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

倉爪 亮[†] 由井 俊太郎[†] 辻 徳生[†]
岩下 友美[†] 原 健二[††] 長谷川 勉[†]

Osher，Sethian らによって提案された Level Set Method は，位相変化が可能な動的輪郭モデルとして注目を集め，現在，移動体追跡や 3 次元幾何形状モデリング，半導体や結晶形成シミュレーションなど，様々な用途に応用され始めている．しかしこの手法は，Snakes など従来の動的輪郭モデルと比較して初期化や更新時の計算コストが高く，高速な計算手法の開発が大きな課題となっている．そこで本論文では高速で安定な Level Set Method の解法として，i) 拡張成長速度の利用と高速な拡張成長速度場の構築（Fast Narrow Band Method，FNB），ii) 補助関数の再初期化処理の高速化と頻繁な再初期化，を特徴とする新たな Fast Level Set Method を提案する．また従来の Level Set Method との比較実験を行い，本手法により境界が安定かつ高速に検出できることを示す．さらに提案する手法の応用例として，ビデオ画像上の移動物体のリアルタイム追跡へ適用した結果を紹介する．

Fast Level Set Method and Realtime Tracking of Moving Objects in a Sequence of Images

Ryo Kurazume,[†] Shuntaro Yui,[†] Tokuo Tsuji,[†]
Yumi Iwashita,[†] Kenji Hara[††] and Tsutomu Hasegawa[†]

The level set method, introduced by S. Osher and J. A. Sethian, has attracted much attention as a topological-free active contour model. This method utilizes an implicit representation of a contour to be tracked, and is able to handle the topological change of the contour naturally. Various applications based on the level set method have been presented including motion tracking, 3D geometrical modelling, and simulation of crystallization or semiconductor growth. 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”. In this paper, we propose an efficient calculation algorithm for the level set method named the Fast Level Set Method (FLSM). Advantages of the proposed FLSM are as follows: i) the use of the extension velocity and the high speed construction of the extension velocity field using the Fast Narrow Band Method, ii) the frequent execution of the reinitialization process of the implicit function which requires little calculation cost. The efficiency of the proposed method is verified through computer simulations, and experiments of realtime tracking of moving objects in video images.

1. はじめに

Kass らの Snakes[7] に代表される動的輪郭モデル（Active Contour Model）はノイズに対して頑強な境界追跡法である．当初，2 次元画像の境界追跡手法として提案された Snakes は，その後 3 次元にも拡張され，3 次元点の集合（points of cloud）からそれを内包する閉曲面を安定に抽出するための deformable surface の手法[5),6),11)] 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た．しかし従来の動的輪郭モデルに共通して，分離や結合など境界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，Osher，Sethian らによって提案された Level Set Method[13),19)] は，本質的に位相変化が可能な動的輪郭モデルとして注目を集め，現在，移動体追跡[16)]，や 3 次元幾何形状モデリング[3),12),25)]，半導体[21)] や結晶形成シミュレーション[26)] など，様々な用途に応用され始めている．この手法は，検出する境界を一次元高い補助関数のゼロ等高面と見なし，境界

† 九州大学大学院システム情報科学研究院
Graduate School of Information Science and Electrical Engineering, Kyushu University
†† 福岡県工業技術センタ
Fukuoka Industrial Technology Center

の進行条件である偏微分方程式（PDE，Partial Differencial Equation）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（zero level set）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，一般に空間を均一グリッド点で離散化し，境界を含むグリッド点とその周囲のグリッド点に設定された値（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，たとえば画像濃淡値などグリッド点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幅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グリッド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．この各グリッドへの初期値の設定，およ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となっている．

この問題に対してこれまでに，Narrow Band Method[1),20)] や Fast Marching Method[2),18),20)]，SFA（Space Field Algorithm）[23)]，Hermes[14),16)]，AOS（Additive Operator Splitting）[9),22)]，あるいは初期境界の最適設定[25)] や多重解像度制御[17)] などの手法が提案されている．

一方，各更新ごとの高速化に加えて，空間内の未知の境界を短時間で検出するには，補助関数を更新する積分時間幅をできるだけ大きくとることが望ましい．しかし PDE に対して数値的に安定な解を得るには CFL（Courant-Friedrich-Levy）条件[8)] を満たす必要があり，これは各グリッド点での値の更新に対する積分時間幅の制約となる．

そこで本論文では高速で安定な Level Set Method の解法として，i) 拡張成長速度（Extention Velocity）の利用と高速な拡張成長速度場（Extention Velocity Field）の構築（Fast Narrow Band Method, FNB[24)]），ii) 補助関数の再初期化処理の高速化と頻繁な再初期化，を特徴とする新たな Fast Level Set Method を提案する．本手法は CFL 条件を満たさない大きな積分時間幅に対しても，実用上安定に境界を抽出できる可能性がある．さらに従来の Level Set Method の手法と提案手法の比較実験，およびビデオ画像上の移動物体のリアルタイム追跡実験を行った結果を紹介する．

本論文の構成は以下のようになる．まず 2 章では Osher，Sethian らによって提案された Level Set Method と，その関連手法を紹介する．次に 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 を提案し，4 章で計算機実験とビデオ画像上の移動物体のリアルタイム検出，追跡実験の様子を紹介する．5 章は本論文のまとめである．

2. Level Set Method

2.1 Upwind Scheme を用いた Level Set Method と Fast Marching Method

Level Set Method は，Osher，Sethian ら[13),19)] 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．この手法は，検出する境界が動的か，あるいは静的かによって，Upwind Scheme を用いた Level Set Method と Fast Marching Method の 2 つの手法に分類できる．

2.1.1 Upwind Scheme を用いた Level Set Method

例として，2 次元 xy 平面での境界領域の決定問題を考える．時刻 t での境界位置を $C(\mathbf{p},t)$ とする．ただし $\mathbf{p}=(p_x,p_y)$ である．この境界に含まれる点 $\mathbf{p}$ 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．これを式で表すと，

$$C_t = F(\kappa)N \tag{1}$$

$$C(p,0) = C_0(p) \tag{2}$$

となる．この問題は差分方程式を利用したラグランジェ法で解くことができるが，トポロジーの変化には対応できないという問題がある．

そこで新たな補助関数 $z=\Psi(x,y,t)$ を導入し，境界位置 $C(\mathbf{p},t)$ 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 $\mathbf{p}(t)$ が境界 $C(\mathbf{p},t)$ 上の点であると仮定すると，これがつねに $\Psi(x,y,t)$ のゼロ等高面上である条件は，

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

で表される．これを偏微分すると，

$$\Psi_t + \nabla\Psi(\mathbf{p}(t),t)\mathbf{p}_t = 0 \tag{4}$$

となる．また曲線上の単位法線ベクトルは，

$$N = \frac{\nabla\Psi}{|\nabla\Psi|} \tag{5}$$

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

$$\mathbf{p}_t \cdot N = F \tag{6}$$

となる．これにより式 (4) は以下のように変換される．

$$\Psi_t = -F(\kappa) \mid \nabla\Psi \mid \tag{7}$$

$$\Psi(C_0(p), 0) = 0 \tag{8}$$

このように境界 $C(\mathbf{p}, t)$ 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．実際に空間上の点 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 を解く必要がある．

$$\Psi_{ij}^{n+1} = \Psi_{ij}^{n} - \Delta t(\max(F_{ij}, 0)\nabla^{+} + \min(F_{ij}, 0)\nabla^{-}) \tag{9}$$

$$\nabla^{+} = (\max(D_{ij}^{-x}, -D_{ij}^{+x}, 0)^2 + \max(D_{ij}^{-y}, -D_{ij}^{+y}, 0)^2)^{\frac{1}{2}} \tag{10}$$

$$\nabla^{-} = (\max(D_{ij}^{+x}, -D_{ij}^{-x}, 0)^2 + \max(D_{ij}^{+y}, -D_{ij}^{-y}, 0)^2)^{\frac{1}{2}} \tag{11}$$

ただし，

$$D_{ij}^{+x} = \frac{\psi_{i+1,j}^{n} - \psi_{i,j}^{n}}{h} \quad D_{ij}^{+y} = \frac{\psi_{i,j+1}^{n} - \psi_{i,j}^{n}}{h}$$
$$D_{ij}^{-x} = \frac{\psi_{i,j}^{n} - \psi_{i-1,j}^{n}}{h} \quad D_{ij}^{-y} = \frac{\psi_{i,j}^{n} - \psi_{i,j-1}^{n}}{h} \tag{12}$$

であり，n は積分回数，h は離散化幅，Δt は積分間隔である．

濃淡画像における領域追跡の具体的な実現法を考える．たとえば空間上の点 (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

$$F(\kappa) = k_I(a - b\kappa) \tag{13}$$

で与えることにする．ただし k_I は輝度勾配に関する項，$a, b \geq 0$ は定数であり，たとえば k_I として，

$$k_I = \frac{1}{1 + \nabla G \otimes I(x, y)} \tag{14}$$

などが考えられる．ただし，$\nabla G\otimes$ は LoG（Laplacian Of Gaussian）フィルタ，$I(x, y)$ は輝度値である．これにより，補助関数は輝度勾配が小さいと速度が大きくなり，境界はその内側方向へ移動する．一方，輝度勾配が十分に大きい場所では速度が 0 に近くなり，境界はその場所で停止する．また曲率が $\kappa \geq \frac{a}{b}$ の場合には速度が負になり，境界は内側方向へ移動する．

なお，式 (13) は κ を 3 次元曲率にすることで，3 次元空間にも拡張できる．また式 (13) 以外にも，指数関数を用いた成長速度の決定法[15] なども提案されている．

2.1.2 Fast Marching Method

Fast Marching Method は，Eikonal 方程式（$|\nabla T(p)|\ F = 1$）の高速な数値解法として提案された．ただし $T(p)$ は境界が点 p に到達した時刻である．この方程式は通常，収束計算により解かれるため，膨大な計算時間が必要である．しかし，Fast Marching Method では，成長速度の符号が固定という条件を加え，到達時刻の小さい点から大きい点へ 1 方向に到達時刻を確定していくことで，収束計算を行うことなく高速に Eikonal 方程式の解を導くことができる．

Fast Marching Method では，まず Eikonal 方程式を次の差分方程式に置き換える．

$$(\max(D_{ij}^{-x}T, -D_{ij}^{+x}T, 0)^2 + \max(D_{ij}^{-y}T, -D_{ij}^{+y}T, 0)^2)^{\frac{1}{2}} = 1/F_{ij} \tag{15}$$

次に，境界は到達時間が小さい場所から大きい場所へと 1 方向に伝播することから，到達時間の小さい場所から順に式 (15) を解き，各点の到達時間を確定する．

具体的には以下のようになる．

Step.1（初期化） まず，次の処理によってすべてのグリッドを 3 つのリストのいずれかに追加する．

（1） 境界に属するグリッドを *known* のリストに追加し，到達時間を 0 にセットする（T = 0）．

（2） *known* の 4 近傍のグリッドのうち，*known* でないグリッドを *trial* に追加し，そのグリッドの到達時間を $T_{ij} = 1/F_{ij}$ により計算して仮登録する．また，これらのグリッドを到達時間に関する昇順の Heap 形式で保存する．

（3） 上記以外のピクセルを *far* に追加し，到達時間を無限大とする（T = ∞）．

Step.2 Heap の先頭に置かれた，*trial* のリスト中で到達時間 T が最も小さいグリッド (i_{min}, j_{min}) を選択し，そのグリッドを *trial* のリストと Heap から除外し，*known* のリストに追加する．Heap から除外する際，同時に Heap を再構築（downheap）する．

Step.3 現在選択されたグリッド (i_{min}, j_{min}) の 4 近傍（$(i_{min} - 1, j_{min})$，$(i_{min} + 1, j_{min})$，$(i_{min}, j_{min} - 1)$，$(i_{min}, j_{min} + 1)$）のうち，*far* のリストに属しているグリッドを *trial* のリストに追加する．

Step.4 現在選択されたグリッド (i_{min}, j_{min}) の 4 近傍のうち，*trial* に属している近傍点の到達時間を式 (15) を用いて計算して仮登録し，Heap を再構築（upheap）する．

Step.5 *trial* に属しているグリッドが存在すれば Step.2 へ戻る．それ以外のときは処理を終了する．

2.2 Narrow Band

境界領域の追跡では，空間全体に対して補助関数を計算する必要はなく，ゼロ等高面に近い領域だけを取り扱えばよい．この手法は Narrow Band Method [1),4)] と呼ばれている．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 領域の境界に近づいたときのみ，Narrow Band 領域内で後述の再初期化処理を行う．

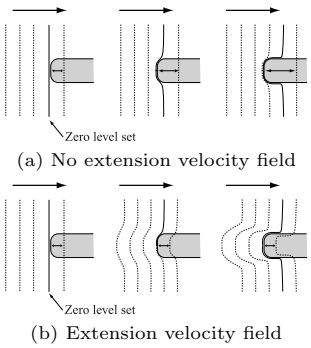
2.3 再初期化

PDE を数値的に解き，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グリッド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[19)] が必要である．しかし再初期化処理で各グリッドにおいて現在の zero level set からの距離を求めるには，各グリッド点の最近傍 zero level set を探索する必要があり，計算コストが高い．そこでこの再初期化を高速に行うために，Adalsteinsson らは補助関数の計算が静的な境界追跡手法と等しいことを利用し，Fast Marching Method を用いた高速な補助関数の計算手法を提案した[2)]．しかし依然として再初期化の計算コストは，補助場更新時の計算コストに比べて大きく，このため安定に PDE を解き，境界を検出するためには，CFL 条件[8)] を満たした小さな積分時間幅で計算を進め，再初期化処理の回数を少なくする必要があった．

2.4 局所成長速度場と拡張成長速度場

Level Set Method において各グリッド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，各グリッドにおける補助関数の成長速度を決定するのに，そのグリッド自身の状態，たと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 のグリッド）での成長速度を決定し，その他のグリッドでは最も近い zero level set のグリッドの成長速度をコピーして用いるものである．

両者の境界進行の様子を**図 1** に示す．局所成長速度（図 1 (a)）は，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，内側では粗になり，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（図 1 (b)）は，zero level set が境界に到達すると，その情報が周囲に伝わって周囲の進行速度も減少させるため，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 [20)] も多くの計算機実験から，拡張成長速度場のほうが解が安定に求まり，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．



(a) No extension velocity field

(b) Extension velocity field

図 1 局所成長速度と拡張成長速度での境界進行の比較
Fig. 1 Comparison of front propagation in no extension and extension velocity field.

3. Fast Level Set Method の提案

これまで拡張成長速度場は，その構築に最近傍 zero level set 点の探索が不可欠であり，局所成長速度場に比べて多くの計算時間が必要であることから，リアルタイムトラッキングなど高速な計算処理が必要なアプリケーションには適さないと考えられてきた[10),19)]．

これに対して本論文では，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）[24)] 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 を提案する．

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

まず，**図 2** に示すような参照マップをあらかじめ作成する．これは，原点周辺にあるグリッド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 であるグリッドの集合を 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，ここで距離にはグリッド中心間のユークリッド距離を用いることとし，また

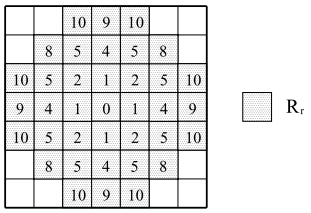


図 2 参照マップ
Fig. 2 An example of reference map.

$\delta > 0$ 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，$\delta(\delta+1) < (\delta+0.5)^2 < \delta(\delta+1)+1$ がつねに成り立つことから，zero level set からの 2 乗距離が $\delta(\delta+1)$ 以下のグリッドの集合とも見なせる．一例として図 2 に，バンド幅 $\delta = 3$ における参照マップ（$R_0 \sim R_{12}$）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 (13) など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 $\delta(\delta+1)$ であるようなグリッドを選択し，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グリッドに仮登録する．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．これによりすべての処理が終了したときには，各グリッドには最も近い 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）[24] と呼ぶことにする．

一例として，**図 3** にバンド幅 $\delta = 3$ の場合の処理の流れ（(a) → (d)）を示す．図 3 (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 であるグリッドをリスト R_{10} を用いて選択し，そのグリッドの成長速度を F_1 とした様子である．図 3 (b) では，すべての zero level set からの 2 乗距離が 10 であるグリッドに，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．図 3 (c) は，成長速度の値が F_1 である zero level set に対し，

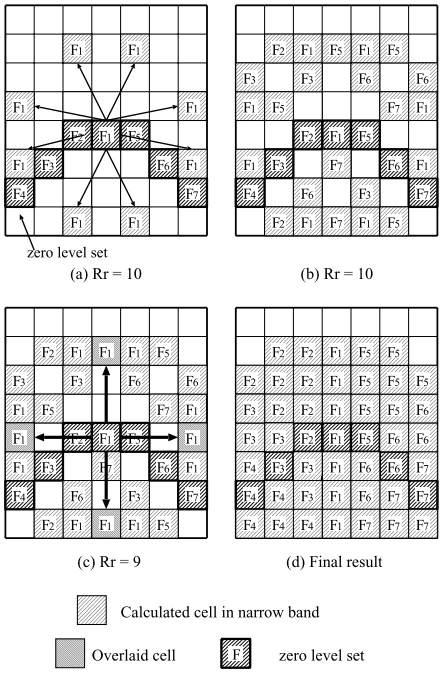


図 3 拡張成長速度場の構築過程
Fig. 3 Procedure of constructing extension velocity field.

そこからの 2 乗距離が 9 $(= 10 - 1)$ であるグリッドをリスト R_9 を用いて選択し，そのグリッドの成長速度を F_1 としている．ただし，図 3 (b) では，成長速度の値が F_1 でないグリッドに F_1 の値を上書きしている．この処理を繰り返すことで，図 3 (d) のように拡張成長速度場が構築できる．

3.2 参照マップの分割

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．たとえば，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 4** 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 zero level set 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

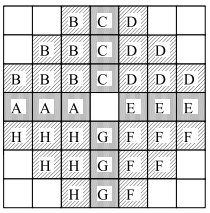


図 4 参照マップの分割
Fig. 4 Division of reference map.

域とする．たとえば，左側に隣接するグリッドが zero level set であれば，A，B，H を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 zero level set があればその 8 近傍方向を書き込まない領域とする．たとえば，左上に隣接するグリッドが zero level set であれば，B は書き込まない領域とする．

(3) (1)，(2) 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．

また，より広い近傍を調べることで，マップをさらに細かく分割し，より書き込みのオーバラップを減らすこともできる．しかし一方で調べなければならない近傍数も増加するため，最適な分割数は Narrow Band の幅 δ の値などに応じて試行錯誤的に決定する必要がある．

3.3 再初期化処理との統合

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，前節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．

3.4 CFL 条件を満たさない積分時間幅への対応

空間内の未知の境界を短時間で検出するには，補助関数を更新する積分時間幅をできるだけ大きくとることが望ましい．しかし PDE に対して数値的に安定な解を得るには CFL 条件[8] を満たす必要があり，これは Level Set Method においては各グリッド点に設定された値の更新に対する積分時間幅の制約となる．

表 1 従来の Level Set Method と提案した Fast Level Set Method の計算量
Table 1 Comparison of calculation cost between conventional Level Set Method and Fast Level Set Method.

	No Extension with Fast Marching (LSM)	Extension (FLSM)
Reinitialization	$O(\delta n \log n)$	$(O(\delta^2 n))$
Construction of velocity field	$O(\delta n)$	$O(\delta^2 n)$
Updating process	$O(\delta n)$	$O(\delta n)$
Detection of zero level set	$O(\delta n)$	$O(\delta n)$

() ··· The reinitialization process of FLSM can be integrated with the construction process of the extension velocity field.

しかし提案した Fast Level Set Method の特徴である拡張成長速度場の利用と頻繁な再初期化処理は，従来の Level Set Method における積分の不安定要因をある程度取り除くことができ，これにより CFL 条件を満たさない大きな積分時間幅を設定しても，実用上問題なく安定に境界を抽出できる可能性がある（4 章）．

3.5 局所成長速度手法との比較

上述したように，これまでの Level Set Method のアプリケーションの多くは，計算コストの問題から局所成長速度場を用いている．しかし局所成長速度場の計算は各グリッドで境界かどうかの判定や補助関数の曲率などをもとに成長速度を計算する必要があり，実際には局所成長速度場の構築にもある程度の時間が必要である．

一方，提案した拡張成長速度場の構築法では，成長速度の計算は境界グリッドだけであり，その他のグリッドは境界グリッドで計算された成長速度をルールに従って上書きするだけである．

したがって提案した手法により最近傍点探索を単純な上書き処理に置き換え，拡張成長速度場の構築に必要な時間を短縮できれば，全体の処理時間を局所成長速度場の構築に近く，あるいは逆転できる可能性がある．

そこで従来の局所成長速度場を利用する手法と本論文で提案した Fast Level Set Method の計算量の比較を**表 1** に示す．ただし，空間は 2 次元で大きさは

$n \times n$ グリッド，Narrow Band の幅は δ とする．

まず再初期化に必要な計算量は，Fast Marching 法を用いた場合が $O(\delta n \log n)$ であるのに対し，提案手法では $O(\delta^2 n)$ である．ただし前述のように，提案手法ではこの再初期化処理を拡張成長速度場の構築に統合でき，このときの全体の計算量は再初期化の有無にはほぼ無関係，すなわち再初期化処理を見かけ上，計算量 0 で実行できる．

次に成長速度場の構築に必要な計算量について考える．まず局所成長速度場を用いた場合，成長速度の計算をすべての Narrow Band 領域で行うため，局所成長速度場の構築に $O(\delta n)$ の計算量が必要である．一方，提案した Fast Level Set Method では，成長速度の計算は zero level set だけで行うため，その計算量は $O(n)$ である．しかし，拡張成長速度場の構築には，zero level set で計算された成長速度を Narrow Band 領域に代入するため，これに加えて $\delta^2 n$ 回の代入処理が必要となる．

以上より，成長速度場の構築に対しては，個々の成長速度の計算が複雑で時間がかかる場合，局所成長速度場に比べて成長速度の計算回数の少ない拡張成長速度場の利用が有利である．しかしメモリーアクセスが遅く，代入に長い処理時間が必要であったり，バンド幅 δ が大きい場合には，局所成長速度場や Fast Marching 法を用いた再初期化が有利であると思われる．

4. 実　　験

4.1 シミュレーション結果

提案した Fast Level Set Method と従来の Level Set Method に対して，境界を検出する計算機実験を行った．実験で用いた空間の大きさは 100×100 グリッド，成長速度の最大値は 1，積分時間幅は 0.1，Narrow Band の幅は 10 グリッドであり，zero level set が Narrow Band の端に到達したときに再初期化を行った．また従来の Level Set Method における再初期化処理には，高速な Fast Marching Method を用いた[2)]．

実験結果を**図 5** に，計算時間の比較を**表 2** に示す．ただし表 2 では，3.1 節，3.2 節で示した拡張成長速度場の構築法を用いた場合と，従来の Fast Marching Method を用いた Leve Set Method の構築法のうち，拡張成長速度場を用いた場合，局所成長速度場を用いた場合の，計 4 つの方法について比較した結果を示している．また，空間の大きさを $100 \times 100 \sim 300 \times 300$ まで変えた場合の計算時間の比較を**図 6** に示す．

次に成長速度の最大値を 1，積分時間幅を 2.0 とし，CFL 条件を満たさない場合に対してシミュレーションを行った．ただし提案した Fast Level Set Method では各更新ごとに Narrow Band の再初期化を行い，従来の局所成長速度場を用いた Level Set Method では zero level set が Narrow Band の端に到達したときに再初期化を行った．実験結果を**図 7** に，計算時間の比較を**表 3** に示す．

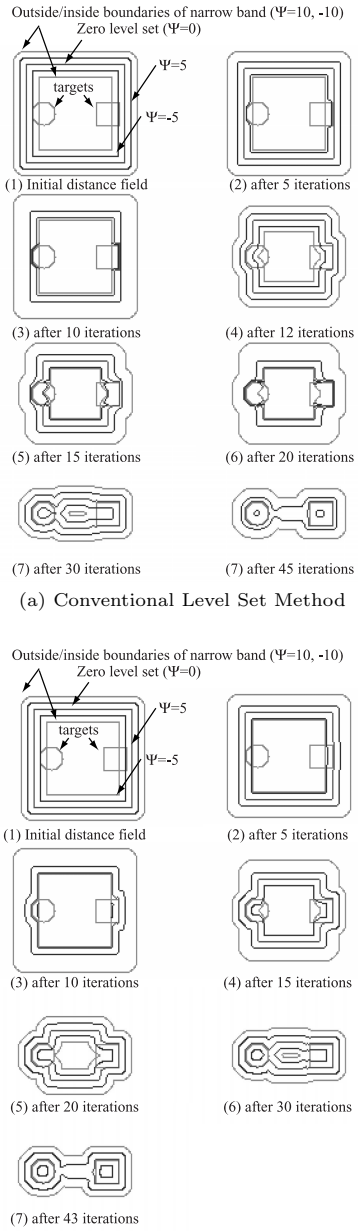


図 5 従来の Level Set Method と Fast Level Set Method のシミュレーション結果
Fig. 5 Simulation results of conventional Level Set Method and Fast Level Set Method.

表 2 Fast Level Set Method と従来の Level Set Method の計算時間の比較

Table 2 Comparison of calculation time between Fast Level Set Method (FLSM) and conventional Level Set Method (LSM). For initialization in LSM, fast marching method is used [19].

	total time (sec.)	for 1 update (msec.)	number of updating (reinitialization)
FLSM (Section 3.1)	1.625	3.983	408 (6)
FLSM (Section 3.2)	0.828	2.029	408 (6)
LSM Extension	1.829	4.324	423 (5)
LSM No Extension	2.828	7.106	398 (5)

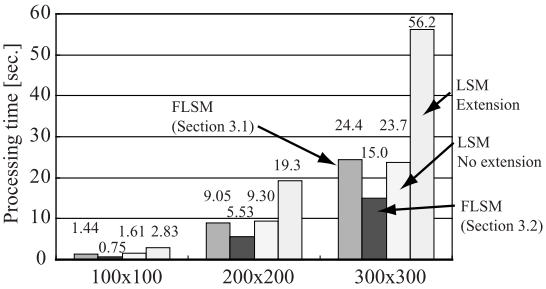


図 6 空間の大きさと計算時間の比較

Fig. 6 Comparison of processing time for various grid sizes.

表 3 Fast Level Set Method と従来の Level Set Method の計算時間の比較

Table 3 Comparison of calculation time between Fast Level Set Method (FLSM) and conventional Level Set Method (LSM) in the case where CFL condition is violated.

	total time (sec.)	for 1 update (msec.)	number of updating (reinitialization)
FLSM (Section 3.1)	0.1100	5.238	21 (21)
FLSM (Section 3.2)	0.0310	1.475	21 (21)
LSM No extension	0.1720	8.190	21 (5)
LSM Extension	0.1400	6.667	21 (21)

これより，従来の Level Set Method では CFL 条件が満たされない場合，更新を続けるうちに補助関数の形状が不安定になるが，Fast Level Set Method では拡張成長速度の利用と頻繁な再初期化により，CFL 条件が満たされない場合でも比較的安定して境界が抽出できることが分かる．

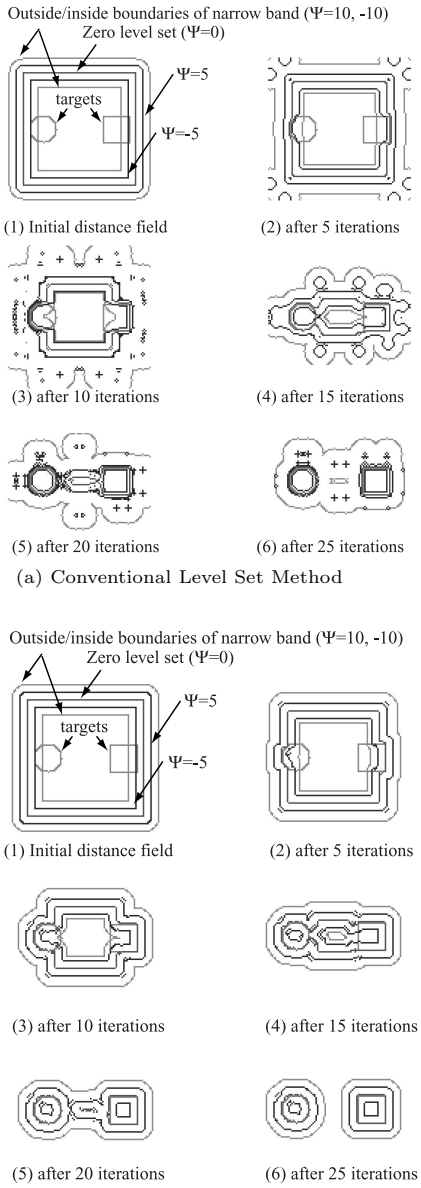


図 7 CFL 条件を満たさない場合の従来の Level Set Method と Fast Level Set Method のシミュレーション結果

Fig. 7 Simulation results of conventional Level Set Method and Fast Level Set Method in the case where CFL condition is violated.

4.2 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 320×240 pixel，30 Hz，使用した計算機は Pentium IV，2 GHz であり，Fast Level Set

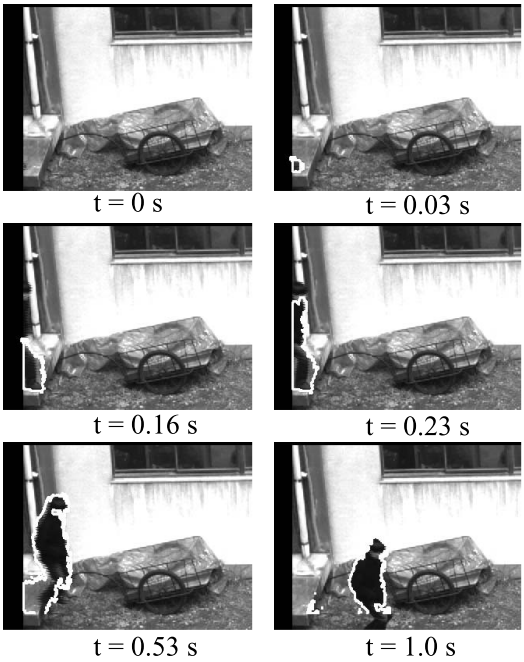


図 8 移動物体のリアルタイム追跡
Fig. 8 Realtime tracking of single object.

Method の処理は約 60 Hz 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．ただしこれらの設定した値は CFL 条件を満たしていない．

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，Fast Level Set Method により検出した移動物体領域の中心から外側へ zero level set を進行させて，濃淡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．ただし式 (13) で用いる輝度勾配項 k_I は，以下の式により背景差分画像 $D(x,y) = I(x,y) - I_{org}(x,y)$ と濃淡画像 $I(x,y)$ の両方から決定した．

$$k_I = \frac{1}{1 + \min(|\nabla I(x,y)|, |\nabla D(x,y)|)} \quad (16)$$

図 8 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分かる．また**図 9** 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．

これらの結果から，拡張成長速度場の利用と再初期化処理を頻繁に行うことで，CFL 条件を満たさない場合でも実用上問題なく安定に移動物体を追跡できることが分かる．

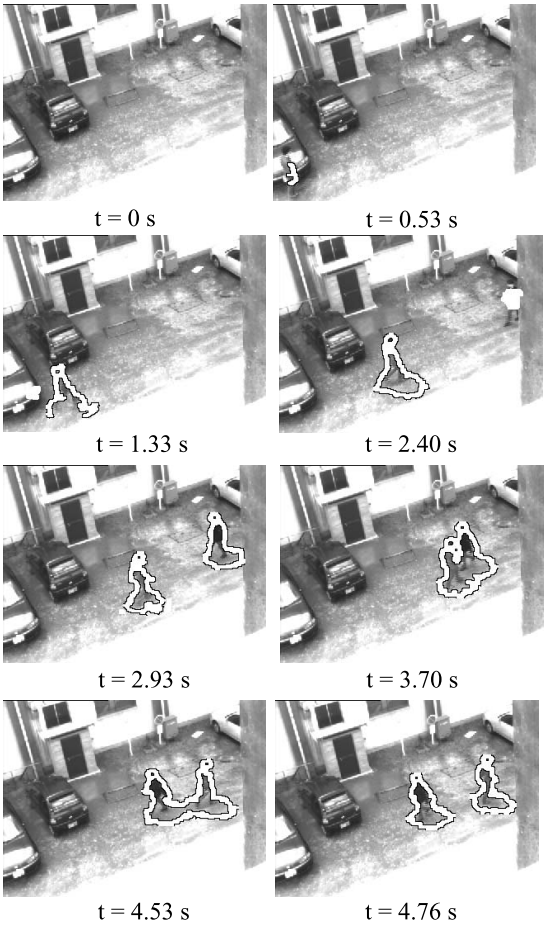


図 9 複数物体のリアルタイム追跡
Fig. 9 Simultaneous tracking of multiple objects.

5. まとめ

本論文では高速で安定な Level Set Method の解法として，i) Extention Velocity（拡張成長速度）の利用と高速な Extention Velocity Field（拡張成長速度場）の構築，ii) 補助関数の再初期化処理の高速化と頻繁な再初期化，を特徴とする新たな Fast Level Set Method を提案し，本手法がこれまでに提案された Level Set Method の手法以上の高速な収束性能を有することを示した．また提案する手法の応用例として，ビデオ画像上の移動物体のリアルタイム追跡への適用例を示した．

本手法は解像度制御などによりさらに高速化でき，今後は提案した Fast Level Set Method を用いて，複数ステレオ画像を安定に統合した全方向モーション

キャプチャシステムを構築する予定である．

謝辞 本研究は，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」の一環として実施された．

参 考 文 献

1) Adalsteinsson, D. and Sethian, J.: A fast level set method for propagating interfaces, *Journal of Computational Physics*, Vol.118, pp.269–277 (1995).
2) Adalsteinsson, D. and Sethian, J.: The fast construction of extension velocities in level set methods, *Journal of Computational Physics*, Vol.148, pp.2–22 (1999).
3) Baillard, C., Hellier, P. and Barillot, C.: Cooperation between level set techniques and dense 3d registration for the segmentation of brain structure, *International Conference on Pattern Recognition*, pp.991–994 (2000).
4) Chop, D.: Computing Minimal Surfaces via level Set Curvature Flow, *Journal of Computational Physics*, Vol.106, pp.77–91 (1993).
5) Cohen, L.D. and Cohen, I.: Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images, *IEEE Tranaction on Pattern Analysis and Machine Intelligence*, Vol.15, No.11, pp.1131–1147 (1993).
6) Johan, M., Herve, D., Nicolas, S. and Nicholas, A.: Representation, Shape, Topology and Evolution of Deformable Surfaces, Application to 3D Medical Image Segmentation, Technical Report, INRIA Report (2000).
7) Kass, M., Witkin, A. and Terzopoulos, D.: Snakes, Active contour models, *International Journal of Computer Vision*, Vol.1, No.4, pp.321–331 (1988).
8) Kimmel, R., Kiryati, N. and Bruckstein, A.M.: Analyzing and Synthesizing Images by Evolving Curves with the Osher-Sethian Method, *International Journal of Computer Vision*, Vol.24, No.1, pp.37–55 (1997).
9) Lu, T., Neittaanmaki, P. and Tai, X.C.: A parallel splitting up method and its application to Navier-Strokes equation, *Applied Mathematics Letters*, Vol.4, No.2, pp.25–29 (1991).
10) Malladi, R., Sethian, J. and Vemuri, B.C.: Shape Modeling with Front Propagation: A level Set Approach, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.17, No.2, pp.158–175 (1995).
11) McInerney, T. and Terzopoulos, D.: Deformable models in medical image analysis, *Medical Image Analysis*, Vol.1, No.2, pp.91–108 (1996).
12) Museth, K., Breen, D., Whitaker, R.T. and Barr, A.H.: Level Set Surface Editing Operaters (2002).
13) Osher, S. and Sethian, J.A.: Fronts propagating with curvature dependent speed: Algorithm based on Hamilton-Javobi formation, *Journal of Computational Physics*, Vol.79, pp.12–49 (1988).
14) Paragios, N. and Deriche, R.: A PDE-based Level Set approach for Detection and Tracking of moving objects, *International Conference on Computer Vision*, pp.1139–1145 (1998).
15) Paragios, N. and Deriche, R.: Geodesic Active Regions for supervised texture segmentation, *International Conference on Computer Vision*, pp.926–932 (1999).
16) Paragios, N. and Deriche, R.: Geodesic Active Contours and level sets for detection and tracking od moving objects, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.22, No.3, pp.266–280 (2000).
17) Peng, D., Merriman, B., Osher, S., Zhao, H.K. and Kang, M.: A PDE-based fast local level set method, *Journal of Computational Physics*, Vol.155, pp.410–438 (1999).
18) Sethian, J.: A fast marching level set method for monotonically advancing fronts, *Proc. National Academy of Science*, Vol.93, pp.1591–1595 (1996).
19) Sethian, J.: *Level Set Methods, 1st ed.*, Cambridge University Press, New York (1996).
20) Sethian, J.: *Level Set Methods and Fast Marching Methods second ed.*, Cambridge University Press, UK (1999).
21) Sethian, J. and Adalsteinsson, D.: An Overview of Level Set Methods for Etching Deposition, and Lithograpy Development, *IEEE Trans. Semiconductor Devices*, Vol.10, No.1, pp.167–184 (1996).
22) Weickert, J., ter Haar, B.M., Romeny and Viergever, M.A.: Efficent and reliable schemes for nonlinear diffusion filtering, *IEEE Trans. Image Processing*, No.3, pp.398–410 (1998).
23) Whitaker, R.: A level-set approach to 3D reconstruction from range data, *International Journal of Computer Vision*, Vol.29, No.3, pp.203–231 (1998).
24) Yui, S., Hara, K., Zha, H. and T. Hasegawa: A fast narrow band method and its application in topology-adaptive 3-D modeling, *International Conference on Pattern Recognition*, pp.122–125 (2002).
25) Zhao, H.K., Osher, S. and Fedkiw, R.: Fast

surface reconstruction using the level set method, *Proc. 1st IEEE Workshop on Variational and Level Set Methods* (2001).
26) 小林　景，杉原厚吉：情報重みつき結晶成長ボロノイ図の近似構成法とその応用，電子情報通信学会論文誌，No.12 (2000).

付　　録

A.1 CFL 条件

PDE を Upwind scheme を用いて数値的に安定に解くためには，積分時間幅が以下の CFL（Courant-Friedrich-Levy）条件を満たす必要がある[8]．

$$1 \geq \Delta t \left(\frac{|\ H_u\ |}{\Delta x} + \frac{|\ H_v\ |}{\Delta y} \right) \tag{17}$$

ただし Δt は積分時間幅，$H(u,v) = \sqrt{u^2+v^2}F = F\ \|\ \nabla\Psi\ \|$ は Hamiltonian，Ψ は補助関数である．もし空間が均一なグリッドに分割されていれば，$\Delta x = \Delta y = 1$ と考えることができ，このとき安定な積分時間幅の最大値は，全積分領域での Hamiltonian の勾配の最大値を用いて，

$$1 \geq \max(|\ H_u\ | + |\ H_v\ |)\Delta t \tag{18}$$

で与えられる．

(平成 15 年 2 月 17 日受付)
(平成 15 年 6 月 3 日採録)

倉爪　亮（正会員）
1967 年 2 月 4 日生．1991 年東京工業大学機械物理工学専攻修士課程修了．同年（株）富士通研究所入社．1995 年東京工業大学機械宇宙学科助手，2000 年スタンフォード大学客員研究員，同年東京大学生産技術研究所博士研究員，2002 年九州大学大学院システム情報科学研究院助教授，現在に至る．群ロボット，歩行機械，レーザ計測の研究に従事．日本ロボット学会，日本機械学会会員．

由井俊太郎
2001 年九州大学工学部電気情報工学科卒業．2003 年同大学大学院システム情報科学府修士課程修了．同年（株）日立製作所中央研究所入社，現在に至る，3 次元物体幾何モデリングおよび医療情報システムの研究に従事．日本ロボット学会会員．

辻　徳生
1977 年 10 月 6 日生．2000 年九州大学工学部電気情報工学科卒業．2002 年同大学大学院システム情報科学府知能システム学専攻修士課程修了．現在，同博士後期課程在学中．コンピュータビジョン，コンピュータグラフィクスの分野における研究に従事．日本ロボット学会，電子情報通信学会，各学生会員．

岩下　友美
1979 年 5 月 8 日生．現在九州大学大学院システム情報科学府修士課程在学中．マルチモーダルコミュニケーションとコンピュータビジョンの研究に従事．日本ロボット学会学生会員．

原　健二（正会員）
1987 年京都大学工学部数理工学科卒業．1989 年同大学大学院修士課程修了．同年武田薬品工業（株）入社．1991 年福岡県工業技術センター入所．2001 年東京大学生産技術研究所協力研究員，現在に至る．1999 年九州大学大学院システム情報科学研究科博士課程修了．工学博士．コンピュータビジョンとコンピュータグラフィックスの研究に従事．電子情報通信学会，IEEE 各会員．

長谷川　勉（正会員）
1950 年生．1973 年東京工業大学電子物理工学科卒業．同年電子技術総合研究所勤務．1992 年より九州大学工学部情報工学科教授．現在同大学大学院システム情報科学研究院教授．知能ロボットの研究に従事．工学博士．計測自動制御学会，電気学会，日本機械学会等の会員．