

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2006-185444

(P2006-185444A)

(43) 公開日 平成18年7月13日(2006.7.13)

(51) Int. Cl.	F I	テーマコード (参考)
G06N 3/00 (2006.01)	G06N 3/00 550Z	5B046
G06F 17/50 (2006.01)	G06F 17/50 620A	5B050
G06T 17/40 (2006.01)	G06F 17/50 622A	
	G06T 17/40 A	

審査請求 未請求 請求項の数 13 O L (全 15 頁)

(21) 出願番号	特願2005-369232 (P2005-369232)	(71) 出願人	503113186
(22) 出願日	平成17年12月22日 (2005.12.22)		ホンダ リサーチ インスティテュート
(31) 優先権主張番号	04030832.2		ヨーロッパ ゲーエムペーハー
(32) 優先日	平成16年12月27日 (2004.12.27)		Honda Research Institute Europe GmbH
(33) 優先権主張国	欧州特許庁 (EP)		ドイツ連邦共和国 デー63073 オ
(31) 優先権主張番号	05021063.2		ッフエンバッハムマイン カール・レギ
(32) 優先日	平成17年9月27日 (2005.9.27)		エン・シュトラーセ 30
(33) 優先権主張国	欧州特許庁 (EP)	(74) 代理人	110000246
			特許業務法人オカダ・フシミ・ヒラノ
		(72) 発明者	マルクス・オルフォファー
			ドイツ国63500セリゲンシュタット、
			シュタインハイマー・シュトラーセ 91

最終頁に続く

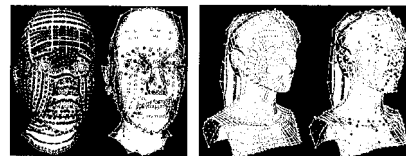
(54) 【発明の名称】 進化的最適化方法および自由形状変形方法

(57) 【要約】

【課題】改善された最適化技法を提供する。

【解決手段】本発明は、設計を最適化するための方法を提供する。設計は変換関数によって変更される。スプライン関数によって変換関数を定義する自由形状変形が用いられる。スプライン関数は、設計が定義される空間を、新しい変更された設計が定義される第2空間に変換する。設計処理中に、スプライン関数のパラメータが、変換関数のパラメータを決定する。自由形状変形は、進化的計算の表現として使用される。

【選択図】図5



【特許請求の範囲】**【請求項 1】**

設計を最適化する方法であって、

前記設計は、変換関数を使用して変更され、前記設計が定義される空間を前記変更された設計が定義される別の空間に変換するスプライン関数によって前記変換関数が定義される自由形状変形が使用されており、

前記スプライン関数のパラメータは、前記変換関数のパラメータを変更し、

前記自由形状変形は、進化的計算の表現として使用される、方法。

【請求項 2】

前記スプライン関数の前記パラメータは、進化的最適化アルゴリズムのパラメータも定義する、請求項 1 に記載の方法。 10

【請求項 3】

前記進化的計算の突然変異演算子は、前記スプライン関数の制御点を導入または削除するために、問題の表現を直接に操作する、請求項 1 または 2 に記載の方法。

【請求項 4】

T スプラインが使用される、請求項 3 に記載の方法。

【請求項 5】

進化的計算は、進化戦略であり、または

進化的計算は、遺伝的アルゴリズムである、

請求項 1 に記載の方法。 20

【請求項 6】

自由形状変形で使用される変換の適合を伴う進化的計算と組み合わせられた自由形状変形方法。

【請求項 7】

進化的計算は、進化戦略であり、または

進化的計算は、遺伝的アルゴリズムである、

請求項 6 に記載の方法。

【請求項 8】

パラメータ適合用の進化的計算と組み合わせられた変換として T スプラインを使用する自由形状変形方法。 30

【請求項 9】

進化的計算は、進化戦略であり、または

進化的計算は、遺伝的アルゴリズムである、

請求項 8 に記載の方法。

【請求項 10】

パラメータ適合用の進化的計算と組み合わせられた変換としての T スプラインと、さらに点挿入および削除による T スプライン構造のオンライン適合と、を使用する自由形状変形方法。

【請求項 11】

進化的計算は、進化戦略であり、または 40

進化的計算は、遺伝的アルゴリズムである、

請求項 10 に記載の方法。

【請求項 12】

請求項 1 乃至請求項 11 のいずれかに記載の方法を実行する計算手段を含むシステム。

【請求項 13】

計算デバイスで実行される時に請求項 1 乃至請求項 12 のいずれかに記載の方法を実施するコンピュータプログラム。

【発明の詳細な説明】**【技術分野】**

【 0 0 0 1 】

本発明は、設計最適化および進化的計算を用いた自由形状変形に関する。

【 背景技術 】

【 0 0 0 2 】

設計最適化問題に最適化アルゴリズムを適用するためには、最適化アルゴリズムによって変更できるパラメータの組によって設計の形態を記述する形を見つけることが必須である。周知の方法は、たとえば、最適化される物体の表面を記述する、2次元表面の場合にスプライン曲線、3次元設計の場合にスプライン曲面を定義することである。この場合に、最適化アルゴリズムによって変更されるパラメータは、制御点およびノット点 (knot point) であり、さらに、NURBS (non-uniform rational B-spline、非一様有理式 B スプライン) の場合に、制御点の重みがある。所与の設計を表すもう1つの形が、基礎設計が与えられる空間を変換する変換関数を定義することである。この場合に、形態の変形は、変換関数のパラメータを変更することによって実現することができる。この方法を、自由形状変形 (free form deformation: FFD) という用語の下で要約することができる。

【 0 0 0 3 】

従来技術：自由形状変形

したがって、自由形状変形は、従来技術で既知であり、次で短く説明する。

【 0 0 0 4 】

拘束付き変形

この原理は、BorrelおよびBechman (非特許文献2) によって定式化された。彼らは、任意の個数のユーザ指定の点の変位拘束 (displacement constraint) によって定義される一般変形方式を開発した。BorrelおよびBechmannの変形方法を、図1に示すが、この方法は、次の原理に基づく：原空間および変形された空間 R^n は、より高次元の空間 R^m の2つの射影である。したがって、変形は、オリジナル空間 R^n の点を R^m の点に変換する関数 $f: R^n \rightarrow R^m$ と、 R^m から R^n への射影 T の合成によって定義される。射影行列は、原空間の所与の点 (拘束点 (constraints points) と呼ぶ) の所与の変位 (拘束 (constraints) と呼ぶ) を達成するように計算される。

【 0 0 0 5 】

数学的に言うと、 R^n の任意の点に対する変換を表す変形関数 d は、関数 $f: R^n \rightarrow R^m$ ($m > n$) と線形変換 $T: R^m \rightarrow R^n$ の合成として表される。 $U \in R^n$ であり、 M が T の対応する行列である場合に、

$$d(U) = M f(U)$$

が得られる。

【 0 0 0 6 】

f および中間次元 m の選択によって、図1からわかるように、可能な変形の組が定義される。

【 0 0 0 7 】

異なる関数 f は、異なるタイプの変形、たとえばグローバル変形またはローカル変形を作る。具体的な変形は、行列 M だけに依存する。対応する変換 T は、潜在的な変形の集合から「適切な」変形を選択する。 n に対して m が大きいほど、可能な変形の範囲が広がる。

【 0 0 0 8 】

一般に、「このモデルは、可能な変形の大きな一群をカプセル化する：たとえば、線形空間変換は、 f が線形の時に得られ、FFDは、 f がテンソル積ベルンシュタイン多項式であり、 M が制御点の変位ベクトルを用いて作られる場合に得られる」。

【 0 0 0 9 】

f および m が指定されたならば、適切な M およびその結果として任意の点 $U \in R^n$ の変換が、 n_c 個の拘束点 V_i の助けを得て、3ステップで実行される。

【 0 0 1 0 】

1. n_c 個の拘束点 V_i 、 $i \in [1, n_c]$ の所望の変位を選択する。これらの点は、本質的に、設計最適化の自由度を確立する。

【0011】

2. 射影行列 M は、 n_c 個の式の n 個の系を解くことによって得られ、 m 個の未知数が M の 1 行を構成する。この系は、任意の $i \in [1, n_c]$ に対する $n \times n_c$ 個の式

$$d(V_i) = M f(V_i)$$

から導出される。

【0012】

残念ながら、この連立方程式は、どの場合にも一意に解くことが可能ではなく、これが、ここで説明する技法の主要な短所を構成する。次の 3 つの状況を区別することができる。 10

【0013】

a) m が n_c より大きく、すべての $f(V_i)$ ベクトルが線形独立である場合に、無限の個数の解が存在する。ユーザが 1 つの M を選択することによって、変形が固定される。それを行う複数の形が、Borrel および Bechmann (非特許文献 2) によって議論されている。

【0014】

b) $m = n_c$ の場合に、1 つの解 M だけが存在する。

【0015】

c) それ以外の場合に、拘束を満足する変形を見つけることはできない。この場合に、最良の近似解が使用される。 20

【0016】

3. すべての点 U の変位を、 $d(U) = M f(U)$ によって計算する。

【0017】

再変換 M が、拘束点 V_i だけに依存するという事実に起因して、最初のオブジェクト U (点 U_1 の集合) の変形は、非常に効率的に実行することができる。値 $f(U_1)$ は、一回だけ計算する必要があり、その後、拘束の複数の組によって定義される複数の変形に使用することができる。その結果、設計最適化は、追加、削除、または変更された拘束についてのみ $f(V_i)$ の値の再計算を必要とし、オブジェクト点 U_1 については不要である。 30

【0018】

既に指摘したように、 M の選択処理は、技術的に要求が厳しく、ユーザとのインタラクションを必要とする。さらに、変形の形態は、拘束に強く相関せず、これによって、この技法が非直観的になっている。これらの不足を迂回するために、Borrel および Rappoport (非特許文献 3) は、一般拘束変形概念を単純化した。この手法は、単純化拘束変形 (Simplified Constraint deformation: Scodef) と呼ばれる。Scodef 手法の中心概念は、再変換 M の統一であり、それによる処理全体の単純化である。これは、それぞれがある拘束点を中心とする B スプライン基底関数 (basis function) のテンソル積としての f の選択によって達成され、各点は、変形処理に影響する。さらに、 n_c 個の拘束点ごとに、各 B スプラインのスコープに影響する半径が導入され、すなわち、各基底関数は、対応する拘束点の半径の外で 0 になる。したがって、異なる半径を、変形処理の微調整に使用することができる。その結果、「Scodef 変形は、空間で任意の個数の可能なオーバーラップする B スプラインの形の「バンプ (bump)」を作成することによって得られる変形と見なすことができる。バンプの位置および高さは、拘束によって定義され、その幅は、影響の拘束半径によって定義される」。 40

【0019】

すべてのタイプの形態を拘束変形手法によって表現できるかどうか (完全性要件 (completeness requirement)) は文献からは不明瞭であるが、B スプラインの使用は、この技法が、B スプラインのように複雑な曲線を効率的かつ正確に表現できることの強い証拠を与える。

【 0 0 2 0 】

E Sを使用する設計最適化について、拘束および/または拘束の変位は、E S標準の遺伝的操作 (genetic operator) の助けを得て変更できる自由パラメータを構成する。単純化拘束変形の場合に、 n_c 個の拘束点 V_i に関連する半径を、追加の設計パラメータとして使用することができる。その結果として、3つのグループの設計パラメータが使用可能であり、これにより設計パラメータの数においてこの技法が非常に柔軟になる。最良の場合に、4つの拘束点だけが、3D物体の変形に必要である。さらに、 n_c 個の式の基礎になる系に起因して、拘束点、その変位、および半径の3つ組を、他の設計パラメータに影響せず、たやすく追加し、または削除することができる。もちろん、そのような適応処理は、変換行列 M の構造を変更することができ、したがって、結果の設計を変更することができる。これは、たとえば、対照的に制御点の再計算を強制する、加算演算子を保存する面の曲線が存在するBスプラインまたはNURBSと対照的である。

【 0 0 2 1 】

自由形状変形

より以前に開発されたが、自由形状変形 (FFD) 手法 (非特許文献9) (図2も参照されたい) は、上で説明した拘束変形手法の特殊事例である。これは、変形関数 f が、ベルンシュタイン多項式 (Bernstein polynomials) のテンソル積 (または、単純化拘束変形の場合にBスプライン基底関数) であり、 M が、制御点の変位ベクトルを用いて作られる場合に当てはまる。拘束変形に非常に似て、FFDは、より一般的な対話型の形態編集概念の形成に焦点を合わせたものである。SederbergおよびParry (非特許文献9) は、ベジエ、Bスプライン、およびNURBSを用いて作られた曲線または曲面の変形の正確な指定が、ほとんどの場合に面倒すぎると考えた。「知覚的に単純な変更であっても、多数の制御点の調整を必要とする場合がある」 (非特許文献5)。

【 0 0 2 2 】

SederbergおよびParryは、立方体または円筒などの標準的な幾何形態のグリッド点の格子に物体を埋め込むことによって、規則的な変形 (固体の R^3 から R^2 への写像) に対する手法を一般化した (非特許文献1)。グリッドのノードの操作によって、3変数テンソル積ベルンシュタイン多項式の助けを得て、グリッドの内部の空間に対する変形が誘導される。これらの変形は、物体を形成する基礎になるグラフィックス・プリミティブを変換する。SederbergおよびParryは、FFDのよい物理的類似を与えた。「変形が望まれる1つまたは複数の物体が組み込まれた透明で柔軟なプラスチックの平行六面体を検討されたい。物体も柔軟であると考えられ、その結果、物体は、その周囲のプラスチックに似た形で変形する」 (非特許文献9)。

【 0 0 2 3 】

既に指摘したように、FFDの最初のステップは、囲む平行六面体領域の局所座標系 S 、 T 、 U の固定である。ある基本的な線形代数の助けを得ると、平行六面体の各点 X を、次のように新しい座標系で表すことができる。

【 0 0 2 4 】

$$X = X_0 + s S + t T + u U$$

FFDのアイデアは、囲むボリュームの変形が、そのボリュームの内部の変形を誘導することなので、次のステップは、平行六面体の制御点 $P_{i,j,k}$ のグリッドの形成である。 S 方向で l 個、 T 方向で m 個、 U 方向で n 個のグリッド点があり、その制御点がグリッド点である3変数テンソル積ベルンシュタイン多項式の助けを得て、変形された囲むボリュームの定式化を可能にする。潜在的な設計最適化の自由パラメータを本質的に作る制御点の変形は、単に、その変位していない格子位置から $P_{i,j,k}$ を移動することによって指定される。任意の点 X の変形された位置 X_{ff} は、まず局所座標系でその (s, t, u) 座標を計算し、次にベクトル値を有する3変数ベルンシュタイン多項式

【数 1】

$$X_{ffd} = \sum_{i=0}^l \binom{l}{i} (1-s)^{l-i} s^i \left\{ \left(\sum_{j=0}^m \binom{m}{j} (1-t)^{m-j} t^j \left\{ \sum_{k=0}^n \binom{n}{k} (1-u)^{n-k} u^k P_{ijk} \right\} \right) \right\}$$

を評価することによって見つかるが、ここで、 X_{ffd} は、変位された点のデカルト座標であり、各 P_{ijk} は、制御点のデカルト座標を含むベクトルである。ほとんどの他の設計表現技法と異なって、（単純化）拘束変形手法の主な動機づけは、形態設計者のための直観的な設計ではない。そうではなく、基礎になる幾何表現に依存しない形態変更ツールが、主な関心事である。 10

【0025】

拡張型自由形状変形（E E F D）

Sederberg他によって導入された基本的な自由形状変形（Extended Free Form Deformation: E E F D）と呼ばれる方法を開発した（非特許文献1）。この方法では、標準的な自由形状変形の制御点の平行六面体ボリュームという制限的要件と異なって、任意の制御ボリュームの使用が可能である。この変更のゆえに、物体を変形する手順が、適合されなければならない、数学的基礎が、Bスプラインに切り替えられなければならない。これは、主に、下で短く説明する3つのステップに分かれる。 20

【0026】

まず、物体全体または物体のうちで変形について対象とされている部分のいずれかを囲む制御点の格子（lattice）を構成しなければならない。第2ステップで、幾何座標を、格子のパラメータ空間に変形しなければならない。これを、「フリージング（freezing）」とも呼ぶ。物体のフリージングによって、スプライン・パラメータ空間での幾何形態の u 、 v 、および w 座標が計算される。これは通常、最も高速の手法である見込みがあるニュートン近似によって行われるが、類似する傾きに基づく方法または進化的最適化（evolutionary optimisation）によって行うこともできる。 u 、 v 、および w 座標をうまく計算した後に、制御点を所望の位置に移動し、表面または固体の x 、 y 、および z 座標を更新することによって、変形を適用することができる。したがって、Bスプライン式を、制御点の新しい空間座標を使用して解かなければならない。この方法は、幾何モデルが固体と等しいか、助変数方程式または暗黙の式によって定義される場合にも有効である。 30

【非特許文献1】A.H. Bar, "Global and Local Deformation of Solid Primitives", Computer Graphics, 18(3):21-30, 1984.

【非特許文献2】P. Borrel and D. Bechmann, "Deformation of n-dimensional objects", International Journal of Computational Geometry Applications, 1(4):427-453, 1991.

【非特許文献3】P. Borrel and A. Rappoport, "Simple Constrained Deformations for Geometric Modeling and Interactive Design", ACM Transactions on Graphics, 13(2):137-155, April 1994. 40

【非特許文献4】S. Coquillart, "Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling", Computer Graphics, 24(4):187-196, August 1990.

【非特許文献5】S. Gibson and B. Mirtich, "A Survey of Deformable Modeling in Computer Graphics", Tech. Report No. TR-97-19, Mitsubishi Electric Research Lab., Cambridge, November 1997.

【非特許文献6】Markus Olhofer, Yaochu Jin and Bernhard Sendhoff, "Adaptive encoding for aerodynamic shape optimization using evolutionary strategies", Congress on Evolutionary Computation (CEC), volume 2, pages 576-583, Seoul, Korea, May 2001. IEEE Press.

【非特許文献7】L. Piegl and W. Tiller, "The NURBS BOOK", Springer-Verlag Berl 50

in Heidelberg 1995 and 1997.

【非特許文献 8】T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng and T. Lyche, “T-spline Simplification and Local Refinement”, ACM Transactions on Graphics, 23(3):276-283, August 2004.

【非特許文献 9】T.W. Sederberg and S.R. Parry, “Free-Form Deformation of Solid Geometric Models”, Computer Graphics, 20(4):151-160, August 1986.

【非特許文献 10】T.W. Sederberg, J. Zheng, A. Bakenov and A. Nasri, “T-splines and T-NURCCs”, ACM Transactions on Graphics, 22(3):477-484, July 2003.

【発明の開示】

【発明が解決しようとする課題】

10

【0027】

本発明の目的は、改善された最適化技法を提案することである。

【課題を解決するための手段】

【0028】

この目的は、独立請求項の特徴によって達成される。従属請求項は、さらに、本発明の中心概念を展開するものである。本発明は、本発明による表現として自由形状変形を使用する進化的計算 (Evolutionary Computation) を提案する。

【0029】

自由形状変形と進化的計算の組合せは、本発明の 1 態様である。

【0030】

20

自由形状変形での本発明のもう 1 つの態様によれば、変形が、問題に適合される。

【0031】

本発明のもう 1 つの態様によれば、T スプラインが、自由形状変形の変換関数としての進化的計算と組み合わせて使用される。

【0032】

本発明のさらなる目的、特徴、および長所は、添付図面を参照して実施形態の次の説明に進む時に明白になる。

【発明を実施するための最良の形態】

【0033】

進化的計算での (拡張型) 自由形状変形

30

進化的最適化に関して、自由形状変形技法の適用は、特に、最適化されなければならない形態の幾何定義がかなり複雑である場合および / またはコストの高い計算流体力学 (Computationally Fluid Dynamics: CFD) 計算または有限要素法 (Finite Element Method: FEM) 計算を実行しなければならない場合に、多数の利益を特徴とする。自由形状変形と、最適化の目標にされているパラメータとの一体化は、形態幾何自体によって定義されるのではなく、格子の制御点によって定義される。その結果、設計者は、グローバル形態変更またはローカル形態変更の度合を選択することができる。

【0034】

一般に、最適化ループを開始できるようになる前に、初期形態を格子に埋め込まなければならない。幾何形態をフリーズして、パラメータ空間内の u 、 v 、および w 座標を入手しなければならない。これらの座標は、進化的最適化の設計評価ステップの前の遺伝子型 - 表現型写像 (genotype-phenotype mapping) で主要な役割を演ずる。パラメータとしての制御点に基づいて、突然変異を適用することができ、適応度評価のためにシステムを準備することができる。したがって、新しい設計は、新しい制御点と結合されるフリーズされた u 、 v 、 w 座標に基づく B スプライン式または Nurbs 式を介して計算される。その後、設計を評価することができ、適応度の値が、各個々に割り当てられる。最後に、新しいパラメータを選択し、次世代の親を、選択された進化的アルゴリズムに従って決定する。

40

【0035】

CFD 評価または FEM 評価と結合された (拡張型) 自由形状変形の効果

50

C F DまたはF E Mと組み合わせられた自由形状変形の多数の利益を理解するために、制御点の位置の変更が、格子のある空間領域に強い影響を有することを考慮しなければならない。サイズがたとえばスプラインのタイプ、スプライン次数、制御点の個数、ノットベクトル (knot vector) などによって定義されるこのボリューム全体が変形される、すなわち、このボリューム内で見つかる形態設計だけではなく、それを囲むすべてのものが変形される。C F D計算に関係する結果として、このボリューム内に位置するすべてのグリッド点も変形される。したがって、複雑な形態に関して、計算的にコストの高いメッシュ生成処理を省略することができる。というのは、自由形状変形が適用される時に、グリッドが、変形された形態に直接に適合されるからである。第2に、C F Dに関して、時間のかかるC F D計算全体を繰り返すのではなく、既に解に収束している、以前の結果を使用することも可能である。 10

【0036】

(拡張型)自由形状変形のパラメータ適応性

自由形状変形のこの拡張は、本発明のさらなる開発に特に有利である。すべての種類の最適化において、適用される表現の高い度合の柔軟性が有利である。設計者は、常に、検索空間とパラメータ最小化の間の適当なトレードオフを見つけることを強いられる。一方で、できる限り多くの設計を表現しなければならない、他方で、少数のパラメータを使用して最適化処理を高速化しなければならない。柔軟性に対するよい代替案は、適応的な表現によって提供される。その結果、最適化の開始時に、少数のパラメータを最適化するだけでよく、収束の過程で、パラメータの個数を増やすことができる。 20

【0037】

自由形状変形は、設計処理で実際の形態を変更せずに、さらなる制御点を追加することによってノットベクトルを拡張する可能性を提供する。この事実は、進化的設計最適化において非常に重要である。非常に少数のパラメータだけを用いて最適化を開始し、必要な時にいつでも、新しい制御点を追加して、実際の最良の設計を失わずに形態に対する影響を拡張することができる。この表現のこれらの非常に前向きの特徴のほかに、制御点挿入に、1つの主要な不利益が付随する。新しい制御点を挿入しなければならない時に、必ず、2次元の場合に制御点の新しい列または行全体、3次元の場合に制御点の平面を制御点グリッドに導入する可能性だけが存在する。したがって、残念ながら、幾何学的理由から直接に望ましい場合に1つの制御点だけを挿入することは不可能であり、結果として、多数の余分な点を考慮に入れなければならない。この影響を最小にするために、Tスプラインという概念を、以下に説明する。Tスプラインとは、制御点の部分的な行または列だけを制御メッシュに挿入することを可能にする特別な種類の点ベースのBスプラインである。 30

【0038】

Tスプライン

したがって、Tスプラインの概念は、たとえばコンピュータ・グラフィックスに関して、既に既知の表現である。

【0039】

従来技術から既知のTスプラインの導入

Tスプラインの概念は、グラフィカル設計の処理でのより高い柔軟性を達成するために、Sederberg他によって導入された(非特許文献1、非特許文献10)。Tスプラインは、基礎になるメッシュでの制御点の配置の指定に関する限りない利益を提供する、不均一Bスプライン曲面の一般化である。水平線および垂直線の各交点で定義される制御点の長方形グリッドを必要とするBスプライン曲面と異なって、Tメッシュの線は、グリッドの端で始まり、終わる必要はなく、T字型接続点で終わることができる(たとえば、図4)。その結果、制御点の総数が劇的に減る。この事実は、より柔軟で効率的な曲面設計を可能にするために1つまたは複数の追加制御点を挿入しなければならない場合に、さらに重要になる。Bスプライン曲面では、制御点の完全な新しい行および/または列を設けなければならないが、Tメッシュでは、ほとんどの場合に、少数の新しい点を追加するだけで 50

よい。

【 0 0 4 0 】

ノット挿入 (knot insettion) の手順について、Sederberg他は、後で詳細に説明するローカル洗練アルゴリズム (local refinement algorithm) を開発し、次いで、「Tスプライン単純化 (T-spline Simplification)」と呼ばれる、余分な制御点を除去する概念を開発した。除去アルゴリズムの適用によって、設計形態がBスプラインを介して定義されている場合に必要な制御点の総数を減らすことが可能になる。

【 0 0 4 1 】

図5に、NURBSモデルのTスプラインへの変換の効果を示す。モデル化された頭の制御点の総数は、4712個から1109個に減り、女性の場合に10305個から3955個に減っている。その結果、設計者は、モデルに対するより効率的な制御を得、単一の制御点の位置の移動が、形態に対するより密接な影響を有する。さらに、Tスプラインの概念は、2次元での曲面モデリングに有効であるだけでなく、他の次元およびTスプラインの任意トポロジ版であるT-NURCC (Non-Uniform Rational Catmull-Clark Surface、非一様有理式カットマル-クラーク局面) に簡単に拡張することができる (非特許文献1、非特許文献8、非特許文献10)。

【 0 0 4 2 】

Tスプラインの数学的定義および規則

Tスプラインは、Sederberg他 (非特許文献10) によって導入されたいわゆる点ベースBスプライン (point based B-splines、PBスプライン) から導き出すことができる。PBスプラインは、制御点の組と、2次元で制御点ごとのノットベクトルの対によって定義される。結果の面の点を計算するために、次の式を解かなければならない。

【数2】

$$P(s, t) = \frac{\sum_{i=1}^n P_i B_i(s, t)}{\sum_{i=1}^n w_i B_i(s, t)}$$

【 0 0 4 3 】

$B_i(s, t)$ は、ブレンディング関数であり、次式のように定義される。

【 0 0 4 4 】

$B_i(s, t) = N[s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}](s) N[t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}](t)$
 $N[s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}](s)$ および $N[t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}](t)$ は、それぞれ、一般形状で次式によって与えられる3次Bスプライン基底関数である。

【数3】

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

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

【 0 0 4 5 】

u_i は、ノットベクトルのノットであり、 p は、スプラインの次数を定義する。 B_i の計算のために、基底関数式を、短いノットベクトル $s_i = [s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}]$ および $t_i = [t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}]$ を使用して評価しなければならない。このベクトルのそれぞれが、制御点 P_i に直接に接続されている。重み w_i は、自由に選択することができる。すべての w_i が1.0である場合に、PBスプライン式は次のように単純に

なる。

【数 4】

$$P(s, t) = \frac{\sum_{i=1}^n P_i B_i(s, t)}{\sum_{i=1}^n B_i(s, t)}.$$

【0046】

一般的に定義された P B スプラインと対照的に、T スプラインは、制御点の位置に対するある順序を定義する、いわゆる T メッシュによって構成される。T メッシュの例を、図 6 に示す。 10

【0047】

このメッシュは、制御点 P_i に密接に接続されるノットベクトルの定義に重要である。T メッシュが満足しなければならない次の要件が定式化されている。

【0048】

1. 「すべての面の 1 側面に沿ったすべてのノット間隔 (knot interval) の合計は、対向する側面のノット間隔の合計と等しくなければならない」。ノット間隔は、2 つのノットの間の差と等しく、したがって、たとえば図 6 では d_i および e_i である。

【0049】

2. 2 つの T 字交差点は、接続の後に新しい面が各対向する側面に等しいノットベクトルの合計を有する場合に限って接続されなければならない (非特許文献 8)。 20

【0050】

ノットベクトル s_i および t_i の定義は、ノット間隔に基づき、B スプライン曲面の計算に必要な定義と全く異なる。これは、(非特許文献 8) でルール 1 として示されている。

【0051】

「パラメータ空間内の光線 $R(\square) = (s_{i2} + \square, t_2)$ を検討されたい。すると、 s_{i3} および s_{i4} は、その光線が交差する最初の 2 つの s 辺の s 座標である。 s 辺とは、定数 s の垂直線分を意味する。 s_i および t_i の他のノットは、同様の形で見つかる」。すべてのノットベクトルが判定されたならば、T スプラインを計算することができる。 30

【表 1】

s	N(s)	c	d
$[s_0, k, s_1, s_2, s_3, s_4]$	$c_0 N[s_0, k, s_1, s_2, s_3](s) + d_0$ $N[k, s_1, s_2, s_3, s_4](s)$	$c_0 = \frac{k - s_0}{s_3 - s_0}$	1
$[s_0, s_1, k, s_2, s_3, s_4]$	$c_1 N[s_0, s_1, k, s_2, s_3](s) + d_1$ $N[s_1, k, s_2, s_3, s_4](s)$	$c_1 = \frac{k - s_0}{s_3 - s_0}$	$d_1 = \frac{s_4 - k}{s_4 - s_1}$
$[s_0, s_1, s_2, k, s_3, s_4]$	$c_2 N[s_0, s_1, s_2, k, s_3](s) + d_2$ $N[s_1, s_2, k, s_3, s_4](s)$	$c_2 = \frac{k - s_0}{s_3 - s_0}$	$d_2 = \frac{s_4 - k}{s_4 - s_1}$
$[s_0, s_1, s_2, s_3, k, s_4]$	$c_3 N[s_0, s_1, s_2, s_3, k](s) + d_3$ $N[s_1, s_2, s_3, k, s_4](s)$	1	$d_3 = \frac{s_4 - k}{s_4 - s_1}$

40

【0052】

ノット挿入 - ローカル洗練アルゴリズム

T スプラインの主要な利益の 1 つが、制御点の新しい行または列全体を挿入する必要なしに、T メッシュ内の任意の位置に新しい制御点を挿入する可能性である。Sederberg 他 50

は、成功したノット挿入に関するルールおよびアルゴリズムを含む手順を開発した。挿入処理は、トポロジ・フェーズおよびジオメトリ・フェーズと呼ばれる２フェーズで行われる。トポロジ・フェーズでは、追加して挿入されなければならない制御点を識別し、ジオメトリ・フェーズでは、新しいＴメッシュの新しい座標および重みを計算する。

【００５３】

Sederberg他（非特許文献８）は、挿入処理中に発生する可能性があり、追加制御点を挿入することによって訂正されなければならない３つの可能な違反を示した。

【００５４】

「違反１：ブレンディング関数（blending function）が、現在のＴメッシュについてルール１によって指定されるノットを含まない。

10

【００５５】

違反２：ブレンディング関数が、現在のＴメッシュについてルール１によって指定されていないノットを有する。

【００５６】

違反３：制御点が、それに関連するブレンディング関数を有しない。」

まず、すべての制御点を、より多くの制御が望まれる位置に挿入する。グリッド検査を行い、違反が発生している場合に追加の点を挿入する。すべての違反を解決した後に、新しい座標を、線形変換によって計算する。

【００５７】

本発明による適応表現

20

Ｔスプラインの概念と進化的最適化の組合せが、本発明のさらなる態様である。

【００５８】

進化的最適化でのＴスプライン

上で既に指摘したように、表現の適応的挙動は、最適化問題に関して非常に重要である。パラメータの個数をできる限り少なく保ち、同時にさまざまな形態設計を保証するように努めなければならない。Ｔスプラインは、設計最適化のこの要件に関する非常によいトレードオフを提供する。一般に、パラメータは、Ｔメッシュの制御点に対してセットされる。したがって、非常に少数の制御点から開始し、最適化の過程で、制御点を任意の位置に追加して、分解能を高めることができる。したがって、この表現を適用することによって、Ｔスプラインの高い柔軟性を使用して、制御点を必要な場所に直接に位置決めして、それまでの最善の設計を保ちながら幾何形態に対する最大の影響を達成することができる。これは、ほとんどの場合に、ごく少数の制御点の追加になるので、パラメータの個数が、できる限り少なく保たれる。これは、非常に良い形で最適化に影響する。

30

【００５９】

変換のパラメータが変更の自由度を決定するという事実に起因して、パラメータを、選択されたパラメータによって必要な変更を実行できる形でセットしなければならない。どの種類の最適化の場合でも、「必要な変更」は、事前にはわからず、経験に基づいてのみ知られる可能性がある。一般に自由度を高めるために、自由形状変形の場合に制御点の個数を単純に増やすことができる。これは、制御点の影響をよりローカルにし、設計に対するグローバル変更は、さまざまな制御点を同時に変更しなければならないので、実現がより困難になる。さらに、拘束付き変形の場合に、変更の自由と、制御点の個数または行列Ｍの次元の変更による最適化の次元数の間にトレードオフがある。この関係は、スプラインを使用する線または面の表現でも観察することができる。

40

【００６０】

上で説明したＴスプラインの特性が、その開発の理由である。たとえばＮＵＲＢＳ面スプラインによって面を表現することによって、単一の制御点の追加が、表現によって与えられる拘束を満足するだけのために、追加制御点の行または列全体の追加を必要とする。同一のことが、別の変更された設計への設計の変換を定義する自由形状変形方法に適用された時にもあてはまる。

【００６１】

50

自由形状変形の既存の方法は、制御点のメッシュを定義することを必要とし、これは、設計の間の変換関数を定義する。この自由形状変形の既存方法に対する拡張が、制御点の自動追加または自動除去である。この方法を用いると、設計の最適化中の変換関数の変更を変更できるようになる。所与の変換関数が、十分に高い自由度をもたらさない場合に、制御点を、制御点の既存の組に追加することができる。この追加に起因して、より高い自由度が、変換関数に存在し、所与の品質尺度に関して設計をさらに改善できるより詳細な変形が可能になる。

【 0 0 6 2 】

スプライン（たとえば、面スプライン）に新しい制御点を導入する時に、所与の面の近似の場合に新しい制御点の最適位置を決定するさまざまな方法が存在する。新しい制御点の最適位置を見つけるためのローカル近似方法およびグローバル近似方法が存在する。しかし、これらの方法は、ローカル品質尺度が存在する場合に限って適用可能である。このローカル品質尺度が存在しない時には、これらの方法を適用することができない。これは、たとえば、表された機体の空気力学特性に基づく品質尺度にあてはまる。たとえば、タービン・ブレードの最適化の場合に、非常にしばしば、圧力損失が、品質尺度として使用される。この圧力損失は、設計の本体全体に依存し、追加制御点を挿入しなければならない前に判定することができない。

10

【 0 0 6 3 】

この場合に、変換関数の追加パラメータの導入は、表現の追加の突然変異演算子（mutation operator）によって実現することができる。ランダムな制御点が、変換関数を表すスプラインに追加される。この変更された表現が、進化の既知の処理を受ける。追加制御点が、変換関数の前向きの変動を可能にするという意味で有益である場合に限って、これらが、進化の処理中に選択される。したがって、自由形状変形の所与の方法、制御点の導入の方法、および進化的計算の組合せが、自由形状変形に使用される変換関数を実際の最適化の必要に適合させる洗練された優雅な手法をもたらす。

20

【 0 0 6 4 】

説明された方法によって変更される変換関数としてスプライン関数を使用する既述の方法が、特殊な必要に対する処理中の最適化の自由を高める洗練された優雅な方法である場合であっても、表現の主な短所はまだ存在する。制御点の導入は、表現の拘束を満足するだけのための複数の他の制御点の導入を必要とする。

30

【 0 0 6 5 】

既に説明したように、Tスプラインは、面の表現の場合にこの問題を解決する。NurbsスプラインではなくTスプラインを使用すると、その場合の単一の追加制御点の導入が可能になる。

【 0 0 6 6 】

面モデリングと同一の形で、Tスプラインの使用は、自由形状変形の制御点追加中の問題を解決することができる。拡張型自由形状変形の場合と同一の形で、新しい制御点の導入を、追加の突然変異演算子によって実現することができる。追加の突然変異演算子は、変換のパラメータの代わりに変換の表現を同一の形で変更する。追加制御点の導入に関して最適の位置の選択を可能にする機構は、既に説明したものと同一である。相違と利益は、この方法が、変換関数のローカル拡張を可能にすることである。

40

【 0 0 6 7 】

要約すると、説明した方法は、変換関数によって変更される基本設計を使用する設計の既知の表現に基づく。自由形状変形技法の場合に、変換関数は、設計が定義される空間を、新しい変更された設計が定義される第2空間に変換するスプライン関数によって定義される。設計処理中に、スプライン関数のパラメータが、変換関数のパラメータを決定する。進化戦略を使用している間に、スプライン関数のパラメータは、それと同時に、最適化アルゴリズムが働くパラメータを決定する。変換関数を所与の問題および必要な変換に最適に適合させるために、制御点の導入および削除を可能にする突然変異演算子が提案される。進化戦略での標準的な突然変異演算子ではなく、この突然変異演算子は、問題/設計

50

の表現を操作する。

【 0 0 6 8 】

上の方法にあてはまる短所は、基礎になるスプライン記述の使用によって導入される拘束であり、これは、単一のパラメータの導入が多数の点の導入を必要にすることである。この問題に対する解決策は、スプライン曲面の表現に関して説明され、Tスプラインと呼ばれる。本明細書で、変換関数としてのこの特殊なスプライン・タイプの超次元拡張を使用することを提案した。この特殊な変換関数のパラメータは、進化的計算の枠組みの中で使用することができ、表現を直接に扱う新しい突然変異演算子によって実際の最適化の特殊な必要に適合させることができる。

【図面の簡単な説明】

【 0 0 6 9 】

【図 1】拘束変形の原理を示す図であり、2つの異なる変形された2D曲線を得るために、3D曲線が、オリジナル2D曲線から計算され、2回射影されている。

【図 2】自由形状変形の原理を示す図である。

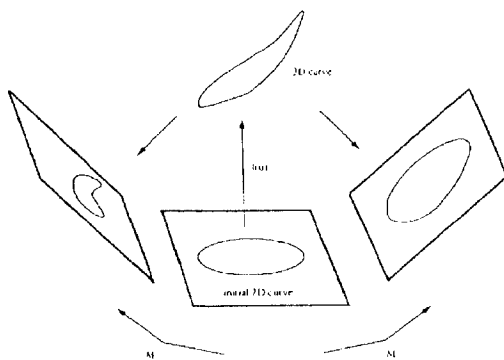
【図 3】EFFDの原理を示す図であり、任意の制御ボリュームの変位前（左）および変更後（右）の制御点を示す図である。

【図 4】Bスプライン・メッシュおよびTメッシュの例を示す図である。

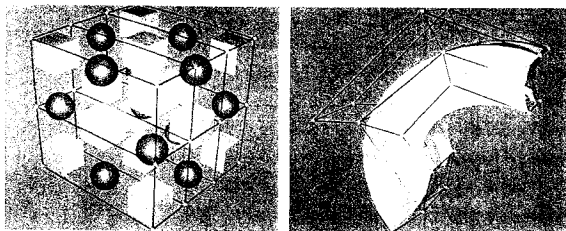
【図 5】Tスプライン単純化の例を示す図である。

【図 6】（左）ブレンディング関数のノット線および（右）Tメッシュを示す図である。

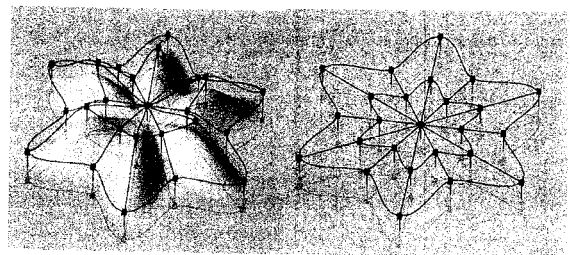
【図 1】



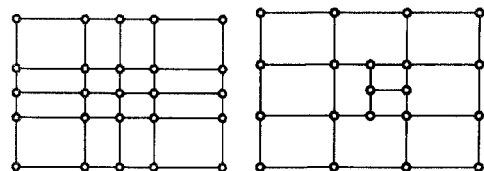
【図 2】



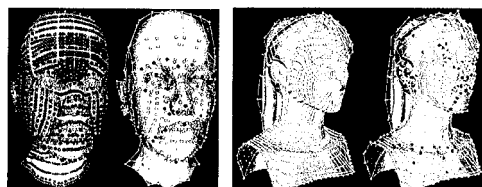
【図 3】



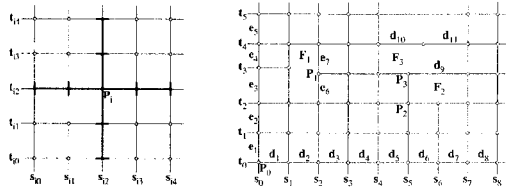
【図 4】



【図 5】



【 図 6 】



フロントページの続き

(72)発明者 ステファン・メンゼル

ドイツ国 6 3 3 0 3 ドライアイヒ、シューベルトシュトラッセ 2 5

(72)発明者 ベルンハルト・センドホフ

ドイツ国 6 3 4 8 6 ブルフコーベル、ウドゥーミュラーリング 7 3

F ターム(参考) 5B046 FA12 FA18

5B050 BA09 BA13 EA28