目 次

図目次

第 1 章　研 究 の 概 要

1.1　研究の目的

熱交換器，コンデンサの曲がり部には，流動液内に含まれる気泡の影響で腐食が生じやすい．また静的安全システムとして用いられる横置き形の蒸気発生器管内の流動は，自然循環形の原子炉などの安全解析上で重要な研究課題である．

気泡が管内液流体中に分散する気泡流現象をより詳しく知る必要性から，気泡流解析モデルは，気液を混合物として取扱う混合流モデルから，気液両相をそれぞれ別個の流体として取扱う二流体モデルが主流になりつつある．二流体モデルにおける気液相間の質量，運動量及びエネルギー輸送項をえるためには，気液相速度及び乱流密度などと並んで，気泡径，ボイド，気液界面面積濃度などの気泡形状の管断面及び管軸方向の空間的，時間的分布の実験的情報が必要となる．しかしながら，気泡の挙動と界面は複雑であり，散在する気泡の分布と形状を正確に測定することは難しい．

本研究では画像処理の手法で気泡の重心を求めて，気泡重心より気泡の空間運動分析を行う [1].

1.2　ステレオ撮影方法の利点

気泡流の研究において気－液二相運動輸送方程式を構築するため，管内液体中に分散する気泡の大きさ，位置などの挙動を測定するさまざまな研究が行われている［2］～ [4]. 従来電極式のボイドプローブが多く用いられているが，しかしながら，この方式には，気相流量が小さく，従って気泡径とボイド率が小さくなる低ボイド領域の気泡流においては，管内のプローブ挿入による液相の乱れのため，測定誤差が大きくなってしまう欠点がある [6]. また気泡径や気泡の界面挙動を直接知ることはできない．ダブルボイドプローブ法によって得られた弦長を基に球体を仮定して気泡直径を求める実験が行われているが，一般的な気泡流においては上述のように気泡は複雑な形状になり，気泡を球体として仮定できる条件はすくないと考えられる．一方，ビデオや写真撮影によって気泡形状を観測する実験も多く行われている [7]. 気泡の画像処理法には，散在気泡の界面を明確にすることが難しいこと，1 方向からの撮影では管断面内の 3 次元的分部を知ることができない欠点があったが，賞雅らは垂直細管を水平面上の直交する 2 つカメラで撮像し，ビデオ気泡画像を処理する方法を開発し，垂直円管内の 3 次元気泡挙動を測定した [4]. この方法は，空間情報を得られるため気泡径，界面形状を直接測定できることや非接触であるため流れを乱さないなどの利点があり，可視化可能なエネルギー線（Ｘ線，中性子線）を用いれば非可視壁状態における測定も可能である．この画像処理においては，気泡を楕円体と仮定して，気泡半径，体積及び界面面積が求められている．

$$\frac{r_p^2\cos^2 f\tilde{O}}{r_x^2}+\frac{r_p^2\sin^2 f\tilde{O}}{r_y^2}+\frac{z_p^2}{r_z^2}=1$$

図 1.1　　気泡の楕円体モデル

1.3　ステレオ方法での研究現状

[1]　１８０°ベント管内気泡流

大気圧常温の空気－水系１８０°ベント管内気泡流の気泡径，ボイド率及び界面面積濃度などの管断面分布をステレオ撮影された気泡画像処理することによって測定した [8]. ボイド率及び界面面積濃度は，気泡を楕円体と仮定してもとめた．画像解析の結果から，ベント管コーナ頂部の気泡は，管中心とベント内周のほぼ中心に分布し，ベント入り口付近でほぼ球体である気泡が偏平な形状になること，などが明らかになった．ベント各部の管断面平均気泡径は変化がなく，ベント各部の管断面の界面面積濃度分布は，ボイド率分布の変化とほぼ同様の変化をした．またレーザトップラー流速計により主流方向の単相液流速度分布及び二次流が強くなっていることを確認した．液単相流時の液気流速測定結果から，気泡流時の気泡の管断面位置と形状の測定結果を推察すると，気泡流時のベント管コーナ頂部の気泡位置は，ベントの遠心力と二次流れによる力との釣合いによるものであり，またその気泡形状は二次流れによって影響されているものでおると考えられた．

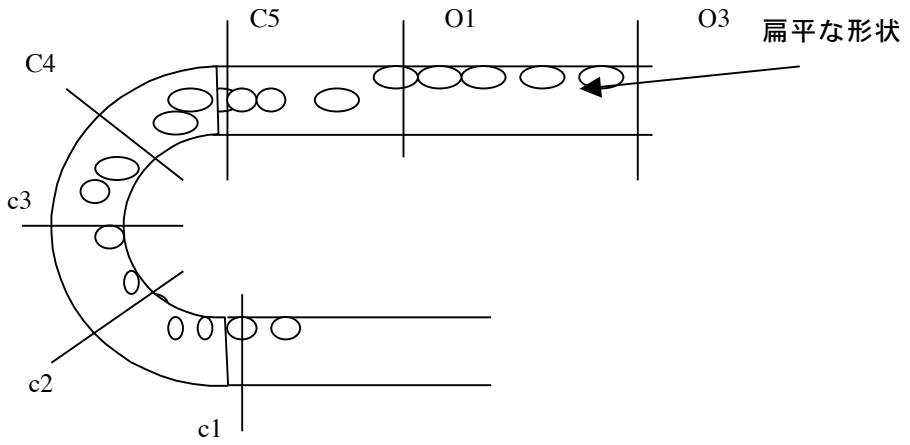


図 1.2　　ベント管内の気泡挙動

[2] 微小重力／通常重力下の気泡流界面測定

上昇気泡流内の気泡にかかる重力項を評価するために，（株）地下無重力実験センターの落下塔を用いて ,1-g とugにおけるガス－水系の気泡流実験を行った [9]. 気泡の径や管内位置，気泡のひずみなどは，ステレオ画像解析法（Stereo Image-processing Method,SIM）によって測定された．実験における特徴は以下のとおりである．

(1) 落下塔実験装置により残留重力のない微小重力環境を得ている．

(2) 1-g からugへ，さらにugから再び 1-g へという径時変化を測定している．
(3) 1-g とugでほとんど同じ気泡径の実験を行っている．
(4) ボイドプローブで測定の難しい 1 ～ 3 ｍｍ径の小さな気泡の管断面測定をしている．
実験結果より以下のことが明らかになった．
(1) ザウター平均径 1.6mm の小さな気泡においても，ボイド分布は 1-g では壁面にピークを持ったくら形分布に ,ug ではコアやくら形のピークは残るが管断面にほぼ一様な分布になる . ug は，気泡径は 1-g のときよりも小さくなる．
(2) 1-g では，気泡はドリフト速度のために管軸方向に延び，壁面に沿って下端を壁面粘性低層につけた底層気泡がある .ug では底層気泡はなくなる．
(3) 1-g においてもugにおいても，径の増加とともに気泡は偏平な形になる．
(4) Lin-Rezakullah に流体モデルを用いて気泡流のシミュレーションを行い，ほぼ実験結果と計算結果の傾向が合っていることを確認した．

今後の課題として，今回の気液流量実験範囲が狭かったのでこれを広げること，特に航空機実験では残留重力および横方向加速度のために計測しにくい低液流速域の気泡流を測定すること[9][10]. および初期気泡径の影響等を調べることがある．また計測された気泡の界面に関する情報の他に，落下塔実験では測定時間の制約のために非常に難しいが液相速度および乱れの同時計測が測定できれば ,ug の気泡流実験をより充実したものにすることができるだろうとしている．

[3] 気泡の衝突現象の研究

一般的に気泡が先行気泡の後流域に入ったとき，気泡は加速し先行気泡と衝突する .Otake らによると，先行気泡が後続気泡に顕著な影響を及ぼし始める臨界距離 L_{WE} がある（図 1.3).Wake entainment model による，界面面積濃度変化率 は次のように表され

$$\langle \Phi_{WE} \rangle = \left(\frac{\langle \alpha \rangle}{\langle a_i \rangle} \right)^2 \frac{\Gamma_{WE} \langle \alpha \rangle^2}{\langle D_b \rangle^4} \left(\langle v_g \rangle - \langle v_f \rangle \right) \exp\left(- K_C \sqrt[6]{\frac{\langle D_b \rangle^5 \rho_f^3 \langle \varepsilon \rangle^2}{\sigma^3}} \right)$$

る [.

1.4 本研究の手法

本研究では気泡の運動を分析するために，賞雅らのステレオ撮影方法により垂直細管内の気泡画像を処理して，単一気泡のより精密な重心を推定し，気泡の空間運動速度，加速度などの 3 次元測定を行う．

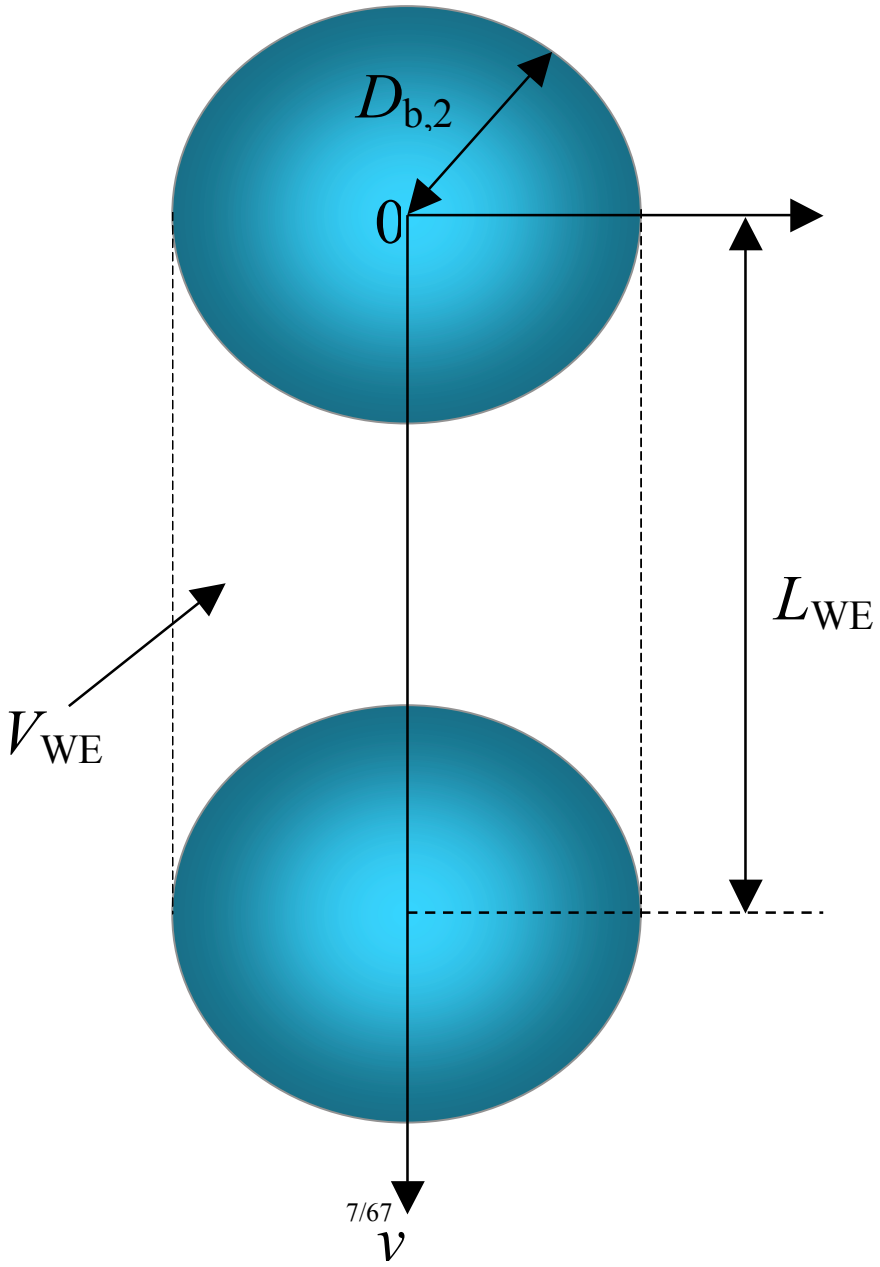

$D_{b,2}$
0
L_{WE}
V_{WE}
v

図 1.3　衝突気泡の運動

第 2 章　研 究 手 法 , 実 験 設 備 と 条 件

2.1 基本的な考え方

装置の概略を図 2.1 に示す．内径9mmの垂直細管内気泡流における気泡は，互いに直交する２つの方向から撮像される，この２つの画像 Image A と Image B を，平面座標系ｘ－ｚとｙ－ｚに対応させる．ｘ－ｙ座標系の原点は画像 image A の左下の点で，ｙ－Ｚ系の原点は画像imageBの右下の点である．実際は空間的に同一点である，ｚは気泡の上昇方向である．

同一時刻に気泡の重心点を表面 x － z 系とｙ－ｚ系に投影された位置を計算して，気泡重心の空間座標が求められる．

気泡の濃淡画像に対し平滑化，二値化等の前処理を施す．このデータについて輪郭線追跡し，気泡の表面形状に相当する画素を求め，含まれる画素で Image A と Image B で同定された対象気泡投影のそれぞれの重心点を求める．ここではこの気泡の重心運動を気泡の運動とみなす．

時刻 t_1 に Image A の平面ｘ－ｙに投影された気泡重心 w_1' （ x_1, z_1 ）とImageBの平面ｙ－ｚに投影された気泡重心 w_1'' （ y_1, z_1 ）を求め，これらより得られる重心点空間座標を $w_1(x_1, y_1, z_1)$ とする（図２．２）. 同様に時刻 t_2 の気泡の重心点空間座標を $w_2(x_2, y_2, z_2)$ とする．時間差Δｔ＝ｔ２－ｔ１，重心の空間位置座標差：Δχ＝χ２－χ１，Δｙ＝ｙ２－ｙ１，Δｚ＝ｚ２－ｚ１，そしてImage A, Image B の間の座標差から

気泡空間移動距離

$$s = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \qquad (2.1)$$

気泡空間移動速度 $v = s/\Delta t$ (2.2)

気泡空間移動加速度 $a = \Delta v/\Delta t$ (2.3)

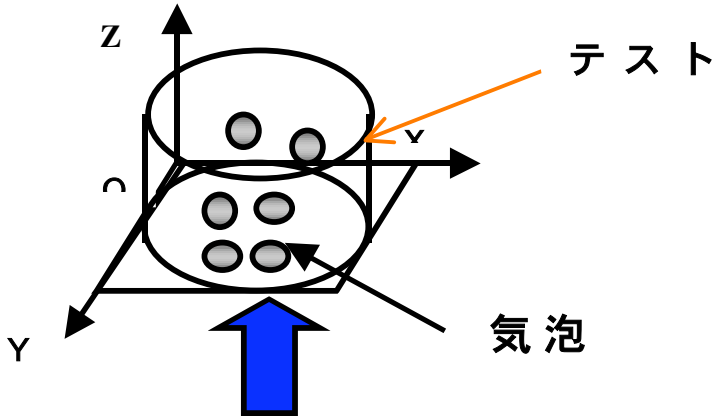
連続画像中の気泡の空間的重心位置を求め，細管内気泡の空間運動追跡を行う．

2.2 実験装置と条件

実験装置本体を図 2.3 に示す，テストチャネルは，内径９ｍｍのアクリル管からなる．気泡吹込み部から軸方向長さ L/D ＝６１，１０７，１５２の３箇所に気泡撮影部がある．この気泡撮影部分は屈折を避けるために４０ｘ４０の長方形断面になっている．無ひずみの表面反射鏡，ストロボおよびCCDビデオカメラの図 2.1 のような配置により，２方向からの画像が透過光で撮影される．また測定部全体はほかの光が入らないように黒布で覆われている．撮影画像範囲は８０ｘ８０ｍｍ，管中心よりカメラレンズまでの直線距離は８８０ｍｍ，撮影距離は１１５０ｍｍである．ビデオカメラのシャッタ速度は 1/1000 s 開放としてある．ストロボの発光時間と発光間隔はそれぞれ ,1.2us と0.5usとした．バックグラウンドの明るさのむらをなくすためにストロボの前面に白紙をはってある．また画像のノイズを低下させるためにCCDビデオカメラからの信号はカラー信号を除去し，輝度信号のみをビデオテープに記録する．気泡吹込み部の気泡発生は，透過粒子径 1.5um の黄銅焼結管４本による気泡発生と八つの直径１ｍｍの気泡発生孔を管周囲に設けたアクリル管１本による気泡発生の二つ方法を用いた．表１に気液流量条件を示す．

表１　　気液流量条件

JL(m/s)	0.4	0.8	1.2	1.6
Jc(m/s)	0.017	0.034	0.068	


Z
テスト
X
O
Y
気泡

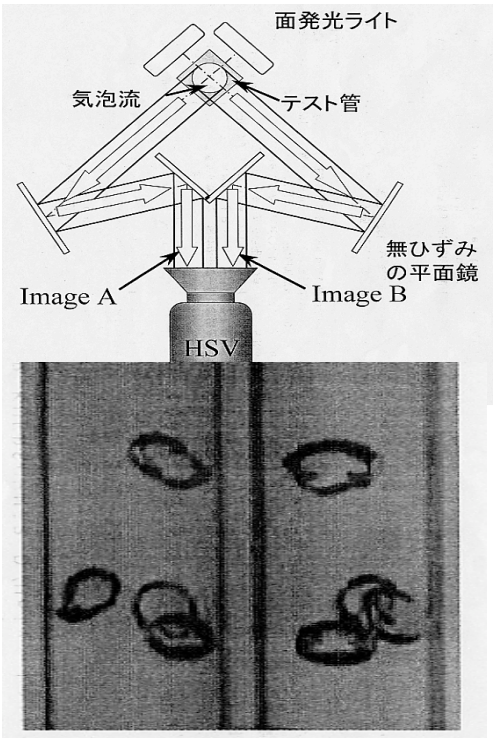
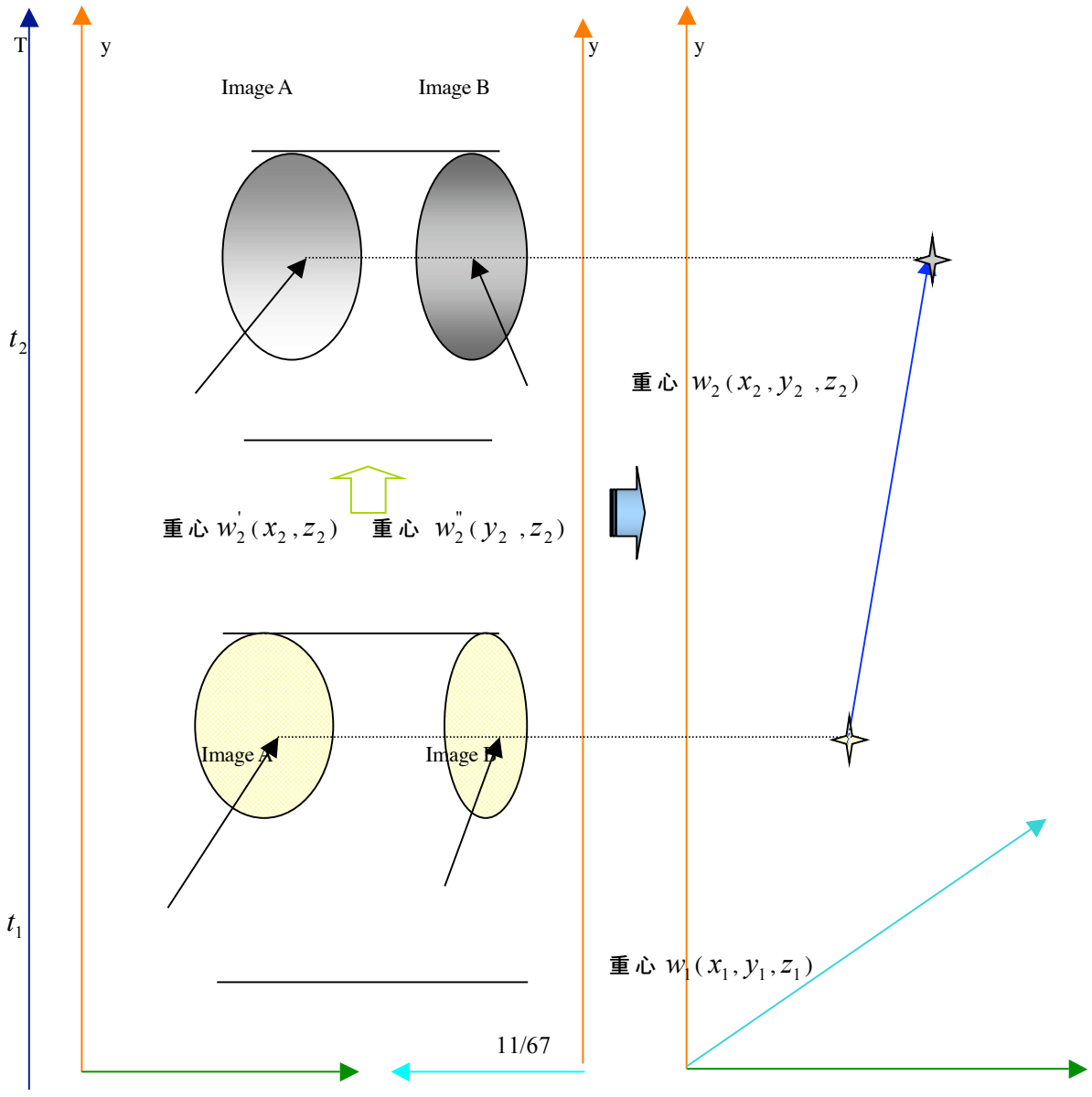

面発光ライト
気泡流
テスト管
無ひずみ
の平面鏡
Image A
Image B
HSV

図 2.1 実験装置と撮った気泡画像



y

重心 $w_1^{'}$ （ x_1, z_1 ） 重心 $w_1^{''}$ （ y_1, z_1 ）

o x y o o x

空間座標系

図 2.2 気泡の空間重心の推定

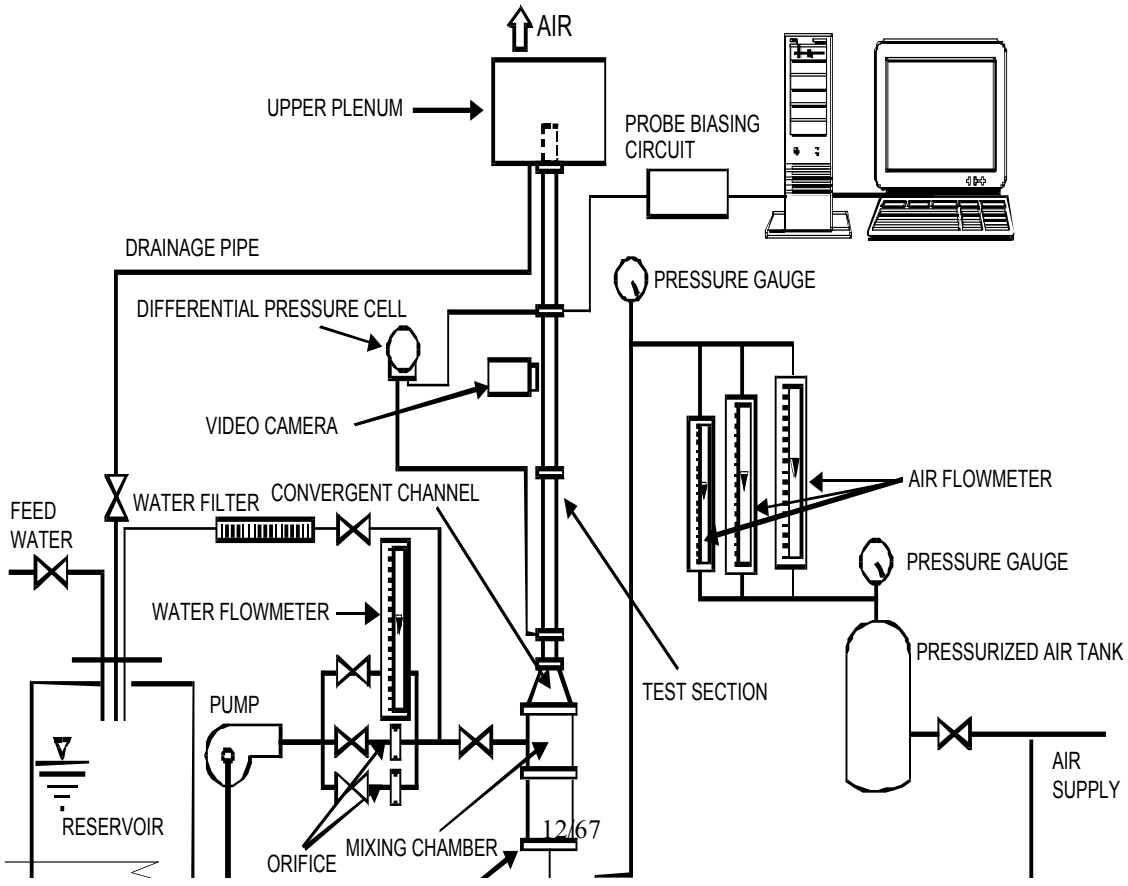


図 2.3 実験設備

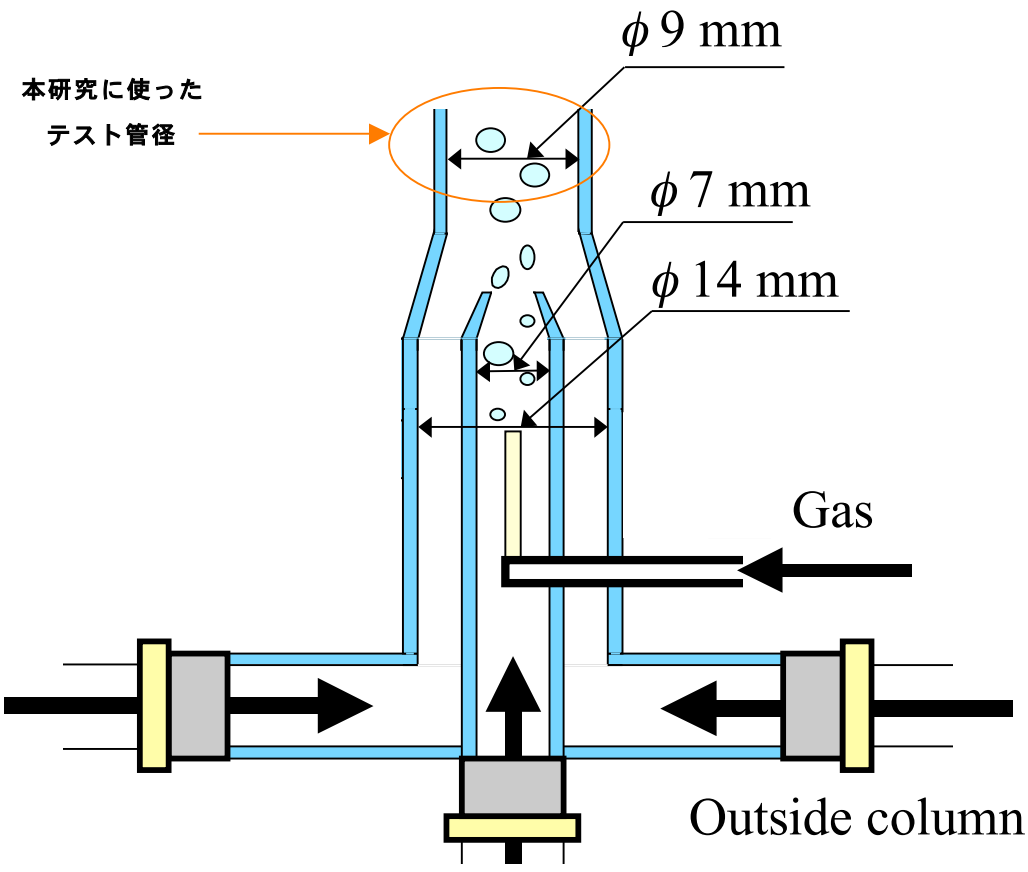


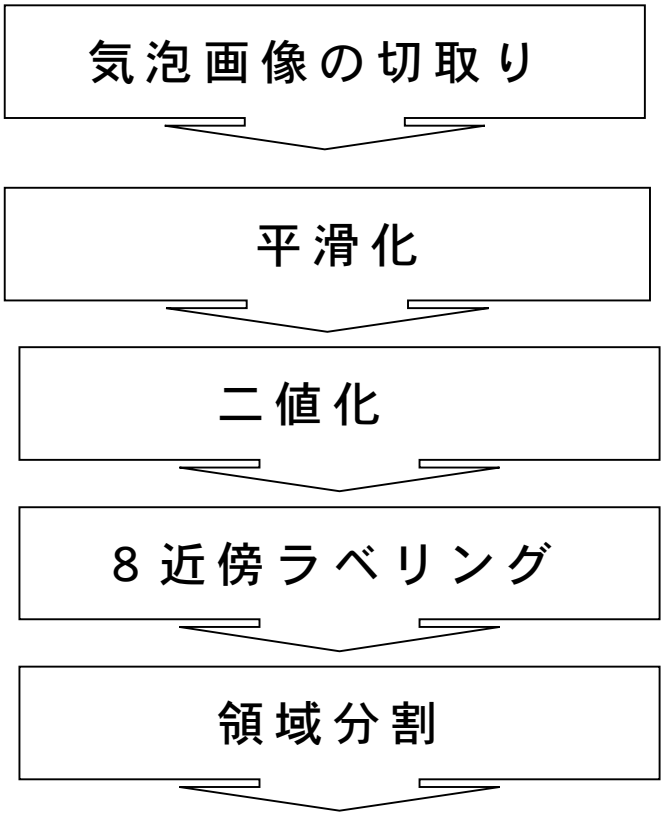
図 2.4 気泡発生装置

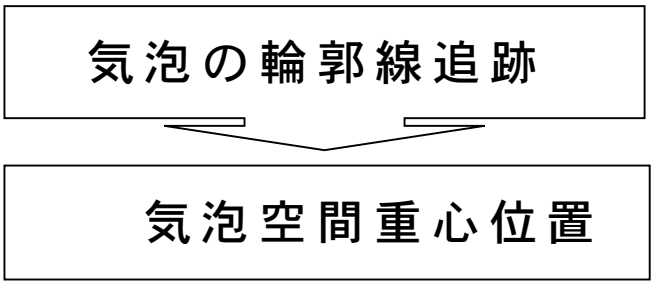

Scene7Trigger [START]
Time
240 98/11/29 02:07:46
+002:2710
0 PLAY 10.0 1000 1/6000
ID Status Play Rec Shutter

図 2.5 ビデオ気泡画像

第 3 章　画 像 処 理

3.1 処理の流れ





3.2 画像の平滑化

画像入力によって得られる画像には一般にいろいろな雑音が含まれている．それをできるだけ低減したいとき，あるいは，画素単位に細かく変動する濃淡を滑らかに変化するようにしたいとき，更に，濃淡変化を非常に少なくして，画像をぼかしたいときに行われる処理を平滑化という．

平滑化の方法は，すべての濃淡変動を一様に平滑化するものと，エッジや線など大きな濃淡変化を保存しながら微小な変動のみを選択的に平滑化しようとするものに大別できる．

移動平均フィルタリング

注目画素を中心とする画像の局所領域の平均値をその注目画素に出力するフィルタを，対象画像のすべての画素に適用する方法である．処理対象画像を $f(i,j)$, 処理対象画像を $g(i,j)$ とし，局所領域として中心画素から i 方向に $\pm M$, j 方向に $\pm N$ の矩形領域を考えると，$g(i,j)$ は次式で与えられる．

$$g(i,j) = \frac{1}{(2M+1)(2N+1)} \sum_{k=-M}^{M} \sum_{l=-N}^{N} f(i+k, j+l) \qquad (3.2.1)$$

実際の処理では，式（3.2.2）に示すような行列を重み係数行列 h とよぶことにすると，式（3.2.1）は h を使用して f と h の積和演算に置き換えることができる（式3.2.3））．

$$h = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \tag{3.2.2}$$

$$g(i,j) = \sum_{k=-M}^{M} \sum_{l=-N}^{N} f(j+k, j+l) h(k,l) \tag{3.2.3}$$

ただし ,$M = 1, N = 1$ であり ,h の要素$h(k,l)$は次のように参照されるものとする .

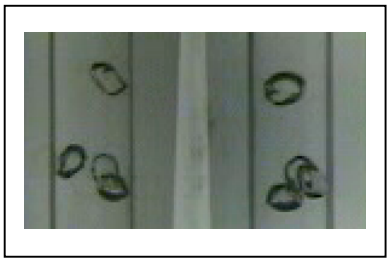
$$h(k,l) : \begin{bmatrix} h(-1,-1) & h(-1,0) & h(-1,1) \\ h(0,-1) & h(0,0) & h(0,1) \\ h(1,-1) & h(1,0) & h(1,1) \end{bmatrix} \tag{3.2.4}$$

式（3.2.3）を見ると，結果としてhはf に 対して平均操作を行うオペレータと見なせるし，f の局所平均値を出力するフィルタと見なすこともできる．このため，h は移動平均オペレータあるいは移動平均フィルタとよばれる．

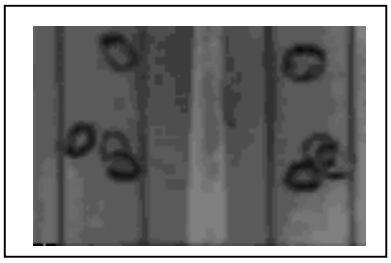
原画像画素配列の濃度 $f_{i,j}$（ 添字i, j はそれぞれ画面垂直及び水平方向に対応する）を次のエッジ保存フィルタを用いて平滑化した画素濃度$g_{i,j}$を求める，処理を行う．

$$g_{i,j} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} f_{i-1,j-1} & f_{i-1,j} & f_{i-1,j+1} \\ f_{i,j-1} & f_{i,j} & f_{i,j+1} \\ f_{i+1,j-1} & f_{i+1,j} & f_{i+1,j+1} \end{pmatrix} / 10 \tag{3.2.5}$$

その前処理としてヒストグラムの部分拡張による画像強調が必要な場合があるが，今回の撮影ではほとんどの場合読込み画像をそのまま平滑化した．図＜ａ＞に示す画像に対して，この３ｘ３画素の移動平均フィルタリングを行った結果を同図＜ｂ＞に示す．



図＜ａ＞



図＜ｂ＞

3.3 二値化

抽出したい対象物や特徴が，背景や他の特徴と明らかに異なる濃度をもっとき，画像から目的物を抽出する最も簡単な方法は閾値処理である．閾値処理は階調変換の一つであ

るが，閾値処理は見やすくするための変換ではなく対象の抽出が目的なので，変換曲線はステップ状になる．閾値処理のうちで，対象と背景の二つの領域に分離する手法を２値化処理と呼ぶ．２値化処理では通常，対象に１を背景には０を与え，１の画素の集合は図形と呼ばれる．

対象物と背景の濃度があらかじめわかっている場合には固定の閾値を設定できるが，撮像条件の変化や前処理によって除去しきれなかったシェージングの影響等を考えると，画像自身から自動的に閾値をきめるほうが望ましく，いくつかの手法がある．

3.3.1 ２値化処理方法

「１」 **P- タイル法** 対象が画像に占める割合，つまり面積比がおおよそわかっている場合には，画像の濃度ヒストグラムが利用できる．割合をＰとすると，対象が背景に比べて大きい値をもっている場合には，大きい濃度値のほうから分布を加算していって，ちょうどＰになる濃度を閾値とすればよい．この手法をＰ－タイル法（Ｐ－ tile method ）と呼ぶ，文書画像や線密度が予期できる論理回路図面等へ適用できる．

「２」 **モード法** Ｐ - タイル法では対象の占有率Ｐが既知である必要があるが，多くの場合不明である．ところで，閾値処理では対象物と背景の濃度差が大きい画像を扱っているので，濃度ヒストグラムは図 3.3.1.1 に示すように，対象物と背景のそれぞれの平均的な明るさでピークをもつ．そこで，この二つのピークの間の谷の底にあたる濃度を閾値とすれば対象物を抽出することができる．この手法をモード法（mode method）と呼ぶ．モードとは峰状の曲線を指し，二つ峰をもつとき双峰的（bimodal），富士山のような形を単峰（ unimodal ）と呼ぶ．

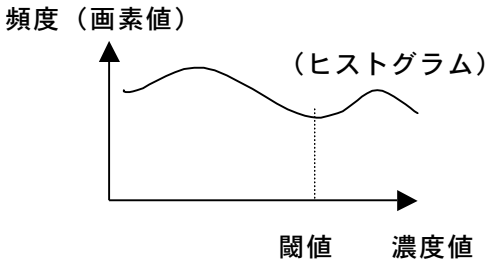


図 3.3.1.1 モード法による閾値選択

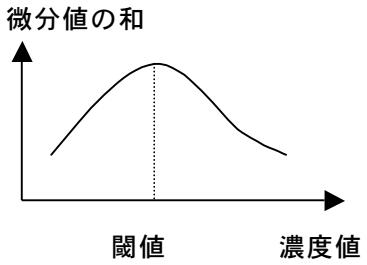


図 3.3.1.2 微分値ヒストグラム

「３」 **微分ヒストグラム法** 画像中の対象物と背景の境界は，濃度値が急に変化する部分（エッジ）に位置すると考えられる．画像の濃度値を直接利用するのではなく，微分値（濃度の変化率）を利用して閾値を決める方法が提案されている．これは次の手順で行われる．画像なかのある画素が濃度値Ｓをもつとする．この画素において微分値（たとえば，近傍の各画素とその画素の濃度値との差の最大値，あるいは各々の差の絶対値の総和等）を求める．与えられた画像中で濃度値Ｓをもつすべての画素における微分値の和を求める．これをすべてのＳに対して求めて微分ヒストグラムを得る（図 3.3.1.2 ）．この結果のヒストグラムの最も高い値をとる濃度値を選択すれば，濃度の変化率が高い部分に対応すると考えられる．

この方法は，図形の境界が一定の濃度値の範囲に納まっているときに有効である．た

だし，実際には境界付近の濃度値が複雑に変化する対象も多く，この方法がうまく働かないこともある．

「4」 **判別分析法** 画像の濃度値のヒストグラムにおいて，濃度値の集合を閾値ｔで2 つのクラス（ｔ以上とｔ未満）に分割したと仮定したとき，2つのクラス間の分離が最もよくなるようにパラメータｔを決めるという考え方に基づいた閾値選択法が提案されている．実際には，2つのクラスの平均値の分散（クラス間分散）と各クラスの分散（クラス内部分散）の比を最大にするという基準によりｔを決める．ｔの決める方を以下に示す．

与えられた画像が１，２,.....,Lの全部でＬレベルの濃度値をもつとする．ここで：閾値をｋとして，ｋ以上の濃度値をもつ画素と，それより小さな値をもつ画素の２つのグループに分け，クラス１，クラス２とする．クラス１の画素を $\omega_1(k)$，平均濃度値を $M_1(k)$，分散を $\sigma_{1(k)}$ と おき，クラス２の画素数を $\omega_2(k)$，平均濃度値を $M_2(k)$),分散を $\sigma_2(k)$ とおくと，クラス内分散は

$$\sigma w^2 = \omega_1\sigma_1^2 + \omega_2\sigma_2^2 \qquad (3.3.1)$$

クラス内分散は

$$\sigma_B^2 = \omega_1(M_1 - M_T)^2 + \omega_2(M_2 - M_T)^2 = \omega_1\omega_2(M \ - M_2)^2 \qquad (3.3.2)$$

で与えられる．ここで σ_B^2/ω_W^2 を最大にするには σ_B^2 を最大にすればよい．すなわち，k を変化させて σ_B を最大にする k の値を求めればよい．

この方法は，ヒストグラムが２つ山をもつ（谷が存在する）ときモード法として働き，山がないときにも閾値が求まるという便利さから，よく用いられている．反面，画像の幾何学的構造は反映されないので，必ずしも判別基準が人間の視覚と一致しないこともある．

[5] 可変閾値法 画像中の場所により平均的な濃度が変化している場合，単一の閾値で全画面をうまく２値化することはできない．この場合は，閾値を部分領域（小領域）ごとに変化させて全画面を処理する．

これは別名，動的閾値処理とも呼ばれる．入力装置のシェ－デイング等により画像が濃度値のゆるやかな勾配をもつときに適用される．

画像の二値化を３ｘ３近傍の平均濃度を閾値として同時に行う．

$$t_{ij} = \begin{pmatrix} f_{i-1,j-1} & f_{i-1,j} & f_{i-1,j+1} \\ f_{i,j-1} & f_{i,j} & f_{i,j+1} \\ f_{i+1,j-1} & f_{i+1,j} & i+1,j+1 \end{pmatrix} / 9 \qquad (3.3.3)$$

$$g_{i,j} = \begin{cases} 1; & f_{i,j} \geq t_{i,j} \\ 0; & f_{i,j} \mathrm{p} t_{i,j} \end{cases} \qquad (3.3.4)$$

対象気泡の 2 値化処理結果を図3.6.4 に示している。

3.4 気泡の同定

二つの気泡画像の気泡の同定には，次の条件のいずれかを用いる．

同定条件 (1)　　気泡が重なっておらず，気泡下端位置と上端位置が AB 画像で同じ気泡．

同定条件 (2)　AB 画像の一方の画像で 2 個の気泡が重なっており，その気泡の下端もしくは上端位置のどちらかが同じ二つの気泡がもう一方の画像にある [5].

3.5 輪郭線追跡

気泡の濃淡画像を二値化によって，対象気泡と背景に分離した画像では，対象気泡の背景との境界線は 1 の領域の縁を順にたどれば抽出できる．この処理を境界追跡と呼ぶ．境界追跡は追跡方向に対して，左手に領域を見るように．つまり反時針方向にたどられることが多い．背景と境界の追跡が反時針方向のときには，領域を左手にみているので孔との境界の追跡は時針方向になる.8-連結の境界線を反時針方向に追跡する手続きを以下に述べる．

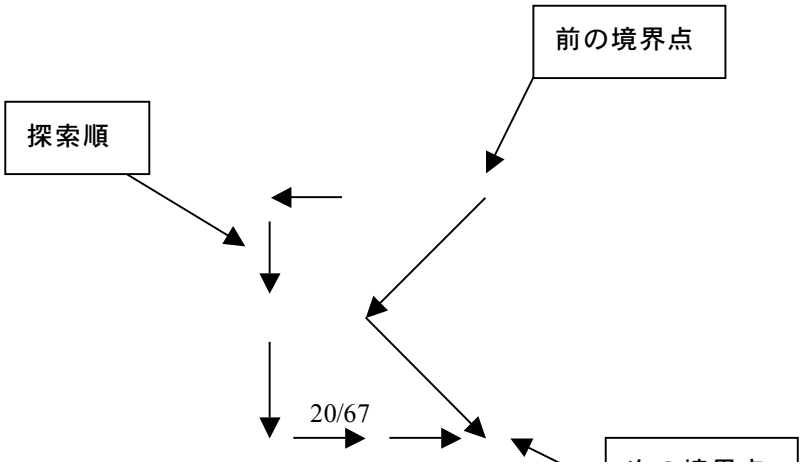
（ステップ 1）画像上を TV ラスタ走査し，まだ追跡ずみのマークのついていない境界点

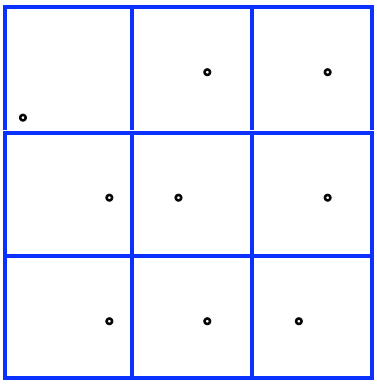
a_0 を 探し．これが見つかったら，1 本の境界線の追跡を開始する上記の境界点が存在しないとき，アルゴリズムを終了する．

（ステップ 2) a_0 の 8- 近傍において，左に隣接する 0 - 画素から始めて反時計まわりに順に，画素の値を調べる（図3.5.1）．このとき，

（1）　a_0 の 8- 近傍がすべて 0 - 画素のときは a_0 は 孤立点であるとして，1 本の境界線の追跡を終える。このとき，ステップ 1 へ戻って画像上のラスト走査を続け，次の境界線の処理へ進む．

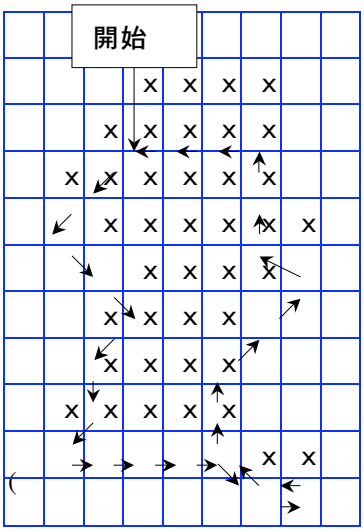
（2）　それ以外のとき，最初に当たった 1- 画素（a_1 と する）を次の境界画素として追跡を進める．





（斜線： 1 －画）

図3.5.1 境界線追跡における次の画素の見つけ方



（ x 印は 1 －画素 ）

図3.5.2　境界線追跡の例

（ステップ３）　a_1 の 8-近傍において，a_0 の次の画素から反時計まわりに1-画素を探し，これを a_2 と おく．以下同様にして，α_3，α_4 ..... を求めていく．ここであるｍにおいて，$\alpha_{m+1}=\alpha_1, \alpha_m=\alpha_0$ となったら 1 本の境界線追跡を終了する（図3.5.2）．このとき，a_0，a_1，…，α_{m-1} が一本の境界線をなす．なお，追跡の途中に各α_i に 追跡ずみのマークをつけておく．

再び，ステップ１へ戻って次の境界線の処理に進む．

このアルゴリズムで追跡される境界線は，各連結成分の外側の境界線と孔の境界線があり，外側の場合は反時計まわりに，内側の場合は時計まわりに追跡される．したがって，境界線の本数は（連結成分の個数＋孔の個数）にと等しい（図3.5.3）

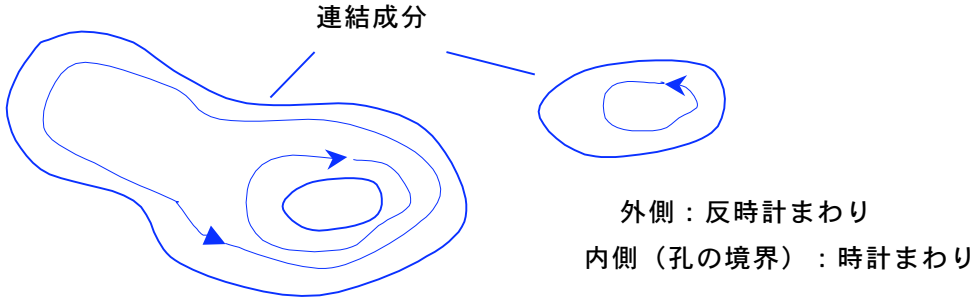


図3.5.3　境界線追跡の順序

抽出された点列の表現は点の座標 (i,j) 系列としてもよいが，ｎ点の系列には２ｎのデータ量が必要である．ここで点列を追跡した手続きをみると，p_{i+1} は p_i からの方向コードで表現できる．方向コード列はフリーマン（ Freeman ）のチェインコード（chain code）と呼ばれ，開始点の座標 (i,j) と n-1 点の方向コードの総計 n+1 のデータ量で座標列と同等の情報をもっている．

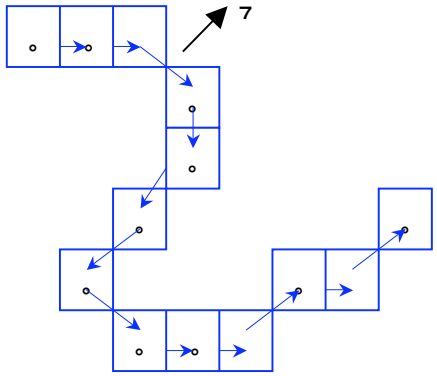
線図形を解析する場合，上記のように単純に画素列を座標値で記憶する以外に，線分の方向に符号を与えるチェイン・コード用いることが多い．

これは，画素列を $a_0, a_1, a_2, ..., a_n$ とすると，a_0 と，a_0 を図の中心位置におき，次の a_1 への方向を図3.5.4に従って符号化する．更に，a_1 を図3.5.4の中心位置におき，a_2 の方向を符号化する．これを繰り返していくことにより，最初の a_0 の 位置とｎ個の符号列（各々０～７のいずれかの値をとる）により，この画素列全体が表される．

この符号は，単位長さの線分の向いている方向を表しているので，全体としての線の方向性が直接表現される．また，隣りあった符号の変化を調べることにより，線の曲がり方を調べることもできる．線図形を座標列｛（x_i, y_j）　，ｊ＝１,……,n｝で直接記憶しておくよりも，このチェイン・コードを用いて記述したデータ量が少なくてすむので，データ圧縮の目的に用いられることもある．

3	2	1
4	i,j	0
5	6	7

図3.5.4　近傍位置の符号化



チェイン・コード　007655700101

図3.5.5 チェイン・コードの例

対象気泡の輪郭線追跡の結果を示す．

4.6 実験用画像処理結果

対象気泡の連続画像は１５６ｘ１０６画素，６４階調，間隔10msでパソコンに取り込まれる（図4.6.1）．平滑化，二値化，領域分割，輪郭線追跡の手順で，対象気泡の形状と周辺座標点が抽出される（図4.6.8），図中の→は追跡によって得た画素点において，明るさ勾配から求められる明るさ変化方向の直交方向を,8 方向量子化して示したものである（図 3.6.1)[5]. 式 (3.6.2) と (3.6.3) により平面重心点座標が求められる．ここで，s は気泡の範囲，Mは対象とする点の数である．なお図 2 における z については式 (3.6.2),(3.6.3) の重み付き平均値を用いた．

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y}\Big/\frac{\partial f}{\partial x}\right) \tag{3.6.1}$$

$$x = \frac{\sum_{xi \in S} xi}{M}, z = \frac{\sum_{zi \in S} zi}{M} \tag{3.6.2}$$

$$y = \frac{\sum_{yi \in S} yi}{M}, z = \frac{\sum_{zi \in S} zi}{M} \tag{3.6.3}$$

時刻t_1からt_{15}まで，１枚づつ対象気泡の空間重心座標が求められ，さらに式（ 2.1 ）（ 2.2 ），（ 2.3 ）により気泡の空間移動距離と空間移動速度求められる．

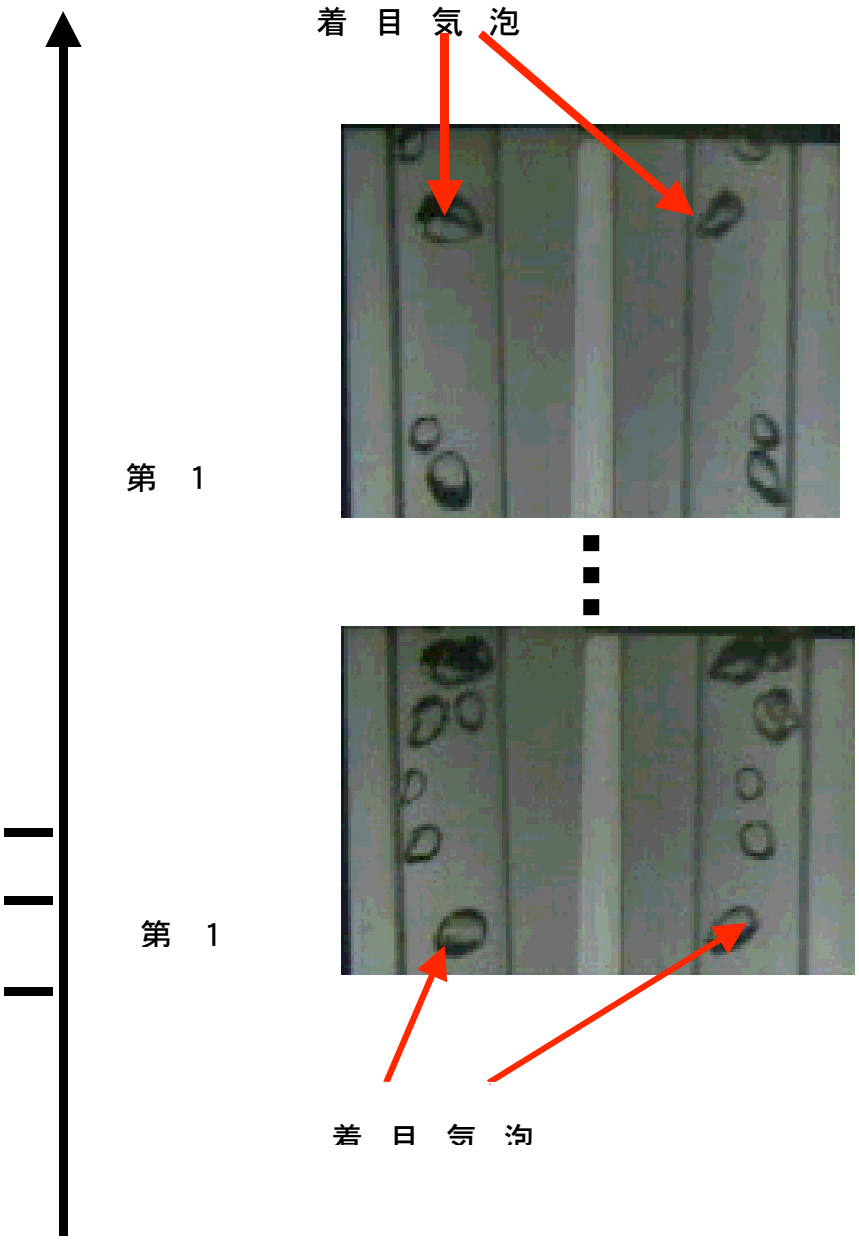


図3.6.1　連続１５枚の原画像

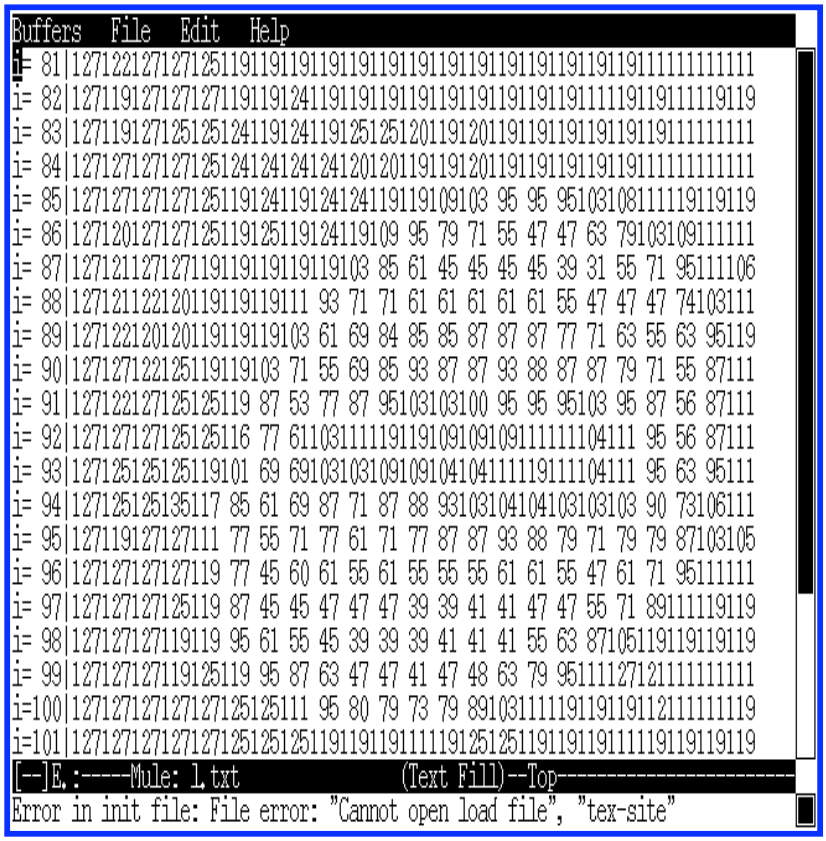

Buffers File Edit Help
i= 81|127122127127125119119119119119119119119119119119119119111111111111
i= 82|127119127127127119119241191191191191191191191191111191191111191119
i= 83|127119127125125124119124119125125120119120119119119119119119111111111
i= 84|127127127127125124124124124120120119119120119119119119119111111111111
i= 85|127127127127125119124119124124119119109103 95 95 95103108111119119119
i= 86|127120127127125119125119124119109 95 79 71 55 47 47 63 79103109111111
i= 87|127121127127119119119119119103 85 61 45 45 45 45 39 31 55 71 95111106
i= 88|127121122120119119119111 93 71 71 61 61 61 61 61 55 47 47 47 74103111
i= 89|127122120120119119119103 61 69 84 85 85 87 87 87 77 71 63 55 63 95119
i= 90|127127122125119119103 71 55 69 85 93 87 87 93 88 87 87 79 71 55 87111
i= 91|127122127125125119 87 53 77 87 95103103100 95 95 95103 95 87 56 87111
i= 92|127127127125125116 77 611031111191191091091091111111104111 95 56 87111
i= 93|127125125125119101 69 691031031091091041041111191111104111 95 63 95111
i= 94|127125125135117 85 61 69 87 71 87 88 93103104104103103103 90 73106111
i= 95|127119127127111 77 55 71 77 61 71 77 87 87 93 88 79 71 79 79 87103105
i= 96|127127127127119 77 45 60 61 55 61 55 55 55 61 61 55 47 61 71 95111111
i= 97|127127127125119 87 45 45 47 47 47 39 39 41 41 47 47 55 71 89111119119
i= 98|127127127119119 95 61 55 45 39 39 39 41 41 41 55 63 87105119119119119
i= 99|127127127119125119 95 87 63 47 47 41 47 48 63 79 95111127121111111111
i=100|127127127127127125125111 95 80 79 73 79 89103111119119119112111111119
i=101|127127127127127125125125119119119111119125125119119119111119119119119
[--]E.:-----Mule: 1.txt (Text Fill)--Top----------
Error in init file: File error: "Cannot open load file", "tex-site"

図3.6.2　原画像の明るさを調べる

3	2	1
4	i,j	0
5	6	7

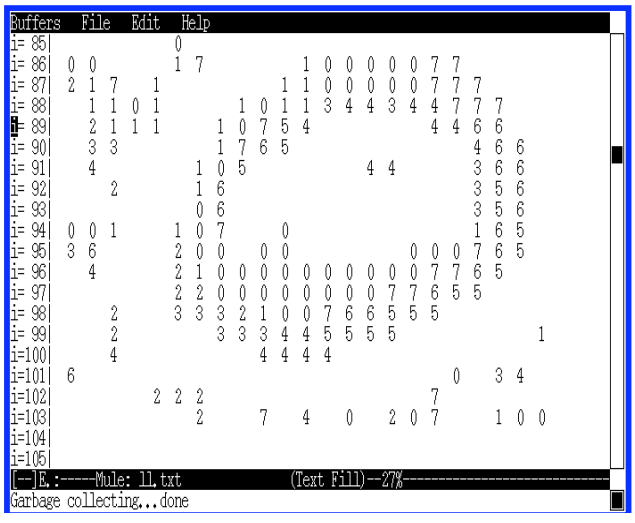


図3.6.3　　エッジの８方向コード

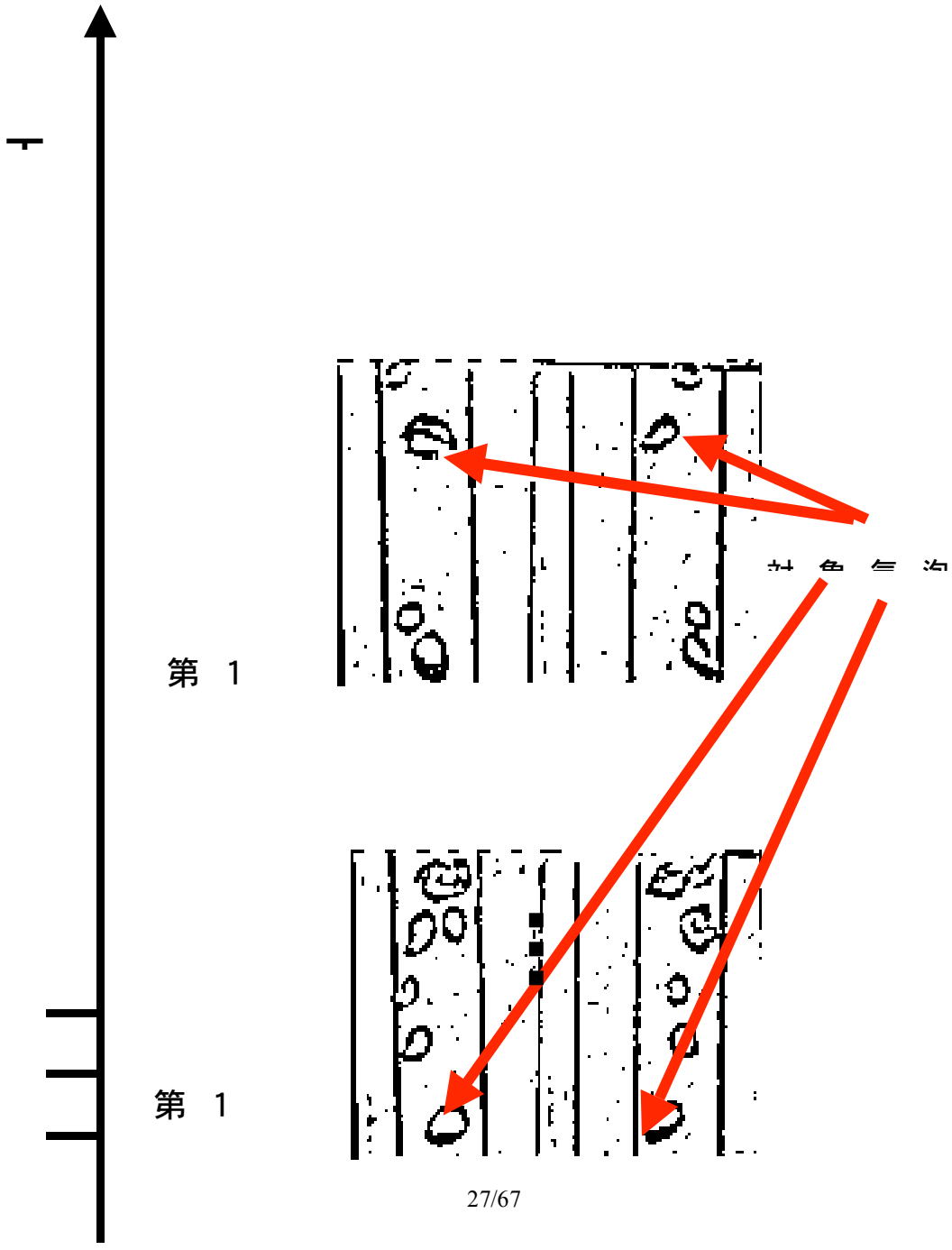
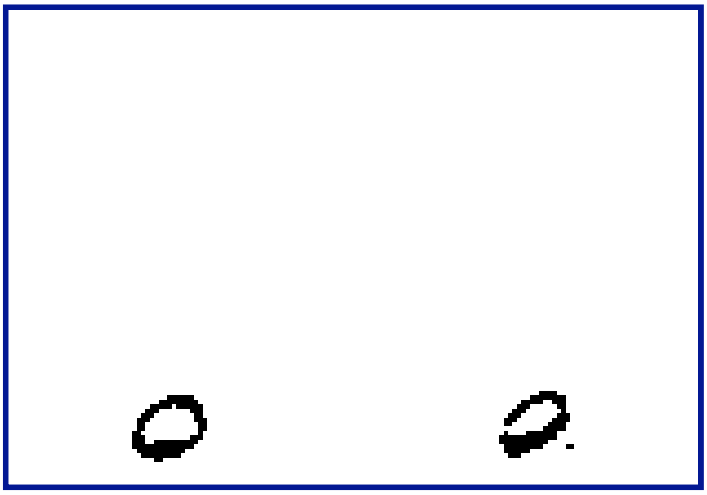


図3.6.4　　2値化した結果



```
i= 84|255255255255255255255255255255255255255255255255255255255255255255255
i= 85|255255255255255255255255255255255255255255255255255255255255255255255
i= 86|255255255255255255255255255255255255255  0  0  0  0  0  0255255255255
i= 87|255255255255255255255255255255255  0  0  0  0  0  0  0  0  0255255255
i= 88|255255255255255255255255255  0  0  0  0  0255  0  0  0  0  0  0255255
i= 89|255255255255255255255255  0  0  0255255255255255255255  0  0  0255255
i= 90|255255255255255255255  0  0  0255255255255255255255255255  0  0255255
i= 91|255255255255255255  0  0  0255255255255255255255255255255  0  0  0255
i= 92|255255255255255255  0  0255255255255255255255255255255255255  0  0255
i= 93|255255255255255255  0  0255255255255255255255255255255255255  0  0255
i= 94|255255255255255  0  0255255255255255255255255255255255255255  0255255
i= 95|255255255255255  0  0  0255255255255255255255255255255  0  0  0255255
i= 96|255255255255255  0  0  0255255  0  0  0  0  0  0  0  0  0  0255255255
i= 97|255255255255255  0  0  0  0  0  0  0  0  0  0  0  0  0  0255255255255
i= 98|255255255255255255  0  0  0  0  0  0  0  0  0  0  0255255255255255255
i= 99|255255255255255255255  0  0  0  0  0  0  0  0  0255255255255255255255
i=100|255255255255255255255255255255  0  0255255255255255255255255255255255
i=101|255255255255255255255255255255255255255255255255255255255255255255255
i=102|255255255255255255255255255255255255255255255255255255255255255255255
i=103|255255255255255255255255255255255255255255255255255255255255255255255
i=104|255255255255255255255255255255255255255255255255255255255255255255255
i=105|255255255255255255255255255255255255255255255255255255255255255255255
---------------------------------------------------------------------------
-- ｷﾑﾂｳ --(29%)
```

図3.6.5　　対象気泡の領域分割

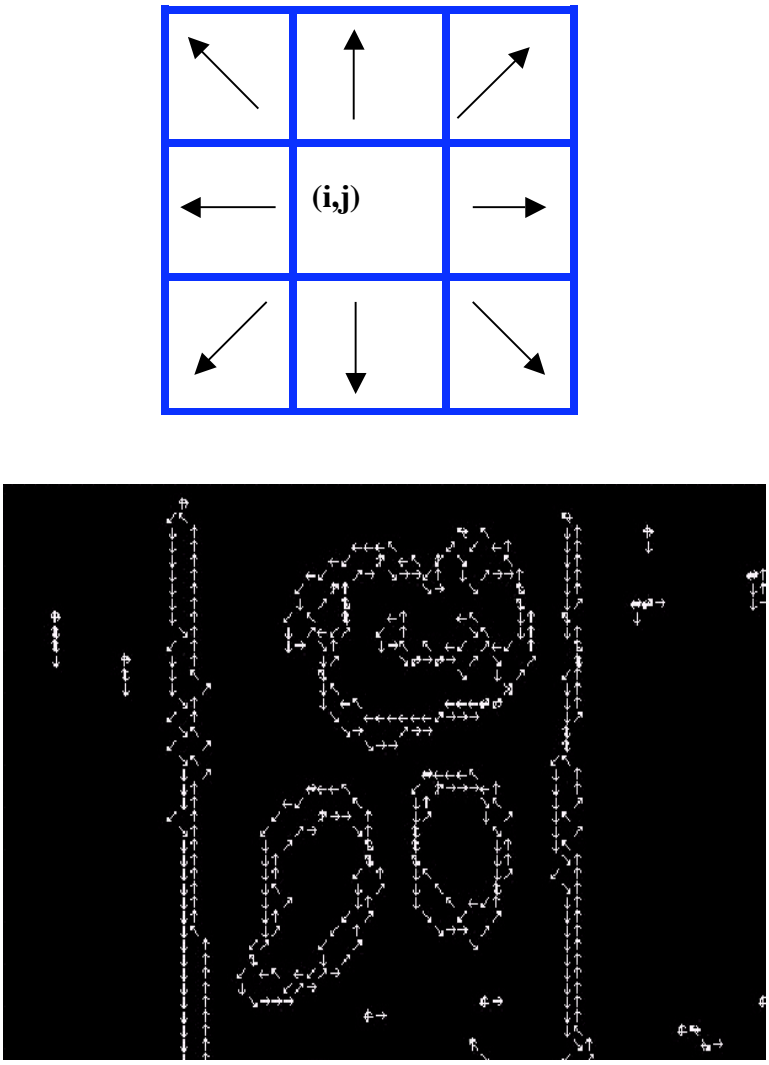
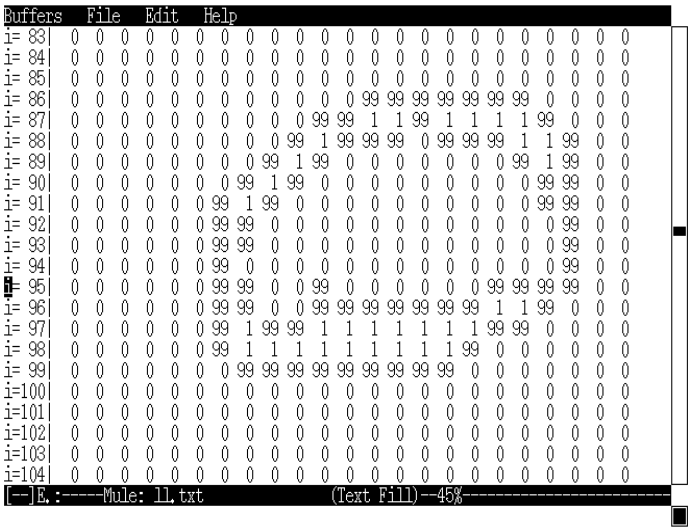


図3.6.6 部分気泡画像輪郭線追跡結果

図3.6.7　全画像の輪郭線追跡結果（→で表示）



Trace_data

20036 21035 22035 23035 24035 25035 26035 27035 28035 29035 30036 31037 31038 31039 32040 31041 30042 29042 28042 27043 26043 25043 24042 23043 22042 21042 21041 20040 20039 20038 20037 20036 0

21037 21038 21039 21040 22041 23041 24042 25042 26042 27042 28042 29041 29040 30039 29038 28037 27036 26035 25035 24035 23036 22036 21037 0

図3.5.8　対象気泡の輪郭線追跡結果

1 本線の追跡順序（外輪郭線）

Start point:(20,36)

i = 20, j = 36
i = 21, j = 35
i = 22, j = 35
i = 23, j = 35
i = 24, j = 35
i = 25, j = 35
i = 26, j = 35
i = 27, j = 35
i = 28, j = 35
i = 29, j = 35
i = 30, j = 36
i = 31, j = 37
i = 31, j = 38
i = 31, j = 39
i = 32, j = 40
i = 31, j = 41
i = 30, j = 42
i = 29, j = 42
i = 28, j = 42
i = 27, j = 43
i = 26, j = 43
i = 25, j = 43
i = 24, j = 42
i = 23, j = 43
i = 22, j = 42
i = 21, j = 42
i = 21, j = 41
i = 20, j = 40
i = 20, j = 39
i = 20, j = 38
i = 20, j = 37
i = 20, j = 36

one trace is over

内輪郭線追跡Start：

i = 21, j = 37
i = 21, j = 38
i = 21, j = 39
i = 21, j = 40
i = 22, j = 41
i = 23, j = 41
i = 24, j = 42
i = 25, j = 42
i = 26, j = 42
i = 27, j = 42
i = 28, j = 42
i = 29, j = 41
i = 29, j = 40
i = 30, j = 39
i = 29, j = 38
i = 28, j = 37
i = 27, j = 36
i = 26, j = 35
i = 25, j = 35
i = 24, j = 35
i = 23, j = 36
i = 22, j = 36
i = 21, j = 37

one trace is over

図3.6.9　気泡の輪郭線の点を出す

第 4 章　　解 析 結 果 と 検 討

4.1 結果

図 4.1 に示されるように気泡はほぼ等速で上昇している．また対象気泡が流れの比較的安定な管中心付近に位置しているために，水平方向への移動はほとんどない．図 5.1 でほぼ等速で上昇しているように見えた気泡も，細かく見るとかなりの速度変動が生じていることがわかる．図 4.3 の移動速度を見ると一層明らかとなる．図 4.4 は速度変動から求められる加速度を示している．周知のように液相から気泡にかかる力は加速度から求めることができるが，この図から，液乱流及び気泡の変形変化によりほぼ 10~20ms の周期で負の加速度が生じていることがわかる．

4.2 検討

1．手
作業の原因で画像の取り込む誤差をおこる．

2．2
つ枚の時間間隔が短くなる必要性がある．

3．観
察時間が長くなる必要性がある．

4．手
作業の原因で重心点座標誤差があるから，プログラムで自動的に出す必要性がある．

5．垂
直方向に補正する必要性がある．

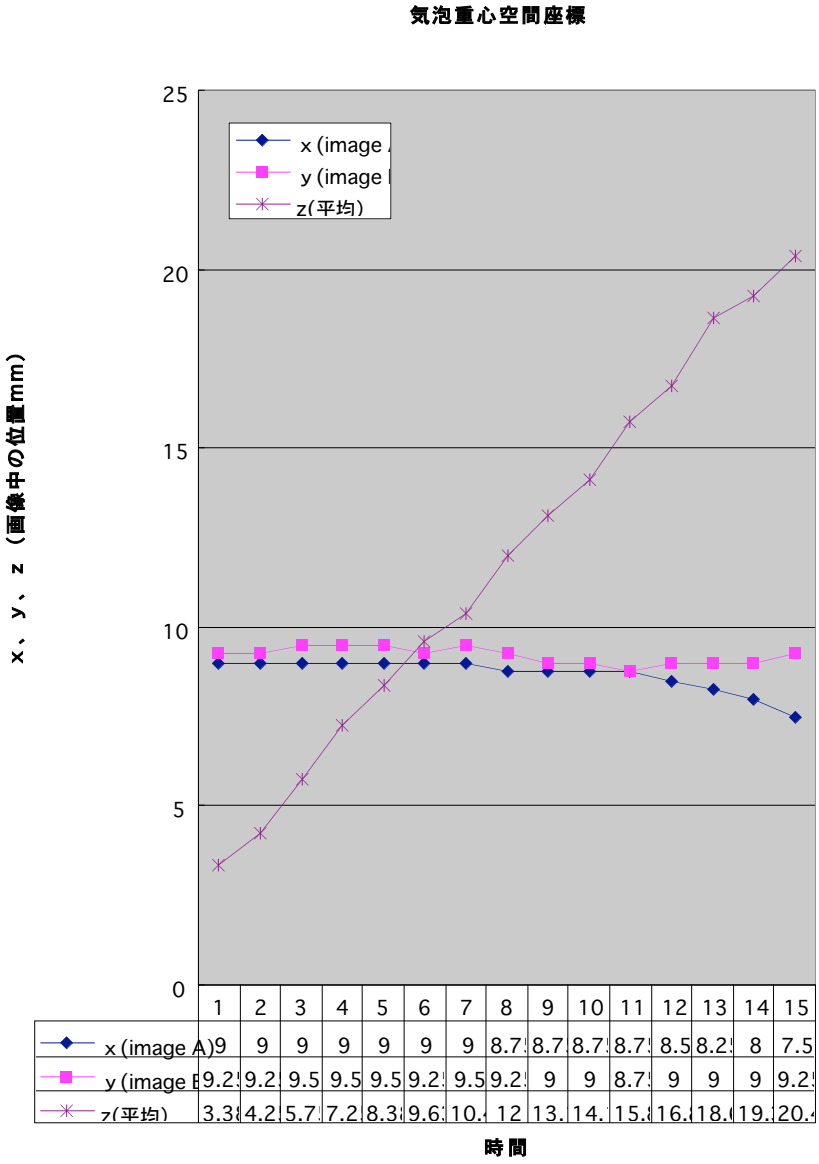


図 4.1　　気泡重心の空間座標

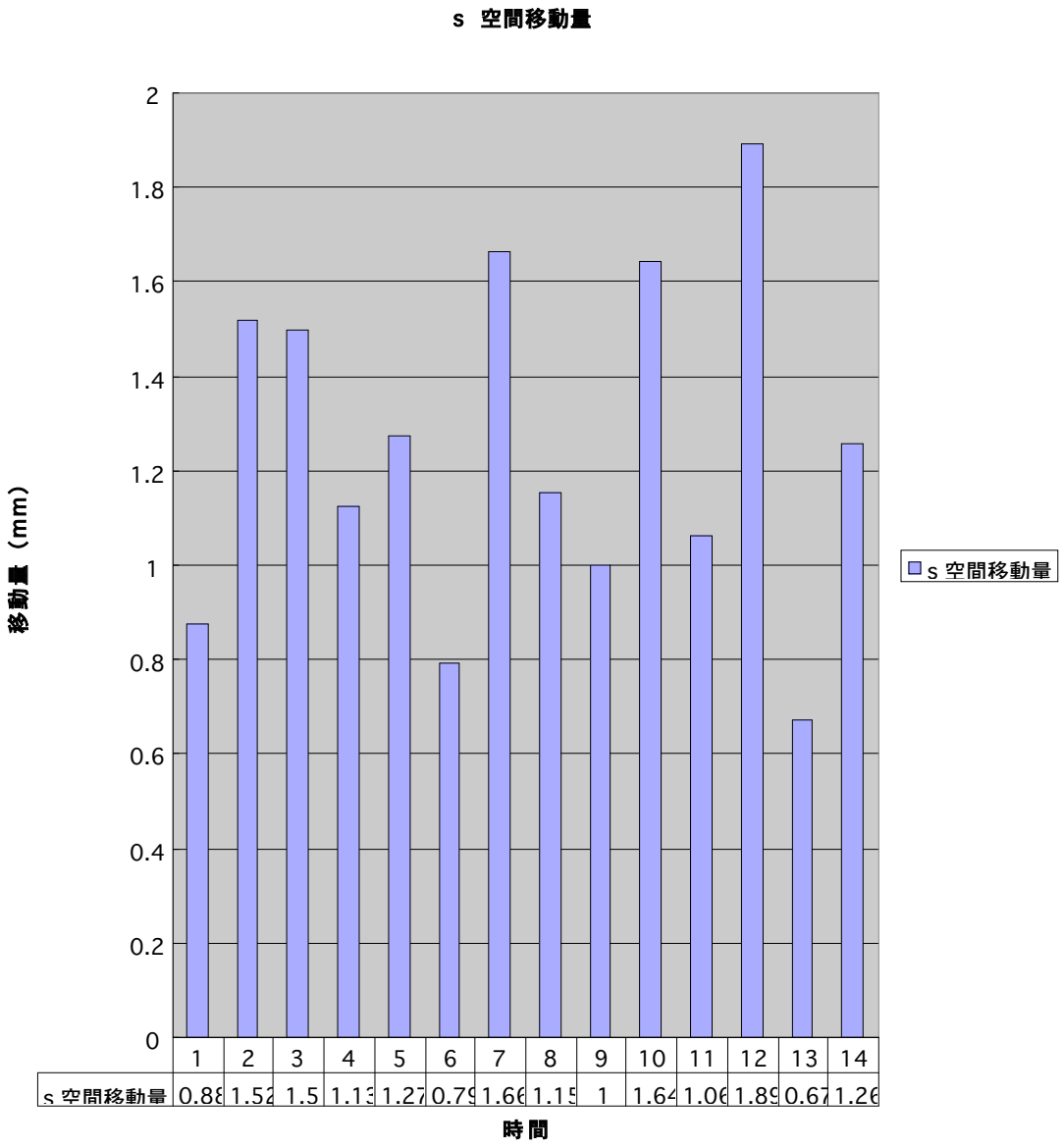

s 空間移動量
移動量（mm）
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14
s 空間移動量 0.88 1.52 1.5 1.13 1.27 0.79 1.66 1.15 1 1.64 1.06 1.89 0.67 1.26
時間
s 空間移動量

図 4.2 空間移動量

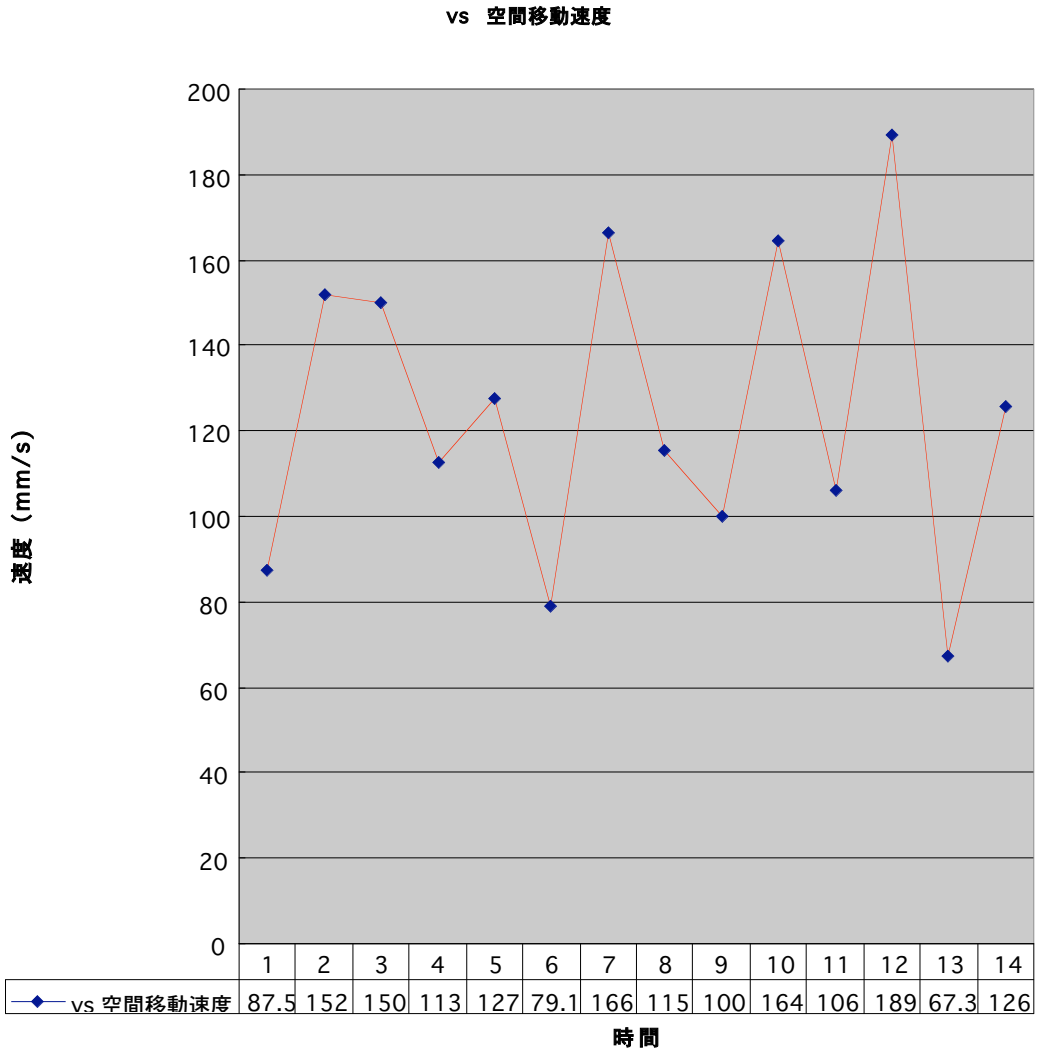


図 4.3　気泡重心空間移動速度

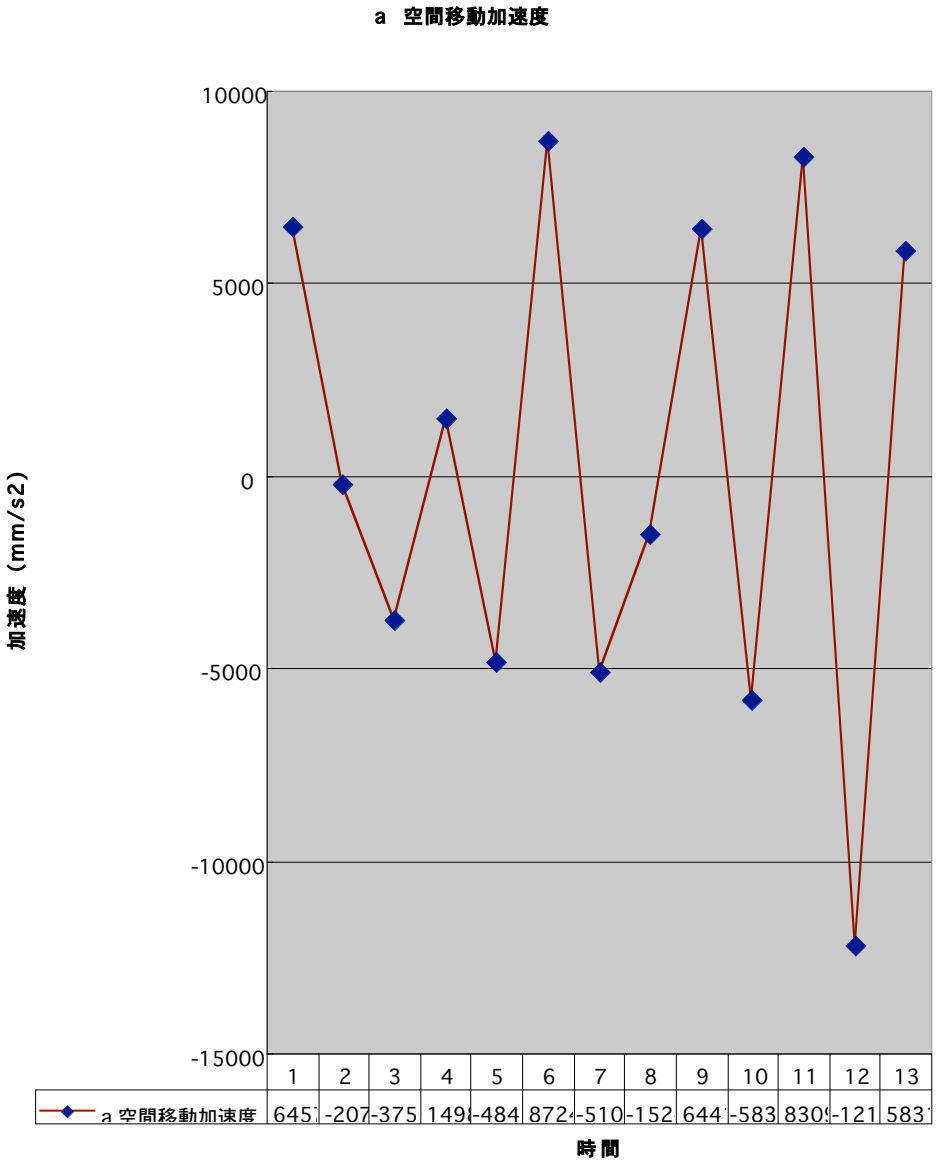


図 4.4　気泡重心空間移動加速度

第 5 章　終 わ り に

5.1 結論

本稿ではステレオ画像解析により垂直細管内単一気泡の 3 次元空間運動を追跡した .

その結果，この手法により気泡の 3 次元移動のミクロ計測が可能であることを示した．またほとんど等速で上昇しているように見える気泡も乱流及び気泡形状変化による浮力変動により，周期的な外力変動があり，その加速度変動は無視できないことが明らかにされた．

5.2 今後の課題

今後は長時間にわたる複数気泡の同時追跡や合体，衝突する 2 気泡を同時に追跡して，微妙な速度変化及び重なっている気泡などの複雑な状況の解明を検討したい．合体，衝突

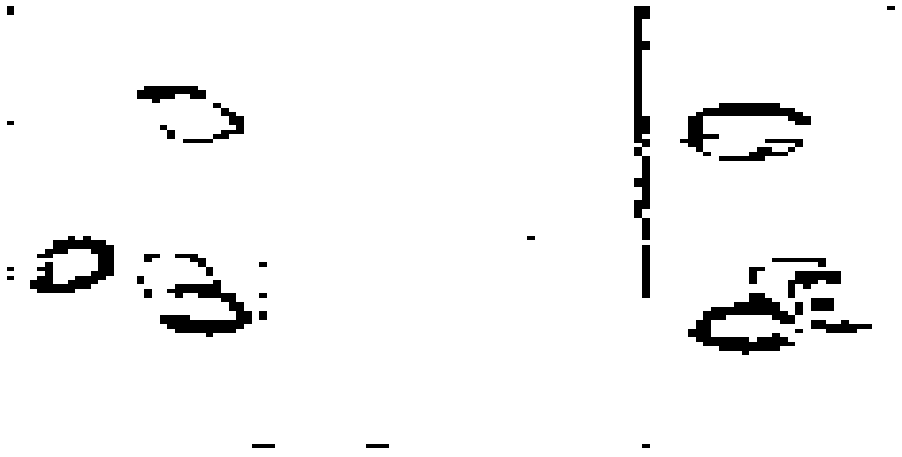


図6.2.1 重なってる気泡

現象の以下の場合を考えています．

（１）前後運動気泡衝突合体現象による気泡変形を観察する．

（２）平行運動の２つ気泡の合体現象を観察する．

謝 辞

本研究を遂行するにあたり，交通電子制御工学講座情報システム研究室の大島正毅教授には丁寧な御指導を頂きました．この場を借りて深く感謝致します．この修士論文の作成にあたり，動力システム工学講座の賞雅寛而教授に絶えず暖かいご指導をいただき，大変感謝しております．実験装置の作成から解析に至るまで多岐にわたってご助力をいただき

ました原子動力研究室波津久達也氏に感謝の意を表す．

非力ながらこの論文を修めることができましたのは，数々の人々の支えるがあったからこそであります．最後に，長谷川さん，本多さんと他情報システム研究室のみなさん，心から感謝しております．

参 考 文 献

[1] 大島正毅,http://carrot.isl.tosho-u.ac.jp

[2] 賞雅寛而，近藤宏一，機論,61-587,B(1995),18.

[3] 松本竹村，機論,58-547,B(1992),645.

[4] 賞雅寛而，後藤建博，木村文彦，日引俊，日本混相流学会年会講演論文,(2001),199.
[5] 賞雅寛而，三好直巳，機論,59-564,B(1993),63.
[6] 臼井，芝浦工大研究報告理工系,29-2(1985),11.
[7] Yamamoto,F., ほか３名,Proc,Int.Conf.Fluid Eng.'94,Kwanju,Korea,(1994),39.
[8] 賞雅寛而，近藤宏一，機論,61-587,B(1995),7.
[9] 賞雅寛而，近藤宏一，ほか３名，機論,63-606,B(1997),2.
[10] Serizawa,A, ほか３名,Proc,Jpn.-US Seminar TPF,Dyn.,(1988),E3.
[11] Lahey,R.T.,Proc.2ndInt.Conf.on Multiphase Flouw,3(1995),MO2-1.
[12] 賞雅．ほか３名，混相流シンポ講論集,9(1990),21.
[13] 大内・ほか２名，機論,57-534,B(1991),372.
[14]Michiyoshi,I.and Serizawa,A.,Nucl.Eng.Des.,95(1986),253.
[15] Serizawa,I.and Kataoka,I.,Int.J.Multiphase Flow,2(1975),235-246.
[16] Kataoka,I., ほか２名,Int.J.Multiphase Flow,12-4(1986),505.
[17]Drew,D.A. and Lahey,R.T., J. Fluid Mech.,117(1982),91-106.
[18]Lin,.Y.Mand Rezkallah,K.S.,Proc.2ndInt.Conf.on MultiphaseFlow '95-Kyoto,2(1995).6-21-6-26
[19]Kamp,A., ほか２名, Proc.2ndInt.Conf.on Experimental Heat Transfer,Fluid Mechanics and Thermodynamics,Hounolulu,USA,(1993),1418-1425.
[20]Kamp,a., ほか２名,Proc,2ndInt.Conf.on MultiphaseFlow '95-Kyoto,(1995),Vol.3,P6-13.
[21]Takamasa,T., ほか３名,J.VisualizationSociety ofJapan,vol.14-Suppl.No.1(1994), 187-190.
[22]Elghobashi,S.E.and Abou-Arab,T.W.,Phys.Fluids.26(1983),931-938.
[23] 須藤・ほか２名，機論,58-548,B(1992),33.
[24] 佐藤正也，青山晃治，大西昇，「耳と顔の特徴の位置関係に基づく顔画像からの視線方向の推定」，信学技報，PRMU97-280，pp.113-120，1998-03．
[25] 市川知弥，佐藤幸男，「距離画像と輝度画像を用いた顔の部位抽出」，第 6 回画像センシングシンポジウム講演論文集，pp.425-428，2000-6
[26] 大野健彦，視線を用いた高速なメニュー選択作業，情報処理学会論文誌，Vol.40 No.2，pp.602-612 （1999.2）
[27] 原島博（監修），元木紀雄，矢野澄男（共編），「3 次元画像と人間の科学」，pp.4-6，株式会社オーム社（2000.4）
[28] 苧阪良二，中溝幸夫，古賀一男，「眼球運動の実験心理学」pp.15,45,52,53，財団法人名古屋大学出版会（1993.5）
大山正，今井省吾，和気典二，「新編　感覚，知覚，心理学ハンドブック」，誠信書房
[29] 岡田　隆三，白井　良明，三浦　純，久野　義徳，“オプティカルフローと距離情報に基づく動物体追跡”，電子情報通信学会論文誌　D-Ⅱ　Vol.j80-D-Ⅱ　No.6 pp.1530-1538，1997年 6 月
[30] 岡田　隆三，白井　良明，三浦　純，久野　義徳，“オプティカルフローと距離情報の統合による 3 次元運動する人間の追跡”，電子情報通信学会論文誌　D-Ⅱ　Vol.j82-D-Ⅱ　No.8 pp.1252-1261，1999年 8 月

[31] 長井　敦，白井　良明，久野　義徳，“複雑変動背景下における移動物体の検出”，電子情報通信学会論文誌　D-Ⅱ　Vol.j80-D-Ⅱ　No.5 pp.1086-1095，1997年 5 月
[32] 大島　正毅，白井　良明，“3 次元情報を用いた物体認識”，電子情報通信学会論文誌　別刷　Vol.j65-D No.5 pp.1086-1095，1982年 5 月
[33] 長井　敦，白井　良明，久野　義徳，“複雑変動背景下における移動物体の検出”，電子情報通信学会論文誌　D-Ⅱ　Vol.j80-D-Ⅱ　No.5 pp.1086-1095，1997年 5 月
[34] 角　保志，冨田　文明，“ステレオビジョンによる 3 次元物体の認識”　，電子情報通信学会論文誌　D-Ⅱ　Vol.j80-D-Ⅱ　No.5 pp.1105-1112，1997年 5 月
[35] 依田　育士，坂上　勝彦，“3 次元動き情報を利用した複数対象物の抽出とその実時間認識”，電子情報通信学会論文誌　D-Ⅱ　Vol.j81-D-Ⅱ　No.9 pp.2043-2051，1998年 9 月
[36] 小野口　一則，武田　信之，渡辺　睦“ステレオ画像の平面投影による移動障害物位置検出”，電子情報通信学会論文誌　D-Ⅱ　Vol.j81-D-Ⅱ　No.8 pp.1895-1903，1998年 8 月
[37] 渡辺　斉，和田　俊和，松山　隆司“照明変化に対して頑健な背景差分法”，コンピュータビジョンとイメージメディア　115 - 3 pp.17-24，1999，3，18
[38] 池田　徹，大中　慎一，溝口　正典“画像の一様変化に高速追従する背景画像生成手法”，信学技報　PRMU97-7 pp.49-56，1997年 5 月
[39] 高橋　一哉，北村　忠明，小林　芳樹“画像処理による交通流監視方法の研究”，信学技報　PRMU97-6 pp.41-48，1997年 5 月
[40] 高藤　政雄，北村　忠明，小林　芳樹“空間微分および差分処理を用いた車両検出”，電子情報通信学会論文誌　D-Ⅱ　Vol.j80-D-Ⅱ　No.11 pp.2976-2986，1997年11月
[41] 林　健太郎，久野　義徳，島田　信敬，白井　義明“動的ロバストキャリブレーションによる人体の姿勢復元”，電子情報通信学会論文誌　D-Ⅱ　Vol.j83-D-Ⅱ　No.3 pp.977-987，2000年 3 月
[42] 星野　准一，増田　功，“局所的な輝度変動にロバストなパノラマ画像生成法”，電子情報通信学会論文誌　D-Ⅱ　Vol.j82-D-Ⅱ　No.2 pp.222-229，1999年 2 月
[43] 河井　良浩，植芝　俊夫，吉見　隆，大島　正毅“多視点レンジデータからの 3 次元形状復元”，電子情報通信学会論文誌　D-Ⅱ　Vol.j75-D-Ⅱ　No.4 pp.737-748，1992年 4 月
[44] 鈴木　充雄，大島　正毅“船上からの時系列画像を用いた航行船舶の検出実験”，日本航海学会第99回講演会，1998年11月
[45] 大崎　喜彦，　山本　正信，　“ステレオ画像からの 3 次元近似モデルのモデルフィッティング”，電子情報通信学会論文誌　D-Ⅱ　Vol.j81-D-Ⅱ　No.6pp.1259-1268,　1998年 6 月
[46] 木本　伊彦，　梶谷　昭彦，　安田　康彦“スティックモデルに基づく単眼視動画像からの人体歩行運動の解析の一手法“，電子情報通信学会論文誌D-Ⅱ　Vol.j74-D-Ⅱ　No.3 pp.1259-1268,　1991年 3 月
[47]　加藤　尚徳，　尺長　健，　“スティックモデルを用いたステレオ画像からの人物姿勢の連続的推定”，信学技報D-Ⅱ　PRMU98-255　No.3　pp.81-88,1999 年 3 月

付 録 (プ ロ グ ラ ム)

1. 輪郭線追跡

平滑化－２値化－領域処理－輪郭線追跡

```
#include <stdio.h>
 #include <stdlib.h>
#include <math.h>
 #include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <X11/Intrinsic.h>
#include <X11/StringDefs.h>
#include <X11/Shell.h>
 #define              IMAX       106
 #define              JMAX       156

 static               Window                    root,win1,win2;
static               Display                  *disp;
 static              Visual                    *vis;
 static              Colormap                  cmap;
 static              GC                        gc;
 static              XColor                    color[256];
static               XSetWindowAttributes      atr;
 static              FILE                      *fpin;
 static              XImage                    *image;

 typedef struct{
        int magic; int width; int heigth; int depth; int length;
         int type; int maptype; int maplength;
    }rasterfile;
 rasterfile          raster;
 char names1[] = { "input image" };              char *name_return1;
 char names2[] = { "output image" };             char *name_return2;
 u_char input_data[IMAX][JMAX];
 u_char output_data[IMAX][JMAX];
 u_char turn_data[IMAX][JMAX];
u_char turn1_data[IMAX][JMAX];
u_char turn2_data[IMAX][JMAX];

 void win_inp(int);
 void win_out(int,int);
  void get_image(char *[], u_char *, int);
 void put_image(u_char *, Window, int, int, u_char);
  void insatu(u_char output_data[][JMAX]);
```

```
main(int argc, char **argv)
{
    int                     i,j,m,k,n,in,jn,iw,jw,ds,nb,nc,na;
   int                      ib[8]={0,-1,-1,-1, 0, 1,1,1};
   int                      jb[8]={1, 1, 0,-1,-1,-1,0,1};
    unsigned        int     id[1000],jd[1000],ic;

   for(i=0;i < IMAX;i++){
       for( j = 0;j < JMAX;j++){
          input_data[i][j]=0;
           output_data[i][j]= 0;
           turn_data[i][j]=0;
          turn1_data[i][j]=0;
          turn2_data[i][j]=0;
       }
    }
    win_inp(argc);
     get_image( argv,( &input_data )[0][0],1);
   put_image(( &input_data ) [0][0], win1, JMAX, IMAX, 0 );
   XFlush(disp);
   for(i=0;i < IMAX;i++){
       for( j=0; j < JMAX ;j++){
         turn_data[i][j]=(input_data[i-1][j-1]+input_data[i-1][j]+
           input_data[i-1][j+1]+input_data[i][j-1]+
          input_data[i][j+1]+input_data[i+1][j-1]+
            input_data[i+1][j]+input_data[i+1][j+1]+
            2*(input_data[i][j]))/10;
       }
    }
   for(i=0;i < IMAX;i++){
       for( j=0; j < JMAX ;j++){
          k=(turn_data[i-1][j-1]+turn_data[i-1][j]+
  turn_data[i-1][j+1]+turn_data[i][j-1]+
 turn_data[i][j+1]+turn_data[i+1][j-1]+
turn_data[i+1][j]+turn_data[i+1][j+1]+
  turn_data[i][j])/9;
           if(turn_data[i][j] >=k ){
turn1_data[i][j]=0;
          }else{
turn1_data[i][j]=1;
```

```
            }
          }
      }
     for(i=0;i < IMAX;i++){
         for( j=0; j < JMAX ;j++){
            if(j >28&&j<44&&i >84&&i<100){
turn2_data[i][j]=turn1_data[i][j];
           }else if(j >110&&j<127&&i>84&&i<100){
turn2_data[i][j]=turn1_data[i][j];
           }else{
turn2_data[i][j]=0;
            }
          }
      }
      printf("turn2_data¥n");
    insatu(turn2_data);
       for(i=0;i < IMAX;i++){
       for(j=0; j < JMAX ;j++){
            if(turn2_data[i][j]==1){
nc=turn2_data[i][j+1]*turn2_data[i-1][j]*turn2_data[i][j-1]*turn2_data                    [i+1][j];
if(nc==0){
     if (turn2_data[i+1][j]==0) ds=6;
    else if(turn2_data[i][j-1]==0) ds=4;
    else if(turn2_data[i-1][j]==0) ds=2;
   else if(turn2_data[i][j+1]==0) ds=0;
     in=iw=i; jn=jw=j;
    n=(ds+4)%8;
    printf("start point is found, i=%3d,j=%3d,ds=%3d,n=%d¥n",i,j,ds,n);
    ds=-1;nb=-1;
    na=turn2_data[in-1][jn-1]+turn2_data[in-1][jn]+
       turn2_data[in-1][jn+1]+turn2_data[in][jn-1]+        turn2_data[in][jn+1]+turn2_data[in+1][jn-1]+
        turn2_data[in+1][jn]+turn2_data[in+1][jn+1];
    if(na >0){
       for(ic=0;ic<1000;ic++) id[ic]=0;jd[ic]=0;
       ic=0;
        do {
           id[ic]=in=iw;jd[ic]=jn=jw;
          turn2_data[in][jn]=99;
           printf(" point is found, iw=%3d,jw=%3d,ic=%3d,m=%d,n=%d¥n",iw,jw,
ic,m,n);
```

```
            ic++;
             n=(n+4)%8;m=0;
             do {
 n=(n+1)%8;if(m==8)    goto End;
iw=in+ib[n];
jw=jn+jb[n];
m++;
             }while(turn2_data[iw][jw]!=1&&turn2_data[iw][jw]!=99);
             if(ds<0){
 if(nb<0) nb=n;
else    ds=nb;
             }
     /*    }while((i!=in&&j!=jn)||(n!=ds));*/
        }while(i!=in||j!=jn||n!=ds);
     }
     else turn2_data[in][jn]=0;
End:;
 }
             }
          }
     }
      printf("turn2_data¥n");
    insatu(turn2_data);
     for(ic=0;ic<1000;ic++)
        output_data[id[ic]][jd[ic]]=31;
       win_out(JMAX,IMAX);
     put_image((&output_data)[0][0],win2,JMAX,IMAX,0);
     printf("output_data¥n");
      insatu(output_data);
    XFlush(disp);
     printf("finish");
     getchar();
  }

 void insatu(u_char output_data[][JMAX])
{
    int      jst,m,r,i,j;
    for(jst=0;jst<JMAX;jst=jst+23){
        r=jst/23+1;
        printf("******DATA OF IMAGE****          part *%2d* ¥n",r);
```

```
        for(m=1;m<=39;m++){            /* "--" 39 times */
          printf("--");
        }
       puts("");
        printf(" **j=|");
       for(j=jst;j<=jst+22;j++){
            if(j<=JMAX){
printf("%3d",j);
            }
        }
       puts("");
        for(m=1;m<=39;m++){
          printf("--");
        }                                          /* "---" 39 times */
       puts("");
        for( i = 0; i < IMAX;i++){
          printf("i=%3dl", i );
            for( j = jst; j <=jst+22;j++){
 if(j<=JMAX){
 printf("%3d",output_data[i][j]);
 }
          }
           puts("");
        }
        for(m=1;m<=39;m++){            /* "---" 39 times" */
          printf("--");
            }
       puts("");
        printf("----------PART* %d * FINISH¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n",r);
    /* printf("----PART* %2d *FINISH¥f",r);*/
    }
}

void get_image(char *argv[], u_char *vd, int no)
{
     int                i,j,pix;
    char                *dt_name,name[50],*dir="";
     u_char             dt[JMAX];
   dt_name=argv[no];
    strcat( strcpy( name, dir ), dt_name );
```

```
    fpin=fopen( name,"r" );
   if( fpin==NULL ){
       printf("Can't open %s.Abort!!¥n",argv[no]);
       fclose(fpin);      exit(1);
    }
   fread(&raster,sizeof(raster),1,fpin);
   pix=raster.maplength/3;
   for( i=0; i<pix; i++ )        color[i].red      = getc(fpin)<<8;
    for( i=0; i<pix; i++ )        color[i].green = getc(fpin)<<8;
   for( i=0; i<pix; i++ ){      color[i].blue    = getc(fpin)<<8;
                                          XAllocColor(disp,cmap,&color[i]);
    }
   for(i=0; i<IMAX; i++){
       fread( dt, sizeof(char), JMAX, fpin );
      for(j=0; j<JMAX; j++, vd++)          *vd =dt[j];
    }
    fclose(fpin);
}

void put_image(u_char *data, Window win, int width, int height, u_char type)
{
    int i;
   u_char *dt;
   dt=(u_char *)malloc(width*height);
     if(type==0)
       for(i=0;i<width*height;i++,data++)    *(dt+i)=(char)color[*data].pixel;
        else
          for(i=0;i<width*height;i++,data++)    *(dt+i)=*data;
      image=XCreateImage(disp,vis,8,ZPixmap,0,(char *)dt,width,height,8,width);
   XPutImage(disp,win,gc,image,0,0,0,0,width,height);
   free(dt);
    }

void win_inp(int argc)
{
   if( argc!=2 ){
        printf("Usage: [Program][data_name]¥n");
       exit(1);
    }
    disp = XOpenDisplay(NULL);
```

```
    root = RootWindow(disp, 0);
    vis    = DefaultVisual(disp,0);
    cmap = DefaultColormap(disp, 0);
   gc      = XCreateGC(disp,root, 0, 0);
     atr.backing_store = Always;
     /*    about win1    */
   win1 = XCreateSimpleWindow(disp,root,100,100,JMAX,IMAX,2,0,0);
   XChangeWindowAttributes(disp,win1,CWBackingStore,&atr);
     XStoreName(disp,win1,names1);XFetchName(disp,win1,&name_return1);
    XMapWindow(disp,win1);
}

void win_out(int x,int y)
{
     disp = XOpenDisplay(NULL);
    root = RootWindow(disp, 0);
    vis    = DefaultVisual(disp,0);
    cmap = DefaultColormap(disp, 0);
   gc      = XCreateGC(disp,root, 0, 0);
     atr.backing_store = Always;
         /*    about win2    */
   win2 = XCreateSimpleWindow(disp,root,100,100,x,y,2,0,0);
   XChangeWindowAttributes(disp,win2,CWBackingStore,&atr);
     XStoreName(disp,win2,names1);XFetchName(disp,win2,&name_return2);
    XMapWindow(disp,win2);
}
```

2.　8 方向コード

```
#include <stdio.h>
 #include <stdlib.h>
#include <math.h>
 #include <X11/Xlib.h>
```

```
#include <X11/Xutil.h>
#include <X11/Intrinsic.h>
#include <X11/StringDefs.h>
#include <X11/Shell.h>
#define            IMAX      106
#define            JMAX      156

static            Window                    root,win1,win2;
static            Display                   *disp;
static            Visual                    *vis;
static            Colormap                  cmap;
static            GC                        gc;
static            XColor                    color[256];
static            XSetWindowAttributes      atr;
static            FILE                      *fpin;
static            XImage                    *image;

typedef struct{
      int magic; int width; int heigth; int depth; int length;
      int type; int maptype; int maplength;
   }rasterfile;
rasterfile          raster;
char names1[] = { "input image" };          char *name_return1;
char names2[] = { "output image" };         char *name_return2;
u_char input_data[IMAX][JMAX];
u_char output_data[IMAX][JMAX];
u_char turn_data[IMAX][JMAX];

void win_inp(int);
void win_out(int,int);
void get_image(char *[], u_char *, int);
void put_image(u_char *, Window, int, int, u_char);
void insatu(u_char output_data[][JMAX]);

main(int argc, char **argv)
{
   int          i,j,a,k,fx,fy;
   double       pai=3.14159;
   double       ans;
   char         moji[9]={'0','1','2','3','4','5','6','7',' '};
```

```
    for(i=0;i < IMAX;i++){
        for( j = 0;j < JMAX;j++){
            input_data[i][j]=0;
            output_data[i][j] = 0;
             turn_data[i][j]=0;
        }
     }
     win_inp(argc);
       get_image( argv,( &input_data )[0][0],1);
    put_image(( &input_data ) [0][0], win1, JMAX, IMAX, 0 );
    XFlush(disp);
    for(i=0;i < IMAX;i++){
        for( j=0; j < JMAX ;j++){
          turn_data[i][j]=(input_data[i-1][j-1]+input_data[i-1][j]+
           input_data[i-1][j+1]+input_data[i][j-1]+
           input_data[i][j+1]+input_data[i+1][j-1]+
             input_data[i+1][j]+input_data[i+1][j+1]+
             2*(input_data[i][j]))/10;
        }
     }
    for(i=0;i < IMAX;i++){
        for( j=0; j < JMAX ;j++){
            a=turn_data[i-1][j-1]+turn_data[i-1][j]+
turn_data[i-1][j+1]+turn_data[i][j-1]+
    turn_data[i][j+1]+turn_data[i+1][j-1]+
       turn_data[i+1][j]+turn_data[i+1][j+1]+
            turn_data[i][j];
           if(turn_data[i][j]*9 >a ){
 output_data[i][j]=moji[8];
          }else{
fy=turn_data[i-1][j]-turn_data[i][j];
  fx=turn_data[i][j+1]-turn_data[i][j];
 /*fy=input_data[i-1][j]-input_data[i][j];
     fx=input_data[i][j+1]-input_data[i][j];*/
 if(fx==0&&fy==0){
             output_data[i][j]=moji[8];
 }else{
     ans=atan2((double)fy,(double)fx);
   /* k=(int)(4*(ans+2*pai)/pai +0.5);        direction turn    */
    k=(int)(4.0*(ans+1.5*pai)/pai+0.5);
```

```
   k=k%8;
    output_data[i][j]=moji[k];
}
           }
        }
     }
     insatu(output_data);
   win_out(JMAX,IMAX);
    put_image((&output_data)[0][0],win2,JMAX,IMAX,0);
   XFlush(disp);
    printf("finish");
    getchar();
}

 void insatu(u_char output_data[][JMAX])
{
    int         i,j,jst,m,r;
   for(jst=0;jst< JMAX;jst=jst+23){
       r=jst/23+1;
       printf("------8 directional code-----        part *%d*¥n",r);
       for(m=1;m<=38;m++){                        /*drawline "--" 38 times */
          printf("--");
        }
       puts("");
        printf(" **j=l");
       for(j=jst;j<=jst+22;j++){
          if(j<=JMAX){
printf("%3d",j);
           }
        }                                                  /*j is from 0 to 22 */
       puts("");
        for(m=1;m<=38;m++){
          printf("--");
       }                                                   /* "--" 38 times */
       puts("");
        for( i = 0; i < IMAX;i++){              /*(size of image or window)*/
          printf("i=%3dl", i );
            for( j = jst; j <=jst+ 22;j++){
 if(j<=JMAX){
    printf("    %c",output_data[i][j]);
```

```
}
            }
           puts("");
         }
        for(m=1;m<=38;m++){                          /* "--" 38 times */
           printf("--");
         }
        puts("");
        printf("------PART* %d * FINISH¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n", r);
    }
}

void get_image(char *argv[], u_char *vd, int no)
{
    int             i,j,pix;
    char            *dt_name,name[50],*dir="";
    u_char          dt[JMAX];
   dt_name=argv[no];
    strcat( strcpy( name, dir ), dt_name );
    fpin=fopen( name,"r" );
   if( fpin==NULL ){
       printf("Can't open %s.Abort!!¥n",argv[no]);
       fclose(fpin);     exit(1);
    }
   fread(&raster,sizeof(raster),1,fpin);
   pix=raster.maplength/3;
   for( i=0; i<pix; i++ )       color[i].red     = getc(fpin)<<8;
    for( i=0; i<pix; i++ )       color[i].green = getc(fpin)<<8;
   for( i=0; i<pix; i++ ){      color[i].blue    = getc(fpin)<<8;
                                        XAllocColor(disp,cmap,&color[i]);
    }
   for(i=0; i<IMAX; i++){
       fread( dt, sizeof(char), JMAX, fpin );
      for(j=0; j<JMAX; j++, vd++)          *vd =dt[j];
    }
   fclose(fpin);
}

 void put_image(u_char *data, Window win, int width, int height, u_char type)
 {
```

```
    int i;
   u_char *dt;
   dt=(u_char *)malloc(width*height);
     if(type==0)
        for(i=0;i<width*height;i++,data++)    *(dt+i)=(char)color[*data].pixel;
         else
            for(i=0;i<width*height;i++,data++)    *(dt+i)=*data;
       image=XCreateImage(disp,vis,8,ZPixmap,0,(char *)dt,width,height,8,width);
   XPutImage(disp,win,gc,image,0,0,0,0,width,height);
   free(dt);
    }

 void win_inp(int argc)
{
   if( argc!=2 ){
        printf("Usage: [Program][data_name]¥n");
       exit(1);
    }
     disp = XOpenDisplay(NULL);
    root = RootWindow(disp, 0);
    vis    = DefaultVisual(disp,0);
    cmap = DefaultColormap(disp, 0);
   gc      = XCreateGC(disp,root, 0, 0);
     atr.backing_store = Always;
        /*    about win1    */
   win1 = XCreateSimpleWindow(disp,root,100,100,JMAX,IMAX,2,0,0);
   XChangeWindowAttributes(disp,win1,CWBackingStore,&atr);
     XStoreName(disp,win1,names1);XFetchName(disp,win1,&name_return1);
    XMapWindow(disp,win1);
}

void win_out(int x,int y)
{
     disp = XOpenDisplay(NULL);
    root = RootWindow(disp, 0);
    vis    = DefaultVisual(disp,0);
    cmap = DefaultColormap(disp, 0);
   gc      = XCreateGC(disp,root, 0, 0);
     atr.backing_store = Always;
```

```
        /*    about win2    */
   win2 = XCreateSimpleWindow(disp,root,100,100,x,y,2,0,0);
   XChangeWindowAttributes(disp,win2,CWBackingStore,&atr);
    XStoreName(disp,win2,names1);XFetchName(disp,win2,&name_return2);
   XMapWindow(disp,win2);
}
```

3　エッジ方向

HEAD　ファイル

```
#define dirmap_width 8
#define dirmap_height 8
 static unsigned char dirmap_bits[8][8] = {
    {0x00, 0x20, 0x40, 0xfe, 0x40, 0x20, 0x00, 0x00},
```

```
    {0x00, 0x70, 0x60, 0x50, 0x08, 0x04, 0x02, 0x00},
    {0x08, 0x1c, 0x2a, 0x08, 0x08, 0x08, 0x08, 0x00},
    {0x00, 0x0e, 0x06, 0x0a, 0x10, 0x20, 0x40, 0x00},
    {0x00, 0x00, 0x04, 0x02, 0x7f, 0x02, 0x04, 0x00},
    {0x00, 0x40, 0x20, 0x10, 0x0a, 0x06, 0x0e, 0x00},
    {0x00, 0x10, 0x10, 0x10, 0x10, 0x54, 0x38, 0x10},
     {0x00, 0x02, 0x04, 0x08, 0x50, 0x60, 0x70, 0x00}};
```

Main プログラム

```
/* trace-k.c revised version of heikatuka-hosei-001-line-cir-ta.c
     revised by M. Oshima 2001.12.28 */
/*heikatuka-hosei-001-line-cir-ta.c      li gefeng 2001.12.8 */
 /* usage example trace image001.ras trace_data */

#include <stdio.h>
 #include <stdlib.h>
#include <math.h>
 #include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <X11/Intrinsic.h>
#include <X11/StringDefs.h>
#include <X11/Shell.h>
 #include"dipmap.h"
 #define              IMAX       106
 #define              JMAX       156

 static               Window                    root,win1,win2;
static               Display                  *disp;
 static               Visual                   *vis;
 static               Colormap                  cmap;
 static               GC                        gc;
 static               XColor                    color[256];
static               XSetWindowAttributes      atr;
 static               FILE                      *fpin;
 static               XImage                    *image;

 typedef struct{
       int magic; int width; int heigth; int depth; int length;
        int type; int maptype; int maplength;
   }rasterfile;
```

```
rasterfile              raster;
 char names1[] = { "input image" };                char *name_return1;
char names2[] = { "output image" };                char *name_return2;
u_char input_data[IMAX][JMAX];
u_char output_data[IMAX][JMAX];
 u_char turn_data[IMAX][JMAX];
u_char turn1_data[IMAX][JMAX];
u_char turn2_data[IMAX][JMAX];
u_char turn3_data[IMAX][JMAX];
u_char turn4_data[IMAX][JMAX];
u_char turn5_data[IMAX][JMAX];

 void win_inp(int);
void win_out(int,int);
 void get_image(char *[], u_char *, int);
void put_image(u_char *, Window, int, int, u_char);
 void insatu(u_char output_data[][JMAX]);
main(int argc, char **argv)
{
    int                      i,j,m,k,n,in,jn,iw,jw,ds,nb,nc,na,nt;
   int                       ib[8]={0,-1,-1,-1, 0, 1,1,1};
   int                       jb[8]={1, 1, 0,-1,-1,-1,0,1};
   unsigned      int      id[1000],jd[1000],ic, ict, ijt;
    FILE *fpo;
    fpo = fopen(argv[2], "w"); /* by M. Oshima 2001.12.28 */
   for(i=0;i < IMAX;i++){
       for( j = 0;j < JMAX;j++){
          input_data[i][j]=0;
           output_data[i][j]= 0;
           turn_data[i][j]=0;
          turn1_data[i][j]=0;
          turn2_data[i][j]=0;
          turn3_data[i][j]=0;

       }
    }
    win_inp(argc);
      get_image( argv,( &input_data )[0][0],1);
   put_image(( &input_data ) [0][0], win1, JMAX, IMAX, 0 );
  XFlush(disp);
```

```
    for(i=0;i < IMAX;i++){
        for( j=0; j < JMAX ;j++){
            turn_data[i][j]=(input_data[i-1][j-1]+input_data[i-1][j]+        input_data[i-1][j+1]+input_data[i][j-
             1]+
            input_data[i][j+1]+input_data[i+1][j-1]+
             input_data[i+1][j]+input_data[i+1][j+1]+
             2*(input_data[i][j]))/10;
        }
     }
    for(i=0;i < IMAX;i++){
        for( j=0; j < JMAX ;j++){
           k=(turn_data[i-1][j-1]+turn_data[i-1][j]+
   turn_data[i-1][j+1]+turn_data[i][j-1]+
   turn_data[i][j+1]+turn_data[i+1][j-1]+
  turn_data[i+1][j]+turn_data[i+1][j+1]+
   turn_data[i][j])/9;
            if(turn_data[i][j] >=k ){
turn1_data[i][j]=0;
          }else{
turn1_data[i][j]=1;
           }
        }
    }
   /* for(i=0;i < IMAX;i++){
        for( j=0; j < JMAX ;j++){
           if(j >28&&j<44&&i >84&&i<100){
turn2_data[i][j]=turn1_data[i][j];
          }else if(j >110&&j<127&&i>84&&i<100){
turn2_data[i][j]=turn1_data[i][j];
          }else{
turn2_data[i][j]=0;
           }
        }
     }*/
    for(i=0;i < IMAX;i++){
        for( j=0; j < JMAX ;j++){
          if(j >33&&j<45&&i >18&&i<33){
turn3_data[i][j]=turn1_data[i][j];
          }else{
turn3_data[i][j]=0;
```

```
            }
          }
        }
    /* for(i=0;i < IMAX;i++){
         for( j=0; j < JMAX ;j++){
            if(turn2_data[i][j]==1){
 if(turn2_data[i-1][j-1]==1||turn2_data[i-1][j]==1||turn2_data[i-1][j+1]          ==1||turn2_data[i][j-
1]==1){
    turn3_data[i][j]=1;
 }else{
    turn3_data[i][j]=0;
 }
           }else{
turn3_data[i][j]=0;
            }
          }
      }*/
      printf("turn3_data¥n");
 /*    insatu(turn3_data);*/
     nt = 0; /* number of traces */
    for(i=0;i < IMAX;i++){
       for(j=0; j < JMAX ;j++){
            if(turn3_data[i][j]==1){
nc=turn3_data[i][j+1]*turn3_data[i-1][j]*turn3_data[i][j-1]*turn3_data          [i+1][j];
if(nc==0){
     nt ++; /* by M. Oshima */
     if (turn3_data[i+1][j]==0) ds=6;
    else if(turn3_data[i][j-1]==0) ds=4;
    else if(turn3_data[i-1][j]==0) ds=2;
    else if(turn3_data[i][j+1]==0) ds=0;
     in=iw=i; jn=jw=j;
    n=(ds+4)%8;
    printf("start point is found, i=%3d,j=%3d,ds=%3d,n=%d¥n",i,j,ds,n);
    ds=-1;nb=-1;
     na=turn3_data[in-1][jn-1]+turn3_data[in-1][jn]+
          turn3_data[in-1][jn+1]+turn3_data[in][jn-1]+
          turn3_data[in][jn+1]+turn3_data[in+1][jn-1]+
          turn3_data[in+1][jn]+turn3_data[in+1][jn+1];
      if(na >0){
        for(ic=0;ic<1000;ic++) id[ic]=0;jd[ic]=0;
```

```
        ic=0;
         do {
            id[ic]=in=iw;jd[ic]=jn=jw;
           turn3_data[in][jn]=31;
           printf(" point is found, iw=%3d,jw=%3d,ic=%3d,m=%d,n=%d¥n",iw,jw,ic,m,n);
           ic++;
            n=(n+4)%8;m=0;
            do {
 n=(n+1)%8;if(m==8)    goto End;
iw=in+ib[n];
jw=jn+jb[n];
m++;
            }while(turn3_data[iw][jw]!=1&&turn3_data[iw][jw]!=31);
            if(ds<0){
 if(nb<0) nb=n;
else    ds=nb;
            }
          }while(i!=in||j!=jn||n!=ds);
    }
    else turn3_data[in][jn]=0;
End:; /* end of a trace */
   printf("nt,ic= %d %d¥n",nt,ic);
     for(ict =0; ict < ic; ict++){
       ijt = id[ict] * 1000 + jd[ict];
      fprintf( fpo, "% d",ijt);
    }
    fprintf( fpo, " %d¥n",0);
}
           }
        }
    }
     fclose(fpo);
     printf("turn3_data¥n");
   insatu(turn3_data);
    for(ic=0;ic<1000;ic++)
      output_data[id[ic]][jd[ic]]=31;
   win_out(JMAX,IMAX);
    put_image((&output_data)[0][0],win2,JMAX,IMAX,0);
    printf("output_data¥n");
   XFlush(disp);
```

```
    printf("finish");
    getchar();
 }

 void insatu(u_char output_data[][JMAX])
{
    int      jst,m,r,i,j;
   for(jst=0;jst<JMAX;jst=jst+23){
       r=jst/23+1;
        printf("******LABELING FOR IMAGE****          part *%2d* ¥n",r);
        for(m=1;m<=39;m++){            /* "--" 39 times */
          printf("--");
        }
       puts("");
        printf(" **j=l");
       for(j=jst;j<=jst+22;j++){
            if(j<=JMAX){
printf("%3d",j);
            }
        }
       puts("");
        for(m=1;m<=39;m++){
          printf("--");
        }                                        /* "---" 39 times */
       puts("");
        for( i = 0; i < IMAX;i++){
          printf("i=%3dl", i );
            for( j = jst; j <=jst+22;j++){
 if(j<=JMAX){
    printf("%3d",output_data[i][j]);
 }
          }
           puts("");
        }
        for(m=1;m<=39;m++){            /* "---" 39 times" */
          printf("--");
           }
       puts("");
        printf("---------PART* %d * FINISH¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n¥n",r);
     }
```

```
}

void get_image(char *argv[], u_char *vd, int no)
{
    int             i,j,pix;
    char            *dt_name,name[50],*dir="";
    u_char          dt[JMAX];
   dt_name=argv[no];
    strcat( strcpy( name, dir ), dt_name );
    fpin=fopen( name,"r" );
   if( fpin==NULL ){
       printf("Can't open %s.Abort!!¥n",argv[no]);
        exit(1);
    }
    fread(&raster,sizeof(raster),1,fpin);
    pix=raster.maplength/3;
    for( i=0; i<pix; i++ )       color[i].red      = getc(fpin)<<8;
    for( i=0; i<pix; i++ )       color[i].green = getc(fpin)<<8;
    for( i=0; i<pix; i++ ){      color[i].blue     = getc(fpin)<<8;
                                        XAllocColor(disp,cmap,&color[i]);
    }
    for(i=0; i<IMAX; i++){
        fread( dt, sizeof(char), JMAX, fpin );
       for(j=0; j<JMAX; j++, vd++)          *vd =dt[j];
    }
    fclose(fpin);
}

 void put_image(u_char *data, Window win, int width, int height, u_char type)
{
    int i;
   u_char *dt;
   dt=(u_char *)malloc(width*height);
     if(type==0)
       for(i=0;i<width*height;i++,data++)    *(dt+i)=(char)color[*data].pixel;
        else
          for(i=0;i<width*height;i++,data++)    *(dt+i)=*data;
      image=XCreateImage(disp,vis,8,ZPixmap,0,(char *)dt,width,height,8,width);
   XPutImage(disp,win,gc,image,0,0,0,0,width,height);
   free(dt);
```

```
    }

 void win_inp(int argc)
{
   if( argc!=3 ){
       printf("Usage: [Program][data_name][trace data name]¥n");
       exit(1);
    }
     disp = XOpenDisplay(NULL);
    root = RootWindow(disp, 0);
    vis    = DefaultVisual(disp,0);
    cmap = DefaultColormap(disp, 0);
   gc      = XCreateGC(disp,root, 0, 0);
     atr.backing_store = Always;

        /*    about win1    */
   win1 = XCreateSimpleWindow(disp,root,100,100,JMAX,IMAX,2,0,0);
   XChangeWindowAttributes(disp,win1,CWBackingStore,&atr);
    XStoreName(disp,win1,names1);XFetchName(disp,win1,&name_return1);
    XMapWindow(disp,win1);
}

void win_out(int x,int y)
{
     disp = XOpenDisplay(NULL);
    root = RootWindow(disp, 0);
    vis    = DefaultVisual(disp,0);
    cmap = DefaultColormap(disp, 0);
   gc      = XCreateGC(disp,root, 0, 0);
     atr.backing_store = Always;

        /*    about win2    */
   win2 = XCreateSimpleWindow(disp,root,100,100,x,y,2,0,0);
   XChangeWindowAttributes(disp,win2,CWBackingStore,&atr);
    XStoreName(disp,win2,names1);XFetchName(disp,win2,&name_return2);
    XMapWindow(disp,win2);
}
```

4. 閾値処理

```
/*This is a program for reading monochrome data in file
                    to do something on it and display it.*/
#include <stdio.h>
#include "vismain.h"

int main(int argc, char *argv[])/*=========================================*/
```

```
{
    char data_name[100],keyin;
   struct rasin *rasin;
    unsigned char *vdata, *vdata2, *vd[MAX_HEIGHT], *vd2[MAX_HEIGHT], *vdisp;
   int width, height,a;
   XColor *color; int pixs;
    Window win1;
    int x, y, fxy, points, depth, gray=1;
  XImage *image;
   Colormap cmap; /*2000.08.10*/
   if( argc<2 ){
        printf("Usage: [Program][data_name]¥n");
       return FAIL;
    }
    sprintf(data_name,"%s",argv[1]);
   rasin = get_image(data_name, &width, &height, &pixs);
    if(rasin == NULL) goto err;
    vdata = rasin->vdata;
  color = mono_cmp(rasin);
   pixs = rasin->pixs;
         color = rasin->color;
    points = width * height;
    vdata2 = (unsigned char *)malloc(points);
   if(vdata2 == NULL) goto err;
  for(y = 0; y < height; y++){
        vd[y] = vdata + y*width; vd2[y] = vdata2 + y*width;
    }
        while(1){
       printf("please input data from 1-255:¥n");
      scanf("%d",&a);
       if(a > 0 && a <= 255){
              for(y = 0; y < height; y++){
       for(x = 0; x < width; x++){
   fxy = f_xy(vd,x,y);
    if(fxy > a ){
        fxy=255;
    }
   else {
        fxy=0;
    }
```

```
    f_xy(vd2,x,y) = (unsigned char) fxy;
fxy = *(vdata + y*width + x)
        }
 }
           vdisp = vdata2;
           rasin->vdisp = vdisp;
          cmap = set_cmap(color, pixs, rasin);
 cmap = set_cmap(color, pixs);    /* 2000.08.31 */
           win1 = win_set(width, height, "output data", cmap);
           printf("Hit Return Key for window setting¥n");
getchar();
           depth =8;
          image = put_image (vdisp, width, height, depth, win1, color, pixs, gray);
              }
       else{
          puts("The data is unuseful!!!");
          break;
        }
    }
    printf("Hit Return Key to end¥n");
    getchar();
   XFree(image);
    free(vdata); free(vdisp);
    return SUCCESS;
     err:       printf("error in main¥n");
    return FAIL;
}
```

END