

急速眼球運動対応の視線検出機能を持つ Vision Chip アーキテクチャ

著者	高木 宏章, 秋田 純一
雑誌名	映像情報メディア学会技術報告
巻	29
号	34
ページ	17-20
発行年	2005-06-23
URL	http://hdl.handle.net/2297/3533

急速眼球運動対応の視線検出機能を持つ

Vision Chip アーキテクチャ

高木 宏章[†] 秋田純一[†]

[†] 金沢大学 〒920-1192 石川県金沢市角間町

E-mail: [†] taka@merl.ec.t.kanazawa-u.ac.jp

あらまし 本論文では急速眼球運動に対応した視線検出機能を持つ Vision Chip アーキテクチャを提案する。視線の検出には瞳孔とプルキニエ像の位置の二つのパラメータが必要であり、今回提案する Vision Chip は、眼球画像からの瞳孔とプルキニエ像の検出方法、およびそれらの中心の検出方法、中心の座標生成を行うものである。本 Vision Chip アーキテクチャの構成要素のうち、二値化後の処理を行うデジタル処理部(PE)を FPGA 上に実装し、正しく動作することが確認できた。また PE 数を 100×100 とした場合のクリティカルパスを考慮したフレームレートは 146[kfps] となり十分に急速眼球運動を検出できることを確認した。

キーワード Vision Chip 急速眼球運動 並列処理 FPGA

Vision Chip Architecture for Saccade Tracking

Hiroaki Takagi[†] Junichi Akita[†]

[†] Kanazawa University kakuma-machi, kanazawa-shi, Ishikawa, 920-1192 Japan

E-mail: [†] taka@merl.ec.t.kanazawa-u.ac.jp

Abstract In this paper, we propose Vision Chip architecture for saccade tracking. Eye direction can be determined by the positions of the pupil and the Purkinje's image in eye image. The proposed Vision Chip has the ability of detecting their positions, as well as generating the coordinates of their center. We implemented the digital processing element (PE) array on FPGA, and the estimated frame rate for 100×100 PEs is 146 [kfps] with considering the critical path., which is fast enough for saccade tracking.

Keyword Vision Chip saccade Parallel Processing FPGA

1. はじめに

急速眼球運動（サッケード）とは我々の意図にかかわらず、ごく自然に起こる、非常に高速な眼球の運動である。急速眼球運動は新式のコンピュータ・インターフェースとして利用できるものと期待されるが[1,2],その速度は、多くの場合、毎秒 600° 以上にも及び、従来の視線検出装置で用いられる、NTSC 規格などのビデオカメラを用いて画像を取り込み、計算機等を用いてソフトウェア処理するという方式では対応することは不可能である。

このような背景から、本論文では高速な画像処理を行う Vision Chip を用いて急速眼球運動を含む視線を検出することを目標とした。Vision Chip は受光素子と処理回路を画素ごとに配置し、1 チップ上に集積させた知的な撮像センサー[3]であり、この構造によって逐次的な処理ではなく並列処理による超高速な画像処理が可能である。本論文では、急速眼球運動に対応する

Vision Chip アーキテクチャとその基礎的な評価について述べる。

2. 視線検出の原理

赤外線眼球画像は視線を見つけるのにふさわしい。赤外線眼球画像の例を図 1 に示す。この中には、2 つの特徴的な領域が存在する。1 つはカメラから角膜に正面方向から光を当てた場合に返ってくる強い反射光であり、プルキニエ像と呼ばれる白い領域であり、もう 1 つはまた黒く丸い領域である瞳孔である。これら 2 つの領域の位置関係を用いると、図 2 に示すようにプルキニエ像から角膜の半径直線を伸ばすことで角膜曲率中心が求められ、そこから瞳孔の中心へと直線を伸ばすことで視線ベクトル(視線)を求めることができる。すなわち視線を求めるためには、赤外線眼球画像からプルキニエ像と瞳孔の中心位置を求めることが必要であることがわかる。

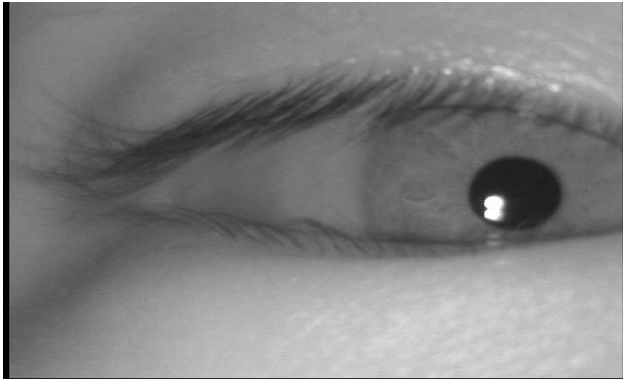


図1 赤外線眼球画像

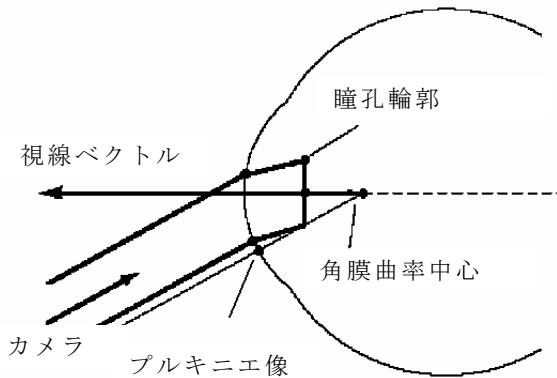


図2 視線検出の原理

3. Vision Chip アーキテクチャ

本稿で提案する Vision Chip の本質的な機能は、急速眼球運動を含む視線を求めて追跡するために、瞳孔およびプルキニエ像の両方の位置を求めることである。この Vision Chip は図3のように画素ごとに配置される処理要素 PE(Processing Element)から構成される。この PE は受光部と処理部からなり、上下左右の PE と連結している。受光部はフォトダイオードと増幅回路からなり、処理部は輝度値の判定回路、拡大収縮回路から構成される。また PE から出力されたプルキニエ像および瞳孔を示すフラグは $x \cdot y$ 軸上に配置されたエンコーダに射影され、座標を生成する。以下では判定回路、拡大収縮アルゴリズム、収縮終了判定アルゴリズム、座標生成の詳細について述べる。

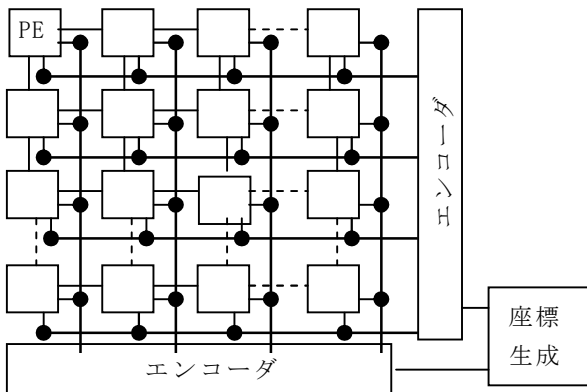


図3 Vision Chip アーキテクチャ

3.1 判定回路

判定回路では視線の検出に必要な瞳孔とプルキニエ像を検出し、それぞれを示すフラグを生成する。実際の検出方法として赤外線を目に照射すると、目の各部での反射率の違いにより、フォトダイオードでの光電流の値が異なることになる。そこで視線検出に必要な瞳孔とプルキニエ像を検出するために図4のような回路によって3値で表す事を考える。この回路ではプルキニエ像と瞳孔を検出できるよう閾値を設定しておき、その閾値と実際の明るさをコンパレータで比較することにより、プルキニエ像と瞳孔がそれぞれ検出される。

なお瞳孔を示すフラグは、プルキニエ像と瞳孔'の論理和によって求められるが、これは以下の理由による。

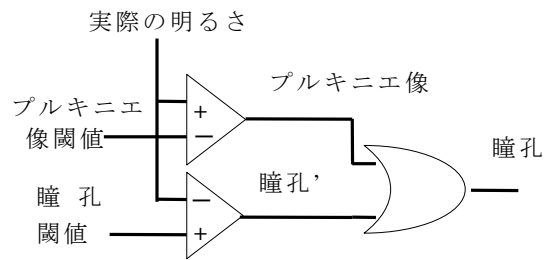


図4 判定回路

瞳孔とプルキニエ像の位置関係として図5の2つの場合がある。(a)のように瞳孔の中にプルキニエ像がある場合、瞳孔'だけで瞳孔の中心を求めようとすると、プルキニエ像の領域のところだけ瞳孔が欠けているように見えるため、瞳孔'の領域の中心が、本来の瞳孔の中心からずれてしまうことになる。そこで瞳孔'とプルキニエ像の論理和を、瞳孔を示すフラグとすることにより、プルキニエ像の部分の補い、本来の瞳孔の中心を求めることができる。また(b)のように瞳孔の外にプルキニエ像がある場合、瞳孔'とプルキニエ像の論理和をとると本来の瞳孔の外部にも、「瞳孔を示すフラグ」の小領域ができてしまうが、図5からもわかるように実際の瞳孔はプルキニエ像より大きいいため、以下で述べる収縮処理の過程でこのプルキニエ像に由来する小領域は消滅し、本来の瞳孔である部分のみが残る。

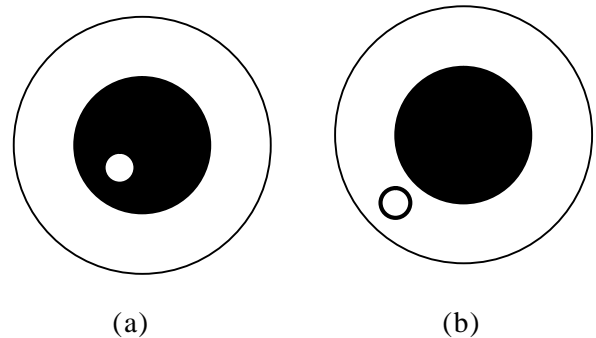


図5 瞳孔とプルキニエ像の位置関係

- (a) プルキニエ像が瞳孔の中にある場合
- (b) プルキニエ像が瞳孔の外にある場合

以上のことから、瞳孔を瞳孔'とプルキニエ像の論理和として求めることで、正しく瞳孔の中心を求めることができると考えられる。

図1の画像を判定回路に通した後の瞳孔を示すフラグの画像を数値シミュレーションによって求めた結果を図6に示す。これを見ると、まつ毛等の「ノイズ」が左やや上の長細い部分として現れている。なお右側の丸い部分が瞳孔である。この結果から、実際には瞳孔'とプルキニエ像の論理和をとったものを瞳孔としても瞳孔を示す領域のすべてが埋まるのではなく、輝度が瞳孔とプルキニエ像の中間となる領域がプルキニエ像の周囲に存在し、そこが抜けてしまうことになる。そこで以下で述べる拡大処理をあらかじめ施すことで、この抜けた領域を補うことにする。



図6 図1の画像を判定回路に通した結果

3.2 拡大収縮アルゴリズム

拡大収縮回路では瞳孔とプルキニエ像を検出したフラグの領域に対して、それぞれの中心を求める。瞳孔の拡大、収縮のアルゴリズムを図7に示す。

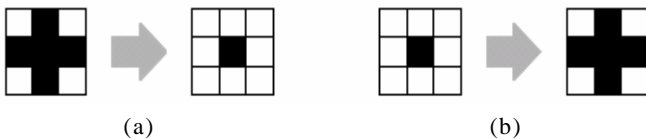


図7 瞳孔の拡大と収縮アルゴリズム

(a) 瞳孔の収縮アルゴリズム

(b) 瞳孔の拡大アルゴリズム

各 PE は瞳孔・プルキニエ像それぞれのフラグを上下左右方向に接続された隣接4近傍の PE へ渡すと同時に、その PE 自身も隣接4近傍の PE からフラグ値を受け取る。ここで PE の値で0(白)は瞳孔やプルキニエ像に該当しないことを示し1(黒)は該当することを示すとする。瞳孔を収縮する場合には、中心の PE と上下左右の隣接した4近傍の PE から受け取った値がすべて1のときその中心を1にするというアルゴリズムとし、これを以下に述べる収縮の終了条件が成立するまで繰り返す。なお瞳孔領域の収縮の前には、前述のよう

に数段階の拡大処理が必要であるが、これは、中心と上下左右の隣接した4近傍の PE から受け取った値どれかひとつでも1があるときその中心を1にするというアルゴリズムとし、これをあらかじめ設定しておいた回数分繰り返す。

3.3 収縮終了判定アルゴリズム

瞳孔、プルキニエ像とそれ以外の領域からなる3値画像に対して収縮を繰り返すと、対象となる瞳孔・プルキニエ像以外のノイズは先に消え、瞳孔・プルキニエ像のみが残る。これらの各領域の中心を求めるためにはさらに収縮を繰り返すことになるが PE 自身は収縮処理を終了するべきタイミングを知る必要がある。これは、PE からの瞳孔・プルキニエ像それぞれのフラグ出力を x, y 軸上に射影し、その射影した像をもとに収縮終了判定部で制御するとする。この収縮終了判定部の機能は、軸上に射影された全 PE からの瞳孔・プルキニエ像それぞれの1の値の数を数えるものである。

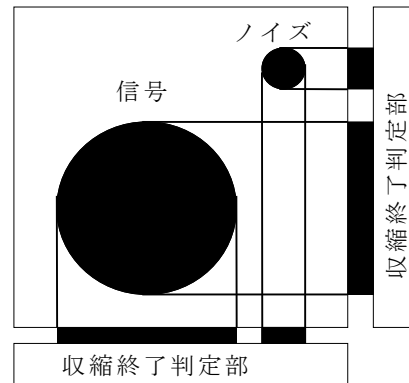


図8 収縮時の全体回路

この判定のために、二つの収縮終了判定アルゴリズムを検討した。一つ目は、遷移の過程の射影の像を D フリップフロップで保持しておき、軸上に存在する、値が1である射影線の数が0になったとき、一つ前の過程の収縮結果を読み出すという方法である。二つ目は軸上に存在する値が1である射影線の数が1または2本の時は収縮処理を終了させ、また x, y 軸ともに3本以上の場合は収縮を継続するという方法である。これは軸上に存在する値が1である射影線の数が0になる直前の収縮過程では、必ず x, y 軸上に存在する値が1である射影線が、少なくとも一方で1,2本になることを利用している。具体的な例として図9で示すと、図9(a)は3×3の PE が黒(値が1)の状態であり、ここから1段階収縮過程が進むと真ん中の PE のみが黒(値が1)の状態になる。すなわちこの図9(a)は、収縮過程を継続すべき状態であるといえる。逆に図9(b)は2×2の PE が黒(値が1)の状態であり、この状態から1段階収縮過程が進むと、全ての PE が値が0になってしまうことになる。

すなわちこの図 9(b)は、は収縮過程を終了するべき状態といえる。

この両者を比較した場合、前者の方法では、全ての PE に D フリップフロップを配置する必要があるために、一つの PE 当りの回路規模が増大してしまうことになるため、本研究では後者の方法を用いることにした。

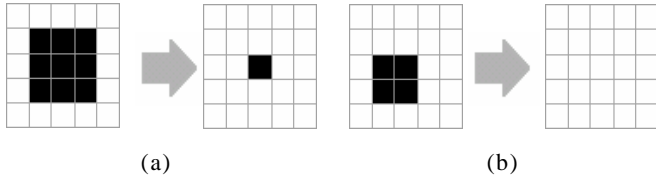


図9 2つ目の方法の収縮終了条件

(a) 収縮を繰り返さなければならない場合

(b) 収縮を終了しなければならない場合

3.4 座標生成

前述の収縮終了判定回路によって収縮過程の終了が検出されると、瞳孔とプルキニエ像のそれぞれの中心の座標を生成する必要がある。この収縮終了時点では、x, y 軸上に存在する値が 1 である射影線は 1 本または 2 本であるため、各軸上に 2 つのプライオリティエンコーダを配置することで、各軸上の値が 1 である射影線の位置の座標の最大値と最小値を出力することができる。

4. FPGA への実装と実測

今回提案された Vision Chip アーキテクチャの処理速度を評価するために、瞳孔・プルキニエ像を示すフラグが生成された後の処理回路を FPGA 上に実装し、その評価を行った。対象の FPGA としては Altera の EP1C20F324 を使い、PE 数が 8×8 の場合と 16×16 の場合のそれぞれで処理回路の実装を行い、遅延時間などの測定を行った。なお瞳孔およびプルキニエ像に対する 2 値化されたフラグの値は、シフトレジスタを用いて値を外部から与えた。この結果、すべての入力パターンに対して正しく中心の座標を求められることを確認した。

PE 数が増えた場合の動作速度を見積もるために、PE 数が 8×8 の場合と 16×16 の場合のそれぞれでクリティカルパスの最大遅延を求めたところ、それぞれ 27.7[ns]と 30.8[ns]となり、この両者の差は 3.1[ns]となる。本研究で考案したアーキテクチャでは、PE からの射影線の出力部分の論理回路の段数が、一辺の PE 数に比例するため、PE 数が増えた場合にはここがクリティカルパスとなると考えられる。これは図 10 のように PE からのフラグ出力と射影線を繋ぐ OR ゲートが原因となるが、この場合は、両者の OR ゲートの段数の差は 15-8=7 段であるため、これを一般化すると PE の数が n × n の場合には n-1 個の OR ゲートを通ることになるた

め、PE 数が n × n の場合の最大遅延 $T_d(n)$ は式(1)のように見積もられる。

$$T_d(n) = 24.6 + 0.44 \times (n-1) \text{ [ns]} \quad (1)$$

ここで定数項である 24.6[ns]は OR ゲート以外の部分での遅延であり、また 0.44[ns]が OR ゲート 1 段あたりの遅延であると考えられる。この結果から、実用上十分な解像度と考えられる PE 数が 100×100 の場合の、クリティカルパスにおける最大遅延は 68.2[ns]となる。この場合、収縮処理には最大で 100 ステップ必要となるため、ワーストケースでのフレームレートは、 $1/68.2[\text{ns}] \div 100 = 146[\text{kfps}]$ となる。これは急速眼球運動を検出・追尾するのに十分な速度であると考えられる。

射影線

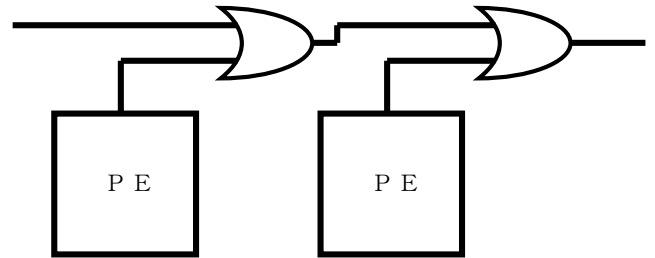


図10 PE と射影線を繋ぐ OR ゲート

5. まとめ

本研究では急速眼球運動に対応する視線検出機能をもつ Vision Chip アーキテクチャを考案し、要素回路の検討を行った。そしてそれらの回路構成およびアルゴリズムの検証のために FPGA を用いた実装と測定を行い、正しい座標の生成を行うことができることを確認した。また PE 数が増加した場合の動作速度の見積も行い、PE 数が 100×100 で 146[kfps]程度の処理が可能であることを示した。受光回路を含む全体の回路設計およびその評価は今後の課題である。

文 献

- [1] J.Triesch, D.H.Ballard, M.M.Hayhoe, B.T.Sullivan, : "What you see is what you need," *Journal of Vision*, No.3, pp.86-94, 2003.
- [2] S.S.Intille, : "Change Blind Information Display for Ubiquitous Computing Environments," *Ubi-Comp2002*, LNCS 2498, pp.91-106, 2002.
- [3] T.Komuro, S.Kagami, M.Ishikawa, : "A Dynamically Reconfigurable SIMD Processor for a Vision Chip", *IEEE Journal of Solid-State Circuits*, Vol.39, No.1, pp.265-268, 2004.1.