The suffix “ \langle datatype \rangle ” specifies the data type of the content. For example, a message bits are embedded in a cover-polygon mesh surface to create a stego-polygon mesh surface.

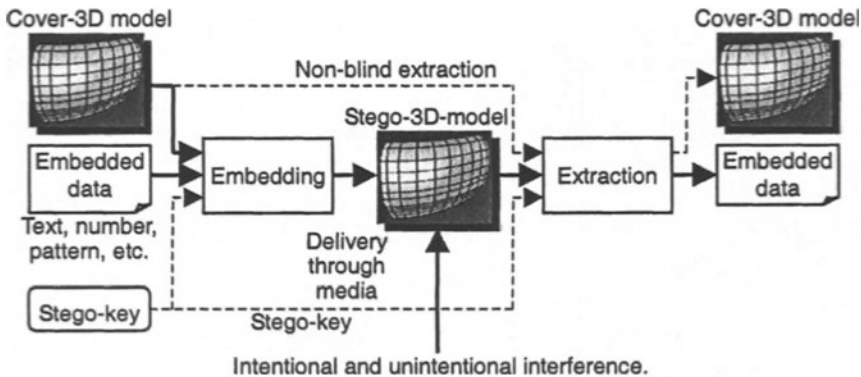


Figure 2. A typical flow of digital watermarking.

Arguably the most important property of a watermark is its *transparency* (sometimes called *invisibility* or *imperceptibility*). Watermarks must be *transparent* to the intended applications. We distinguish two kinds of transparencies, *functional* and *perceptual*. For most of the traditional data types, such as image and audio data, transparency of a watermark is to be judged by human beings. If the cover-data and stego-data are indistinguishable to human observers, the watermark is *perceptually transparent*. For other data types, such as 3D GCAD data, transparency of the watermark is judged by if *functionality* of the data is compromised or not. A perceptually transparent watermark may or may not be functionally transparent. Likewise, a functionally transparent watermark may or may not be perceptually transparent. For example, a perceptually transparent watermark added to CAD data of an engine cylinder may alter the shape of the cylinder enough to interfere with the function of the engine.

A very important classification of watermark is by *robustness*. A *robust watermark* should withstand both intentional and unintentional modifications of the stego-data. On the other hand, a *fragile watermark* must be affected by modifications so that tampering and other damage to the data can be detected. The modification is applied as a part of normal use of the object (e.g., image compression) or as an attack with an intention to remove the watermark. A robust watermark can be used to claim copyright of illegally duplicated data, while a fragile watermark finds application in tamper detection.

Another classification is by the use of cover data for extraction. A *blind-detection* algorithm extract messages using stego-data only, while a *non-blind* detection algorithm requires original cover-data as well as the watermarked stego-data for extraction. While a watermarking scheme with non-blind detection enables more robust and accurate watermark extraction, a blind-detection watermarking algorithm usually has wider applications.

Note that a watermarking technology itself is not sufficient for the watermark to be effective. Legal infrastructure, social acceptance, as well as technical infrastructures such as verifiable digital signature and trusted places to escrow cover-3D models are necessary.

1.2 Previous Work on Watermarking 3D Models

Numerous paper on watermarking “classical” data types, such as text, 2D still image, 2D movie, or audio data, have been published [Katzenbeisser00]. However, relatively small number of research on watermarking 3D models has been published. Most of the watermarking algorithms for 3D geometric models have targeted geometric shapes of 3D polygonal meshes [Ohbuchi97, Kanai98, Ohbuchi98a, Ohbuchi98b, Benedens99, Yeo99, Praun99, Benedens00, Wagner00, Ohbuchi01]. For non-polygonal mesh shape definitions, Ohbuchi et al targeted parametric surfaces [Ohbuchi99] and Fornaro and Sanna targeted Constructive Solid Geometry (CSG) models [Fornaro00].

These existing watermarking algorithms that target 3D shapes are not suitable, however, for watermarking shapes found in most GCAD models. There are two reasons for this. First, most of the previous watermarking algorithms for 3D shapes targets shapes defined by using polygonal meshes. They are not readily applicable to most of the GCAD models that use parametric curves and surfaces for shape definition. Second, almost all of the previous watermarking algorithms for 3D shapes alter shape geometry and/or topology for watermarking. However, GCAD models rarely tolerate changes in geometry and/or topology. For most GCAD models, watermarking algorithms that exactly preserves shape are required.

Fornaro and Sanna [Fornaro00] proposed a method to “watermark” a CSG model by employing an approach similar to *message authentication code*, which protects integrity of emails. To authenticate a CSG model, their method compute a hash value from the CSG model and encode it into a special invisible CSG node, for example, a null-sized sphere, attached to the CSG model. Such special nodes, however, can easily be found and removed. For this reason, this method may not qualify as a “watermarking” scheme, although it would be useful for model integrity verification.

2. WATERMARKING 3D GEOMETRIC CAD MODELS

Watermarking in general exploits redundancy in the watermarking target to encode extra information that is the watermark. There are three classes of

redundancies that can be exploited to add structures to 3D GCAD data for watermarking.

- **Innate redundancy:** A shape itself may have redundancy. Information may be encoded in the shape by altering parts of the shape without interfering with the function of the shape. A shape of a machine part has a set of functions. The shape itself often has certain amount of arbitrariness that can be exploited to encode watermarks without affecting intended function of the shape. A trivial example is an engraving of a part number found on a component.
- **Representation redundancy:** Representation of a shape may have redundancy. Information may be encoded for watermarking by modifying the representation while exactly preserving the shape. For example, knots can be inserted into a NURBS surface without changing its geometric shape. Once inserted, knots are very difficult to remove if exact preservation of the model's geometric shape is enforced.
- **Encoding redundancy:** Encoding of a shape representation has redundancy so that watermark information can be encoded by modifying the encoding. Geometric shape and its representation do not change. A trivial example would be to use the lower 4 decimal digits of 10 digit floating number format to encode information, assuming that the CAD system has an effective accuracy of at most 6 digits.

3. WATERMARKING PARAMETRIC CURVES AND SURFACES

This section lists methods to watermark parametric curves and surfaces. These methods are classified by their two properties, preservation of shape and data size. A *shape-preserving* method preserves the exact shape of the original model. Likewise, a *size-preserving* method preserves the original data size, that is the numbers of shape-defining parameters such as control points and knots.

Note that, while most of the descriptions in the following use parametric curves as examples, the same descriptions apply with little modifications to tensor-product surfaces created by using the parametric curves.

3.1 A Shape-Preserving, Size-Preserving Method

Rational-linear reparameterization is the only one method we came up so far in this category, which preserves both shape and data size. A watermarking method based on this property was first presented in [Ohbuchi99].

3.1.1 A Rational-Linear Reparameterization Based Algorithm

If is reparameterized by using a *rational linear function*, a NURBS curve preserves its geometric shape. The curve also preserves data size, measured in terms of number of knots, control points, and weights. We utilized this property to create a shape preserving watermarking algorithm. The scheme employs non-blind detection (i.e., the extraction requires both cover- and stego-NURBS models). The watermark embedded using this method is not robust; another reparameterization will destroy the watermark. Also note that the method does change shapes of objects that rely on parameterization, e.g., a ruled surface.

Let us first define a NURBS curve. (For the details on NURBS curves and surfaces, see, for example, Piegl and Tiller [Piegl97].) A p th-degree NURBS curve $C(u)$ defines a point that traces a trajectory in 3D space as the scalar parameter value u varies in the interval $[a,b]$. A set of control points $\{P_i\}$ forms a control polygon, and $\{w_i\}$ are the weights of the control points. U is a non-periodic and non-uniform knot vector, which is a nondecreasing sequence of real numbers, that is, $a \leq u_i \leq u_{i+1} \leq b$ for $i = 0, \dots, m-1$. Recursively defined function $N_{i,p}(u)$ is the i th B-spline basis function of degree p (order $p+1$).

$$C^w(u) = \sum_{i=0}^n N_{i,p}(u) P_i^w \quad (1)$$

$$U = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1}\} \quad (2)$$

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \quad (3)$$

A NURBS curve $C(u)$ defined on u can be reparameterized by a function $u = f(s)$ so that the curve is computed as a function of a new parameter s . We require that the function $f(s)$ be increasing ($f'(s) > 0$ for all $s \in [c,d]$ in which $a = f(c)$ and $b = f(d)$) so that the same point x is not traced more than once. As mentioned above, we reparameterize a NURBS curve by using a rational linear function, as studied by Lee and Lucian [Lee91], to preserve the shape and the data size of the curve.

There are four coefficients α , β , γ , and δ in $g(u)$, which is an inverse of $f(s)$, but its degrees-of-freedom (DOF) of is three. This is obvious by rewriting $k_1 = \alpha/\gamma$, $k_2 = \beta/\gamma$ and $k_3 = \delta/\gamma$.

$$s = g(u) = \frac{\alpha u + \beta}{\gamma u + \delta} = \frac{k_1 u + k_2}{u + k_3}, \quad u = f(s) = \frac{-\delta s + \beta}{\gamma s - \alpha} \quad s \in [c, d] \quad (4)$$

To ensure that $g(u)$ and $f(s)$ are well-behaved, we assume that

$$\begin{aligned} \alpha\delta - \gamma\beta &> 0, \\ \mu(u) = \gamma u + \delta &\neq 0 \quad \text{for all } u \in [a, b], \text{ and} \\ \lambda(s) = \gamma s - \alpha &\neq 0 \quad \text{for all } s \in [c, d] \end{aligned} \quad (5)$$

The reparameterized curve $C(s)$ has the same control points $\{P_i\}$. The new knots s_i are the image $s_i = g(u_i)$ of the original knots u_i . The new weights $\{\bar{w}_i\}$ (modulo a common nonzero factor) are obtained from either one of the equations (6) below:

$$\bar{w}_i = w_i \prod_{j=1}^p \lambda(s_{i+j}) \quad \text{or} \quad \bar{w}_i = w_i / \prod_{j=1}^p \mu(u_{i+j}). \quad (6)$$

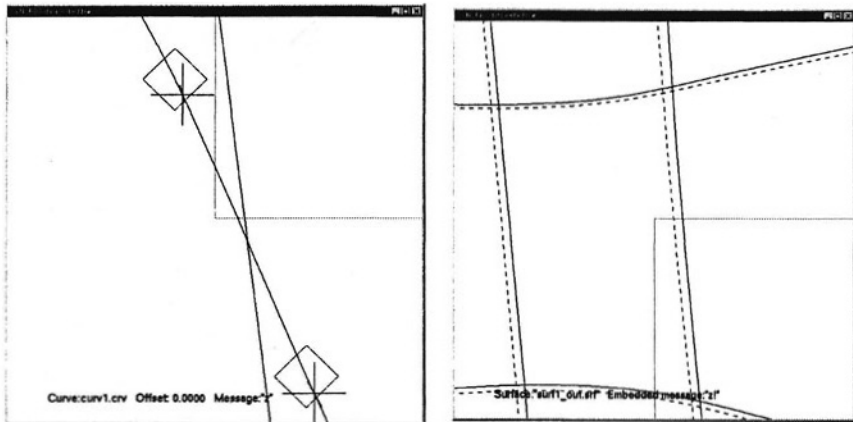
where s_{i+j} and u_{i+j} are the new and old knots, respectively.

The function $g(u)$, which has 3-DOF, can be specified by three points (u_1, s_1) , (u_2, s_2) , and (u_3, s_3) through which the function $g(u)$ must pass using the following formula.

$$\begin{aligned} k_1 &= \frac{(u_1 s_1 - u_2 s_2)(s_1 - s_3) - (u_1 s_1 - u_3 s_3)(s_1 - s_2)}{(u_1 - u_2)(s_1 - s_3) - (u_1 - u_3)(s_1 - s_2)} \\ k_2 &= u_1 s_1 + k_3 s_1 - k_1 u_1 \\ k_3 &= \frac{(u_1 s_1 - u_3 s_3)(u_1 - u_2) - (u_1 s_1 - u_2 s_2)(u_1 - u_3)}{(u_1 - u_3)(s_1 - s_2) - (u_1 - u_2)(s_1 - s_3)} \end{aligned} \quad (7)$$

We fix the two endpoints, that is, $u_1 = s_1 = a$ and $u_3 = s_3 = b$, so that the range of parameter is the same before and after the reparameterization. We encode the watermark information in the one remaining DOF by choosing the offset $D = s_2 - u_2$. The larger the magnitude of D , the more the curve of the function $s = g(u)$ deviates from the straight line $s = u$. We keep the magnitude of D small so that the curve after reparameterization has a ‘‘good’’ parameterisation.

For the proof of concept system, we experimented with a simple amplitude modulation to encode a number into the offset D . Given the message data d of size L bits and the predefined range of the offset $[D_{\min}, D_{\max}]$, the offset D is computed by using the following formula $D = (D_{\max} - D_{\min})(d + 0.5)/2^L + D_{\min}$. This D offsets an *interior* knot of the NURBS curve. We used the knot in the middle of the knot sequence whose index i in the knot sequence is given by $i = \lfloor m/2 \rfloor$.



(a) A letter "z" is embedded in a NURBS curve. Cross and rectangular markers are at the same parameter values but plotted before and after the watermarking.

(b) A 2-byte string "aZ" was embedded in a Cartesian product NURBS surface. Solid and dotted lines are iso-parametric lines before and after the watermarking.

Figure 4. NURBS curves and surfaces are watermarked by using reparameterization, which exactly preserves their geometric shapes.

Extraction of a watermark starts with a comparison of the value of the i th knots in the knot vectors of the cover-model (original model) and the stego-model (watermarked model) to find the offset $D = s_2 - u_2$. From this value, the original data d can be recovered by using the formula to compute the value D in reverse.

Figure 4 shows examples of embedding data into a NURBS curve and a NURBS surface using the method described above. Figure 4a shows a NURBS curve reparameterized to encode 8 bit of information. Figure 4b is a close-up of a NURBS surface to show a small shift in the iso-parametric lines caused by embedding 16 bit of information.

3.2 Shape-Preserving, Size-Altering Methods

A watermarking algorithm that exactly preserves the model's shape but alters its data size can be realized by injecting redundancy into

representations of parametric curves and surfaces. Such redundancy injection can be achieved by using such techniques as knot insertion, degree elevation, and reparameterization involving degree elevation.

- **Knot Insertion:** For those parametric curves with multiple spans, such as nonrational B-spline curves and NURBS curves, new knots can be inserted into the curve. The values of the inserted knot, the mere presence of the new knot, or the location of the new knot in the knots vector may encode information to be embedded.
- **Degree Elevation:** For nonrational curves, such as Bezier and B-spline curves, elevating the degree of a curve introduces new control points. For example, the amount of the increase in degree or the locations of new control points may encode information.
- **Reparameterization Involving Degree Elevation:** Reparameterization by using a polynomial or rational polynomial of a degree greater than one can be applied to rational parametric curves, such as rational Bezier and NURBS curves. Such reparameterization will raise the degree of the curve, introducing new control points and knots, and increasing the data size.

3.2.1 A Knot Insertion Based Algorithm

Knot insertion into NURBS curves and surfaces can be used for a robust watermarking algorithm that exactly preserves shape. Watermarks added by the method are quite robust against attempted removal assuming that the intended application of the watermark demands exact shape preservation. This is because knots removal applied in an attempt to remove watermark will inevitably involves minute shape change, which is not tolerated by the application. In addition to shape preservation and robustness, this method offer significantly larger data capacity than the reparameterization based algorithm described in Section 3.1.1, since multiple knots may be inserted into a NURBS curve.

Given a NURBS curve

$$\mathbf{C}^w(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i^w \quad (8)$$

defined on a knot vector $U = \{u_0, \dots, u_m\}$, a knot insertion adds a new knot to the knot vector so that geometric shape of the curve remains unchanged. Let a new knot value that is added to encode watermark message be $\bar{u} \in [u_k, u_{k+1})$. We insert the knot \bar{u} into U to form the new knot vector \bar{U} ;

$$\bar{U} = \{\bar{u}_0 = u_0, \dots, \bar{u}_k = u_k, \bar{u}_{k+1} = \bar{u}, \bar{u}_{k+2} = u_{k+1}, \dots, \bar{u}_{m+1} = u_m\}. \quad (9)$$

After the knot insertion, $C^w(u)$ has the representation of the form;

$$C^w(u) = \sum_{i=0}^{n+1} \bar{N}_{i,p}(u) Q_i^w \quad (10)$$

where $\{\bar{N}_{i,p}(u)\}$ are the p th-degree basis function on \bar{U} and $\{Q_i^w\}$ is the new set of control points. The curve is the same both geometrically and parametrically before and after the knot insertion. While the new set of control point $\{Q_i^w\}$ is different from $\{P_i^w\}$, the new knot vector \bar{U} includes all the knots in the original knot vector U .

New set of control points $\{Q_i^w\}$ can be computed by the following formula [Piegl97].

$$Q_i^w = \alpha_i P_i^w + (1 - \alpha_i) P_{i-1}^w \quad (11)$$

where

$$\alpha_i = \begin{cases} 1 & i \leq k - p \\ \frac{\bar{u} - u_i}{u_{i+p} - u_i} & k - p + 1 \leq i \leq k \\ 0 & i \geq k + 1 \end{cases} \quad (12)$$

Information is encoded into this watermark by choosing the value of the inserted knot \bar{u} . We call a knot inserted for watermarking a *watermark knot*. Since a NURBS curve can be inserted with more than one knot, data capacity of the watermark per curve can be significant.

The value of watermark knot \bar{u} can take any value in the range $u_0 < \bar{u} < u_m$. In our proof-of-concept implementation, we arbitrarily decided to insert the watermark knot near the middle of a pair of adjacent knots. Deviation from the midpoint is determined by the data value and an amplitude scale parameter γ . Assuming the message data d of size L bits, we compute the watermark knot \bar{u} as

$$\bar{u} = \frac{u_{k+1} - u_k}{2} \left(1 + \gamma \frac{d - 2^{L-1}}{2^L} \right) \quad (13)$$

Here the amplitude scale parameter γ is chosen small enough (e.g., $\gamma \leq 0.01$) so that \bar{u} stays fairly close to the middle.

Extraction of the watermark is non-blind, so that the original cover-NURBS curve is required for extraction. Comparing the knot vector of the original curve with that of the watermarked curve finds inserted watermark

knot(s) and their value(s). Embedded message can be recovered easily from the value(s) of the watermark knot(s).

If multiple watermark knots are added to a NURBS curve, they must be ordered somehow to correctly embed and extract information. (Imagine a 16 bit message, in which each of the four 4 bit fragments is encoded into one of four watermark knots in a knot vector. Without ordering among the fragments, correct extraction of the 16 bit message is not possible.) Ordering of a set of knots on a NURBS curve can be introduced, for example, by sorting the watermark knots by their value. To embed a larger chunk of data, multiple NURBS curves must be ordered. Such ordering can be introduced, for example, by using topology among curves in the CAD model. Extending this watermarking method to a tensor product NURBS surface is trivial.

Figure 5 shows an example of watermarking a NURBS curve by inserting three knots. While the shape of the control polygon changes, that of the NURBS curve itself remains unchanged.

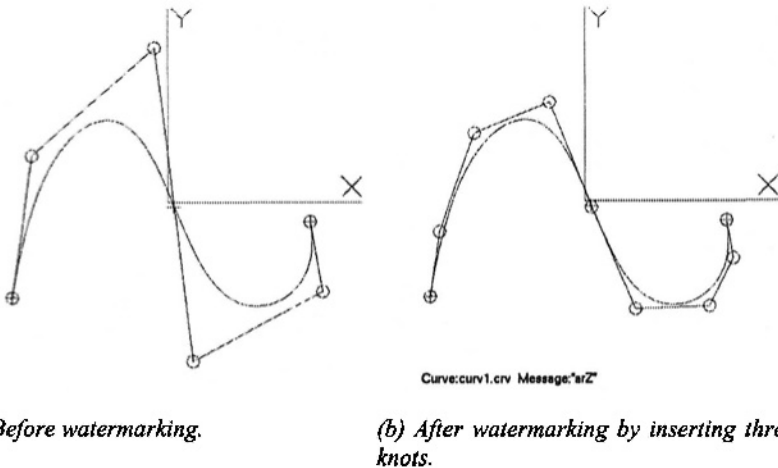


Figure 5. A NURBS curve before (a) and after (b) watermarking by using knot insertion. The value of the inserted knot encodes information to be embedded into the watermark.

3.3 Shape-Altering, Size-Preserving Methods

While tolerating shape change may sound odd for a CAD model, applications exist that do tolerate such shape changes. For example, interference checks performed to evaluate manufacturability, a maintenance manual, or an on-line parts catalogue do not require exact shape preservation. If exact shape preservation is not an issue, additional approaches exist for watermarking parametric curves and surfaces.

- **Control Points and/or Weights Modulation:** The control points for a curve form a one-dimensional sequence of 3D (or 4D if weights are included) points. These values can be modulated for watermarking.
- **Knot Vector Modulation:** The values of knots in a knot vector, which is a 1D sequence of scalar values, can be modified to encode information.

Note that the modulation of numerical values, such as control point coordinates, weights, and knot values may be performed either in their original domain or in a transformed domain. Watermarking in a transformed domain is one of the most popular approaches to watermarking for image data type. As the transformation, Fourier, wavelet, discrete-cosine and other transformations have been employed. Transformed-domain watermarking techniques can be applied to any kind of data that consists of an ordered set of numbers. For example, algorithms described in Section 3.1.1 and Section 3.2.1 could also benefit from this transformed domain approach.

4. CONCLUSION

The paper first examined 3D GCAD model as a multimedia data type, exploring a set of operations that are necessary it to become a full-fledged multimedia data type. The paper then focused on digital watermarking as a possible means to provide such security related operations as intellectual property management, authentication, and tamper detection for 3D GCAD models. The paper listed three avenues to watermark 3D GCAD models classified by the redundancy exploited for watermarking. This paper then focused on approaches to watermark parametric curves and surfaces, which are the mainstay objects for 3D GCAD systems. Details of two watermarking algorithms for NURBS curves and surfaces, one based on knot insertion and the other based on rational linear reparameterization, have been presented. Both algorithms exactly preserve geometric shape of the target NURBS curves and surfaces. The knot insertion based method produces watermarks that are quite robust if exact preservation of the model's geometric shape is required. One avenue of our future work is to apply the algorithms described in this paper in the real world. To do that, data representation formats of GCAD systems and GCAD data exchange standards must be taken into account so that the watermarks survive format conversions. Another avenue of our future work is to further explore the alternative approaches to watermark 3D GCAD models, perhaps employing a different class of redundancy for watermarking.

ACKNOWLEDGEMENTS

A part of this research was conducted while the first author was a senior research staff member at the IBM Tokyo Research Laboratory. The first author is supported by grants from the Ministry of Education, Culture, Sports, Science and Technology (MECSST) of Japan (Grant No. 12680432).

REFERENCES

- [Benedens99] O. Benedens, Geometry-Based Watermarking of 3D Models, *IEEE CG&A*, pp. 46-55, January/February 1999.
- [Benedens00] O. Benedens, C. Busch, Towards Blind Detection of Robust Watermarks in Polygonal Models, Proc. of the *EUROGRAPHICS 2000*, vol. 19 (2000), No. 3.
- [Fornaro00] C. Fornaro, A. Sanna, Public key watermarking for authentication of CSG models, *Computer-Aided Design*, vol. 32 (2000), pp. 727-735.
- [Kanai98] S. Kanai, H. Date, and T. Kishinami, Digital Watermarking for 3D Polygons using Multiresolution Wavelet Decomposition, Proc. of the *Sixth IFIP WG 5.2 International Workshop on Geometric Modelling: Fundamentals and Applications (GEO-6)*, pp. 296-307, Tokyo, Japan, December 1998.
- [Katzenbeisser00] S. Katzenbeisser, F. A. P. Petitcolas, *Digital Watermarking*, Artech House, London, 2000.
- [Kim00] Tae-hoon Kim, Jehee Lee, Sung yong Shin, Robust Motion Watermarking based on Multiresolution Analysis, Proc. of the *EUROGRAPHICS 2000*, vol. 19 (2000), No. 3.
- [MPEG4] ISO/IEC JTC1/SC29/WG11 *MPEG-4 Visual and MPEG 4 SNHC*.
- [Ohbuchi97] R. Ohbuchi, H. Masuda, and M. Aono, Watermarking Three-Dimensional Polygonal Models, *Proc. of the ACM Multimedia '97*, Seattle, Washington, USA, November 1997, pp. 261-272.
- [Ohbuchi98a] R. Ohbuchi, H. Masuda, and M. Aono, Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications, pp. 551-560, *IEEE Journal on Selected Areas in Communications*, May 1998.
- [Ohbuchi98b] R. Ohbuchi, H. Masuda, and M. Aono, Geometrical and Non-geometrical Targets for Data Embedding in Three-Dimensional Polygonal Models, *Computer Communications*, Vol. 21, pp. 1344-1354, Elsevier (1998).
- [Ohbuchi99] R. Ohbuchi, H. Masuda, and M. Aono, A Shape-Preserving Data Embedding Algorithm for NURBS Curves and Surfaces, In Proc. *Computer Graphics International '99 (CGI '99)*, pp. 180-187, Canmore, Canada, June 4-June 11, 1999.
- [Ohbuchi01] Ryutarou Ohbuchi, Shigeo Takahashi, Takahiko Miyazawa, and Akio Mukaiyama, Watermarking 3D Polygonal Meshes in the Mesh Spectral Domain, in Proc. of the *Graphics Interface 2001*, pp. 9-17, Ontario, Canada, June, 2001.
- [Piegl97] L. Piegl, W. Tiller, *The NURBS Book*, 2nd Edition, Springer, Berlin, 1997.
- [VRML] ISO/IEC 14772-1 Virtual Reality Model Language (VRML).
- [Wagner00] M. G. Wagner, Robust Watermarking of Polygonal Meshes, Proc. of the *Geometric Modeling & Processing 2000*, pp. 201-208, Hong Kong, April 10-12, 2000.
- [Yeo99] B-L. Yeo and M. M. Yeung, Watermarking 3D Objects for Verification, *IEEE CG&A*, pp. 36-45, January/February 1999.