イメージ・プロセッシング20−2
（1978. 9. 19）

計算機動画像処理による，移動するスイ細胞顆粒像の解析

高木幹雄　・　坂上勝彦
（東京大学 生産技術研究所）

1．はじめに

一般に動画像処理とは何らかの方法で準備した時間的に差のある複数枚の画像間の変化を解析することを意味する。近年ハードウェアコストの低下によりICメモリーを使ったフレームメモリーが急速に広まりつつあり，TVカメラ，あるいはVTRなどからフレーム単位でディジタル画像を取込むことが可能となっている。この技術によって，今後ディジタル動画像処理の応用分野はさらに広がるものと考えられる。

本研究では，動画像の例としてスイ細胞内にある移動する顆粒を取上げ，その2次元方向の動きを追跡する。この顆粒の数は非常に多く人手でそれぞれの移動粒子を追跡するのは極めて困難である。特に扱う画像の枚数が多くなった場合には計算機による処理がどうしても必要となる。この目的で，時間的に連続したディジタル画像から移動する粒子を自動的に抽出し追跡するアルゴリズムを開発した。同じ様な微少移動粒子を追跡するシステムとして Bugsystem[1] が挙げられるが，これは2値化によって簡単に粒子を抽出できる単純な画像を対象としており，扱う粒子数もそれほど多くない。本稿で提案するアルゴリズムは，たくさんの顆粒の中から移動する顆粒を検出し，さらに顆粒の重心を求めるときなどに必要な閾値はそれぞれの顆粒についてアダプティブに決定されるように考慮されている。

サンプル動画像は連続写真をメカニカルスキャナーでディジタイズすることによって準備したが，フレームメモリーを持つカラーディスプレイ装置の使用を想定した手法であり，そのハードウェアによる演算機能及び専用コントローラによるVTRとの接続によりさらに効果的な移動解析が将来は行なえる予定である。

2．スイ細胞顆粒について

本研究で対象とするのはスイ臓の内分泌細胞である。この細胞を培養液で灌流しつつ顕微鏡下にセットし，1秒間隔で360枚の連続写真が撮影されている。今回はこのうち最初の10枚を解析した。

図1に写っている一面に散らばった暗い粒子が顆粒であり，直径は0.2μm程度である。これらの顆粒は内分泌細胞内のリボゾーム上で作られたインスリンを細胞膜まで運ぶ働きをし，その動きを定量的に把握することがホルモン分泌機構解析の上で極めて重要な課題となっている。しかし図1を見てわかるように顆粒の数は極めて多く（図1には500以上の顆粒が写っている）人手ですべての移動顆粒を追跡するのは極めて困難であり どうしても計算機による処理が必要となるのである。

動画像としての顆粒には次のような特色がある。

(1)　顆粒の移動そのものを追跡したい。
顆粒の移動から細胞質の流れを知りたいのではなく顆粒の動きそのものが知りたいのである。よって1つ1つの移動を完全に追跡する処理方法を開発する必要がある。ただし重心だけでよい。

(2) 顆粒はバックグラウンドよりずっと暗い粒状物質である。
この特徴のため移動検出は比較的容易に行なえそうである。

(3) 移動速度にはかなりの開きがある。
ほとんど動かないものから,かなりのスピードで動くものまでいろいろなものがあり，そのどちらも検出できるような処理方法を開発する必要がある。

(4) 顆粒は広い領域に散らばっている。
つまり全領域を一度に処理することはむずかしい。

(5) 3次元方向の移動追跡は困難である。
本稿では深さ方向への動きは無視している。

(6) 粒子の大きさは一定でない。
深さ方向への移動のため粒子の大きさが変化して見えることがあり得る。さらに発生，消滅もあり得る。

(7) 移動のパターンにはいろいろなものがある。
ほぼ同じ速度で同じ方向に移動するものや不規則な動きをするものなどいろいろなものがある。

ヌイ細胞顆粒は以上のような特色を持つ移動物体であり，動画像処理の立場からモデル化すると，『バックグラウンドと濃度の異なる複数個の粒状物体の2次元方向の移動解析』ということになる。次章でその処理方法について解説する。

3. 処理手順

3.1 ヌイ細胞顆粒移動追跡の問題点

顆粒の数は極めて多く，他の移動追跡の対象の場合とは異なる問題点が数多くある。

(1) 同じ形でしかも速度が異なる粒子がたくさんある中で，いかにして動いたものを検出し，さらに2枚の画像間でどの粒子とどの粒子が実際に移動したpairであるかをいかにして判定するか。

(2) 追跡処理を直接行なうことのできるのはハードウェア，ソフトウェアの制限から一般に128×128程度の正方形画像に限られてしまう。しかし実際に粒子の追跡を行ないたい領域はさらに広く，このような領域に対し処理が行なえなければ意味がない。

(3) 動画像データにおいて画面数が多くなるともはや各画面に対し対話形でパラメータを設定することはできない。よってアルゴリズムは，自動的に追跡が行なわれしかも本質的に誤りの生じないものでなくてはいけない。

(4) 連続写真で与えられた動画像をスキャナーで入力する場合には画像のregistration（位置合せ）という重要な問題がある。

これらの問題点を考慮しつつ追跡用ソフトウェアを開発した。

3.2 動画像の取込み

まず10枚の連続写真（time 1 ～ time 10）をスキャナーを使って入力した。この写真の全領域について調べたいし，またregistrationのためのアドレスも知りたい。そこで全領域をline density = 13本/mm，画像サイズ = 2048×2250でディジタイズ（8bit），MTに格納した。図1はtime 1のディジタル画像を9点おきにサンプルして蓄積管に表示したものである。ただし後述する20のsubregionの境界線を重ねて表示してある。

取込まれた動画像に以下の処理が行なわれる。3.4～3.7は時間的に近接した

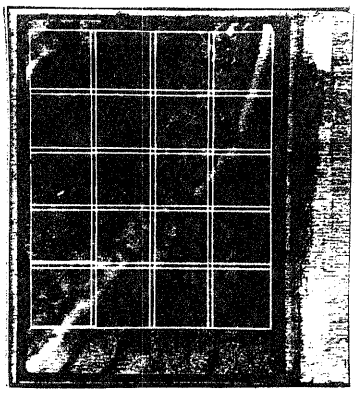


図1. スイ細胞顆粒ディジタル画像(time 1)と 20のsubregion.

subregion# 1 2 3 4
5 6 - -
-

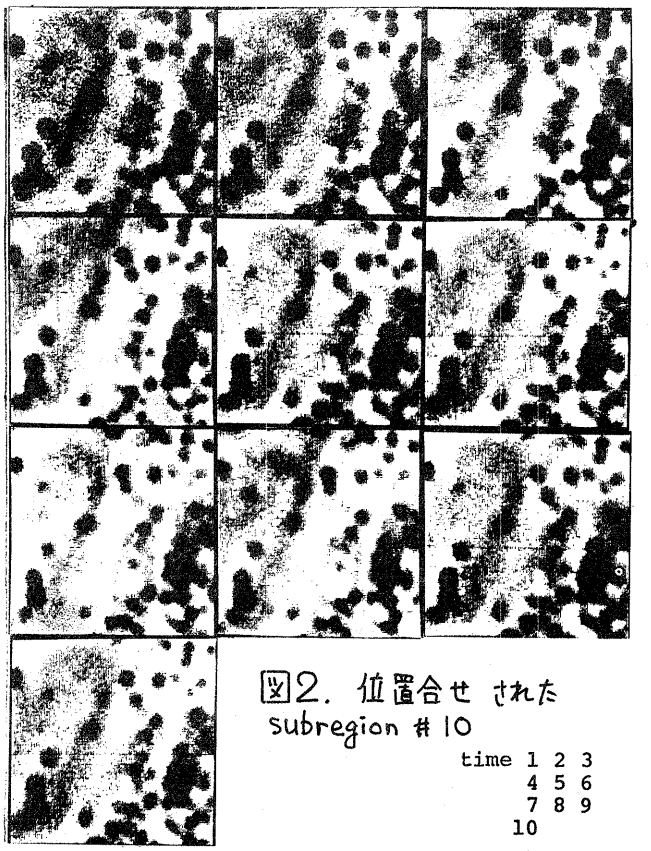


図2. 位置合せされた subregion #10

time 1 2 3
4 5 6
7 8 9
10

2枚の画像間で行なわれる処理である。3.3～3.8はある1つのsubregionにおける処理でありこれをsubregionの数だけ繰返す。

3.3 連続画像の準備

直接追跡処理を行なう画像は原データ(2048×2250)から 128×128, sample rate = 3 で切出した。time 1 における Subregion の切り方は図1に示した。各subregionにはそれぞれ30画素巾の重なりを持たせてある。これは境界付近の粒子をスムーズに追跡するためのものである。

registration はアフィン変換を使って行なった。この場合何らかのマーカーが必要となるが、図1で周辺の黒枠の内側の線はカメラのフレームであるため位置合せの情報として使うことができる。図2は registration 処理後の subregion #10 の time 1 ～ time 10 である。それぞれ画像サイズは 128×128 である。

3.4 時間微分

まず画像間の減算を行なって移動があった部分を強調した画像を作る。図3-3は人為的に作った移動粒子データの time 2(図3-2)から time 1(図3-1)を減算したものである。この画像に対し閾値処理を施す。まず図3-3のposi側について行なう。すなわちある閾値よりも値の大きい画素のアドレスを抽出するのである。その画素数が256以内に収まるように閾値は自動的に調整される。次に nega 側についても同様にある閾値よりも値の小さい画素のアドレスを抽出する。posi側から抽出したアドレスの集合を backward addresses, nega側から抽出したアドレスの集合を forward addresses と名づける。backward addresses と forward addresses の内容をプロットしたのが図3-4である。白く塗りつぶしてあるのが backward 側である。これを見てわかるようにこのアドレスの情報から各粒子の移動の様子がほぼわかるのである。図3-4において各粒子に対しほぼ同じ形のアドレス

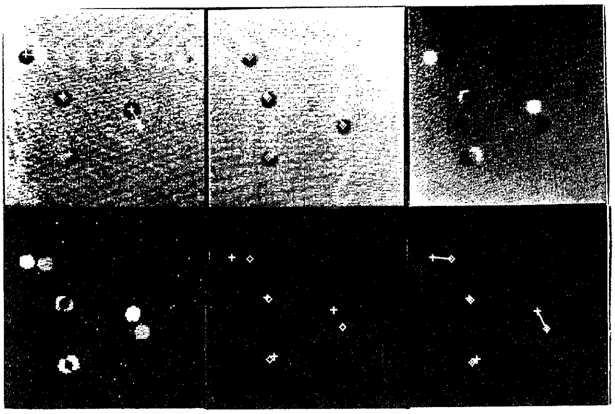


図3　人為的に作った移動粒子データ（time 1 , time 2 ）.

図3-1 time 1.（ベクトルを重ねて表示）

図3-2 time 2.

図3-3 time 2 − time 1.

図3-4 抽出されたアドレス.
□:Backward
⁝⁝:Forward

図3-5 shadowの重心.
+ : Backward
◇ : Forward

図3-6 移動ベクトル.
+ : 始点
◇ : 終点

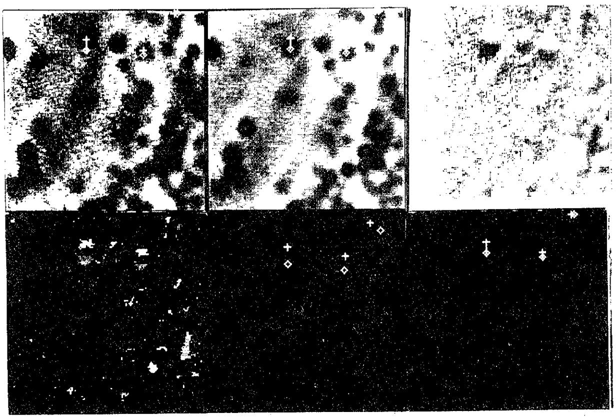


図4　Subregion #10 （time 2 , time 3 ）.

図4-1 time 2.

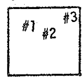
図4-2 time 3.

図4-3 time 3 − time 2.

図4-4 抽出されたアドレス.

図4-5 shadowの重心.

図4-6 移動ベクトル.



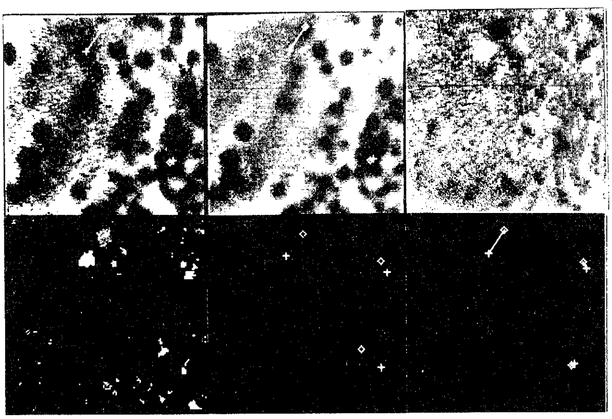


図5　Subregion #10 （time 3 , time 4 ）.

図5-1 time 3.

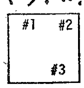
図5-2 time 4.

図5-3 time 4 − time 3.

図5-4 抽出されたアドレス.

図5-5 shadowの重心.

図5-6 移動ベクトル.



のかたまりが pair になっている。backward 側のかたまりを backward shadow forward 側のかたまりを forward shadow と名づける。言うまでもなく移動の方向は backward shadow から forward shadow の方向である。

図3のような単純な画像の場合はきれいな shadow (backward と forward) を抽出できる。しかし実際の画像ではこれほどうまくはいかない。(図4-4, 図5-4) 雑音によって,移動とは関係のない画素のかたまりがかなり多く現われている。しかし backward shadow と forward shadow の pair は他の雑音とは違ってかなり明確な特徴を持っている。

こうして抽出したアドレスは追跡処理に先だってファイル (Address File) にたくわえておく。移動量の小さい粒子については shadow を抽出することができない。そこで差分とアドレス抽出操作は時間的にやや離れた画面間でも行なうことが必要である。今回のサンプルに対しては,時間間隔1秒と2秒について時間微分を行ない,抽出したアドレスをファイルに記録した。

カラーディスプレイ装置[5]は画像間の減算及び ある閾値以上の値を持つ画素のアドレスを順次補助メモリー (256 w × 18 bit) に記録する機能をハードウェア的に持っており,これを使うことにより時間微分の部分はかなりの計算時間短縮が期待できる。

本手法では2枚の画像間の変化を一律に "256×2 画素のアドレス" という形にコンパクトにまとめてしまい,以下の処理では画像間の変化に関する情報はすべてこのアドレスから得ている。記憶させるアドレス数を制限するということは処理の融通性を考えると不利かも知れないが,後の処理において極めて扱いやすいデータであり,膨大な数の連続画像の画像間変化の記録方式として極めて有効なものであると考えられる。

3.5 移動の検出

実際のスイ細胞顆粒像について時間微分をすると shadow 以外の画素がかたまりとなって抽出されてしまうことはすでに前節で述べた。本節では抽出したアドレスを調べ真の移動による shadow すなわち backward shadow と forward shadow の pair を見つける手法について解説する。

1) セグメンテーション　backward addresses と forward addresses の内容をそれぞれアドレスの隣接性を使って群分けし同定番号をつける。このとき各セグメントの面積,重心,楕円近似した場合の長軸/短軸 及び長軸の角度を特徴量として同時に計算する。面積がある値 (IL) より小さいものと,ある値 (IH) より大きいものはこの段階で雑音として delete してしまう。

2) 対応テーブルの作成　1)で番号付けされた backward 側のセグメントと forward 側のセグメントのうち backward shadow と forward shadow の関係にあるセグメントを見つけ,それを記述する対応テーブルを作成する。これにはまず backward 側の あるセグメントに対し forward 側のセグメントの中で最も重心間の距離の近いものを選び出す。この pair を図6の Decision Tree に通し,真の pair であると判定されれば backward 側から forward 側への対応テーブルに記録する。真の pair でないとみなされれば次に距離の近いものについて調べる。さらに逆方向つまり forward 側の各セグメントから backward 側の各セグメントへの対応テーブルを作成し,両テーブルが hand shake したセグメント同士を backward shadow, forward shadow の pair とみなし対応テーブルに記録する。

Decision Tree は次のような考え方で作られた。図4-4を見てわかるように,

移動量が粒子の直径以下の場合 shadow はそれぞれ長円形となりその主軸はどちらもほぼ同じ方向を向いている。またどちらも重心間を結ぶ直線に対しほぼ直角である。ただし主軸の方向を考慮することができるのはセグメントが仮に長円形になっている場合だけである。また forward と backward shadow の面積はほぼ同じである。重心間の距離もある程度以上離れているものは pair とみなすことはできない。

このようにして検出された移動による shadow の pair を図3-5, 4-5, 5-5 に示す。ただし各閾値は以下の値を用いた。

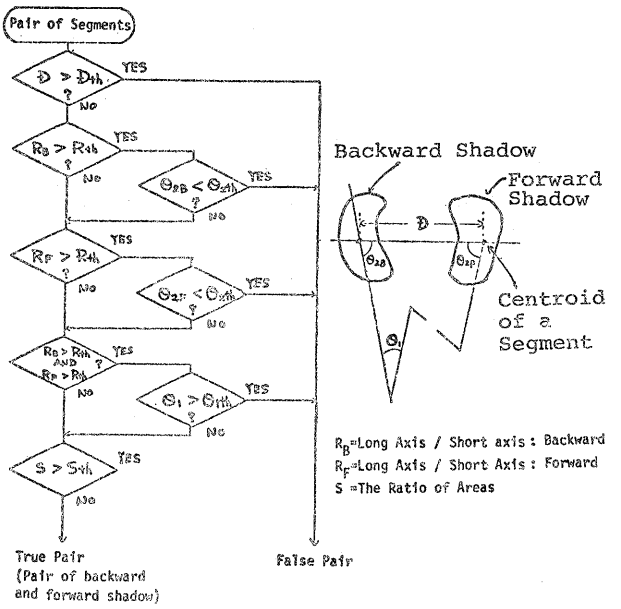


図6 Decision Tree

IL = 10 (画素) : 面積の下限
IH = 200 (画素) : 面積の上限
Dth = 20 (画素) : 重心間の距離, Rth = 1.5 : 長軸/短軸 (長円形性判定)
θ1th = 60 (deg) : 主軸の平行性判定, θ2th = 45 (deg) : 主軸-OO' の直角性判定
Sth = 4 : 面積比

図4における2個のほぼ同方向への移動粒子の間にある粒子の移動, また図5の最も大きい移動粒子のすぐ右側にある粒子の移動はどちらもセグメントの面積が10に満たないため検出されていないが, 両方とも2秒間隔の画面について差分をとった場合には検出されている。

各閾値のうち特に移動検出能力に強く影響するのは θ1th と θ2th である。上記の値は極めてゆるい閾値であるが, この場合に人間と全く同じ検出能力が得られた。

3.6 移動ベクトルの計算

粒子の移動ベクトルは SSDA[2] を用いて求める。前節「移動の検出」の段階で移動粒子の始点と終点についてのおおよその情報がすでに得られている。これを利用し, さらに閾値の自動決定法[3]を応用すると SSDA はより効果的になりしかも処理を完全に自動化することができる。

本システムでは次のようにして移動ベクトルを求めた。窓画像としては backward shadow の重心 G_B を中心とした 8×8 の領域を使う。これを forward shadow の重心 G_F から G_B まで移動させ最小の残差と2番目に小さな値を与える2点を選ぶ。つぎにそれらの点を中心として上下左右に1画素ずつ移動させ, その中で最小値を与える点をとって同様の1画素ずつの移動をくり返す。はじめの2点からのそれぞれのくり返しが終わった時点で残差の小さい方を選び, そこにおいて重ね合せが達成されたとみなす。

shadow が真の移動によるものであれば窓画像を G_F に重ね合せたときに閾値が一挙に引き下げられると考えられ極めて効果的な SSDA が行なわれると考えられる。また shadow が真の移動によるものでない場合でも, 上下左右への1画素ずつの窓画像の移動の段階で2枚の画像間で同一粒子を重ね合せることができるであ

ろう。これによって粒子がほとんど移動していないことがわかればこの段階でその粒子は対応テーブルから削除することができる。

以上の手法で移動ベクトルはほぼ正確に求めることができる。次に問題となるのはベクトルの始点の求め方である。近似的にはbackward shadowの重心（G_B）でもよいと考えられるが本システムでは次のような簡単な方法で粒子の重心を求めた（図7）。これは直線G_BG_Fと粒子の輪郭との交点A,Bを求めその中点Mを粒子の重心とする方法である。ただし雑音による影響をさけるために直線は3本並べて引き，最もG_Bから離れている点をそれぞれA,Bとした。半径r（図7）から近似的に粒子の面積を求めることもできる。輪郭追跡によって粒子の重心と面積を求める方法も試みてみたが，思わぬところに迷い込んでしまうことがかなりあり使えないことがわかった。

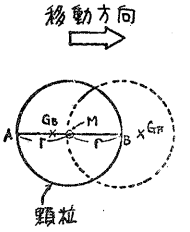


図7
移動ベクトルの始点．

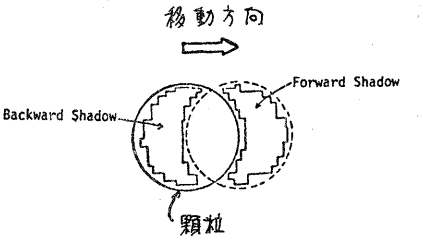


図8
閾値の自動決定．

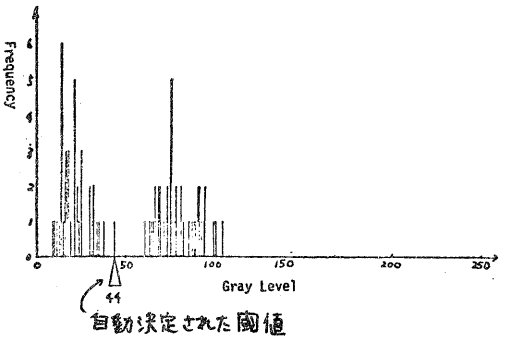


図9 Shadowに相当する画素のヒストグラム．

粒子の輪郭を検出するための閾値は各顆粒について自動決定を行なう。図8でbackward shadowの画素は顆粒内の画素であり，forward shadowの画素はバックグラウンドに相当する。よって両shadowのすべての画素についてのヒストグラムは典型的な2モード型となり容易に閾値を自動決定することができる。ここでは大津[4]によるアルゴリズムを用いた。図9は図4で検出されている移動のうち一番左側にあるものについて求めたヒストグラムと算出された閾値を示したものである。図3-6，4-6，5-6は実際に移動ベクトルを求めた例である。図5-6の#3は顆粒の固まりの部分でありSSDAの結果ほとんど動きがないと判定されている。その他の部分はすべて正しく移動が追跡されている。移動の方向もより正しい方向に微妙に調整されているのが図4-5と図4-6を比較してみるとよくわかる。

3.7 ベクトルの記録

算出された各顆粒の移動ベクトルはすべてベクトルテーブルに記録される。ベクトルテーブルはディスク内にファイルとして用意されている。

3.8 トラッキングテーブルの作成

ある1つのsubregionについてのベクトルのテーブルへの記録が終わると次の段階としてこのテーブルに対する処理を行ないベクトルをつなぎ合わせて移動粒子に番号をつける。処理は次のような手順で行なう。

(1) ベクトルテーブルの冗長をなくす。

処理前のベクトルテーブルには図10に示すような冗長がある。まず粒子が図10-1のような動きをした場合3つのベクトル$\vec{AB}$，$\vec{BC}$，$\vec{AC}$はいずれもベクトルテーブルに記録される。しかし図から明らかなようにベクトル$\vec{AC}$は考慮する

必要のないベクトルであり テーブルから削除しなくてはならない。図10-2は time i-2 から time i-1 の間に粒子が動かなかったか、あるいは移動量が小さいために検出されなかった場合である。このときベクトル $\vec{AB}$ はベクトルテーブルの2つのスロット（i-2, 2）と（i-1, 1）

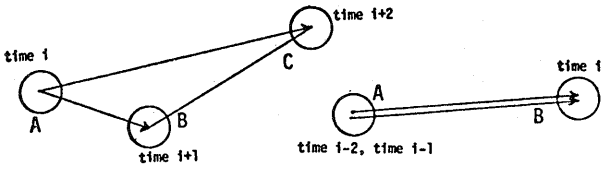


図10-1　　図10-2

図10　ベクトルテーブルの冗長。

（ただし スロット（i, j）には time i と time i+j のフレーム間での粒子の移動ベクトルが記録されている）に記録されてしまう。そこでこのような場合にはスロット（i-2, 2）に記録されているベクトル $\vec{AB}$ をテーブルから削除する。このようにして1つの移動を時間的に重複する2つ以上のスロットで記述することのないようにベクトルテーブルを整理するのである。

(2) ベクトルの連続性を調べ 同じ粒子の移動に相当するベクトルに同じ番号を付ける。

ベクトルの連続性はあるベクトルの終点と時間的にこれに続くベクトルの始点との距離によって判定することもできるが、これだと場合によっては誤りが生じることもあり得るし、距離の閾値の選定にも問題があろう。ここでは誤りが少なく、より正確な方式を提案する。

今、顆粒が time i-1, time i, time i+1 で図11のように移動したとする。このときベクトルaの forward shadow とベクトルbの backward shadow が図11のようにできる。これを見てわかるようにベクトルaとbが同一顆粒によるベクトルであることを証明するには、2つの shadow が time i の画面上で1つの顆粒内にあることを証明すればよい。これには次の2つの条件を用いる。

① 2つの shadow に1画素でも重なりがある。

② まず2つの shadow にはさまれた領域を抜き出す。これは両 shadow を合わせた図形の外接曲線（外側から輪ゴムをはめたときにできる曲線；図13で＋）の内部の画素から shadow を除けば得られる（図13で◇と×）。これらの画素のう

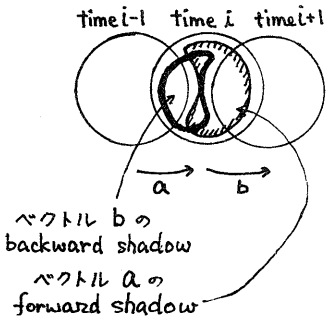


図11　ベクトルの連続性。

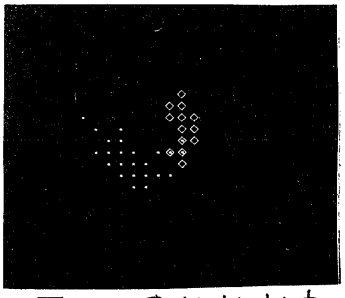


図12　連続性判定（条件①）

◇ : forward shadow

■ : backward shadow

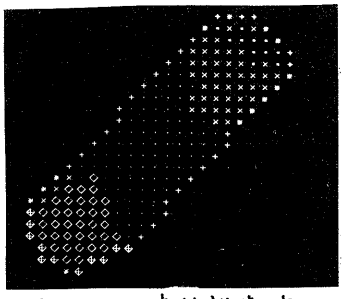


図13　連続性判定（条件②）

◇ : forward shadow

■ : backward shadow

＋ : 外接曲線

ちベクトルa及びbの始点を求めるときに用いた閾値の平均値によって顆粒内であると判定された画素（図13で　x　）の割合が80%を越える。

条件①，②のどちらかを満たした場合にベクトルa，bが同一顆粒によるものであると判定する。ただし②は比較的時間がかかるため①だけでは判定できなかったものについて②を調べるプログラムにしてある。図12は①の条件を満たした例である。このとき2つのshadowは同一顆粒内にある。図13は②の条件を満たさなかった例で，閾値にひっかかった画素の割合は0.345であった。2つのshadowは明らかに異なる顆粒に属している。実際の例では①を満たさず，②だけを満たして連続性が確認されたベクトルは極めて少なかった。

以上の処理が終了したベクトルテーブルの内容は，全調査領域における顆粒のトレースを記録するトラッキングテーブルに書き加えられる。2つのsubregionに渡って移動する顆粒のトレースの結合もこの段階で行なわれる。

4. 追跡結果

図14は人為的に作った移動粒子データに対して作成したトラッキングテーブルの内容をプロットしたものである。図15はスイ細胞顆粒画像のsubregion #6，7，10，11についてそれぞれベクトルテーブルを作り，それらを結合して作ったトラッキングテーブルの内容をプロットしたものである。人為的に作ったデータに対しては完全な追跡が行なわれているのがよくわかる。図15のtrace Aは図4-6，図5-6の#1のベクトルがつなげられたものであり，time3からtime4の間に今までと逆方向に大きく移動した様子がよくわかる。

5. おわりに

以上スイ細胞顆粒（一般的には移動する粒状物体）の移動追跡法について述べた。結論として次のようなことが言える。

(1) 2枚の時間差のある画像間の差分をとり，その値の大きい（小さい）画素のアドレスを定まった数（本研究では256個）抽出すれば，それに対して処理を加えるだけで移動した粒子の位置，移動方向についてのかなり正確な情報を得ることができる。画像差分及びアドレス抽出はハード化による高速化が期待できる。またそのアドレス処理は完全にコア内で行なえるため移動検出の部分は極めて高速に行なうことができると考えられる。

(2) 移動ベクトルの正確な値はSSDAを使って計算することによって極めてよい結果が得られた。また移動検出の処理において，発生した移動をすべて検出するためにはdecision treeにおける閾値をかなりゆるいものにしなくてはいけない。しかしこうすると移動していない部分を誤って移動とみなす可能性がある。これに対してはSSDAにより重ね合せが達成される場所が探し出され自動的に補正が行なわれる手法が効果的であることがわかった。

(3) 粒子の重心を求めるためにはその輪郭を検出する閾値を知りたいが，それには抽出したアドレスの

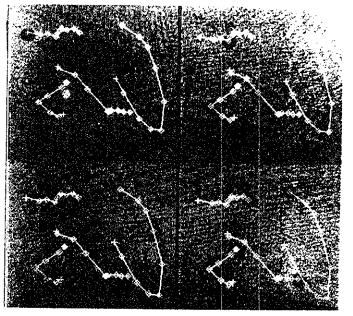


図14　人為的に作った移動粒子データに対する追跡結果．time1　time4　time7　time10

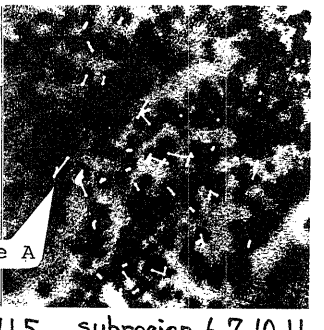


図15　subregion 6,7,10,11に対する追跡結果．

うち知りたい粒子の shadow に相当するものを選び，その画素の値を実際に調べることによってそれぞれの粒子にとって最も適した閾値を自動的に決定することができた。

(4) 移動ベクトルに対する番号付けを明確に行なう場合にも，抽出したアドレスが役に立つことがわかった。

(5) いくつかの小さい領域についての移動を記録したベクトルテーブルを合成してトラッキングテーブルを作るという手法により一度に処理できない全領域についての移動追跡が可能となった。

今後の課題としてまず追跡結果の表示方法があげられる。図15のようにトラッキングテーブルの内容をただ画像上にプロットしたのでは極めてわかりにくい。

また，registration には時間と手間が著しくかかるため，ＶＴＲ，ＴＶカメラ，カラーディスプレイを組込んだ本格的な動画像処理システムの開発を急がねばならない。

最後に日頃御指導頂く尾上教授，試料を提供して頂いた東大医学部 金沢先生に感謝致します。

References

(1) J.O.B.Greaves:"The Bugsystem: The Software Structure for the Reduction of Quantized Video Data of Moving Organisms", Proc. IEEE, Vol.63, No.10, Oct. 1975.

(2) D.I.Barnea and H.F.Silverman:"A Class of Algorithm for Fast Digital Image Registration", IEEE Trans. Compt., Vol.C-21, Feb. 1972.

(3) 尾上，前田，斎藤：“残差逐次検定法による画像の重ね合せ”，情報処理，Vol.17, No.7 (Jul. 1976).

(4) 大津：“濃度分布からの閾値決定法”，S52，通信学会 情報部門全国大会 145.

(5) 尾上，高木：“複数機能を有するカラーディスプレイ” 第7回 画像工学コンファレンス 1-2 (1976).