計測自動制御学会東北支部 第192回研究集会 (2000.12.12)
資料番号 192-5

ボール軌道予測アクセラレータを用いた捕球ロボットシステム

Ball-Catching Robot System Using a Ball-Trajectory Prediction Accelerator

○風間 英樹, 佐々木 明夫, 張山 昌論, 亀山 充隆

○Hideki Kazama, Akio Sasaki, Masanori Hariyama, Michitaka Kameyama

東北大学情報科学研究科

Graduate School of Information Sciences
Tohoku University

キーワード： 知能集積システム(intelligent integrated system), ステレオビジョン(stereo vision), ボール抽出(ball extraction), メモリアロケーション(memory allocation)

連絡先： 〒980-8579 仙台市青葉区荒巻字青葉05 東北大学大学院情報科学研究科亀山研究室
風間 英樹, Tel.: (022)217-7155, Fax.: (022)263-9167, E-mail: hideki@kameyama.ecei.tohoku.ac.jp

1. まえがき

高安全知能自動車や家庭用サービスロボットなどのリアルワールド応用においては, 膨大な量の入力データを高速に実行できる高性能な専用プロセッサの開発が重要である[1, 2, 3]. リアルワールド応用においては, 運動物体が将来どのような位置軌道をとるかを予測する, 軌道予測が重要な基礎技術となる. 軌道予測システムの一例として, 動いているボールの軌道を予測し, 捕球するシステムを考える. このようなシステムは, 球技ロボットなどを実現するための基礎となるものである. ボール捕球システムでは, 精度の高いボールの軌道予測を行うために, 単位時間にできるだけ多く, ボールの3次元位置を取得する必要がある. しかし, その基本処理であるボール抽出には, 非常に多くの計算時間が必要となる.

ボール抽出の代表的な方法として, (1)ボールの色の事前知識に基づき, その色の領域をボールとして抽出する方法, (2)画像間の差分を用いて動いている領域をボールとして抽出する方法, (3)ハフ変換により画像のエッジに含まれる円弧を抽出し, その円弧をボールのエッジとする方法, (4)ボールの形状と大きさを表すテンプレートを用意し, そのテンプレートに近い形状と大きさを持つ領域をボール領域として抽出する方法などがある. 本稿では, 以下の理由から(4)を用いたボール抽出を行っている. まず, (1)では, 背景にボールと同じ色のものがある場合や, 照明の変化によりカメラに写るボールの色が変化した場合に, ボールを抽出できないという問題がある. これに対し, (4)では, ボールの形状を利用して抽出を行うため, 背景にボー

ルと同じ色のものがある場合でもボールを抽出できる．また，ボールの色を事前知識として必要としないため，カメラに写るボールの色が変化した場合でもボールを抽出できる．次に，(2)では，ボール以外に動く部分がある場合に，ボールを抽出できないという問題がある．これに対し，(4)では，静止画からボールを抽出しているため，ボールを抽出できる．(3)では，円弧というエッジの情報だけを用いてボールの抽出を行うため，ボール以外のエッジが多い複雑な画像において，誤抽出が多く発生するという問題がある．これに対し，(4)では，円弧内部を含む円形領域との照合により抽出を行うため，複雑な画像においても誤抽出が少ない．

(4)は，階層的にテンプレートとのマッチングを行うことにより，計算量を減少できることが知られている．階層的テンプレートマッチングでは，サンプリングして画素数を減らした画像を用いてボールの大まかな位置を抽出し，ボールが存在する可能性の高い領域に対してだけ，元の画像で精度の高い抽出を行う．しかしながら，階層的テンプレートマッチングを用いた場合でも，計算量が膨大となるという問題がある．この問題に対し，本稿では，階層的テンプレートマッチングに適合するVLSIプロセッサの構成を提案している．

シミュレーションにより決定したボールを捕球するために要求されるボール抽出時間を制約として，チップ面積最小化を考える．時間制約を満たすVLSIプロセッサを実現するためには，パイプライン処理だけでなく，空間的並列処理も活用する必要がある．そのため，メモリシステムは，階層的テンプレートマッチングの全ての階層において，並列にデータ供給ができる構成とする必要がある．本稿では，並列データ供給方式として，メモリシステムを複数のメモリモジュールで構成し，各メモリモジュールから独立にデータを読み出せるようにする方式を考える．このような方式においては，多数のメモリモジュールと多数の演算器間の相互結合回路網の面積増大が最大の問題となる．そこで，相互結合回路網を最小化するために，メモリモジュールと演算器を一対一で接続する構成とする．この構成では，相互結合回路網の面積は無視できる程度となるため，全体のチップ面積は，メモリ部の面積と演算部の面積の和となる．メモリ部の面積は，必要なメモリ容量が一定であることから，一定となる．一方，演算部の面積は，演算器数で決まり，その数はメモリモジュール数と同数となるため，メモリモジュール数が最小の時，面積が最小となる．そこで，メモリモジュール数を最小化することにより，チップ面積の最小化を実現する．

階層的並列メモリアクセスに対応可能で，かつ，メモリモジュール数が最小となる，メモリアロケーションを提案する．階層的テンプレートマッチングでは，処理階層により並列に読み出すべき画素のパターンが変化する．そのため，通常，階層的並列メモリアクセスを実現するためには，多数のメモリモジュールが必要となるという問題がある．これに対し，サンプリング間隔と互いに素となる周期で画素を周期的にメモリモジュールを割り当てることにより，最小のメモリモジュール数で，並列にアクセスを可能とするメモリアロケーションを提案する．

この設計方針に基づき，処理時間の制約条件10msec以内で640 × 480画素の画像中から8~31dotの大きさのボールを抽出できるボール抽出VLSIプロセッサを0.35μm CMOS設計ルールにより設計した．その結果，通常の構成と比較して，チップ面積を12%(125mm^2)にまで減少することができた．

2. ボール捕球システム

2.1 全体の構成

Fig.1に示すような，前方から投げられたボールを2台のカメラを用いたステレオビジョンにより計測し，マニピュレータによりボールを捕球する

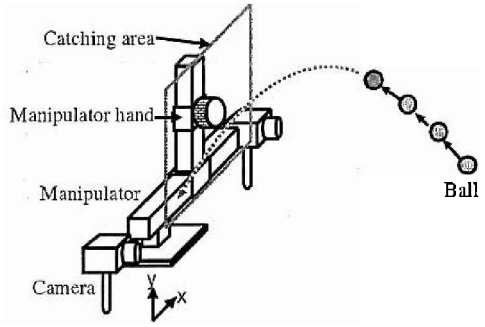


Fig. 1　ボール捕球システムの構成

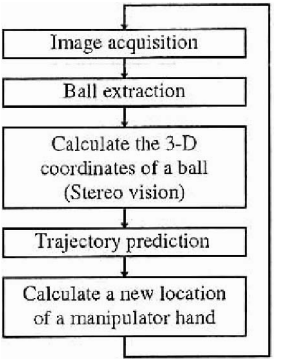


Fig. 2　ボール捕球アルゴリズムのフローチャート

システムを考える. 通常, マニピュレータの移動速度はボールの移動速度に比べてかなり遅い場合が多く, ボールが捕球位置に近づいてからマニピュレータの移動を開始するのでは間に合わない. そこで, ボールの到達位置を予測し, あらかじめ到達位置に向かって動いておくことが必要となる. しかし, 早い時刻ではボールが遠方にあるため計測精度が低く, さらにボールの過去の軌跡に関する情報も少ないため, 到達位置の予測精度が低いという問題がある. この問題を解決するために, Fig.2に示すように, カメラからの画像取得, 取得画像中のボール領域の抽出, 抽出したボールの中心座標を用いた三角測量によるボールの3次元座標計算, ボールの3次元座標の時系列を用いたボールの軌道予測, ボールの予測到達位置へ向けたマニピュレータの目標位置姿勢の更新を捕球するまで繰り返し行う. ただし, 軌道予測においては, ボールは空気抵抗の影響を受けずに放物線軌道をとると仮定し, 最小2乗法を用いて, 過去のボールの軌跡に最も近い放物線軌道を予測軌道とする. Fig.2のように処理を行う事により, 早い時刻から到達位置へ向かってマニピュレータをグローバルに移動させておくことができ, また, より新しい計測データを用いて予測軌道を更新するため, 予測制御の精度を高めることができる.

Fig.2のループ1回当りの処理時間を短くし, ボールの3次元位置を多く取得するほど, 予測軌道の精度が向上し, マニピュレータを効率よくボール到達位置へと移動させることができるため, 捕球成功率が高くなる. したがって, 軌道予測の処理要素のうち計算量が特に多いボール抽出の高速化が, 捕球成功率を高くするために重要となる.

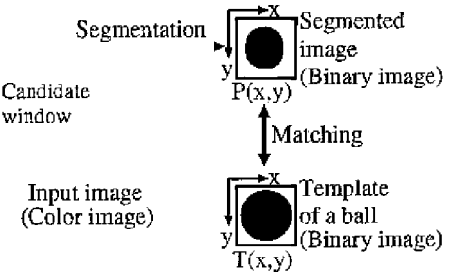


Fig. 3　ボール抽出アルゴリズム

2.2　階層的テンプレートマッチングに基づくボール抽出アルゴリズム

抽出対象は模様のない任意の色のボールとする. また, 画像の大きさを$W \times H$, 抽出するボールの半径の最小値, 最大値をそれぞれR_{min},R_{max}とする.

ステップ1: 画像の色情報をRGB空間からUCS空間へ変換する. UCS空間では色情報をスカラー量である明度と2次元のベクトル量である色度で表すため, 明度と色度を分離することが可能である. そのため, ステップ4において, ボールの影などの影響のない領域分割ができる.

ステップ2: テンプレートマッチングを行う候補ウィンドウの左上の座標を$(i_w, j_w) = (0, 0)$に設定する.

ステップ3: 抽出するボールの半径を$r = R_{min}$に設定する. ボールのテンプレートの大きさは, 半径rのボールを含む大きさにする. このとき, Fig.3に示すように, テンプレートはボール領域を1, 背景領域を0とした2値画像とする.

ステップ4: Fig.3の破線で示すように, 候補ウィンドウ内にテンプレートと同じボール領域が存在したと仮定し, その仮定したボール領域内の画素の集合をFとする. また, 集合Fに属する画素数をN_F, 任意の座標(i, j)の画素の色度を$(U(i,j), V(i,j))$とする. テンプレートのボール領域と同じ大きさのボールが候補ウィンドウ内にある場合には, 集合F内のほとんどの画素はボール領域に対応している. このとき, 抽出対象は模様のないボールであるため, 集合F内の画素はほぼ同一の色度をとる. そこで, 式(1)で表される, 集合F内の画素の平均の色度(U_{ave}, V_{ave})と候補ウィンドウ内の画素Pの色度(U_p, V_p)のチェス盤距離$\max\{|U_p - U_{ave}|, |V_p - V_{ave}|\}$がしきい値$\varepsilon$以内の場合, Pをボール領域の画素とし, それ以外の場合, Pを背景領域の画素として, 領域分割を行う. この領域分割によりボール領域の画素値を1, 背景領域の画素値を0とした2値画像を生成する.

$$(U_{ave}, V_{ave}) = \left(\frac{1}{N_F} \sum_{(i,j)\in F} U(i,j), \frac{1}{N_F} \sum_{(i,j)\in F} V(i,j) \right) \quad (1)$$

ステップ5: 領域分割された候補ウィンドウとテンプレート内の任意の座標(x, y)の画素値をそれぞれ$P(x,y)$,$T(x,y)$とする. 領域分割された候補ウィンドウとのテンプレートマッチングによる評価は式(2)で表されるSAD(Sun of Absolute Differences)値Sにより行う. $P(x,y)$,$T(x,y)$は2値のデータであるため, 式(2)に示すように, 絶対値差分は排他的論理和により計算できる.

$$\begin{aligned} S &= \sum_{(x,y)\in Window} |P(x,y) - T(x,y)| \\ &= \sum_{(x,y)\in Window} \{P(x,y) \oplus T(x,y)\} \end{aligned} \quad (2)$$

式(2)の値が, しきい値よりも小さい時, ボールであるとする. ただし, しきい値はあらかじめ実験的に求めた適切な値とする.

ステップ6: $r \leftarrow r + 1$. $r \leq R_{max}$ならば, ステップ4へ戻る.

ステップ7: $i_w \leftarrow i_w + 1$. $i_w < W$ならば, ステップ3へ戻る.

ステップ8: $i_w \leftarrow 0$, $j_w \leftarrow j_w + 1$. $j_w < H$ならば, ステップ3へ戻る.

以上のアルゴリズムでは, 一画面内の全ての位置のウィンドウに対して, 全ての大きさのテンプレートとのマッチングを行う必要があるため, 計算量が膨大となるという問題がある. この問題を解決する方法として, Fig.4に示す階層的テンプレートマッチングが知られている. 初めに最も粗くサンプリングした画素数の少ない画像を用いたテンプレートマッチング(大枠マッチング)を行い, ボールである可能性が高いと判定された領域に対してだけ, 段階的にサンプリング密度を上げていき精度の高いマッチング(精密マッチング)を行う. このようにすることで, 処理を行う候補ウィンドウ数を減らし, 計算量を削減している. また, サンプリング間隔を大きくするほど, 計算量は減少するが, ボール抽出の精度が低下する. そこで, 大枠マッチング時のサンプリング間隔Sを, 式(3)に示すよう

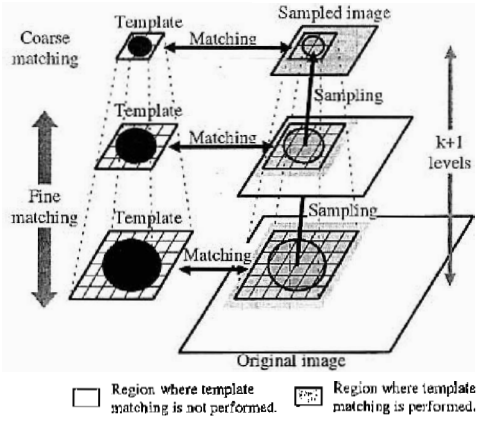


Fig. 4　階層的テンプレートマッチング

に，サンプリング後のボールの半径がR_{samp}より小さくならない範囲で最大にする．ただし，R_{samp}は実験により決定したボールと他の物体が判別可能なサンプリング画像上でのボールの最小半径，Rは抽出対象のボールの入力画像における半径である．$\lfloor x \rfloor$はxの整数部分を表すものとする．

$$\left.\begin{aligned} S &= 2^k \\ k &= \left\lfloor \log_2 \frac{R}{R_{samp}} \right\rfloor \end{aligned}\right\} \qquad (3)$$

これにより，Fig.5に示すように，入力画像でのボールの大きさが大きいボールに対しては粗く，小さいボールに対しては密にサンプリングを行うため，大枠マッチングでのボールの大きさがほぼ同じになり，入力画像上で異なる大きさのボールをほぼ同じ精度で抽出できる．精密マッチングでは，Fig.4に示すように，サンプリング間隔Sを順に$2^{k-1}, 2^{k-2}, \cdots, 1$となるように小さくしていく．したがって，最終的な抽出結果を得るまでに$k+1$階層の処理を行う．このように，サンプリング間隔を全て2のべき乗とすることで，3.3節で述べる，階層的並列メモリアクセスのためのメモリアロケーションを用い，ボール抽出VLSIプロセッサの面積最小化を実現することができる．

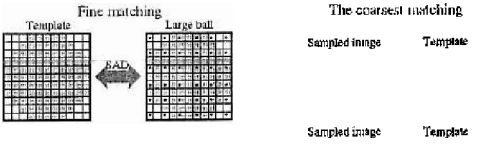


Fig. 5　階層的テンプレートマッチングのサンプリング密度

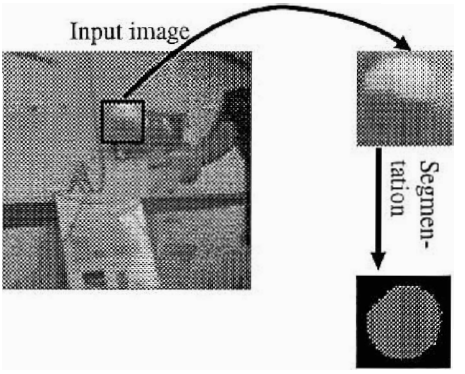


Fig. 6　階層的テンプレートマッチング結果

2.3　シミュレーション結果

階層的テンプレートマッチングに基づくボール抽出アルゴリズムのシミュレーションをワークステーション(SUN UltraSPARC-II 360MHz)を用いて行った．ただし，画像の大きさ$W \times H = 640 \times 480$，抽出するボールの半径の最小値$R_{min} = 8$，最大値$R_{max} = 31$，サンプリング後のボールの最小半径$R_{samp} = 4$，入力画像におけるRGB空間およびUCS空間のビット数を24ビットとして，シミュレーションを行った．このときの処理結果をFig.6に示す．また，この処理に要する計算時間は平均29秒となった．

Fig.1に示すようなボール捕球システムを想定し，その捕球成功率のシミュレーションを行った．ただし，マニピュレータは，2自由度の直角座標マニ

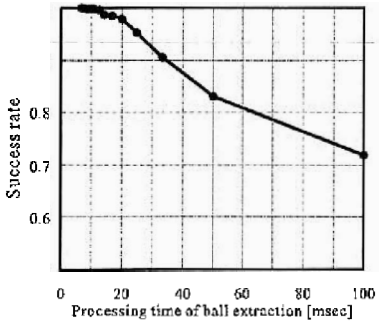


Fig. 7 シミュレーション結果

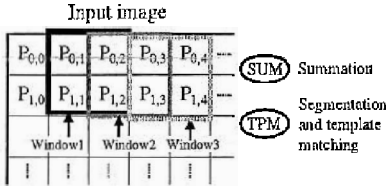
ピュレータで, 各軸の最大速度が1m/sec, 最大加速度が13m/sec^2, マニピュレータハンドの中心から5cm以内のボールを捕球できるものとする. また, ボールは, 人間が軽く投げた場合の速度に相当する4m/secで, マニピュレータの前方4mから, マニピュレータが捕球可能な1m×1mの範囲内に到達するように投げるものとする. 軌道予測においては, ボールは空気抵抗の影響を受けずに放物線軌道をとると仮定し, 最小2乗法を用いて, 過去のボールの軌跡に最も近い放物線軌道を予測軌道とした. 以上の条件でボールを10000回投げた場合のシミュレーションを行った. Fig.7にボール抽出時間と捕球成功率の関係を示す. このグラフから, ボール抽出時間が短くなるほど捕球成功率も高くなることがわかる. これは, 2.1節で述べたように, ボール抽出時間を短くし, ボールの3次元位置を多く取得するほど, 予測軌道が実際の軌道に近づくため, マニピュレータを効率よく実際のボール到達位置へと移動させることができるためである. また, グラフから, ボール抽出時間が10msec以下の時に捕球成功率が100%となることがわかる. そこで, ボールの3次元位置取得時間が10msec以内という制約条件で, ボール抽出VLSIプロセッサのチップ面積最小化を考える.

3. ボール抽出VLSIプロセッサのアーキテクチャ

3.1 全体の構成

同一階層での異なる候補ウィンドウに対するテンプレートマッチングのデータフローグラフをFig.8に示す. 同一階層の候補ウィンドウの出力間には依存関係が無く, 処理すべき候補ウィンドウ数は多い. このような場合には, パイプラインのセットアップ時間を無視できるため, 空間的並列処理だけではなく, パイプライン処理も全体の処理時間を減少するために有用となる. そこで, Fig.8に示すように, 1つの候補ウィンドウに対する処理を画素値総和計算(SUM), しきい値計算, 領域分割・テンプレートマッチング(TPM)の3ステージに分割しパイプライン処理を行う. また, 時間制約を満たすために, パイプライン処理に加えて, 画素値総和計算と領域分割・テンプレートマッチングの処理では候補ウィンドウ内の画素に対し空間的並列処理を行う.

Fig.9に, ボール抽出VLSIプロセッサの構成を示す. まず, イメージセンサから読み出した画素の色情報をRGB空間からUCS空間へ変換する. 並列処理のためには, ウィンドウ内の画素を演算部へ並列に供給する必要がある. そこで, 高速かつ並列アクセスを実現するために, 画像メモリを複数のメモリモジュールから構成し, 異なるメモリモジュールに記憶された画素に対する並列アクセスを可能としている. 画素値総和計算部では式(1)に示す平均の色度をFig.8に示すスケジューリングに従い空間的並列に計算する. しきい値計算部では領域分割のしきい値εを計算する. 領域分割・テンプレートマッチング部では, 領域分割を行い, その結果を用いて式(2)に示すテンプレートマッチングを並列に行う.



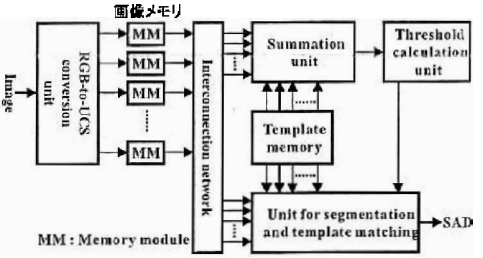


Fig. 9　VLSIプロセッサのブロック図

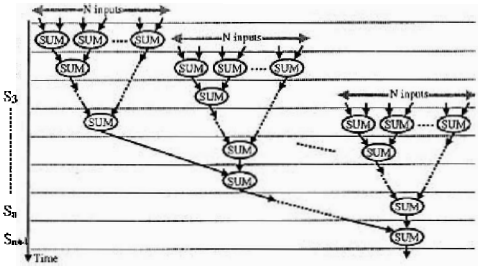


Fig. 10　1つの候補ウィンドウに対する画素値総和計算のデータフローグラフ

Spatially parallel processing

Fig. 8　ボール抽出処理のデータフローグラフ

3.2　演算部の構成

はじめに, 時間制約を満たす, 空間的並列処理に基づくスケジューリングを考える. ボール抽出の主要な処理であるテンプレートマッチングのデータフローグラフを, Fig.10に示す. Fig.10に示すように, 1ステップに処理を行う画素数Nを大きくするほど, 全体の処理時間は短くなる. そこで, 時間制約を満たす最小のNを求める. 演算部の最大動作周波数は, 最も語長の長い加算を行う, 累積加算器の最大動作周波数により決定される. また, 累積加算器で行う加算の語長は, Nによらず一定であるため, 演算部の最大動作周波数も, Nによらず一定となる. 0.35μm CMOS設計ルールを用いてボール抽出プロセッサを設計した場合の, 演算部の最大動作周波数は, HSPICEシミュレーションより, 100MHzとなった. 演算部の最大動作周波数100MHzと, 各大きさの候補ウィンドウの数から, 時間制約を満たすNを求めたところ, N=17×9画素となった.

N=17×9画素を並列に処理するような, 並列度の高い処理を行う場合, 通常, Fig.9に示すような構造のプロセッサでは, 相互結合回路網の面積が膨大になるという問題がある. そこで, Fig.11に示すように, 演算器数がメモリモジュール数と同数になるよう冗長に演算器を使用することで, メモリと演算器間の通信を局所化し, 相互結合回路網最小の構成にする.

3.3　階層的並列メモリアクセスのためのメモリアロケーション

Fig.8のスケジューリングに従って, 同一階層の候補ウィンドウに対する処理を空間的並列処理と

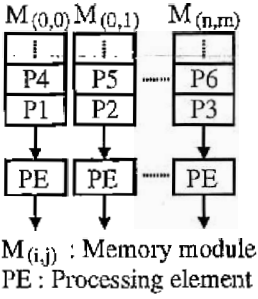


Fig. 11　相互結合回路網最小の構成

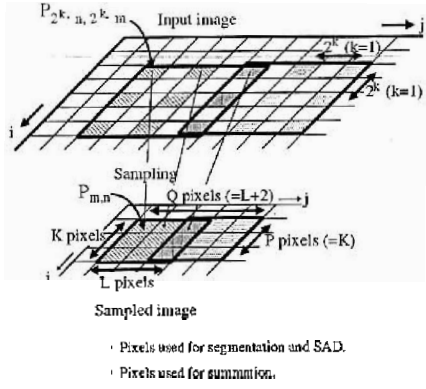


Fig. 12　階層的テンプレートマッチングのアクセスパターン

パイプライン処理で行う場合，画像メモリから演算部へ画素データを並列に供給する必要がある．画像メモリは，Fig.11に示すような，複数のシングルポートのメモリモジュールからなる構成を用いる．この構成では，異なるメモリモジュールの画素(Fig.11では$P1$,$P5$,$P3$など)は，並列にアクセスが可能である．したがって，完全並列アクセスを可能とするためには，並列に読み出す画素を異なるメモリモジュールに割り当てるメモリアロケーションが必要である．

並列に読み出すべき画素の集合(アクセスパターン)を考える．一例として，Fig.8に示すように，2×2の候補ウィンドウの処理を1×2画素ずつ並列に行う場合を考える．ステップS_5では，パイプライン処理によりWindow1の画素$(P_{0,1}, P_{0,2})$に対する領域分割・テンプレートマッチングの処理とWindow1から2画素右にあるWindow3の画素$(P_{0,3}, P_{0,4})$に対する画素値総和計算を並列に行うため，この場合のアクセスパターンは1×(2+2)の長方形領域$(P_{0,1}, P_{0,2}, P_{0,3}, P_{0,4})$となる．一般に，候補ウィンドウ内の画素を，K×L画素ずつ並列に処理を行う場合のアクセスパターンは，Fig.12に示すように，領域分割・テンプレートマッチングのための入力画素と画素値総和計算のための入力画素を合わせたK×(L+2)画素の長方形領域となる．

上述の並列に読み出すべき画素に対する並列メモリアクセスを，Fig.11に示す構成の画像メモリで実現するための，メモリアロケーションを考える．ただし，チップ面積最小化のために，以下の条件1～3をメモリアロケーションの条件とする．

条件1: 並列に読み出すべき画素が互いに異なるメモリモジュールに割り当てられる．すなわち，完全並列アクセスが可能である．並列に読み出すべき画素は，任意の2のべき乗のサンプリング間隔でサンプリングを行った画像の任意の位置におけるP×Qの長方形領域の画素とする．この並列に読み出すべき画素が存在する入力画像上での行および列の集合は，それぞれ式(5)および式(6)で表すことができる．そのため，Fig.12に示す，並列に読み出すべき画素の入力画像における集合は，式(4)で表すことができる．ただし，Zを整数全体の集合とする．

$$W = \{(i,j) \mid i \in I, j \in J\} \quad (4)$$

$$I = \{i \mid i = F_i(a), a \in A\} \quad (5)$$

$$J = \{j \mid j = F_j(b), b \in B\} \quad (6)$$

$$A = \{a \mid 0 \leq a < P, a \in \mathbf{Z}\} \quad (7)$$

$$B = \{b \mid 0 \leq b < Q, b \in \mathbf{Z}\} \quad (8)$$

$$F_i(x) = 2^k \cdot (m + x) \quad (9)$$

$$F_j(x) = 2^k \cdot (n + x) \quad (10)$$

条件2: メモリモジュール数が最小である．Fig.11に示す構成においては，演算器数をメモリモジュール数と同数にするため，この条件により，演算器数を最小化でき，演算部の面積最小化が実現できる．

条件3: 各画素は1個のメモリモジュールにだけ割り当てられる．これによりメモリ容量を最小化できる．

また，画素の位置からその画素が割り当てられているメモリモジュールを求めるための，アドレス発生回路が簡単になるという理由により，周期的なメモリアロケーションが望まれる．そこで，画像のi行j列の画素$P_{i,j}$を，式(11)に示す，アドレス生成関数$G((i,j))$により，メモリモジュール$M_{G((i,j))}$に割り当てるメモリアロケーションを考える．

$$G((i,j)) = (V(i), H(j)) \quad (11)$$

$$V(i) = i \bmod T_i \qquad (T_i \geqq 0, T_i \in \mathbf{Z}) \quad (12)$$

$$H(j) = j \bmod T_j \qquad (T_j \geqq 0, T_j \in \mathbf{Z}) \quad (13)$$

条件1〜3を満たし，式(11)で表される，最適なメモリアロケーションとして，周期T_iをP以上でありかつ最小の奇数，周期T_jをQ以上でありかつ最小の奇数とするメモリアロケーションを提案する．

一例として，$P = 3, Q = 5$の場合において，各画素を割り当てるメモリモジュールをFig.13に示す．Fig.13に示すメモリアロケーションの場合，サンプリング画像上の画素は，Fig.14に示すように，メモリモジュールに割り当てられる．Fig.14に示すように，このメモリアロケーションは，2のべき乗のサンプリング間隔でサンプリングされた画像上の，任意の位置において，3×5画素の並列メモリアクセスが可能となっているため，条件1を満たしている．1つのメモリモジュールからは同時に1画素だけを読み出せるため，P×Q=15画素を並列に読み出すためには，最低限P×Q=15個のメモリモジュールが必要となる．Fig.13に示す，メモリアロケーションにおいて必要となるメモリモジュール数は，その最低限必要な数と同数の15個であるため，メモリモジュール数は最小となっており，条件2も満たしている．また，各画素をどれか1つのメモリモジュールだけに割り当てているため，条件3も満たしている．したがって，Fig.13に示す，メモリアロケーションの一例は，$P = 3, Q = 5$の場合において，最適となっている．式(11)で表される，最適メモリアロケーションは，数学的にも最適であることを保証されている．

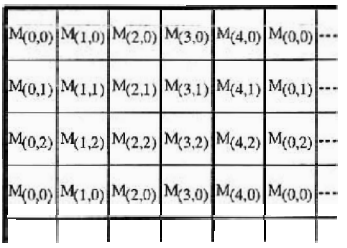


Fig. 13 階層的並列メモリアクセスのためのメモリアロケーション

3.4 評価

$W \times H = 640 \times 480$の大きさの画像から半径$R_{min} = 8$〜$R_{max} = 31$の大きさのボールを抽出できるボール抽出VLSIプロセッサの設計を0.35μm CMOS設計ルールで行った．そのチップレイアウトを，Fig.15に示す．HSPICEによりシミュレーションを行ったところ，その最大動作周波数は100MHz，ボール抽出時間は10msecとなった．提案手法によるメモリモジュール数最小化の効果を評価する．比較対象として，階層的でない画像処理プロセッサに使用されている，長方形メモリアロケーションを考

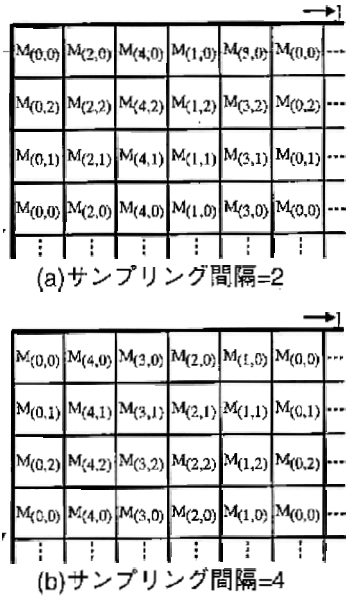


Fig. 14　サンプリング画像上でのアロケーション

える．長方形メモリアロケーションでは，並列にアクセスする画素を全て含む最小の長方形領域内を全て異なるメモリモジュールに記憶する．長方形メモリアロケーションを用いた場合，必要となるメモリモジュール数は，2409個となり，全体のチップ面積は984mm^{2}となる．それに対し，階層的並列メモリアクセスのためのメモリアロケーションを用いた場合，メモリモジュール数を8%(171個)，全体のチップ面積を12%(125mm^2)までに減少できる．

4.　まとめ

本稿では，階層的テンプレートマッチングにおいて必要となる階層的並列メモリアクセスのためのメモリアロケーションを提案した．本稿では，ある時刻における画像だけを用いてボール抽出を行ったが，過去のボールの動き情報を用いたボール抽出範囲の絞り込みを行うことにより，さらなる計算量の削減および軌道予測精度の向上が行えるものと考えられる．また，階層的な画像処理アルゴリズムは，画像理解などにおいても，計算量減少に有用である．本稿のプロセッサの構成法は，このような階層的な画像処理アルゴリズムをプロセッサ化する場合においても有用である．

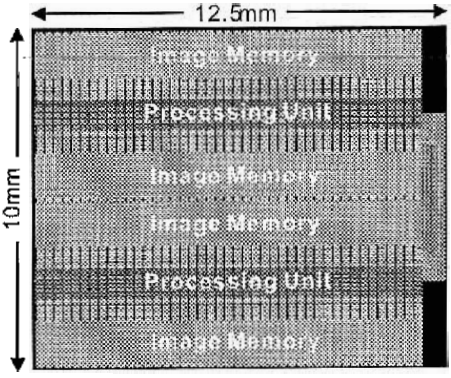


Fig. 15　ボール抽出VLSIプロセッサのレイアウト

参考文献

1) Masanori Hariyama, Hideki Kazama, Michitaka Kameyama, "VLSI Processor for Hierarchical Template Matching and Its Application to a Ball-Catching Robot System," , Proc. 2000 IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2000), pp.613-618, 2000.

2) M.Hariyama and M.Kameyama, "Design of a Collision Detection VLSI Processor Based on Minimization of Area-Time Products," Proc. IEEE International Conference on Robotics and Automation, pp.3691-3696, 1998.

3) 張山昌論, 李昇桓, 亀山充隆, "転送ボトルネックのないセンサ・メモリアーキテクチャに基づくモーションステレオVLSIプロセッサの構成," 電学論(E), Vol.120-E, No.5, pp.237-243, May, 2000.