

画像処理用超並列プロセッサ AMP における 連結成分のラベル付けとその評価

1 C-4

山元規靖[†] 鶴田直之[†] 谷口倫一郎[†] 雨宮真人[†]
[†]九州大学総合理工学研究科

1 まえがき

著者らは、画像処理のための柔軟で高能率な並列処理を実現する自律型非同期超並列プロセッサ AMP^[1]とそのプログラミング言語 Valid-A^[2]を開発している。AMP 上の各 PE(Processor Elements) は、外部メモリ(共有メモリ)を持たず、非同期に動作する。PE 間の結合は、格子状のネットワークを採用している。AMP では画像の早期処理から画像理解のレベルまで種々の処理を実行させることを設計の基本方針としている。

二値画像における連結成分(図形)のラベル付けは、以後の画像理解を行なううえで非常に重要な処理である。本稿では、AMP 上での連結成分のラベル付けの並列アルゴリズムとそのインプリメント、および AMP のソフトウェアシミュレータ上でのラベル付け処理の評価について述べる。

2 並列プロセッサにおける連結成分のラベル付け

多数の PE を持つ並列マシン上での画像処理では、その期待される並列性から、各画素に対する処理ごとに PE に割り付ける方法が考えられる。連結成分のラベル付けでは、連結成分を構成する画素を導出し、各連結成分に固有の番号(ラベル)を割り振らねばならない。この場合、最も単純なラベル付けの手法としては、以下の二つの手法が考えられる。ここでは、各画素が唯一固有の番号(例えば、画素座標)を持っているとする。

Algorithm A 連結成分の周辺をトレースすることにより連結領域を求め、トレースした周辺を構成する画素のうち任意の画素の番号をその連結成分のラベルとする。

Algorithm B 連結成分を一画素に縮退させることにより連結領域を求め、縮退した画素のうち任意の画素の番号をその連結領域のラベルとする。

Algorithm A の手法は、各連結成分間では並列処理される。しかし、周辺のトレースは逐次処理であるため、 $n \times n$ 画素の入力画像に対して最悪の場合(連結成分が画像全体に広がる複雑な領域である場合) $O(n^2)$ の処理が必要である。

Algorithm B においても、各連結成分および縮退は並列に処理されるが、複雑な連結領域の場合やはり $O(n^2)$ の処理が必要である。

並列マシン上での効率の良いラベル付けのためには、上記の手法中の逐次処理の部分をしてできるだけ並列に処理しな

ければならない。そこで、一般的には次に述べる divide-and-conquer の手法を用いて並列にラベル付け処理を行なう^[3]。

まず、入力画像($n \times n$ 画素)をいくつかの細領域に分割し、各細領域ごとにラベル付けを行なう。(図1参照。)この得ら

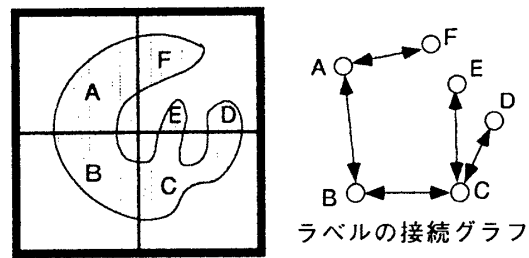


図1: divide-and-conquer によるラベル付け

れた局所ラベル(図1中の A, B など)をさらに大きな領域へと階層的にマージしていくことにより、完全な画像全体のラベル付けを行なうことができる。この手法では、 $\sqrt{n} \times \sqrt{n}$ 個の PE を用いて $O(n)$ での処理が可能である。

以下では、divide-and-conquer によるラベル付けの AMP へのインプリメントと評価について述べる。

3 AMP における連結成分のラベル付け

AMP において上記の divide-and-conquer に基づいたラベル付けを行なう際、各細領域では連結領域の周辺を細領域の境界画素からトレースする。(図2参照。)これにより細領域

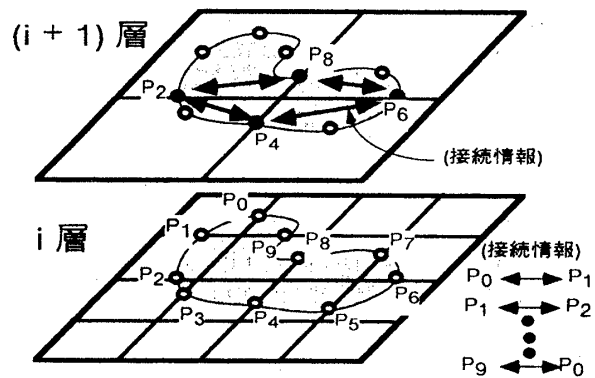


図2: AMP におけるラベル付けアルゴリズム

内の連結成分の周辺の接続関係が求められる。さらに、この

Labeling Connected Components on Massively Parallel Image Processor AMP

Noriyasu YAMAMOTO, Naoyuki TSURUTA, Rin-ichiro TANIGUCHI, Makoto AMAMIYA

Department of Information Systems, Graduate School of Engineering Sciences, Kyushu University

隣接する4つの細領域を新たに1つの領域として、4つの細領域で求めた周辺の接続情報を基に新たな領域での接続情報を求める。以上の操作を階層的に繰り返すことにより最後には画像全体での連結成分の周辺の接続情報が得られ、ラベル付けを行なうことができる。

次に、 $n \times n$ 画素の入力画像に対し、 $m \times m$ 個の PE を用いるとした場合のこの手法の手間を考える。

まず、各 PE に割り付けられる $l \times l$ 画素について連結成分周辺のトレースを行なう。ここで、 $l = (n/m)$ とする。この処理は、各 PE において逐次処理となるので $O(l^2)$ の手間がかかる。

連結領域の周辺トレースは、 $\log m$ 層行なわれ、 $1 \geq i \geq \log m$ とすると、 i 層の細領域はそれぞれ $kl \times kl$ 個の画素を持つ。(ここで、 $k = 2^i$ とする。) また、各細領域には k^2 個の PE が含まれる。連結成分の周辺をトレースする際の最大 PE 間通信距離は k^2 であり、この場合細領域の境界を構成する1つの PE から発するトレースは最大 $(l/2)$ 本である。したがって、 i 層における処理の手間は $O(k^2 + (l/2))$ である。よって、階層全体の手間は、次のようになる。

$$\sum_{i=1}^{\log m} (2^{2i} + \frac{n}{2m}) = O(m^2)$$

したがって、ラベル付け全体の手間は、

$$O(m^2 + \frac{n^2}{m^2})$$

となる。この式より、divide-and-conquer を用いたラベル付けのアルゴリズムは、 $m = c\sqrt{n}$ のとき最小値 $O(n)$ となる。ただし、 c は定数である。

4 評価

表1は、3種類の入力画像における前述のラベル付けアルゴリズムの AMP での実行時間をシミュレートした結果を示す。ここで、divide-and-conquer による手法と比較して挙

表1: 種々の入力画像における Labeling 処理時間

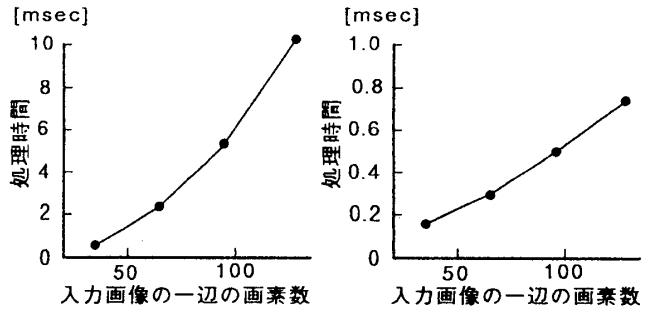
入力画像 (128x128)	Algorithm A	divide-and-conquer approach
	0.67 (55.2%)	0.16 (87.1%)
	0.20 (81.5%)	0.12 (83.6%)
	10.14 (12.6%)	0.77 (65.1%)

PE数: 256 個 単位: [msec] ()内はFU稼働率

げている Algorithm A の結果は、前述した $O(n^2)$ のアルゴリズムの実行結果である。当然のことながら、複雑あるいは大きい連結成分を持つ入力画像では divide-and-conquer によるアルゴリズムが高速である。

図3は、入力画像の大きさに対するラベル付けアルゴリズムの処理時間を表1でも示した Algorithm A と divide-and-conquer による手法について比較したグラフである。こ

で、入力画像は、表1で用いた渦巻型の画像である。使用した PE 数は、前述の各画像の大きさに対する最適数より多いため、(b) の divide-and-conquer によるアルゴリズムの場合、 $O(n)$ よりも多少処理時間が長くなっている。(a) の Algorithm A の場合は、やはり $O(n^2)$ の処理時間となっていることがわかる。



(a) Algorithm A (b) divide-and-conquer approach

図3: 入力画像の大きさに対する処理時間

5 むすび

本稿では、AMP 上での連結成分のラベル付けの並列アルゴリズムとそのインプリメントについて述べ、AMP のソフトウェアシミュレータ上でのラベル付け処理の実行時間の評価について述べた。

AMP では、divide-and-conquer によるラベル付けの手法により $n \times n$ 画素の入力画像を n 個の PE を用いて $O(n)$ で処理できる。AMP において、この手法は、複雑で大きな連結成分を持つ画像に対しても有効であることを示した。divide-and-conquer による手法は、ラベル付けのみならず、多角形認識^[4]など他の処理においても $O(n)$ で処理を実行するために有効な方法である。

本稿で示した divide-and-conquer による手法において、各階層で実際に処理を実行している PE は、細領域の境界に位置する PE のみで、他の PE はデータの転送のみを行なっている。今後は、これらの使用していない PE に負荷を分散することにより、さらに効率の良い処理の実現が考えられる。

参考文献

- [1] 谷口, 雨宮: “画像処理と理解のための自律型非同期超並列プロセッサ AMP”, 「画像理解の高度化と高速化」シンポジウム講演論文集, pp.53-58(1989).
- [2] 山元, 鶴田, 谷口, 雨宮: “画像処理用超並列プロセッサ AMP のプログラミングと性能評価について”, 情報処理学会論文誌, Vol.32, No.7, pp.933-940(1991).
- [3] D. Nassimi and S. Sahni, “Finding connected components and connected one on a mesh-connected parallel computer”, SIAM Journal on Computing, vol. 9, No. 4, 1980, pp.744-757.
- [4] 松尾, 山元, 鶴田, 谷口, 雨宮: “画像処理用超並列プロセッサ AMP における図形認識法とその評価”, 情報処理学会第44回全国大会講演論文集(2), pp.255-256(1992).