情報処理学会論文誌採録決定 （2000 年 3 月頃を予定）

# NURBS曲線・曲面を対象とした幾何形状を保存する電子透かし埋め込み手法

## A Shape-Preserving Data Embedding Algorithm for URBS Curves and Surfaces

大渕 竜太郎
ohbuchi@acm.org
山梨大学工学部
コンピュータ･メディア工学科
〒400－8511 山梨県甲府市武田 4-3-11

増田 宏
masuda@race.u-tokyo.ac.jp
東京大学 人工物工学研究センター
〒113－8656 東京都目黒区駒場 4-6-1

**青野 雅樹**
aono@acm.org
**日本アイ･ビー･エム東京基礎研究所**
〒242－8502 神奈川県大和市
下鶴間 1623-14

## 要旨

各種のデジタルコンテンツを対象とし，電子透かしと呼ばれる情報をコンテンツ自体に埋め込み，この情報を著作権管理などの目的で利用しようとする研究が行われている．最近その埋め込み対象として 3 次元モデルが加わったが，これまでの電子透かし手法は 3 次元ポリゴンメッシュやその属性を対象として開発されおり，そのままでは（形状）CAD モデルには適用できない場合が殆どである．これは，CAD モデルの多くは主たる形状定義プリミティブとしてポリゴンではなくパラメタ曲線・曲面を用いていること，また CAD モデルでは既存の手法による透かし埋め込みに伴うトポロジや幾何形状の変更を許さない用途が圧倒的に多いこと，の２つの理由による．

本論文は，non-uniform rational B-spline (NURBS) 曲線および曲面を対象とした新たなデータ埋め込みアルゴリズムを提案する．このアルゴリズムは，再パラメタ化に用いる有理線形関数の持つ自由度を使って情報を埋め込む．再パラメタ化は NURBS 曲面・曲線の幾何形状を厳密に保存し，また，有理線形関数による再パラメタ化は NURBS の次数や節点数を変えないためモデルのデータ量も保存する．

本論文ではさらに，NURBS に限定しない種々のパラメタ曲線・曲面を対象とした情報埋め込み手法の概略を，幾何形状の保存とデータ量の保存という２つの要件で分類･整理して列挙した．

## Abstract

Digital watermarking adds various information to digital contents for copyright management and other applications. 3D model has recently been recognized as a watermarking target data type. However, existing watermarking algorithms target polygonal meshes and their attributes for watermarking so that they can’t be readily applied to the majority of (geometric) computer aided design (CAD) models for two major reasons. First, these CAD models employ parametric curves and surfaces, not polygonal meshes, as their main shape-defining primitives. Second, most CAD applications do not tolerate modifications of model topology and/or geometry that are introduced by existing watermarking algorithms.

This paper proposes a new watermarking algorithm for *non-uniform rational B-spline* (*NURBS*) curves and surfaces, which employs rational linear reparameterization for embedding messages. The algorithm exactly preserves the shape, — that is, the geometry and topology — of its watermarking targets. Furthermore, it preserves the data size of the model. We consider these two properties, exact preservation of shape and preservation of data size, to be important in various applications of CAD models.

In addition to the shape- and data size-preserving data embedding algorithm for NURBS curves and surfaces, this paper outlines additional methods for embedding data in various types of parametric curves and surfaces.

# 1. 初めに

データ埋め込み，または電子透かしと呼ばれる技術は，watermark（透かし）と呼ばれる何らかの構造体を，その存在が埋め込み対象となるコンテンツ本来の目的（例えば人による表示・鑑賞）を阻害しないようコンテンツに付加する [1]. 埋め込まれた透かしは，説明の付加，改ざんの検出，あるいは正規の購入者の認証など，そのコンテンツを何らかの形で管理する目的で用いることが出来る．

これまでのデータ埋め込みの研究の多くは「古典的」マルチメディアデータ型，例えば文字文書，静止画像，動画像，および音声データに対する埋め込みを中心としていた．しかし最近，3 次元（3D）データがその重要性を増すにつれ [2), 3)]，3D モデルに対する埋め込みアルゴリズムがいくつか発表された．

我々は 3D モデルの形状およびその属性を対象として情報を埋め込む手法を何種か提案してきた [4), 5), 6)]．我々が発表した手法は主に，3D ポリゴンメッシュによって定義された形状を埋め込みの対象としている．我々が提案した複数のアルゴリズムの幾つかは，あるクラスの幾何変換に対する不変量を利用して透かしを埋め込んだ．この不変量は頂点座標から誘導されたものである．不変量を用いたことで，3D モデルが日常的にさらされる可能性のある幾何変換，例えばアフィン変換によって透かしが壊れない．しかし，この透かしは頂点座標へのノイズの重畳により破壊される．我々はこの他，3D モデルのトポロジ，つまり頂点の結合関係を変更して埋め込む透かしや，3D モデルの形状に付随する属性，例えば頂点に付随するテクスチャ座標を変更してデータを埋め込む手法 [6)] も提案した．

Kanai らは 3 角メッシュをウェーブレット変換し，その変換領域の係数を操作して情報を埋め込む手法を提案した [7)]．彼らの透かしはアファイン変換に耐え，かつ頂点座標に重畳されたランダムノイズに対してもある程度の頑強性を有した．Praun らの手法 [8)] の考え方は Kanai らの手法に近いが，メッシュを直接ウェーブレット変換するのではなく，与えられたメッシュ上に多重解像度の基底関数（正規性や直交性は持たない）を立て，その係数を変更することで透かしの埋め込みを行った．Yeo らは fragile watermark（脆い透かし，後述）の概念を，2 次元の静止画像を対象として初めて提唱した．この脆い透かしの概念を同じ Yeo らが 3D モデルに適用した例が [9)] に述べられている． Benedens[10)] は 3D モデルの形状から求められた法線ベクタの集合を対象として，あるクラスの座標変換に対し頑強性を持つ透かしの埋め込み手法を述べた．また，形状そのものを対象としてはいないが，Hartung ら [11)] は MPEG-4 の facial animation parameters（顔アニメーションパラメタ群; FAPs) 列に対し情報を埋め込んだ．この研究で興味深いのは，顔アニメーションをレンダリングして得られた 2 次元動画を FAPs 抽出用の画像処理プログラムに入力し FAPs を抽出しすることで，レンダリング後の 2 次元動画から透かしの情報を取り出すことに成功した点である．

3 次元モデルは形状 CAD の分野でも大きな役割を果たしている．しかし，3次元形状を対象とするこれまでのデータ埋め込みアルゴリズムは，そのままでは形状 CAD データに適用できない場合が多い．これには 2 つの理由がある．第一に，大多数の CAD モデルはその形状の定義にポリゴンメッシュではなくパラメタ形式の曲線や曲面，例えば Bézier や non-uniform rational B-spline (NURBS)形式の曲線や曲面を用いる．従って，ポリゴンを埋め込みの対象とする透かしのアルゴリズムをそのまま適用することはできない．第二に，殆どの CAD モデルでは，その幾何形状やトポロジに少しでも変更を加えられるとそのモデルが使い物にならなくなる．したがって座標値や頂点のトポロジを変更するこれまでの電子透かしアプローチをそのまま使用することはできない．

本論文では，パラメータ形式の曲線や曲面を対象とした情報埋め込みアルゴリズムを提案する．本論文の貢献は以下のようにまとめることができる．

- NURBS 曲線・曲面からなる 3D モデルを対象に，埋め込みの前後でその幾何形状を厳密に保存し，さらにそのデータ量も保存する透かし埋め込みの具体的手法を提案した．この手法は rational linear reparameterization（有理線形再パラメタ化）の前後で NURBS 曲線・曲面の幾何形状が変わらない性質を利用している．
- NURBS に限らず，各種のパラメタ形式の曲線や曲面にデータを埋め込む複数の手法の概略を，形状の保存性とデータ量の保存性の 2 つの性質によって分類し列挙した．

以下，まず本節の残りの部分で電子透かしあるいはデータ埋め込み呼ばれる技術の一般的概念について簡単に紹介する．第 2 節では，NURBS 曲線・曲面を対象に，形状とデータ量を変えずに透かしを埋め込むアルゴリズムを紹介し，第 3 節ではこのアルゴリズムの実装と実験結果について述べる．次いで第 4 節では，NURBS に限定しない種々のパラメータ曲線・曲面を対象とした電子透かしのアプローチを複数提案する．最後に，第 5 節でまとめを述べ，これからの研究方向についてコメントする．

## 1.1. 情報埋め込み

本論文では，透かしをコンテントに追加することを（データ）埋め込み(embedding)と呼び，また埋め込まれた情報を目的に応じて取り出すことを取り出し(extraction)と呼ぶ．透かしが埋め込まれる対象物を cover-<datatype> (または被

覆-<datatype>), 透かしの入った対象を stego-<datatype> （または隠号-<datatype>），そして埋め込まれた情報を embedded-<datatype> (または埋め込み-<datatype>) と呼ぶ [12]. ここで "<datatype>" の部分はデータ型によって変わり，テキスト，画像，あるいは NURBS 曲線・曲面などとなる．

電子透かしはその可視性 (visibility), より一般的にはその可知性 (perceptibility) により分類できる．可知な透かしは，コンテンツの通常の使用環境下で観測者（通常は人）にとって可知であるが，不可知な透かしは何らかの機械的な処理を経てはじめて可知となる．不可知性や可知の度合いは，電子透かしがコンテンツ本来の使用目的を妨げないために重要な性質である．これまでの電子透かしはその殆どが不可知な透かしであるが，しかし，可知な透かしも，その可知な性質を積極的に使いコンテンツの無許可の再配布を牽制する，などの使い道がある．

ここで，3 次元モデルの場合，今までの電子透かしと比べて可知性の定義がより複雑となることを注意しておきたい．画像や音などを対象とした従来の電子透かしの可知性を議論する場合，被覆データの観測者は人であり，観測の過程は直接的である．例えば，画像の電子透かしを論ずるとき，その画像は画像表示プログラムやプリンタ等を使い変更を加えずに表示させるものだと仮定してきた．もちろん，例えば画像の表示も，表示デバイスの特性（輝度，色再現性，等），環境の明るさ，などにより影響を受けるが，次に述べる 3D モデルの場合に比べるとそのモデルは単純で直接的である．

3D モデルの場合には，観測者に関する仮定も，観測の課程も，より複雑である．まず，CAD モデルが被覆データの場合，被覆データである 3D モデルの観測者は人とは限らない．例えば，さらなるモデリングを行うための 3D CAD プログラムや，製造のための数値制御切削装置が，透かしの存在に影響されずに正しい結果を出すことが求められるかもしれない．さらに，たとえ観測者が人であったとしてもその観測はより間接的である． たとえば，ある 3D モデルが人にとって可知になるのは，レンダリングおよび表示の処理を行った後，あるいはそのモデルで定義された機械部品などが切削などによって実際に製造された後である．３D モデルの電子透かしの可知性を論ずるには，３D モデルの持つこれらの特性を考慮した新たな手法が必要と思われる．

電子透かしはまた頑強な透かし (robust watermark) と脆弱な透かし (fragile watermark) に分類できる．前者はコンテンツに加えられる変更が意図的 (intentional) か非意図的 (unintentional)かにかかわらず耐えて保存される必要がある．脆弱な透かしは，コンテントに対する全ての意図的な変更や，一部の非意図的な変更によって改変または破壊されることにより，そのコンテントの改ざんや変更，あるいは改ざん部分の検出をしようとするものである．ここで非意図的な変更とは，コンテントの通常の使用において予期されるもの，例えば静止画像における JPEG 圧縮・伸長や 3 次元モデルに対するアファイン変換である．これに対し意図的な変更とは，透かしを変更しあるいは破壊するなどの目的で意図的に加えられるものを指す．

別な分類として，透かしの取り出しに際して透かしの入った隠号データに加え，オリジナル（透かし埋め込み前）の被覆データを必要とするとき，その透かしは秘密透かし (private watermarking) と呼び，オリジナルがいらないとき公開透かし (public watermarking) と呼ぶ．例えば Hartung ら [11]の手法や本論文第 2 章で述べる NURBS を対象とした透かし手法は秘密透かしである．一般に，公開透かしのほうが使い道が広い．

また，暗号の考え方を使い，鍵を使って埋め込み情報を保護することもできる．例えば第三者による情報取り出しを避けるため，埋め込み時に（擬似）乱数系列を用いて透かしを埋め込んだ場所をかく乱すること，あるいは埋め込み時に透かしのメッセージ自体をかく乱するなどすることもある．この乱数系列生成の為の鍵を隠号鍵(stego-key)と呼ぶ．隠号鍵と乱数系列の関係は，秘密鍵暗号式でも公開鍵暗号式でも良い [13]. 乱数列によるかく乱にはまた，Hartung らの手法 [11]のように拡散帯域通信の考え方を使って埋め込みによる変更を分散し，妨害耐性を高めたりする使い方もある．

気をつけねばならないのは，その他の各種セキュリティ技術と同様に，電子透かし技術単体では，著作権保護などの現実の応用で効果を上げることはできない点である．実際に何が必要なのかはそれぞれの応用で異なるが，透かしと組み合わせて使う電子署名や，透かしを入れたデータを預託する為の信用できる預託機関などのインフラストラクチャ，そしてこれらインフラストラクチャの社会的・法的な認知など，満たされねばならない要件は数多い．

## 2. NURBS への形状不変な情報埋め込み

本節では，NURBS 曲面･曲線を対象とし，幾何形状を厳密に保存し，かつデータ量も変化しない情報埋め込み手法を述べる．本手法は秘密透かしで，従って取り出しにはオリジナルの被覆 3D モデルと，透かしの入った隠号 3D モデルの両方が必要である（図 1 参照）．

形状を保存する本透かし手法の基本的アイデアは，NURBS 曲面･曲線はその幾何形状を全く変えることなく再パラメタ化することができる，という事実に基づいている．以下，本節ではまず NURBS 曲線についてアルゴリズムを説明し，後にテンソル積として作られた NURBS 曲面に拡張する．

幾何形状を変えない性質（形状保存性）は CAD の用途の多くで要求される．例えば，多少とも幾何形状が変形したモデルを使った Constructive Solid Geometry 集合演算は失敗する可能性がある．

また，データ量を変えない性質（データ量保存性）も通信や蓄積のコスト削減に重要である．NURBS 曲面・曲線モデルのデータ量は主に control point（制御点）と weight（重み）ペアの数と，knot vector（節点ベクタ）の要素数で決まるが，本節で述べる手法は制御点・重みペアの数や節点ベクタの要素数を変えないという意味でモデルのデータ量に変化が無い．ただ，本論文の手法は節点ベクタなどの数値を変えるため，「データ圧縮をかけた後のビット数」の意味でのデータ量は埋め込み前後で変わり得る．

## 2.1. NURBS 曲線

本節ではまず簡単に NURBS 曲線を定義する．NURBS そのほかのパラメータ曲線･曲面について詳しくは Farin[14] や， Piegl と Tiller[15]，などの本を参考にしてほしい．本論文では Piegl と Tiller の本の定式化を用いた．

$p$ 次の NURBS 曲線 $\mathbf{C}(u)$ は，パラメタ（スケーラ量） $u$ が区間 $[a,b]$ で変化するにつれ，ある点が 3 次元空間をたどる軌跡を定義する．

$$\mathbf{C}(u)=\frac{\sum_{i=0}^{n} N_{i,p}(u)w_i\mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(u)w_i} \quad u\in[a,b], \tag{1}$$

ここで制御点の集合 $\{\mathbf{P}_i\}$ は control polygon（制御ポリゴン）をなし，また $\{w_i\}$ は重みである．重み $w_i$ の増加は曲線を制御点 $\mathbf{P}_i$ に引き寄せる効果がある．また $N_{i,p}(u)$ は，次数 $p$ （階数 $p+1$）の $i$ 番目の B-spline 基底関数で，再帰的に次のように定義される．

$$N_{i,o}(u)=\begin{cases}1 & \text{if} \quad u_i\le u<u_{i+1}\\ 0 & \text{otherwise}\end{cases} \tag{2}$$

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

パラメータの区間 $[a,b]$ の両端点において制御点を通過する（つまり $\mathbf{C}(a)=\mathbf{P}_0$ かつ $\mathbf{C}(b)=\mathbf{P}_n$ ）NURBS 曲線における Non-periodic（非周期的）かつ non-uniform（非一様）な knot vector（節点ベクタ）は区間 $[a,b]$ 内で非減少（増加するか値が変わらない）な実数値の列であり，以下のように定義される．

$$U=\{\underbrace{a,\ldots,a}_{p+1},u_{p+1},\ldots,u_{m-p-1},\underbrace{b,\ldots,b}_{p+1}\}. \tag{4}$$

ここで $a\le u_i\le u_{i+1}\le b$ and $i=0,\ldots,m-1$ である．

## 2.2. NURBS 曲線の再パラメタ化

パラメータの区間 $u\in[a,b]$ で定義された NURBS 曲線 $\mathbf{C}(u)=\{x(u),y(u),z(u)\}$ を関数 $u=f(s)$ で reparameterize（再パラメタ化）すると，この曲線は，もとのパラメタ $u$ の代わりに新たなパラメタ $s$ の関数として計算される．ここで，曲線上の点の軌跡が同一の点を 2 回以上通らないよう， $f(s)$ は区間 $s\in[c,d]$ 内で単調増加関数（つまり， $f'(s)>0$ ， $\forall s\in[c,d]$ ）と限定する．再パラメタ化についての詳細は [16] や [15] を参照してほしい．重要なのは，再パラメタ化はその前後で NURBS 曲線・曲面の幾何形状を厳密に保存する点である．

再パラメタ化関数 $f(s)$ は無数に存在する．しかし，例えば次数 2 以上の多項式で再パラメタ化した結果は NURBS 曲線であるが，その次数は元の曲線 $\mathbf{C}(s)$ より高く，従ってより多くの制御点を必要とし，モデルのデータ量が増える．

本論文の手法では有理線形式で再パラメタ化する．こうすると再パラメタ化の前後で NURBS 曲線の次数に変化が無く，よって制御点数の増加などのデータ量の増加を伴う変化がない．有理線形（有理 1 次，M bius，双線形，などとも呼ばれる）式による再パラメタ化は Lee と Lucian[17] が調べており，本論文の手法はその結果を用いている．以下 Lee らの結果を要約する．

有理線形関数 $g(u)$ は以下のように定義される.

$$s = g(u) = \frac{\alpha u + \beta}{\gamma u + \delta}. \tag{5}$$

$$u = f(s) = \frac{-\delta s + \beta}{\gamma s - \alpha} \quad s \in [c,d]. \tag{6}$$

ここで $f(s)$ は $g(u)$ の逆関数である. ここで

$$\mu(u) = \gamma u + \delta \quad \lambda(s) = \gamma s - \alpha. \tag{7}$$

とおく. $g(u)$ と $f(s)$ が互いに逆関数となる, (5)(6)の分母がゼロにならない, など式(5)(6)の振る舞いが良いためには

$$\begin{aligned} &\alpha\delta - \gamma\beta > 0, \\ &\mu(u) \neq 0 \quad \text{for all } u \in [a,b] \text{ , および} \\ &\lambda(s) \neq 0 \quad \text{for all } s \in [c,d] \end{aligned} \tag{8}$$

が必要である. NURBS 曲線 $\mathbf{C}(u)$ が与えられたとき, これを $g(u)$ で再パラメタ化した $\mathbf{C}(s)$ は以下のようにして求まる:

- 制御点 $\{\mathbf{P}_i\}$ は不変.
- 新しい knot(節点)ベクタは元の節点ベクタの個々の節点を $g(u)$ で写像したもの, つまり $s_i = g(u_i)$.
- 制御点の新たな重み $\{\overline{w}_i\}$ は (9)式のいずれか一方で計算する;

$$\overline{w}_i = w_i \prod_{j=1}^{p} \lambda(s_{i+j}) \text{ または } \overline{w}_i = \frac{w_i}{\prod_{j=1}^{p} \mu(u_{i+j})}. \tag{9}$$

ここで $s_{i+j}$ と $u_{i+j}$ はそれぞれ新・旧の節点である.

## 2.3. 埋め込みプリミティブとしての NURBS 曲線

本論文ではデータ埋め込みのためコンテンツに変更を加える最小単位を embedding primitive（埋め込み単位）と呼ぶ. 従って有理線形再パラメタ化を行って情報を埋め込んだ NURBS 曲線は, その 1 つ 1 つが埋め込みプリミティブである. 埋め込みプリミティブ 1 つあたりの埋め込み情報量は一般に小さいが, 複数の埋め込みプリミティブを組み合わせて相互の順序関係を導入すると有用な量の情報を埋め込むことができる.

自由度 3 を持つ有理線形再パラメタ化関数 $g(u)$ に情報を埋め込むには, その係数 $\alpha$, $\beta$, $\gamma$, および $\delta$ を操作すれば良い. この関数の自由度は, 係数が 4 つあるにもかかわらず 3 であることは以下の変形から明らかである:

$$s = g(u) = \frac{\alpha u + \beta}{\gamma u + \delta} = \frac{\frac{\alpha}{\gamma} u + \frac{\beta}{\gamma}}{u + \frac{\delta}{\gamma}} = \frac{k_1 u + k_2}{u + k_3}. \tag{10}$$

ここで $k_1 = \alpha/\gamma$, $k_2 = \beta/\gamma$ および $k_3 = \delta/\gamma$ と置いた.

これら 3 つの係数 $k_1$, $k_2$, および $k_3$ は, 例えば, 関数 $g(u)$ が通る 3 点 $(u_1,s_1)$, $(u_2,s_2)$, および $(u_3,s_3)$ を決めることで決定される（図2）. この 3 点を（10）式に代入して連立式として $k_1$, $k_2$, および $k_3$ について解くと以下のようになる:

$$k_1 = \frac{(u_1 s_1 - u_2 s_2)(s_1 - s_3) - (u_1 s_1 - u_3 s_3)(s_1 - s_2)}{(u_1 - u_2)(s_1 - s_3) - (u_1 - u_3)(s_1 - s_2)}$$

$$k_2 = u_1 s_1 + k_3 s_1 - k_1 u_1 \tag{11}$$

$$k_3 = \frac{(u_1 s_1 - u_3 s_3)(u_1 - u_2) - (u_1 s_1 - u_2 s_2)(u_1 - u_3)}{(u_1 - u_3)(s_1 - s_2) - (u_1 - u_2)(s_1 - s_3)}$$

式(10)の持つ 3 自由度全てを操作して情報を符号化することも可能だが，ここでは 2 端点を$u_1 = s_1 = a$および$u_3 = s_3 = b$で固定し，パラメタの区間が再パラメタ化の前後で同じになるようにした．情報は残った 1 自由度を使い，$D = s_2 - u_2$を変更して符号化する．ここで，偏移値$D$は，その値が大きいほど関数$s = g(u)$が直線$s = u$から離れる．

偏移値 $D$ の大きさの選択には注意が必要である．$D$が大きすぎるとパラメタ化が偏り，「悪い」パラメタ化とされる．（パラメタ化が「良い」とは，パラメタ空間上で等間隔の点が実空間の曲線上でもほぼ等間隔に写像される事を言う．）$D$の値が小さければ，「良い」パラメタ化をもつ曲線は，ほぼ同様に「良い」パラメタ化を持つ曲線に写される．もちろん$D$が小さすぎると計算誤差などによる影響を受けやすいので，適度な値を選ぶ必要がある．我々のプロトタイプでは$D$のとり得る範囲をユーザが指定することとした．

埋め込みは，具体的には偏移値$D$を直接に振幅変調して行った．$D$ のとり得る範囲$[D_{min}, D_{max}]$に $L$ ビットのデータ $d$（区間 $[0, 2^L - 1]$ の整数）を符号化して埋め込むとき，偏移値$D$の値は以下の式で求める:

$$D = \frac{(D_{\max} - D_{\min})(d + 0.5)}{2^L} + D_{\min} \tag{12}$$

この $D$ の値に基づいて，NURBS 曲線の interior knots（内部節点群）の位置を変更する．幾何形状を不変に保つため，節点の位置を変更した結果として各節点の値および制御点の重みが変化するが，制御点座標値そのものはまったく変わらないし，前述のように幾何形状は保存される．

ここで，埋め込み対象となる NURBS 曲線には少なくとも 1 つの内部節点が必要である．（内部節点を持たない NURBS 曲線は rational Bézier（有理 Bézier）曲線である．）現在の実装では，内部節点が $m$ 個ある場合には，常に$i = \lfloor m / 2 \rfloor$で求まる $i$ 番目の内部節点を偏移させて埋め込みを行った．しかし，どの節点を偏移させるかは，鍵を用いて作った乱数列などで決定することもできる．

情報の取り出しは，元々の曲線（被覆 NURBS 曲線）と再パラメタ化後の曲線（隠号 NURBS 曲線）の節点ベクトル値を比較し，埋め込み時と同じ節点でその差$D = s_2 - u_2$を求め，これから式（12）を用いてデータ $d$ を計算する．

埋め込みに際し，乱数系列を用いて埋め込みメッセージをかく乱することでデータのもつ規則的なパターン埋め込み結果に現れないようにすることが望ましい．また，複数の曲線･曲面に対して埋め込みをする場合，埋め込みをしない曲線・曲面も再パラメタ化を行い，情報の埋め込まれた曲線･曲面を容易に同定できないようにすることが望ましいだろう．

## 2.4. NURBS 曲面への埋め込み

ここまでは簡単のため曲線を対象として説明したが，このアルゴリズムは 2 つの NURBS 曲線のテンソル積として作られる NURBS 曲面に容易に拡張できる．NURBS 曲面が $u$ と $v$ でパラメタ化されていたとするなら，前節の手法をこれら 2 つのパラメタそれぞれに適用し，2 つの偏移値 $D_u$ と $D_v$ を使って曲面を再パラメタ化すれば良い．NURBS 曲線に $L$ ビット埋め込めたとすると，NURBS 曲面には $2L$ ビット埋め込むことができる．

## 2.5. 埋め込みプリミティブの順序付け

埋め込みプリミティブ１つ当たりに埋め込めるデータ量は小さい．しかし埋め込みプリミティブ間で順序をつけることにより，複数のプリミティブ全体でより大きな容量のデータを埋め込むことが可能である．CAD モデルの場合は，ひとつのモデルを構成する曲面，曲線に識別番号がついており，これによってプリミティブの順序付けを実現できる．現実の CAD モデルは数百，数千を超える曲面，曲線（特にトリム曲線）からなっている．したがって CAD モデル全体としては種々の用途に十分な量のデータを埋め込むことが可能であろう．

既に述べたように，埋め込みに際しては，プリミティブの順序とプリミティブに埋め込まれるメッセージの間の対応を，公開鍵暗号などを用いた擬似乱数列によってかく乱することが考えられ，これにより第三者による情報の取り出しなど

を防ぐことができる．

## 2.6. 本埋め込み手法の特性と使い道

ここで述べた再パラメタ化による埋め込み手法は，厳密な形状保存性とデータ量保存性という 2 つの大きな特長を持っている．しかしその欠点はその破壊が容易な事で，透かしの埋め込まれた曲面･曲線にさらに任意の再パラメタ化を加えれば透かしが消えてしまう．従って，本節で述べた手法は脆弱な透かしとして使い，認証や改ざんの検出，例えば「正規の認証コードが入っていない場合は何らかの予期せぬ変更が加えられた」と検出するような使い方が主になるであろう．

この手法はまた，対象となる曲線や曲面がパラメタ化を積極的に利用する場合には適用できない場合が多い．例えば ruled surface はその形状がパラメタ化に依存するし，トリム曲線もパラメタ化が重要である．このような曲線，曲面に対して再パラメタ化を施すことは許されず，これまで述べた手法は適用できない．

# 3. 実験と結果

我々は第 2 節に述べた NURBS 曲線・曲面を対象とした情報埋め込みアルゴリズムを実装した．現在の実装では，曲線当たり $L$ ビット ($L$=8 程度)，曲面当たり $2L$ bit の情報を埋め込む．曲線･曲面の埋め込みプリミティブ間の順序は CAD システムの持つ順序番号を用いる事とした．プログラムは C++で書き，表示には OpenGL を用いた．

図 3.(a)-(b) はある 3 次 NURBS 曲線を 2 種類の偏移値 $D$ で再パラメタ化した結果を示す．折れ線は制御点からなる制御ポリゴンであり，各制御点は丸で示されている．曲線には，区間 $[a,b]$ 上に等間隔に置かれた，両端を含めて 10 個のパラメータ値に対応する 10 個のマーカーを付けてパラメタ化の状況を視覚化してある．マーカーは，再パラメタ化前は十字，再パラメタ化後は四角（ひし形）で表現している．比較的小さな偏移値 $D=0.01$ では再パラメタ化の前後でパラメタ化の変化は小さい．逆に大き目の偏移値，例えば $D=0.05$ では明らかに偏ったパラメタ化が認められる．

図 3. (c)-(d) は $L=8\text{bit}$ の情報を埋め込んだ NURBS 曲線を示す．$L=8\text{bit}$ の情報は節点を表現する倍精度浮動小数点数（64 bit）に埋め込まれている．

図4．と図5．はある NURBS 曲面を再パラメタ化して情報を埋め込んだ例である．図 5. (a) では 2 つのパラメタ $u$ と $v$ の偏移値として $D_u=0.001$ および $D_v=0.02$ を用いた．実線が再パラメタ化前，点線が再パラメタ化後のパラメタ化をあらわしている．図 5. (b) は ASCII 文字 “z!” の 2 文字の文字コードを合計 16 bit のデータとして NURBS 曲面に埋め込んだ例である．偏移値 $D$ の範囲は $[D_{min},D_{max}]=[0.001,0.01]$ と意図的に大きめに取り，再パラメタ化によるパラメタ化の変化が見えるようにしてある．ASCII 文字コードでは “z” のほうが “!” より値が大きいので，こちらのほうがパラメタ化の変化も大きい．ちなみに埋め込みビット数 $L=8$ の場合には，$D_{\max}$ をこれよりかなり小さくしても埋め込みの安定性に影響は無い．

# 4. 曲線・曲面に情報を埋め込むその他の手法.

第 2 章では NURBS 曲面･曲線を対象とし，形状不変かつデータ量不変に電子透かしを埋め込む方法を述べた．本節では，NURBS を含め，Bézier, 有理 Bézier, B-spline などの種々のパラメトリック曲線・曲面を対象に情報を埋め込む場合に考えられる手法幾つかの概略を述べる．これらの埋め込み手法は，形状保存性，すなわち幾何形状を厳密に保存するかどうか，およびデータ量保存性，すなわちモデルデータのサイズは保存するか，の二つの特徴で分類した．ただし，形状，データ量いずれの保存も不要な場合については特に言及しない．

## 4.1. 形状を保存し，データ量を保存しない手法

形状保存し，データ量を保存しない手法は，以下のようにして実現できる．以下では簡単のため主に曲線を対象に話しを進めるが，テンソル積曲面への拡張も容易に可能である．

- **Knot insertion (節点挿入)**：複数の区間からなるパラメータ曲線，例えば非有理 B-Spline や NURBS 曲線に節点を挿入する．その節点の値，節点の存在そのもの，あるいは節点ベクトル中での新たな節点の（順序）位置，などを使って情報を符号化し埋め込みむ．
- **Degree elevation (次数上げ)**：非有理な Bézier や B-spline 曲線の次数上げは制御点の追加を起こす．次数がどれだけ増えたか，どの区間に制御点を挿入したか，などに情報を埋め込むことができる．

- **次数上げの伴う再パラメタ化**: 有理 Bézier や NURBS では，次数上げを伴う再パラメタ化により情報を埋め込むことができる． 次数上げを伴う再パラメタ化は，新たな制御点や節点を導入する．

これらのデータ埋め込み手法で導入された制御点や節点を，形状を保存しつつ除去することはかなり困難であろう．従って，ここで述べたアプローチは形状の保存を重んじる CAD 用途における著作権保護や秘密漏洩の追跡を目的とした頑強な埋め込みに向いている可能性がある．ただしデータ量保存性を要求する用途には適当でない．

## 4.2. 形状を保存せず，データ量を保存する手法

形状の厳密な保存は必要ないがデータ量は保存したい場合の手法もいくつか考えられる．以下で述べる手法は，第 4.2 節の手法ほどではないにしろ，比較的高い頑強性を実現できる．

- **制御点座標値の変更**: 制御ポリゴンをポリゴンメッシュとして扱えば，ポリゴンの頂点座標値を変えて埋め込む情報埋め込みアルゴリズムの考え方がほぼそのまま適用できる．また，一般のポリゴンの頂点群と異なり，曲線や曲面の制御点群には既に 1 次元あるいは 2 次元の順序がついているため，これら制御点座標値の列を単なる数値列として扱ってその数値を変更し，情報を埋め込むことが出来る．この手法はほとんど全てのパラメタ曲線に適用できる．
- **節点ベクトル要素値の変更**: 節点ベクトルを単なる数値列として扱い，その値を変更して情報を埋め込むこともできる．NURBS や B-スプラインなど複数区間からなり節点ベクトルを持つパラメタ曲線に適用できる．

制御点座標値，節点ベクタ要素値のいずれかを変更する場合は，値をそのまま変更しても良いし，離散コサイン変換（DCT）などの変換を施した変換領域でその係数を変更しても良い．例えば，曲面の制御点座標を DCT 変換した係数を変更して透かし埋め込む手法は，我々が [18] で提案した DCT を用いる lossy な(完全な復元が不可能な)曲面形状圧縮手法と組み合わせることが可能だろう．

## 5. まとめと今後の課題

本論文では，パラメトリック曲線や曲面で定義された 3 次元形状を対象とする電子透かし埋め込み手法について述べた．

まず，第 2 章では，形状が NURBS 曲面・曲線で表現される 3 次元モデルを対象とし，その幾何形状を厳密に保存しつつ情報を埋め込む手法を提案した．この手法は NURBS 曲面・曲線が有理線形関数による再パラメタ化前後で幾何形状を変えない性質を利用している．この手法はまた，埋め込み前後でモデルのデータ量を変えない．第 3 章ではこの手法を実装し実験した結果を示した．

第 4 章では，NURBS に限らずパラメトリック曲線・曲面に対し情報を埋め込む複数の手法の概略を，形状保存性とデータ量保存性を使って分類して列挙した．

われわれは，今後，第 4 節でその概要を述べた一般のパラメタ曲面･曲線を対象とした各種の情報埋め込み手法の中で，特に，データ量は保存しないが形状を保存しかつ頑強な手法を具体化し実装したい．さらに，これら手法を使って，実際の CAD データを対象とした埋め込み実験も行いたい．

### 参考文献


1) **松井 甲子雄，電子透かしの基礎，森北出版，東京都，1998年8月．**

2) ISO/IEC 14772-1 Virtual Reality Model Language (VRML).

3) ISO/IEC JTC1/SC29/WG11 *MPEG-4 Visual* and *MPEG 4 SNHC*.

4) R. Ohbuchi, H. Masuda, and M. Aono, Watermarking Three-Dimensional Polygonal Models, *Proceedings of the ACM Multimedia '97*, Seattle, Washington, USA, pp. 261-272, November 1997.

5) R. Ohbuchi, H. Masuda, and M. Aono, Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications, Volume 16, No. 4, pp. 551-560, *IEEE Journal on Selected Areas in Communications*, May 1998.

6) 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).

7) 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 Modeling: Fundamentals and Applications (GEO-6)*, pp. 296-307, Tokyo, Japan, December 1998.

8) Emil Praun, Hugues Hoppe, Adam Finkelstein, Robust Mesh Watermarking, Proceedings of the *ACM SIGGRAPH '99*, Los Angeles, CA, USA, August, 1999, pp. 49-56.

9) B-L. Yeo and M. M. Yeung, Watermarking 3D Objects for Verification, *IEEE CG&A*, pp. 36-45, January/February 1999.

10) O. Benedens, Geometry-Based Watermarking of 3D Models, *IEEE CG&A*, pp. 46-55, January/February 1999.

11) F. Hartung, P. Eisert, and B. Girod, Digital Watermarking of MPEG-4 Facial Animation Parameters, *Computer and Graphics*, Vol. 22, No. 4, pp. 425-435, Elsevier, 1998.

12) B. Pfitzmann, Information Hiding Terminology, in R. Anderson, Ed., *Lecture Notes in Computer Science*, No. 1174, pp. 347-350, Springer, 1996.

13) A. J. Menezes, P. C. van Oorshot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

14) G. E. Farin, *Curves and Surfaces for Computer-Aided Geometric Design, A Practical Guide*, Fourth Edition, Academic Press, San Diego, CA, 1997.

15) L. Piegl, W. Tiller, *The NURBS Book*, 2nd Edition, Springer, Berlin, 1997.

16) W. Tiller, Rational B-splines for curve and surface representation, *IEEE Computer Graphics and Applications*, Vol. 3, No.6, pp. 61-69, 1983.

17) E. T. Y. Lee, and M. L. Lucian, Mobius Reparameterizations of Rational B-splines, Computer Aided Geometric Design, Vol. 8, pp. 213-215, Elsevier, 1991.

18) **増田 宏，大渕 竜太郎，青野 雅樹, 周波数領域での曲面データの圧縮と転送, 情報処理学会論文誌，第40巻 第7号，** pp.1188-1195**，** 1999**年3月号**.

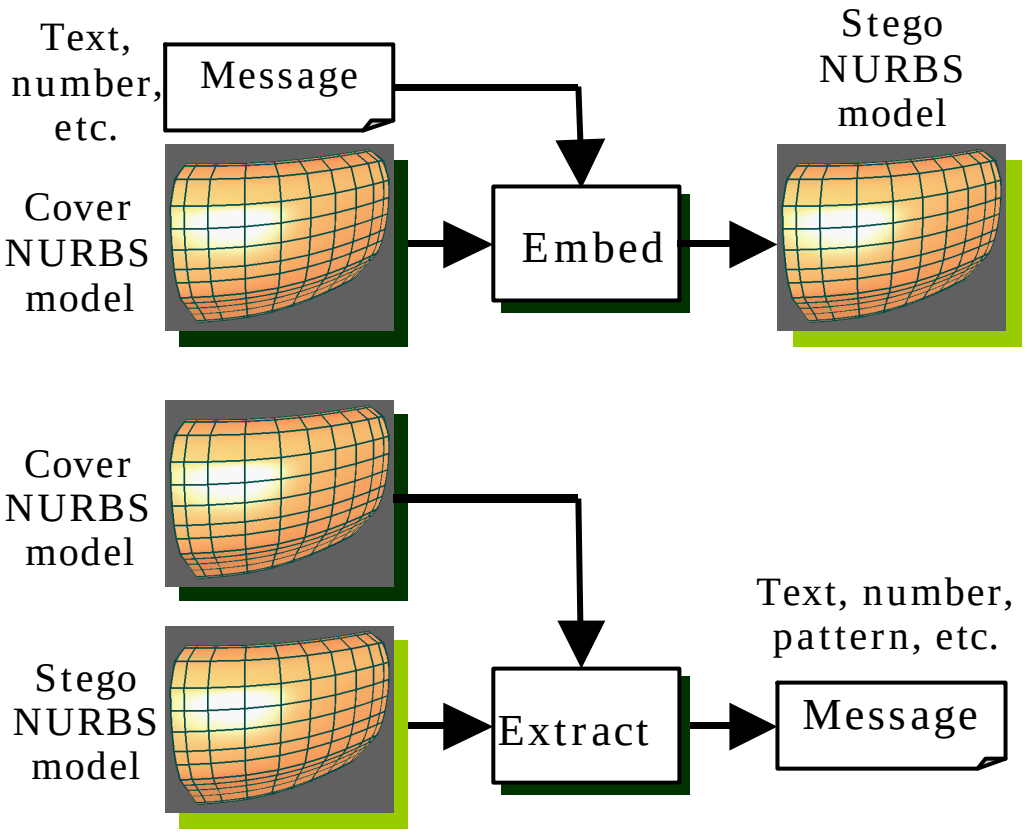



図 1. NURBS 曲面･曲線を対象とする秘密透かし手法のデータの流れ．取り出しには透かしの入っていない被覆 NURBS モデルが必要である．

Figure 1. Data flow of the private watermarking algorithm for NURBS curves and surfaces. Extraction requires the original cover-NURBS model without the watermark.

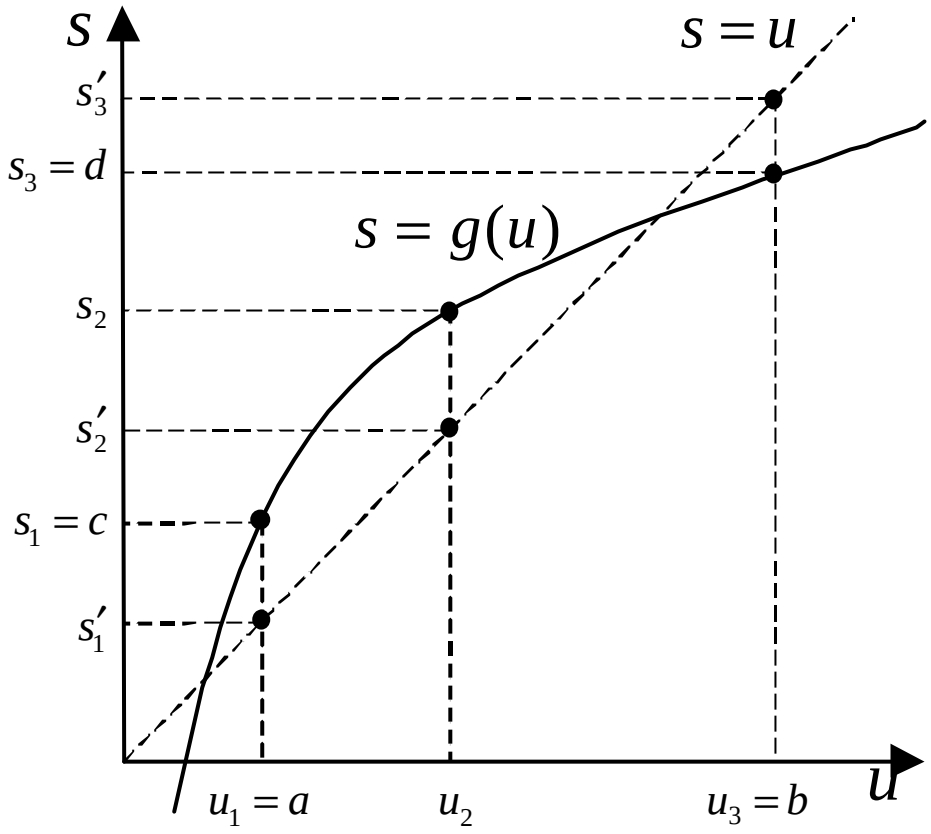
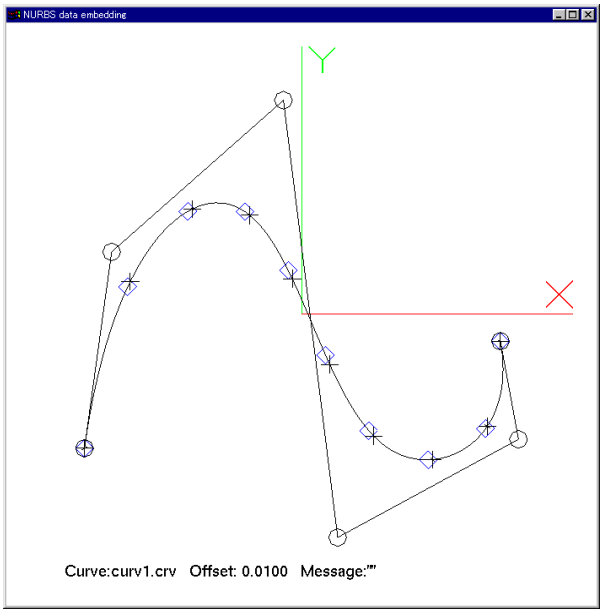


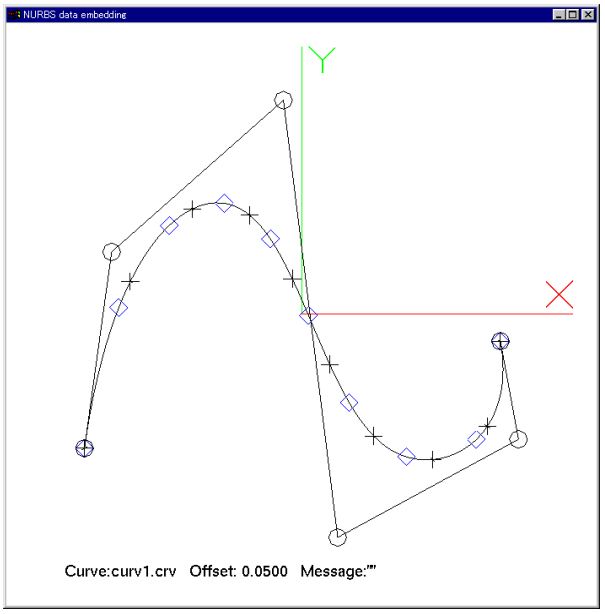

図 2. 再パラメタ化に使用する自由度 3 の有理線形関数を決定する.

Figure 2. Determining a rational linear reparameterization function with degrees-of-freedom 3 for the reparameterization.



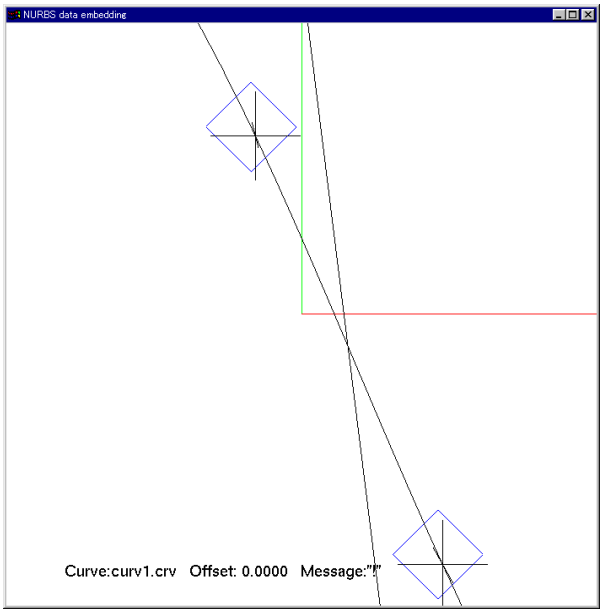

(a) **偏移値** $D$ = 0.01.

(a) Displacement $D$=0.01.



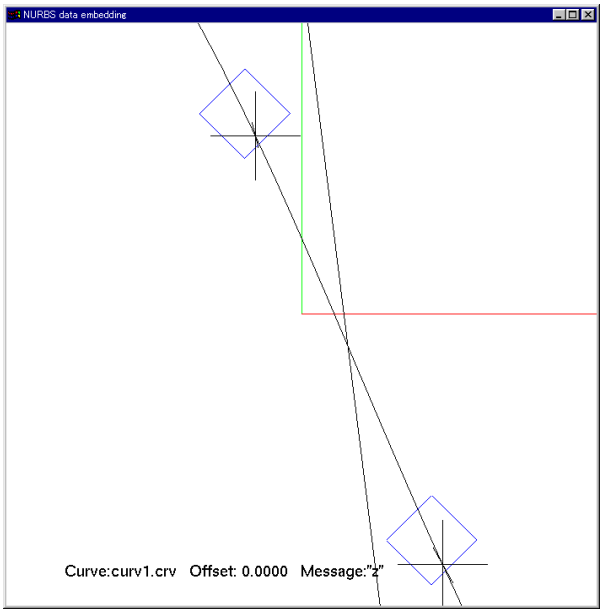

(b) **偏移値** $D$ = 0.05.

(b) Displacement $D$=0.05.




(c) 文字 “!” の埋め込み後（拡大図）.

(c) Result of embedding a letter “!” (a close-up view).




(d) 文字 “z” の埋め込み後（拡大図）.

(d) Result of embedding a letter “z” (a close-up view).

図 3. ある NURBS 曲線を，図(a) は $D$ = 0.01, (b) は $D$ = 0. 5 の偏移値で再パラメタ化した例．図 の (c) と(d) は同じ曲線に文字 “!” と “z” のコードを埋め込んだ例.

Figure 3. Example of reparameterizing a NURBS curve with offsets (a) D=0.01 and (b) D=0.5. In figures (c) and (d), letters “!” and “z”, respectively, are embedded.

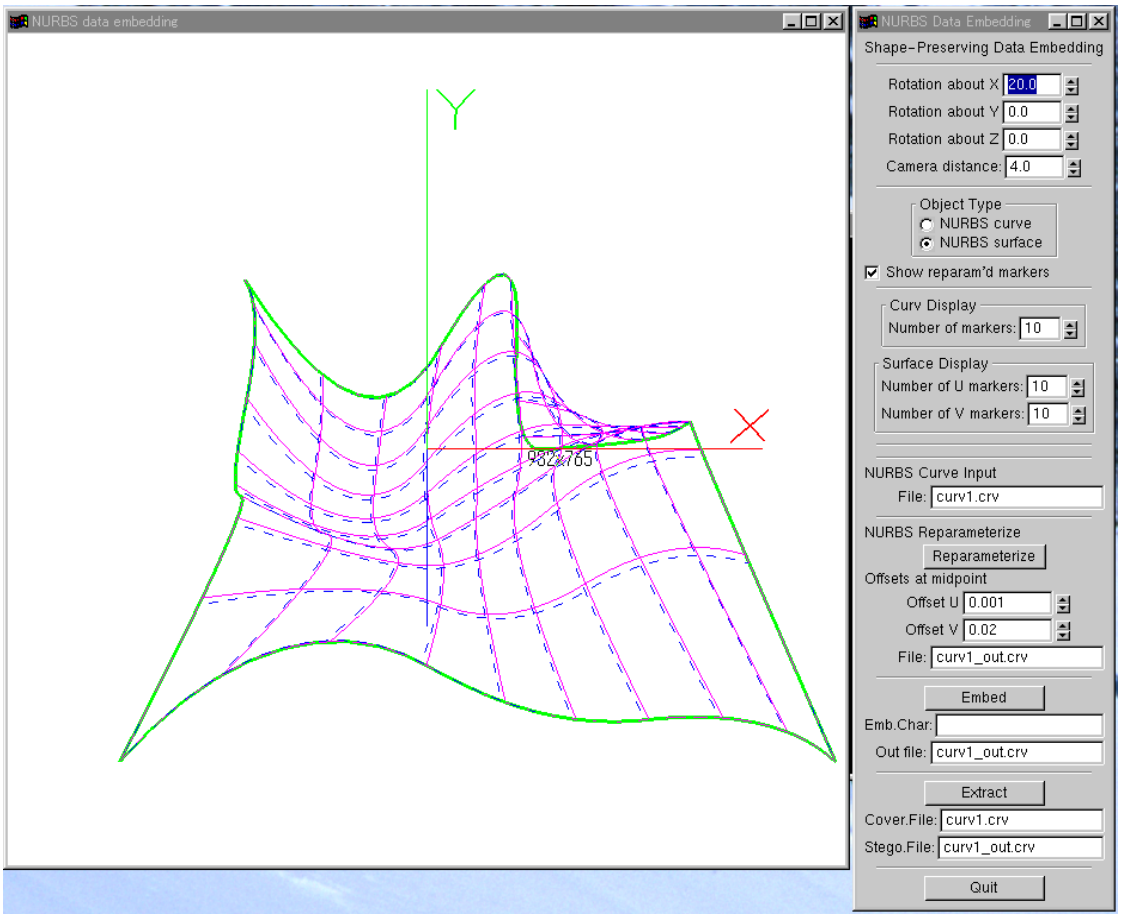
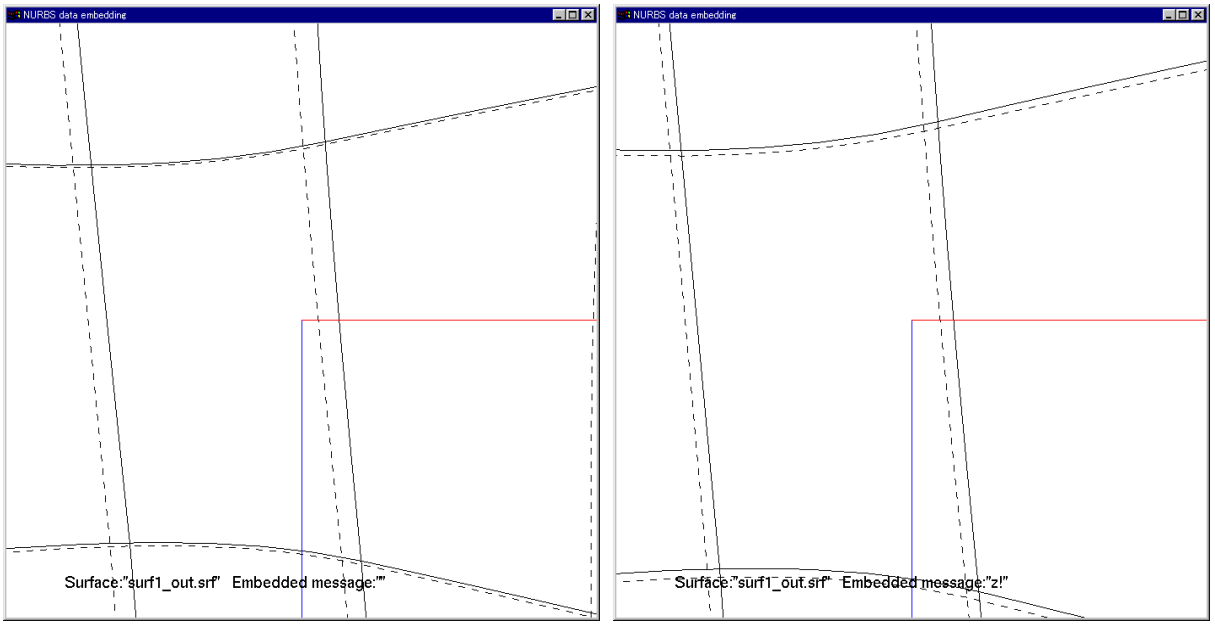



図 4. NURBS 曲面を $D_u = 0.001$, $D_v = 0.02$ の偏移値を用いて再パラメタ化した結果. 実線が再パラメタ化前, 点線が再パラメタ化後.

Figure 4. A NURBS surface of degree 3 in both u and v is reparameterized with dislacements $D_u = 0.001$, $D_v = 0.02$.




(a) **偏移** $D_u = 0.01$ **及び** $D_v = 0.001$ で再パラメタ化した結果.

(a) Reparameterizd with displacements $D_u = 0.01$ and $D_v = 0.001$.

(a) 2 文字，16bit のデータ“z!” を埋め込んだ結果.

(b) A 2-byte string “z!” was embedded.

図 5. 前出の図 4. と同一の曲面を拡大．いずれも実線が再パラメタ化前，点線が再パラメタ化後.
Figure 5. A close-up view of the surface shown in Figure 4. In both of the figures, solid and dotted lines, respectively, indicate parameterizations of the surface before and after the reparameterization.