

## オプティカルフローに基づく 高精度 3 次元計測 VLSI プロセッサの構成

### Design of a High-Precision Three-Dimensional Instrumentation VLSI Processor Based on Optical Flow

○李昇桓, 張山昌論, 亀山充隆

○Seunghwan Lee, Masanori Hariyama, Michitaka Kameyama

東北大学 大学院 情報科学研究科

Graduate School of Information Sciences, Tohoku University

キーワード: 専用 VLSI プロセッサ (special-purpose VLSI processor), 並列処理 (parallel processing), アレーブ  
ロセッサ (array processor), メモリインタリーブ方式 (memory interleaving)

連絡先: 〒980-77 仙台市青葉区 荒巻字 青葉東北大学 大学院 情報科学研究科 亀山研究室  
李昇桓, Tel.: (022)217-7155, Fax.: (022)263-9401, E-mail: seung@kameyama.ecei.tohoku.ac.jp

#### 1. はじめに

ロボットが未知環境で自律的に作業を遂行するためには、作業対象や障害物などの 3 次元距離情報を獲得することが重要である。従来の代表的な 3 次元計測方法としてはステレオビジョン<sup>1</sup>が挙げられる。この方法は三角測量の原理に基づいた方法であり、2 台のカメラで同時に撮った画像間の視差を利用する。計測精度は視点差に比例するが、視差が大きくなると画像間の変化が大きいため誤対応がとれる可能性が大きくなり、高精度な計測が困難であった。オプティカルフローに基づく計測は 1 台のカメラを用いるため、フレーム間のサンプリング間隔を減少することによってフレーム間の変化を少なくし、対応点探索の問題を容易にでき、演算量も減少できる利点がある<sup>2</sup>。さらに、マルチフレーム間のオプティカルフローを用いる

ことにより、視差が大きい画像間のオプティカルフローが得られ、3 次元計測の精度向上が可能である。

本稿では、サンプリング間隔の減少により高精度な計測を可能にする 3 次元計測 VLSI プロセッサを提案する。オプティカルフローの計算は相関演算の繰り返しであり、膨大な演算量と膨大な数の画素値を被演算子として必要とする。従来の並列アーキテクチャではフレームメモリと演算部間のデータ転送がボトルネックになる問題があった。提案する VLSI プロセッサでは空間的並列処理と時間的並列処理が融合され、相関演算が最大並列で処理される。フレームメモリから一回読み出した画素値はレジスタに記憶し、レジスタ間の転送のみで再利用できるため、フレームメモリへのアクセス回数を大幅に減少できる。さらに、フレームメモリを複数のバンクに分割し、各バンクから

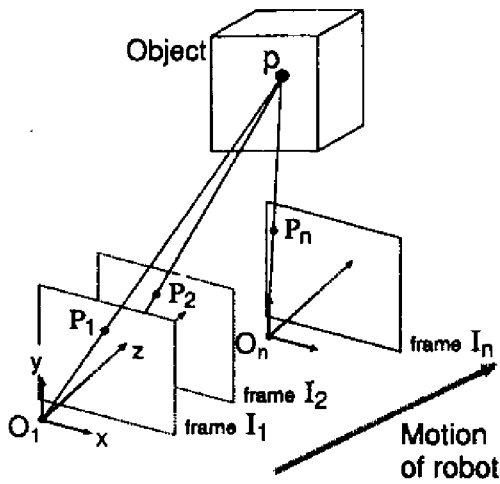


Fig. 1 Geometry of 3-D instrumentation.

並列読み出すことによりメモリ-演算部間のバンド幅を広げる。以上のアーキテクチャに基づいたプロセッサは2枚間のオプティカルフローの計算を52msで行ない、28MIPSワークステーションに比べ、2百万倍の性能が得られる。

## 2. 計測アルゴリズム

### 2.1 オプティカルフローに基づく3次元計測の原理

図1にオプティカルフローに基づく3次元計測の原理を示す。1台のCCDカメラが自律移動ロボットにマウントされ、移動速度と方向は既知であるとする。ロボットが移動すると、カメラから作業環境内の物体を異なる視点から見た一連の画像が得られる。時刻  $t = 1, 2, \dots, n$  に撮影したフレームを各々  $I_1, I_2, \dots, I_n$  とする。物体表面上のある物理的な点  $p$  の  $I_1, I_2, \dots, I_n$  への投影点を各々  $P_1, P_2, \dots, P_n$  とする。イメージ平面上での  $P_i$  から  $P_j$  へのベクトルをオプティカルフローと呼ぶ。2つのフレーム間のオプティカルフローが求まると、三角測量の原理に基づき物体の3次元座標が計算できる。

3次元計測の誤差には画像の離散化による誤差とオプティカルフローの計算における誤差があ

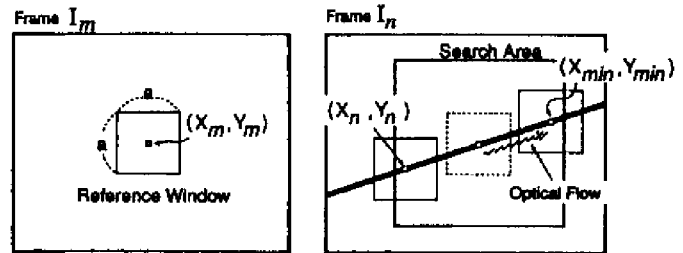


Fig. 2 Calculation of optical flow using MAD

る。画像の離散化による誤差は、フレーム間の視差を大きくすることにより減少できる。逆に、オプティカルフロー計算における誤差は視差を小さくし、画像間の変化を小さくすることにより減少できる。2枚のフレームを用いた3次元計測では以上のようなトレードオフにより高精度が困難となる。

### 2.2 オプティカルフローの計算

図2のように、フレーム  $I_m$  と  $I_n$  間のオプティカルフローは式(1)の相関演算を用いてフレーム間の対応点を計算することにより求められる。

$$MAD(X_m, Y_m, X_n, Y_n) =$$

$$\sum_{i=-a}^{a-1} \sum_{j=-a}^{a-1} |I_m(X_m+i, Y_m+j) - I_n(X_n+i, Y_n+j)| \quad (1)$$

ここで、 $(X_m, Y_m)$  はフレーム  $I_m$  上のオプティカルフローを求めるとするピクセルの座標、 $I_m(X_m, Y_m)$  はそのピクセルの画像値、 $a$  は参照ウインドの一辺の長さである。 $(X_n, Y_n)$  は対応点の候補ピクセルの座標であり、次のように限定される。 $(X_m, Y_m)$  の対応点は、図1の  $p, P_i, O_i$  を通る直線をフレーム  $I_{i+1}$  に投影したエピポーラ線<sup>3</sup>上にある。そのため、対応点の候補ピクセルをエピポーラ線上のピクセルに限定できる。エピポーラ線はフレーム間の位置関係によって決定され、既知のロボットの移動速度と方向から計算できる。

探索領域内のエピポーラ線上の全ピクセル  $(X_n, Y_n)$  に対して式(1)を計算する。その中から相関値が最小になるピクセル  $(X_{min}, Y_{min})$  を求め、 $(X_m,$

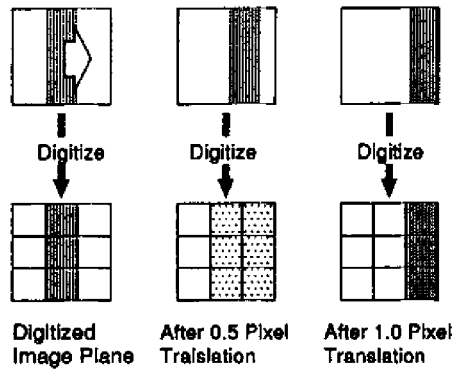


Fig. 3 Variation of brightness on a digitized image plane during translation.

$X_{min}, Y_m - Y_{min}$ ) を  $(X_m, Y_m)$  のオブティカルフローとする。

### 2.3 マルチフレームを用いたオブティカルフローの高精度化

視差が大きいフレーム間で、例えば、図1のフレーム  $I_1$  と  $I_n$  間の高精度なオブティカルフローを求めるため、視差が小さいフレーム間のオブティカルフローを求め、それらをベクトル加算することにより、最終的なオブティカルフローを得るアルゴリズムを提案する。カメラの移動は連続的であるにも関わらず、イメージ平面が離散化されているために、入力画像の劣化が生じる場合もある。この様子を図3に示す。各ピクセルの画素値はピクセルの中に投影された点の輝度の加算で決定される。そのため、画像上に投影された物体の移動がピクセルの一辺の長さの整数倍にならない場合には画像値が変化し、対応点であるにも関わらず相関値が最小にならない場合がある。この問題を解決するためには移動がピクセルの整数倍になるフレーム間のオブティカルフローを求めるのが望ましい。実験上では移動が整数倍になるフレームでは整数倍でないフレームより相関値が小さい傾向があるため、フレーム  $I_i$  と  $n$  枚のフレーム間の相関を求め、その中から相関値が最小になるフレーム間のオブティカルフローが離散化による精度低

下が最も少ないと見なす。そのオブティカルフローをベクトル加算し、 $n$  枚間のオブティカルフローを求めるアルゴリズムを提案する。

[オブティカルフロー計算アルゴリズム]

Step 1  $i = 1$

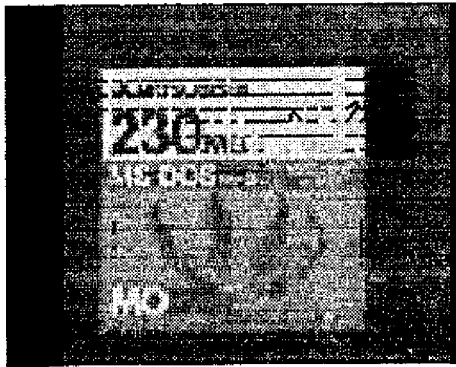
Step 2 フレーム  $I_i$  と  $I_j$ , ( $j = 2, \dots, n$ ) 間のオブティカルフローを計算し、相関演算値が最小になるフレーム  $I_k$ ,  $i \leq k \leq j$  までのオブティカルフロー  $V_k$  を求める。

Step 3  $k \neq n$  ならば、 $i = k$  として Step 2 を繰り返す。 $k = n$  ならば、これまで求めた  $V_k$  をベクトル加算し、フレーム  $I_1$  と  $I_n$  間のオブティカルフローとする。

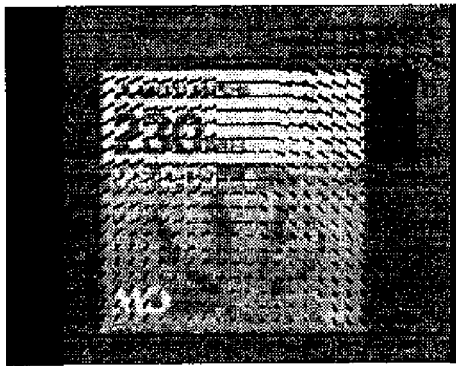
以上の方法により計算したオブティカルフローの例を図4に示す。物体をイメージ平面に垂直な軸の回りに回転している例である。図4の(a)は連続的な2枚のフレーム間のオブティカルフローを計算し、それらをベクトル加算した結果である。図4の(b)は相関値が最小になるフレーム間のオブティカルフローをベクトル加算した結果であり、図4の(a)より精度よく計算されていることがわかる。

### 3. プロセッサアーキテクチャ

図5に提案する3次元計測VLSIプロセッサの構成を示す。 $n$  個のフレームはフレームメモリ FM1 から  $n$  に記憶され、フレームの画素値は相関演算部に転送する。Step 2 のフレーム間のオブティカルフロー計算を並列に行なうために、複数の相関演算部を内蔵している。フレーム  $I_k$  と  $I_i$  間のオブティカルフローは配線された相関部で行ない、最小の相関値とオブティカルフローを最小値判定部に転送する。最小値判定部に転送されてきた相関値の中から最小オブティカルフローを選び、 $n$  フレーム間のオブティカルフローをベクトル加算



(a) Optical flow calculated between successive frames and summed from frame 1 to 10



(b) Optical flow calculated by the proposed algorithm

Fig. 4 Simulation result of optical flow calculation.

し、最終的なオプティカルフローを計算する。距離計算プロセッサはそのオプティカルフローを入力データとし、距離を計算する。

### 3.1 相関演算部の構成

図 7と 8に相関演算部の構成と処理要素 (PE) の構成を示す。処理時間を減少するため、 $n \times n$  のウィンドに対して、 $n \times n$  個の PE を備えピクセル間の絶対差を並列に計算する。さらに、各 PE ごとに加算器を備え、それらをパイプライン化し累積加算することによりスループットを向上させる。配線による面積と PE 間データ転送遅延時間を減少するため、行方向に中間和を転送しながら累積加算する。さらに、各行の中間和を同様に累積加算する。

提案する、演算部は並列処理とパイプライン化

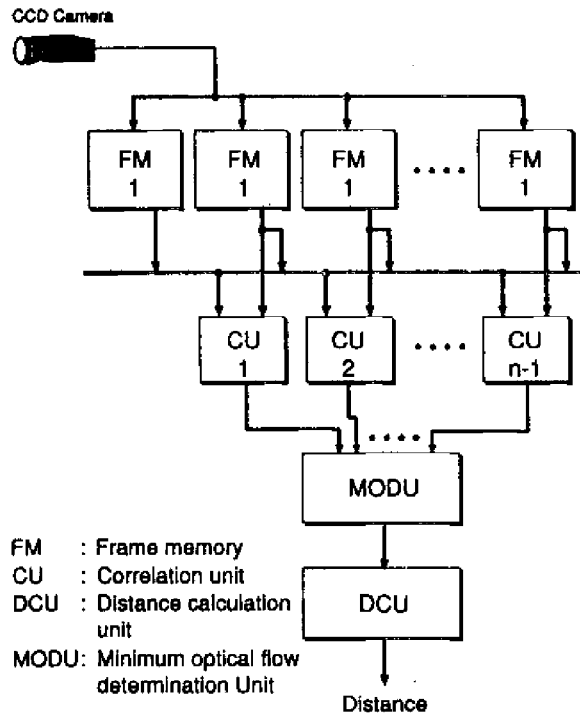


Fig. 5 Structure of the 3-D instrumentation VLSI processor.

により処理時間は減少できるが、毎クロック  $n \times n$  個の画素値を演算部に転送する必要がある。フレームメモリのアクセス時間は高速 SRAM の場合、1ワード当り数十 ns になるため、数 ns の遅延時間を持つ加算器の稼働率が低下する。

そのため、図 6 のように各ウィンド間にオーバーラップする領域に着目し、その領域の画素値を再利用することにより毎クロックメモリから読み出す画素値数を  $2n - 1$  個まで減少できる。2.2 節で述べたようにエピポーラ線はフレーム間の位置関係、つまり、ロボットの移動速度と方向より決定され、傾きは任意である。したがって、画素値の再利用のためにはレジスタに記憶されたデータの更新のための制御や配線が問題になる。

ウィンドの中心  $(X_n, Y_n)$  をエピポーラ線上で移動しながら相関演算する場合、ウィンドの移動は斜め、水平あるいは垂直方向に 1 ピクセル移動される。図 6 はウィンドが斜め方向に移動した場合を示す。各レジスタをウィンド内の各ピクセル

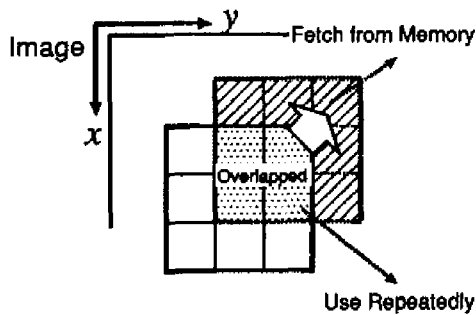


Fig. 6 Example of windows.

に対応させ、ウィンドの移動方向に従ってレジスタ間の転送のみで画素値の再利用ができる。さらに、各レジスタ間のデータ転送は同じ方向であるため、レジスタ間に配線の供有ができ、配線量が減少される。以上の考えに基づき  $n \times n$  個レジスタを2次元アレー状に並べ、どの方向のウィンドの移動にも対応し各レジスタに記憶された画素値がレジスタ間に移動するように配線したキャッシュレジスタアレーを提案する。ウィンドの1ピクセル移動するため、隣接レジスタ間のデータ転送のみでPEにデータを供給できる。配線のオーバーヘッドを減少するため、各レジスタに一個のPEを配線する。

### 3.2 フレームメモリの構成

ウィンドが移動する場合、オーバーラップしない領域は図6に示すように同じ行又は列なる。どの場合でも並列アクセスできるように次のような対角インタリーブ方式<sup>2</sup>を利用する。図9にフレームメモリの構成と対角インタリーブ方式を示す。同じ対角線上の画素値は一つのバンクに記憶することによって同じ行、列上の画素値が異なるバンクに均等に分散記憶される。その結果、行、列、行と列のアクセスに対して並列アクセスができ、 $2n-1$  個の画素値を1メモリアクセス時間で同時に読み出される。

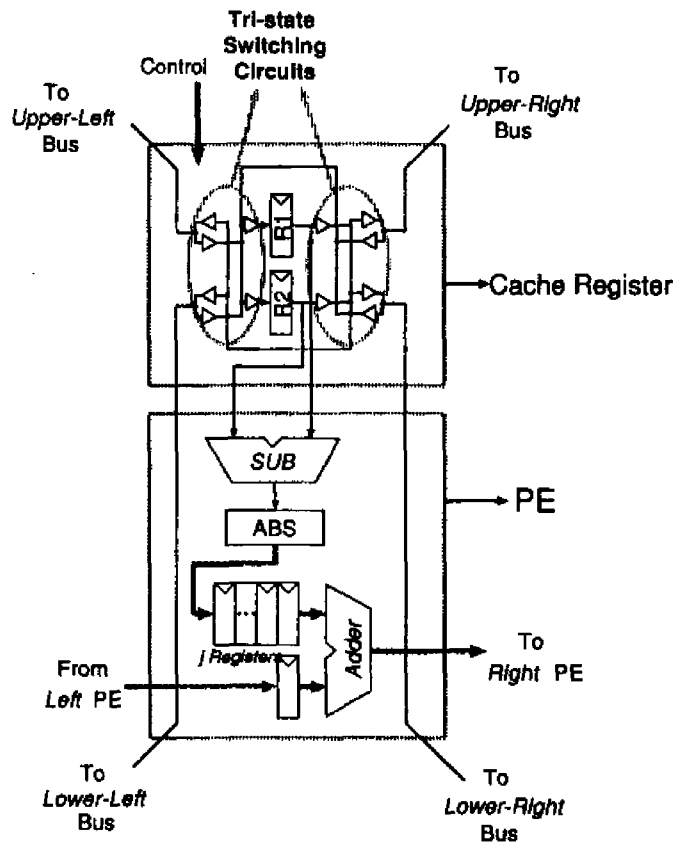


Fig. 8 Structure of the PE.

## 4. 評価

式(2)に示すように2フレーム間のオプティカルフロー計算時間  $T_{corr}$  は  $2a^2$  個の画素値をキャッシュレジスタアレーにロードする時間と  $p \times p$  の探索領域に対する相関演算時間の和とピクセル数  $M \times N$  の積である。

$$T_{corr} = (2a^2 t_{eff} + 2p^2 T_{CLK}) MN \quad (2)$$

ここで、 $t_{eff}$  は対角インタリーブ方式メモリのワード当たり平均アクセス時間であり、 $m$  個のメモリバンクを用いた場合、 $t_{eff} = t_{mem}/m$  である。 $t_{mem}$  は各メモリバンクのアクセス時間である。

$T_{img}$  を画像のサンプリング間隔とすると、 $n$  フレームの撮影から3次元計測の終了までの最大遅延時間  $T_{inst}$  は式(3)で表される。

$$T_{inst} = nT_{img} + (n-1)T_{corr} + T_{dist} \quad (3)$$

ここで、 $T_{dist}$  は距離計算時間である。

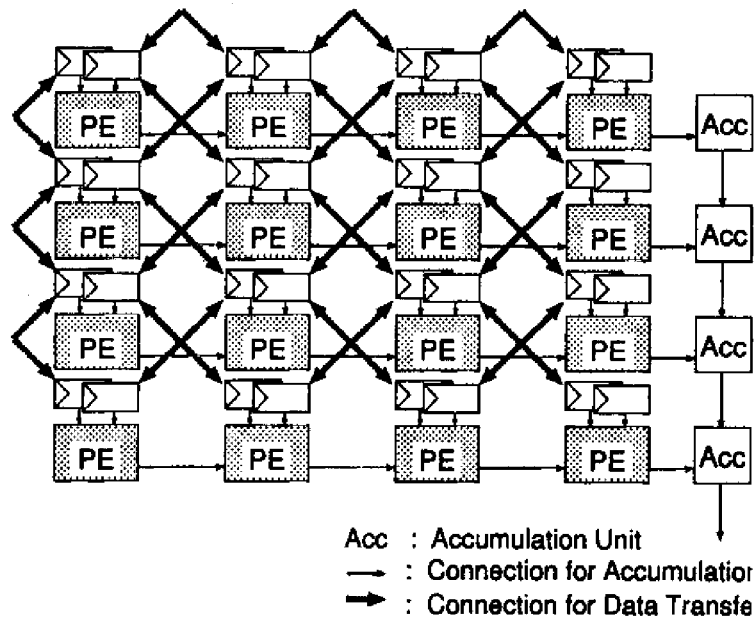


Fig. 7 Structure of the CU.

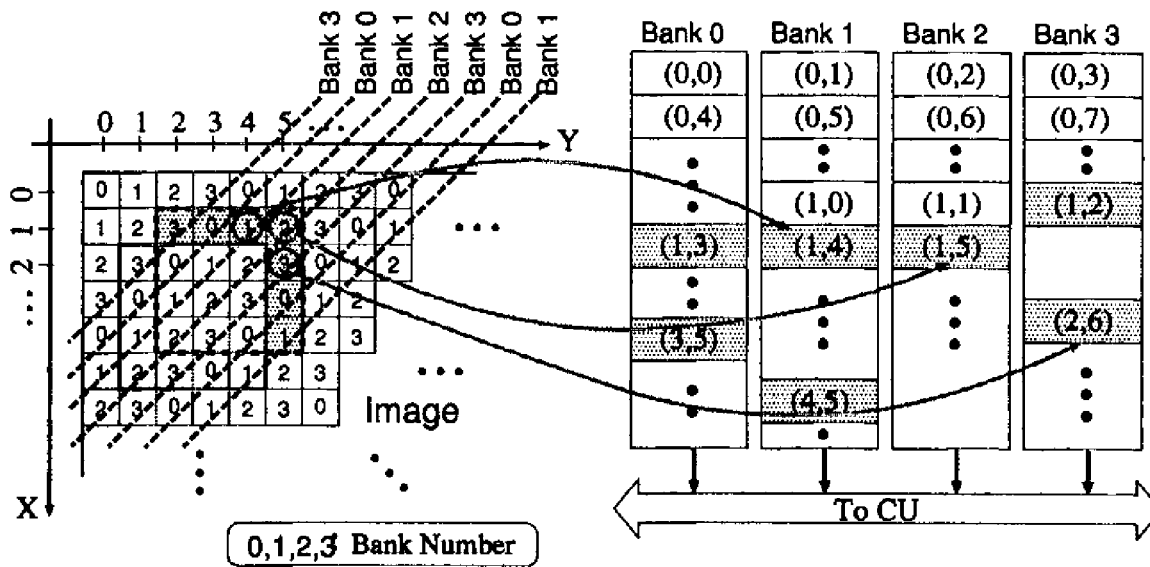


Fig. 9 Diagonal interleaving.

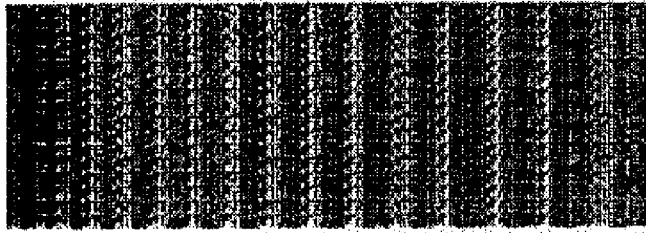


Fig. 10 Layout of the CU.

Table 1 Features of the VLSI Processor

Technology	0.8 $\mu$ Double-Metal CMOS
Chip Size	7.3 $\times$ 2.6mm <sup>2</sup>
Number of Tr.	275K
Number of PE	16 $\times$ 16

30 フレーム/秒の性能の CCD カメラを用いると  $T_{img}$  は 33ms になる。図 10 に 0.8 $\mu$ m CMOS 設計ルールを用いた相関演算 VLSI プロセッサのレイアウト、表 1 に諸元を示す。SPICE シミュレーションにより、 $T_{CLK}=20ns$  と見積られ、式 (2) より  $T_{coor} = 520ms$  と算出される。同様に、 $T_{dist} = 320ns$  と見積られる。式 (3) により  $T_{dist} = 0.9sec$  と算出され、28MIPS のワークステーションより 2 百万倍の高速化が達成される。

## 5. むすび

本稿では、イメージ平面の離散化によりオブティカルフローが低下する問題を、マルチフレーム間で相関演算値が最小になるオブティカルフローを求めることにより高精度なオブティカルフローを計算するアルゴリズムを提案した。また、空間的並列と時間的並列処理を融合したアーキテクチャを提案し、相関演算の処理時間を減少させた。対角インタリーブ方式並列メモリと同時データ転送するキャッシュレジスタアレーによりフレームメモリと演算部間のデータ転送時間を減少するメモリアーキテクチャについて述べた。提案する VLSI プ

ロセッサは最大 0.9 秒で 3 次元計測を行ない、数秒単位の計測が要求される自律ロボットの衝突チェックに十分に応用可能である。

## 参考文献

- 1) R.M. Haralick and L.G. Shapiro: Computer and Robot Vision vol.2, 156/160, Addison-Wesley Publishing, Inc.(1993).
- 2) S. Lee, M. Hariyama and M. Kameyama: High-Performance VLSI Architecture for Three-Dimensional Instrumentation Based on New Concurrent Memory-Access Scheme, APCCAS'96, 500/503, (1996).
- 3) M.Suk and S.M.Bhandarkar: Three-Dimensional Object Recognition from Range Images, Springer-Verlag, Tokyo, 25/30, (1992).