

# Fast Level Set Method を用いた複数移動物体の実時間追跡

九州大学 岩下友美 倉爪亮 辻徳生 原健二 長谷川勉

## Realtime Tracking of Multiple Human Bodies using Fast Level Set Method

Kyushu University Yumi Iwashita Ryo Kurazume Tokuo Tsuji Kenji Hara  
Tsutomu Hasegawa

**Abstract**— This paper presents an efficient implementation technique for the level set method(LSM) named the Fast Level Set Method (FLSM). Various applications based on the LSM have been presented including motion tracking and 3D geometrical modeling. However, the calculation cost of reinitialization and updating of the implicit function is considerably expensive as compared with the cost of conventional active contour models such as "Snakes". To tackle this problem, we have proposed an efficient algorithm of the LSM named the FLSM. This paper introduces some experiments of realtime tracking of moving objects in video images and 3D stereo range images using the FLSM.

**Key Words:** Motion tracking, 3d shape reconstruction, level set method, stereo range images

### 1. はじめに

動的輪郭モデル (Active Contour Model) の代表的な手法である Snakes<sup>1)</sup> は、ノイズに対して頑強な 2 次元画像の境界追跡法である。この手法を 3 次元に拡張した deformable surface<sup>2)</sup> は、3 次元点の集合からそれを内包する閉曲面を安定に抽出するための手法として、幾何モデリングや領域追跡の分野で研究が進められてきた。しかし、従来の動的輪郭モデルに共通して、境界の分離や結合など位相変化への対応が困難であるということが問題とされていた。

これに対し、本質的に位相変化が可能な動的輪郭モデルの手法として Level Set Method(LSM)<sup>3) 12)</sup> が提案されており、移動体追跡<sup>4)</sup> や 3 次元幾何モデリング<sup>5)</sup> などの様々な分野で用いられている。この手法は、検出する境界を 1 次元高い補助関数のゼロ等高面とみなし、境界の進行条件である偏微分方程式を数値的に解いて補助関数の形状を変更し、そのゼロ等高面を次々に検出することで境界形状を動的に制御する手法である。

この手法を計算機内で実現するには、一般に空間を均一ボクセルで離散化し、境界を含むボクセルとその周囲のボクセルに設定された値 (通常は境界からの距離や境界の予想到達時刻) を、境界の曲率や、例えば画像濃淡値などのボクセル周辺の状況から計算される成長速度を用いて、ある設定した積分時間間隔で繰り返し更新していく。しかし更新とともに積分誤差も蓄積するため、安定な解を得るには、一定回数更新後に各ボクセルの補助関数の値を再計算し、以降の計算の初期値として設定する必要がある。この各ボクセルへの初期値の設定、及び各更新ごとの成長速度の計算には多くの計算量が必要であり、計算の高速化が大きな課題である。

この問題に対し、Level Set Method の高速化手法が提案されており、代表的なものとして、Narrow Band Method(NB)<sup>6)</sup> と Fast Marching Method<sup>7)</sup> が挙げられる。NB では、空間全体に対して補助関数を計算するのではなく、ゼロ等高面に近い領域だけに対して補助関数を計算することで処理の高速化を図ったものである。しか

し、後で述べる拡張成長速度場を用いる場合には、NB 内の各ボクセルごとに最近傍境界を探索しなければならず、計算コストは依然として高い。一方、Fast Marching Method はより計算量の少ない効率的な手法であるが、曲面の膨張と収縮が同時に実現できないという制限がある。

そこで、我々はこれまでに高速で安定な Level Set Method の解法として、拡張成長速度場の利用と高速な拡張成長速度場の構築を特徴とする新たな Fast Level Set Method(FLSM) を提案した<sup>12)</sup>。さらに FLSM の応用例として、ビデオ画像上の移動物体のリアルタイム追跡、およびステレオカメラを用いた人体概形の 3 次元リアルタイム追跡を試みた。

本報告では、まず第 2 章では Osher,Sethian らによって提案された Level Set Method を紹介する。次に第 3 章で、拡張成長速度場の高速な構築手法である Fast Narrow Band Method(FNB) と、それを用いた高速で安定な Level Set Method である Fast Level Set Method の概要を示す。さらに、第 4 章でビデオ画像上の移動物体のリアルタイム検出、追跡実験の様子を、第 5 章でステレオカメラを用いた人体の 3 次元概形形状のリアルタイム復元、追跡実験を紹介する。第 6 章はまとめである。

## 2. Level Set Method とその高速化手法

### 2-1 Level Set Method

Level Set Method は、Osher,Sethian ら<sup>3)</sup> によって提案された位相変化が可能な動的輪郭モデルである。例として、3 次元 xyz 空間内での LSM を用いた境界追跡法について説明する。まず、時刻  $t$  での境界位置を  $C(p, t)$  とする。ただし  $p = (p_x, p_y, p_z)$  である。この境界に含まれる点  $p$  は、移動速度  $F(\kappa)$  で法線方向  $N$  に移動していると考えられる。ここで  $\kappa$  はその点での境界の曲率であり、 $F$  を成長速度という。これを式で表すと、

$$C_t = F(\kappa)N \quad (1)$$

$$C(p, 0) = C_0(p) \quad (2)$$

となる。この問題は差分方程式を利用したラグランジュ法で解くことができるが、トポロジーの変化には対応しにくいという問題がある。

そこで新たな補助関数  $z = \Psi(x, y, z, t)$  を導入し、境界位置  $C(p, t)$  はその関数の一部、すなわち  $z = \Psi(x, y, z, t) = 0$  を満たす  $\Psi$  で表されると考える。ここで、点  $p(t)$  が境界  $C(p, t)$  上の点であると仮定すると、これが常に  $\Psi(x, y, z, t)$  のゼロ等高面である条件は、

$$\Psi(p(t), t) = 0 \quad (3)$$

で表される。これを時間で偏微分すると、

$$\Psi_t + \nabla \Psi(p(t), t) p_t = 0 \quad (4)$$

となる。また曲線上の単位法線ベクトルは、

$$N = \frac{\nabla \Psi}{|\nabla \Psi|} \quad (5)$$

で表され、さらに移動速度  $F$  は境界  $C(p, t)$  の法線方向速度であるから、

$$p_t \cdot N = F \quad (6)$$

となる。これにより式 (4) は以下のように変換される。

$$\Psi_t = -F(\kappa) |\nabla \Psi| \quad (7)$$

$$\Psi(C_0(p), 0) = 0 \quad (8)$$

このように境界  $C(p, t)$  を直接的に移動する代わりに、補助関数  $\Psi(x, y, t)$  を更新し、 $\Psi(x, y, z, t) = 0$  として境界を求めることで、トポロジーの変化に対応した領域追跡が可能となる。実際に空間上の点  $i, j, k$  において補助関数  $\Psi_{ijk}$  を更新するには、以下のいわゆる Upwind Scheme が使われることが多い。

$$\begin{aligned} \Psi_{ijk}^{n+1} &= \Psi_{ijk}^n - \\ &\Delta t (\max(F_{ijk}, 0) \nabla^+ + \min(F_{ijk}, 0) \nabla^-) \end{aligned} \quad (9)$$

$$\begin{aligned} \nabla^+ &= (\max(D_{ijk}^{-x}, -D_{ijk}^{+x}, 0)^2 \\ &+ \max(D_{ijk}^{-y}, -D_{ijk}^{+y}, 0)^2 \\ &+ \max(D_{ijk}^{-z}, -D_{ijk}^{+z}, 0)^2)^{\frac{1}{2}} \end{aligned} \quad (10)$$

$$\begin{aligned} \nabla^- &= (\max(D_{ijk}^{+x}, -D_{ijk}^{-x}, 0)^2 \\ &+ \max(D_{ijk}^{+y}, -D_{ijk}^{-y}, 0)^2 \\ &+ \max(D_{ijk}^{+z}, -D_{ijk}^{-z}, 0)^2)^{\frac{1}{2}} \end{aligned} \quad (11)$$

ただし、

$$\begin{aligned} D_{ijk}^{+x} &= \frac{\Psi_{i+1,j,k}^n - \Psi_{i,j,k}^n}{h} & D_{ijk}^{+y} &= \frac{\Psi_{i,j+1,k}^n - \Psi_{i,j,k}^n}{h} \\ D_{ijk}^{+z} &= \frac{\Psi_{i,j,k+1}^n - \Psi_{i,j,k}^n}{h} & D_{ijk}^{-x} &= \frac{\Psi_{i,j,k}^n - \Psi_{i-1,j,k}^n}{h} \\ D_{ijk}^{-y} &= \frac{\Psi_{i,j,k}^n - \Psi_{i,j-1,k}^n}{h} & D_{ijk}^{-z} &= \frac{\Psi_{i,j,k}^n - \Psi_{i,j,k-1}^n}{h} \end{aligned} \quad (12)$$

であり、 $n$  は積分回数、 $h$  は離散化幅、 $\Delta t$  は積分間隔である。

3次元距離画像における領域追跡の具体的な実現法を考える。例えば空間上の点  $(x, y, z)$  における時刻  $t$  での  $\Psi(x, y, z, t)$  の値を、その時刻における境界  $\Psi(x, y, z, t) = 0$  からの符号付距離（境界の内側が負、外側が正）とし、また成長速度  $F(\kappa)$  を

$$F(\kappa) = k_I(a - b\kappa) \quad (13)$$

で与えることにする。ただし  $k_I$  は距離データ点の密度勾配に関する項、 $a, b \geq 0$  は定数である。

さて、Level Set Method において各ボクセルの成長速度を決定するために、局所成長速度場 (No extension velocity field) と拡張成長速度場 (Extension velocity field) を構築する2つの手法が提案されている。局所成長速度場とは、各ボクセルにおける補助関数の成長速度を決定するのに、そのボクセル自身の状態、例えば境界かどうかや補助関数の曲率などの情報を用いるものである。一方、拡張成長速度場とは、まず zero level set (補助関数値が0のボクセル) での成長速度を決定し、その他のボクセルでは最も近い zero level set のボクセルの成長速度をコピーして用いるものである。

局所成長速度は、各ボクセル毎に成長速度を決定するため、zero level set が境界に到達し、その部分の進行速度が減少しても、その情報が周辺に伝わらず、周辺の速度には影響がない。そのため、進行を続けるうちに補助関数場が境界外側では密に、内側では疎になる場合がある。さらに境界周辺以外では関数場は変化し続け、停止することはない。一方、拡張成長速度は、zero level set が境界に到達すると、その情報が周囲に伝えられ、周囲の進行速度も減少するため、進行を続けても局所成長速度の場合のように補助関数場が不均一になることがない。Sethian ら<sup>8)</sup> も多くの計算機実験から、拡張成長速度場は解が安定に求まり、最終的に得られる zero level set の形状も高精度であることを示している。

また、Upwind Scheme により補助関数を更新する場合、更新とともに積分誤差も積算されるため、安定な解を得るには一定回数更新後に各ボクセルごとに補助関数の値（一般には現在の zero level set からの距離）を再計算し、以降の計算の初期値として設定する「再初期化」の処理が必要である。

しかし、上記の拡張成長速度場の構築処理や再初期化処理において、各ボクセルで現在の zero level set からの距離を得る際、各ボクセルからの最近傍 zero level set の探索処理を行わなければならない。この計算コストは非常に高く、これが Level Set Method の大きな問題となっている。

## 2.2 Narrow Band Method

Level Set Method の計算コストを削減するために、これまで様々な高速化手法が提案されており、その代表的な手法として、Narrow Band Method(以下、NB) が挙げられる<sup>6)</sup>。一般に境界領域の追跡において、空間全体に対して補助関数を計算する必要はないことから、この手法ではゼロ等高面に近い領域だけに処理を限定することで処理の効率化を図っている。さらにこの手法では計算コスト削減のために、ゼロ等高面が Narrow Band 領域の境界に近付いたときのみ、Narrow Band 領域内で再初期化処理を行う。

### 3. Fast Level Set Method の提案

NB は Level Set Method に比べると高速な手法であるが、依然として計算コストは高い。そこで、我々は i) 最近傍点探索処理をあるルールに基づく単純な数値の上書き処理に置き換えることで、高速に拡張成長速度場を構築する Fast Narrow Band Method (FNB)<sup>5)</sup> と、ii) 補助関数の再初期化処理の高速化と頻繁な再初期化を特徴とする、高速で安定な Level Set Method である Fast Level Set Method を提案した<sup>12)</sup>。以下、これらの手法を紹介する。

#### 3.1 高速な拡張成長速度場の構築法 (Fast Narrow Band Method)

ここでは 3 次元空間での実装法について説明する。まず、断面図が Fig.1(b) のように表される球形の参照マップ球 Fig.1(a) をあらかじめ作成する。これは、原点周辺にあるボクセルを原点からの距離に応じて分類したものである。つまり、原点からの 2 乗距離が  $r$  であるボクセルの集合を  $R_r$  とし、 $r = 0 \sim \delta(\delta + 1)$  に対するリスト  $R_0, R_1, \dots, R_{\delta(\delta+1)}$  をそれぞれ作成する。なお、ここで距離にはボクセル中心間のユークリッド距離を用いることとし、また  $\delta > 0$  は Narrow Band のバンド幅である。また、バンド幅  $\delta$  の Narrow Band 領域は、各 zero level set からの距離を小数点以下で四捨五入した整数値が  $\delta$  以下になるようなグリッドの集合と定義する。これは、 $\delta(\delta + 1) < (\delta + 0.5)^2 < \delta(\delta + 1) + 1$  が常に成り立つことから、zero level set からの 2 乗距離が  $\delta(\delta + 1)$  以下のボクセルの集合ともみなせる。一例として Fig.1 に、バンド幅  $\delta = 3$  における参照マップ球 ( $R_0 \sim R_{12}$ ) の断面を示す。グリッドに書き込まれている数字 ( $r$ ) は、属しているリスト ( $R_r$ ) を示す。

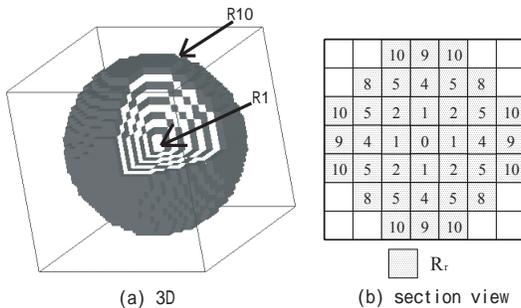


Fig.1 An example of reference map.

次に、作成した参照マップ球を用いて拡張成長速度場を構築する。ただし、zero level set での成長速度は式 (13) 等によりあらかじめ決定されているものとする。まず、リスト  $R_{\delta(\delta+1)}$  を用いて、ある zero level set からの 2 乗距離が  $\delta(\delta + 1)$  であるようなボクセルを選択し、その zero level set に格納されている成長速度の値を選択されたボクセルに仮登録する。この処理をすべての zero level set に対して行う。次に、添字の値を 1 小さくして同じ処理を行い、これを添字の値が 0 になるまで繰り返す。ただし、仮登録の際、異なる値がすでに仮登録されていた場合には、新たな値を上書きすることにする。これにより全ての処理が終了した時には、各ボクセルには最も近い zero level set における成長速度

の値が登録されている。このように、参照マップ球内の距離に応じたリストを利用することで、距離比較を行うことなく代入処理だけで拡張成長速度場が構築できる。以上の手法を Fast Narrow Band Method (FNB)<sup>5)</sup> と呼ぶことにする。

一例として、2 次元  $xy$  平面での拡張成長速度場の構築処理の流れを説明する。Fig.2 にバンド幅  $\delta = 3$  の場合の処理の流れ ((a) → (d)) を示す。Fig.2 (a) は、成長速度の値が  $F_1$  である zero level set に対し、そこからの 2 乗距離が 10 であるボクセルをリスト  $R_{10}$  を用いて選択し、そのボクセルの成長速度を  $F_1$  とした様子である。Fig.2 (b) では、全ての zero level set からの 2 乗距離が 10 であるボクセルに、各々の zero level set に格納されている成長速度の値を登録している。Fig.2 (c) は、成長速度の値が  $F_1$  である zero level set に対し、そこからの 2 乗距離が  $9 (= 10 - 1)$  であるボクセルをリスト  $R_9$  を用いて選択し、そのボクセルの成長速度を  $F_1$  としている。ただし、Fig.2 (b) では、成長速度の値が  $F_1$  でないボクセルに  $F_1$  の値を上書きしている。この処理を繰り返すことで、Fig.2 (d) のように拡張成長速度場が構築できる。

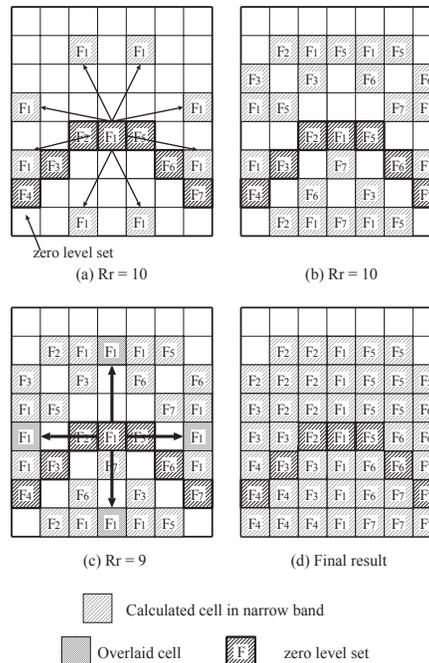


Fig.2 Procedure of constructing extension velocity field.

#### 3.2 参照マップ球の分割と再初期化処理との統合 (Fast Level Set Method)

前項の FNB は書き込む領域を限定することでさらに効率化できる。例えば、ある zero level set (a) の左側に zero level set (b) が隣接している場合 (a) の左側の領域には (b) よりも (a) に近い点は存在しない。同様に (b) の右側の領域には (b) よりも (a) に近い点は存在しない。このように、隣接するグリッドが zero level set かどうかを調べ、その位置関係によって、書き込む領域を限定することができる。

そこでまず、Fig.3 のように参照マップ球を原点からの方向により、扇形の領域を 12 つ、直方体の領域を 6

つ、球を8等分にした領域を8つ、合計26つの領域に分ける。ただし扇形の領域をA領域、直方体の領域をB領域、球を8等分にした領域をC領域と呼ぶことにする。

次に各 zero level set に対し、次の手順により拡張成長速度場を構築する。

1. A領域において zero level set に隣接するボクセルを調べ、そこに他の zero level set があればそのA方向を書き込まない領域とする。また同時に、そのA領域に隣接するC領域も書き込まない領域とする。
2. B領域において zero level set に隣接するボクセルを調べ、そこに他の zero level set の点があるときは、その方向のB領域を書き込まない領域とする。また同時に、そのB領域に隣接するA領域、C領域も書き込まない領域とする。
3. C領域において zero level set に隣接するボクセルを調べ、そこに他の zero level set があればそのC方向を書き込まない領域とする。
4. (1),(2),(3)で残った領域に対し、前項と同様にリストを用いて成長速度を書き込み、拡張成長速度場を構築する。

上記の手法を用いると、書き込みのオーバーラップを減らすことができ、拡張成長速度場を高速に構築できる。

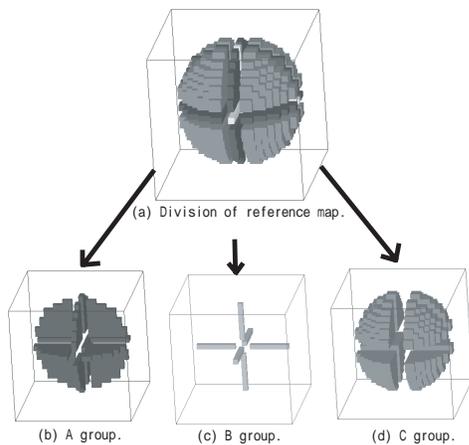


Fig.3 Division of reference map.

さて、Level Set Method により境界を安定に検出するには、一定回数更新後に各グリッドにおいて、現在の zero level set からの距離を再計算し、以降の計算の初期値として設定する再初期化処理が必要となる。

ところが、前項までに提案した拡張成長速度場の構築処理は、各グリッドで現在の zero level set からの距離に応じて成長速度を上書きする処理であり、その過程で距離も同時に上書きすることで、各グリッドに zero level set からの距離を簡単に設定できる。この際、追加される処理は単なるメモリアクセスだけであり、全体の計算量はほとんど変化しない。

拡張成長速度場の計算は各更新時ごとに行われるので再初期化の処理もほとんど計算量を増やすことなく、最大で各更新時ごとに行うことができる。

## 4. ビデオ画像上の移動物体のリアルタイム検出と追跡

次に実際のビデオ画像に対して提案した Fast Level Set Method を適用し、移動物体の検出とリアルタイム追跡実験を行った。使用した画像のサイズ、入力速度はそれぞれ 320x240pixel, 30Hz, 使用した計算機は Pentium IV, 2GHz であり、Fast Level Set Method の処理は約 60Hz で実行されている。また成長速度の最大値は1ピクセル、積分時間幅は4であり、Narrow Band の幅は5ピクセルとした。

実験ではまず背景差分により移動物体の領域を大まかに検出した。検出した移動物体領域の中心から外側へ、Fast Level Set Method により zero level set を進行させて、濃淡値あるいは背景差分値が急激に変化する領域を移動物体の境界として検出、追跡した。すなわち式(13)で用いる輝度勾配項  $k_I$  は、以下の式により濃淡画像  $I(x, y)$  と背景差分画像  $D(x, y) = I(x, y) - I_{org}(x, y)$  の両方から決定した。

$$k_I = \frac{1}{1 + \min(|\nabla I(x, y)|, |\nabla D(x, y)|)} \quad (14)$$

Fig.4 に追跡開始から時間ごとの処理結果を示す。これより、移動物体が画面上を約 100 [pixel/sec.] で移動している場合でも、正確に移動物体の輪郭を抽出し、遅れなく追跡できていることがわかる。また Fig.5 に複数の領域が交差し、分離している場合を示す。この場合も、当初2つの閉曲線で表されていた移動物体の境界が、移動物体が交差したことで1つの閉曲線に統合され、次の時刻で再度2つの閉曲線に分離しており、境界の位相変化に柔軟に対応できている。

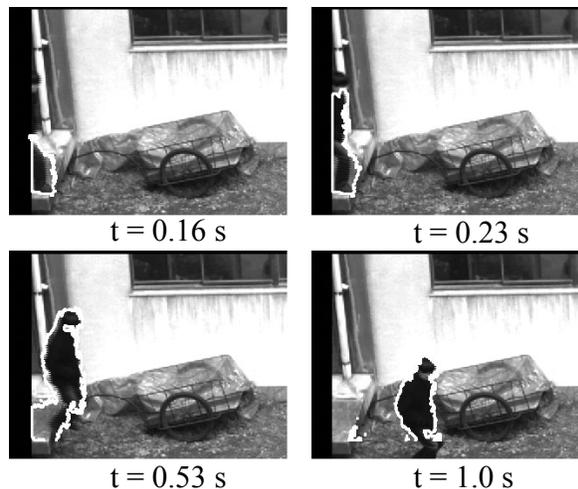


Fig.4 Realtime tracking of single object.

## 5. ステレオカメラを用いた人体の3次元リアルタイム追跡

対象物体全周の3次元形状をテレビカメラを用いて復元する手法として、ステレオ視と視体積交差法<sup>9)10)11)</sup>などが挙げられる。視体積交差法は、任意の視点から対象を撮影し、その時得られる2次元シルエットを実空間に投影した錐体(視体積)を考えると、対象はその



Fig.5 Simultaneous tracking of multiple objects.

錐体中に含まれるという制約条件に基づいている．従って，対象物体の詳細な形状を求めるには多くのカメラを配置する必要があり，またシルエットが正確に切り出されるように背景も工夫する必要がある．また，凹物体やオクルージョンがある場合には，原理的に形状を復元できない．

そこで，本研究ではより少数のステレオカメラ (Point-Grey 社製, Bumblebee) を用い，提案した FLSM を適用して凹物体やオクルージョンがある場合でも人体概形のリアルタイム追跡を行うことができる試作機を構築した．

### 5.1 複数距離画像の統合と FLSM による人体概形の追跡

ステレオカメラから距離画像を獲得した後，space carving の手法を用いて 3 次元のボクセル空間で複数距離画像の統合を行う．まず，各ステレオカメラに対して Fig.6 に示すようなボクセル空間を 1 つずつ設定する．また，各ボクセルは IN (人体の内部)，OUT (人体の外部) の投票箱を持つものとし，最初全てのボクセルは初期値として IN が投票されている．次に，Fig.7 に示すように，ステレオカメラの距離画像上の各画素をボクセル空間に逆投影し，その画素に対応する距離データ点を含むボクセル A とカメラの投影中心との間に存在するボクセルに対して OUT を投票する．このような投票を全ステレオカメラの全ての画素に対して行い，全てのボクセル空間に共通して IN が投票されているボクセルの集合  $S$  を求める．

次に，求めたボクセル集合  $S$  を用い，FLSM による人体概形の追跡を行う．まず，ボクセル集合  $S$  の中で OUT が投票されているボクセルに隣接するボクセルを

停止領域 (stopping region)，その他のボクセルを内部領域 (inner region) と定義する．全ボクセル空間を内包する初期曲面を考え，FLSM を用いて曲面をある速度で収縮させ，停止領域内に zero level set が到達した場合には，成長速度を小さくすることにより人体概形を復元する．Fig.8(a) にボクセル空間内に複数人体が存在する様子の水平断面図を，Fig.8(b) にその時の停止領域と内部領域の様子を示す．このように両方のカメラから複数人体の全ての表面が観測できれば，FLSM を適用して正確な人体概形の復元が可能である．しかし，Fig.8(c) のように人体が移動すると相互オクルージョンが発生し，また腕や足などによってもセルフオクルージョンが生じることがある．これらの場合，停止領域が Fig.8(d) のようになるため，2 つが結合した概形が復元される．そこで，オクルージョン領域が発生する場合でもそれぞれの形状を分離して復元するために，オクルージョンの領域内では zero level set を進行させない，つまり Fig.9 に示すように，zero level set 点 A がオクルージョンの領域内に存在する場合は，点 A の成長速度を小さくすることにする．具体的には，各ステレオカメラから獲得した距離データ点を含むボクセルと点 A との間の距離をそれぞれ  $d_1, d_2$ ，またその和を  $D$  とし， $d_1$  と  $d_2$  の最小値と  $D$  との比  $r_d$  を

$$r_d = \frac{\min(d_1, d_2)}{D} \quad (15)$$

と定義する．次に，以下の手順に従って式 (13) で用いる係数  $k_I$  を決定する．

1. zero level set 点が停止領域内の場合， $k_I = 0$  とする．
2. zero level set 点が内部領域内に存在し，かつ  $D > \alpha$  であり，

$$\begin{cases} r_d > \beta \text{ の場合, } k_I = 0 \text{ とする} \\ r_d \leq \beta \text{ の場合, } k_I = C_1 \text{ とする} \end{cases} \quad (16)$$

3. zero level set 点が内部領域内に存在し，かつ  $D \leq \alpha$  である場合， $k_I = C_1$  とする．
4. zero level set 点が内部領域・停止領域のどちらにも存在しない場合， $k_I = -C_2$  とする．

ただし， $C_1 < C_2$  であり， $\alpha = N/2$ ， $\beta = N/8$  とした ( $N$  はボクセル空間での各座標軸方向のボクセル数)．これにより，距離データ点から離れた内部領域に含まれるボクセルでは小さな成長速度が設定され，2 つの分離した領域はオクルージョンが発生しても併合しにくくなる．

本システムの特徴として，次の 3 つが挙げられる．まず，本手法では距離データを時空間的に統合するため，距離画像内のノイズに対して頑強である．次に，本手法はシルエット法に比べて背景と対象の分離が容易であり，背景差分や特殊なスタジオは必要ない．また，FLSM により閉領域境界を追跡するため，一時的にデータ欠損が生じても滑らかな境界の保存が可能である．すなわち，一時的にオクルージョン領域が発生しても，その領域では以前の境界を保つことで物体形状の概形を得ることができ，オクルージョンに強い．

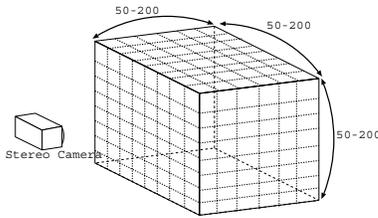


Fig.6 Voxel space.

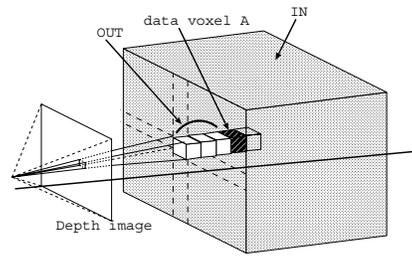


Fig.7 Voxel voting.

## 5.2 実験

提案した FLSM を適用して人体の追跡実験を行った。使用した距離画像のサイズは  $640 \times 480$  pixel, 計算機は Pentium IV, 2.8GHz である。Fig.10 に示すように, 実験では  $1.2m$  立方のボクセル空間中心から 2 台のステレオカメラをそれぞれ  $2.3m$  離して設置した。実験で使用したボクセル空間の解像度は  $50 \times 50 \times 50$  (ボクセルの一辺は  $2.2cm$  であり, 以下, これを低解像度と呼ぶ) と,  $200 \times 200 \times 200$  ( $0.6cm$ , 高解像度) であり, また, 実験は低解像度での人間動作の追跡 (実験 1), 高解像度での人間動作を追跡 (実験 2), 低解像度での複数対象物の追跡 (実験 3) の 3 通りを行った。実験 1 と実験 2 の全体の処理速度はそれぞれ  $15.4Hz$ ,  $0.84Hz$  である。

Fig.11 に実験 1 と実験 2 の結果を示す。ただし, Fig.11(a) は実際の実験の様子, Fig.11(b) は実験 1, Fig.11(c) は実験 2 の結果を示したものである。実験 1 では人間動作をほぼ遅れなく追跡することができたが, 実験 2 では実験 1 に比べて詳細に人体形状を復元できているものの, ボクセルサイズが 64 分の 1 になったため, 処理速度は 8 分の 1 に低下した。また, Fig.11(d) には, 高解像度の場合に得られた人体形状にマーチングキューブ法を適用し, その後テクスチャマッピングを行った結果を示す。

次に, 複数対象物の追跡を行った結果を Fig.13 に示す。ただし, Fig.13(a) は実際の実験の様子, Fig.13(b) はボクセル空間を上方から示したもので, 複数対象物は Fig.12(a) から Fig.12(b) に示すように移動する。また, Fig.13(c) はボクセル空間を斜め上方から示したもので, 複数対象物は Fig.12(c) から Fig.12(d) に示すように移動する。Fig.13 より, 複数対象物が交差し, 相互オクルージョン領域を含む場合でも, 人体概形を保存しつつ追跡できていることがわかる。これらの実験より, シーンに人体が複数存在し, お互いが交差するような場合でもそれぞれを分離した 3 次元形状の復元と追跡が可能であることを確認した。しかし, 本システムは単純な閾値処理により複数対象物をそれぞれ分離しているため, オクルージョン時間や物体間の距離によって複数対象物を分離することが困難になる場合がある。そこで, 今後はこの問題の解決策として, ラベリングした対象物の概形や体積などの情報を保存し, それらの情報を用いて複数対象物の分離を行う手法を検討する予定である。

## 6. まとめ

本報告では高速で安定な Level Set Method の解法として, 拡張成長速度場の利用と高速な拡張成長速度場の構築を特徴とする Fast Level Set Method の概要を示した。また, その応用例として, ビデオ画像上の移動

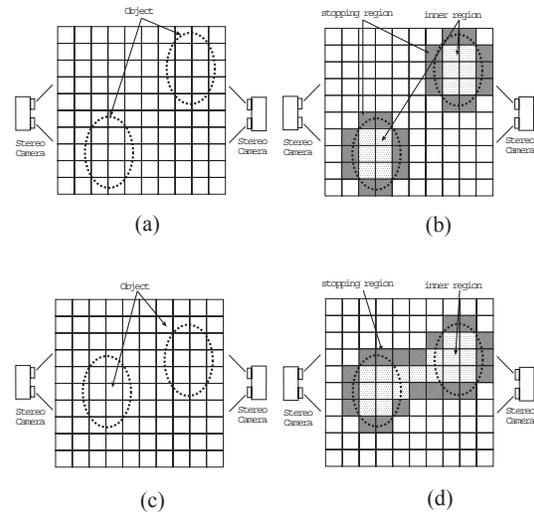


Fig.8 Stopping region and inner region.

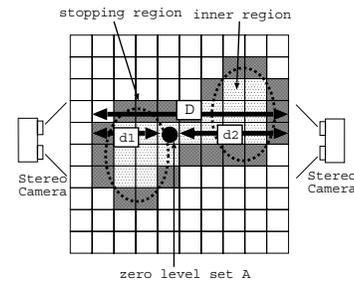


Fig.9 Zero level set in occlusion region.

物体の実時間追跡, およびステレオカメラを用いた人体の 3 次元実時間追跡を行うシステムを開発した。今後は, 本システムを PC クラスタに実装し, より詳細な対象物体の 3 次元形状をより高速に復元するシステムを開発する。

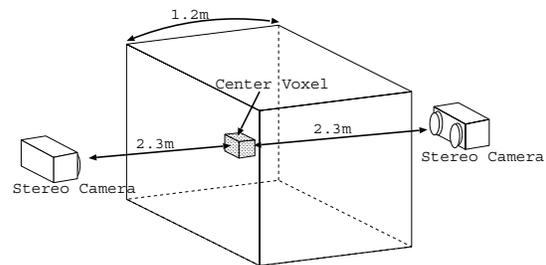


Fig.10 The camera settings.

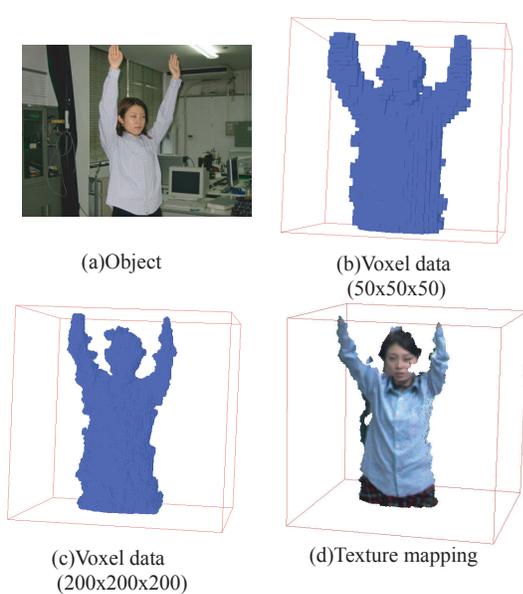


Fig.11 3D reconstruction of human body.

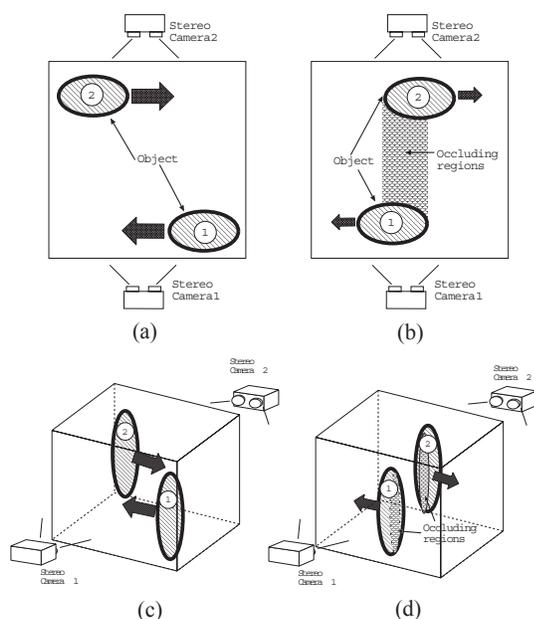


Fig.12 Top and cross views of experiment.

### 謝辞

本研究の一部は、21世紀COEプログラム「システム情報科学での社会基盤システム形成」および総務省戦略的情報通信研究開発推進制度の支援を受けた。

### 参考文献

- 1) M. Kass, A. Witkin and D. Terzopoulos, "Snakes, Active contour models", *Int. J. Computer Vision*, Vol.1, No.4, pp.321-331, 1988
- 2) D. Terzopoulos, A. Witkin and M. Kass, "Constraints on deformable models: Recovering 3D shape and nonrigid motion", *Artif. Intell.*, *Artif. Intell.*, vol.36, pp.91-123, 1988
- 3) S. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithm based on Hamilton-Jacobi formation", *J. Computational Physics*, Vol.79, pp.12-49, 1988

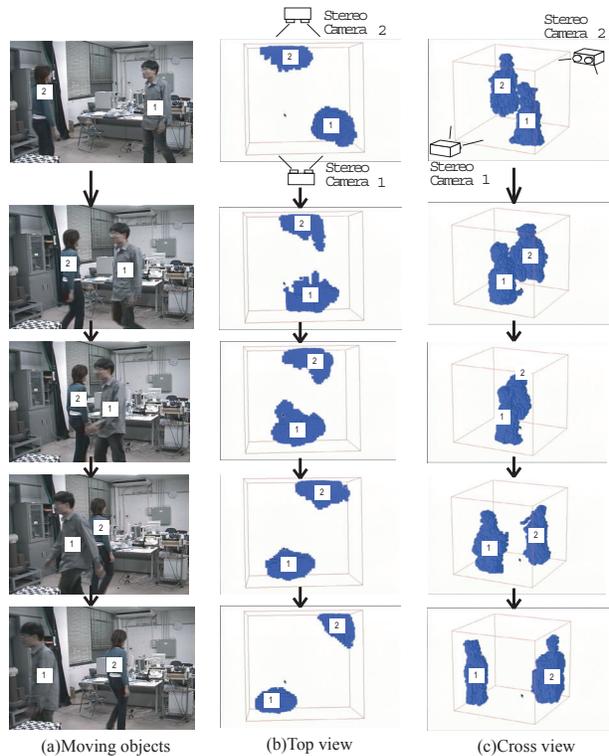


Fig.13 3D reconstruction of multiple human body.

- 4) N. Paragios and R. Deriche, "Geodesic active contours and level sets for detection and tracking moving objects", *IEEE Trans. on PAMI*, Vol.22, pp.266-280, 2000
- 5) S. Yui, K. Hara, H. Zha and T. Hasegawa, "A fast narrow band method and its application in topology-adaptive 3-D modeling", *Proc. ICPR02*, vol.IV, pp.122-125, Aug, 2002
- 6) D. L. Chopp, "Computing minimal surfaces via level set curving flow", *J. Computational Physics*, Vol.106, pp.77-91, 1993
- 7) J. A. Sethian, "A fast marching level set method for monotonically advancing fronts", *Proc. Nat'l Acad. Sci. USA*, Vol.93, pp.1591-1595, 1996
- 8) J. A. Sethian, "Level Set Method and Fast Marching Methods second ed.", *Cambridge University Press*, UK(1999)
- 9) W. N. Martin and J. K. Aggarwal, "Volumetric description of objects from multiple views", *IEEE Trans. on PAMI*, Vol.5, No.2, pp.150-158, 1983
- 10) A. Laurentini, "The visual hull concept for silhouette-based image understanding", *IEEE Trans. on PAMI*, Vol.16, No.2, pp.155-162, 1994
- 11) ウ 小軍, 和田 俊和, 東海 彰吾, 松山 隆司, "平面間透視投影を用いた並列視体積交差法", *情報処理学会論文誌: コンピュータビジョンとイメージメディア*, Vol.42, No.SIG6(CVIM2), pp.33-43, 2001
- 12) 倉爪 亮, 由井 俊太郎, 辻 徳生, 岩下 友美, 原 健二, 長谷川 勉, "Fast Level Set Method の提案とビデオ画像の移動体のリアルタイム追跡", *情報処理学会論文誌*, Vol.44, No.8, pp.2244-2254, 2003