

濃淡画像の連結成分を求める並列アルゴリズム

藤原 暁宏[†] 増澤 利光[†] 藤原 秀雄[†]

Parallel Algorithms for Gray-Scale Digitized Picture Component Labeling

Akihiro FUJIWARA[†], Toshimitsu MASUZAWA[†], and Hideo FUJIWARA[†]

あらまし デジタル画像上で連結成分を求める問題は、画像処理や画像認識の分野において基本的な問題である。これらの分野において、実際の処理の多くはカラー画像や濃淡画像を対象としているが、連結成分に関する既存の並列アルゴリズムの研究は、ほとんどが2値画像を対象としている。濃淡画像を対象とする連結成分としては、 k -有効連結成分という概念が知られており、本論文では、この k -有効連結成分の概念を用いた、極大連結成分問題、および連結成分分割問題を解く並列アルゴリズムを各々示す。入力は $n \times n$ の濃淡画像とする。極大連結成分問題に対しては、EREW PRAM上で、 $O(\log k \log n)$ 時間、 $n^2 / \log n$ プロセッサのアルゴリズムを示す。また、連結成分分割問題に対しては、EREW PRAM上で $O(\log^2 n)$ 時間、 n^2 プロセッサのアルゴリズム、Arbitrary CRCW PRAM上で $O((\log k + \log^* n) \log n)$ 時間、 n^2 プロセッサのアルゴリズムを示す。

キーワード 並列アルゴリズム, 画像処理, 連結成分, PRAM

1. ま え が き

デジタル画像上で連結成分を求める問題は、画像処理や画像認識において基本的な問題であり、並列アルゴリズムの分野でも近年多くの研究が行われている。その研究の多くは2値画像に関する研究であり [1], [6], [10], [13], PRAMモデルでは $n \times n$ の2値画像の連結成分を求める最適加速^(注1)な $O(\log n)$ 時間、 $n^2 / \log n$ プロセッサのアルゴリズムも提案されている [2]。

一方、現在画像処理や画像認識の分野において、実際に処理の対象となる画像の多くは、カラー画像や濃淡画像である。これらの画像を物質や物体ごとに連結領域に分割する処理のことを領域分割という。領域分割に関しては決め手となる手法がなくさまざまな方法が提案されているが、基礎的かつ重要な方法の一つとして、似通った特性（色や濃度など）をもつ連結領域を一つの物質若しくは物体と考えて分割を行う領域拡張法 (region growing) がある [12], [15], [16]。領域拡張法では、最初に画像を同じ特性をもつ連結領域に分割し、そののちに隣接する二つの連結領域について、

それらが類似した特性をもつならば合併を行うという操作を合併ができなくなるまで繰り返す。この領域拡張法において、合併を行う連結領域の類似性を表す指標として、濃度を考慮した連結成分である k -有効連結成分という概念が提案されている [3], [9]。 k -有効連結成分とは、連結成分に属する画素の濃度の最大値と最小値の差が k 以下である連結成分である。このような k -有効連結成分をもとにした領域拡張法による領域分割を、前述の2値画像に対するアルゴリズムを用いて効率よく求めることは困難である。

本論文では、 $n \times n$ の画素からなる濃淡画像（以下単に画像と呼ぶ）を入力とする、以下の二つの問題に対する並列アルゴリズムを示す。

- (1) 極大連結成分問題: 入力画像中の一つの画素が指定されたとき、その画素を含む極大な k -有効連結成分を求める問題。
- (2) 連結成分分割問題: 入力画像を k -有効連結成分に極大分割する問題。

極大連結成分問題は、指定された画素の濃度に近い濃度をもつ画素からなる極大連結成分を求める問題であり、 k を適当に指定することにより、画像からの指

[†] 奈良先端科学技術大学院大学情報科学研究科, 生駒市 Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630-01 Japan

(注1): 最適加速な並列アルゴリズムとは、プロセッサ数と時間計算量の積が、最速の逐次アルゴリズムの計算量とオーダー的に等しい並列アルゴリズムである。

定された画素を含む物体や物質の抽出に利用できる。また、連結成分分割問題は、入力画像の物体や物質への分割（領域分割）に利用できる。

この二つの問題は、いずれも逐次アルゴリズムでは、深さ優先探索の手法を用いることにより、 $O(n^2)$ 時間で解ける [16]。並列アルゴリズムでは、連結成分分割問題に対して、2次元メッシュの並列計算モデル上で $O(n)$ 時間、 n^2 プロセッサのアルゴリズム [9] が示されている。また、この問題に対する実際の並列計算機上のアルゴリズムの実験的評価も行われている [3], [5]。

本論文では、極大連結成分問題に対しては、EREW PRAM 上で、 $O(\log k \log n)$ 時間、 $n^2/\log n$ プロセッサのアルゴリズムを示す。また、連結成分分割問題に対しては、EREW PRAM 上で $O(\log^2 n)$ 時間、 n^2 プロセッサ、Arbitrary CRCW PRAM 上で $O((\log k + \log^* n) \log n)$ 時間^(注2)、 n^2 プロセッサのアルゴリズムを示す。

本論文の構成は以下のとおりである。2. では k -有効連結成分をはじめとするいくつかの定義を行う。3. では極大連結成分問題を解く並列アルゴリズムを示す。4. では濃淡画像の連結成分が満たす性質を示し、連結成分分割問題を解く並列アルゴリズムを示す。

2. 諸 定 義

D を $n \times n$ の濃淡画像とする (D は画素の集合である)。 D の各画素は $p = (i, j) (0 \leq i, j \leq n-1)$ と表す。各画素はそれぞれ非負整数の濃度をもつ。以下、画素 p の濃度を $d(p)$ と表す。

二つの画素 $p(i_0, j_0), p(i_1, j_1)$ に対して、 $|i_0 - i_1| + |j_0 - j_1| = 1$ ならば、この二つの画素は互いに隣接すると言う。

画素の集合を C とする。 C に含まれる任意の二つの画素 p, q に対し、 C に含まれる画素の列 $(a_0 (= p), a_1, \dots, a_m (= q))$ が存在し、 a_i と $a_{i+1} (0 \leq i \leq m-1)$ が隣接しているとき、 C を連結成分と言う。一つの画素のみからなる画素の集合も連結成分と言う。連結成分 C について $\max_C = \max\{d(p) | p \in C\}$, $\min_C = \min\{d(p) | p \in C\}$ とする。非負整数 k に対して、 $\max_C - \min_C \leq k$ が成り立つとき、 C を k -有効連結成分であると言う。定義より、0-有効連結成分は、同じ濃度をもつ画素からなる連結成分である。

$C \cap C' = \phi$ なる連結成分 C, C' について、画素 $p \in C$ と隣接する画素 $q \in C'$ が存在するとき、連

結成分 C と C' は隣接すると言う。このとき、集合 $C \cup C'$ は連結成分である。特に $C \cup C'$ が k -有効連結成分であるとき、 C' を C の k -隣接成分と言う (C は C' の k -隣接成分でもある)。また、 $C' = \{q\}$ のとき、連結成分 C と画素 q は隣接すると言う。

本論文の並列計算のモデルは EREW (Exclusive Read Exclusive Write) PRAM, および, Arbitrary CRCW (Concurrent Read Concurrent Write) PRAM である。PRAM は、複数のプロセッサが共有メモリの任意のメモリセルに対して1単位時間でアクセスを行うモデルである。複数のプロセッサが同期して一つのメモリセルに対して行うアクセスの制限により、PRAM はいくつかの種類に分類される。EREW PRAM では同じメモリセルに対して同時に複数のプロセッサがアクセスを行うことは禁止されている。CRCW PRAM では同じメモリセルに対する同時読出し、同時書込みが認められており、今回使用する Arbitrary CRCW PRAM では、同じメモリセルに対して同時に複数の書込みが行われた場合、任意の一つの書込みが成功するという仮定をもつ。

3. 極大連結成分問題

3.1 問題の定義

最初に、極大連結成分問題の定義を行う。

[定義1] (極大連結成分問題)

極大連結成分問題は、 $n \times n$ の画像 D 、 D に含まれる画素 s 、および、非負整数 k が与えられたとき、 s を含む極大な k -有効連結成分を求める問題である。つまり、以下の3条件を満たす連結成分 C を求める問題である。

- (i) $s \in C$.
- (ii) C は k -有効連結成分である。
- (iii) 任意の画素 $p \notin C$ について、 $C \cup \{p\}$ は k -有効連結成分ではない。 □

図1に極大連結成分問題の例を示す。各画素の中の数字は濃度を表す。また、太線で囲まれた画素を入力として与えられる画素 s とする。 $k = 10$ の場合、網線で囲まれた連結成分は極大連結成分問題の解である。

$k = 0$ ならば、極大連結成分問題の解は一意に定まるが、 $k > 0$ ならば、極大連結成分問題の解は一般に複数存在する。 $k = 0$ の場合の極大連結成分問題は、

(注2) : $\log^* n = \min\{t | \log^{(t)} n \leq 2\}$. ここで、 $\log^{(t)} n = \log \log^{(t-1)} n$, $\log^{(1)} n = \log n$ である。

38	38	40	43	42	42
31	34	37	37	47	47
31	32	32	34	47	47
38	38	14	44	44	47
11	11	14	14	45	45
11	10	10	10	10	10

図1 極大連結成分問題 ($k = 10$)
 Fig. 1 Maximum component labeling problem. ($k = 10$)

入力の画素と同じ濃度をもつ画素を1, その他の画素を0と考える2値化を行うことにより, 2値画像の連結成分を求めるアルゴリズム [2] を用いて容易に解くことができる. 従って, 以下では $k > 0$ の場合についてのみ考える.

3.2 アルゴリズム

ここでは 極大連結成分問題を解く EREW PRAM 上の並列アルゴリズムを示す. このアルゴリズムは, たかだか $\log k + 1$ フェーズからなる. アルゴリズムは, s を含む連結成分を保持するための変数 C をもち, 各フェーズでは連結成分 C を拡張していく. 第 m フェーズを最終フェーズとすると, 第 i フェーズ ($1 \leq i \leq m - 1$) では $\max_C - \min_C \geq (1 - \frac{1}{2^i})k$ となる k -有効連結成分 C を求める. アルゴリズム終了時の連結成分 C が, 求める極大な連結成分である.

[アルゴリズム 1]: (極大連結成分を求める)

入力: $n \times n$ の画像 D , 画素 $s \in D$, 非負整数 k

出力: s を含む極大な k -有効連結成分 C

(C を保持するデータ構造は, 各画素が C に属するか否かを表すサイズ $n \times n$ の論理型配列で実現する.)

方法:

- (1) $\max_C = \min_C = d(s)$ とする.
- (2) 以下のフェーズを繰り返す.
- (2.1a) 各画素 p について, その濃度 $d(p)$ により $d'(p)$ を求める.

$$d'(p) = 1 \left(\begin{array}{l} \lfloor \frac{\max_C + \min_C}{2} - \frac{k}{2} \rfloor \leq d(p) \\ \leq \lfloor \frac{\max_C + \min_C}{2} + \frac{k}{2} \rfloor \end{array} \right)$$

$$= 0 \quad (\text{それ以外})$$

(2.1b) 各画素の d' により, $n \times n$ の2値画像 D' が得られる. D' に対して, s の属する極大な0-有効連結成分 C を求める. $\max_C = \max\{d(p) | p \in C\}$, $\min_C = \min\{d(p) | p \in C\}$ とする.

(2.1c) $\max_C - \min_C = k$ ならば, C を求める極大連結成分として出力し, アルゴリズムを終了する.

(2.2) $\max_{old} = \max_C, \min_{old} = \min_C$ とする.

(2.3a) 各画素 p について, その濃度 $d(p)$ により $d'(p)$ を求める.

$$d'(p) = 1 \quad (\max_C - k \leq d(p) \leq \max_C)$$

$$= 0 \quad (\text{それ以外})$$

(2.3b), (2.3c) (2.1b), (2.1c) と同じ.

(2.4a) 各画素 p について, その濃度 $d(p)$ により $d'(p)$ を求める.

$$d'(p) = 1 \quad (\min_C \leq d(p) \leq \min_C + k)$$

$$= 0 \quad (\text{それ以外})$$

(2.4b), (2.4c) (2.1b), (2.1c) と同じ.

(2.5) $\max_{old} = \max_C$ かつ $\min_{old} = \min_C$ ならば, C を求める極大連結成分として出力し, アルゴリズムを終了する. □

3.3 正当性の証明

[補題 1] アルゴリズム 1 の (2) のフェーズ数はたかだか $\log k + 1$ である. つまり, アルゴリズム 1 は必ず停止する.

(証明)

(2) の第 i フェーズ終了時において, $W_i = \max_C - \min_C$ とする. まず, アルゴリズム 1 が第 i フェーズで終了しなければ,

$$W_i > (1 - \frac{1}{2^i})k$$

が成り立つことを, i に関する帰納法で証明する.

まず, 第1フェーズの実行を考える. アルゴリズム 1 は, 第1フェーズ実行中に終了するか, 終了せずに第2フェーズに進むかのいずれかである. 第2フェーズに進む場合には, (2.3b)若しくは(2.4b)で \max_C 若しくは \min_C が変更されている. (変更がなければ (2.5) でアルゴリズム 1 が終了する)

第1フェーズ終了時の W_1 を考える. 第1フェーズの (2.1b) では $\lfloor d(s) - \frac{k}{2} \rfloor$ 以上, $\lfloor d(s) + \frac{k}{2} \rfloor$ 以下の濃度をもつ画素からなる連結成分が求められる. 第1フェーズの (2.3b) 若しくは (2.4b) の実行により, \max_C , 若しくは, \min_C が変更されるので, $\lfloor d(s) - \frac{k}{2} \rfloor$ 未満の濃度をもつ画素, または, $\lfloor d(s) + \frac{k}{2} \rfloor$ より大きい濃度をもつ画素が C に加えられたことになる.

従って, $W_1 > \frac{k}{2}$ が成り立つ.

次に, 第 i フェーズの実行でアルゴリズム 1 が終了

しなかったとし、 $W_i > (1 - \frac{1}{2^i})k$ であるとする。このときの第 $i+1$ フェーズの実行を考える。

第 i フェーズ終了時の max_C, min_C の値をそれぞれ max_C^i, min_C^i と表す。帰納仮定より、 $max_C^i - min_C^i = W_i > (1 - \frac{1}{2^i})k$ が成り立つ。第 $i+1$ フェーズの (2.1b)

では $\lfloor \frac{max_C^i + min_C^i}{2} - \frac{k}{2} \rfloor$ 以上、 $\lfloor \frac{max_C^i + min_C^i}{2} + \frac{k}{2} \rfloor$ 以

下の濃度をもつ画素からなる連結成分が求められる。

第 $i+2$ フェーズに進む場合には、第 $i+1$ フェーズの (2.3b) 若しくは (2.4b) の実行により、 max_C 、若しくは min_C が変更されるので、(2.1b) では属すこと

のなかった、 $\lfloor \frac{max_C^i + min_C^i}{2} - \frac{k}{2} \rfloor$ 未満の濃度をもつ画

素、若しくは、 $\lfloor \frac{max_C^i + min_C^i}{2} + \frac{k}{2} \rfloor$ より大きい濃度をもつ画素が C に加えられることになる。

$\lfloor \frac{max_C^i + min_C^i}{2} + \frac{k}{2} \rfloor$ より大きい濃度をもつ画素が C

に加えられた場合、

$$\begin{aligned} W_{i+1} &\geq \lfloor \frac{max_C^i + min_C^i}{2} + \frac{k}{2} + 1 \rfloor - min_C^i \\ &> \frac{max_C^i + min_C^i}{2} + \frac{k}{2} - min_C^i \\ &= \frac{max_C^i - min_C^i}{2} + \frac{k}{2} \\ &= \frac{W_i}{2} + \frac{k}{2} \\ &> (\frac{1}{2} - \frac{1}{2^{i+1}})k + \frac{k}{2} \quad (\text{帰納仮定より}) \\ &= (1 - \frac{1}{2^{i+1}})k \end{aligned}$$

である。 $\lfloor \frac{max_C^i + min_C^i}{2} - \frac{k}{2} \rfloor$ 未満の濃度をもつ格子点

が C に加えられた場合も同様に $W_{i+1} > (1 - \frac{1}{2^{i+1}})k$ が示される。

以上より、 $W_i > (1 - \frac{1}{2^i})k$ が示され、 $W_{\log k} > k-1$ である。よってアルゴリズム 1 は、次のいずれかで終了する。

(i) (2) の $\log k$ フェーズ以内の実行において終了する。

(ii) (2) の $\log k + 1$ フェーズの実行により、 $W_{\log k+1} = max_C - min_C = k$ となり、(2.1c)、(2.3c)、(2.4c) のいずれかで終了する。

つまり、(2) のフェーズ数はただか $\log k + 1$ である。□

[補題 2] アルゴリズム 1 で求められる C は s を含む極大な k -有効連結成分である。

(証明)

C が s を含む k -有効連結成分であることはアルゴリズムから明らか。従って、以下ではその極大性のみを背理法により証明する。

ある画素 $p \notin C$ が存在し、 $C \cup \{p\}$ が k -有効連結成分であると仮定する。

アルゴリズム 1 は (2.1c)、(2.3c)、(2.4c)、(2.5) のいずれかで終了する。

アルゴリズム 1 が (2.1c) で終了したと仮定する。このとき $max_C - min_C = k$ である。また、 $C \cup \{p\}$ が k -有効連結成分であるので、 C と p は隣接し、 $min_C \leq d(p) \leq max_C$ である。従って、最後の (2.1a) の実行において、 $d'(p) = 1$ である。また、 $min_C = \min\{d(q) | q \in C\}$ 、 $max_C = \max\{d(q) | q \in C\}$ なので、任意の $q \in C$ について、最後の (2.1a) の実行において、 $d'(q) = 1$ である。従って、アルゴリズム 1 の最後の (2.1b) の実行により、 $C \cup \{p\}$ は一つの連結成分となる。これは仮定に矛盾する。

アルゴリズム 1 が (2.3c)、(2.4c) で終了した場合についても、同様に矛盾を導ける。

次にアルゴリズム 1 が (2.5) で終了したと仮定する。このとき、 $min_C \leq d(p) \leq max_C$ ならば、上と同様に p が C に含まれることが導かれるので、 $d(p) < min_C$ 若しくは $max_C < d(p)$ であると仮定する。

$d(p) < min_C$ であると仮定する。 $C \cup \{p\}$ が k -有効連結成分であるので、 $max_C - k \leq d(p)$ 、かつ、 p と C は隣接する。また、任意の $q \in C$ について、 $max_C - k \leq min_C \leq d(q) \leq max_C$ である。従って、最後の (2.3a) の実行により、 $d'(p) = d'(q) = 1$ が成り立ち、最後の (2.3b) の実行により $C \cup \{p\}$ は一つの連結成分となる。これは仮定に矛盾する。同様に、 $max_C < d(p)$ の場合も矛盾を生じる。

以上より、アルゴリズム 1 が (2.1c)、(2.3c)、(2.4c)、(2.5) のいずれかで停止した場合も、仮定に矛盾する。□

3.4 計算量の評価

[定理 1] $n \times n$ の濃淡画像、および、非負整数 k を入力とする極大連結成分問題は、EREW PRAM 上で $O(\log k \log n)$ 時間、 $n^2 / \log n$ プロセッサで解くことができる。

(証明)

以下、アルゴリズム 1 が EREW PRAM 上で $O(\log k \log n)$ 時間、 $n^2 / \log n$ プロセッサで実現できることを示す。

(1)は明らかに $O(1)$ 時間、1プロセッサで実行できる。

(2)の1フェーズの計算量を評価する。

(2.1a) 各プロセッサが max_C, min_C を知り、各画素の d' を計算するのは $O(\log n)$ 時間、 $n^2/\log n$ プロセッサで実行できる。

(2.1b) 2値画像の連結成分は、EREW PRAM上で $O(\log n)$ 時間、 $n^2/\log n$ プロセッサで求められる [2]。このアルゴリズムを適用することにより、同じ計算量で0-有効な連結成分を求めることができる。また、 C に属する画素の濃度の最大値、最小値も、同じ計算量で求められる [7]。

(2.1c) 一つのプロセッサで終了条件を検証し、それを $n^2/\log n$ 個のプロセッサに伝達する。この操作は2分木を利用したブロードキャストのアルゴリズム [11] により、 $O(\log n)$ 時間で行うことができる。次に、 C を参照し各画素について C に属するか否かを判定する。 C に属する画素については1のラベル、それ以外の画素については0のラベルを与える。この操作は各画素に関する局所的な演算のみなので、 $O(1)$ 時間、 n^2 プロセッサで実現できる。

(2.2) 明らかに $O(1)$ 時間、1プロセッサで実行できる。

(2.3a) ~ (2.3c), (2.4a) ~ (2.4c) 時間、プロセッサ数とも (2.1a) ~ (2.1c) に等しい。

(2.5) (2.1c) に等しい。

以上より、(2)の1フェーズの実行は $O(\log n)$ 時間、 $n^2/\log n$ プロセッサで実現できる。補題1より、(2)はただか $\log k + 1$ フェーズ繰り返されるので、アルゴリズム全体の計算量は $O(\log k \log n)$ 時間、 $n^2/\log n$ プロセッサである。 □

4. 連結成分分割問題

4.1 問題の定義

最初に、連結成分分割問題に関するいくつかの定義を行う。

[定義2] (画像の k -有効分割)

画像 D の連結成分の任意の集合 CS について、 D の任意の画素が CS のちょうど一つの連結成分に属するとき、 CS を D の分割と呼ぶ。つまり、 $CS = \{C_1, C_2, \dots, C_m\}$ が次の3条件を満たすとき、かつ、そのときに限り、 CS を D の分割と呼ぶ。

(i) 各 $C_i (1 \leq i \leq m)$ は D の連結成分。

(ii) $\bigcup_{i=1}^m C_i = D$ 。

38	38	40	43	42	42
31	34	37	37	47	47
31	32	32	34	47	47
38	38	14	44	44	47
11	11	14	14	45	45
11	10	10	10	10	10

図2 連結成分分割問題 ($k = 10$)
Fig.2 Maximal component labelings problem. ($k = 10$)

(iii) 任意の $i, j (1 \leq i < j \leq m)$ について、 $C_i \cap C_j = \phi$ 。

更に、非負整数 k に対して、画像の分割 CS に含まれる各連結成分 $C_i (1 \leq i \leq n)$ が k -有効であるとき、 CS を D の k -有効分割と呼ぶ。 □

[定義3] (連結成分分割問題)

連結成分分割問題は、 $n \times n$ の画像 D 、および、非負整数 k が与えられたとき、 D の極大な k -有効分割を求める問題である。つまり、以下の2条件を満たす D の分割 CS を求める問題である。

(i) CS は D の k -有効分割である。

(ii) 任意の異なる連結成分 $C_1, C_2 \in CS$ について、 $C_1 \cup C_2$ は k -有効連結成分ではない。 □

図2に連結成分分割問題の例を示す。各画素の中の数字は濃度を表す。 $k = 10$ の場合、異なる線で囲まれた三つの連結成分の集合が連結成分分割問題の解である。

極大連結成分問題同様、 $k = 0$ ならば、連結成分分割問題の解 CS は一意に定まるが、 $k > 0$ ならば、解 CS は一般に複数存在する。また、 $k = 0$ の場合の連結成分分割問題は、文献 [2] のアルゴリズムの手法を用いて、同じ計算量で容易に解くことができる。従って、以下では、 $k > 0$ の場合についてのみ考える。

4.2 グラフの定義

本節ではグラフを利用して連結成分分割問題を解くので、最初にグラフについて、いくつかの定義を行う。

グラフ G は空でない有限集合 V と、 V の相異なる二つの要素の非順序対の集合 E からなる。以下 $G = (V, E)$ と表す。 V の各要素を頂点と呼び、 E の各要素を辺と呼ぶ。頂点 a, b の非順序対である辺を (a, b) と表す。また、このとき頂点 a と b は隣接すると言い、辺 (a, b) は頂点 a, b に接続すると言う。

グラフ $G = (V, E)$ において、相異なる頂点からな

る系列 $\langle v_0, v_1, \dots, v_m \rangle$ が各 $i (0 \leq i \leq m-1)$ に対して、 $(v_i, v_{i+1}) \in E$ を満たすとき、この系列を v_0 から v_m への経路 (v_0 - v_m 経路) と言う。また、経路 $\langle v_0, v_1, \dots, v_k \rangle$ が存在し (但し、 $k \geq 2$)、 $(v_k, v_0) \in E$ であるとき、頂点系列 $\langle v_0, v_1, \dots, v_k, v_0 \rangle$ を閉路と言う。

グラフにおいて、任意の2頂点 u, v に対して u - v 経路が存在するならば、このグラフは連結であると言う。閉路がないグラフを森と呼び、閉路がない連結グラフを木という。特に根と呼ばれる頂点を指定した木を根つき木と言う。以下、根つき木を単に木という。 r を根とする木 T の任意の頂点を x とする。木では r - x 経路が一意に定まるが、 r - x 経路中の頂点 y について、 $(y, x) \in E$ ならば y を x の親と言い、 x を y の子と言う。木では根だけが親をもたない頂点である。子をもたない頂点を葉と言い、葉以外の頂点を内部頂点と言う。頂点集合が根と葉だけからなる木を星と呼ぶ。

平面上にどの辺も交差しないように埋め込むことができるグラフを平面グラフと言う。平面グラフについては以下の定理が成り立つ。

[定理2] [14] 平面グラフの頂点数が v 、辺数が e であるとき、 $v \geq 3$ ならば、

$$e \leq 3v - 6$$

が成り立つ。 □

4.3 濃淡画像の連結成分の性質

ここでは、濃淡画像の連結成分について成り立つ性質を示す。

[定義4] 画像の任意の分割を $CS = \{C_1, C_2, \dots, C_m\}$ とする。このとき CS の上の全順序関係“ \leq ”を以下のように定義する。ここでは CS の任意の二つの連結成分を C_i, C_j とし、一般性を失わず $\max_{C_i} \leq \max_{C_j}$ とする。

- (1) $\max_{C_i} < \max_{C_j}$ のとき、 $C_i \leq C_j$ 。
- (2) $\max_{C_i} = \max_{C_j}$ のとき、
 - $\min_{C_i} > \min_{C_j}$ ならば、 $C_i \leq C_j$ 。
 - $\min_{C_i} < \min_{C_j}$ ならば、 $C_j \leq C_i$ 。
 - $\min_{C_i} = \min_{C_j}$ のとき、 $i \leq j$ ならば $C_i \leq C_j$ 、 $i > j$ ならば $C_j \leq C_i$ 。 □

[性質1] 非負整数 k に対して、画像の任意の k -有効分割を CS 、 CS の任意の連結成分を C'_0 とする。また、 CS の任意の部分集合を $CA = \{C'_1, C'_2, \dots, C'_h\}$ とする。ここで、各 $i (1 \leq i \leq h)$ に対し、 $C'_i \leq C'_0$ 、かつ、 C'_i は C'_0 の k -隣接成分 (C'_i は C'_0 に隣接し、

$\max_{C'_0} - \min\{\min_{C'_0}, \min_{C'_i}\} \leq k$) であるとする。

このとき、 $C' = \bigcup_{j=0}^h C'_j$ は k -有効連結成分であり、

$\min_{C'} = \min\{\min_{C'_j} | 0 \leq j \leq h\}$ 、 $\max_{C'} = \max_{C'_0}$ である。 □

以下では、一つの連結成分とそれに隣接する一つ以上の連結成分を併合することにより、一つの連結成分を作る操作を、合併と呼ぶ。性質1は、合併により k -有効連結成分を得るための十分条件を示している。

4.4 EREW PRAM 上のアルゴリズム

4.4.1 アルゴリズム

ここでは、EREW PRAM上で連結成分分割問題を解く並列アルゴリズムを示す。このアルゴリズムは $O(\log n)$ のフェーズからなる。アルゴリズムは、画像 D の k -有効分割を保持するための変数 CS をもつ。第1フェーズ開始時は、 CS は0-有効極大分割を保持している。

各フェーズでは、 CS の各連結成分を一つの頂点とするグラフ (森) を作成し、 CS のいくつかの連結成分を合併し、新たな CS を作成する。以下にその概要を示す。

まず、 CS を頂点集合とし、 CS の各連結成分 C_1 に対し、 $C_1 \leq C'_1$ を満たす k -隣接成分 C'_1 が存在するならば、そのうち任意の一つを C_1 の親とする森 G_1 を考える。性質1より、 CS の任意の連結成分 C と G_1 における C のすべての子 (連結成分) を合併して得られる連結成分は k -有効である。また、互いに独立な合併は並列に実行できるので、 G_1 の頂点を隣接する頂点の色が異なるように彩色し、同じ色の連結成分に対して、合併を並列に実行する。

次に G_1 において接続する辺をもたなかった各連結成分 C_2 に対し、 C_2 の任意の一つの k -隣接成分 C'_2 を C_2 の親とする森 G_2 を考える。 G_2 は星と孤立頂点からなる森になるので (補題3)、星の根とその子との合併は、他の星の合併とは独立である。従って、各星について、星に属する子の濃度の最大値を求め、その最大値をもとに合併可能な子と根を並列に合併する。

以上の操作を、 CS のすべての連結成分に k -隣接成分がなくなるまで繰り返す。

以下に連結成分分割問題を解くアルゴリズム2を示す。

[アルゴリズム2]: (D の極大な k -有効分割を求める)

入力: $n \times n$ の濃淡画像 D 、非負整数 k

出力: D の極大な k -有効分割 CS

(CS を保持するデータ構造は、 $n \times n$ の整数型配列であり、同じ連結成分に属する画素に対して同じラベルを格納する。)

方法：

(1) D の 0-有効な極大分割を求め、 CS に格納する^(注3)。また、各連結成分 $C \in CS$ について、 $\min_C = \max_C = d(p)$ とする (p は C に属する任意の画素)。

(2) $C_1 \cup C_2$ が k -有効連結成分となる $C_1, C_2 \in CS$ が存在する限り、以下のフェーズを繰り返す。

(2.1a) 各連結成分 $C \in CS$ について、 $C \preceq C'$ を満たす k -隣接成分 C' が存在するならば、それらのうち任意の一つを選び (C_p とする)、 $P(C) = C_p$ とする。それ以外なら、 $P(C) = C$ とする。

各連結成分 $C \in CS$ について、 $P(C)$ が C の親を表すと考えると、 $G_1 = (CS, CE_1)$ は森である。ここで、 $CE_1 = \{(C, P(C)) | C \in CS, C \neq P(C)\}$ である。

(2.1b) 森 G_1 の各木の根の色を α とする。 G_1 を隣接頂点が同色にならないように α, β の 2 色に塗り分ける。

(2.1c) 色が α かつ子をもつ各連結成分 C と C のすべての子をもつ合併する (性質 1 より、合併によって新たに得られる連結成分は k -有効である)。つまり、 CS, G_1 を次のように更新する。但し、色が α かつ子をもつ連結成分の集合を $CS_\alpha (\subseteq CS)$ 、 G_1 における $C (\in CS_\alpha)$ の子の集合を $Ch(C)$ と表す。また、 C と C のすべての子の合併により得られる連結成分を $comb(C)$ と表す。つまり、 $comb(C) = C \cup (\bigcup_{C' \in Ch(C)} C')$ である。

(i) $CS = (CS - (CS_\alpha \cup (\bigcup_{C \in CS_\alpha} Ch(C)))) \cup \{comb(C) | C \in CS_\alpha\}$ とする。

(ii) 各連結成分 $comb(C)$ ($C \in CS_\alpha$ に対し、 $\min_{comb(C)} = \min\{d(p) | p \in comb(C)\}$ 、 $\max_{comb(C)} = \max_C$ とする。また、 $comb(C)$ の色を α, β 以外の色 δ にし、 $P(comb(C)) = comb(C)$ とする。

(iii) 各連結成分 $C \in CS_\alpha$ に対し、 $Ch(C)$ に属する連結成分を親としていた連結成分 (親としていた連結成分の合併によりできた連結成分も含む) C_g について、 $\max_{comb(C)} - \min_{C_g} \leq k$ ならば (C_g が $comb(C)$ の k -隣接成分ならば)、 $P(C_g) = comb(C)$ とする。それ以外なら $P(C_g) = C_g$ とし、 C_g の色を δ にする。

(2.1d) 色が δ かつ子をもつ連結成分 C と、 C のす

べての子を (2.1c) と同様に合併する。

(2.2a) 色が α, β のいずれかである各連結成分 $C \in CS$ について、 k -隣接成分が存在するならば、それらのうち任意の一つを選び (C_p とする)、 $P(C) = C_p$ とする。それ以外なら、 $P(C) = C$ とする。

各連結成分 $C \in CS$ について、 $P(C)$ が C の親を表すと考えると、 $G_2 = (CS, CE_2)$ は星若しくは孤立頂点からなる森である (補題 3)。ここで $CE_2 = \{(C, P(C)) | C \in CS, C \neq P(C)\}$ である。

(2.2b) 星の根の各連結成分 C と、 G_2 における C の子の部分集合を以下のように合併する (CS を以下のように更新する)。但し、星の根である連結成分の集合を $CS_r (= \{C \in CS | P(C) = C\})$ 、 G_2 における $C (\in CS_r)$ の子の集合を $Ch(C)$ と表す。

(i) 各連結成分 $C \in CS_r$ に対し、以下の操作を行う。

(A) $\max'_C = \max\{\max_{C'} | C' \in Ch(C) \cup \{C\}\}$ とする。

(B) $Ch_1(C) = \{C' \in Ch(C) | \min_{C'} \geq \max'_C - k\}$ とする。また、 $comb_1(C) = C \cup (\bigcup_{C' \in Ch_1(C)} C')$ とする ($comb_1(C)$ は k -有効連結成分である)。

(C) $\max_{comb_1(C)} = \max'_C$ 、 $\min_{comb_1(C)} = \min\{d(p) | p \in comb_1(C)\}$ とする。

(ii) $CS = (CS - (CS_r \cup (\bigcup_{C \in CS_r} Ch_1(C)))) \cup \{comb_1(C) | C \in CS_r\}$ とする。 □

4.4.2 正当性の証明

アルゴリズム 2 の (2.2a) で構成されるグラフ G_2 に関しては、次の補題が成り立つ。

[補題 3] アルゴリズム 2 の (2.2a) で求められるグラフ G_2 は星若しくは孤立頂点からなる森である。

(証明)

(2) の任意のフェーズについて考える。(2.1a) において、 $P(C) \neq C$ となる任意の連結成分 C は、(2.1c)、(2.1d) で他の連結成分と合併する (合併によって得られる新たな連結成分の色は δ) か、あるいは、 $P(C) = C$ となり、色が δ になる。従って、(2.2a) において、色が α または β である任意の連結成分 C は、(2.1a) において、 $P(C) = C$ となる連結成分である。つまり、 $C \preceq C'$ なる k -隣接成分 C' をもたない連結成分である。従って、(2.2a) において、色が α または β である任意の相異なる連結成分を C_1, C_2 ($C_1 \preceq C_2$) とする

(注 3) : CS の初期値を、一つの画素からなる連結成分の集合としても、このアルゴリズムは連結成分分割問題を解く。但し、同じ色の隣接画素が異なる連結成分に属する解を求めることがある。操作 (1) を加えることにより、同じ色の隣接画素は同じ連結成分に属する解を求める。

と, C_2 は C_1 の k -隣接成分ではない. 従って, (2.2a) において, $P(C_1) \neq C_2$, かつ, $P(C_2) \neq C_1$ が成り立ち, 色が α または β である連結成分の親は, 色が δ である. また, 色が δ である連結成分は, 親をもたない. このことから補題が証明される. \square

以下, アルゴリズム 2 が連結成分分割問題を解くことを示す.

[補題 4] アルゴリズム 2 の実行終了時の CS は D の k -有効極大分割を保持している.

(証明)

(i) CS が画像 D の分割である (D の任意の画素が CS のただ一つの連結成分に属する) ことを示す.

(1) の終了時, CS は 0-有効極大分割である.

(2.1c), (2.1d) において, 任意の連結成分 C_1, C_2 について, C_1 と $Ch(C_1)$ の合併, C_2 と $Ch(C_2)$ の合併が並列に起こったとする. このとき, $P(C_1) \neq C_2$, かつ, $P(C_2) \neq C_1$ が成り立つことを示せば十分である. (2.1c) では彩色の条件より, 明らかに成り立つ. また, (2.1d) では, 色が δ の連結成分 C について, $P(C) = C$ であるので, 明らかに成り立つ.

(2.2b) において, 任意の連結成分 C_1, C_2 について, C_1 と $Ch(C_1)$ の合併, C_2 と $Ch(C_2)$ の合併が並列に起こったとする. このときも, $P(C_1) \neq C_2$, かつ, $P(C_2) \neq C_1$ が成り立つことを示せばよい. これは G_2 が星もしくは孤立頂点からなる森である (補題 3) ことから, 明らかである.

(ii) CS の各連結成分が k -有効連結成分であることを示す.

(1) の終了時, CS は 0-有効極大分割である.

(2.1c), (2.1d) において, 合併により生じた新たな連結成分は, 性質 1 より k -有効連結成分である.

(2.2b) の合併によって生じた新たな連結成分が k -有効連結成分であるのは, 合併の方法より明らかである.

(iii) CS の極大性はアルゴリズム 2 の終了条件より明らかである. \square

アルゴリズム 2 の停止性を示す前に, 以下の補題を証明する.

[補題 5] アルゴリズム 2 の (2) の各フェーズにおいて, フェーズの開始時に, k -隣接成分をもつ連結成分 $C \in CS$ は, (2) の 1 フェーズの実行により, 次のいずれか一つを満たす.

(a) k -隣接成分と合併する.

(b) 合併はしないが, k -隣接成分の数が 1 以上減少する.

(証明)

背理法で証明する. アルゴリズム 2 の (2) の任意のフェーズについて考える. このフェーズ開始時に, k -隣接成分をもち, このフェーズにおいて, k -隣接成分と合併をせず, かつ, k -隣接成分の数も減少しなかった連結成分 C が存在すると仮定する. この C について場合分けを行い, (2) の実行を検証する.

(i) (2.1a) において, $P(C) = C_1(C \neq C_1)$ と仮定する ($C \subseteq C_1$ を満たす k -隣接成分 C_1 が存在すると仮定する). このとき C は, (2.1c), (2.1d) で, 他の連結成分と合併するか, あるいは $P(C) = C$ となる. $P(C) = C$ となるのは, C_1 が他の連結成分と合併し, この合併によって新たに得られた連結成分が C の k -隣接成分ではない場合である. このとき C の k -隣接成分の数は少なくとも 1 減少する. これは仮定に矛盾する.

(ii) (2.1a) において, $P(C) = C$ と仮定する. このとき C は, (2.1c), (2.1d) で他の連結成分と合併されず, (2.1d) 終了時の C の色は α または β である. (2.1c), (2.1d) で C の k -隣接成分の数が減少しなければ, (2.2a) で C は k -隣接成分をもち, $P(C) = C_2(C_2 \neq C)$ となる. このとき, (2.2b) で, C は C_2 と合併するか, C_2 が他の連結成分と合併し, この合併で得られた連結成分は C の k -隣接成分でなくなる (つまり, C の k -隣接成分が一つ減る) か, のいずれかである. どちらの場合もこれは最初の仮定に矛盾する. \square

[補題 6] アルゴリズム 2 の (2) のフェーズ数は $O(\log n)$ である. つまり, アルゴリズム 2 は必ず停止する.

(証明)

画像 D の分割 CS について, グラフ $G = (CS, CE)$ を考える. 但し, $CE = \{(C_1, C_2) | C_1 \text{ は } C_2 \text{ の } k\text{-隣接成分}\}$ とする. このとき G は明らかに平面グラフである. (2) の第 i フェーズ開始時の G の辺の数を e_i とする. 但し, 第 i フェーズが最終フェーズの場合, 第 i フェーズ終了時の G の辺の数を e_{i+1} とする. 第 i フェーズにおいて, 補題 5 の (a) を満たす連結成分の数を n_i , (b) を満たす連結成分の数を m_i とする. 一つの合併において, a 個の連結成分が合併すれば, G の辺は少なくとも $a-1$ 個減少する. 一つの合併には少なくとも二つの連結成分が必要なので, 補題 5 の (a) を満たす連結成分の数を n_i とすると, 合併により G の辺は少なくとも $\frac{1}{2}n_i$ 個減少する. このことから,

次式が成り立つ。

$$\begin{aligned} e_{i+1} &\leq e_i - \frac{1}{2}n_i - m_i \\ &\leq e_i - \frac{1}{2}n_i - \frac{1}{2}m_i \\ &\leq e_i - \frac{1}{2}\left(\frac{1}{3}e_i + 2\right) \quad (\text{定理 2 より}) \\ &= \frac{5}{6}e_i - 2 \end{aligned}$$

$e_1 \leq 2n(n-1)$ であるので、上の式を解くことにより、(2)のフェーズ数は $O(\log n)$ である。□

補題 4, 6 より、アルゴリズム 2 は連結成分分割問題を解く。

4.4.3 計算量の評価

[定理 3] $n \times n$ の画像の連結成分分割問題は EREW PRAM 上で $O(\log^2 n)$ 時間、 n^2 プロセッサで解くことができる。

(証明)

各連結成分を表す CS のラベルとしては、その連結成分の代表画素の座標を用いる。 $max_C, min_C, P(C)$, 連結成分の色 はそれぞれ $n \times n$ の配列を用意し、各連結成分の代表画素に適当な値を格納することにより実現する。

以下、それぞれのステップの計算量を示す。

(1) 既知の 2 値画像の連結成分を求めるアルゴリズム [2] を応用することにより、 D の 0-有効極大分割は $O(\log n)$ 時間、 $n^2/\log n$ プロセッサで求められる。各連結成分の min_C, max_C は、[4] のアルゴリズムを用いて、画素を連結成分のラベルと濃度をキーとしてソートすることにより、 $O(\log n)$ 時間、 n^2 プロセッサで求められる。(以下ソートを行う場合はすべて [4] のアルゴリズムを使用する)

(2) については、まず、1 フェーズの計算量を評価する。なおフェーズの終了判定は、 $O(\log n)$ 時間、 n^2 プロセッサで行うことができる。

(2.1a) 各連結成分 $C \in CS$ について、 $C \preceq C'$ を満たす C の k -隣接成分 C' から任意の一つを選択するのは、各連結成分の境界上の画素について隣接画素を探索することにより k -隣接成分が存在するかどうかを確認し、条件にあう隣接連結成分の中から一つの連結成分を選ぶことによって実現される。これは境界上の画素を k -隣接成分の有無、連結成分のラベル、および、画素の座標をキーとしてソートし、同じ連結成分の中で最も小さい画素番号をもつ画素の隣接連結成分を親とすることにより、 $O(\log n)$ 時間、 n^2 プロセ

ッサで実行できる。

(2.1b) 各連結成分について根からの距離を $O(\log n)$ 時間、 n^2 プロセッサで求め [11], 求めた距離が偶数か奇数かにより 2 色に彩色を行う。

(2.1c) 合併の操作をそれぞれ検証する。

(i) 連結成分 C と $Ch(C)$ の合併によって得られる連結成分 $comb(C)$ の代表画素は、 C の代表画素とする。従って、 $Ch(C)$ の連結成分の各画素に対し、その連結成分のラベルを C のラベルに更新すればよい。 $Ch(C)$ の各連結成分の代表画素は $O(1)$ 時間で C のラベルを知ることができる。 $Ch(C)$ の代表画素以外の画素については、画素を合併前のラベルをキーとしてソートし、ポインタジャンプのアルゴリズム [11] により、代表画素のラベルの伝達を行えば、 $O(\log n)$ 時間、 n^2 プロセッサで実行できる。

(ii), (iii) $min_{comb(C)}$ は、画素をラベルと濃度をキーとしてソートすることにより、 $O(\log n)$ 時間、 n^2 プロセッサで求められる。その他の操作は $O(\log n)$ 時間、 n^2 プロセッサで簡単に実行することができる。

(2.1d) (2.1c) に等しい

(2.2a), (2.2b) それぞれ、(2.1a), (2.1c) に等しい
従って、(2) の 1 フェーズは $O(\log n)$ 時間、 n^2 プロセッサで実行できる。

補題 6 より、(2) のフェーズ数は $O(\log n)$ であるので、補題は示される。□

4.5 Arbitrary CRCW PRAM 上のアルゴリズム

4.5.1 アルゴリズム

ここでは、Arbitrary CRCW PRAM 上で連結成分分割問題を解く並列アルゴリズムを示す。このアルゴリズムは前述の EREW PRAM 上の並列アルゴリズム (アルゴリズム 2) を改良したものである。このアルゴリズムでは、以下のような手段を用いて、アルゴリズム 2 の (2) では $O(\log n)$ 時間であった 1 フェーズの計算量を、 $O(\log k + \log^* n)$ 時間としている。

(i) アルゴリズム 2 の (2.1b) では、 $O(\log n)$ 時間で森を 2 色に彩色した。Arbitrary CRCW PRAM 上においては、森を 2 色に彩色する効率のよいアルゴリズムは知られていないが、3 色に彩色する効率のよい並列アルゴリズムは既知であるので [8], このアルゴリズムを利用する。但し、このアルゴリズムでは、各木の根の色が α になるようには彩色できないので、根の色が β, γ である連結成分を根とした星の合併も行う。

(ii) アルゴリズム 2 の (2.1c) では、各連結成分

$comb(C)$ の最小の濃度 $min_{comb(C)}$ を, ソートにより $O(\log n)$ 時間で求めている. Arbitrary CRCW PRAM 上では $min_{comb(C)}$ の取り得る値はただか k 種類であることを利用し, $O(\log k)$ 時間で $min_{comb(C)}$ を求める.

以下に Arbitrary CRCW PRAM 上のアルゴリズムを示す

[アルゴリズム 3]:

入力, 出力: アルゴリズム 2 に同じ.

方法:

- (1) ~ (2.1a) アルゴリズム 2 に同じ.
 (2.1b) 森 G_1 を隣接頂点が同色にならないように α , β , γ の 3 色に塗り分ける.
 (2.1c) アルゴリズム 2 に同じ.
 (2.1d) 色が β かつ子をもつ連結成分 C と, C のすべての子を (2.1c) と同様に合併する.
 (2.1e) 色が γ かつ子をもつ連結成分 C と, C のすべての子を (2.1c) と同様に合併する.
 (2.1f) 色が δ かつ子をもつ連結成分 C と, C のすべての子を (2.1c) と同様に合併する.
 (2.2a) 色が α , β , γ のいずれかである各連結成分 $C \in CS$ について, k -隣接成分が存在するならば, それらのうち任意の一つを選び (C_p とする), $P(C) = C_p$ とする. それ以外なら, $P(C) = C$ とする.
 (2.2b) アルゴリズム 2 に同じ. □

アルゴリズム 2 と同様にして, アルゴリズム 3 の正当性は示される.

4.5.2 計算量の評価

[定理 4] $n \times n$ の画像の連結成分分割問題は Arbitrary CRCW PRAM 上で $O((\log k + \log^* n) \log n)$ 時間, n^2 プロセッサで解くことができる.

(証明)

各変数を格納するデータ構造はアルゴリズム 2 と同じものを使用する. アルゴリズム 3 の (1) は $O(\log n)$ 時間, n^2 プロセッサで, (2) のフェーズの終了判定は $O(1)$ 時間, n^2 プロセッサで行うことができる. 以下, (2) の 1 フェーズの計算量を評価する.

(2.1a) 明らかに, $O(1)$ 時間, n^2 プロセッサで実行できる.

(2.1b) 文献 [8] のアルゴリズムを適用することにより, $O(\log^* n)$ 時間, n^2 プロセッサで実行できる.

(2.1c) 合併の操作をそれぞれ検証する.

(i) $O(1)$ 時間, n^2 プロセッサで実行できる.

(ii), (iii) $min_{comb(C)}$ を求める計算量を示す.

$min_{comb(C)}$ の取り得る値が k 種類であるので, 2 分探索の手法により, $min_{comb(C)}$ を $O(\log k)$ 時間, n^2 プロセッサで求めることができる. その他の操作については, $O(1)$ 時間, n^2 プロセッサで実行できる.
 (2.1d) ~ (2.1f) (2.1c) に等しい.

(2.2a), (2.2b) それぞれ, (2.1a), (2.1c) に等しい.

以上より, (2) の 1 フェーズは $O(\log k + \log^* n)$ 時間, n^2 プロセッサで実行できる. □

5. む す び

本論文では, 濃淡画像の連結成分に関する並列アルゴリズムを示した. 極大連結成分問題については, EREW PRAM 上で $O(\log k \log n)$ 時間, $n^2 / \log n$ プロセッサのアルゴリズムを示した. また, 連結成分分割問題については, EREW PRAM 上で $O(\log^2 n)$ 時間, n^2 プロセッサ, Arbitrary CRCW PRAM 上では $O((\log k + \log^* n) \log n)$ 時間, n^2 プロセッサのアルゴリズムを示した.

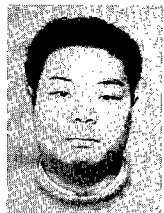
今後の課題としては, k に依存しない高速なアルゴリズムを提案することが考えられる. また, 濃淡画像の連結成分について, 今回扱った k -有効連結成分だけでなく, 異なる定義による連結成分に関する問題も考えられる.

謝辞 本研究に対し, 御助言, 御討論頂いた奈良先端科学技術大学院大学の井上智生助手および日立製作所の吉田大輔氏に深く感謝致します.

文 献

- [1] A. Agrawal, L. Nekludova, and W. Lim, "A parallel $O(\log n)$ algorithm for finding connected components in planar images," Proc. 1987 International Conference on Parallel Processing, pp.783-786, Aug. 1987.
- [2] H. M. Alnuweiri and V. K. Prasanna, "Parallel architectures and algorithms for image component labeling," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.14, no.10, pp.1014-1034, Oct. 1992.
- [3] D. A. Bader, J. JáJá, D. Harwood, and L. S. Davis, "Parallel algorithms for image enhancement and segmentation by region growing with an experimental study," Technical Report CS-TR-3449 and UMIACS-TR-95-44, UMIACS and Electrical Engineering, University of Maryland, May 1995.
- [4] R. Cole, "Parallel merge sort," SIAM Journal of Computing, vol.17, no.4, pp.770-785, 1988.
- [5] N. Coptly, S. Ranka, G. Fox, and R.V. Shankar, "A data parallel algorithm for solving the region growing problem on the connection machine," Journal of Parallel and Distributed Computing, vol.21, pp.160-168,

- 1994.
- [6] R. E. Cypher, J. L. C. Sanz, and L. Snyder, "Practical algorithms for image component labeling on SIMD mesh connected computers," Proc. 1987 International Conference on Parallel Processing, pp.772-779, Aug. 1987.
 - [7] A. Gibbons and W. Rytter, "Efficient Parallel Algorithms," Cambridge University Press, 1988.
 - [8] A. V. Goldberg, S. A. Plotkin, and G. E. Shannon, "Parallel symmetry-breaking in sparse graphs," Proc. 9th Annual ACM Symposium on Theory of Computing, pp.315-324, 1987.
 - [9] S. Hambrusch, X. He, and R. Miller, "Parallel algorithms for gray-scale digitized picture component labeling on a mesh-connected computer," Journal of Parallel and Distributed Computing, vol.20, no.1, pp.56-68, 1994.
 - [10] W.-J. Hsu, L.-R. Wu, and X. Lin, "Optimal algorithms for labeling image components," Proc. 1990 International Conference on Parallel Processing, vol. III Algorithms and Applications, pp.75-82, 1990.
 - [11] J. JáJá, "An Introduction to Parallel Algorithms," Addison-Wesley Publishing Company, 1992.
 - [12] A. Rosenfeld and A. C. Kak, "Digital Picture Processing," Academic Press, 1982.
 - [13] H. Shi and G. X. Ritter, " $O(n)$ -time and $O(\log n)$ -space image component labeling with local operators on SIMD mesh connected computers," Proc. 1993 International Conference on Parallel Processing, pp.III 98-101, 1993.
 - [14] M. ベザット, G. チャートランド, L. L. ホスター, "グラフとダイグラフの理論," 共立出版, 1981.
 - [15] 谷内田正彦, "人工知能シリーズ11 ロボットビジョン," 昭晃堂, 1990.
 - [16] 田村秀行, "コンピュータ画像処理入門," 総研出版, 1985.
(平成7年2月27日受付, 9月7日再受付)



藤原 暁宏 (学生員)

平5阪大・基礎工・情報卒。平7奈良先端大学院博士前期課程了。現在、同大学院博士後期課程在学中。並列アルゴリズム、並列計算モデルなどの研究に従事。IEEE, 情報処理学会会員。



増澤 利光 (正員)

昭57阪大・基礎工・情報卒。昭62同大学院博士課程了。工博。同年同大情報処理教育センター助手。同大基礎工学部助教授を経て、平6奈良先端大・情報科学助教授、現在に至る。平5コーネル大学客員教授(文部省在外研究員)。分散アルゴリズム、並列アルゴリズムに関する研究に従事。ACM, IEEE, EATCS, 情報処理学会各会員。



藤原 秀雄 (正員)

昭44阪大・工・電子卒。昭49年同大学院博士課程了。同大・工・電子助手, 明治大・工・電子通信助教授, 情報科学教授を経て, 現在奈良先端大・情報科学教授。昭56年ウォータールー大客員助教授。昭59年マッギル大客員準教授。論理設計論, フォールトトレランス, 設計自動化, テスト容易化設計, テスト生成, 並列処理, 計算複雑度に関する研究に従事。著書「Logic Testing and Design for Testability」(MIT Press)など。大川出版賞, 情報処理学会会員, IEEE Fellow。