Fast Level Set Method の PC クラスタへの実装

Implementation of Fast Level Set Method on a PC-cluster

岩下 友美（九州大） ◯ 山崎 智弘（九州大） 倉爪 亮（九州大） 長谷川 勉（九州大）

Yumi IWASHITA †Kyushu University, Hakozaki 6-10-1, Fukuoka

Tomohiro YAMASAKI † Ryo KURAZUME† Tsutomu HASEGAWA†

The level set method(LSM) has been widely used for various applications such as motion tracking and 3D geometrical modeling. However, the calculation cost of reinitialization and updating of the implicit function is considerably expensive. To tackle this problem, we have proposed an efficient algorithm of the LSM named the Fast Level Set Method(FLSM). This paper introduces an improvement for FLSM in an adaptive fashion using multiresolutional space representation as well as an implementation of the FLSM on a PC-cluster.

Key Words: Level set method, 3D shape reconstruction, range image

1 はじめに

Snakes[1)] や Deformable surface[2)] に代表される動的輪郭モデル (Active contour model) は，対象物体を内包する閉曲面を安定に抽出する手法として 2 次元，3 次元空間での移動体追跡や幾何モデリングの分野において広く利用されている．しかし，従来の動的輪郭モデルに共通して，閉曲面の分離や結合などの位相変化への対応が困難であることが問題とされていた．

これに対し，位相変化が可能な動的輪郭モデルとして，Level Set Method(LSM)[3)4)] が提案され，移動体追跡や 3 次元幾何モデリングなどの分野で応用例が報告されている．しかし、その実現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った。この問題に対し、我々はこれまでに LSM の高速化手法として，Fast Level Set Method(FLSM) を提案し[7)]，物体の 3 次元形状を高速に復元できることを示した．

本報告では，これまでに提案した FLSM を概説し，PC クラスタへの実装と，解像度を適切に切り替えることで，対象物体の 3 次元形状を高速に復元する手法について述べる．

2 Level Set Method とその高速化手法

2.1 Level Set Method

Level Set Method は，Osher,Sethian ら[3)4)] 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，2 次元 xy 平面内での 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 $p=(p_x,p_y)$ 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る．ここで κ はその点での境界の曲率であり，F を成長速度という．

次に，時刻 t における補助関数 $z=\Psi(x,y,t)$ を導入し，境界位置 $C(p,t)$ はその関数の一部，すなわち $z=\Psi(x,y,t)=0$ を満たす Ψ 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 $\Psi(x,y,t)$ のゼロ等高面である条件は，$\Psi(p(t),t)=0$ で表される．これを時間で偏微分すると，

$$\Psi_t + \nabla\Psi(p(t),t)p_t = 0 \tag{1}$$

となり，さらに式 (1) は以下のように変換される．

$$\Psi_t = -F(\kappa) \mid \nabla\Psi \mid \tag{2}$$

$$\Psi(C_0(p),0) = 0 \tag{3}$$

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，$\Psi(x,y,t)=0$ 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．実際に平面上の点 i,j において補助関数 Ψ_{ij} 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．

$$\psi^{n+1}_{i,j} = \psi^n_{i,j} - F(i,j)|\nabla\psi^n_{i,j}|\Delta t \tag{4}$$

ただし，Δt は積分間隔である．

2 次元濃淡画像における領域追跡の具体的な実現法を考える．例えば平面上の点 (x,y) における時刻 t での $\Psi(x,y,t)$ の値を，その時刻における境界 $\Psi(x,y,t)=0$ 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

$$F(\kappa) = k_I(a-b\kappa) \tag{5}$$

で与えることにする．ただし k_I は濃度勾配に関する項，κ はその点での境界の曲率，$a,b \geq 0$ は定数とする．

さて，拡張成長速度場[3)7)] を用いた Level Set Method では，各ピクセルの成長速度を決定するために，まず zero level set（補助関数値が 0 のピクセル）での成長速度を決定し，その他のピクセルでは最も近い zero level set のピクセルの成長速度をコピーして成長速度場を構築する．

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ピ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，上記の成長速度場の構築処理や再初期化処理において，コピー元の zero level set を決定したり各ピクセルで現在の zero level set からの距離を得るには，各ピ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，これまでに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[6)]．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，高速に成長速度場を構築する Fast Narrow Band Method(FNB)[5] 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[7]．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 2 次元空間での実装法について説明する．まず，図 1(a) のように表される参照マップをあらかじめ作成する．これは，原点周辺にあるピ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 であるピクセルの集合を R_r とし，$r = 0 \sim \delta(\delta+1)$ に対するリスト $R_0, R_1, \cdots, R_{\delta(\delta+1)}$ をそれぞれ作成する．なお，ここで距離にはピ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 δ 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 δ 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 $\delta(\delta+1)$ 以下のピクセルの集合ともみなせる．一例として図 1(a) に，バンド幅 $\delta = 3$ における参照マップ（$R_0 \sim R_{12}$）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．

次に，作成した参照マップ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（5）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 $\delta(\delta+1)$ であるようなピクセルを選択し，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ピ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，各ピ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）[5] と呼ぶことにする．

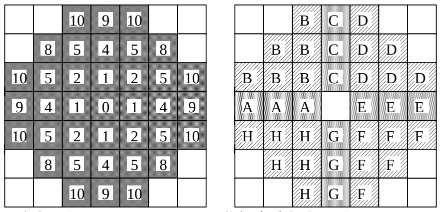
3.2 参照マップの分割と再初期化処理との統合 (Fast Level Set Method)

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，図 1(b) のように参照マップを原点からの方向により，A〜F の 8 つの領域に分ける．ただし A,C,E,G を 4 近傍領域と呼び、B,D,F,H を 8 近傍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. 上下左右の 4 近傍を調べ、そこに他の境界上の点があるときは、その方向の 4 近傍領域を書き込まない領域とする。また同時に、その 4 近傍に隣接する 8 近傍領域も書き込まない領域とする。
2. 斜め方向の近傍を調べ、そこに他の境界点があればその 8 近傍方向を書き込まない領域とする。
3. （1），（2）で残った領域に対し、前節と同様にリストを用いて成長速度を書き込み、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．



(a)Reference map (b)Divided reference map

Fig.1: 参照マップ.

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ー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 FLSM の高速な実現手法

4.1 解像度の切り替えによる収束の高速化

物体形状をより正確に復元するためには，空間のグリッド幅を小さくし，高解像度で曲面の更新する必要がある．しかし，高解像度になればなるほど，低解像度の場合に比べ，計算コストが増加する．そこで，モデリングの初期段階では，低解像度の状態で曲面を更新し，得られた曲面を初期曲面とし徐々に解像度を高めながら曲面を更新する．このように解像度を徐々に高めながら曲面の更新を行うことで，より正確な物体形状を高速に復元することが可能となる．以上を解像度制御型 FLSM と呼ぶことにする．解像度制御型 FLSM の流れを Fig.2 に示す．

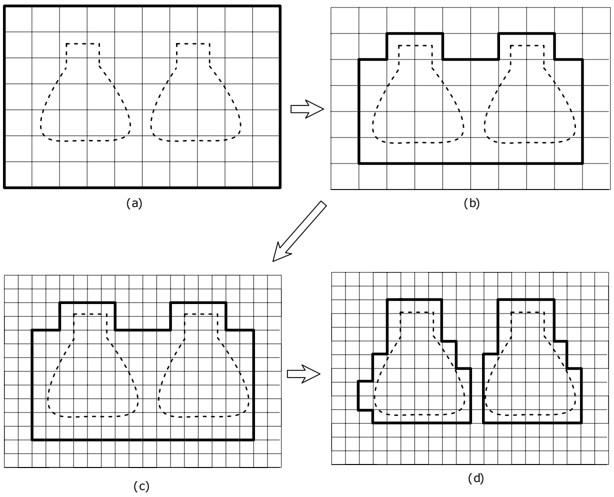


Fig.2: 解像度制御型 FLSM の処理の流れ

5　FLSM の PC クラスタへの実装

FLSM の処理は処理空間を分割することで並列化できる．そこで我々は，今回 PC クラスタを用いた並列処理により FLSM の計算の高速化及び高精度化を実現した．FLSM を実装した PC クラスタシステムは，4 台の計算機がギガビットイーサで相互に接続されたもので，CPU は Intel Xeon-3.06GHz ×2，Memory 2GB である．

まず，3 次元空間全体が N 個のボクセルに分割されているとする．処理を行う PC の台数 M でこのボクセル空間を分割し，それぞれの PC で分割された N/M のボクセル空間に対し FLSM 処理を適用する．Fig.3 は PC を 4 台用いた時の空間の分割の様子である．空間の分割を行う際，それぞれの空間で 3%程度の重複する領域を作ることで境界の繋ぎ目が現れないようにしている．

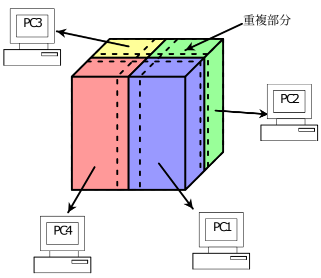


Fig.3: ボクセル空間の分割

6　実験

本手法の有効性を確認するため，Fig.4 に示すステゴサウルスの模型の距離画像データをレーザレンジファインダ (Minolta, VIVID700) を用いて計測し，FLSM を用いたモデリング実験を行った．計測により得られたデータ点数は 619,054 点である．

Fig.5 に解像度制御型 FLSM を用いてモデルを生成した結果を示す．また，解像度制御を用いた FLSM の処理結果と解像度制御を用いない従来の FLSM の処理結果との計算時間の比較を Table 1 に示す．ボクセル空間の最終解像度は 256 × 256 × 256 である．

これより，解像度制御型 FLSM は，従来手法と比べ計算時間が大幅に削減されており，解像度制御型 FLSM の有効性が確認できた．

Fig.4: ステゴサウルスの模型

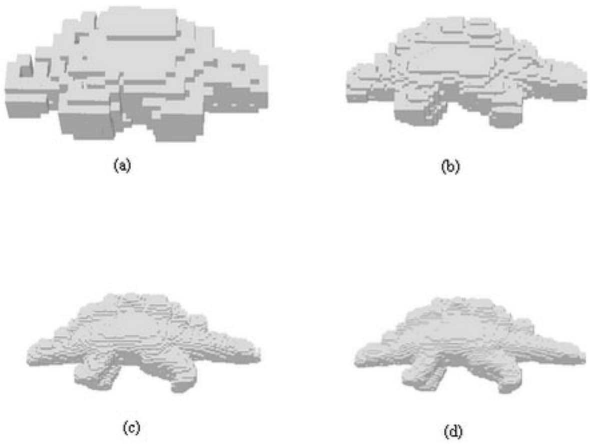


Fig.5: 解像度制御型 FLSM を用いたモデル復元過程

Table 1: 解像度制御型 FLSM と従来の FLSM との計算時間の比較

ボクセル空間の解像度	従来の FLSM[sec.]	解像度制御型 FLSM[sec.]
128 × 128 × 128	9.53	1.59
256 × 256 × 256	83.61	4.37

次に，PC クラスタへ実装した FLSM を用いてモデル生成した結果を示す．使用した計算機は 4 台，ボクセル空間の解像度は 700 × 700 × 700 である．使用した 4 台のうち 1 台を FLSM の処理と同時に，曲面の更新される様子を表示するために用いた．物体形状が復元される過程を Fig.6(a)-(d) に示す．Fig.7(a)(b) は Fig.6(d) に Marching Cubes 法を適用したものである．得られたモデルの頂点数は 11,754,912 点，パッチ数は 3,918,304 個であった．実験の結果，1 台の計算機による FLSM の処理では，ボクセル空間の解像度は 517 × 517 × 517 が最高であったのに対し，PC クラスタを用いることにより，より高い解像度で対象物体の正確な 3 次元形状を復元できることが確認できた．

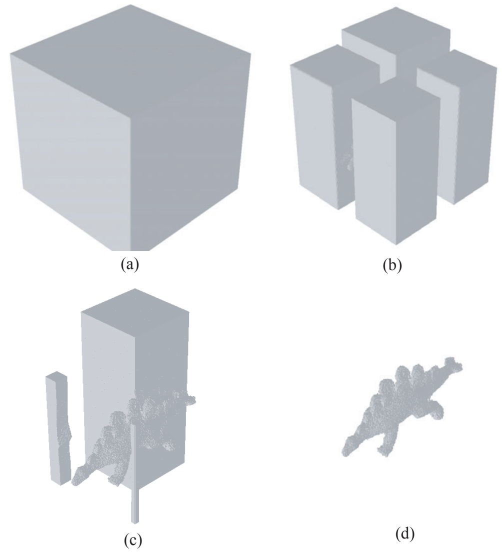
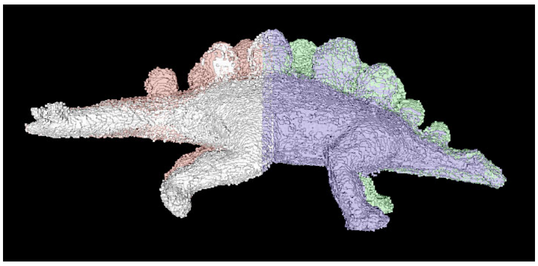
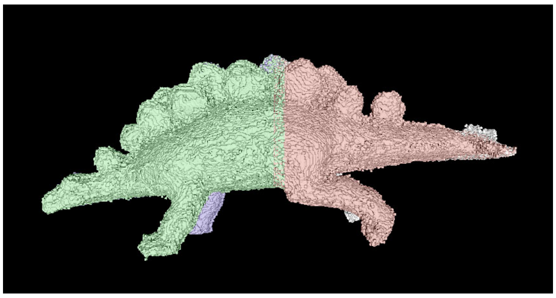


Fig.6: PC クラスタを用いたモデル復元過程



(a)



(b)

Fig.7: 復元されたモデル (700 x 700 x 700)

7 まとめ

本報告では，これまでに提案した FLSM を概説し，PC クラスタへの実装と，解像度を適切に切り替えることで，対象物体の 3 次元形状を高速に復元でき手法を提案した．また，実際の距離画像を用いたモデリング実験により，提案した手法の有効性を確認した．

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

4) J. Sethian: Level Set Methods, 1st ed. New York, Cambridge University Press, 1996

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 curvature flow,” *J. Conputational Physics*,Vol.106, pp77-91,1993

7) 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物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)