

I_026

海馬時系列符号化モデルに基づく移動体検出手法のデジタル LSI アーキテクチャ
 A Digital LSI Architecture for Moving Detection Based on Sequence Coding
 in a model of Hippocampus

梁 海超† 森江 隆† 中山 浩之† 中田 一紀† 林 初男†
 Haichao Liang Takashi Morie Hiroyuki Nakayama Kazuki Nakada Hatsuo Hayashi

1. はじめに

近年、自動車の安全性を高めるために多くの運転支援システムが開発されている [1,2]。それらのシステムでは、カメラ画像からの物体の移動検出が課題となっている。従来の手法では、複数カメラ画像を用いて、画像マッチング処理により対象物の移動検出を行うものが主流であった。しかし、画像マッチング処理は計算量が膨大であり処理時間がかかる一方、使用されるカメラのフレームレートは 30 fps 以上が要求される。したがって高速に移動物体を検出する手法が必要となる。今回、我々は海馬の放射状伝播を利用した時系列符号化機構に基づく移動体検出手法を考案した。この手法は、実時間処理に適した並列アルゴリズムとなっており、集積回路化による実時間処理が期待できる。本研究では、この手法のデジタル回路構成について提案する。

2. 海馬モデルに基づく移動物体検出手法

海馬は脳の部位のひとつであり、主に歯状回、CA3 領域、CA1 領域から形成される。それらの各領域には領域特有の神経細胞が層状に並んでおり、各神経細胞はシナプスを介して相互に結合している。この海馬では、入力された時系列信号を神経細胞の発火の時空間活動パターンに符号化していると考えられている。つまり、海馬において神経活動が領域内を放射状に伝播し、これを利用することによって入力情報の時間的順序を符号化することができる [3]。さらにこの機構を使うと、物体の位置と移動時間を検出することができる。

この移動物体検出手法のアルゴリズムでは、放射状の格子で区切られた 3 種類の 2 次元マップを用いる (図 1)。これらのマップをそれぞれ実画像マップ、CA3 マップ、CA1 マップと呼ぶ。実画像マップは上下左右の 4 領域に分割され、それらに対応して 4 枚の CA3 マップを用意する。さらに、各 CA3 マップにはその領域での上下左右 4 方向の移動検出に対応して 4 枚ずつの CA1 マップ (合計 16 枚) を用意する。

CA1 マップ上の各ユニットは同じ座標を持つ実画像マップ上のユニット内にある物体の通過時間を表す値を保持する。実画像マップにはカメラ画像に適当な画像処理を施した結果 (例えばエッジ抽出した結果) が入力され、物体エッジと重なったユニットを発火ユニットと定義し、同じ座標を持つ CA3 マップ上のユニットに発火信号を送る。CA3 マップ上のユニットは発火信号を受け取り、CA1 マップ上のユニットに 2 種類の信号を投射する。一つは減衰信号であり、同じ座標を持つ各 CA1 マップ上のユニットの一つづれた位置のユニットに投射し、それらのユニットが既に減衰中でなければ保持している値を初期値から減衰させる。

もう一つの信号は値確定信号であり、同じ座標を持つ各 CA1 マップ上のユニットに投射し、それらのユニットが減衰中であるときに限り、その値を確定させる。確定された値により、対応する実画像マップのユニット内の物体の移動時間を計算する。また、確定された値は次のフレームで初期値に戻る。

ニューロン活動と同様に、このユニットにも不応期を導入した。不応期内では、新たに来た発火信号に応答しない。さもないと、誤検出を起こす場合がある。不応期長は検出した速度帯の最低速度に応じて調整する。

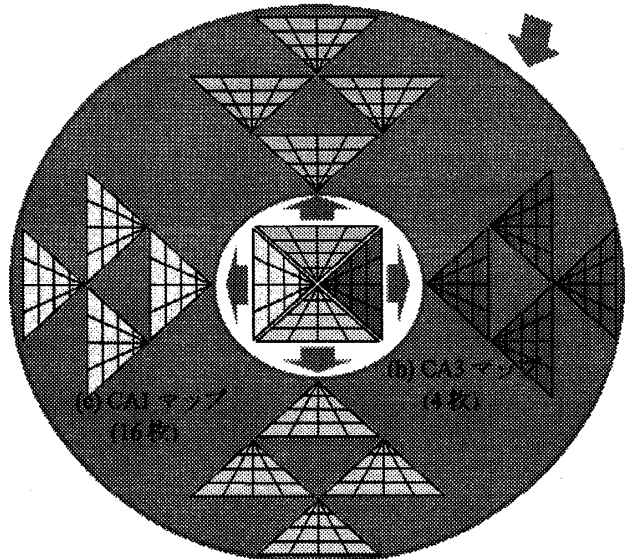
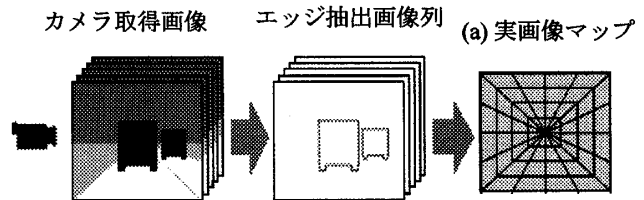


図 1 移動物体検出モデル

表 1 モデルの仕様

入力画像サイズ	320pixels×256pixels (W×H)
画像サンプリング時間	8msec
ユニット数	32×32unit (行×列)
CA1 減衰パラメータ	線形 64 段階
CA3 不応期の長さ	512msec (8msec×64)

† 九州工業大学 大学院生命体工学研究科

3. 海馬記憶モデルのデジタル LSI 化

3.1 アーキテクチャ

本研究で提案するシステムの構成を図 2 に示す。また、モデルの仕様を表 1 に示すように決定した。ここでは実画像マップ上のユニットの発火情報を入力情報とする。提案するデジタル LSI アーキテクチャでは、減衰器をカウンタで構成し、さらに消費電力を抑えるために、共通カウンタ型回路を用いる。CA3 マップと CA1 マップは、ユニットごとにレジスタを設置する。実画像マップ上のあるユニットが発火すると、対応する座標を持つ CA3 マップ上のユニットはそのタイミングのカウンタ値をレジスタに保持する。発火情報処理では、CA3 マップ上の発火信号を受けたユニットは各 CA1 マップ上の対応する座標の一つづれた位置のユニットにそのカウンタ値を伝送する。移動計算処理では、CA3 マップ上のユニットと各 CA1 マップ上の同じ座標を持つユニットとカウンタ値の引き算を行い、移動体を検出し、移動時間を計算する。ユニット間の距離が決まっているため、移動時間で相対速度が計算できる。その相対速度により、危険警告や画像背景抽出などの処理を行う。

モデルの回路実装において、次の3つの点を工夫した。

I. CA3 マップおよび CA1 マップの実現：

共通カウンタ信号のサイズが N ビットであれば、CA3 マップと CA1 マップのレジスタのサイズは N+1 ビットにする。レジスタの初期値はすべてのビットを 1 に設定する。初期値を持つユニットは、初期状態であると見なし、そのユニットの移動計算処理は行わない。

CA3 マップと CA1 マップには 1 フレーム分のデータを記憶し、処理時間内のデータを重ね書きする。ただし、初期状態でないユニット(活動中ユニットと定義)に対し、不応期内では上書きしない。

II. 減衰信号の実現：

減衰信号をダウンカウンタ回路によって構成した(図 3)。カウンタ信号周期は不応期長より長くした。カウンタ信号を制御するクロック信号はシステムクロック信号ではなく、入力画像が更新するたびに動作するように、サンプリング時間に対応して 125Hz クロック信号とした。カウンタ値が最大値(COUNT_MAX)に戻った場合、活動中ユニットに保持されているカウンタ値(SC)を $SC+COUNT_MAX+1$ に変換する。

III. 不応期の導入：

ある時点のカウンタ信号値に不応期長に対応するカウンタ値を足した結果は、その時点の活動中ユニットが不応期内にあるかどうかの判断基準になる。すなわち、ユニットに保持されているカウンタ値がそれより大きければ、不応期を脱したと見なし、そのユニットをリセットし、初期状態に戻す。

3.2 回路での処理フロー

回路処理プロセスを 5 つに分けて、待機状態、発火情報処理、移動計算処理、出力処理、不応期処理とし、図 4 のフローチャットに示すような流れで処理を行う。

- 待機状態：CA3 マップおよび CA1 マップ上の各ユニットをリセットする。移動計算結果を保存する各方向別メモリをリセットする。

- 発火情報処理：実画像マップに発火したユニットがあると、CA3 マップ上の対応する座標を持つユニットが初期状態である限り、その時点のカウンタ値をラッチし、各ユニットに保持する。
- 移動計算処理：CA3 マップ上の各ユニットに対し、4 つの CA1 マップ上の対応する座標を持つユニットを抽出し、上下左右の 4 つの方向での移動計算処理を同時に行う。移動計算をする時、まず各ユニットの状態を確認し、CA3 マップ上のユニットとある方向の CA1 マップ上のユニットがともに活動中であり、かつ CA3 マップ上のユニットに保持されているカウンタ値がより小さい場合に限り、その方向の移動計算を行い、計算結果を対応する方向別メモリへ書き込む。
- 出力処理：CA1 マップ上の各ユニットの移動計算結果を Windows 画面上の輝度で表示する。移動体を検出したら、移動始点のユニットをリセットし、初期状態に戻す。

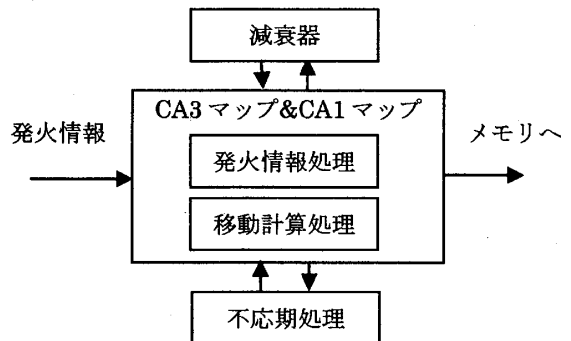


図 2 デジタル LSI アーキテクチャ

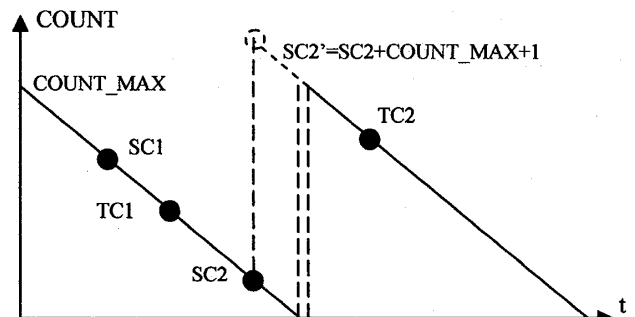


図 3 ダウンカウンタによる減衰信号の実現法

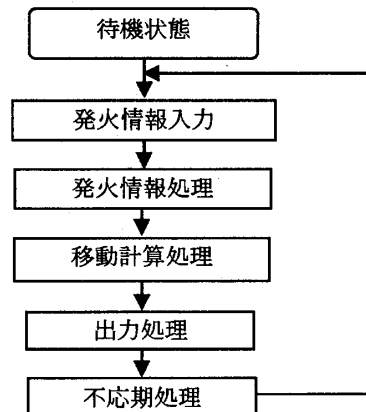


図 4 処理フロー

- 不応期処理：CA3 マップ及び CA1 マップ上の各ユニットごとに図 5 に示すフローチャットに従った処理を行う。

4. シミュレーション結果

ユニット数 3×3 の CA3 マップを構成し、その一部を抽出し、ModelSim によるシミュレーションを行った。

図 6(a) - (c) に示すような 3 つのパターンを用意した(黒い部分が発火ユニット)。なお、見やすくするために、放射状のマップを正方形格子に変換し、ユニットの番号を図 6(d) のように定義した。回路は物体エッジ抽出処理を含めていないため、画像サンプリング時間を 4ms、カウンタを制御するクロック信号を 250Hz、システムクロック信号を 50MHz に設定し、シミュレーションをした。まずパターン 1 を入力し、12ms 後にパターン 2 を入力し、さらに 16ms 後にパターン 3 を入力した場合の結果を図 7 に示す。パターン 2 を入力した後、3 番ユニットに上方向移動が、5 番ユニットに下方向移動が検出された。カウンタ値の差が 3 であるため、移動時間が $3 \times 1/250s = 12ms$ と算出できる。パターン 3 を入力した後、0 番ユニットに上方向移動が、8 番ユニットに下方向移動が検出された。カウンタ値の差が 4 であるため、移動時間が $4 \times 1/250s = 16ms$ と算出できる。これらは入力情報と一致している。パターン入力時間を調整し、不応期処理の妥当性も確認した。

5. デジタル回路の FPGA 実装

ALTERA 社の Stratix™ II 系列 EP2S60F672C 型ボードを用い、回路の FPGA 実装を行った。論理合成の結果は下記の通りであり、最大動作周波数は 84.3Hz となった。

Device Utilization for EP2S60F672C

Resource	Used	Avail	Utilization
LUTs	45156	48352	93.39%
Registers	43072	51182	84.15%
Memory Bits	279552	2544192	10.99%

6. まとめ

海馬の時系列符号化機構に基づく移動検出手法を提案し、そのデジタル回路実装を行った。今後は、実画像を処理できる物体エッジ抽出処理を含めた移動体検出システムの構築を目指す。

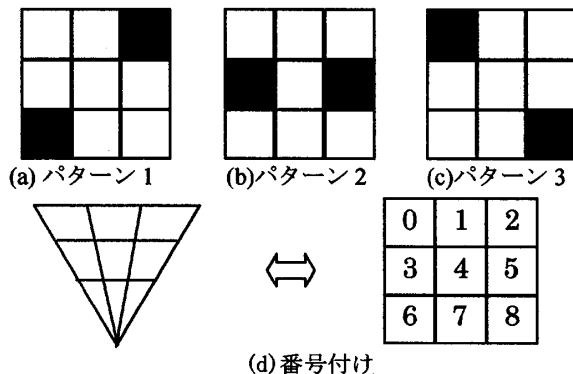


図 6 シミュレーションに用いた発火パターン

参考文献

[1] 佐々木一人, 河本新二, 安全運転を支援する車載画像処理システム, 東芝レビュー, Vol.59, No.4, pp.28-31, 2004.
 [2] N. Shima, M. Babasaki, Y. Akidzuki, K. F. Honda, T. Higuchi, H. Higashida, and R. Nakamura, Fusion Sensor for Driving Assistance System, Fujitsu Ten Tech. J., No.17, pp.35-44, 2001.
 [3] M. Yoshida and H. Hayashi, Organization of cell assemblies that code temporal sequences in a hippocampal CA3-CA1 model, Int. Joint Conf. on Neural Networks (IJCNN), pp.495-500, 2004.

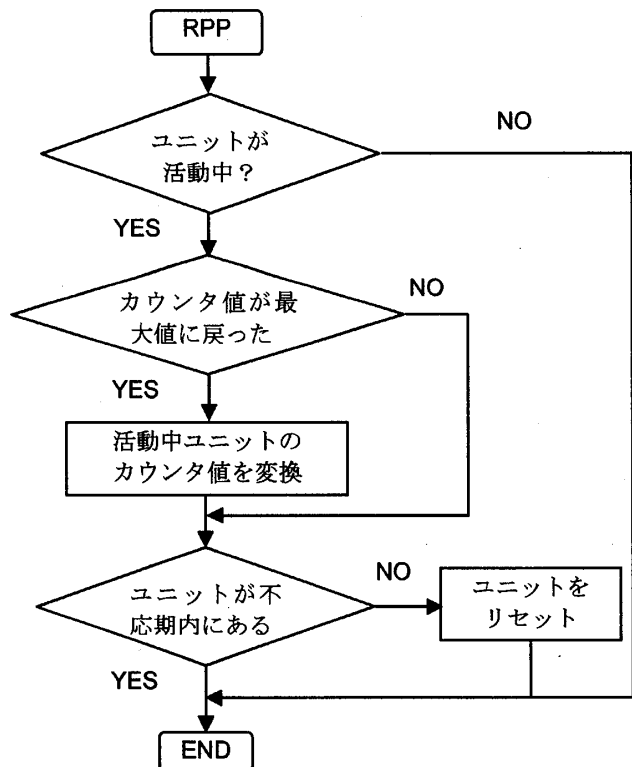


図 5 不応期処理

(1) パターン 2 入力後 (2) パターン 3 入力後

/REG_test/car/OUT_U0	0	/REG_test/car/OUT_U0	4
/REG_test/car/OUT_U1	0	/REG_test/car/OUT_U1	3
/REG_test/car/OUT_U2	0	/REG_test/car/OUT_U2	0
/REG_test/car/OUT_U3	3	/REG_test/car/OUT_U3	0
/REG_test/car/OUT_U4	0	/REG_test/car/OUT_U4	0
/REG_test/car/OUT_U5	0	/REG_test/car/OUT_U5	0
/REG_test/car/OUT_U6	0	/REG_test/car/OUT_U6	0
/REG_test/car/OUT_U7	0	/REG_test/car/OUT_U7	0
/REG_test/car/OUT_U8	0	/REG_test/car/OUT_U8	0
/REG_test/car/OUT_D0	0	/REG_test/car/OUT_D0	0
/REG_test/car/OUT_D1	0	/REG_test/car/OUT_D1	0
/REG_test/car/OUT_D2	0	/REG_test/car/OUT_D2	0
/REG_test/car/OUT_D3	0	/REG_test/car/OUT_D3	0
/REG_test/car/OUT_D4	0	/REG_test/car/OUT_D4	0
/REG_test/car/OUT_D5	3	/REG_test/car/OUT_D5	0
/REG_test/car/OUT_D6	0	/REG_test/car/OUT_D6	0
/REG_test/car/OUT_D7	0	/REG_test/car/OUT_D7	0
/REG_test/car/OUT_D8	0	/REG_test/car/OUT_D8	4

図 7 シミュレーション結果