

ビデオレート・ラベリングプロセッサ

太田 善之, 古明地 正俊, 後藤 敏行, 吉田 真澄

株式会社 富士通研究所

本報告は、画像中の複数領域を分離する処理であるラベリングをビデオレートで行うことのできるプロセッサに関するものである。筆者らは既にパイプライン処理を可能にしたラベリングアルゴリズムを提案した。本プロセッサはこのアルゴリズムに基づいて構成しており、前処理の導入により従来の方式に比べ、ラベル接続数を低減することができる。また、ラベルの接続関係を直接探索するラベル分類方法を用いているため、暫定ラベル数に比例する演算量で処理を終了することができる。試作したプロセッサは、 512×512 サイズの実画像に対して最大 255 個の領域までビデオレートでラベル付け処理を実行することができる。

Video-rate labeling processor

Yoshiyuki OHTA, Masatoshi KOMEICHI, Toshiyuki GOTOH, Masumi YOSHIDA

Pattern Information Processing Laboratory, FUJITSU LABORATORIES LTD.,

1015 Kamikodanaka, Nakahara-ku, Kawasaki, 211 JAPAN

This paper describes the labeling processor, which can process labeling connected components in an image at video rate. We have already proposed a high speed algorithm for labeling amenable to pipeline processing. Based on the algorithm, the processor reduces provisional labels using the pre-processing which expands regions in an image and smooths their shape. As the processor directly searches the equivalent labels on the plane graph which is composed of provisional labels and their connectivities, the processing time for classifying the equivalent labels is proportional to the number of the provisional labels. The developed processor can process labeling up to 255 connected components in an 512×512 image at video rate.

1. はじめに

近年、画像処理の応用分野の拡大、画像処理技術の進歩に伴い、静止画像処理から動画画像処理への要求が増えている。それに伴い、各種ビデオ機器を入出力とし、それらの信号を直接処理できる動画画像処理装置の開発が盛んに行われるようになってきた。しかし、画像中に散在する複数の対象物を分離する処理であるラベリングは、逐次演算による画像の連結性判定が不可欠であるため⁽¹⁾、高速化が困難であった。

筆者らはこれまでに、パイプラインアーキテクチャを基本とし、カラー画像をビデオレート（30画像/秒）で処理できる画像処理装置「章駄天」を開発した⁽²⁾。また、従来のラスタ走査型ラベリングアルゴリズムに改良を加え、パイプライン処理を可能としたラベリング方式を提案した⁽³⁾。今回、この方式に基づき、ビデオレートでラベリングを実行するプロセッサを開発し、「章駄天」の演算機能として実現した。

本報告では、我々の提案したラベリングアルゴリズムについて簡単に述べた後、試作したハードウェア諸元と動画画像を用いた実験例について述べる。

2. アルゴリズム⁽³⁾

2.1 ラスタ走査型ラベリングアルゴリズム

従来のラスタ走査型アルゴリズムの概略を図1に示す。このアルゴリズムでは、最初に画像を左から右、上から下へラスタ走査して、ウィンドウ内の近傍画素を参照しながら領域内の各画素に暫定的にラベル（暫定ラベル）を付与する〔暫定ラベル付処理〕。また、隣り合った画素に異なる暫定ラベルが付与されている場合には、それらの暫定ラベルを接続関係として検出する〔接続関係検出処理〕。次に、検出された接続関係を基に、直接・間接を問わず互いに接続しているラベルをグループ化し、分類する〔ラベル分類処理〕。最後に、ラベルの分類結果に基づいて暫定ラベルを領域毎に個別のラベルに変換する〔ラベル更新処理〕。

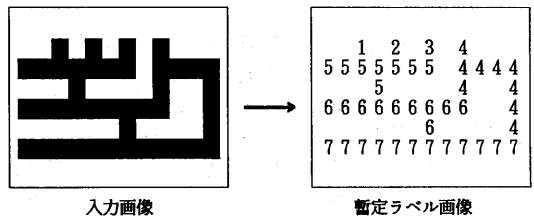
2.2 問題点

ラスタ走査型アルゴリズムにおいて、暫定ラベル付処理、接続関係検出処理およびラベル更新処理は画像を走

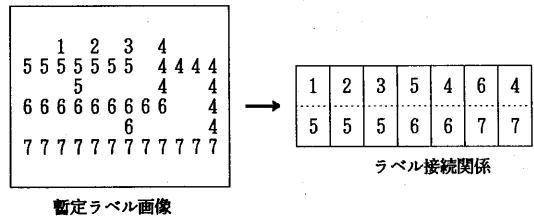
査する演算であり、パイプライン方式により高速化を図ることができる。ラベリングの高速化はラベル分類処理をいかに速くするかによっており、それは暫定ラベルの数、ラベルの接続数およびラベル分類処理の能力によって決まる。

(1) 暫定ラベル数、ラベル接続数の問題

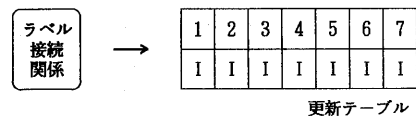
暫定ラベル付処理は、局所的なウィンドウ内の画素のパターンによってラベルを付けている。よって、局



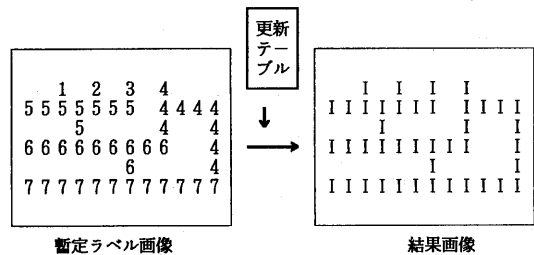
(a) 暫定ラベル付け処理



(b) 接続関係検出処理



(c) ラベル分類処理



(d) ラベル更新処理

図1. ラスタ走査型ラベリング方式

対して多くの暫定ラベルが付与される。これに伴ってラベル同士の接続関係の数も増大してしまう。

(2)ラベル分類処理の問題

従来のラベル分類処理では、ラベルの接続数の2乗に比例する演算が必要であった。例えば、巡回リスト法⁽⁴⁾では、各暫定ラベルに対して1語分の記憶場所を用意しておき、同一領域に属する暫定ラベルが巡回リストをなすように接続関係を格納する。新たに接続関係が検出されると、リストを参照して同一の暫定ラベルが既に存在しているかをテストする必要がある。最悪の場合、リスト上のすべての暫定ラベルを参照しなければならず、分類処理の演算量は暫定ラベル数の2乗のオーダーになる。

2.3 解決策

暫定ラベル数・ラベル接続数の増大、ラベル分類処理の演算量に関する問題点に対して、以下のような改良を行った。

(1)局所的形状変化への対処

暫定ラベル付処理を実行する前に前処理を施す。前処理では、画像内の各領域の形状を、別の領域が連結しないことを条件として凹部分を埋めて各領域の局所的な形状変化をできるだけ抑え、暫定ラベル付処理で付与されるラベル数を低減する。(図2参照)

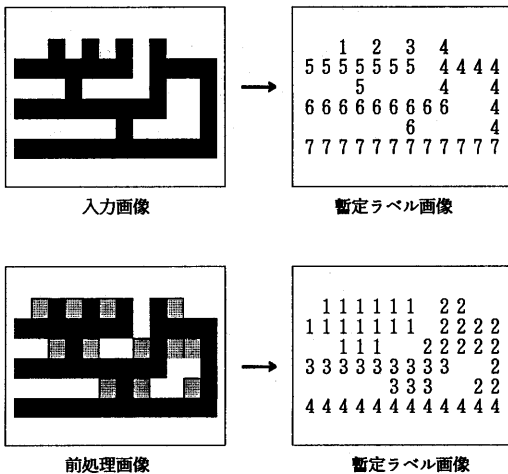


図2. 前処理による暫定ラベルの低減

(2)ラベル分類方法の改良

図3に示すように、暫定ラベル付処理が施された画像(暫定ラベル画像)では、暫定ラベル同士の接続の様子を、暫定ラベルをノード、接続関係をアークとみなした平面グラフを構成している。よって、ラベルの分類処理はこのグラフの探索問題に置き換えることができる。グラフ理論によれば、ノード数Lとアークの数Cmは

$$C_m \leq 3 \times L - 6$$

という関係がある⁽⁵⁾ので、暫定ラベルの平面グラフ上でアークを直接たどる探索を行えば、ラベル分類処理を暫定ラベル数に比例する時間内に終了させることができる。

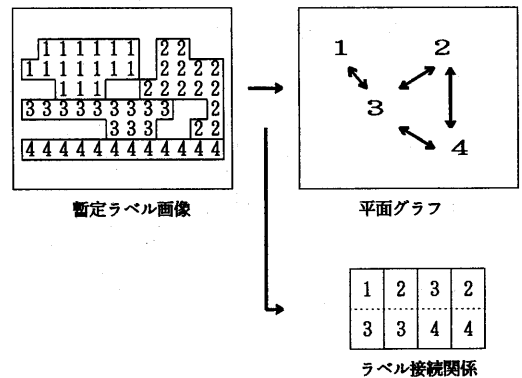


図3. 接続関係の平面グラフ表現

3. ハードウェア構成

第2章に述べたような改良を加えた、本ラベリングプロセッサの処理フローを図4に示す。

3.1 前処理部

前処理部では、各領域に付与される暫定ラベルの数を低減するために、入力2値画像の各領域の凹部分を膨張させる。本プロセッサでは、3×3のウィンドウをラスト走査させ、中心画素の値が0の際、周囲の8画素の組み合わせにより、隣り合う領域が接続しない場合に中心

画素の値を1にする論理フィルタ処理を施している。本プロセッサでは4連結を考慮しており、膨張処理の論理は図5に示すようになる。また、膨張処理の効果を高めるため、図6に示すように、ある画素に対する処理結果を次の画素の処理に利用する再帰型のフィルタ処理を施している。

膨張された各領域は、暫定ラベルを付与した後、原画像と論理積をとることにより形状を復元する。

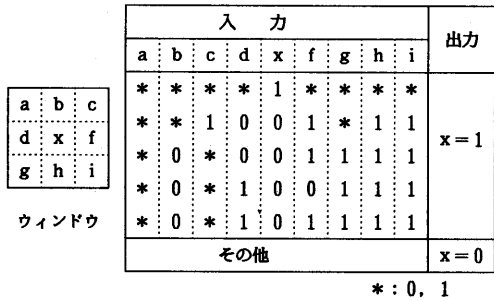


図5. 前処理の演算論理

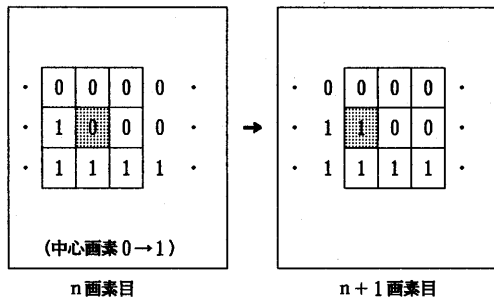


図6. 再帰型論理フィルタ処理

入 力					出 力	
x	d	b	c	f	暫定ラベル	接続関係
0	*	*	*	*	0	—
1	1	0	*	*	D	—
1	1	1	*	*	D	B, D
1	0	1	*	*	B	—
1	0	0	1	*	C	—
1	0	0	1	0	(new)	—
1	0	0	0	*	(new)	—

* : 0, 1. また小文字の画素に付与されているラベルを大文字で表す。

図7. 暫定ラベル付の論理

3. 2 暫定ラベル付部

暫定ラベル付部では、図7に示すような3×2のウィンドウをラスタ走査させながら各領域に暫定ラベルを付与する。同時に、xの画素とbの画素に付与されている暫定ラベルの値が異なる場合には、両者の値を接続関係として出力する。暫定ラベル付けおよび接続関係抽出の論理を図7に示す。

接続関係は画素毎に検出されるため、画像中の離れた2点で同一接続関係が抽出される可能性がある。重複して同一接続関係を抽出することを避けるため、フラグテーブルを用いてチェックを行っている。初めて抽出された接続関係の場合はフラグテーブル内に1を立て、同一接続関係が再び抽出された時にはテーブル内にビットが立っているため、その接続関係を出力しない。従来では、予想される暫定ラベルの最大値をLとすると、図8(b)

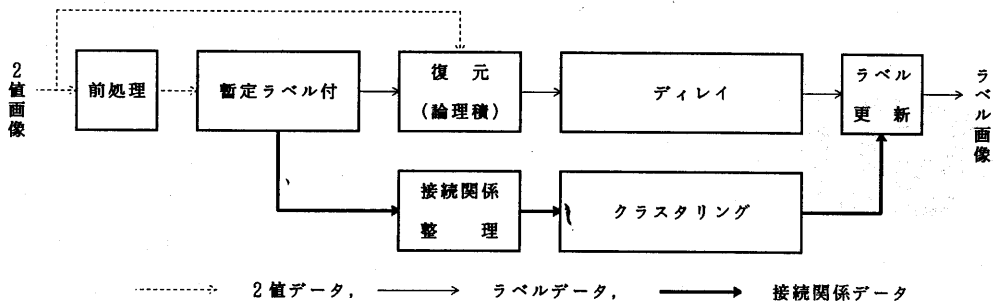


図4. ラベリングプロセッサ・ハードウェアブロック図

に示すように $L^2 / 2$ の大きさのフラグテーブルを用意していた。しかし、図 8 (c) のように、画像を横方向にいくつかの部分画像に分割すると、暫定ラベルの接続関係は部分画像内および隣り合う部分画像間に限られる。そこで、フラグテーブルの参照範囲を図 8 (d) に示すように限定することができ、フラグテーブルの容量を減らすことができる。

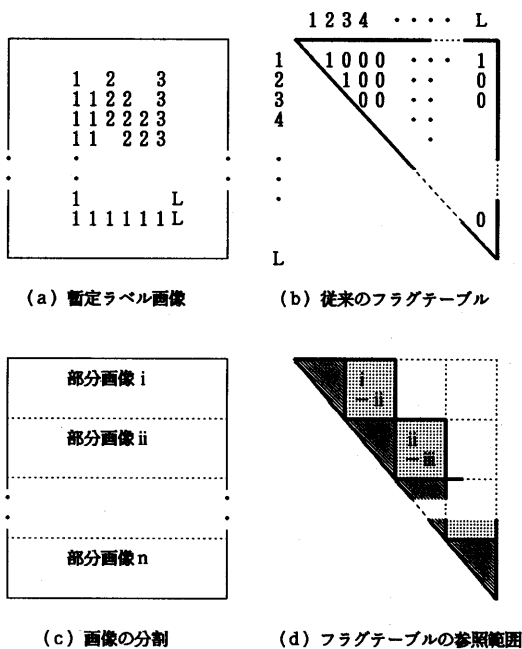


図 8. 重複接続関係の除去

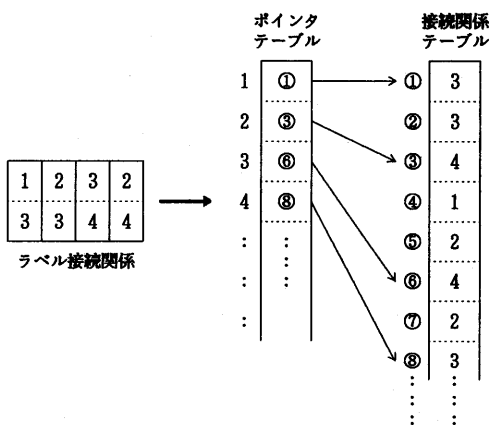


図 9. 接続関係整理の様子

3. 3 接続関係整理部

接続関係整理部では、暫定ラベル付部で画素毎に抽出された接続関係を暫定ラベル別に参照することができるように、図 9 に示すような接続関係テーブルとポインタテーブルを作成する。

◇接続関係テーブル

1 次元の配列であり、暫定ラベル 1 と接続しているすべてのラベルをテーブルの先頭から順に格納する。

暫定ラベル n と接続しているすべてのラベルを、ラベル $n-1$ に接続しているラベルのうち、最後のラベルが格納されている次のアドレスから順に格納する。

◇ポインタテーブル

接続関係テーブルにおいて、ある暫定ラベルと接続しているラベルのうち最後のラベルが格納されているアドレス値を暫定ラベル毎にセットしている。このような構成のテーブルを作成することによって、暫定ラベル n に対するポインタテーブルの値から $n-1$ に対する値までをアドレスとして接続関係テーブルをアクセスすることにより、ラベル n に直接接続しているラベルをすべて参照することができる。

ポインタテーブル、接続関係テーブルは以下の手順で作成する。

1. 暫定ラベル付部で抽出された接続関係をすべて書き、同時に暫定ラベル毎に接続数を係数する。
2. 接続数の係数値を暫定ラベル 1 から累積し、ポインタテーブルを作成する。
3. 書いておいた接続関係を読みだし、ポインタテーブルを参照して 1 次元の接続関係テーブルを構成する。

3. 4 クラスティング部

クラスティング部では、接続関係テーブルの内容を参照しながら、直接および間接的に接続しているすべての暫定ラベルを探索し、それらに同一の更新ラベルを付与する。本プロセッサでは、探索手順をプログラム化したマイクロプロセッサでクラスティングを行い、結果は更新テーブルに格納する。探索手順は以下の通りである。

[Step 1]

1 画像中に存在する暫定ラベルにおいてその最大値からクラスタリングを開始する。

更新ラベル値 = 1。

[Step 2]

最大暫定ラベルから降順に更新テーブルの内容を参照し、更新ラベルがまだ付与されていない暫定ラベル（以下「ラベルA」）を捜す。

そのようなラベルがない場合には処理終了。

[Step 3]

ラベルAに接続している暫定ラベルを接続テーブルから1つずつ読み、更新ラベルがセットされていなければ、Aと同じ更新ラベルをセットすし、同時に、読みこんだ暫定ラベルをスタックに蓄える。

[Step 4]

ラベルAに接続している暫定ラベルがなくなったら、スタックの内容を1つずつ引き出し、Step 3の処理を繰り返す。スタックが空の場合は、ラベルAに対する探索を完了し、更新ラベル値を+1して、Step 2へ。

図3のような接続関係が得られた場合の探索の様子を図10に具体的に示す。ポイントテーブル、接続テーブルは図9に示したようになっており、クラスタリングは暫

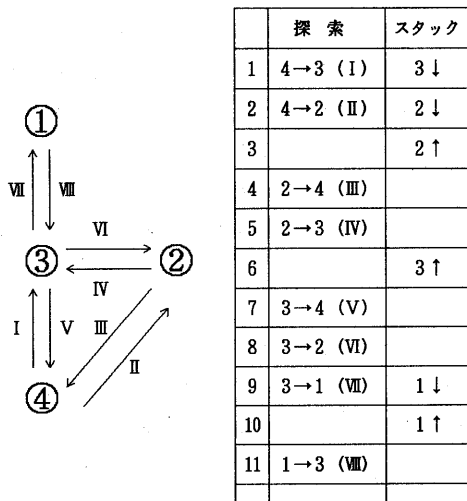


図10. ラベル接続関係の探索

定ラベルの最大値4から始まって、4→3→2→1の順に平面グラフのアーキを直接たどって探索を行っている。

3. 5 ラベル更新部

ラベル更新部では、クラスタリング部で作成した更新テーブルをLUTとして用い、暫定ラベルを更新して各領域に個別のラベル値を付与する。

3. 6 パラメータの最適化

我々はシミュレーションを行い、ハードウェア化に伴う各種パラメータの最適化を行った⁽²⁾。

暫定ラベル数とクラスタリングに必要な演算量の関係は図11に示したようになる。図において、演算量とは探索においてあるラベルから次のラベルへ遷移する回数である。テストコーディングではラベルの遷移に27ステップ必要であり、100nsec/命令で動作するマイクロプロセッサで実行した場合、暫定ラベルが約3000以下の場合にクラスタリング処理がビデオレートで実行可能となる。一方、暫定ラベルを3000以下に抑えるた

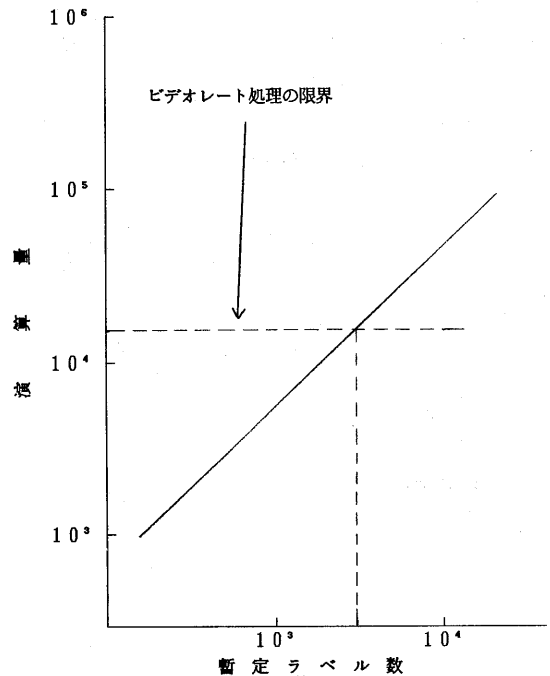


図11 暫定ラベル数と演算量の関係

めには、前処理が9回以上必要である。

以上により、インプリメントに際しては、前処理回数12回、暫定ラベル付部では暫定ラベルが255を越える度に画像を分割し、最大8分割まで可能とした。暫定ラベルが2040を越えた場合および更新ラベルが255を越えた場合にはラベルオーバーフローとし、ラベリング処理を行わず、入力2値画像をそのまま出力する。

4. 試作プロセッサと実験例

試作したプロセッサの性能は以下の通りである。

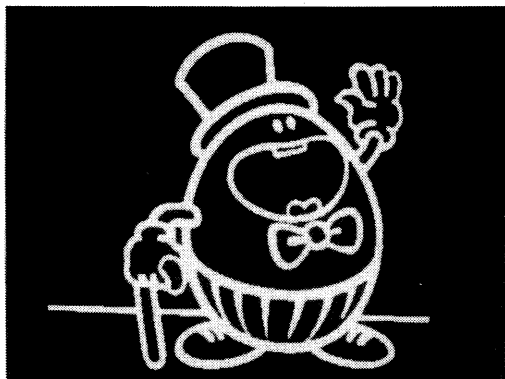
- ・対象画像サイズ: 512×512 (2値画像)
- ・最大暫定ラベル数: 2040 ラベル
- ・分離可能領域数: 255 領域 (出力8ビット)
- ・処理速度: 30 画像/秒

- ・パイプラインピッチ: 120 nsec/画素
- ・クラスタリング部マイクロプロセッサ
マシンサイクル: 120 nsec
マイクロプログラム: 46ステップ (初期化を含む)

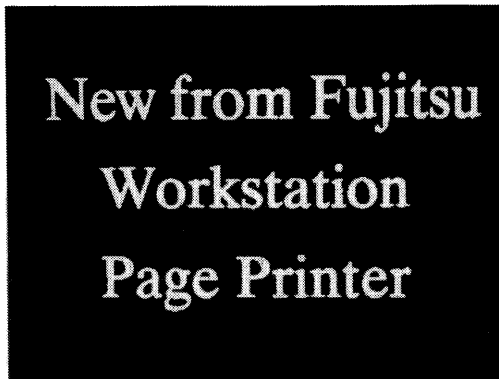
・試作ハードウェア量

- 前処理部 ⇒ A4版プリント板3枚*
- 暫定ラベル付部 ⇒ " 1枚
- 接続関係整理部 ⇒ " 4枚
- クラスタリング+ラベル更新部
⇒ A4版プリント板1枚
- ディレイ部 ⇒ " 3枚*

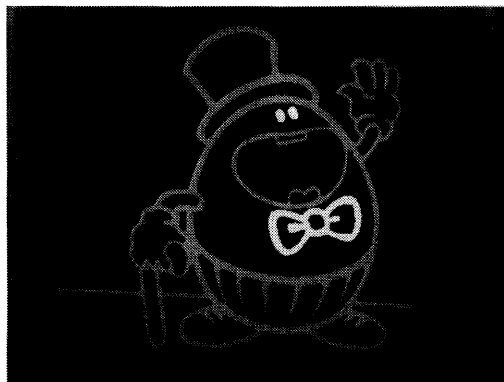
[*前処理部、ディレイ部は「韋駄天」⁽²⁾の既開発モジュールを利用している。]



(a) 2値画像 (入力)

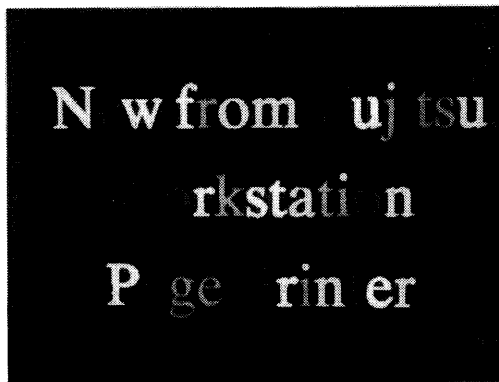


(a) 2値画像 (入力)



(b) ラベル画像 (出力)

図12. 処理例1



(b) ラベル画像 (出力)

図13. 処理例2

実際の動画像に対する処理の例を図12, 13に示す。両図において(a)は本プロセッサへ入力される2値画像, (b)は本プロセッサの出力であり, ここではラベル値ある濃度値に割り当てて表示した。図12の例では, 暫定ラベル約630個, ラベルの接続数約650, 更新ラベル4個であり, クラスタリングに要した時間は約5 msecである。また, 図13の例では, 暫定ラベル約370個, ラベルの接続数約350, 更新ラベル約40個であり, クラスタリングに要した時間は約2 msecである。

5. まとめ

ビデオレートでラベリング処理を行うことのできるプロセッサについて述べた。本プロセッサでは, 領域の形状を整形して暫定ラベル数を低減する前処理を導入した。また, 暫定ラベル数に比例する演算量で暫定ラベルの分類処理を終了するクラスタリングを有し, 処理の高速化を図っている。試作したハードウェアでは, 512×512 の大きさの画像に対して255領域まで, ビデオレートでラベリング処理を実行することができる。

謝辞

本プロセッサ実現にあたり, 有益な御助言を頂いた電総研、白井部長ならびに関係各位に感謝する。

参考文献

- (1) Minsky, M., Papert, S.: "Perceptron", 8.73, The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, and London, England, 1969.
- (2) 古明地, 佐々木, 尾崎, 佐藤, 太田: カラー動画像処理システム「章駄天/カラー」, 情処学会CV研, CV49-2, 1987.
- (3) 後藤, 太田, 吉田, 白井: バイブライン型高速ラベル付けアルゴリズム, 情処学会CV研, CV43-1, 1986.
- (4) 白井, 杉原: 連結領域を求める手法とそのハードウェア, 信学会画像工学研資, IE78-9, 1978.
- (5) 例えば Harary, F. (池田貞雄訳): 「グラフ理論」 共立出版, 1974.
- (6) Gotoh, T., Ohta, Y., Yoshida, M.: "Component labeling algorithm for video rate processing", Proc's of Advances in Image Processing, pp. 217 - 224, SPIE Vol. 804, 1987.
- (7) 太田, 古明地, 後藤, 吉田: ビデオレート・ラベリングプロセッサ, 情処学会36回全国大会3V-5, 1988.