

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4675042号
(P4675042)

(45) 発行日 平成23年4月20日(2011.4.20)

(24) 登録日 平成23年2月4日(2011.2.4)

(51) Int. Cl.		F I			
G06F 17/50	(2006.01)	G06F 17/50	622A		
G06T 19/00	(2011.01)	G06F 17/50	626C		
		G06T 17/40	A		

請求項の数 2 (全 14 頁)

(21) 出願番号	特願2003-432771 (P2003-432771)	(73) 特許権者	506301140
(22) 出願日	平成15年12月26日(2003.12.26)		公立大学法人会津大学
(65) 公開番号	特開2005-190311 (P2005-190311A)		福島県会津若松市一箕町大字鶴賀字上居合
(43) 公開日	平成17年7月14日(2005.7.14)		90番地
審査請求日	平成18年12月1日(2006.12.1)	(74) 代理人	100091904
前置審査			弁理士 成瀬 重雄
		(72) 発明者	ニコライ ミレンコフ
			福島県会津若松市一箕町鶴賀 会津大学内
		(72) 発明者	廣富 哲也
			福島県会津若松市一箕町鶴賀 会津大学内
		(72) 発明者	成田 祥
			福島県会津若松市石堂町7-79
		審査官	田中 幸雄

最終頁に続く

(54) 【発明の名称】 立体形状物の生成方法

(57) 【特許請求の範囲】

【請求項1】

以下の処理をコンピュータに行わせることを特徴とするコンピュータプログラム：

- (1) 平面形状 f の入力を前記コンピュータが受け付ける処理；
- (2) 第1の方向への奥行きを持ち、さらに、前記第1の方向から前記平面形状 f に見える第1の立体形状データを前記コンピュータの記憶部に保存する処理；
- (3) 前記第1の方向とは交差する第2の方向への奥行きを持ち、さらに、前記第2の方向から前記平面形状 f に見える第2の立体形状データを前記コンピュータの記憶部に保存する処理；
- (4) 前記第1の立体形状と前記第2の立体形状との論理積演算によって、これらの形状を重ね合わせたときにこれらの形状が重複する領域を残し、非重複領域を除去することにより、前記コンピュータの処理部が立体形状候補を生成する処理；

ここで、前記処理部は、「前記平面形状 f に見える」かどうかの判定を、以下の条件が成立するか否かに基づいて行う。

(条件)

$$f(x, y) = 0 \Rightarrow \forall z, g'(x, y, z) = 0$$

$$f(x, y) = 1 \Rightarrow \exists z, g'(x, y, z) = 1$$

ただし、f(x, y) : x - y 平面に置いたときの平面形状 f ；

$g'(x, y, z)$: 判定対象となる立体形状
であり、

前記平面形状 f はビットマップで与えられるものである。

【請求項 2】

コンピュータを備え、前記コンピュータは以下の処理を行うことを特徴とする立体形状生成装置：

- (1) 平面形状 f の入力を前記コンピュータが受け付ける処理；
- (2) 第 1 の方向への奥行きを持ち、さらに、前記第 1 の方向から前記平面形状 f に見える第 1 の立体形状データを前記コンピュータの記憶部に保存する処理；
- (3) 前記第 1 の方向とは交差する第 2 の方向への奥行きを持ち、さらに、前記第 2 の方向からは前記平面形状 f に見える第 2 の立体形状データを前記コンピュータの記憶部に保存する処理；
- (4) 前記第 1 の立体形状と前記第 2 の立体形状との論理積演算によって、これらの形状を重ね合わせたときにこれらの形状が重複する領域を残し、非重複領域を除去することにより、前記コンピュータの処理部が立体形状候補を生成する処理；

ここで、前記処理部は、「前記平面形状 f に見える」かどうかの判定を、以下の条件が成立するか否かに基づいて行う。

(条件)

$$f(x, y) = 0 \Rightarrow \forall z, g'(x, y, z) = 0$$

$$f(x, y) = 1 \Rightarrow \exists z, g'(x, y, z) = 1$$

ただし、 $f(x, y)$: $x - y$ 平面に置いたときの平面形状 f ；

$g'(x, y, z)$: 判定対象となる立体形状
であり、

前記平面形状 f はビットマップで与えられるものである。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、立体形状物の生成方法および立体形状物に関するものである。

【背景技術】

【0002】

現在用いられている立体文字は、平面形状（例えば文字、記号、図案その他の形状）に対して奥行き方向の厚みを加えたただけのものである。このような立体文字は、単一の方向またはその近傍から見た場合以外は、同じ平面形状として認識されない。また、このような立体文字は、構造が単純であり、装飾性も乏しい。

【発明の開示】

【発明が解決しようとする課題】

【0003】

本発明は、複数方向から見たときに、実質的に同じ平面形状として認識できる立体形状物およびその生成方法を提供することを目的としている。

【課題を解決するための手段】

【0004】

本発明に係る立体形状物の生成方法は、以下のステップを備えている：

- (1) 第 1 の方向への奥行きを持ち、さらに、前記第 1 の方向からは平面形状に見える第 1 の立体形状を準備するステップ；
- (2) 前記第 1 の方向とは交差する第 2 の方向への奥行きを持ち、さらに、前記第 2 の方向からは、前記平面形状に見える第 2 の立体形状を準備するステップ；
- (3) 前記第 1 の立体形状と前記第 2 の立体形状とを重ね合わせたときに、これらの形状が重複する領域を残し、非重複領域を除去することにより、立体形状候補を生成するステ

10

20

30

40

50

ップ。

【0005】

この生成方法は、さらに以下のステップを備えていてもよい：

(4) 前記第1および第2の方向とは交差する第3の方向への奥行きを持ち、さらに、前記第3の方向からは、前記平面形状に見える第3の立体形状を準備するステップ；

(5) 前記ステップ(3)の後、前記立体形状候補と前記第3の立体形状とを重ね合わせ、これらの形状が重複する領域を残し、非重複領域を除去することにより、新たな立体形状候補を生成するステップ。

【0006】

本発明における立体形状物の生成方法は、以下のステップを備えていてもよい：

(1) z方向への奥行きを持ち、さらに、前記z方向からは平面形状 $f(x, y)$ に見える第1の立体形状 $g_1(x, y, z)$ を準備するステップ；

(2) x方向への奥行きを持ち、さらに、前記x方向からは平面形状 $f(z, y)$ に見える第2の立体形状 $h_1(x, y, z)$ を準備するステップ；

(3) 前記第1の立体形状 g_1 と前記第2の立体形状 h_1 との論理積 $g_2 = g_1 \cdot h_1$ を算出し、この論理積 g_2 に基づいて立体形状候補を生成するステップ。

【0007】

この生成方法は、さらに以下のステップを備えていてもよい：

(4) y方向への奥行きを持ち、さらに、前記y方向からは平面形状 $f(x, z)$ に見える第3の立体形状 $h_2(x, y, z)$ を準備するステップ；

(5) 前記ステップ(3)の後、前記論理積 g_2 と前記第3の立体形状 h_2 との論理積 $g_3 = g_2 \cdot h_2$ を算出し、この論理積 g_3 に基づいて立体形状候補を生成するステップ。

【0008】

前記生成方法は、「前記立体形状候補を生成した後、立体形状候補が第1および第2の方向から前記平面形状に見えるかどうかの確認を行うステップ」をさらに備えてもよい。

【0009】

前記生成方法は、「前記ステップ(5)において新たな立体形状候補を生成した後、この立体形状候補が第3の方向から平面形状に見えるかどうかの確認を行うステップ」をさらに備えることもできる。

【0010】

前記生成方法は、「立体形状候補を生成した後、立体形状候補における非連結領域のうちの少なくとも一つを減らし、その後、前記方向のいずれかまたは全てから同じ平面形状に見えるかどうかを確認するステップ」をさらに備えることもできる。

【0011】

前記生成方法は、「前記立体形状候補を生成した後、前記立体形状候補における非連結領域の数を増加させずに、前記立体形状候補の体積を減少させるステップ」をさらに備えることもできる。

【0012】

前記ステップ(3)における第1の立体形状 g_1 を以下のステップ(a)および(b)により生成することができる：

(a) z方向への奥行きを持ち、さらに、前記z方向からは平面形状 $f(x, y)$ に見える初期の立体形状 $h_0(x, y, z)$ を準備するステップ；

(b) 設定範囲における全ての領域が実質的に密である立体形状 $g_0(x, y, z)$ と前記立体形状 h_0 との論理積により、前記第1の立体形状 g_1 を生成するステップ。

【0013】

本発明における立体形状物の生成方法は、

第1の方向への奥行きを持ち、さらに、前記第1の方向からは平面形状に見える第1の立体形状と、前記第1の方向とは交差する第2の方向への奥行きを持ち、さらに、前記第2の方向からは、前記平面形状に見える第2の立体形状とを重ね合わせた状態において、これらの形状が重複する領域を残し、非重複領域を除去することにより、立体形状候補を

10

20

30

40

50

生成する
構成であってもよい。

【0014】

前記生成方法における第1の立体形状と第2の立体形状とは、実質的にほぼ同じ大きさとされていることが好ましい。

【0015】

また、本発明に係る立体形状物は、前記生成方法のいずれかにより生成されたものである。

【0016】

本発明に係る立体形状物は、交差する2以上の方向から同じ平面形状に見えるものである。この交差とは、例えば直交である。

10

【0017】

本発明に係るデータは、前記した生成方法により生成された立体形状物候補、または前記立体形状物候補から選択された立体形状物を示すものである。このデータは、一般的には、デジタルデータである。

【0018】

本発明に係るコンピュータソフトウェアは、以下の処理をコンピュータに行わせるものである：

(1) 第1の方向への奥行きを持ち、さらに、前記第1の方向からは平面形状に見える第1の立体形状データをコンピュータに保存する処理；

20

(2) 前記第1の方向とは交差する第2の方向への奥行きを持ち、さらに、前記第2の方向からは、前記平面形状に見える第2の立体形状データをコンピュータに保存する処理；

(3) 前記第1の立体形状と前記第2の立体形状とを重ね合わせたときに、これらの形状が重複する領域を残し、非重複領域を除去することにより、立体形状候補を生成する処理。

【0019】

本発明に係る立体形状生成装置は、コンピュータを備えている。もちろん、立体形状物生成装置がコンピュータそのものであっても良い。このコンピュータは以下の処理を行うものである：

(1) 第1の方向への奥行きを持ち、さらに、前記第1の方向からは平面形状に見える第1の立体形状データを前記コンピュータに保存する処理；

30

(2) 前記第1の方向とは交差する第2の方向への奥行きを持ち、さらに、前記第2の方向からは、前記平面形状に見える第2の立体形状データを前記コンピュータに保存する処理；

(3) 前記第1の立体形状と前記第2の立体形状とを重ね合わせたときに、これらの形状が重複する領域を残し、非重複領域を除去することにより、立体形状候補を生成する処理。

【発明の効果】

【0020】

本発明によれば、複数方向から見たときに、実質的に同じ平面形状として認識できる立体形状物を得ることができる。

40

【発明を実施するための最良の形態】

【0021】

以下、本発明の一実施形態に係る立体形状物の生成方法を説明する。まず、この方法の実施に用いる立体形状生成装置の概要を図1により説明する。

【0022】

この装置のハードウェア構成は、一般的なコンピュータと同様であって、処理部1と、記憶部2と、入力部3と、出力部4と、これらを接続するバス5とを備えている。処理部1は、記憶部2に格納されたプログラムに基づいて演算を行う部分である。記憶部2には、後述する生成方法をコンピュータに実行させるプログラムが格納されている。入力部3

50

は、例えばキーボード、マウス、タッチパッドなどの任意の入力手段により構成される。出力部 4 は、例えばプリンタ、ディスプレイ、立体形状物造形装置（例えば光造形装置）などの任意の出力手段により構成される。処理部 1、記憶部 2、入力部 3、出力部 4 などの機能要素は、互いに離間した位置に存在しても良い。この場合、これらは、ネットワークにより接続される。

【 0 0 2 3 】

つぎに、立体形状物の生成方法の詳細を、図 2 ~ 図 6 に基づいて説明する。なお、本実施形態では、立体形状や平面形状という用語を、以下の意味で用いる。また、以下では、 x, y, z の直交座標系を前提としているが、これらの座標軸が斜行していることも可能である。

10

【 0 0 2 4 】

(1) 平面形状

平面形状とは、 R^2 もしくは Z^2 上の部分空間

$$\{H(x,y) \mid x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}\}$$

上で定義された二値関数で定義することができる。ここで、 R は実数空間、 Z は虚数空間を示す。例えば、二値関数の出力が 1 か 0 で表されるとすれば、平面形状は、

$$f(x, y) \in \{1, 0\}$$

のように表せる。この形状は、 z 軸方向から見た平面形状を意味する。平面図形の例を図 3 (a) に、これを二値化した図を同図 (b) に示す。他の軸から見た平面形状は、異なる引数を取る。例えば、 x 軸方向から見た平面形状は、 $f(z, y)$ のように表される。同じ関数 $f()$ で表される平面形状は、実質的に同一であるとする。

20

【 0 0 2 5 】

(2) 立体形状

立体形状とは、 R^3 もしくは Z^3 上の部分空間

$$\{l(x,y,z) \mid x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}, z_{\min} \leq z \leq z_{\max}\}$$

上で定義された二値関数で定義することができる。例えば、二値関数の出力が 1 か 0 で表されるとすれば、立体形状は、

$$g(x, y, z) \in \{1, 0\}$$

30

のように表せる。

【 0 0 2 6 】

(3) 平面形状として見える

ある立体形状 g を回転して得られたいずれかの g' が、ある平面形状 f に対してつぎの関係

$$f(x, y) = 0 \Rightarrow \forall z, g'(x, y, z) = 0$$

$$f(x, y) = 1 \Rightarrow \exists z, g'(x, y, z) = 1$$

が実質的に成り立つとき、立体形状 g は平面形状 f として見えることになる。したがって、得られた立体形状またはその候補が所望の平面形状として見えるかどうかは、コンピュータにより判別可能である。

40

【 0 0 2 7 】

(図 2 のステップ 2 - 1)

まず、第 1 および第 2 の立体形状を準備する。まず、任意の平面図形 $f(x, y)$ において、任意の（一様な）奥行き z を与える。これにより、第 1 の初期立体形状 h_0 を得ることができる（図 4 a 参照）。

【 0 0 2 8 】

一方、任意の平面図形 $f(z, y)$ において、任意の（一様な）奥行き x を与える。これにより、第 2 の立体形状 h_1 を得ることができる（図 4 b 参照）。これにより、 x 方向

50

への奥行きを持ち、さらに、前記 x 方向からは平面形状 $f(x, y)$ に見える第 2 の立体形状 $h_1(x, y, z)$ を得ることができる。ここで、

$$|z| \cong |x|$$

であることが好ましい。

【0029】

さらに、ある設定範囲における全ての領域が密な形状である立体形状 $g_0 = 1$ と h_0 との積関数を取り、これを第 1 の立体形状 g_1 とする（図 4 c 参照）。ここで、 g_0 の設定範囲は、 h_0 の領域を全て含む大きさを持つことが望ましい。

10

$$g_1(x, y, z) = g_0(x, y, z) \cdot h_0(x, y, z)$$

これにより、z 方向への奥行きを持ち、さらに、z 方向からは平面形状 $f(x, y)$ に見える第 1 の立体形状 $g_1(x, y, z)$ を得ることができる。ただし、この g_1 は、一般に、 h_0 と常に等しいため、第 1 の初期形状 h_0 をそのまま第 1 の立体形状としてもよい。

【0030】

これらの演算操作は、記憶部 2 に格納されたコンピュータプログラムに従って、処理部 1 により行われる。以降の操作も原則として同様である。演算の結果得られた各データは、記憶部 2 に格納され、必要に応じて読み出される。

20

【0031】

（図 2 のステップ 2 - 2）

ついで、 g_1 と h_1 との積を計算し、これを立体形状候補 g_2 とする（図 4 d 参照）。

$$g_2(x, y, z) = g_1(x, y, z) \cdot h_1(x, y, z)$$

これにより、 g_1 と h_1 において二値化されたデータの論理積が算出され、その結果、両者とも 1 である部分のみが残る。すなわち、この操作により、「第 1 の立体形状と第 2 の立体形状とを重ね合わせ、これらの形状が重複する領域を残し、非重複領域を除去して、立体形状候補を生成する」という処理をコンピュータにより行うことができる。

30

【0032】

このようにして得られた立体形状候補 g_2 は、一般に、二つの方向（この例では z 方向と x 方向）から、平面形状 f として見える。したがって、本実施形態の方法では、立体形状候補 g_2 により、二つの方向から平面形状 f として見える立体形状を得ることができる。

【0033】

（図 2 のステップ 2 - 3）

つぎに、この実施形態では、得られた立体形状候補 g_2 が、所期の 2 方向（この例では z および x 方向）から平面形状 f として見えるかどうかを判断する。見えない場合は、その候補 g_2 を破棄し、ステップ 2 - 1 に戻って、他の方向（例えば y 軸方向）から平面形状に見える立体形状を用いて、前記の処理を再び行う。

40

【0034】

（図 2 のステップ 2 - 4）

さらに、この実施形態では、y 方向への奥行きを持ち、さらに、y 方向からは平面形状 $f(x, z)$ に見える第 3 の立体形状 $h_2(x, y, z)$ を準備して（図 4 e 参照）記憶部 2 に格納する。この形状 h_2 は、前記した第 2 の立体形状 h_1 と同様にして準備できる。

【0035】

（図 2 のステップ 2 - 5）

ついで、 g_2 と h_2 との論理積 g_3

50

$$g_3(x, y, z) = g_2(x, y, z) \cdot h_2(x, y, z)$$

を生成する。この論理積 g_3 が、新たな立体形状候補となる。論理積 g_3 を生成する処理は、「立体形状候補 g_2 と第3の立体形状 h_2 とを重ね合わせ、これらの形状が重複する領域を残し、非重複領域を除去して、新たな立体形状候補を生成する処理」に対応する。

【0036】

ついで、この立体形状候補 g_3 が、所期の方向（この例では x, y, z の三方向）から平面形状 f に見えるかどうかを確認する。所期の方向から平面形状 f に見える候補 g_3 は、求める立体形状の一つに該当する。3方向から平面形状 f に見える立体形状が得られない場合もある。ただし、確認工程は必須ではなく、立体形状候補 g_3 を直ちに立体形状として扱うことも、可能性としては考えられる。

10

【0037】

さらに、第 n の方向（ n は4以上の整数）から平面形状 f として見える立体形状を生成することも可能である。例えば第4の方向から平面形状 f として見える立体形状を生成するには、 x, y, z のいずれとも交差する方向への奥行きを持つ第4の立体形状（図示せず）を生成する。ついで、第4の立体形状と前記立体形状候補 g_3 との論理積を取り、初期の方向（この例では4方向）から平面形状 f として見えるかどうかを確認する。5以上の方向から平面形状 f として見える立体形状の生成も同様にして可能である。

【0038】

（図2のステップ2 - 6）

20

上記の方法で得られた立体形状は、冗長部分を含むことがある。つまり、一般には、上記した方法で得られた立体形状は、複数方向から平面形状として見える立体形状のうち最大体積のものとなる。

【0039】

そこで、以下、冗長部分を除去する方法を説明する。なお、冗長部分を除去するとは、換言すれば、得られた立体形状の部分集合を取ることに相当する。この部分集合が、所望の複数方向から平面形状 f に見えるのであれば、その部分集合も、求める立体形状となりうる。最適化の一例を図4（f）に示す。

【0040】

（連結領域最小化アルゴリズム）

30

最適化の一例として、連結領域最小化アルゴリズムを説明する。前記の処理により得られた立体形状（候補）を g とする。 g は、所期の複数方向から、平面形状 f として見えるものである。また、 g は、 n 個の領域 s_i ($0 < i < n$) により構成されているとする。

【0041】

この方法では、分離された領域 s_i のうちのひとつまたは複数を減らす（図5のステップ5 - 1）。その後、所望の方向の全て（いずれかであってもよい）から同じ平面形状に見えるかどうかを確認する（ステップ5 - 2）。この処理を、各領域について行い、ステップ5 - 2において確認された立体形状のいずれかを出力する（ステップ5 - 3）。好ましくは、領域 s_i の全ての組み合わせについて確認を行い、最小領域数または最小体積の立体形状を出力する。

40

【0042】

この処理を具体的に説明すると、つぎのようになる。まず、入力（引数）として、領域集合 $\{s\}$ を取る。この集合の要素は s_i である。領域集合 $\{s\}$ に含まれる領域の総和からなる形状を S とする。形状 S が、指定された全ての可視方向から g と同一の平面形状に見えるかどうかを判定する。もし見えなければ、エラーを返して処理を終了する。この処理は、あくまで確認のためのものである。

【0043】

もし見えるならば、集合 $\{s\}$ に属する全ての要素

$$t \in \{s\}$$

50

を、それぞれ領域集合 $\{s\}$ から除去する。t を除去した領域集合のうち、所望の方向から平面形状に見えるものがなければ、エラーを返して処理を終了する。

【0044】

そうでなければ、所望の方向から平面形状に見えるもののうち、もっとも要素数の少ない領域集合 $\{s\} - t$ を、立体形状として返す（ステップ5-3）。

【0045】

図6(a)には、最適化前の立体形状の一例を示す。この例では、この立体形状は、2方向から、平面形状「回」に見える。しかし、この立体形状は、冗長な領域（例えば、前後および両側面の「口」形の部分）が存在する。すると、この冗長な領域が障害となって、正確な方向から見ないと平面形状として見えにくくなることもある。

10

【0046】

この立体形状に、前記した連結領域最小化アルゴリズムを適用した例を、同図(b)に示す。この立体形状も、2方向から所望の平面形状に見える。さらに、この立体形状では、冗長領域が除かれているため、平面形状として観察しうる角度が広がるという利点もある。さらに、実際の形状物を作成する場合は、材料や加工工数を削減できるという利点もある。

【0047】

（他の最適化アルゴリズム）

最適化アルゴリズムとしては、上記に限らず、種々のものが考えられる。例えば、所望の方向から平面形状 f に見え、かつ、もっとも体積や表面積や回転モーメントの少ない立体形状を選ぶ方法がある。

20

【0048】

体積最小化のアルゴリズムの一例を図7に基づいて説明する。この例では、最適化前の立体形状は、3方向から、平面形状「口」に見えるものである。この立体形状は、冗長部分を含んでいる（図7a参照）。

【0049】

これに対して、もっとも体積が少なく、かつ、3方向から所望の平面形状に見える立体形状を図7(b)に示す。この図では、体積は最小であるが、非連結な領域の数が増加している。用途によっては、このような最適化が不適切な場合がある。例えば、この立体形状は、実際の立体物を作成する用途には、作製が難しくて不適切なことがある。

30

【0050】

このような場合には、例えば、連結領域数を維持する（または増やさない）という条件下で、最小体積となる立体形状を生成することができる（図7c参照）。この立体形状は、体積自体は図7(b)のものより多いが、非連結領域の数は増加しないという利点がある。

【0051】

さらに他の最適化アルゴリズムとしては、例えば次のようなものがある。

表面積最小：もっとも表面積が少なく、かつ、3方向から所望の平面形状に見える立体形状を生成するものである。これを用いると、CGなどでポリゴンモデルを作成する際、構成ポリゴン数を抑えることができ、メモリやレンダリングコストを節約することができる。また、実際の物体を製作した場合、物体の質量が小さくなるため、物体の強度や安定性が向上する。

40

断面積最大：このアルゴリズムは、他の最適化アルゴリズムの制約条件として用いられることが多い。すなわち、これは、他の最適化アルゴリズムによって得られる立体形状のうち、もっとも断面積が大きい立体形状を生成するものである。一般には、最小断面積が最大となる形状を選択することが好ましい。このアルゴリズムは、体積最小化や表面積最小化のアルゴリズムとは相反する傾向にある。各部（特に断面積が小さい部分）の断面積を大きくすることにより、製品の強度を向上させることができる。

回転モーメント最小：回転モーメントが最小で、かつ、3方向から所望の平面形状に見える立体形状を生成するものである。

50

【実施例 1】

【0052】

図 8 ~ 図 1 1 には、前記した方法により生成された、最適化前の立体形状の例を示す。図 8 は、2 方向から文字「水」と見えるもの、図 9 は、2 方向から文字「凸」と見えるもの、図 1 0 は、3 方向から文字「匹」と見えるもの、図 1 1 は、2 方向から特定の図形（「陰陽巴紋」）に見えるものである。

【0053】

前記した立体形状の応用、生成方法（作成方法）、使用材料、適切な最適化アルゴリズムの例を下記の表 1 に示す。

【表 1】

10

応用例	生成方法	使用材料	最適化アルゴリズム
小サイズの製品 (アクセサリ、キーホルダーなど)	掘削	金属、木材	連結領域数最小
	鋳造または集層造形方法	金属、樹脂、ゴム	連結領域数最小 体積最小 断面積最大
	手作業	セラミック(陶磁器)、木材、石材、金属、含プラスチック	連結領域数最小 断面積最大
掘削または手作業	連結領域数最小 回転モーメント最小		
大サイズの製品 (看板、モニュメントなど)	掘削または手作業	セラミック(陶磁器)、木材、石材、金属、含プラスチック	連結領域数最小 回転モーメント最小
証明、花火など	機械造形または手作業	各種発光体(ネオン管、LED、火薬など)	連結領域数最小 体積最小
CGモデル	サーフィス・モデル	-	連結領域数最小 表面積最小
	ソリッド・モデル ポリウム・データ	-	体積最小
	ホログラム、ステレオグラムなどの立体視モデル	-	連結領域数最小

20

30

【0054】

なお、本発明は、上記した実施形態に限定されるものではなく、本発明の要旨を逸脱しない範囲内において種々変更を加え得るものである。

また、前記実施形態を実現するための各部（機能ブロックを含む）の具体的手段は、ハードウェア、ソフトウェア、ネットワーク、これらの組み合わせ、その他の任意の手段を用いることができる。

40

さらに、機能ブロックどうしが複合して一つの機能ブロックまたは装置に集約されても良い。また、一つの機能ブロックの機能が複数の機能ブロックまたは装置の協働により実現されても良い。さらに、機能ブロックどうしは、ネットワークで連結されて相互に通信可能なものでもよい。

また、前記実施形態での各種の処理は、コンピュータで実行できるだけでなく、手作業により実行することも可能である。

【産業上の利用可能性】

【0055】

本発明の立体形状物の生成方法により得たデータを用いて、立体形状物を示すCGモデ

50

ルを生成することができる。また、このデータをCADデータに変換すれば、機械造形や手作業により立体形状物を実際に生成するための支援を行うことができる。さらに、この方法で得られた立体形状物は、複数方向から所望の平面形状に見えるので、装飾物としても高い利用価値を有する。

【図面の簡単な説明】

【0056】

【図1】本発明の一実施形態に係る立体形状物の生成方法に用いるコンピュータの構成の一例を示す概略的なブロック図である。

【図2】本発明の一実施形態に係る立体形状物の生成方法の概略的な手順を示すフローチャートである。

【図3】図3(a)は、平面図形の一例を示す説明図、同図(b)は、この平面図形を二値化した例を示す説明図である。

【図4】図4(a)～(f)は、図2に示す生成方法で生成される立体形状の一例を説明するための説明図である。

【図5】立体形状物の最適化方法の概略的な手順を示すフローチャートである。

【図6】図6(a)および(b)は、図5に示す最適化方法で生成される立体形状の一例を説明するための説明図である。

【図7】図7(a)～(c)は、他の最適化方法で生成される立体形状の一例を説明するための説明図である。

【図8】図8(a)および(b)は、本発明の一実施形態に係る生成方法で生成される立体形状の一例を説明するための説明図である。

【図9】図9(a)および(b)は、本発明の一実施形態に係る生成方法で生成される立体形状の一例を説明するための説明図である。

【図10】図10(a)および(b)は、本発明の一実施形態に係る生成方法で生成される立体形状の一例を説明するための説明図である。

【図11】図11(a)および(b)は、本発明の一実施形態に係る生成方法で生成される立体形状の一例を説明するための説明図である。

【符号の説明】

【0057】

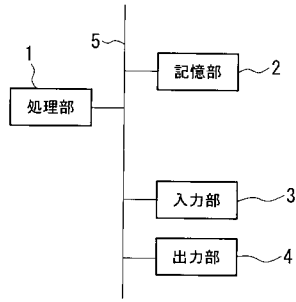
- x 第2の方向
- y 第3の方向
- z 第1の方向
- 1 処理部
- 2 記憶部
- 3 入力部
- 4 出力部
- 5 バス

10

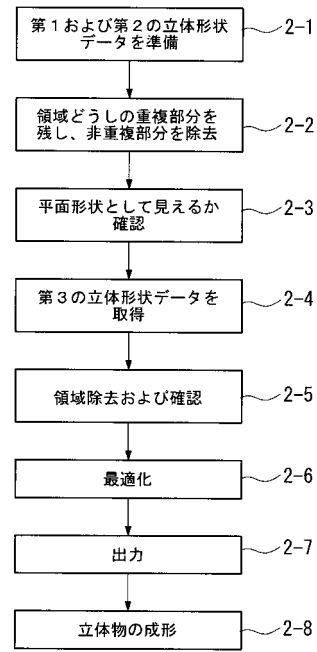
20

30

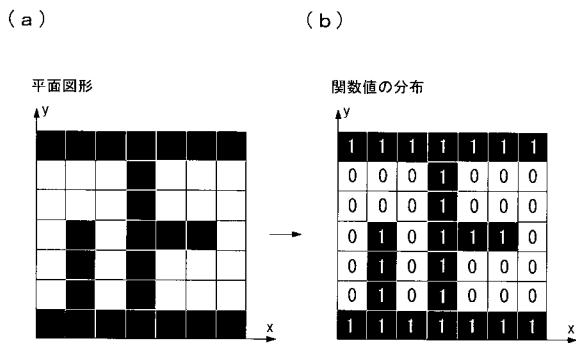
【図1】



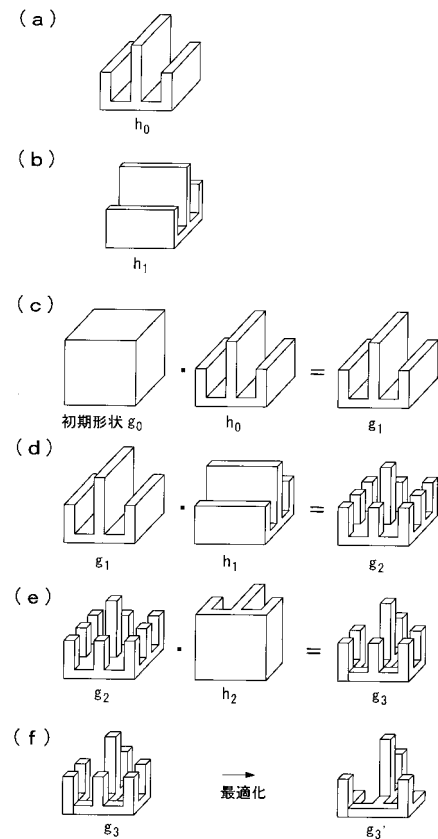
【図2】



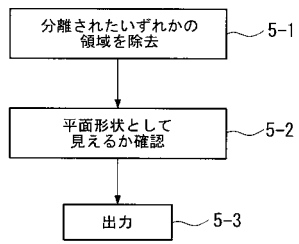
【図3】



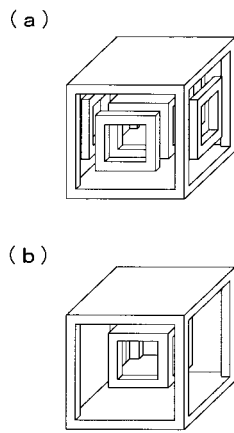
【図4】



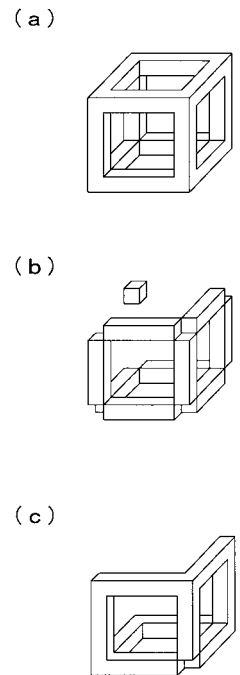
【図5】



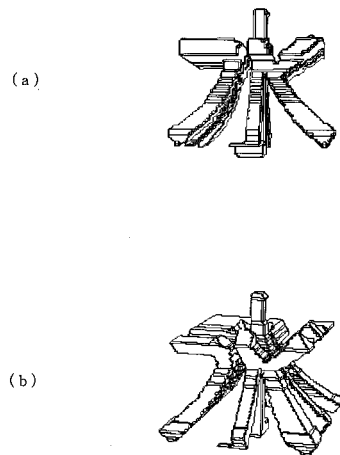
【図6】



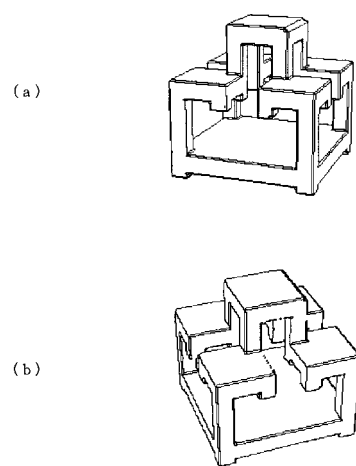
【図7】



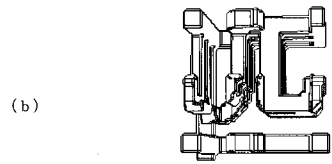
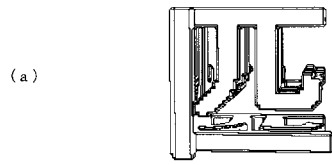
【図8】



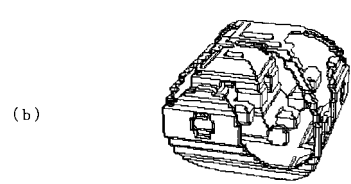
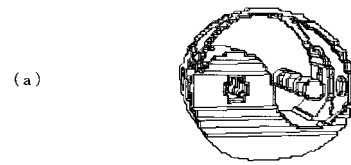
【図9】



【 図 10 】



【 図 11 】



フロントページの続き

(56)参考文献 特開平10-111882(JP,A)
特開平07-200872(JP,A)

(58)調査した分野(Int.Cl., DB名)
G06F 17/50
G06T 19/00