

Superquadrics による物体の部品分割手法

堀越 力 浜野輝夫 石井健一郎

NTTヒューマンインタフェース研究所

あらまし

距離画像から物体の概略形状を自動的に求める手法として、新たに Superquadrics による Wrapping 法を提案する。本手法は、袋に包まれた物体を推測する過程に類似している。最初に、全データに対して近似した関数を袋に見立てて物体を包み込み、その関数を徐々に収縮させていく。そして、関数を突き破って出てきた部分を物体の凸部分として認識し、分離していく。これにより、物体の距離画像から凸部分を自動的に分離し、個々の凸部分に再度関数を近似することにより、物体の概略形状を記述できる。また実際に、Wrapping 法を距離画像に適用し、良好な結果を得ることができた。

Decomposing an Object into Convex Parts by using Superquadrics

Tsutomu HORIKOSHI Teruo HAMANO Kenichiro ISHII

NTT Human Interface Laboratories

1-2356 Take Yokosuka-Shi Kanagawa 238-03, Japan

Abstract

A new method "Wrapping" is proposed which can automatically decompose an object into its primitives from range data. This method roughly duplicates the process we use when determining the content of a bag by feeling it with one's hand. The superquadrics, which is fitted to the range data, is likened to the bag. As the superquadrics "bag" is shrunk, convex parts gradually become obvious. Such parts are separated from the range data, and the parts are refitted. In this manner, the shape of an object is recovered by fitting superquadrics to each convex part. We have successfully applied this method to a actual range data set and typical results are shown.

1 はじめに

紙袋の中身が分からないとき、我々は、その袋の回りを手で触ってみることがよくある。つまり、人は中の物体の凹凸具合を調べて大まかな形状を類推して、何であるか当てることができる。

さて、物体の距離画像からその形状を記述することを考えてみると、一般に、形状がわからないデータに簡単なプリミティブを当てはめて、物体の三次元的な構造を獲得する。このときの処理過程に対しても、先の袋の場合と同じような過程を経て大まかな形状を類推することができる。

一般に、物体を効率よく記述する為には、大局的な記述と詳細な記述が可能であることが重要であるといわれている [1, 2]。そこで、距離画像などから物体形状を求める手順としても、まず概略形状を求め、徐々に詳細な形状を記述していくといった手段が有効であると考えられる。

これまで、距離画像から三次元モデルを獲得する試みは数多くなされている。距離画像への直接的な近似としては、Bajcsyら [4, 5] が Superquadrics に *Tapering* や *Bending* など変形関数を組み込み、物体の概略形状に近似させる方法を検討した。また、遮蔽輪郭や物体表面の曲率などを利用することで、基本要素分割を試みている。しかし、このような Bottom-up 的方法の場合、局所的な変動 (誤差など) に左右され易いという問題がある。また、Top-down 的アプローチとしては、Pentland [3] が、Superquadrics で表現されたテンプレート当てはめをしていくことで、物体の部品を大きい順に求めることに成功している。この方法は必ず関数で表現できる部品に分割されるが、二次元レベルでの分割で、奥行き情報が失われる場合がある。また、物体を概略形状から記述する方法として、村木 [6] は、分割という考えを全く使わずに、等関数値曲面記述により、物体を少ないプリミティブで記述する手法を提案し、自然物体に対する検討を行っている。しかし、現在のところ、かなりの計算量を必要としている。

本研究の目的は、物体の大局的な形状を容易に求め

ることであり、人が、袋で包まれた物体の形状を類推するような方法として、新たに Superquadrics を用いた Wrapping 法を提案する。これは、Superquadrics を紙袋と考え、物体を包み込み、この袋を小さくしていくことで現れる物体の凸部分を徐々に分離していくことで、物体形状を記述する方法である。

以降では、まず2章にて、Wrapping 法の原理及び詳細に付いて述べ、3章にて、実際の距離画像に適用した例を示す。

2 Wrapping 法

2.1 基本的原理

図1のような曲線 C に、上部から直線 l を降ろしていくことを考える。このとき、直線と曲線の上下関係

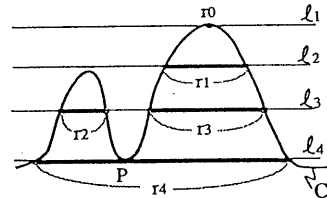


図1 Wrapping 法の基本的原理

に注目すると、始めは直線 l_1 と曲線 C との交わりは、 r_0 の1点だけである。徐々に直線 l を下げていき、直線 l_2 の位置まで下がったとき、線分 r_1 の部分で曲線 C が直線 l より上側に存在する。直線 l_3 のときは、2つの線分 (r_2, r_3) の部分で曲線 C が直線 l より上側になっている。直線を下げるに従い、2つの部分 (r_1, r_2) の長さは徐々に伸びていき、直線 l_4 の位置になったとき2つの線分であった部分が1つの線分 r_4 となる。このときの接点 P を見ると、それは曲線 C の谷の中心点を表していることがわかる。つまり、この接点を見つけることで2つの山を分離することができる。

Wrapping 法とは、この曲線 C を物体表面と考えたときに、この直線 l として超二次関数面を用いた方法である。

2.2 Wrapping 法の概要

Wrapping 法は、概略形状として近似された関数を収縮させながら、関数外部に存在するデータを関数表面に投影する。各収縮段階で、関数表面上に形成されるデータの集まりのラベリングを行い、複数の異なるラベル領域が重なったときを検出する。そして、重なった領域が検出されたなら、1段階前のラベリング結果において、統合した複数のラベル領域に属していたデータ毎にデータの分割を行う。以上の処理を繰り返すことにより、物体形状の凸に相当する部分が徐々に分離されていくことになる。

Wrapping 法の大まかなフローを図2に示す。

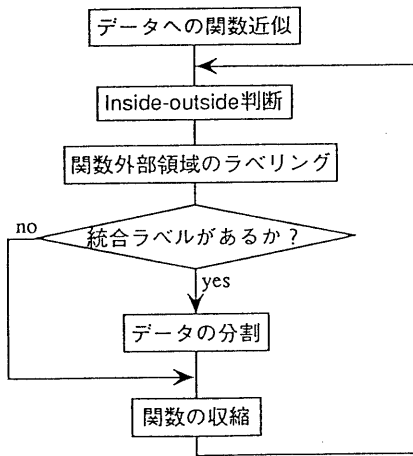


図2 Wrapping 法処理フロー

以後各ステップについて述べていく。

2.3 関数の概略形状近似

物体の概略形状を求めるために、まず1つの Superquadrics を入力データに近似させる。

2.3.1 Superquadrics による形状近似

Superquadrics は次式で表わされる。

$$S = \left\{ \left(\frac{x}{a_1} \right)^{2/\epsilon_2} + \left(\frac{y}{a_2} \right)^{2/\epsilon_2} \right\}^{\epsilon_2/\epsilon_1} + \left(\frac{z}{a_3} \right)^{2/\epsilon_1} = 1 \quad (1)$$

この関数をデータに近似するために、非線形最小二乗法 (Levenberg-Marquadt Method) が良く使われ、

次の二乗誤差が最小となる関数を求めている。

$$\chi^2 = \sum_{i=1}^N \{1 - S(x_i, y_i, z_i; \mathbf{a})\}^2, \quad i = 1, \dots, N \quad (2)$$

ここで、 \mathbf{a} は Superquadrics のパラメータ $(a_1, a_2, a_3, \epsilon_1, \epsilon_2)$ を表す。しかし、最小二乗法は、局所的に大きなノイズが入ったデータの影響を受け易いため、関数で表現できない物体形状に近似させた場合、物体の概略形状とは全く異なる形状に収束してしまう場合がある。そのため、制約つき最小化を行う必要がある。

2.3.2 概略形状への近似

物体全体の概略形状を求めるといった観点から、以下の様な条件を設定し、この条件の下で関数を近似させることにした。

1. 二乗誤差 χ^2 が最小であること。
2. 全てのデータが関数内部に含まれること。
3. データの占有体積 \geq 関数の占有体積

つまり、これらの不等式の条件を満たす最適化問題を解くことに置き換え、次式で表す最適化問題を考える。

$$\begin{cases} \min_{\mathbf{a}} \chi^2 \\ S(x_i, y_i, z_i; \mathbf{a}) < 1, \quad i = 1, \dots, N \\ a_1 a_2 a_3 - V_0 \leq 0 \end{cases} \quad (3)$$

ここで、 N は3次元データ数、 V_0 はデータの占有体積であり、入力されたデータを直方体と仮定する。

この問題は不等式制約問題となっているが、スラック変数 s を導入することにより

等式制約問題

$$\begin{cases} \min_{\mathbf{a}} \chi^2 \\ \text{条件1: } S(x_i, y_i, z_i; \mathbf{a}) - 1 + s_i^2 = 0, \quad i = 1, \dots, N \\ \text{条件2: } V_0 - a_1 a_2 a_3 + s_0^2 = 0 \end{cases} \quad (4)$$

に変換することができ、Lagrange 乗数法により解くことができる [7]。つまり、(5) 式に示す Lagrange 関数 L に関して、その値が最小となる関数パラメータ \mathbf{a} を求めることで、物体の最外郭形状としての概略形状を求めることができる。

$$L(x, y, z, a, \lambda, t) = \sum_{i=1}^N \{S(x_i, y_i, z_i; a) - 1\}^2 + \sum_{i=1}^N \frac{1}{2t_i} [\max\{0, \lambda_i + t_i (S(x_i, y_i, z_i; a) - 1)\}^2 - \lambda_i^2] + \frac{1}{2t_0} [\max\{0, \lambda_0 + t_0 (a_1 a_2 a_3 - V_0)\}^2 - \lambda_0^2] \quad (5)$$

但し、 λ は Lagrange 乗数、 t は Penalty parameter を表す。

計算手順としては、まず修正ニュートン法等により L が最小になるパラメータの組合せを求める。次に、Lagrange 乗数及び Penalty Parameter を更新し、再度 L が最小になるパラメータの組合せを求める。この操作を繰り返し、データとの誤差が閾値以下になったら、そのときのパラメータを最適値とする。ここで、初期値としては $\lambda_i^0 = 0$ 、 $t^0 = 1$ であり、 λ_i^k 、 t^k の $k+1$ 回目の更新は、

$$\lambda_i^{k+1} = \lambda_i^k + t^k \{S(x_i, y_i, z_i; a) - 1 + (y_i^k)^2\} = \max\{0, \lambda_i^k + t^k \{S(x_i, y_i, z_i; a) - 1\}\} \quad (6)$$

$$t^{k+1} = \alpha t^k \quad (7)$$

で行われる¹。

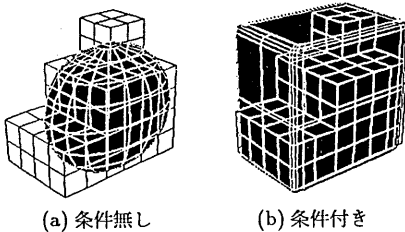


図 3 関数近似形状結果

図 3 は、ブロックを積み重ねた形状の物体に関数を近似させた例であり、条件を付けた場合と、付けない場合での近似結果の違いを示している。黒い部分が近似された関数を表す。条件式 (4) を付けることにより、物体を包み込み、かつ物体表面に最も近似された形状（物体の凸閉包のような形）が求められていることがわかる。

¹ α は定数であり、一般に $\alpha \leq 10$ として使われる。

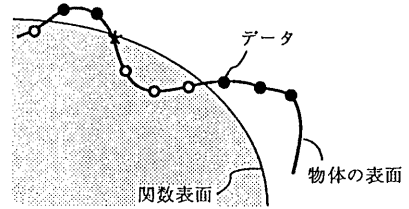
2.4 データの inside-outside 判断

物体の凸部分を検出するため、各データが関数表面より外部に存在するかどうかを判別する。そのために (1) 式を inside-outside 関数として用いる。

即ち、三次元座標 (x, y, z) の位置を判断する Inside-outside 関数 $S(x, y, z)$ を

$$S(x, y, z) \begin{cases} < 1 & \text{関数内部 (○)} \\ = 1 & \text{関数表面上 (*)} \\ > 1 & \text{関数外部 (●)} \end{cases} \quad (8)$$

と定義する。（図 4 参照）



- は関数外部と判断されたデータ、
- * は関数表面上に存在するデータ、
- は関数内部と判断されたデータを表す。

図 4 inside-outside 関数

2.5 関数外部領域のラベリング

各データの inside-outside 判断の結果を関数表面上に投影すると、内部に存在するデータや、外部に存在するデータによるいくつかのデータの集まった領域が形成される。ここでは、関数外部に存在するデータの領域に注目し、この領域のラベリングを行う。即ちこのラベリングされた領域が、関数を突き破っている物体の凸部分を表すことになる。

2.6 関数の収縮

2.6.1 一様収縮

Superquadrics の大きさを収縮させるにはスケールパラメータ a_1, a_2, a_3 を小さくすれば良い。各 a_1, a_2, a_3 の値から一定量 (δ) ずつ差し引くことで、関数を一様に収縮させていくと、凸部分が徐々に関数外部に出てくる。物体表面の一部分に凸部分が有る場合（例えば、コーヒーカップの取っ手など）は関数全体を一様に収縮させることでその凸部分を分離することが可能である

[8]. しかし、ブロックを積み重ねたような形状に対しては、この一様収縮では分割する事ができない。そのような場合は、収縮方向を限定することで、分割を可能にしている。

2.6.2 関数収縮方向の区別

収縮方向として、スケールパラメータ a_1, a_2, a_3 の変化する方向が考えられる。各パラメータを別々に変化させると、xyz 軸各正負方向の 6 方向へ別々に収縮が可能となる。

各収縮方向は、関数表面を次のように 6 つの区域（収縮方向）に分割している。

$$\left\{ \begin{array}{l} 1: -\pi/4 \leq \eta < \pi/4, \quad -\pi/4 \leq \omega < \pi/4 \\ 2: -\pi/4 \leq \eta < \pi/4, \quad \pi/4 \leq \omega < 3\pi/4 \\ 3: -\pi/4 \leq \eta < \pi/4, \quad 3\pi/4 \leq \omega < 5\pi/4 \\ 4: -\pi/4 \leq \eta < \pi/4, \quad 5\pi/4 \leq \omega < -\pi/4 \\ 5: \pi/4 \leq \eta \leq \pi/2, \\ 6: -\pi/4 \leq \eta \leq -\pi/2, \end{array} \right. \quad (9)$$

ここで、 ω が経度、 η が緯度を示している。

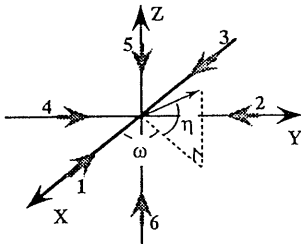


図 5 関数収縮方向

この分割された各区域において、ラベリングされた領域が収縮により統合されるかを調べていく。

2.6.3 各方向の収縮可否の判断

関数を収縮させることにより凸部分が出てくるということは、その方向に関数内部に存在するデータがあるということである。そこで、各収縮方向において、先のラベリングされた領域の形状を調べ、図 6 に示すように各収縮方向 i ($i = 1, \dots, 6$) 内のラベリング領域の横方向最大長 L_{1i} と縦方向最大長 L_{2i} と各区域のそれぞれの最大長 M_{1i}, M_{2i} とを比較している。

そして、収縮方向 i に対して、

規則 1: $(L_{1i} < M_{1i}) \wedge (L_{2i} < M_{2i})$ を満たす i が存在するならば、 i 方向に収縮させる。

規則 2: 規則 1 を満たす i が存在せず、かつ

$(L_{1i} < M_{1i}) \vee (L_{2i} < M_{2i})$ を満たす i が存在するならば、 i 方向に収縮させる。

といった 2 つの規則のもとで、各方向に関数を収縮させるかどうかを判断する。

ところで、スケールパラメータを変化させると、各軸正負方向共に変化してしまう。そこで、パラメータを小さくした後、そのパラメータの軸方向に平行移動することで結果的に、各軸正負方向別々に収縮を行っている。例えば、1 回の収縮される量が δ のとき、収縮方向が (9) 式における “1” の区域での収縮では、スケールパラメータ a_1 の新しい値は $a_1 - \delta/2$ となり、更に関数自体を $-\delta/2$ だけ x 軸方向に平行移動させるという処理を行う。

図 7 は、各収縮段階でのラベリング結果を示しており、関数表面の位置を横軸が経度、縦軸を緯度として展開している。そして、A, B はラベリング領域、S および ● はそれぞれ、関数表面上、関数内部にデータの存在する領域を表している。空白の部分は、そこにデータが存在しないことを意味する。

2.7 データの分離

各収縮段階でデータの分離は次の 2 段階で行われる。

1. 規則 1 を満たす方向が無くなった時点で、ラベリングされている領域を分離する。
2. 収縮が終了した時点で、ラベリングされている領域のデータを分離する。

まず、図 6(b) から分かるように、規則 1 に従って、関数の上面に相当する部分（収縮方向 5）のみの収縮が行われる。そして、図 7 の (a) から (b) になった時点で、関数表面が図 6(a) の “←” の部分まで収縮している。このとき、規則 1 に当てはまる収縮方向が無くなったため、図 7(b) のラベル A に属するデータが分離される。

次に、規則 2 に従って、収縮方向 “1” と “5” のみの方向に関数が収縮していく。関数を収縮させていくに従って、図 7(b) → (c) → (d) → (e) のようにラベリ

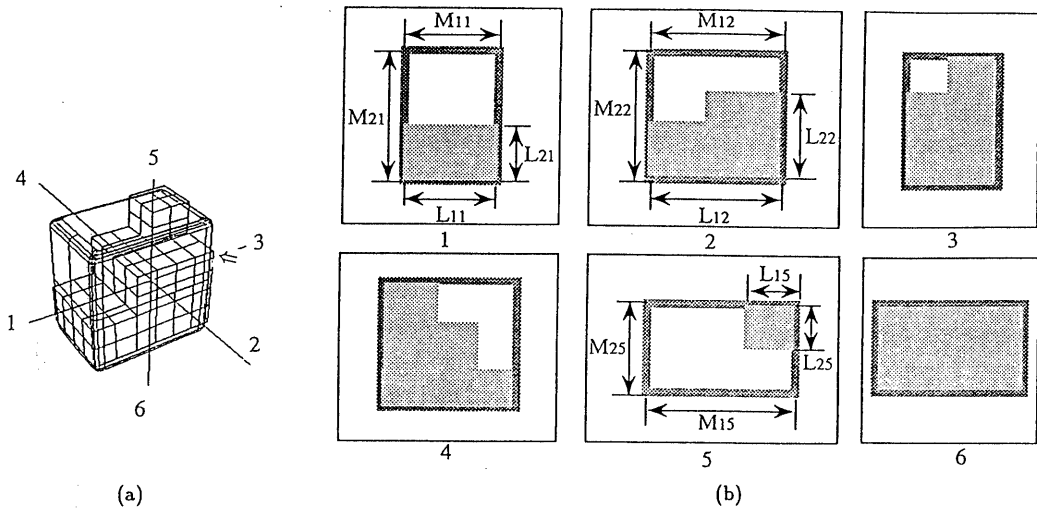


図6 収縮方向の各区域でのラベリング領域

■ ラベリング領域
 □ 各収縮方向での各区域の関数表面形状

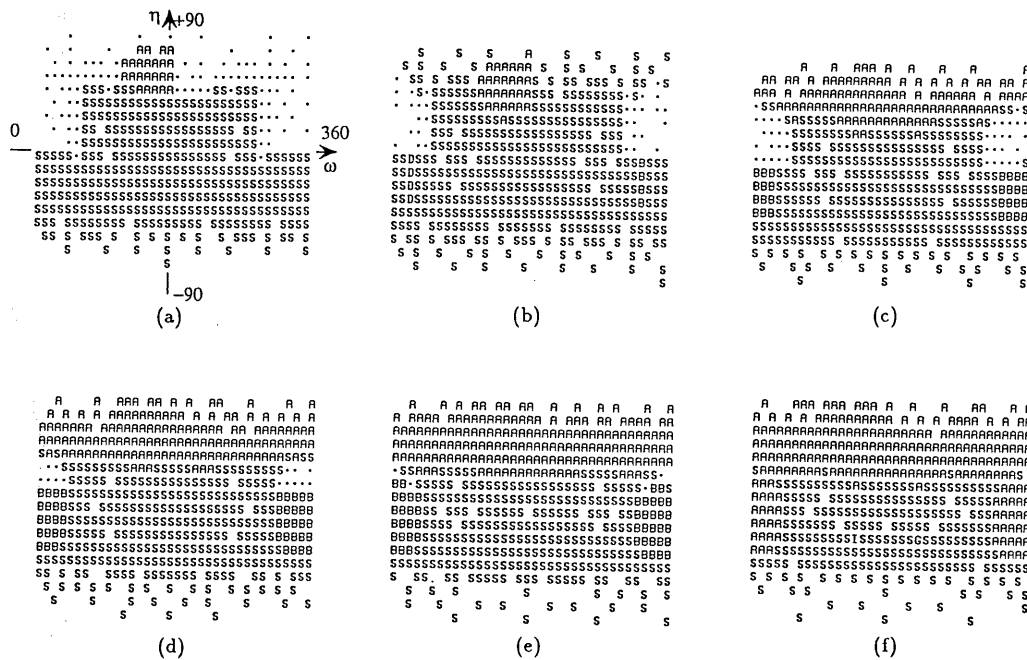


図7 各収縮段階でのラベリング結果

横軸が経度 ω , 縦軸が緯度 η を表す.

A, B: ラベリング領域, S: 関数表面上のデータ, •: 関数内部のデータ

ング領域が徐々に拡大していく。そして、図7(e)から(f)に変わるとき、ラベルAとBが1つのラベルAに変化し、BがAに統合されている。即ち、2.1で説明したように、2つの凸部分間のつながっている部分に関数表面が存在することになる。

そこで、2つの領域が統合したときの1段階前のラベリング結果(図7(e))をもとに、各領域(A,B)に属するデータを分離する。

以上の操作によって分割されたデータへの近似結果を図8(b)に示す。ここでは、分離された各データへ近似された関数と最後に残った関数(ブロック右下の部分)を同時に表示している。

本手法はデータを極座標に変換しているため、inside-outside判断により、同一方向に関数内部のデータと外部のデータが混在する可能性がある。そこで、分離する場合は、外部に存在するデータのみを分離している。

3 実データへのWrapping法の適用例

次に実際の三次元データに適用した。

3.1 距離画像への適用

三次元データとしてはEcho スキャナー(能動光による距離計測装置)により計測した自動車のモデルの距離画像を用いた。この距離画像は、物体の全方位の距離データである。図9(a)は、各表面座標データを●で示しており、このデータに対して(5)式的最適値を求めた結果、図9(b)の様な形状が得られた。そして、Wrapping法により、データを分割した結果、図9(c)に示すように、車のボディと屋根の部分に分割することができた。分割された各データに関数を近似した結果を図9(d)に示す。本手法の原理からすると、車を分割した場合、車のボンネット部分と後ろのトランクの部分は別々に分離されるべきであるが、図9の様に側面データに欠けている部分が有ったため、1つのボディとして分割されている。

この近似結果は、かなり粗い近似ではあるが、(a)に示すような部分的に疎なデータであっても、凸部分を分離、近似することが可能であることが分かる。

Echo スキャナーは物体表面の場所によりデータが計

測できない部分(水平面に相当する部分は計測されない)や、半径の違いによる計測間隔のばらつきがあるため、1つの面であるべき領域が複数のラベリング領域として分離される場合がある。そこで、ラベリングを行う際に、ラベリング領域がある閾値以上になるまで凸領域としての識別を行わないようにしている。

4 おわりに

物体の表面データから基本要素に分割し、記述する手法としてWrapping法を提案した。本手法は、知識などを使わず、物体の凸部分の分割が可能であり、距離画像などのノイズの多い画像への適用が可能である。本手法は、一般に最初に近似される近似形状により、分割されるデータ(ラベリングされる領域)が変わってしまうが、Lagrange乗数法を用いることで、物体の凸包に近い形状を比較的安定に求めることが可能であり、物体の概略形状を記述する段階での分割はほぼ自動的に行うことができる。

一般に、物体の部品分割の妥当性やその判断は、目的により異なり、一意に決めることは困難であるが、本手法の場合、収縮する方向を対象により変化(限定)させることで目的に合った分割を自動的に行う為の1つの手法として利用可能である。

今後は、更に多くの距離画像などを対象として、疎なデータやノイズの多く含まれるデータに適用し、ラベリング領域の統合の仕方などに関して検討していく。また、より多様な形状、及び詳細形状の記述等に関する検討も行っていく予定である。

謝辞 本研究を進めるにあたり、ご指導頂いたNTTヒューマンインタフェース研究所小林幸雄音声部長、遠藤隆也マルチメディア研究部長に深く感謝いたします。また、本研究に対し有益なご討論を頂いたマルチメディア研究部の皆様にも感謝致します。

参考文献

- [1] D.Marr 乾, 安藤訳, ビジョン-視覚の計算理論と脳内表現-, 産業図書, 1989

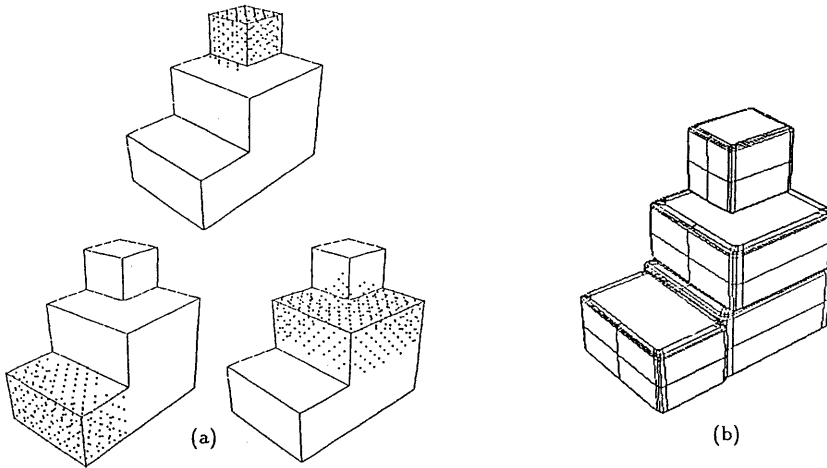


図 8 ブロックの分割例

- (a) 分割された各凸部分のデータ (●が1つのデータを表す。)
 (b) 各分割されたデータに関数を近似した結果

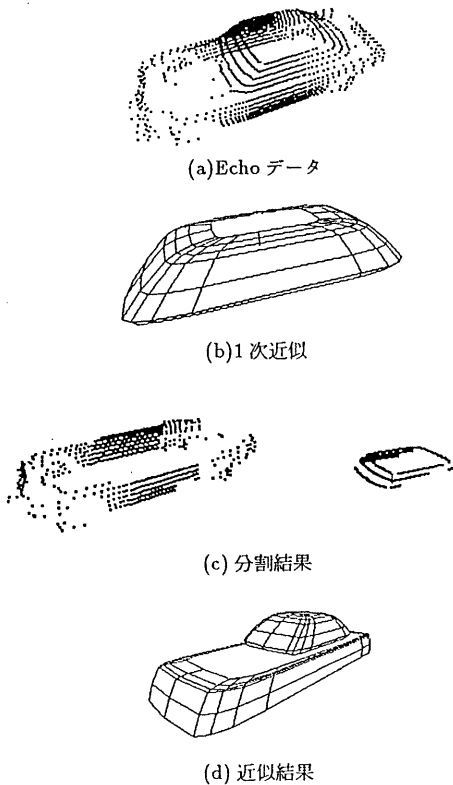


図 9 車の分割

- [2] D.Terzopoulos, D.Metaxas, "Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics", 3rd Int. Conf. Computer Vision, pp606-615, Osaka, Japan, 1989.
- [3] A.Pentland, "Automatic Extraction of Deformable Part Models", Int. J. Computer Vision, Vol.4, No.2, pp107-126, 1990.
- [4] F.Solina and R.Bajcsy, "Recovery of Parametric Models from Range Images: the Case for Superquadrics with Global deformations", IEEE PAMI, Vol12, No.2, pp131-147, 1990.
- [5] A.Leonardis, Alok Gupta, and R.Bajcsy, "Segmentation as the Search for the Best Description of the Image in Terms of Primitives", 3rd Int. Conf. Computer Vision, pp121-125, Osaka, Japan, 1989.
- [6] 村木, "距離データからの等関数値曲面記述の生成", 情報処理学会, CV70-11, 77-84, 1991.
- [7] 今野, 山下非線形計画法, 日科技連, 1978.
- [8] 堀越 "Superquadrics による物体の基本要素分割", 信学会秋季全国大会, D-399, 1990.