

1次元プロセッサレイによる逐次型画像処理アルゴリズムの並列化について

4N-7

許 昭倫

NEC 情報メディア研究所

1 はじめに

多数のPEを一次元に結合しSIMD (Single Instruction Multiple Data) の原理で動作するLPA (Linear Processor Array) 型の画像処理システムは、フィルタリングに代表されるような局所並列型の処理に対しては非常に効率よく動作するが、画素の処理に順序性がある場合等は、システムが持つ並列性を処理にうまく生かさないという問題点が存在する。これまで、個別に解決策が提案された例はあるが、一般性を持った方法は提案されていない。本稿では、そうした逐次型の画像処理に対しLPAによる一般的な並列化手法を提案する。

2 画像処理の処理形式による分類

画像処理は処理の対象画素の範囲によって、全画素を処理する全面型と一部分の画素のみを処理する領域型、また画素間の処理順序によって、各画素に対する処理を並列に行える並列型と画素間での処理順序に依存関係が存在する制約型とに分類できる。そしてこれらの組合せを考えると4つの処理形式、すなわち全面並列型、全面制約型、領域並列型、領域制約型とに分類できる。

全面並列型の例にラプラシアンのようなフィルタリング処理、全面制約型の例にランレングス符号化や全面ラスタスキャンによるラベリング、領域並列型の例にモルフロジー演算やSnakesのような弛緩法による演算、そして領域制約型の例に細線化、輪郭線追跡等がある。

3 各処理形式のLPAによる並列化

前記4つの処理形式の内、全面並列型は、従来のLPAの行単位方式(図1左)による実行で十分に並列化できる。ここでは全面制約型の処理に対しシストリック方式、そして領域並列型、領域制約型の処理に対しては仮想スタック方式を中心とする実行方式を提案する。

Efficient Implementation of Sequential Type Image Processing Algorithms on Linear Processor Arrays
Sholin Kyo
Information Technology Research Laboratories
NEC Corporation

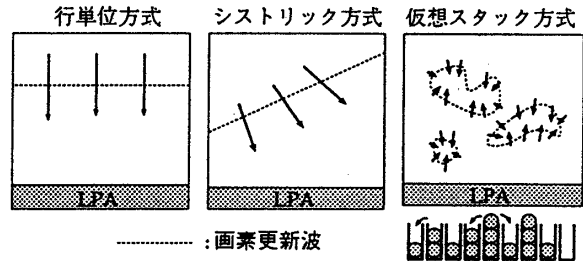


図1: 各処理形式に対するLPAによる並列化方式

3.1 シストリック方式

全面制約型では、各画素の処理順序は他の画素に依存して決定されるが、多くの場合それは静的に決まる。そうした画素同士の依存関係は再帰マスク(図2)の形で記述できる。シストリック方式とは、この再帰マスクが課す制約を満足するために、LPA上の各PEをシストリックに、すなわちあるタイムディレイのもとで次々と起動するように制御する方式である。その結果、タイムディレイの値に応じた角度を持った斜め方向への画素更新波が得られる(図1中、再帰マスクA使用時)。タイムディレイの大きさは通常再帰マスクのサイズに依存し、サイズが $(2M+1) \times (2M+1)$ の再帰型マスクの場合、タイムディレイはMとなる。本方式を用いることで $N \times N$ の画像に対し、MがNより充分に小さい場合、全面制約型の処理はN個のPEを有するLPAのもとで $O(N)$ の時間で処理される[1]。

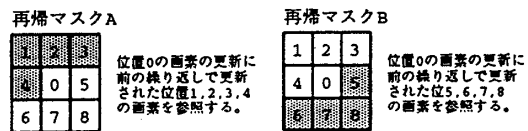


図2: 再帰マスクの例

3.2 仮想スタック方式

領域型の処理では、どの画素が処理すべき画素かは、画像の内容に応じ動的に決まる。仮想スタック方式(図1右)では、そうした動的な要素を扱えるようにするため、発火可能となった画素へのポイントをその都度貯

めておくためのスタックを各PE上に設ける。処理は、まず処理の開始点となる画素へのポインタを各PEのスタックに投入することによりスタックの初期化を行う。その後、各PEは自分のスタックが空でなければそこから画素へのポインタを取り出し、それが指す画素に対し所定の処理を行った後、当該画素の所定近傍に存在する画素が所定の発火条件を満たすかどうかを調べ、満たす場合はそれらの画素へのポインタを、それらの画素の処理を担当PEのスタックに投入する。それを、全PEのスタックが空になるまで繰り返す。こうしたスタックを用いた画素の処理の順序付けにより、各PEは同一の処理を複数の領域に対し重畳的に施すことが可能となる。

仮想スタック方式を用いた場合、もっとも多くの画素を処理する必要のあったPEが全体の処理時間を支配する。しかし通常、サイズ $N \times N$ の画像に対し N 個のPEを有するLPAを用いた場合、各PEは最大で N 個の画素に対し処理を行えばよいため、多くの場合 $O(N)$ の処理時間で処理が終了する。

3.3 両方式の評価および領域サイズの影響

領域型処理で領域のサイズが大きい場合、全面型と見なし処理を行うことが可能である。例えば領域制約型の処理で静的に画素の処理順序が決まり、かつ領域サイズが大きい場合では、仮想スタック方式とシストリック方式のどちらを使用することも考えられる。領域並列型の場合でも、シストリック方式と行単位方式を読み替えば、同じことがいえる。

表1に、領域型であり画素同士の依存関係が静的に決まる処理の一つである距離変換¹[2]を、両方式それぞれによって実現した場合の、画像内の領域サイズの変動に対する実行時間の変化を調べた結果を示す。領域サイズが小さい場合、シストリック方式では背景画素への無駄な処理によるロスが相対的に顕著となる。一方領域サイズが大きい場合、仮想スタック方式では発火すべき画素数が増え、負荷最多PEへの負担が増加し、処理時間全体が伸びる結果となる。なお、画像サイズは 256×256 、LPAはIMAP-VISION²[3]、プログラムはLPA用並列C言語1DC[1]で記述した。ワークス

¹2値画像の各前景画素位置に、それへのもっとも近い背景画素への距離を書き込む処理

²256PE(8bit,40MHz動作)構成のワンボード画像処理システム

テーション³上で全画素を逐次的に処理した場合の処理時間は約500msで、動作周波数の違いを考慮すれば、どちらの手法でも単一PEの場合と比べ170~280倍の性能を実現していることになる。

手法	画像1	画像2	画像3	画像4
仮想スタック方式	10.1ms	7.9ms	6.1ms	4.8ms
シストリック方式	8.0ms			

表1: 領域サイズの異なる画像に対する処理時間

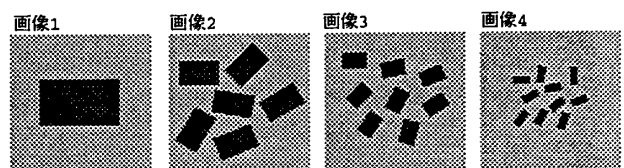


図3: 画像1~4

4 まとめ

各処理形式の画像処理のLPAによる並列化の方式をまとめると以下ようになる。

全面並列型 従来の行単位方式で十分に並列化される。

全面制約型 シストリック方式を用いる。

領域並列型 領域サイズが小さい場合は仮想スタック方式、大きい場合は行単位方式が有効である。

領域制約型 領域サイズが小さい場合は仮想スタック方式、領域サイズが大きかつ画素間の依存関係が静的な場合はシストリック方式が有効である。

5 おわりに

本稿では、LPAによる逐次型画像処理の並列化方式として、シストリック方式および仮想スタック方式を提案した。また両方式を実機上で評価した結果、高い並列性抽出効果が得られることを確認した。

参考文献

- [1] 許, 佐藤: 1次元プロセッサアレイ用データ並列言語1DCによる画像処理アルゴリズムの記述とそのコンパイラ, 情報処理学会報告(SwoPP96),1996.
- [2] G. Borgefors: *Distance Transformations in Digital Images*, Computer Vision, Graphics, and Image Processing, Vol.34, pp.344-371, 1988.
- [3] 藤田 他,10GIPS IMAP-VISION ボード—ハードウェア—, 信学春季大会, (1996).

³MicroSparc,110MHz,C言語使用