

# 医用3次元画像処理のための基本アルゴリズム

安田 孝美

名古屋大学 工学部 情報工学科

あらまし

医学領域におけるCT (Computed Tomography) に代表される3次元画像は、実際の臨床の場において診断のみならず治療目的にも用いられる可能性を秘めている。従来、人体内部をできるだけ明確に表示するための3次元表示法がいくつも提案されているが、最近では、これらの手法によって生成された3次元画像を画面上で外科医が手術をするように操作することも研究段階では可能となっている。

本文では、3次元医用画像の表示法およびそれらを用いて手術計画に利用するための最近の研究動向について述べる。

## Fundamental Algorithms for Medical 3-D Image Processings

*Takami YASUDA*

Dept. of Information Engineering,  
Fac. of Engineering, Nagoya University  
Chikusa-ku, Nagoya 464-01, Japan

### **Abstract**

Three-dimensional information in medical images has a great deal of potential to be used not only for diagnostic but also therapeutic purposes. Several 3-D visualization techniques have been reported to render human internal organs as clearly as possible. It is now possible to interactively manipulate these 3-D images just as a surgeon would do during a surgery.

This paper examines the current developments for both visualization and computerized surgical planning.

## 1. はじめに

デジタル画像は工業をはじめとしたさまざまな産業分野で生成・取得され、それらを有効に利用するためにはコンピュータによる画像処理が重要になっている。医学においては、レントゲンをはじめとして多くの医用画像が実際の臨床の場で活躍している。その中で人体内部を断面の系列で可視化するCT (Computed Tomography)画像は本質的に3次元情報を有しており、診断・治療に最も有効な画像の一つになっている。CT画像はX線吸収量などの物理量をデジタル・データとして記録しているので、計算機によりその3次元情報を処理して可視化することが可能である。一般にこの計算機内での処理は関心領域の抽出後、《形状表現》→《投影》→《描画》→《操作》の流れとなる。従来、画像処理やコンピュータ・グラフィックスの手法を基本としたさまざまな表示法が考案され、その全般的な基本的技法については横井ら<sup>(1)</sup>や鳥脇<sup>(2)</sup>により解説されている。しかしその後、画質の優れた新たな描画法がいくつか報告されており、また単に表示のみの処理から、何らかの操作を会話的に行なうことにより手術シミュレーションに用いる実用的な研究も盛んになっており、本文では主にこれらについて解説する。

## 2. 形状表現

断面の系列で表現されている3次元物体の元の形状を再構成する主な方法として、ワイヤ・フレーム法、面素構成法、ボクセル法が提案されている。以下ではこれら代表的な形状表現法について解説する。

### 2.1 ワイヤ・フレーム法

表示したい関心領域の輪郭線をすべてのCT断面から抽出し、それらをディスプレイに投影することにより3次元形状を表現する。実際には輪郭は点列として抽出され、表示の際スプライン曲線などで滑らかに結合させる場合もある<sup>(3)・(4)</sup>。また、輪郭線をCT断面に垂直に断面間隔の半分だけ上下方向に平行移動してできる板を積み重ねる輪郭線挿引法を著者らが報告している(図1)<sup>(5)</sup>。

これらの手法は共に断面上で輪郭さえわかっていればその3次元形状をある程度表現することが可能なものであるが、詳細な形状を表現することは難しい。

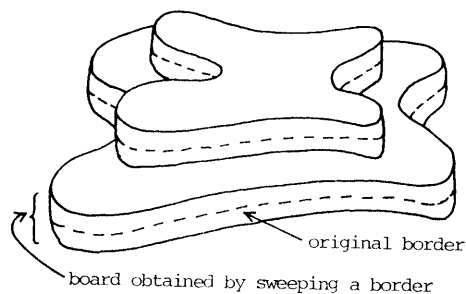


図1. 輪郭線挿引法

### 2.2 面素構成法

上下に隣接する断面の輪郭線の節点からグラフ処理の手法を用いて、何らかのコスト最小の三角形面素を構成していくことで3次元表面形状を得る(図2)。コストとしてKeppelら<sup>(6)</sup>は三角形の体積を最小にする関数を用い、Fuchsら<sup>(7)</sup>は面積を用いている。またChristiansenら<sup>(8)</sup>は上下の断面を接続する辺の長さが最小となる三角形を選んでいくことでコスト計算を簡略化している。これらはいずれも表面形状をかなり滑らかに表現するものであるが、輪郭形状が隣接する断面で大きく異なる場合や、複雑な分岐をする場合には対応できない。Linら<sup>(9)</sup>は隣接断面での輪郭を力の場を用いて歪ませることにより新たな仮想輪郭を生成して、輪郭が滑らかにつながる手法を報告している。

ところで人体形状には概して円筒形あるいは球形の部分が多い。著者ら<sup>(10)</sup>は輪郭を円柱座標系あるいは極座標系により表現し、頭蓋表面のより自然な形状を簡易的に生成している(図3)。また、これは手術後に予想される皮膚表面の生成にも有効に用いられる(図4)<sup>(11)</sup>。

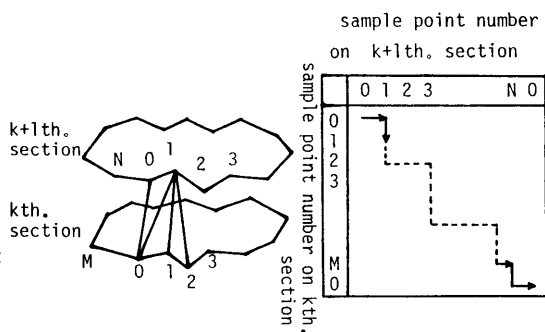


図2. 三角形面素構成

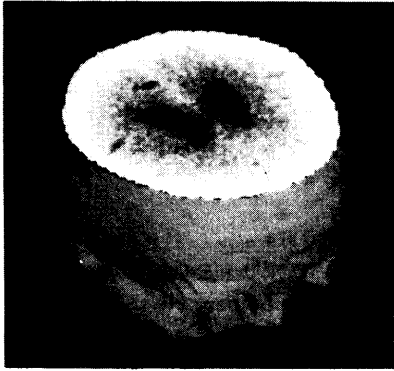


図3. 円柱座標系表現による頭蓋表面像

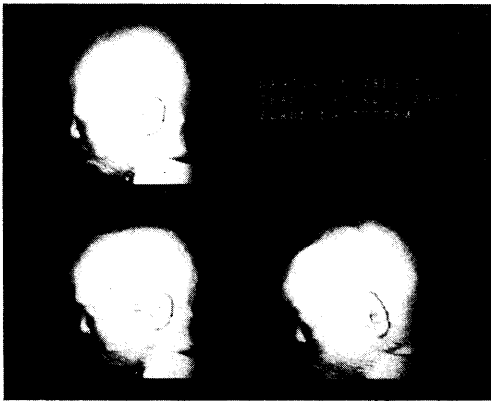


図4. 皮膚表面像  
 左：術前  
 右：術後  
 上：想像  
 下：現実

### 2. 3 ボクセル法

領域抽出後の形状は各断面上での画素の集合で与えられる。各画素を断面間隔の高さをもつ直方体（ボクセル）として考えると、新たな面素を構成することなくボクセルそのものから3次元形状を構成することができる<sup>(12)</sup>。ボクセル法は簡素な方法であるが、どのような複雑な形状も表現できることから、4章で示すように現在さまざまな描画法が工夫され、画質も飛躍的に向上している。本来、ボクセル法は原データの空間解像度が低くなるとボクセルが大きくなり、ボクセル固有の形状が表示像上で目立ってしまうという欠点がある。

Lorensenら<sup>(13)</sup>はボクセルの局所的な位置関係から表面を構成する微小面を推定し、微細な形状をより忠実に得る方法を提案している。隣接するボクセルの0と1の値によって256種類の面の組合せができるがトポロジ的に同一なものも存在し、結局14種類となる（図5）。多角形（ポリゴン）の表示が高

速に行えるグラフィック・エンジンを搭載したワークステーション等での実現を考えた場合に本方法は有効である。

以下3章、4章では、各々ボクセル法により記述された形状に対する投影、描画の諸手法について概説する。

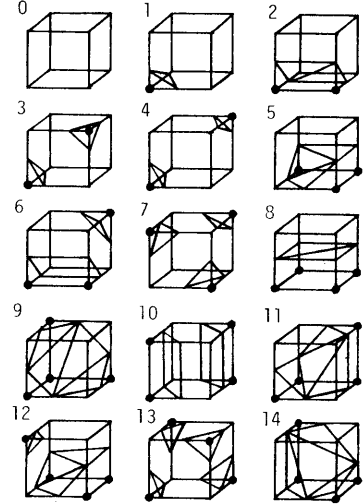


図5. ボクセルの位置関係による微小面

### 3. 投影

一般に3次元物体をスクリーンに投影するとき隠面処理をする必要がある。ボクセル型のデータに対してはスクリーン上の画素から視線方向に並ぶボクセル群までの距離（Z値）を比較して、各画素について視点に最も近いボクセルの面を投影することにより、隠面処理を行うZ-バッファ法を用いることが多い<sup>(3)・(14)</sup>。しかしながら、表示対象を構成するボクセル数は一般に非常に多く、3次元像を生成する際にはこの投影変換に最も時間がかかる。したがって、投影処理の高速化は重要な問題となっている。

Artzy<sup>(15)</sup>はボクセルを構成する6つの面のうち視点方向から可視となる面は常に3つに限定されるので、すべての面を投影する必要がないことを指摘した。

Back-to-front アルゴリズムは視点の位置から判断して視点から最も遠いボクセルから順に投影を行えば自然に隠面処理が行えることを利用したもので、Z-バッファによる比較演算を不必要とした<sup>(16)</sup>。また、ここでは逆に視点に近い順にボクセル投影を実行し、スクリーン上の画素のうち既にボクセルが投影されているものについては以後の投影は行わないことにより比較演算をなくすFront-to-back アルゴリズムも提案されている。

Frieder ら<sup>(17)</sup>はボクセルの辺の長さがスクリーンの画素間隔と等しいかあるいは短いとき、ボクセルの中心点のみを投影することにより面の塗りつぶしをすることなく投影点間で隙間が生じないことを指摘した。また、ルック・アップ・テーブルを利用することにより投影計算を高速に行った。一般に、ボクセル $V(I, J, K)$ のスクリーン座標 $(X, Y, Z)$ への変換式は式(1)で表される。

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} L1 & L2 & L3 \\ M1 & M2 & M3 \\ N1 & N2 & N3 \end{pmatrix} \begin{pmatrix} I \\ J \\ K \end{pmatrix} + \begin{pmatrix} I0 \\ J0 \\ K0 \end{pmatrix} \quad (1)$$

但し、行列は拡大・縮小および回転を含んだ変換行列、ベクトル $(I0, J0, K0)$ は投影の際の平行移動量である。ここで、ある断面に対して $K$ は固定される。したがって、 $L3 \cdot K, M3 \cdot K, N3 \cdot K$ は各断面について1度だけ計算すればよい。I, J方向に対しては、すべての可能な $L1 \cdot I, M1 \cdot I, N1 \cdot I, L2 \cdot J, M2 \cdot J, N2 \cdot J$ を6つの1次元配列に記憶し、必要なときにI, Jをインデックスとして取り出すことにする。原画像の大きさが $N \times N \times N$ のとき、テーブルを用いない投影方法では積算回数は $N^3$ 回必要であるが、この方法では $9 \times N$ 回の乗算を実行すればよい。

横井ら<sup>(18)</sup>は上述のFriederらの方法を更に改良して、より高速な投影変換法を開発した。すなわち、各ボクセルに対する式(1)の変換式において逐次計算を行うことにより積の演算回数を減らし、更に1次元配列を不要にした。いま、ある断面 $K$ について $L3 \cdot K, M3 \cdot K, N3 \cdot K$ が既に求まっているとすると、これを用いて次の断面での値を次式で求める。

$$L3 \cdot (K + \Delta K) = L3 \cdot K + L3 \cdot \Delta K \quad (2)$$

$$M3 \cdot (K + \Delta K) = M3 \cdot K + M3 \cdot \Delta K \quad (3)$$

$$N3 \cdot (K + \Delta K) = N3 \cdot K + N3 \cdot \Delta K \quad (4)$$

但し、 $\Delta K$ はボクセルの $K$ 方向の長さである。したがって、始めに増分 $L3 \cdot \Delta K, M3 \cdot \Delta K, N3 \cdot \Delta K$ を一度計算しておけば、これらの値は常に加算で求められることになる。また、列方向、行方向についても同様のことが行える。結局本方法では、初期値及び変化分を前もって計算するのに乗算を各々9回だけ実行しておけば、以後各値は、テーブルを用いることなく加算によるのみ順次求められる。本方法は3次元形状を構成するボクセルが空間的につながっているとき有効に働くが、通常このような場合が多い。この特徴を利用してTrivedi<sup>(19)</sup>はボクセルデータをランレングス表現することにより、データの圧縮と共に投影変換の高速化を行った。

#### 4. 描画

断面の系列で表現されている3次元形状は2.3節で述べたボクセルの集合体として考えられる。こ

のボクセルを3章で述べた方法でスクリーンに投影して像を生成するが、このままでは対象物体の形状とは別のボクセル固有の小さな凹凸が表面に現れてしまうという問題が生ずる。そこで、できるだけ物体本来の形状を忠実に表現する描画アルゴリズムが考案されてきた。

#### 4.1 乱反射モデル

Hermanら<sup>(12)</sup>により最初にボクセル表示法が提案された際に用いられた。これはボクセルの各面をスクリーンに投影したときにできる多角形内の各画素を乱反射面を想定した輝度計算により表示するものである。実際の計算式は式(5)で与えられる。

$$I(\theta) = \alpha \{ \cos(\theta/n) \}^p \quad (5)$$

ここで、 $I(\theta)$ は画素に与える輝度値、 $\theta$ は光源方向と表面での法線方向とのなす角、 $\alpha$ はディスプレイ装置により決まる最高輝度値に関するパラメータ、 $n$ と $p$ はスムージングのパラメータで例えば $n = 2, p = 0.6$ が用いられる。

また、表面上での微小な凹凸を軽減するため画面上で2次元の平滑化フィルタ処理を行っている。

#### 4.2 Depth coding

画素に与える輝度値を式(6)に示すように画素とボクセルとの距離 $d$ に比例した値を用いて表示する<sup>(20)</sup>。

$$I(d) = \alpha (R - d) \quad (6)$$

ここで、 $R$ はボクセルが存在すると考えられる最も遠い距離である。

この方法はボクセル形状が見えなくなり、表面形状が滑らかになるということで一時期よく用いられた。しかし、物体本来の表面の微妙な形状も消失してしまう欠点があるため、現在では主に他の輝度計算法と組み合わせて遠近感の効果を出すために用いられている。

#### 4.3 Gradient shading

3章で求めたZ-バッファから物理空間における物体表面の法線方向を推定して輝度計算を行うものである<sup>(21)</sup>。Z-バッファはスクリーンあるいは視点から表面までの距離(Z値)を記憶しており、表面は次式で表現される。

$$z = z(x, y) \quad (7)$$

この場合、表面上の任意の点での法線は次のベクト

ルで与えられる。

$$(\partial z / \partial x, \partial z / \partial y, -1) \quad (8)$$

この  $x, y$  成分はデジタル画像上では各々左右、上下の差分と考えられる。いま、 $\delta_r$  を現在の画素と右の画素との差分、 $\delta_b$  を左の画素との差分とすれば  $\partial z / \partial x$  は以下のように求められる。

$$\frac{\partial z}{\partial x} = \frac{W(|\delta_b|) \cdot \delta_b + W(|\delta_r|) \cdot \delta_r}{W(|\delta_b|) + W(|\delta_r|)} \quad (9)$$

ここで  $\delta \geq 0$  について、

$$W(\delta) = \begin{cases} 1 & \delta \leq a \text{ のとき} \\ \varepsilon & \delta \geq b \text{ のとき} \\ (1-\varepsilon)/2 + (1-\varepsilon)/2 \cdot \cos((\delta - a)\pi / (b - a)) & \text{otherwise} \end{cases} \quad (10)$$

但し、

$$a = 2.0$$

$$b = 5.0$$

$$\varepsilon = 10^{-5}$$

$\partial z / \partial y$  についても同様である。

重み関数  $W$  はスムージング効果を含んでおり、以上の手順より得られる法線ベクトルを用いることで、ボクセル固有の凹凸をなくしつつ物体本来の表面形状をある程度忠実に表現する方法として現在よく用いられている。著者ら<sup>(18)</sup>はGradient shadingを行う前にZ-バッファに対して2次元平滑化フィルタ処理をすることで画質がより向上することを報告している。本方法による表示例を図6に示す。

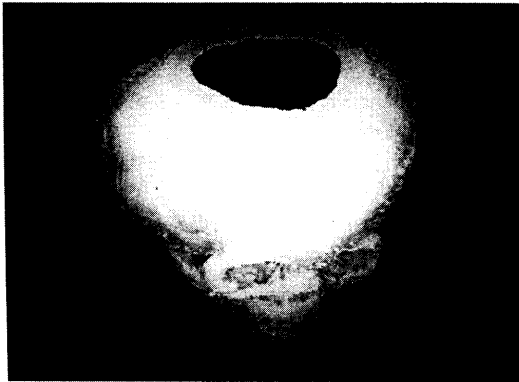


図6. Gradient shading法による頭蓋像

#### 4.4 Normal-based contextual shading

物理空間におけるボクセルの局所的な位置関係から表面での法線ベクトルを推定して輝度計算を行う

方法で、2次元画像であるZ-バッファからの推定に比べて観察方向に対して不変な法線推定が可能なおとこに大きな特徴がある<sup>(22)</sup>。

物体表面を構成するあるボクセルの1つの面上の点  $p$  を考える。物体表面を局所的に見た場合、点  $p$  を通る互いに直行するベクトルを各々  $\mu, \nu$  とすれば、点  $p$  での法線ベクトル  $\omega$  は  $\omega = \mu \times \nu$  (ベクトル  $\mu$  と  $\nu$  の外積) として与えられる (図7)。ここで、隣接するボクセル面の位置関係は  $\mu$  方向のみを考慮した場合、図8のように9通りとなる。また、ベクトル  $\mu$  は図からも明らかなように5通りに制限される。 $\nu$  方向についても同様のことが言え、結局隣接ボクセル面の局所的な位置関係により法線ベクトルは25通りの可能性があることになる。本来81通りのボクセル面の位置関係があるにも関わらず法線ベクトルは25通りに制限されることになるが、表示結果の画質はよい。

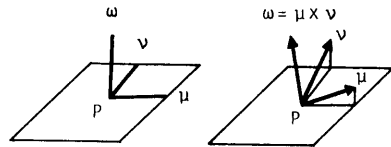


図7. 法線ベクトル  $\omega$

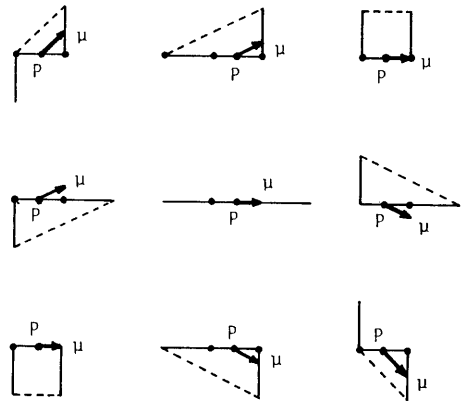


図8. 5種類のベクトル  $\mu$

#### 4.5 Grey level gradient shading

物体表面での法線ベクトルを推定するのに、小さい値処理後の2値画像であるボクセル集合から求めるのではなく、元のCT値を持ったボクセルから推定することでダイナミック・レンジを広げ、より正確な推定をしようとする方法である<sup>(23)</sup>。この基本的な考え方は、隣接するボクセル間でCT値の差が大きいときに何らかの境界面があるという事実を立脚している。具体的な計算は次式で行われる。

$$I(i, j) = \alpha \cos \left[ \frac{-1 - \sum_{m=-1n=-1}^1 \sum_{n=-1}^1 \{g(i+m, j+1, k_T+n) - g(i+m, j-1, k_T+n)\}}{\beta} \right] - \gamma \cdot k_T \quad (11)$$

ここで、 $g(i, j, k_T)$ はボクセル空間での位置 $(i, j, k_T)$ でのボクセルのもつCT値、 $k_T$ は画素 $(i, j)$ からボクセル空間へ投影方向とは逆に探索していったときにあるしきい値以上のCT値をもつボクセルまでの距離を表す。また、 $\alpha, \beta, \gamma$ はスケール・ファクタとして用いられている。

#### 4. 6 Volume rendering

上述の手法を融合して各手法の長所を生かした描画法である<sup>(24)</sup>。基本的には、視点からスクリーン上の各画素を通りボクセル空間に向けて、あるサンプル間隔で透過率 $\xi$ と輝度値 $I$ とを積算してその画素における最終的な輝度値 $I_{FINAL}$ を求めるものである(式12, 図9)。

$$I_{FINAL} = \sum_x \xi(x) \cdot I(x) \quad (12)$$

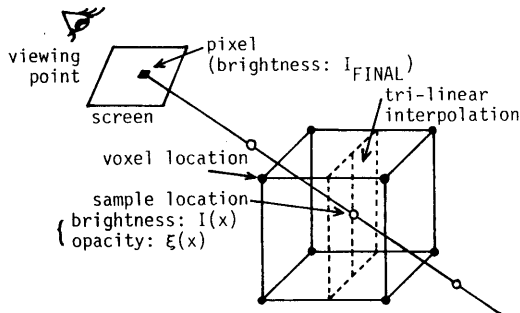


図9 Volume rendering法

本方法の特徴は(1)単一しきい値処理で対象領域を強制的に抽出することなく、自然な境界を表示できること、(2)透過率をうまく用いることにより半透明表示が可能なことである。

サンプル点における輝度値はPhong<sup>(25)</sup>のモデルを基本に次式で計算する。

$$I = \alpha_a + \frac{1}{\beta_1 + \beta_2 \cdot d} [\alpha_d (N \cdot L) + \alpha_s (N \cdot H)] \quad (13)$$

ここで、 $\alpha_a, \alpha_d, \alpha_s$ は各々コンピュータ・グラフィックスで通常用いる環境光、拡散光、正反射光成分の割合を表す。 $d$ は視点からサンプル点までの距

離を表し、 $\beta_1, \beta_2$ はDepth codingのためのパラメータである。 $N, L$ はサンプル点での法線ベクトルと光源ベクトル、 $H$ は正反射光を計算するためのベクトルで視線方向ベクトル $V$ により次式で求める。

$$H = \frac{V + L}{|V + L|} \quad (14)$$

法線ベクトル $N$ はサンプル点を囲むボクセルのもつ密度(この場合はCT値)の差分により次式で推定する。

$$N = \frac{1}{2} \left[ \begin{array}{l} \{g(i+1, j, k) - g(i-1, j, k)\}, \\ \{g(i, j+1, k) - g(i, j-1, k)\}, \\ \{g(i, j, k+1) - g(i, j, k-1)\} \end{array} \right] \quad (15)$$

透過率 $\xi$ は、あるCT値 $g_0$ をもつボクセルに対して $\xi_0$ を与え、 $g_0$ から次第に離れると $\xi_0$ も減衰するような関数で定義する。

#### 5. 操作

CT断面系列から3次元形状を再構成して表示することにより、形状把握が容易に行われるようになると、今度は本来の形状を再現している投影像をディスプレイ装置から会話的に操作して何らかの変形が加えられないかという要求が起こってきた。このような機能は医学分野において、形状に重点がおかれる手術の術前における計画立案等に有効である。特に先天的あるいは事故などの後天的な理由により、形状が著しく変形している頭蓋骨を修復する頭蓋形成外科<sup>(26), (27)</sup>への応用に関心が高まり、手術のシミュレーションや評価に利用可能なコンピュータ・グラフィックスや画像処理の諸手法の開発が行われている。

Vannierら<sup>(28)</sup>は頭蓋の変形の程度を定量的に計測することを可能にし、実際に臨床で用いている。現在では市販のCT装置にも簡単な計測機能が付加されているものが多い。

形状の操作は以下のような手法に大きく分けられる。

- (1) CT断面上での会話型操作
- (2) 基本的幾何学形状による操作
- (3) スクリーン上からの会話型操作
- (4) 手術シミュレーション
- (5) 異なる隣接する骨の自動分離

##### 5. 1 CT断面上での会話型操作

CT断面から切断したい領域の輪郭線を会話的に入力してその領域を除いた3次元像を生成するもの

で, Bloch ら<sup>(28)</sup>が報告している. これは元の3次元物体から仮想的に一部分を削除した像を生成するという最も初期の方法で, 興味深いものであるが, 領域指定を各CT断面上で行わなければならないという問題がある.

CT断面上での処理の応用として, Brewsterら<sup>(29)</sup>は正常な頭蓋形状がほぼ左右対称であることを利用して, 左右非対称な異常形状の頭蓋について, 仮想的に正常な半領域の鏡像を異常領域に用いることで正常な形状を推定している(図10). これは頭蓋形成手術計画への一つの指針になることから意義深いものである.

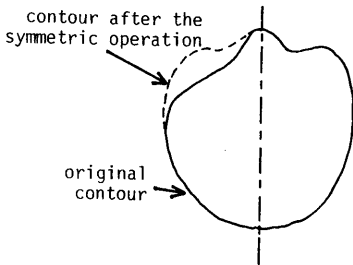


図10 頭蓋の左右対称性を利用した正常形状推定

### 5.2 基本的幾何学形状による操作

Brewsterら<sup>(29)</sup>やTrivedi<sup>(30)</sup>は球や角柱といった基本的な3次元幾何学形状及びそれらを適当に組み合わせた形状によって物体の切断を行う方法を提案した. これは2値の論理演算として実現される. 例えば, 元の物体形状が1で表されているような3次元物体Oにある幾何学形状が同じく1で表されている物体Mを付加するとき, この操作後の形状O'は次式で表現される.

$$O' = O \cup M \quad (16)$$

同様にOからMを削除する操作は次式のようなになる(図11).

$$O' = O \cap \overline{M} \quad (17)$$

この場合のMは他の複数の幾何学形状による論理演算で定義されていてもよく, 例えば幾何学形状A, Bの論理和として,

$$M = A \cup B \quad (18)$$

と表現することが可能である.

組合せによって望む切除領域が決定されない場合は, 切断領域を含むすべてのCT断面について2次元幾何学形状(例えば円や長方形)による切断操作を実行し, 結果的に3次元的な切断を得ることもできる. 但し, 領域が多くなるCT断面にわたって存在する場合には会話的処理に要する時間は長くなる.

この種の形状操作は工業用CAD等においては用いられているが, ボクセル型の要素で構成されるCT画像による形状への操作は従来行われていなかった.

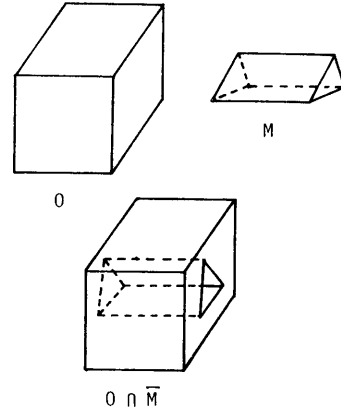


図11 幾何学形状による操作

### 5.3 スクリーン上からの会話型操作

会話的に自由な操作を実現しようとする方法で現在様々な研究・開発が進んでいる. 現実的な会話型操作として, Hermanら<sup>(31)</sup>は4.2節のようなCT断面上での領域指定法の他に, ディスプレイ上で削除領域を囲むような切断線を入力し, その切断線を断面にもつようなスクリーンに垂直な面で物体を二分する手法を示唆している.

Hermanらの提案を受けて, Brewsterら<sup>(29)</sup>やTrivedi<sup>(30)</sup>はディスプレイ装置から会話的に描かれた曲線を断面にもつスクリーンに垂直な切断面により切断を行う機能を開発した.

Fellinghamら<sup>(32)</sup>は臨床的に有用なCT像の3次元処理装置CEMAX-1000を開発し, 既に多くの症例に対し商業的に利用している. これは主に以下のような機能を有する.

- (1) 長さ, 角度, 体積の測定
- (2) 移植シミュレーションを目的とした人工骨との合成表示
- (3) スクリーン上での簡単な手術シミュレーション
- (4) カスタム人工骨のデザイン
- (5) 手術計画や放射線治療計画を目的としたMR

等その他のCT像との合成表示

手術シミュレーションは単一の骨片を輪郭により切断する機能のみであり、実際行われている手術に適用するのは難しい。また、CT断面系列から人工骨を設計する場合には久野ら<sup>(33)</sup>が報告しているように断面間の補間が問題となる。他のCT像との合成は各々の特徴を生かした画像が得られることから有用であるが、位置合わせをうまく解決しなければならない。CEMAX-1000はこのような問題を含んでいるものの、商業的、臨床的に実用されている点が評価できる。

5. 1節から5. 3節までの方法は撮影されたCT像から特定の領域を抽出して表示するのに用いられる。この場合の領域分割は一般に画像処理で行われる濃度値特性の異なる部位での分割ではなく、例えば骨盤と大腿骨との分離のように濃度値構造は骨の特性値(X線CTの場合はX線吸収量)として等しいが、解剖学的に異なる部位の分割を意味する。(骨盤、大腿骨の自動分離については5. 5節に述べる。)このような切除処理により、臨床的に重要であるが周りの部位により隠されているような部分を観察することが可能となる。よって、これらの機能は手術計画への利用というよりも疾患の診断に有効なものと考えられる。

#### 5. 4 手術シミュレーション

著者らは実用的な手術シミュレーション・システムの開発を形成外科医、整形外科医、および放射線科医と共に行っている<sup>(11)、(34)、(35)</sup>。その最も重要な機能はラベルにより各ボクセルを管理することによって、スクリーン上から3次元的に任意形状の骨切断と移動の操作が可能なことである。以下では本機能を実現するための処理手法について簡単に紹介する。

##### (1) 形状表現とボクセル管理

骨領域を表すボクセルを単純しきい値処理により抽出し、それらの座標値I, J, Kを各々3つの1次元配列 $L_I, L_J, L_K$ に記憶していく。

$$\begin{aligned} L_I(k) &= I \\ L_J(k) &= J \\ L_K(k) &= K \end{aligned} \quad (k=1, 2, \dots, n) \quad (19)$$

但し、nはボクセルの総数である。

ここで、切断・加工された骨片を各々個別に管理するため、各ボクセルにその属する骨片を示すラベルを持たせ、1次元配列 $L_L$ に記憶する。したがっ

て、切断によって新たな骨片が生成される毎に新しいラベルが生成され、対応するボクセル集合にこの新しいラベルが割り当てられることになる。また、ディスプレイ上に投影像を表示するための輝度値用2次元配列Aと共に、切断・移動の操作をラベルで管理するためのラベル用2次元配列B(投影面上の座標(X, Y)の画素が画面に表示されている骨片に属するとき、その画素のラベルをX行Y列要素に入れる。)を用意する。

##### (2) 表示と操作

いま、k番目のボクセル $V_k(I=L_I(k), J=L_J(k), K=L_K(k))$ の投影を考える。まず、このボクセルに現段階で付いているラベル $L_k$ を $L_L(k)$ から求め、このラベルに対応する移動・回転による座標値の変換を行う。同一のラベルをもつボクセル集合は切断された一つの骨片を表すが、これは画像上で指示された移動・回転の操作を受けている。この操作を受ける毎にそのラベルに対応する移動・回転を行う変換行列の係数が更新されている。そこで、ラベル $L_k$ の骨片に対する変換係数 $a_{pq} = a_{pq}(L_k)$ 、 $b_p = b_p(L_k)$ を式(20)に代入することにより、ボクセル $V_k$ の変換後の座標値(I', J', K')を求める。

$$\begin{pmatrix} I' \\ J' \\ K' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} I \\ J \\ K \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \quad (20)$$

この座標値(I', J', K')を投影変換して4. 3節で述べたGradient shadingにより表示を行う。

スクリーンに垂直な平面による切断はディスプレイ上に表示されている同一ラベルを持つ画素集合を各々別のラベルを持つ2つの画素集合に分けることにより実行する。以下に具体的な処理手順を述べる。

##### 【切断処理手順】

- ① ヘア・カーソルにより切断したい骨領域を選択する。この操作により、切断したい骨片のみが再表示され、切断処理が可能となる。この骨片を表すラベルを $L_0$ とする。
- ② 次に、切断線の点列を入力することにより切断が行われる。このとき2次元配列Bを予め初期化しておき、点列が入力される毎に、点列間を連結する点集合 $\{(X_c, Y_c)\}$ を求め、全ての $\{(X_c, Y_c)\}$ に対する要素B(X\_c, Y\_c)に切断線を示すラベルを記入し、この切断線により画像を2つの領域に分割する。
- ③ 領域の分割後、関心のある領域内で1点を指定する。これにより、この点を含み切断線と境界線と



で囲まれた閉領域の点集合  $\{(X_R, Y_R)\}$  を求め、2次元配列  $B$  の対応する要素  $B(X_R, Y_R)$  にこの新たに生成された領域を示す新しいラベル  $L_1$  を記入する。

④ 最後に、手順①で選択された骨片のラベル値  $L_0$  を持つ全てのボクセルをスクリーン上に再投影し、対応する2次元配列  $B$  の要素に手順③で生成した閉領域を示すラベル  $L_1$  が書き込まれていればそのボクセルのラベルを  $L_0$  から  $L_1$  に更新する(図12)。

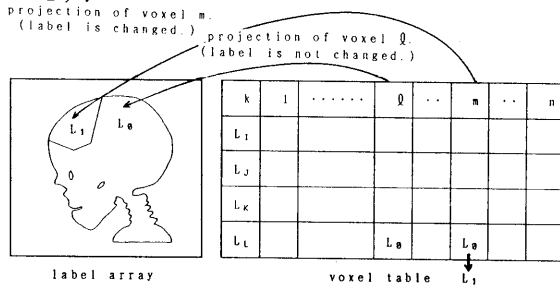


図12 ラベル管理による切断処理

3次元的に任意形状の切断は、以上のようなラベル管理されたボクセル集合を2分割する操作をいくつかの方向から実行することにより可能となる。

また、移動操作として平行移動と回転移動が可能である。平行移動の方向はディスプレイ画面に対して平行であり、回転移動の回転軸は画面に対して垂直である。これらの移動操作も複数の方向からの移動を組み合わせることにより任意の移動が実現できる。上述の機能により現在実際に行われているかなり複雑な頭蓋形成手術のシミュレーションを形成外科医が行った結果を図13に示す。

なお、これらの機能は頭蓋形成手術シミュレーションを対象として開発されているが、工学的観点から見ると3次元デジタル画像に対する複雑な操作(切断・移動・回転)をコンピュータ・グラフィックスによる表示画面からの対話的指示で実行するための基本機能の実現である。したがって、本システムは今後他の部位の手術計画立案や3次元画像解析システムの骨格部分になり得ると考えられる。実際、股関節部の手術のシミュレーション<sup>(36)</sup>やミイラの解析<sup>(37)</sup>への応用などの例がある。更には3次元形状をボクセル型データで表現して図形操作するCAD等にも応用の可能性があると考えられる。

その他にも手術シミュレーションに用いられることを目的とした会話型操作法の開発がいくつか行われている。石井ら<sup>(38)</sup>は脳外科手術を対象にドリル切削のように頭蓋骨を切開して、内部の疾患部分を観察する機能を報告している。また、同様な手法を

整形外科領域に用いる研究が周藤ら<sup>(39)</sup>によって報告されている。

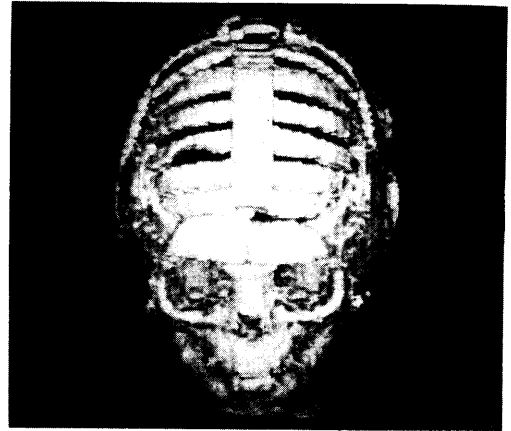


図13 手術シミュレーション例

### 5.5 異なる隣接する骨の自動分離

骨盤と大腿骨のように本来わずかに離れている骨もCTの解像度の制限によりつながってしまうことが多い。このような場合、各CT断面上で単純にしきい値処理を用いて二値化するだけでは分離できない場合が多い。したがって人手によって分離・抽出することになるが、断面の数が多くなるとこれには多大な労力を必要とする。また、これらの骨は複雑な形状をしており、上述の手法で会話的に分離操作することは難しい。そこで3次元画像処理の手法により自動的に分離する方法が曾山ら<sup>(36)</sup>により報告されている。3次元デジタル画像データとして与えられる処理対象に対して3次元画像処理はCT断面上での2次元画像処理と比べ以下の点で優れている。

(1) エッジ検出において、3次元差分フィルタではCT断面に対して平行に存在するエッジも他の方向と同様に出力されるが、2次元差分フィルタでは出力されない。

(2) 連結成分のラベル付けにおいて、3次元的に連結している成分も2次元ラベリングではしばしば別のラベルになってしまうことがあるが、3次元ラベリングでは1個の連結成分として同一ラベルで扱われる。

整形外科においては、本来他の領域により隠されている領域を自動的に分離して観察可能にすることが意義は大きい。最近、3次元画像処理手法を用いた骨盤・大腿骨の自動分離をYlä-Jääskiら<sup>(40)</sup>も報告している。

