閉曲線点列の3次スプライン補間問題における「うねり」除去の一手法†

原田耕一††　中前栄八郎††

3次元物体を立体表示する簡易な方法として，一組の透視平面上に投影された線画を用いる方式がある．ここで用いられる線画を得る方式として，2種類の方法，すなわち，(a)3次元空間内で線画表示されたものを2次元平面に透視変換する方法，および(b)3次元図形のデータ点のみ透視変換し，2次元空間上で補間を行う方法，が考えられる．しかし，透視変換される点の数を制限するという立場からは(b)の方が有利である．

本論文では，(b)の方式を3次スプライン曲線による補間によって実行する立場をとり，この補間曲線がもつ「うねり」を除去する方法について論じる．「うねり」を除去する方法として，すでにいくつかの手法が提案されているが，ここでは補間曲線の節点を移動することを新たに提案する．この方法は，(i)補間関数が簡単である，(ii)アルゴリズムが比較的簡単である，という特徴を持っており，効率的に「うねり」を取り除いて視覚的に自然な立体像を与えている．適用例として，3次元空間内にあるトーラスを取り上げている．

1. まえがき

最近のコンピュータ・グラフィックス技術の応用のうち，重要なものの1つは，濃淡付けされた物体の立体表示である[1]．これは，航空機や船舶の運転技術習得を目的としたシミュレーションに主として利用されている．この場合の立体表示の方式は，一組の平面図形によるものである．すなわち，左右両眼にそれぞれ提示されるべき一組の画像を創成し，何らかの方法によって，これらを，相互に干渉することなくそれぞれの眼に与え，両眼視差[2]による立体感を得ている．このようにして得られる立体画像は，実存の物体に類似しているという点で理想的であるが，画像を創成するために必要な計算量およびメモリ数という観点からは，現在の計算機技術の発達を考慮しても，簡易的なものであるとはいい難い．しかしながら，2次元的に表示された物体を観察することによって元の3次元物体を脳裏に描くことは，人間の不得意な能力の1つであると考えられるので，計算機による処理結果を，立体的に表示すること自身は，きわめて有用である．

そこで，本論文では，CAL (Computer Assisted Learning)[3]で現われるような，比較的簡単な3次元図形を取り上げ，これを線画によって立体的に表示する問題を取り扱う．すなわち，上記の立体視手法を，線画図形を用いて実行する場合に必要な補間問題に着目し，検討を行う．ところで，線画による立体視の方式は，補間を施す空間の次元によって**図1**(a)，(b)のように区分することができる．これらの方式の本質的な違いは，3次元図形を2次元図形に変換するための透視変換[4]と，少数のデータ点列を滑らかに結ぶ補間操作との順序に表われている．この図から理解できるように，(a)は(b)よりずっと多くの透視変換を必要とすることになる．ところで，透視変換のための演算は，図形表示の高速化において1つの障害となっていることが知られている[4]．したがって，インテリジェント・ターミナルに具備されているような小規模計算機で立体図形表示を行う場合は，(b)の方式を用いる方が適切であるといえる．本論文では，さらに，補間のために必要な演算量，およびメモリ数を縮少化のために，3次スプライン方式による補間手法[4]を用いることにする．このような理由により，以下では，平面上にある閉曲線のデータ点列を補間する問題を取り扱う．

3次スプライン関数を補間問題に適用する場合，データ点列の配列によっては，視覚的に不自然な「う

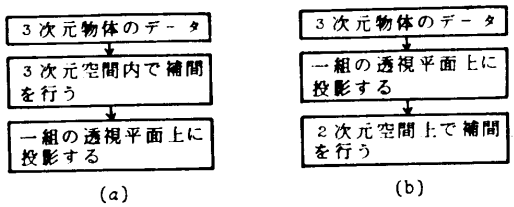

図1 線画による立体視の方式
Fig. 1 Two methods to generate stereoscopic line-drawing image.

† A method to get rid of "Undulations" of cubic spline interpolants for closed curves. by KOICHI HARADA and EIHACHIRO NAKAMAE (Faculty of Engineering, Hiroshima University.)

†† 広島大学工学部

ねり」[5]を生じることがある．当然のことながら，この「うねり」は不自然な立体感をもたらす．この「うねり」は本質的にはデータ点不足からくるものであり，データ点を増加する方法は「うねり」除去に効果があることはいうまでもない．しかしながら，この方式を単純に採用することはできない．なぜなら，データ点が不足しがちなのは，被補間関数の曲率が大きい部分においてであるが，このような部分においてのみデータ点数を増加すれば，データ点間隔の不均一性が増大し，これに起因する「うねり」が生じるからである．さらに，透視変換後の図形において，曲率が大きくなる場所を原図形であらかじめ定めておくことは，視点の位置を種々に変えた場合の図形を実時間で表示する場合には不可能であるといってよい．したがって，「うねり」除去の方策としてデータ点増加は，被補間点列全体にわたってのデータ点数増加を要求し，前述の計算量およびメモリ数の縮小化要求と相反することになる．そこで，与えられたデータ点列を操作することはせずに，この点列を補間する場合の補間手法に依存する「うねり」を極力制限することを考える．なお，与えられた点列を直線によって補間して得られる関数には「うねり」は含まれないものとし，この補間関数を「うねり」除去の基準関数として用いることにする．

「うねり」除去には後述のように種々の方法があるが，本論文では節点移動による方式を用いる．節点を移動する方法は従来，B-spline 関数について広く研究されている[6]．この関数を補間問題[7]あるいはデータ平滑化問題[8]に適用した報告もすでになされている．ここで取り上げる3次スプラインは B-spline に比べ，節点数が少ないという特徴がある．この点は前者に処理手順を簡単化するという長所を与える反面，後者にはない制限をも課している．すなわち，B-spline の節点（データ点に一致しないもの）は Schoenberg-Whitney の条件[8]を満足する範囲で自由に選ぶことができるが，一般的な手法における3次スプラインの節点移動は，補間曲線のデータ点通過という条件を満足させない．したがって，本論文では，節曲線という概念を導入することによりこのような不合理を回避するとともに，3次スプライン曲線のもつ長所を最大限利用すべく解析を行っている．

2. 3次スプライン曲線

3次のスプライン曲線は種々のスプライン曲線[9]の中で最も簡単なものであり，きわめて良好な補間を与えるため，広範に使用されている[4]．この関数は，データ点間ごとに区分的に計算される．いま，データ点を $P_1, P_2 \cdots P_n$ で表現することにすれば，P_i, P_{i+1} 間のセグメントは3次スプライン曲線により，次式で表わされる．

$$P_i(t)=B_1+B_2t+B_3t^2+B_4t^3,$$
$$0\leq t\leq t_i,\ t_i\triangleq|P_{i+1}-P_1|, \qquad (1)$$

ここで，B_1, B_2, B_3, B_4 は次の4つの条件によって決定される定数である．

$$P(0)=P_i,\ P(t_i)=P_{i+1},\ \frac{d}{dt}P(t)\Big|_{t=0}=P_i'$$
$$\frac{d}{dt}P(t)\Big|_{t=t_i}=P_{i+1}', \qquad (2)$$

上式のうち，P_i および P_{i+1} の座標はデータとして与えられるが，2つの導関数 P_i' および P_{i+1}' は算出しなければならない．このための条件式として，3次スプライン曲線が2階導関数まで連続であることによる条件以外に，2個の条件が必要である．これらの条件は通常，両端にあるデータ，すなわち，P_1 および P_n において課せられる．**1.** で述べたように，本論文は3次元物体の線画表示を目的としているもので，ここでは，周期的な境界条件を仮定する．すなわち，

$$P_1=P_n,\ P_1'=P_n',\ P_1''=P_n'', \qquad (3)$$

とする．この仮定のもとで導関数算出方程式は次式となる．

$$\begin{bmatrix} 2(t_{n-1}+t_1) & t_{n-1} & & & 0 & t_1 \\ t_2 & 2(t_1+t_2) & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & t_{n-2} & \\ & & & t_{n-1} & 2(t_{n-2}+t_{n-1}) & \end{bmatrix} \begin{bmatrix} P_1' \\ P_2' \\ \vdots \\ P_{n-1}' \end{bmatrix}$$
$$= \begin{bmatrix} \frac{3}{t_{n-1}t_1}(t_{n-1}^2D_1-t_1^2D_{n-1}) \\ \frac{3}{t_1t_2}(t_1^2D_2-t_2^2D_1) \\ \vdots \\ \frac{3}{t_{n-2}t_{n-1}}(t_{n-2}^2D_{n-1}-t_{n-1}^2D_{n-2}) \end{bmatrix}, \qquad (4)$$

$$D_i\triangleq P_{i+1}-P_i\ (i=1,2,\cdots,n-1).$$

以上の手続きによって得られる3次スプライン曲線は，パラメトリックな手法[4]により，容易に2次元，もしくは3次元へと拡張することができる．つまり，与えられた位置ベクトルの各成分について，データ点間距離をパラメータとして式(1)から式(4)までの手続きを個別に適用すれば，希望する次元の補間曲線を

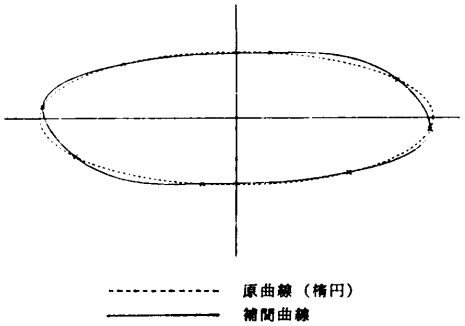

図 2 「うねり」の例
Fig. 2 An example of "undulation".

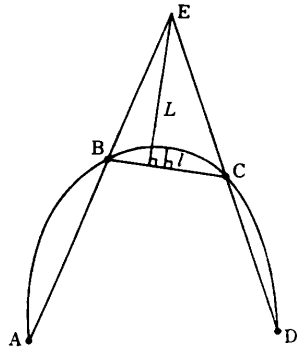

図 3 セグメントの「ふくらみ」
Fig. 3 "Bulge" of a segment.

得ることができる．したがって，一般性を失うことなく $P(t)$ は一価のスカラ値関係であるとして解析を進めることができる．また，以下では図1(b)の方式を用いるので，補間曲線の次元は2次元とする．

前述のように，データ点の配置によっては，スプライン曲線は，視覚的に不自然な「うねり」を生じることがある．**図2**に一例を示す．「うねり」の特徴として，この図からも理解できるように，曲率の大きい部分の近傍で発生しやすい点が挙げられる．一般に，透視変換された後の点列は，もとの点列のもつ特徴を維持しない．換言すれば，3次元空間内でたとえ曲率が小さくても，透視平面の位置によって，曲率の大きいデータ点列を生じることになる．したがって，(b)の方式を採用する際は，「うねり」が生じやすいことになる．そこで，次章において「うねり」の定式化を行い，数学的な取り扱いが可能な表現を与える．

3.「うねり」の定式化

図2から判るように，スプラインセグメントの原曲線からの偏差は，データ点間ごとに大きく異なっている．ところで，我々が，線画図形を観察する際に，視覚的に不自然な「うねり」として感じるのは，このようなデータ点間の偏差の差異であると考えられる．したがって，「うねり」を定式化するには，まず，各セグメントの形状，すなわち，「ふくらみ」を定式化する必要がある．この目的のために基準関数として直線補間関数を用い次の定義を行う．

[定義 1] セグメントの「ふくらみ」

図3において，A, B, C, D を連続する4つのデータ点列であるとする．いま，B, C 間のセグメントに注目した場合，線分 $\overline{AB}$ および $\overline{DC}$ を，それぞれ延長して得られる半直線の交点をEとする．Eと弦 $\overline{BC}$ との距離，および BC 間のセグメントと弦 $\overline{BC}$ との最大距離をそれぞれ L, l とした場合，l/L を $\overline{BC}$ 間のセグメントの「ふくらみ」という．なお，$\overline{AB}/\!/\overline{CD}$ の場合は $L\to\infty$ とし，「ふくらみ」は0と考える．

前述の「ふくらみ」を用いることにより，「うねり」は次のように定義できる．

[定義 2] セグメント間の「うねり」

3次スプラインセグメントの「ふくらみ」が隣接するデータ点間で，ある規定された量 ε 以上の差をもって変化している場合は，これらのセグメント間で「うねり」を生じているという．

この定義中の ε の値は，表示される図柄に許容される「うねり」の程度，および処理を実行する計算機の演算速度を考慮して決定しなければならない．なお次章のアルゴリズム中に，この量の一例が示されている．

以上の定義を，アルゴリズム中に具体的に組み込む場合，l の算出のみが問題となる．(1)式から判るように，3次スプラインセグメント上の点は，x 座標および y 座標についてそれぞれ t に関する3次式となる．一方，弦の方程式は，これらの座標についてそれぞれ t の一次式となるので，両者の差として与えられる関数は，結局，x 座標および y 座標についてそれぞれ，t の3次式となる．これらを，$x_d(t), y_d(t)$ と書けば，

$$l=\max_{0\le t\le t_i}\sqrt{x_d^2(t)+y_d^2(t)}, \qquad (5)$$

となる．ここで，$x_d(t)$ および $y_d(t)$ の各係数は，セグメントが決定される段階で定まるので，(5)式は，係数が既知の6次関数の最大値を求める問題となる．したがって，計算機により，これを数値的に解析することができる．しかしながら，「ふくらみ」の厳密な値が要求されているわけではないので，セグメント上にいくつかの点をサンプルし，これらの点と弦との最

大距離を求める方が効率的である．そこで，本論文では，後者の方法を用いることとし，次章において，具体的なアルゴリズムを示す．

4. 節点移動による「うねり」の除去

スプライン曲線の「うねり」除去の方式として，いくつかの方法が提案されている．de Boor は，ある手続きにより，与えられたデータ点の間に，新たに点を追加して「うねり」を回避しようとしている[6]．また，Nielson は，張力を導入することによって「うねり」を除去しようとしている[10]．しかし，前者の方法は，アルゴリズムが複雑であり，また，後者は指数関数が補間関数の中に含まれているため，ここでの応用に適しているとはいえない．そこで，節点移動による「うねり」除去の方法を，新たに提案する．なお，節点とは，スプラインセグメントの接続点[9]のことであり，補間問題においては，これまで，節点はデータ点に一致しているものとして解析が行われてきた．本章以降では，節点は必ずしもデータ点に一致しているとは限らないので，次の定義を施すことは有用である．

[定義 3] 固定節点・自由節点

3次スプラインセグメントの節点は，節点を移動する前は，すべてデータ点に一致している．「うねり」除去の目的で節点の位置を移動する際に，動かす必要のない節点を固定節点，移動により，データ点と一致しなくなる節点を，自由節点と呼ぶ．

与えられたデータ点上にある各節点を，固定節点と自由節点とに分類するアルゴリズムの例を**図4**に示す．図中，d_I とは，$(I-1)$, I, $(I+1)$ 一番目のデータ点* を直線で補間したとき，I 番目のデータ点上に生じる角のことを意味している．また，Q_I は次式で定義される量である．

$$Q_I=\max\{l_{I-1},\ l_I\}/\min\{l_{I-1},\ l_I\}$$
$$l_{I-1}\triangleq|P_{I-1}-P_I|,\quad l_I\triangleq|P_{I+1}-P_I|, \qquad (6)$$

α_I が十分小さいとき，および十分大きいときは，それぞれ曲率が十分大きいとき，および小さいときと考えられるので，前者を自由節点，後者を固定節点とするのが妥当であると考えられる．また，Q_I は連続する3つのデータ点列において，中間にあるデータ点の他の点に対する相対的な位置を表現しており，この値が大きいほど，曲率の大きい部分を発生しやすい．なお，固定節点，あるいは自由節点と判定される Q_I の大きさは角 α_I の関数となる．図4中の各定数は以上の考察のもとに経験的に定めたものであるが，閉曲線に対しては，良好な判定を与えることが見出されている．

このようにして，自由節点であると判定された節点を，データ点上から引き離すと，一般に，この節点を共有する2つのセグメントは，節点を失ったデータ点を通過しなくなる．したがって，得られる補間曲線は，補間の条件に合致しなくなる．このような不都合を排除するため，節曲線という概念を導入する．

[定義 4] 節 曲 線

与えられたデータ点列において，ある1つの節点を除く他の節点がすべてそれぞれのデータ点に一致しているものとする．補間曲線が，すべてのデータ点を通過するという条件のもとで，この節点の位置を変えるとき得られる曲線を，注目しているデータ点の節曲線という．

節曲線の一例を**図5**に示す．この定義から，節曲線の性質として，次の2つが得られる．

[性質 1]

補間曲線と節曲線とは，データ点および節点の2点で互いに交わる．

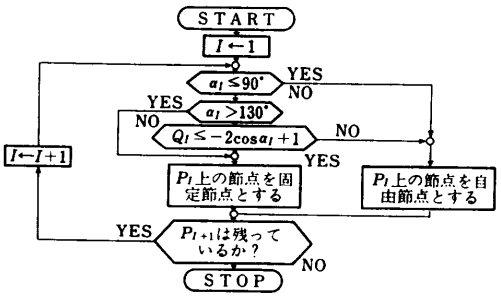

図 4 固定節点・自由節点の分類アルゴリズム
Fig. 4 The algorithm to discriminate between fixed and free knots.

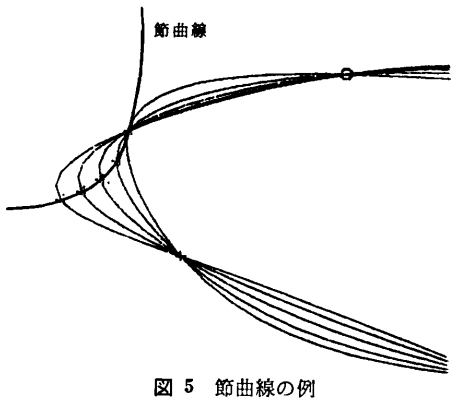

図 5 節曲線の例
Fig. 5 An example of a knot line.

* 本論文では，閉曲線のデータ点列を取り扱っているので，$P_0=P_{n-1}$, $P_{n+2}=P_2$ 等が仮定されている．

[性質 2]

特定のデータ点を通過する節曲線は，ただ1つだけである．

性質1は節曲線の定義から明らかであり，性質2は，3次スプライン曲線の唯一性[11]を用いて，簡単に証明することができる．

ところで，1つの閉曲線データ点列において，本章のアルゴリズムにより自由節点であると判定されるものは，一般に複数個存在する．したがって，定義4を，そのままの形で実際問題に適用することができないように考えられる．しかしながら，スプラインセグメント算出における局所性[12]を用いて，次の2つの性質が証明できる．

[性質 3]

1つの節点の位置を変えることにより，この節点が補間曲線に与える影響は，この節点から離れるにしたがって，指数関数的に減少する．

[性質 4]

1つの節点の位置を変えることにより，各データ点列を通過する節曲線の受ける影響* は，この節点から離れるにしたがって，指数関数的に減少する．

たとえば，自由節点となる2つのデータ点が互いに2単位** あるいは3単位離れていれば，相互の影響は，データ点が隣接している場合と比較して，約25%あるいは約6%程度に減少する[12]．したがって，定義4で与えられている節曲線は，自由節点が隣接あるいは，1つのデータ点を間にはさんで近接しているような特殊な場合を除いて，各データ点ごとに個別に考えてよい．また，この結果として得られる補間曲線は，十分の精度で，すべてのデータ点列を通過することになる．節点曲線を用いた具体的アルゴリズムを次章で与える．

5.「うねり」除去アルゴリズム

提案している方法の主要部の流れ図を**図6**に示す．なお，このアルゴリズムが適用される前処理として，前章の固定節点と自由節点との分類（図4参照）はすでになされているものとする．ここで，図6中の変数の意味は次のとおりである．

（i） U_{max}：各々の隣接するセグメントで定義される「ふくらみ」の差（定義1参照）の絶対値 $U_i(i=1,2,\cdots,N$；N はデータ数）を用いて次のように定義される量である．

$$U_{max} \triangleq \max\{U_1, U_2, \cdots, U_N\}. \qquad (7)$$

（ii） ε：「うねり」が発生しているかどうかを決定する指標（定義2参照）であって，ここでは $\varepsilon=0.01$ としている．

（iii） M：自由節点の個数である．なお，自由節点が存在しない場合（$M=0$）は，本アルゴリズムを適用する必要はない．

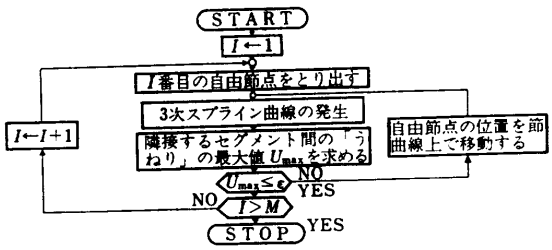

図 6 節点移動のアルゴリズム
Fig. 6 The algorithm to move knots on knot lines.

図6のアルゴリズムを適用する上での1つの問題点は，節曲線の決定にある．性質1および性質2で述べたように，節曲線は，データ点列が与えられれば，一意に決定されるものである．したがって，解析的にこれを求めることができれば，自由節点を通過する節曲線が直ちに求められると同時に，この曲線の形状に基づいた節点移動法を導き出すことができる．しかし，パラメトリックな手法による節曲線の解析的表現は困難であり，現在のところ，この表現は得られていない．そこで，図6のアルゴリズムでは，計算機による収束計算によって節曲線を数値的に取り扱っている．したがって，節点を節曲線上で移動するというのは概念的な表現であって，実際のプログラムでは，離散的な節点を，必要に応じて収束計算によって定めている．つまり，節点を移動すべき場所をまず節点の初期位置とする．そして，この位置を少しずつ変えながら，節点が節曲線上に乗るように収束計算を行っている．

次に，前章の最後で述べた，自由節点が局所的に集中する場合についての対策を考える．性質4およびその下の考察から判断して，隣接する自由節点が3単位以上離れていれば，十分の精度で相互の干渉を無視できる．そこで，自由節点が2単位以内（データ点間隔は一般に均一ではないので，ここでは，自由節点を含むデータ点が隣接，もしくは，1つのデータ点を中に挟んで近接しているという意味である）に集中している場合は，これらを1つの集団であるとみなす．この

* この影響は，補間曲線の節点における導関数で測るものとする．
** データ点間隔がすべて等しいと仮定した場合の隣接データ点間隔を1単位としている．

集団の中で，前章の Q_I が最も小さいものを，最も自由節点とする必要のあるものとし，この集団の自由節点とする．この手続きにより得られる自由節点を通過する節曲線は，定義1にしたがって求めることができる．以上のような理由から，前章のアルゴリズムを適用する際は，このような手続きが含まれているものと仮定する．

6. 適 用 例

本論文で提案しているアルゴリズムの適用例として，簡単な立体図形のトーラスを取り上げる．すなわち，

$$x_{ij}=\{g-h\cdot\cos(2\pi i/K)\}\cos(2\pi j/L),$$
$$y_{ij}=\{g-h\cdot\cos(2\pi i/K)\}\sin(2\pi j/L),$$
$$z_{ij}=h\cdot\sin(2\pi i/K)+\bar{h}\cdot\sin(\bar{h}_l\cdot 2\pi j/L),$$
$$i=1,2,\cdots,K;\ j=1,2,\cdots,L, \qquad (8)$$

を考える．実際例では $g=3.5$, $h=1.5$, $K=8$, $L=16$ としている．このトーラスを極座標 $(r, \theta, \phi)=(500, 10, 30)$ の位置に両眼の中心におき，両眼の間隔を6として得られる2つの視点から観察する．ただし，θ, φ の単位は度とする．また，2つの透視平面は点 (0, 0, 0) (物体の中心) を含み，この点とそれぞれの眼とを結ぶ線分に垂直となるように設定する．このようにして得られる透視平面上に(8)式で表現されるデータ点を投影し，それぞれの平面で2次元補間を行う．**図7**および**図8**は，$\bar{h}=\bar{h}_l=0$, すなわち，通常のトーラスについて，スプライン補間および提案しているアルゴリズムを適用した結果を示す．また，**図9**および**図10**は，これら2つの方法を，変曲点を含むデータ点列（ここでは $\bar{h}=0.5$, $\bar{h}_l=3$ としている）に用いた例を示している．図7，図8，図9，および図10中の2つの像を，左右両眼で，それぞれ独立に観察すれば，図8および図10からは，もとの図形に対応した立体感が得られる．以上のことにより，本アルゴリズムは，平面図形としても，また，立体感のある図形としても，視覚的に自然な補間を与えていることが理解できる．

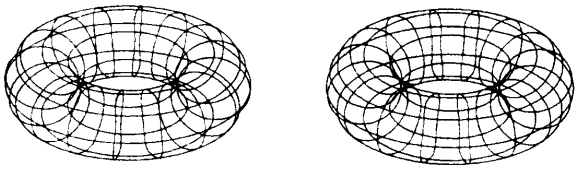

図 7 3次スプライン補間による線画
Fig. 7 Line drawing representation by using cubic spline interpolation.

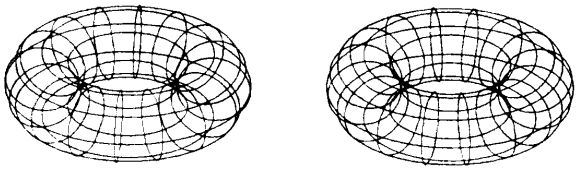

図 8 節点移動処理を施した後の線画
Fig. 8 Line drawing representation by applying the proposed free knot method.

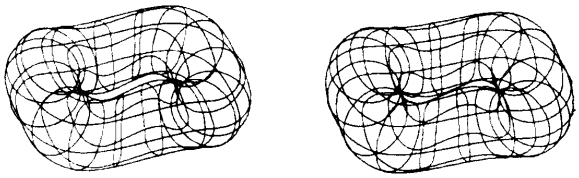

図 9 3次スプライン補間による線画
Fig. 9 Line drawing representation by using cubic spline interpolation

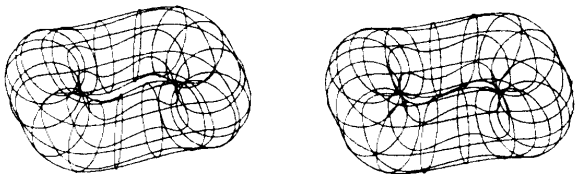

図 10 節点移動処理を施した後の線画
Fig. 10 Line drawing representation by applying the proposed free knot method.

7. あ と が き

本論文で提案している節点移動による「うねり」除去の方法が，閉曲線による線画表示にきわめて有効であることを示した．しかしながら，この方法は，変曲点を境に，曲線の形状が大きく変化している場合は適用に問題がある．そもそも，本論文中の「ふくらみ」の定義は，補間曲線の直線補間からの偏差の最大値のみを用いて定式化したものである．したがって，変曲点を含まない曲線は，この偏差が極大となる位置が1つであり問題ないが，変曲点を1つ含めば，この極大値が2つとなり，最大偏差のみを用いて「うねり」除去を考えることには疑問がある．さらに，変曲点を含む場合は，$L\to\infty$ となりやすい点も問題である．これらの問題点とともに，節曲線の解析的な取り扱いは現在検討中である．

参考文献

1) Myers, W.: Computer Graphics: Reaching the User, Computer Vol. 14, No. 3, pp. 7-17 (1981).
2) Okoshi, T.: Three-Dimensional Imaging Techniques, Academic Press (1976).
3) Mckengie, J., Elton, L. R. B., and Lewis, R.,: Interactive Computer Graphics in Science Teaching, John Wiley and Sons (1978).
4) Rogers, D. F. and Adams, J. A.: Mathematical Elements for Computer Graphics, McGraw-hill, (1976).
5) 原田，河村，中前：3次スプライン曲線の「うねり」除去に関する一手法，昭和 56 年前期情報処理学会全国大会，2F-7 (1981).
6) de Boor, C.: A Practical Guide to Splines, Springer-Verlag. (1978).
7) 吉本：自由節点のスプライン関数を用いた補間について，情報処理学会論文誌，Vol. 22, No. 4 pp. 304-311 (1981).
8) 市田，吉本：スプライン関数とその応用，教育出版 (1979).
9) 秦野：スプライン関数，情報処理，Vol. 22, No. 1, pp. 19-27 (1981).
10) Nielson, G. M.: Some Piecewise Polynomial Alternatives to Splines under Tension, in Computer Aided Geometric Design (R. E. Barnhill and R. F. Riesenfeld eds.), Academic Press, (1974).
11) 研野：自動設計法，コロナ社 (1969).
12) Harada, K. and Nakamae, E.: Local Curve Fitting Procedures Using Cubic Splines, J. Inst. Electron and Comm. Japan Vol. E64, No. 5, pp. 309-313 (1981).

(昭和 56 年 6 月 22 日受付)
(昭和 56 年 11 月 18 日採録)