

視覚装置における位置／大きさ検出アルゴリズムの検証

味岡義明[†] 嶋田雅洋^{††} 天野英晴^{††}

[†] (株) エッチャンデス 〒443-0057 愛知県蒲郡市中央本町 12-7

^{††} 慶應義塾大学理工学研究科天野研究室

〒223-8522 神奈川県横浜市港北区日吉 3-14-1

E-mail: tajioka@cyberdoc.co.jp, ††{masahiro,hunga}@am.ics.keio.ac.jp

あらまし 我々人間は容易に交差点で信号機を見付けたり、街中でトイレマークを見付けることができるのは、網膜に1億個以上ある視細胞が検知した150度以上に渡る広角多帯域動画の中から、それらしい物体だけを数百ミリ秒程度で選り出した後、その物体を1秒程度注視することにより、その物体を識別しているからである。このような脳の視覚機能をモデル化したものが視覚装置であり、MIMD型格子状アーキテクチャと局所並列画像処理アルゴリズムによって特徴付けられる。そこで現在我々のプロジェクトは、視覚装置を搭載した人工眼球を用いて盲導犬ロボットを開発するために、この視覚装置のLSI化と、開発環境の整備を実行中である。本稿では、視覚装置の中心的な役割を果たす位置／大きさ検出アルゴリズムについて説明した後で、データの転送量についての評価を行なう。

キーワード 視覚装置

Size/Position detection algorithm on The Visual Device

Yoshiaki AJIOKA[†], Masahiro SHIMADA^{††}, and Hideharu AMANO^{††}

[†] Ecchandes Inc, 12-7, Chuohommachi, Gamagori, Aichi, 443-0057, Japan

^{††} Faculty of Science and Technology, Keio University

3-14-1, Hiyoshi, Kohoku-ku, Yokohama, 223-8522, Japan

E-mail: tajioka@cyberdoc.co.jp, ††{masahiro,hunga}@am.ics.keio.ac.jp

Abstract We, human beings, can find signals at intersections and lavatory marks easily because we can recognize an object by gazing at it for about one second after selecting only some suitable objects in some hundred milliseconds in a wide and multi-band animation image detected by more than 100 million cones and rods in a retina, whose view angle is more than 150 degrees. A visual device is a model of such a visual mechanism of our brains, characterized by a MIMD-type array architecture and some local and parallel image processing algorithms. In order to develop a seeing-eye robot with an artificial eye carrying the visual device, now, our project is making an LSI of the visual device and its development environment. In this report, we estimate amount of transmit data after explaining about position/size detection algorithm that plays an important role in the visual device.

Key words Visual Device

1. はじめに

携帯用カメラとLSI技術の発達により、人間の視覚機能の一部を代行することにより、視覚障害者の日常生活をサポートするシステムの開発が可能になっている。

筆者らの一人により提案された視覚装置[1][2]は、人間の視覚に代って、ノイズの多い街中で、信号機やトイレマークなどの形状認識を行うシステムである。視覚装置は、従来のビジョンチップや並列画像処理で用いられている計算コストを必要とする厳密な画像処理アルゴリズムに代って短時間でそこそこの結果を得ることのできる局所並列画像処理アルゴリズムを2次元格子構造を持つ単純なMIMD型並列アーキテクチャ上で実行する。このことでノイズを多く含む画像情報から高速かつ低消費電力で物体の大まかな形状認識を可能とする。

本報告では、視覚装置の概略と共に、局所並列画像処理アルゴリズムの代表として位置/大きさ検出アルゴリズムを紹介する。さらに、2次元格子状MIMD型並列アーキテクチャ上で実行した場合の性能およびハードウェアコストを評価することで、その有効性を明らかにする。

2. 視覚装置

2.1 局所並列アルゴリズム

並列画像処理の研究は60年代から続けられており、様々なアルゴリズムと並列処理手法が提案されている[3][4][5][6][7]。一方、これらの画像処理アルゴリズムを基にして、1970年代中頃にはビジョンチップの開発も始まり[8]、人工網膜LSI[9]やS³PE[10]が開発されている。これらの、並列画像処理アルゴリズムの多くは、4近傍ないし8近傍の画像データを参照し、フーリエ変換、アフィン変換、ハフ変換などの画像処理アルゴリズムを並列に処理することで性能の向上を図っている。

これに対して、視覚装置は、パン、チルト及びロールをする移動カメラが、人間の眼球と同じように物体を探索する状況を想定する。すなわち、探索に際してはカメラを移動させたり、物体の切り出しに際しては、ズームをして物体を拡大することが可能であることを前提とする。また、人間の視覚能力を目標としているため、比較的移動速度の速くない対象物体に対して、大まかな認識を低電力、低コストで行うことを狙っている。

最終的に視覚装置は、ノイズの多い世界において、人間の視細胞の数と同等の1億画素以上の入力画像

に対して、50MHz程度のクロック周波数を用いて、画像中の多数の物体の初期探索を約33ミリ秒、一定条件を満たす物体の大まかな形状認識を約330ミリ秒、1個の物体の種別認識を約1秒で実現することを目標としている。

上記の状況の基で、この目標を達成するために開発されたアルゴリズムが局所並列画像処理アルゴリズムである。このアルゴリズムは、従来の並列画像処理と異なり、厳密な数値計算をする代りに、単純な数式と局所的なデータ交換に基づき、そこそこの結果を短時間で得ることを目的としている。

例えば、画像中の複数の物体の大まかな位置及び大きさを特定する場合、従来は水平方向及び垂直方向に対してヒストグラムを計算する方法が頻繁に用いられてきた[11]。この方法は、物体の形状が幾何的でないとしても、2、3個程度の濃淡が明瞭な物体に対して、画像中の全ての画素をスキャンすることにより濃淡のしきい値を計算し、各々の物体の大まかな位置及び大きさを導出することができる。しかし、従来の手法は、(1)物体の数や大きさに関係なく、画素数が増えるほど計算量が増える。(2)物体の色をしきい値によって背景の色から区分しなければならぬ。(3)集合写真のように、物体の配置によっては全く機能しない。そのため、本来ならば位置と大きさが特定された物体に対してテンプレートとマッチングをするべきところを、物体の位置と大きさを特定するために、画像を多数の領域に分割し、さらにこれらの領域のヒストグラムを、1つずつテンプレートとマッチングする方法が研究されている[12]。

これに対し視覚装置は、8近傍以内の関数だけから構成されるエッジ情報生成アルゴリズムを用いて、しきい値を計算することなく、カラー画像から粗エッジ情報を生成する。この際、ノイズ自身の性質を用いてノイズを低減することができる。さらに数10近傍以内の関数から構成されるエッジ情報形成アルゴリズムを用いて、粗エッジ情報を、より明瞭な形成エッジ情報に変換する。そして、一定の近傍以内の関数から構成される位置/大きさ検出アルゴリズムを用いて、全ての物体の位置及び大きさを並列に計算し、一定時間内に全ての計算を終了する。

また、画像中の複数の物体の大まかな幾何図形を特定する場合、従来はフーリエ変換、アフィン変換及びハフ変換が頻繁に用いられてきた。例えば画像中から人物の顔を探索する場合、物体のエッジ情報を生成した後、ハフ変換によってこのエッジ情報が円(楕円)図形であるか識別し、それにより顔であるか

どうか判定する方法が用いられる[13]。ハフ変換は、エッジ情報の数と等しい数の連立方程式を解いて近似解を見付ける方法で、エッジ情報が増えると計算量は膨大となる。このため、物体の大きさをさらに幾つかに区分するなどして、計算量の低減を図っているが、それでも多くの計算量を必要とする。

一方、視覚装置は、一定の近傍以内の関数から構成される位置／大きさ検出アルゴリズムを変形した幾何解析アルゴリズムを用いて、画像中の位置に関係なく一定以内の大きさの多数の物体の形状を並列に計算し、一定時間内に全ての計算を終了することができる。しかも図形の種類が○△□－＋ＴＹＶ＊など簡単な図形であれば、図形を事前に指定しなくても動作する。

上記の他にも、色情報生成アルゴリズム、物体／背景分離アルゴリズム、位置／大きさ／傾き検出アルゴリズム、領域正規化アルゴリズム、パターンマッチングアルゴリズムなど主立った視覚装置用のアルゴリズムが開発済みである。

本報告では、このうち位置／大きさ検出アルゴリズムを紹介するが、それに先だってアルゴリズムを動作する視覚装置のアーキテクチャの概観を述べる。

2.2 視覚装置のアーキテクチャ

局所並列処理のアルゴリズムの実行には、格子状に配列された論理的な配列演算ユニット(AOU)から構成されるMIMD型格子状アーキテクチャを想定する。各々のAOUは、少なくとも1帯域画素値を有する1画素を処理する論理的な単位であり、隣接のAOUに対して非同期にデータをやり取りすることができる。アルゴリズムの記述は、このAOUを必要量想定して行う。ただし、処理する画像によっては、AOUが膨大な数となるため、 $m \times m$ 個のAOUを1つにまとめて、仮想配列演算ユニット(VAOU)を構成する。視覚装置が局所処理を行う性質上、このVAOUは、あたかも近傍サイズが約 $1/m^2$ になり、さらに帯域画素値の数が m^2 倍されたAOUのように振る舞う。そこで実際に視覚装置をLSIに実装する場合には、このVAOUをプロセッシングエレメント(PE)に割り付ける。各PEは、プロセッサ、メモリ及び少なくとも1個のコントローラを有し、個々のコントローラが隣接のPEに備えられた相異なる1個のコントローラと非同期に通信する。

3. 位置／大きさ検出アルゴリズム

位置／大きさ検出アルゴリズムは、図1に示すように、図形の輪郭を表すエッジ情報から構成される

エッジ情報画像を入力した後で、これらのエッジ情報をおおよその重心位置に集めることにより、この図形のおおよその重心位置の画素と、図形を表すエッジ情報の数(これが図形の大きさを表す重複情報となる)を求めるものである。

従来の典型的な手法では、全ての帯域画素値を1回だけ参照することにより、水平方向と垂直方向のヒストグラムを計算する方が用いられてきた[?]。しかし、この方法では、多数の物体の重心位置を同時に検出することができない。

そこで、ここでは、参照したエッジ情報が指し示す重心方向に向かってエッジ情報を移動させることにより、全てのエッジ情報がおおよその重心位置に集まることを利用し、各々のAOUが一定の近傍サイズ内のAOUのエッジ情報を参照することのみで、並列におおよその重心位置とエッジ情報を求める局所並列アルゴリズムを提案する。

以下ではまず、図1で示す関数及びオペレータの詳細について説明する。

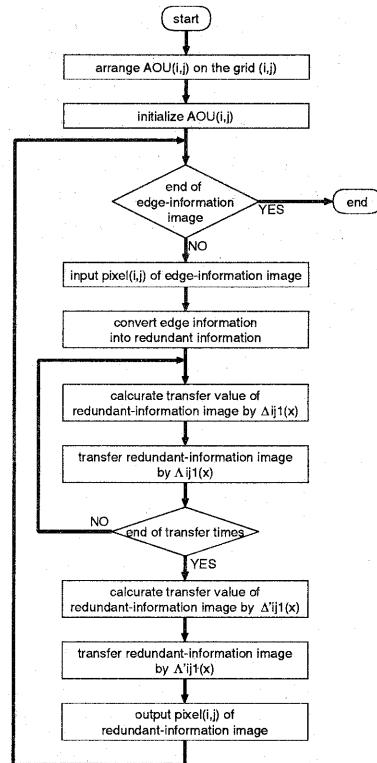


図1 位置／大きさ検出アルゴリズム。

3.1 アルゴリズムの説明

a) 画像の表記

幅 w 、高さ h 、帯域数 b の任意の 2^n 階調画像を x

及び $\underline{y}$ とする。 $\underline{x}$ 及び $\underline{y}$ は、それぞれ AOU に割り付けられており、その位置 $p(i, j, k)$ の帯域画素値 x_{ijk} 及び y_{ijk} を用いて数式 1 及び 2 のように表される。なおアンダーラインが付された文字はベクトルを示す。また n は非負の整数、 w, h, b, i, j, k は自然数である。

$$\underline{x} = \{x_{ijk} | x_{ijk} \text{ is value at } p(i, j, k), \\ 1 \leq i \leq w, 1 \leq j \leq h, 1 \leq k \leq b\} \quad (1)$$

$$\underline{y} = \{y_{ijk} | y_{ijk} \text{ is value at } p(i, j, k), \\ 1 \leq i \leq w, 1 \leq j \leq h, 1 \leq k \leq b\} \quad (2)$$

画像 $\underline{x}$ を二値画像に変換する場合、3 に従って帯域画素値を二値化する。

$$\Phi_{ijk}(\underline{x}) = \begin{cases} 1 & \text{if } x_{ijk} > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

b) 近傍の画像の表現

さて、画像の位置 $p(i, j, k)$ における AOU 上での q 近傍の位置の集合 $P_{ijk}(q)$ は数式 4 によって表される (図 2)。ただし近傍サイズ q は 4、8、24、48、80、120、 $(2l+1)^2 - 1$ と続く数列であり、近傍長 l は自然数である (図 3)。なお画像サイズをはみ出した位置が集合 $P_{ijk}(q)$ に含まれる場合には、特に指定がない限り、画素値が 0 に相当し、しかも画像に含まれない架空の位置を代用する。これにより辺縁処理は自動的に行われる。したがって集合 $P_{ijk}(q)$ の要素の数 N_{ijk} は常に q となる。

$$P_{ijk}(q) = \begin{cases} \{p(i+1, j, k), p(i, j+1, k), \\ p(i-1, j, k), p(i, j-1, k)\} \\ \text{if } q = 4, \\ \{p(l, m, k) | i-r \leq l \leq i+r, \\ j-r \leq m \leq j+r, \\ p(l, m, k) \neq p(i, j, k)\} \\ \text{if } q = (2r+1)^2 - 1. \end{cases} \quad (4)$$

c) 画像の移動

さて、各々の AOU は、近傍サイズ内の重複情報を用いて重心方向を計算し、自らが有する重複情報を 8 近傍のうち重心方向に対応する AOU に移動させる。以下、画像の移動の方法について述べる。この処理は全て画像データをそのままの形で扱うことが可能で、しかも割り算などハードウェア量や計算時間が掛る処理は必要としない。

位置 $p(l, m, o)$ の l, m, o 各々の値を画像データとして帯域画素値に変換するオペレータを $\sharp$ とし、変換

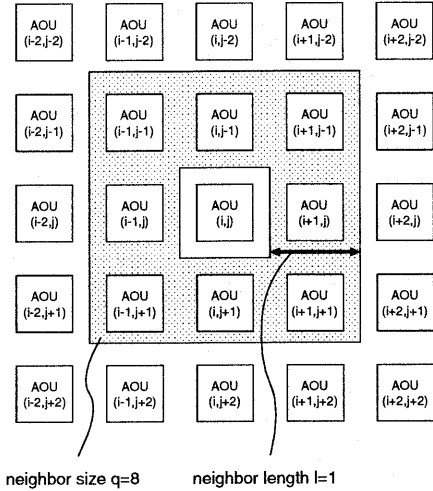


図 2 配列演算ユニットに対する近傍。

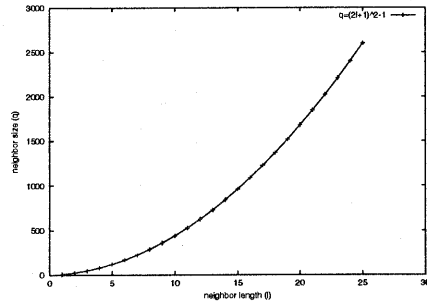


図 3 近傍長と近傍サイズの関係。

された帯域画素値を $\sharp p(l, m, o)$ とする。次に帯域画素値が位置 $p(i, j, k)$ から位置 $p(i+l, j+m, k+o)$ へ移動する場合を考える。このとき帯域画素値の移動量は位置 $p(l, m, o)$ として表されるものとする。つまり移動量はある位置からのベクトルと見なすことができる。最後に帯域画素値から位置を取り出すオペレータを $\sharp^{-1}$ とする。したがって $\sharp^{-1} \sharp p(l, m, o) = p(l, m, o)$ となる。

画像 $\underline{x}$ があり、画像 $\underline{x}$ が単帯域二値画像である場合、画像 $\underline{x}$ の位置 $p(i, j, 1)$ における重心位置への移動量は数式 5 に従い計算される。なお、本来重心を計算するには除算を行う必要があるが、8 近傍内への移動量を計算する際に除算は相殺されてしまうので、数式 5 では除算が省かれている。

$$G_{ij1}(\underline{x}) = p \left(\sum_{p(l, m, 1) \in P_{ij1}(q)} (l-i)x_{lm1}, \sum_{p(l, m, 1) \in P_{ij1}(q)} (m-j)x_{lm1}, 0 \right) \quad (5)$$

$$\Theta(p(i, j, k)) = \begin{cases} \#p(1, 0, k) & \text{if } i > 0, |j| < |i|/2, \\ \#p(1, -1, k) & \text{if } i > 0, j < 0, \\ & |i|/2 \leq |j| \leq 2|i|, \\ \#p(0, -1, k) & \text{if } j < 0, 2|i| < |j|, \\ \#p(-1, -1, k) & \text{if } i < 0, j < 0, \\ & |i|/2 \leq |j| \leq 2|i|, \\ \#p(-1, 0, k) & \text{if } i < 0, |j| < |i|/2, \\ \#p(-1, 1, k) & \text{if } i < 0, j > 0, \\ & |i|/2 \leq |j| \leq 2|i|, \\ \#p(0, 1, k) & \text{if } j > 0, 2|i| < |j|, \\ \#p(1, 1, k) & \text{if } i > 0, j > 0, \\ & |i|/2 \leq |j| \leq 2|i|, \\ \#p(0, 0, k) & \text{otherwise.} \end{cases} \quad (6)$$

$$\Theta'(p(i, j, k)) = \begin{cases} \#p(1, 0, k) & \text{if } i > 0, \\ & |j| < |i|/2, \\ \#p(1, 0, k) & \text{if } i > 0, \\ & j < 0, |i|/2 \leq |j| \leq 2|i|, \\ \#p(0, 1, k) & \text{if } i < 0, j > 0, \\ & |i|/2 \leq |j| \leq 2|i|, \\ \#p(0, 1, k) & \text{if } j > 0, 2|i| < |j|, \\ \#p(1, 1, k) & \text{if } i > 0, j > 0, \\ & |i|/2 \leq |j| \leq 2|i|, \\ \#p(0, 0, k) & \text{otherwise.} \end{cases} \quad (7)$$

移動量 $p(i, j, k)$ から、数式 6 及び 7 に従い 8 近傍内への移動量を計算し、移動量画像に画像化する。なお数式 7 は、画像の離散化により数式 6 では対応しきれない場合にのみ利用する。

したがって数式 5、6 及び 7 を用いると、数式 8 及び 9 に従い、単帯域二値画像 $\underline{x}$ の重心方向への移動量画像の帯域画素値を簡単に記述することができ

る。なお移動量画像の帯域数は 1 となる。

$$\Delta_{ij1}(\underline{x}) = \Theta(G_{ij1}(\underline{x})) \quad (8)$$

$$\Delta'_{ij1}(\underline{x}) = \Theta'(G_{ij1}(\underline{x})) \quad (9)$$

2 つの画像 $\underline{x}$ 、 $\underline{y}$ があり、画像 $\underline{y}$ が移動量画像である場合、数式 10 に従い、画像 $\underline{y}$ で指し示された移動位置に画像 $\underline{x}$ の帯域画素値を移動した後、同じ帯域画素に移動した帯域画素値の合計を濃淡画像にする。

$$\Gamma_{ijk}(\underline{x}, \underline{y}) = \sum x_{lmk} \text{ for } p(l, m, 1) \in P_{ij1}(q) \\ \text{and } \#^{-1} y_{lm1} = p(i-l, j-m, 0) \quad (10)$$

そこで数式 3、8、9 及び 10 を用いることにより、数式 11 又は数式 12 に従い、単帯域濃淡画像 $\underline{x}$ を近傍の重心方向に移動した後、同じ帯域画素に移動した帯域画素値の合計を簡単に記述することができる。

$$\Lambda_{ij1}(\underline{x}) = \Gamma_{ij1}(\underline{x}, \Delta(\Phi(\underline{x}))) \quad (11)$$

$$\Lambda'_{ij1}(\underline{x}) = \Gamma_{ij1}(\underline{x}, \Delta'(\Phi(\underline{x}))) \quad (12)$$

4. データ転送量と接続手法

前節で紹介した局所並列アルゴリズムは、実際に複数の AOU をまとめて VAOU として、LSI 上の 2 次元アレイに割り当てた際の PE 間の通信量が問題になる。まず、この転送量の評価を行ない、これに適した PE 間の転送機構を提案する。

4.1 VAOU 間のデータ転送量の評価

ここでは、 $m \times m$ 個の AOU をまとめて VAOU とし、これを PE に割り付ける。ここで、図 4 に示す通り、近傍長 l と転送領域内の AOU の数 N'_{ijk} の関係は、数式 13 となる。

$$N'_{ijk} = (2l + m)^2 - m^2 \quad (13)$$

次に、検証に用いた円図形のエッジ情報は、図 5 に示す通り、幅 100、高さ 100 の画像において、中心座標 (50, 50)、半径 $r = 1, 2, 3, \dots, 45$ 、線幅 $lw = 1, 3, 5$ である。このとき線幅 $lw = 1$ における半径 r とエッジ情報の総数 e は、図 6 に示す通りである。このことから、実質的に、重複情報が 0 でないものだけを VAOU 間で転送すれば、VAOU 間の転送データ量を近傍長 l に対して大幅に減らせることができる。以下では、特に断りが無い限り、線幅 $lw = 1$ とする。なお、円図形は端点がなく、また角もないことから、位置/大きさ検出アルゴリズムにとっては最も制約条

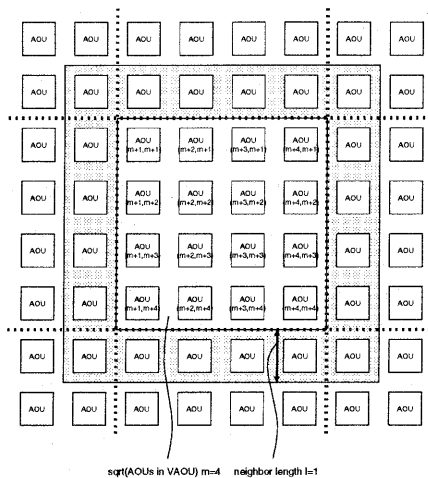


図4 VAOU に対する近傍

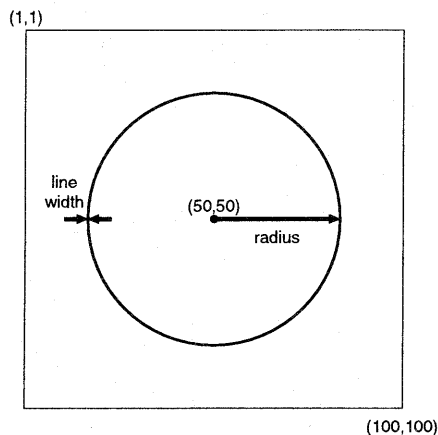


図5 評価に用いた円図形

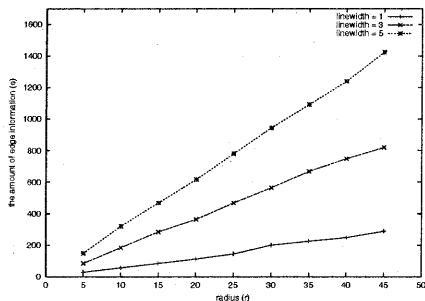


図6 円図形の半径とエッジ情報の数の関係

件が厳しい、つまり近傍長 l を長くしなければならない図形である。

次に図 13 に、近傍長 l に対する円図形の半径 r と位置／大きさ検出アルゴリズムの移動回数 t の関係を

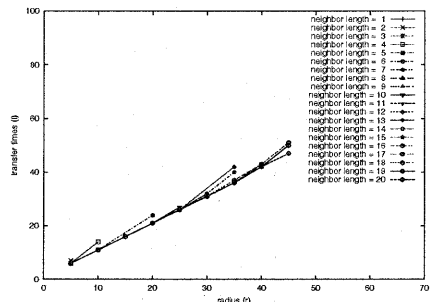


図7 円図形の半径と移動回数の関係

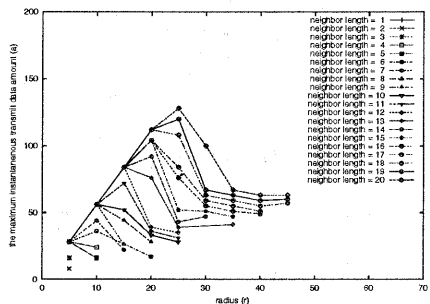


図8 $m = 1$ における円図形の半径と瞬間最大転送データ量の関係

を示す。この結果から、

$$t = r + \alpha \quad (14)$$

であることが判る。さらに、この図より、位置／大きさ検出アルゴリズムが重心を得るために必要な近傍長 l の最小値 l_{min} は、

$$\frac{r}{3} < l_{min} < \frac{r}{2} \quad (15)$$

を満たすことが判る。

このとき近傍長 l に対する円図形の半径 r と転送領域内の重複情報の最大量、すなわち瞬間最大転送データ量 a の関係を、VAOU のサイズ ($m \times m$) を変えて評価した。この結果を図 8、9、10 に示す。

このことから、円図形の半径 r が増加しても、瞬間最大転送データ量 a は増えないことが判る。例えば、VAOU 中の AOU の数の平方根 $m = 4$ の場合、近傍長 $l = 20$ であるとしても、半径 $r = 45$ の円図形に対して、瞬間最大転送データ量 a は 65 である。したがって、この場合には、視覚装置はデータを、AOU 間で水平方向及び垂直方向にそれぞれ 20 回転送しなければならないが、実際には VAOU 間を水平方向及び垂直方向にそれぞれ 5 回転送するだけで済み、しかも瞬間最大転送データ量 $a = 65$ は転送領域内の AOU の数 $N'_{ijk} = 1920$ の約 $1/29.5$ にもなり、円図

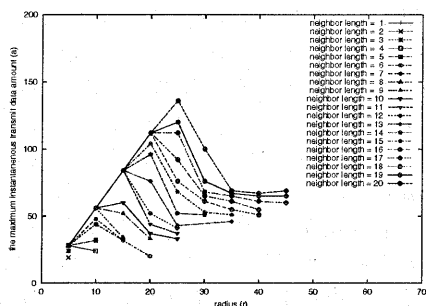


図9 $m = 4$ における円図形の半径と瞬間最大転送データ量の関係

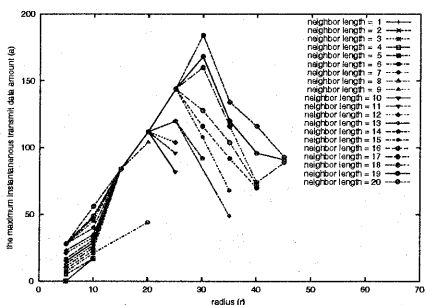


図10 $m = 20$ における円図形の半径と瞬間最大転送データ量の関係

形のエッジ情報の総数 $e = 288$ に対しても約 $1/4.4$ になる。この瞬間最大転送データ量 $a = 65$ は、近傍長 $l = 3$ の近傍サイズ $q = 48$ に近い。

この瞬間最大転送データ量 $a = 65$ は VAOU 全体からみると瞬間的な最悪値であるので、実質的な転送データ量はさらに少なくなる。このように瞬間最大転送データ量 a が円図形の半径 r に対して比例しないのは、エッジ情報を用いていることに加え、移動回数が少ないうちに近傍 AOU 同士の重複情報を集めた後で、移動回数が多くなってから、これらの重複情報を重心に向けて集めることによる。以上の検討により、視覚装置の LSI 化について PE 間のデータ転送量は限られたものとなり、これに適した簡単な通信機構を設ければ十分対処可能と考えられる。位置／大きさ／傾き検出アルゴリズム及び幾何解析アルゴリズムも、基本的に同様の特徴を有する。

4.2 PE 間の通信機構

以上、評価した局所並列アルゴリズムにおける VAOU 間のデータ転送経路は、図 11 に示す反時計回り又は時計回りで渦巻き状にまとめることができる。このようにすることで、各々の VAOU に転送データが集中することを避けると共に、転送プロトコルを

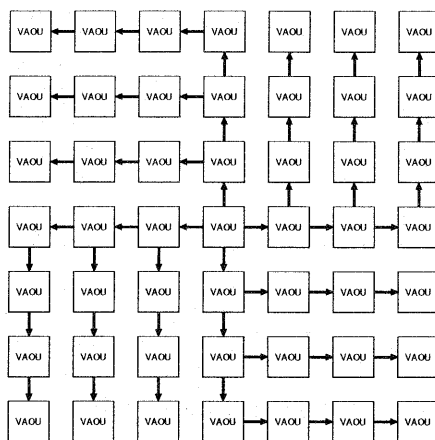


図11 データ転送経路

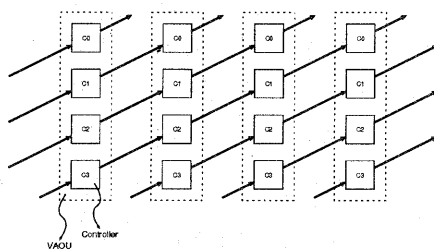


図12 転送用コントローラの接続

簡単化する。

上記のデータ転送を実現するために、単純なハンドシェイク機構を持った転送コントローラを(最大転送回数+1)個、それぞれの PE に装備する。これらのコントローラに番号付けを行ない、図??に示すように、これらを隣接 PE に対して番号が順に減るコントローラに対して結線する。この接続により、衝突を起こさずに転送を行なうことができる。この際、番号が0になったデータは廃棄して良いため、データの転送終了判定はプロセッサではなく、コントローラ内で簡単に行なうことができる。

5. 性能予測

5.1 アルゴリズムの検証

まず、正しく円図形の重心位置が検出できるかどうかを確認するため、前節で示したアルゴリズムを実行し、実際の円図形の重心位置と位置／大きさ検出アルゴリズムで求めた重心位置を比較した。この結果、13 に示す通り、一致していることが判る。

5.2 PE の動作周波数とハードウェア量の見積もり

簡単な 16bit の CPU に、視覚装置のアルゴリズム

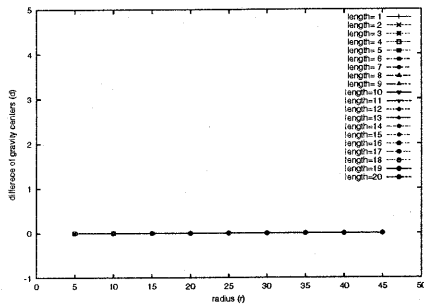


図 13 円図形の半径と重心の差の関係。

に必要な整数／固定小数乗算命令、転送用スタック操作命令を付け加え、コントローラおよび入出力部を設計して、ハードウェア量を評価した結果を表 1 に示す。ただし、ハードウェア量は、Altera 社 APEX20KE シリーズの FPGA の Logic Block 数で評価した。

表 1 PE の要求 Logic Block 数

CPU	2430
転送コントローラ	500
入出力	271

さらに、視覚装置の PE は、上記に加えローカルメモリを 8Kbyte を必要とする。しかし、最大級の APEX は、51840 Logic Block を搭載することが可能であることから、現状で 3x3 の程度のアレイが FPGA 上に余裕をもって搭載可能であることがわかる。最大動作周波数は 30.5MHz である。

5.3 性能見積もり

上記の PE アレイを ASIC 上に実装する場合、現状の技術でも 8x8 程度が可能であり、近い将来さらに大きなサイズの実装が可能である。また、クロック周波数は 100MHz を容易に越えることができるが、消費電力を考えて、50MHz で動作させることを考える。

まず、クロック周波数を f とし、 $m=1, l=1$ の場合におけるクロック数を $C(1, 1)$ とすると、任意の m 及び l に対するクロック数 $C(m, l)$ は、概算で数式 16 となる。

$$C(m, l) = \frac{am}{4(l+m)} C(1, 1) \quad (16)$$

そこで関数 $\Delta_{ij1}(x)$ を計算するために必要な、最大転送時間 $T_{\Delta}(m, l)$ は、数式 17 となる。

$$T_{\Delta}(m, l) = \frac{r}{f} C(m, l) \quad (17)$$

上記 PE によるアレイを命令レベルシミュレータ ISIS によりシミュレーションした結果、現在のプロセッサと転送ルーチンの組み合わせでは、 $C(1, 1) = 2400$ となる。そこでクロック周波数 $f = 50\text{MHz}$ とした場

合、線幅 $lw = 1$ の円の半径 $r = 25$ に対して $a = 60$ であるので、 $T_{\Delta}(4, 10) = 5.1\text{ms}$ である。関数 $\Gamma_{ij1}(x, y)$ が 8 近傍に対してデータを転送するため、 $a \approx 0$ と見做して良い。つまり、 $T_{\Gamma}(4, 1) \approx 0\text{ms}$ である。したがって、目標とする 33ms と比較すると、約 27.9ms 程余裕がある。つまり、16 個の AOU が関数 $\Delta_{ij1}(x)$ における積和演算及び関数 $\Theta(p(i, j, k))$ における IF 文をそれぞれ 25 回計算するために、約 139.5 万クロックサイクルを掛けることができるので、1 個の AOU 当り、1 回の計算に約 3487 サイクルを掛けることができる。また、 $r = 45$ に対して $a = 112$ の場合、 $T_{\Delta}(4, 18) = 11.0\text{ms}$ となる。

なお、線幅 lw が大きくなれば、 lw に比例して最大転送時間 $T_{\Delta}(m, l)$ も長くなる。また、ここでは、 $C(1, 1) = 2400$ としたが、プロセッサの性能を改善すれば、その分最大転送時間が短くなる。

文 献

- [1] 味岡義明、視覚装置、PCT 国際公開番号 WO 00/16259.
- [2] 味岡義明、情報収集システム、人工眼球、イメージセンサ及び連動装置、PCT 国際公開番号 WO 01/41448.
- [3] 棟上昭男、高度並列情報処理装置による図形の記述 (I)、電算報、vol.31, no.8, pp.930-946, 1967-08.
- [4] 大谷木重夫、森亮一、小林敦仁、渡辺祥司、局所並列演算による手書き”ひらがな”文字認識、信学技法、IE76-87, PRL76-85, 1977-03.
- [5] 加藤直也、鳥脇純一郎、福村晃夫、線図形の距離変換のための反復型局所並列処理とその応用、信学技法、PRL77-72, 1978-03.
- [6] 鳥脇純一郎、横井茂樹、画像処理のアルゴリズム、情報処理、vol.21, no.6, pp.613-619, 1980-06.
- [7] 鳥脇純一郎、横井茂樹、実行形式面から見た画像処理のアルゴリズム、信学技法、PRL81-60, 1981-12.
- [8] M.J.B. Duff, CLIP-4: A Large Scale Integrated Circuit Array Parallel Processor, Proc. 3rd IJCPR, pp.728-733, 1976.
- [9] K. Kyuma, E. Lange, J. Ohta, A. Hermanns, B. Banish, M. Oita, Artificial Retinas-Fast, Versatile Image Processing, Nature, vol.372, no.6502, pp.197-198, Nov. 1994.
- [10] T. Komuro, I. Ishii, M. Ishikawa, Vision Chip Architecture Using General-Purpose Processing Elements for 1ms Vision System, Proc. 4th IEEE Int. workshop on Computer Architecture for Machine Perception (CAMP'97), pp.276-279, 1997.
- [11] 石川正俊、マトリクス状センサからの出力分布の中心位置と総和の検出方法、計測自動制御学会論文集、vol.19, no.5, pp.23-28, 1983.
- [12] 村瀬洋、V.V.Vinod、局所色情報を用いた高速物体探索法—アクティブ探索法—、電子情報通信学会論文誌、vol.J81-D-II, no.9, pp.2035-2042, 1998.
- [13] 馬場功淳、大橋健、乃万司、松尾英明、江島俊朗、「人を見るシステム」のための人物追跡と頭部の分類、情報処理学会研究報告、HI-95, CVIM-129, pp.79-84, 2001.