

パイプライン型高速ラベル付けアルゴリズム

後藤 敏行 * 太田 善之 * 吉田 真澄 *

白井 良明 **

* 株式会社 富士通研究所

** 電子総合技術研究所

筆者らは画像中の複数領域を分離するラベル付け処理をパイプラインで実行できるアルゴリズムを開発した。従来のラスタ走査型のラベル付け処理は、各領域に対して暫定的に付与したラベルの分類処理が高速化のネックとなっていた。これに対して、前処理によるラベル接続数の低減、分割処理による重複接続関係の削除および接続関係の直接探索によるラベル分類方法の考案により、暫定ラベル数に比例する演算量で実行できる新アルゴリズムを開発した。本論文ではこのアルゴリズムについて説明するとともに、シミュレーションによる検証実験の結果について報告する。本アルゴリズムの実現によってラベリング処理をビデオレートで行える見通しを得た。

Pipe-lined Algorithm for Labeling Connected Regions at Video Rate

Toshiyuki GOTOH * Yoshiyuki OHTA * Masumi YOSHIDA *

Yoshiaki SHIRAI **

* FUJITSU LABORATORIES LTD.

** Electro Technical Laboratory

We have developed a high speed algorithm for region labeling amenable to pipelining.

In this paper, we propose a new pre-processing method for reducing the provisional labels. This method expands regions in an image. We also propose a classification method for labels whose processing time is proportional to the number of the provisional labels.

The experiments using computer simulation verify that images can be processed in pipeline by using this algorithm.

1. はじめに

筆者らはパイプライン・アーキテクチャを基本とした画像処理システムの開発を進めており、これまでに「章駄天」と名付けたビデオレート画像処理システムを発表した⁽¹⁾。今回、このアーキテクチャのもとで動作するラベリング方式を構築した。

ラベリングは画像中に散在する複数の対象物（領域）を分離するための処理であり、その応用範囲も極めて広い。しかし、この処理は逐次演算による画像の連結性の判定が不可欠なために⁽²⁾、その実現においては高速化が難しい。これまで、ラベリングの高速化をめざした様々な検討がなされてきたが、それらはランダム走査型とラスタ走査型のアルゴリズムに大別できる⁽³⁾。ランダム走査型では画像をランダムに走査しながら領域の探索とラベルの伝播を繰り返して行う。この方法はラベルの伝播に際して、画像をランダムにアクセスする必要があり、パイプライン化は根本的に不適である。一方、ラスタ走査型では最初に画像を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各領域に暫定的にラベル（暫定ラベル）を付与し、次にこのラベル同士の接続関係に基づいてラベルの分類を行い、最後にこの結果を使ってラベルを更新する。この方法は暫定ラベルの分類処理に膨大な演算量を必要とし、さらにこの演算量の限界値が理論的に把握しづらいために、パイプライン処理を実現できないという問題があった。

以下、本稿では従来のラスタ走査型ラベリング・アルゴリズムにおいて、高速化を阻害している要因について論じた後、従来のアルゴリズムの問題点を解決しパイプライン化を可能にした新しいアルゴリズムを提案する。

2. 問題点と解決策

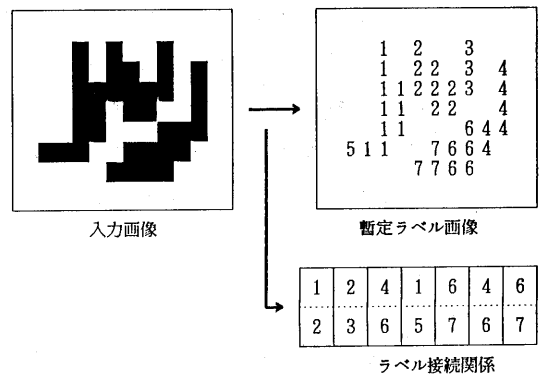
2.1 ラスタ走査型ラベリング・アルゴリズム

従来のラスタ走査型アルゴリズムの概要を図1に示す。ここでは、最初に暫定ラベル付け処理により各領域に暫定ラベルを付与するとともに、ラベル間の接続関係を検出する（図1-(a)）。次にラベル分類処理では画素ごとに検出された接続関係を整理し分類する（図1-(b)）。そして、最後のラベル更新処理では、ラベルの分類結果に基づいて、暫定ラベルを個々の領域に対応したラベル

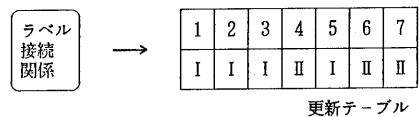
に変換する（図1-(c)）。

2.2 問題点

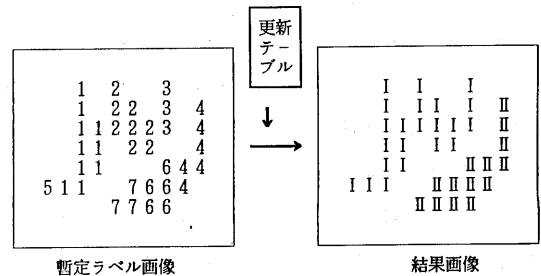
ラベリングの高速化は、基本的に入力画像をラベルの世界におとすラベル接続関係の数とラベルの世界で行うラベル分類処理の能力によって決まる。ラベル接続関係の数が少なくても、ラベル分類処理の能力が低ければ高速処理が達成できず、一方、ラベル分類処理の能力が高くても、ラベル接続関係が多ければ高速処理は達成できない。したがって、ラベリングの高速化を実現するため



(a) 暫定ラベル付け処理



(b) ラベル分類処理



(c) ラベル更新処理

図1 ラスタ走査型ラベリング方式

には、ラベル接続数を低減させるとともに、ラベル分類処理そのものを高速化する必要がある。

◆ラベル接続関係の問題

ラベル接続関係の数が膨大になるのは2つの原因がある。第一は局所的形状変化の激しい画像においては、単一の領域に対して、数多くの暫定ラベルが付与され、これにともなって接続関係の数も増大してしまうことである。これに対処するために、暫定ラベル付け処理用ウィンドウの形状を工夫することにより暫定ラベル数を低減する試みが報告されているが⁽⁵⁾、ウィンドウ形状の工夫だけでは、単純な凹凸形状に対して効果があるが、複雑な形状に対して効果が少ないという問題がある。第二は接続関係の検出が局所的なウィンドウ内の情報に基づいて行われるために、図1-(a)の接続関係において、ラベル6とラベル7のように同一領域内で同じ接続関係が重複して検出されるという問題である。これに対処するために、三角メモリ法という重複接続関係検出手段が提案されている⁽⁴⁾。しかし、この方法においても1つの画像ごとに接続関係の数の二乗に比例したテーブルを参照しなければならないという問題がある。

◆ラベル分類方法の問題

従来のラベル分類処理はラベルの接続数の2乗に比例した演算量が必要となり、高速化が難しい。例えば、巡回リスト法⁽⁴⁾では、接続関係が検出されるごとに、リストを参照して同一の暫定ラベルがその中にすでに存在しているかどうかテストする必要があるが、最悪の場合リスト中の暫定ラベルをすべて参照することになり、その演算量は暫定ラベル数の2乗のオーダーとなる。また、MIN-MAX法⁽⁵⁾では、接続関係が検出されるごとにテーブル内の暫定ラベル数を付け替える操作を必要とするが、この操作が暫定ラベルの2乗のオーダーの演算量となる。

2.4 解決策

筆者らはラベルの接続関係数の低減とラベル分類処理の演算量削減の両面から検討を行い、

- ・局所的形状変化への対処

- ・重複接続関係の削除方法の改良

- ・ラベル分類方法の改良

を柱とした新しいアルゴリズムを開発した。

局所的形状変化への対処については、画像内の各領域の形状をそれぞれの領域が接続しないことを条件とした画像の整形を行うことで暫定ラベルを削減する。

これによって、接続関係の低減が図られるが、そのラベルの接続関係数は次式で与えられる。

$$C_m \leq 3 \times L - 6 \quad \text{..... (式 2-1)}$$

ここで、Lは暫定ラベル数、C_mは接続関係の数である。なお、この条件式はラベルの接続関係が平面グラフ⁽⁶⁾を構成することから導いたものである。

重複接続関係の削除については、三角メモリ法⁽⁴⁾を基本として、分割処理の導入により演算量を削減できるように改良した。

ラベル分類については、ラベルの接続関係に基づいて、暫定ラベルを直接に探索することにより、ラベルの接続数のオーダーで処理できる方法を考案した。

3. 開発したアルゴリズム

3.1 ラベル接続数の低減

本方式では前処理による領域形状の整形を行うことにより、暫定ラベル数を削減し、これによってラベル接続数の低減を可能にした。

図2に従来技術と開発した技術を比較した原理を示す。従来技術では、入力画像に田型ウィンドウを直接に走査し暫定ラベル付け処理を行う。これによって、この画像には1~7のラベルが付与される。これに対して、開発した技術では、画像内の領域が結合しない条件で各領域を△形状になるように膨大化し、その画像に暫定ラベル付け処理を行う。これによって、画像に付与される暫定ラベルは1~3となり、暫定ラベル数を大幅に削減することができる。

実際には図3に示すように、前処理による画像の整形の後に、ラベリングを行い、その結果と入力画像の論理積をとることにより、前処理によって生じた領域形状の歪を矯正し、入力画像に対する正当なラベル画像を得る。

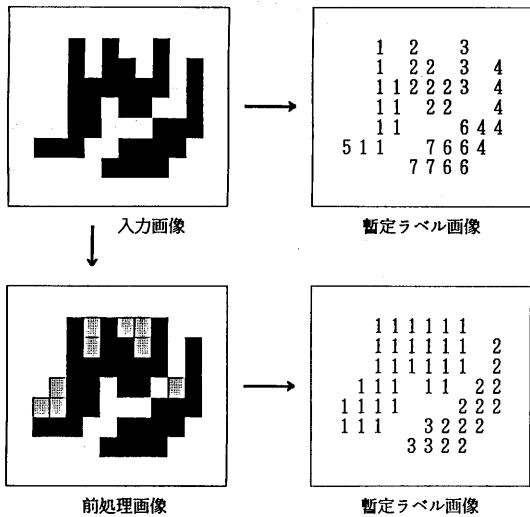


図2 前処理による暫定ラベルの低減

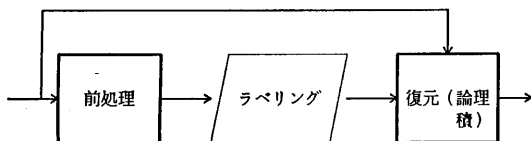


図3 前処理と形状復元

3.2 重複接続関係の削除

一般に、ラベル接続関係の検出は局所的なウィンドウにより画素ごとに判定するために、同一の接続関係が領域内の異なる部分で重複して検出される。これに対処するために、三角メモリ法では図4に示すような三角形のテーブルを用意し、接続関係が検出されると、このテーブルの対応するアドレスに1を書き込む。そして、全画像の処理が終了した後に、この内容を参照することによって、重複接続関係を削除する。

しかし、この方法によるとメモリ容量が $L^2/2$ となり、また、これを参照するための演算量は $L^2/2$ に比例することになる。これに対して、筆者らは画像の分割処理を導入することにより、メモリ容量および演算量を削減できる方法を考案した。図5にその原理を示す。ここでは、図5-(a)に示すように入力画像を分割し、それぞれの部分画像に対して独立に暫定ラベルを付与するとともに、

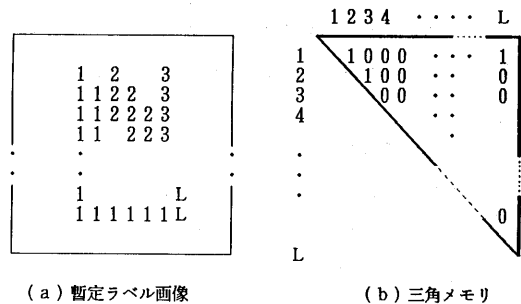


図4 三角メモリ法の原理

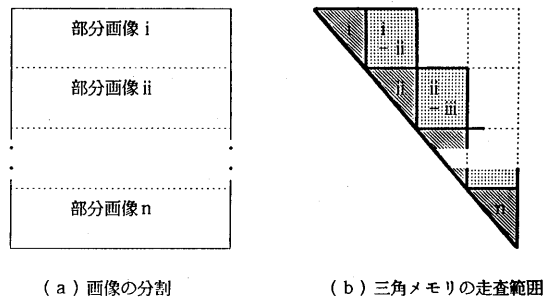


図5 三角メモリ法の改良

部分画像内の接続関係と部分画像間の境界における接続関係を検出する。各部分画像は独立に暫定ラベルを付与されているので、検出される接続関係は部分画像内では三角メモリの斜線の部分にのみ現れ、また部分画像間では点線の部分にのみ現れる。したがって、テーブルの参照はこれらの部分に対してのみ行えばよい。ここで、画像全体に暫定ラベル付けを行ったときの最大ラベル数 L と分割処理を行ったときのラベル総数 L' がほぼ等しい($L \approx L'$)と仮定した場合、演算量は

$$(\text{重複接続関係整理の演算量}) \propto 2L^2 / 3n$$

でおさえることができる。ただし、 n は画像の分割数である。

3.3 ラベル分類処理の改良

ラベル分類処理の目的は“2つのラベルが接続している”という情報の集合から、直接的および間接的に接続しているラベルを1つのグループにまとめ上げることに

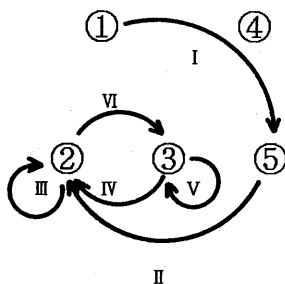
ある。前章で述べたように、暫定ラベルをノード、接続関係をアークと見なすと、ラベル分類処理はグラフ理論における平面グラフの探索問題に置き換えて考えることができる。したがって、暫定ラベル数が与えられたときに、ラベルの接続数の最大値は(式2-1)で示すような比例関係にあるので、接続関係に基づいて暫定ラベルを直接に探索する方法をとれば、接続関係の数に比例する演算量でラベル分類処理を行うことができる。

図6に実際のラベル分類処理の方法を示す。ラベル分類処理は接続関係テーブルの作成と接続関係の探索の2つの処理から成る。

接続関係テーブル		ポインタ テーブル
1	2 5	2
2	1 3 5	3
3	2	1
4	6	1
5	1 2	2
...
L		0

(a) 接続関係テーブルとポインタテーブル

探索	接続 関係	スタ ック
I	1→5	5 ↓
II	5→2	2 ↓
III	—	5 —
IV	2→3	3 ↓
V	—	2 —
VI	3→2	— 2 ↑
VII	2→1	1 ↓



(b) ラベル接続関係の探索

図6 ラベル分類処理の改良

(i) 接続関係テーブルの作成

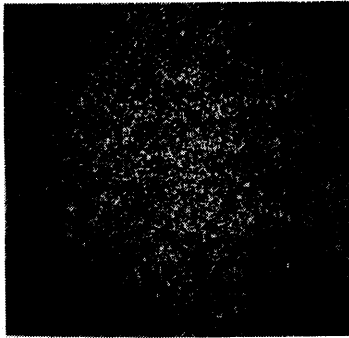
最初に重複を削除した接続関係を入力し、図6-(a)に示す接続関係テーブルとポインタテーブルを作成する。接続関係テーブルは縦方向が暫定ラベル、テーブルの内容はそれと直接に接続しているすべての暫定ラベルである。また、ポインタテーブルはそれぞれの暫定ラベルに接続しているラベルの総数を表す。このテーブルには、たとえば1と5が接続している場合には、1には5、5には1が接続しているというように往復の情報を記入することにより、双方向からのグラフ探索ができるように工夫した。

なお、このテーブルに記入される接続関係の合計は最大でも $6 \times L - 12$ であるのに対して、論理的なテーブルは $L \times L$ の大きさを必要とする。接続関係を効率的に格納し、テーブルのメモリ容量を削減するために、インプリメントにあたっては、この構成を索引型とした。

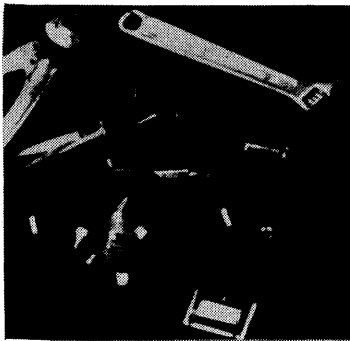
(ii) 接続関係の探索

次に接続関係テーブルの内容を参照しながら、直接的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する。探索の流れは以下の通りである。

- [ステップ1] 探索を始めるラベルを決定する。ラベルがない場合はラベル分類処理を完了する。スタックの初期化を行う。
- [ステップ2] 接続テーブルより内容を1つ読み出すとともに、対応するポインタテーブルの内容が空の場合はステップ4を実行する。
- [ステップ3] 読み出したラベルがスタックの最上位の内容(遷移前のラベル)と等しい場合はステップ2を実行する。それ以外の場合はそのラベルに遷移し、ステップ2を実行する。
- [ステップ4] スタックの内容を引き出し、遷移し、ステップ2を実行する。スタックが空の場合は探索を完了し、ステップ1を実行する。



(a) ランダム画像の一例



(b) 実画像

図7 テスト用画像

図6-(a)に示す接続関係テーブルの探索を行う場合の動作を図6-(b)に具体的に示す。

4. 評価実験

4.1 テスト画像

本方式の有効性を検証するために、図7に示す512×512の2種類のテスト用画像を用いて評価実験を行った。

・ランダム画像

0~1の一樣乱数を発生させ、暫定ラベル数が最大となるような閾値で2値化した後に、領域数を255に制限した画像(5種類)。

・実画像

テレビカメラより入力した8ビット/画素の画像(工具・部品)を2値化したもの。

a	b	c
d	e	f
g	h	i

ウィンド

入力									出力
a	b	c	d	e	f	g	h	i	
*	*	*	*	1	*	*	*	*	e = 1
*	*	1	0	0	1	*	1	1	
*	1	*	0	0	1	1	1	1	
*	1	*	1	0	0	1	1	1	
その他									e = 0

*: 0, 1

図8 前処理の演算論理

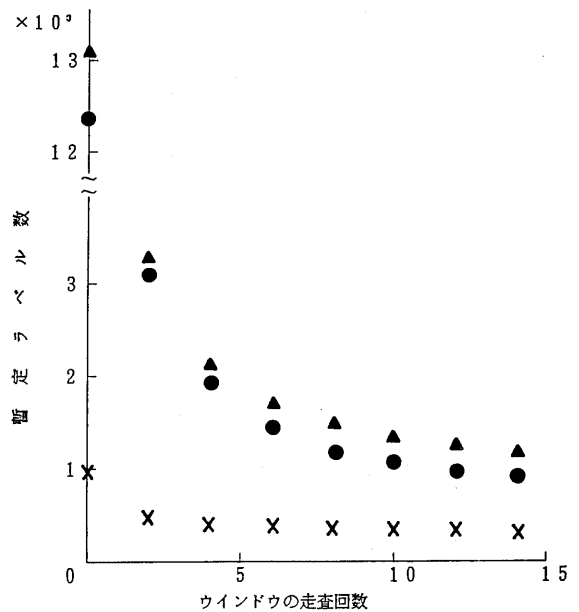


図9 ウィンドウの走査回数と暫定ラベル数の関係

4.2 接続関係の低減

接続関係の低減効果を暫定ラベル数を指標として判断した。図8に示す論理のウィンドウを複数回走査し、画像の整形を行った場合の暫定ラベル数の変化を図9に示す。Xは実画像、●はランダム画像(最小値)、▲はランダム画像(最大値)を示す。この結果より、走査回数の増加にともなって暫定ラベル数が減少し、ランダム画像では1/10、また実画像では1/3程度に削減されることが分かった。

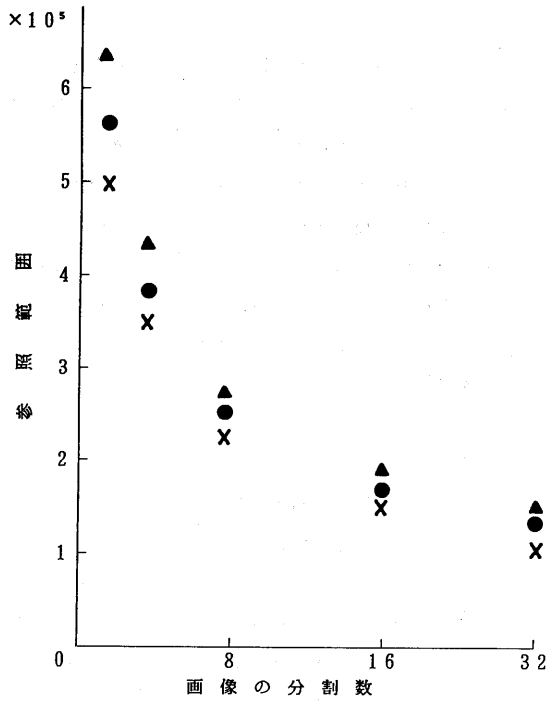


図10 画像の分割数と三角テーブルの参照範囲の関係

4.3 重複接続関係の削除

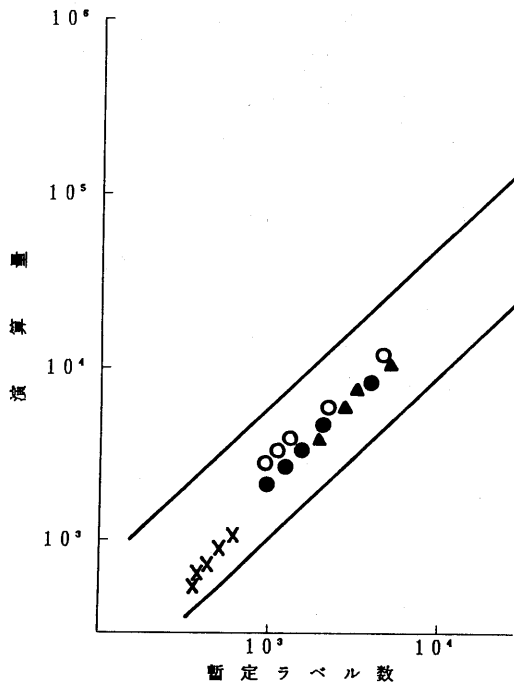
図10に画像の分割数と三角テーブルの参照範囲の関係を示す。×は実画像、●はランダム画像（最小値）、▲はランダム画像（最大値）を示す。分割数の増大にもなって参照範囲が減少する。

4.4 ラベル分類処理

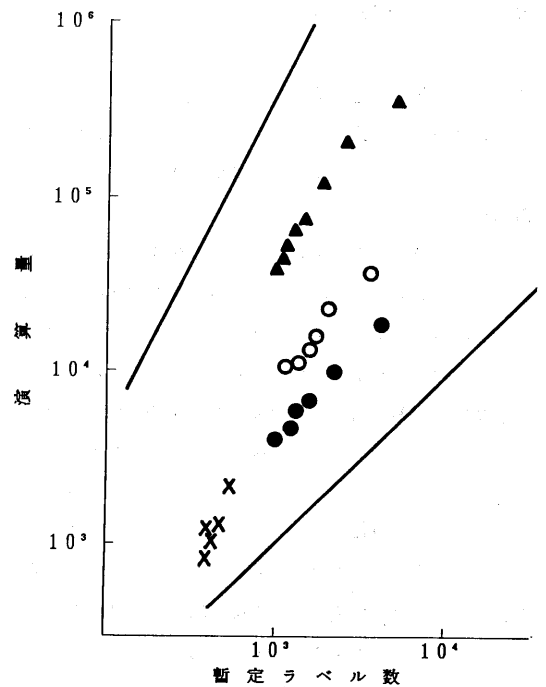
本方式と巡回リスト法について、暫定ラベルの分類処理に必要な演算量を実際の画像に適用した場合の比較結果を図11に示す。×は実画像、●はランダム画像（最小値）、○はランダム画像（中間値）、▲はランダム画像（最大値）を示す。図中の実線は演算量の理論的限界値を示し、演算量はこの範囲内におさまることになる。

なお、演算量の単位は、本方式に対してはあるラベル（頂点）から次のラベル（頂点）に遷移する回数、また巡回リスト法に対しては巡回リストの変更あるいは同一ラベル存在のチェックを行うために、リストの内容をアクセスする回数とした。

図11より、本方式では暫定ラベル数の増大にほぼ比



(a) 本アルゴリズム



(b) 巡回リスト法

図11 暫定ラベル数と演算量の関係

表1. パラメータの最適化

前処理回数	分割数	暫定ラベル数(部分画像内)		ラベル制限値	オーバーフロー発生確率(%)
		平均値	標準偏差		
6 9 12 15	8	200.4 166.1 148.9 138.7	16.7 18.1 19.5 19.9	256	11.7 ≒0 ≒0 ≒0
6 9 12 15	16	113.0 93.5 84.3 78.6	13.4 14.4 15.2 15.7	128	13.1 0.8 0.2 0.1

例して増大する。一方、巡回リスト法では暫定ラベル数の増加の1~2乗に比例して増加することを確認した。これら2つの方式の実際の演算量の比較は定義が異なるために、直接比較することはできないが、これらの処理を1命令1マシンサイクル(100ナノ秒/命令)で動作するマイクロプロセッサで実行させると仮定すると、本方式では1演算単位が約2.7マイクロ秒、巡回リスト法では約600ナノ秒となる。この条件から、両方式を比較すると、暫定ラベル数が100以上の場合に本方式が有利であることが分かった。

4.5 パラメータの最適化

ビデオレート画像処理システム「章駄天」への搭載を考慮し、ハードウェア化にともなう各種パラメータの最適化を行った。

各部分画像内で重複接続関係の削除を並列処理に基づいて実行するものとし、テーブルの容量および画像の分割数を決定するためにテスト画像(ランダム)を用いて行った実験結果を表1に示す。表1において、オーバーフロー確率は各部分画像内での暫定ラベルの平均値、分散およびテーブルの大きさから統計的に推定した値である。オーバーフロー確率を1/1000以下におさえた場合、ハードウェアの規模の点から、分割数が8、前処理回数が9回の条件で最適となることが分かった。第4.3節及び第4.4節の結果より、重複接続関係の削除を4多重で並列動作させた場合、この演算時間は1画像当たり24ミリ秒、ラベル探索の時間は32ミリ秒となる。いずれも、テレビ画像の走査時間(1/30秒)以内で実行できパイプライン処理を構成できることが分かった。

5. まとめ

本論文ではパイプライン処理を可能にする高速ラベリング・アルゴリズムについて述べた。ここでは、特に従来のラベリング・アルゴリズムにおいて、高速処理を阻害している要因を分析するとともに、これらの問題を解決するための各種手法を提案した。また、実験により本方式の有効性を確認するとともに、ハードウェア化にともなう各種パラメータの最適化を行った。これによって、ラベリングをビデオレートでパイプライン処理できる見通しを得た。

参考文献

- (1) 佐々木, 佐藤, 岩瀬, 後藤: "構造可変型ビデオレート画像処理システム「章駄天」", 情処学会コンピュータビジョン研究会資料, CV37-1 (1985)
- (2) Minsky, M. and Papert, S.: "Perceptron", 8.73, The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, and London, England, 1969.
- (3) Rosenfeld, A.: "Digital Picture Processing", pp. 347, Academic Press, New York, 1976.
- (4) 白井, 杉原: "連結領域を求める手法とそのハードウェア", 信学会画像工学研究会資料, IB78-9 (1978)
- (5) 奥山, 小林, 武長, 浅田, 藤原: "高性能ラベリングプロセッサのアーキテクチャ", 情処学会第31回(昭和60年後期)全国大会講演論文集, 5P-7, pp.1315-1316 (1985)
- (6) Harary, F. (池田貞雄訳): "グラフ理論", 共立出版 (1974)