

## 実時間視覚処理のためのビジョンチップシステムの開発

鏡 慎吾<sup>†</sup> 小室 孝<sup>††</sup> 石井 抱<sup>†††</sup> 石川 正俊<sup>††</sup>

Development of a Vision Chip System for Real-Time Visual Processing

Shingo KAGAMI<sup>†</sup>, Takashi KOMURO<sup>††</sup>, Idaku ISHII<sup>†††</sup>, and Masatoshi ISHIKAWA<sup>††</sup>

あらまし 本論文では、汎用ビジョンチップを用いた実時間視覚処理システムのアーキテクチャと試作システムについて述べる。設計されたビジョンチップコントローラは、ビジョンチップにインストラクションを供給するとともに処理結果の情報を取得し、それをフィードバックするといった一連の処理をボトルネックなく行う。新しく提案するストールの発生しないパイプラインアーキテクチャ SPARSIS (Stall-less Pipeline Architecture for Real-time Sensory Information processing System) の採用により、高い時間分解能での実時間性を保証することができる。試作されたシステムはクロック周波数 40 MHz で動作し、100 ns の時間分解能を実現した。また実際にビジョンチップを実装して視覚処理実験を行った。

キーワード ビジョンチップ, 高速ビジョン, 並列処理, 実時間処理

## 1. ま え が き

ロボットビジョン, マルチメディア, マン・マシンインタフェースなど, 高速かつ汎用なビジョンシステムへの需要が高まる中, 光検出器 (PD) と処理要素 (PE) を画素ごとに直結してワンチップ上に集積することによって画像入力におけるボトルネックを取り除き, 高速な視覚を実現するビジョンチップの研究が注目を集めている。

これまでは主としてその集積化のしやすさから, アナログ回路を用いて固定の処理を行うビジョンチップが数多く開発されてきた [1]。しかし近年の半導体集積技術の進歩をそのよりどころとして, デジタル回路を PE として採用したプログラマブルなビジョンチップの研究も行われている。例えば小室らは PE に汎用な処理回路を用いながら高集積化を目指した S<sup>3</sup>PE (Simple and Smart, Sensory Processing Element)

アーキテクチャを提案している [2]。S<sup>3</sup>PE では外部からインストラクション列を与えることにより, 高速でなおかつ汎用な視覚情報処理を実現できる。

このような汎用ビジョンチップの実際の利用に際しては, ビジョンチップにインストラクションを供給するとともに処理結果の情報を取得し, それをフィードバックするといった一連の処理をいかにボトルネックなく行うかが重要となる。今後様々な分野へビジョンチップの応用を広げて行く上で, その潜在的能力を最大限に引き出すことを可能とするためには, デバイスとしてのビジョンチップの研究のみではなく, 実際のアプリケーションとのインタフェースを含めたシステム全体について検討を行う必要がある。

実時間で視覚処理を行うシステムはこれまでも多数のものが提案されている [3]~[5]。しかしそのほとんどが実時間性として暗にビデオフレームレート (NTSC で 30 frames/s) を仮定しており, ビジョンチップでの処理に要求されるような, 例えば 1000 frames/s といったレートでの処理や入出力, あるいは PD を能動的に制御するといった場面で課せられる, ビジョンチップ特有の更に厳しい時間制約については考慮されていない。一方, 他の汎用ビジョンチップの例を見ても中心となるのはデバイスの研究であり [6], [7], システムとしての在り方についての研究はあまり見られない。

我々は以上のようなことを踏まえ, S<sup>3</sup>PE アーキテ

<sup>†</sup> 東京大学大学院工学系研究科, 東京都  
Graduate School of Engineering, The University of Tokyo,  
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656 Japan

<sup>††</sup> 東京大学大学院情報理工学系研究科, 東京都  
Graduate School of Information Science and Technology,  
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo,  
113-8656 Japan

<sup>†††</sup> 東京農工大学工学部, 小金井市  
Faculty of Technology, Tokyo University of Agriculture  
and Technology, 2-24-16 Naka-cho, Koganei-shi, 184-8588  
Japan

クチャのビジョンチップを用いて、実際に様々なアプリケーションシステムと接続して使用することを目的とした実時間視覚処理システムの開発を行った。本論文では、その設計方針とアーキテクチャ設計、及び試作システムの開発について述べる。

## 2. ビジョンチップとその処理構造

### 2.1 ビジョンチップアーキテクチャ S<sup>3</sup>PE

S<sup>3</sup>PE は、例えばビジュアルフィードバックを用いたロボット制御などに十分な 1000 frames/s 以上のフレームレートでの汎用な視覚処理を実現することを目指したビジョンチップアーキテクチャである [2]。

図 1 にその概略を示す。画素ごとに配置された PE が外部から与えられたインストラクション列に従って SIMD 型制御で動作する構造で、各 PE はビットシリアル ALU と 24 ビットのローカルメモリをもち、4 近傍の PE と直接通信可能である。回路的、アーキテクチャ的な様々な工夫により、汎用性を確保しながら回路面積を小さく抑えている。

画像入力の A-D 変換もインストラクションに従って画素ごとに行う。蓄積された電荷が光電流によって放電されることにより電位がしきい値を切るまでの時間をソフトウェアでカウントする形になっており、専用の A-D 変換回路を用意するのに比べて回路規模を小さくできるのみならず、環境や目的に応じた柔軟な処理を画像取得の段階から行うことができる。

現在 S<sup>3</sup>PE に基づいたビジョンチップとして、0.35 μm CMOS プロセスにて試作された 16 × 16 及び 24 × 24 画素のテストチップが動作確認されている [8], [9]。同じ技術では 64 × 64 画素が 8mm 角のチップに集積可能であり、我々の当面の技術目標で

ある。

### 2.2 ビジョンチップの処理モデル

ビジョンチップを用いて処理を行う場合の一般的な処理モデルは図 2 のように考えられる。

- (1) 画像獲得
- (2) 画素ごとの SIMD 型並列演算
- (3) 特徴量抽出
- (4) 画素並列性をもたない逐次的・大域的演算
- (5) 特徴量のアプリケーションでの利用
- (6) (1)(2) へのフィードバック

例えばビジュアルフィードバックを用いたロボットの制御といった場面を考えると (1)(2) で取得した画像の前処理や対象の抽出など、画素ごとのデータ並列性を生かした処理が高速に行われ (3)(4) で対象の重心などの特徴量が抽出される。これらの情報は、必要に応じて並列度の低い処理を経て (5) でロボットへの指令値として用いられるのと同時に (6) とし

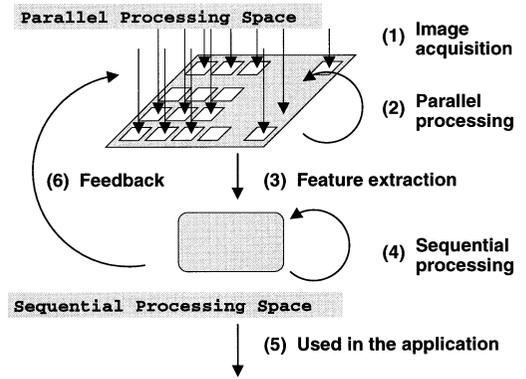


図 2 ビジョンチップの処理モデル  
Fig. 2 Processing model with vision chip.

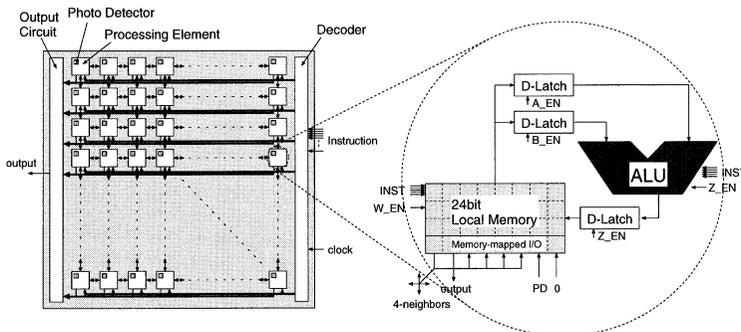


図 1 S<sup>3</sup>PE アーキテクチャ  
Fig. 1 S<sup>3</sup>PE architecture.

て A-D 変換や前処理のパラメータの更新や処理内容の変更といった形でビジョンチップにフィードバックすることにより、適応的・能動的な視覚処理を実現できる。

### 2.3 ビジョンチップシステム

ビジョンチップを様々なアプリケーションシステムに応用していく際、その能力を最大限に生かすためには、前節で述べたような一連の処理をボトルネックなく実現することが要求される。これをそれぞれのアプリケーション自身が行うのは大きなオーバーヘッドであり、また、アプリケーションシステムの処理能力、I/O の速度によってビジョンチップの動作が制限されてしまう。

これを解決するため、ビジョンチップとそれ専用の制御・処理系を統合した視覚処理システム — ビジョンチップシステムを導入する。ビジョンチップへのインストラクションは、システム内のコントローラにより安定して供給される。ビジョンチップの処理結果はコントローラが直ちに受け取り、特徴量への圧縮、逐次的処理を経てアプリケーション側へ伝達する。プログラム制御もアプリケーションシステムからは独立して行われ、処理結果に応じて遅延なくビジョンチップの制御に反映することを可能とする。

これにより MHz オーダ以上の高速な動作はビジョンチップシステムの内部の閉じた環境で行われ、例えばアプリケーションシステムとの間の情報交換はフレームレート程度 (kHz オーダ) の比較的遅いレートで行うことで、通信ボトルネックによる性能の低下や、アプリケーション側のオーバーヘッドを回避できる。

我々はこれら各機能を実現するビジョンチップコントローラの設計を行い、それを核としてビジョンチップシステムの開発を行った。以下ではそのアーキテクチャの詳細について述べる。

## 3. SPARSIS アーキテクチャ

### 3.1 設計方針

ビジョンチップシステムにおいて時間的に最もクリティカルなのはインストラクションの供給である。これには以下に述べるような時間的制約が課せられる。

一つはフレーム時間の制約である。例えばフレーム時間 1ms といった高速な視覚処理を実現するためには、インストラクションがビジョンチップに十分高速に供給され、各フレームごとの処理がその時間内に確実に完了することが保証されなくてはならない。これ

はビジョンチップシステムが必要とする実時間性の最も端的な例である。

それに加えて、入力画像の A-D 変換のために PD の出力を読み出すなど、物理量のセンシングを直接制御するインストラクションの実行タイミングについては、これよりもはるかに厳しい時間的制約が課せられる。2.1 で述べたとおり、A-D 変換は PD の出力がしきい値を切るまでの時間をカウントすることで行われる。例えば光蓄積時間 1ms で 8 ビットの A-D 変換を行うためには、1ms の間に 255 回の PD 出力読出しを行う必要があるが、その各読出しタイミングが完全に決定的でなければ、A-D 変換の結果はアプリケーションの意図するものとは異なってしまう。更にダイナミックレンジの拡大や環境への適応のために PD 出力読出しタイミングを動的に変化させることを考えると、ビジョンチップへのインストラクションの供給には極めて高い時間分解能での実時間性が要求されるといえる。

インストラクション供給の時間分解能を上げるためには、パイプライン化による単位時間当りのインストラクション発行数の向上が有効である。しかし、通常のマイクロプロセッサにおけるパイプライン設計 [10] では動的にストールが発生し、ビジョンチップが要求するインストラクション発行タイミングの実時間性を満たすことができない。

実行時間の動的な変動を排除した設計の試みは既に見られ、例えば高性能計算を指向した並列処理用プロセッサにおいて、プロセッサ間通信の同期の問題を解決するために、静的スケジューリングの可能なパイプラインアーキテクチャが提案されている [11]。しかし、我々の目的であるビジョンチップの細粒度の実時間制御を行うためには、インストラクション・入力データの供給や出力データの取得といった制御フローもパイプライン化する必要があり、逐次処理との相互間の干渉も考慮に入れる必要が生じるため、既存の手法を単純に適用するだけでは不十分である。

我々はビジョンチップコントローラのコアアーキテクチャとして、ビジョンチップの制御フローをパイプライン化し、逐次処理を行う RISC 型整数パイプラインに同期してそれを動作させ、なおかつ全体としてパイプラインストールが一切生じないよう留意して設計されたアーキテクチャ SPARSIS (Stall-less Pipeline Architecture for Real-time Sensory Information processing System) を提案する。

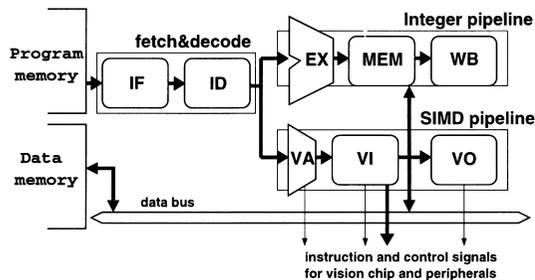


図3 SPARSISのパイプライン構成  
Fig. 3 Diagram of the SPARSIS pipeline.

### 3.2 SPARSISのパイプライン構成

SPARSISは図3に示すように整数パイプラインとSIMDパイプラインの2本のパイプラインをもつ。プログラムメモリから命令がフェッチされデコードされると、整数演算命令は整数パイプラインへ、ビジョンチップ上でのSIMD演算命令はSIMDパイプラインへと送られる。整数パイプラインはビジョンチップの演算結果を用いてアプリケーションが要求するデータを生成するほか、分岐条件やループカウンタなどプログラム制御に必要な値の演算等を行う。

整数パイプラインは、画像特徴量としてモーメント量などを扱うのに十分な32ビット幅のデータバスを有する。パイプラインの制御からは、例えば分岐予測などの投機的要素は一切排除されている。分岐命令やロード命令といった、フォワーディングでハザードの回避できない命令については遅延スロットを設け、ストールは全く生じない構成となっている。コンパイラが遅延スロットを埋められない場合は無効命令が挿入されるが、これはコンパイル時に確定するため実時間性を損なわない。これにより毎サイクル必ず1命令が発行され、内蔵されたタイマカウンタなどと組み合わせることで、通常のマイクロプロセッサとリアルタイムOSの組合せでは実現し得ないインストラクションレベルでの実時間性保証が可能となる。

命令セットは、汎用レジスタ16本をもつRISC型のロード/ストアアーキテクチャである。命令スケジューリングのしやすさが、センサを制御するインストラクションの実時間性保証に貢献できる。

主記憶は、プログラムメモリに関しては画像処理は短いフレームの繰返しである点やフレーム内でも同じ処理の繰返しが多い点、データメモリに関しては原則として画像特徴量としてのスカラー量を扱う点などを考

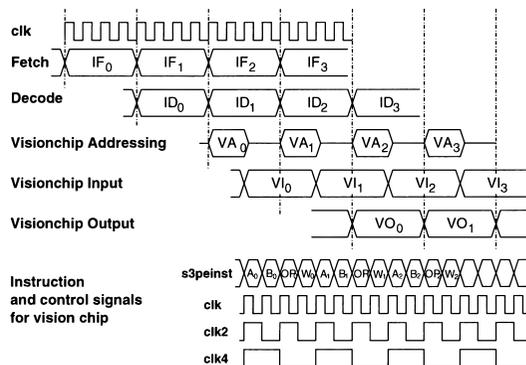


図4 SIMDパイプラインの動作タイミング  
Fig. 4 Timing chart of the SIMD pipeline.

慮し、容量を限定した上でアクセスのレイテンシが常に一定となるよう構成されている。

ビジョンチップへ送られるインストラクションは、デコードされると直ちにSIMDパイプラインに発行される。SIMDパイプラインはVA, VI, VOの各ステージからなる。入力信号・インストラクションの前処理をVAで行った後VIでビジョンチップへ送り、必要な場合はVOで後述する出力ユニットを制御し、出力データの処理を行う。

このようにビジョンチップの制御もパイプライン化されて行われるため、整数演算の結果をVAステージでの前処理に用いる場合やビジョンチップの出力を整数演算の入力とする場合など、整数/SIMD両パイプライン間でデータハザードが生じ得る。これを回避するため、整数パイプライン内でのデータフォワーディングと同様に、パイプライン相互間についてもハザードの起こり得るステージ間でフォワーディングを行う。

S<sup>3</sup>PEは1インストラクションが4サイクルからなるため、整数パイプラインの1ステージ分のサイクルタイムを4分割し、4回に分けてインストラクションバスに送出する。SIMDパイプラインの動作タイミングを図4に示す。ビジョンチップの各PEは送られたインストラクションに従ってSIMD型で並列に実行する。

### 3.3 ビジョンチップのアドレッシング

VAステージでは送出するインストラクションの前処理が行われる。S<sup>3</sup>PEの場合、インストラクションはすべて3オペランド形式で、各オペランドでローカルメモリ内の絶対アドレスを指定する[2]。そのため、ビジョンチップへ送るインストラクションをそのまま

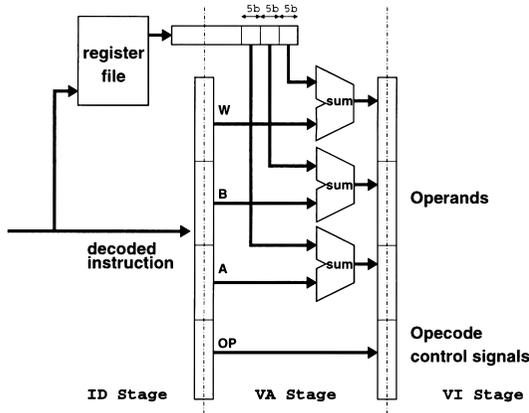


図5 S<sup>3</sup>PE インストラクションのアドレッシング  
Fig.5 Displacement addressing for vision chip instruction.

の形でプログラムメモリ内に格納していたのでは、サブルーチンの汎用性がなくなってしまう。

コンパイル時に解決可能なのであれば、すべて展開してプログラムメモリに格納することで対応可能であるが、前述したように主記憶の容量は限定されているため、プログラムの肥大化は許容できない。また、動的な処理内容の変化への対応もできなくなる。

そのため、汎用レジスタの内容で各オペランドに独立なオフセットを与えるディスプレースメントアドレッシングをVAステージで行う。これを図5に示す。IDステージでフェッチした汎用レジスタの内容のうち下位15ビットが5ビットずつ、ビジョンチップに与える3オペランドへのオフセットとして用いられる。汎用レジスタを用いることで柔軟性が高くなり、また、命令間の依存も少なくなると期待できる。

これをアドレス計算と考えれば、ビジョンチップは整数パイプラインのメモリアクセスと同じタイミングで制御されていると見ることができる。整数パイプラインからはビジョンチップとの入出力をメモリマップとして扱い、また必要に応じて整数・SIMDパイプライン間でフォーワーディングを行うことで、データハザードによるストールを回避することができる。

#### 4. ビジョンチップシステムの構成

##### 4.1 全体の構成

システム全体の構成を図6に示す。コントローラコアは、SPARSISに基づく32ビットのマイクロコントローラである。ビジョンチップとの間のデータ入出力

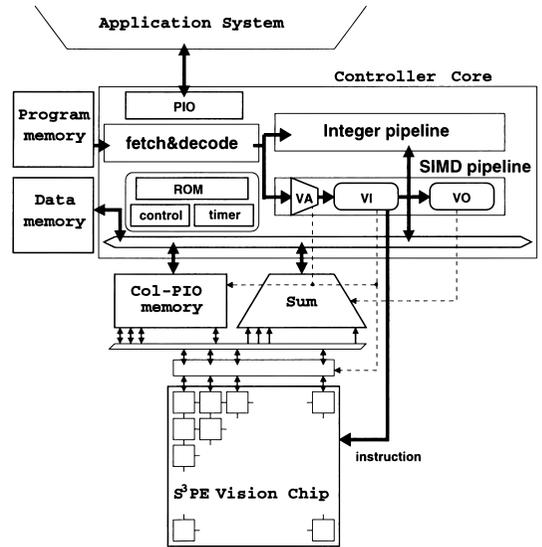


図6 システム全体の構成  
Fig.6 Diagram of the designed vision chip system.

インタフェースとしては、画像から特徴量を抽出するための総和出力ユニットや、初期化やモニタリングのための列並列入出力ユニットを備える。外部システムとの通信はパラレルI/Oインタフェースを通して共有メモリ型で行う。

現在これらの各機能は、FPGAを用いて実現している。コントローラに必要とされる処理演算機能などはアプリケーションによっても大きく異なり、ソフトウェア資産の蓄積も未だ十分ではないため設計初期の段階から完全に固定することは困難であるが、インシステムプログラミングが可能なデバイスを用いることで、命令セットアーキテクチャの拡張などを順次行っていくことや、異なるバージョンのビジョンチップへの対応などが可能となった。

##### 4.2 入出力インタフェース

###### 4.2.1 総和出力ユニット

処理結果出力におけるボトルネックを取り除くためには、2次元パターン情報をそのまま取り出さず、アプリケーションが必要とする特徴量のレベルまで圧縮した形で情報を取り出すことが有効である。画像から抽出される特徴量として、ここでは特にロボットビジョンを主なアプリケーションと考え、ビジュアルサーボなどに有効な特徴量として像の面積や重心、分散などのモーメント情報を考える。これは画素ごとの重み付けと画像全体の総和計算を組み合わせることで効率良

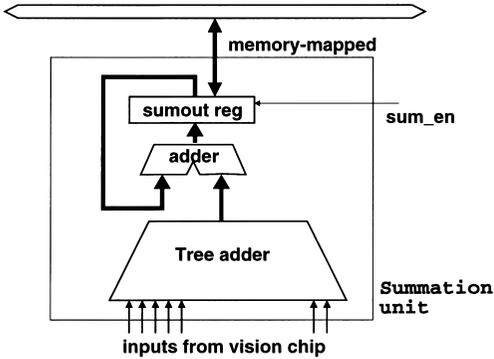


図 7 総和出力ユニット  
Fig. 7 Diagram of the summation unit.

く求められる。

総和出力回路の構成を図 7 に示す。画像をシフトしながら順次出力し、外周 PE からの出力を入力とするトリ型加算器によって計算された総和を累積していく形で、画像全体の総和を計算する。

この方法では、画素数  $N \times N$  に対して  $O(N)$  のステップ数を総和出力に要してしまうが、ビジョンチップとともに集積することを前提とした総和出力回路が既に提案されており [12]、今後のビジョンチップへの実装が検討されている。これを採用することで画素数の増加にも対応できると考えられる。

#### 4. 2. 2 列並列入出力ユニット

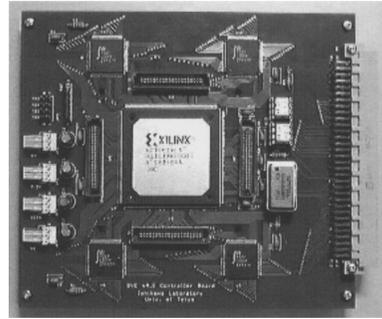
ローカルメモリの初期化や画像のモニタリングを行うため、ビジョンチップの外周 PE の入出力を通して列並列に入出力を行う、列並列入出力ユニットを用意した。これはバッファリングのためのメモリと、データバスとビジョンチップ側の入出力との間での幅の変換を行うロジックを含んだユニットである。

総和出力ユニット、列並列入出力ユニットとも SIMD パイプラインにより制御され、SPARSIS の要請に基づき、整数パイプラインとの間でフォーディングバスを設け、データハザードを回避している。

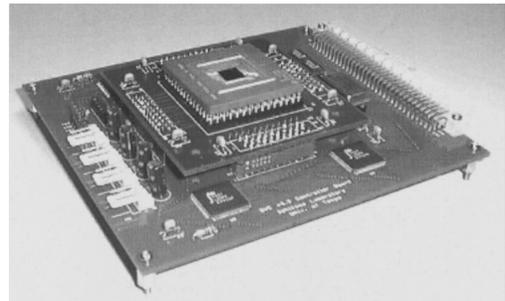
### 5. 試作システムと評価

#### 5. 1 システムの試作

以上で述べたアーキテクチャに基づいて、システムの試作を行った。コントローラの設計は Verilog-HDL で行い、Cadence 社の Verilog-XL によるシミュレーションで論理動作を確認した。FPGA デバイスとしては XILINX 社の XC4062XLA を用い、XILINX 社の



(a) Main board (without a vision chip).



(b) The system with a vision chip mounted.

図 8 試作されたビジョンチップシステム

Fig. 8 Photograph of the developed system board.

開発環境 Foundation にて論理合成、テクノロジマッピング、配置配線を行った。

回路規模はゲート数換算で約 27000 ゲートで、 $64 \times 64$  画素の  $S^3PE$  ビジョンチップと比較するとおよそ 1 割弱である。これは将来的にビジョンチップとともにワンチップに集積することで、更なる小型化・高速化を図ることも十分現実的な規模であると考えられる。

実装されたビジョンチップコントローラをビジョンチップとともにコンパクトなボード上に搭載したビジョンチップシステムの試作を行った。ビジョンチップとしては  $24 \times 24$  画素を集積したテストチップ [9] を用いている。試作されたシステムの写真を図 8 に示す。ビジョンチップはドータボードに実装し取り外し可能とすることで、将来のビジョンチップの変更にも対応可能とした。

#### 5. 2 試作システムの動作実験

##### 5. 2. 1 基本性能

試作されたビジョンチップシステムの動作を確認するため、動作実験を行った。その結果、ビジョンチップも含めたシステム全体でクロック周波数 40 MHz で

表 1 各処理の実行時間

Table 1 Execution time of sample programs.

program	steps	(time)
dilation, erosion (binary)	5	(0.5 $\mu$ s)
4-neighbor $x, y$ edge (binary)	9	(0.9 $\mu$ s)
4-neighbor smoothing (4 bit)	59	(5.9 $\mu$ s)
4-neighbor smoothing (8 bit)	99	(9.9 $\mu$ s)
0th moment* <sup>1</sup> (binary input)	55	(5.5 $\mu$ s)
1st moment* <sup>1</sup> (binary input)	275	(27.5 $\mu$ s)
center of gravity* <sup>1</sup> (binary input)	605	(60.5 $\mu$ s)

\*<sup>1</sup>For 24  $\times$  24 pixels.

表 2 試作システムの諸元

Table 2 Specification of the developed system.

System clock frequency	40 MHz
Temporal resolution of controlling vision chip	100 ns
Equivalent gate count for the vision chip controller	$\cong$ 27000
Program memory (external)	32b $\times$ 32kword
Data memory (external)	32b $\times$ 32kword
Main board size	140 $\times$ 123 mm
Power supply	3.3 V, 5 V

動作することを確認した。ビジョンチップのインストラクションレートはその 1/4 の 10 MHz であるため、100 ns の時間分解能で完全に実時間性を保ったビジョンチップの制御が実現できる。これは、これまで我々が応用システム開発に用いてきたスケールアップモデル [13] のインストラクションレートと比較して 100 倍以上高速な値である。例えば A-D 変換を行う際には、入力光の蓄積時間を 1 ms としてその 1/10000 の分解能で各サンプリングのタイミングを任意にスケジュールすることができる。

主な処理のステップ数及び実行時間を表 1 に示す。基本的なフィルタリング処理は数  $\mu$ s のオーダで実行でき、従来システム [13] と比較して、100 倍程度の高速化が実現されている。また、処理全体に占める割合の大きい特徴抽出処理については、従来は 1 フレーム時間の半分である 500  $\mu$ s を費して画像を出力した後 DSP で逐次的に重心を計算していたのに対し、本システムではモーメント計算のような大域的な特徴量抽出もシステム内で数～数十  $\mu$ s で行うことが可能である。アプリケーションシステムへは数ワードの特徴量を転送するだけで済み、出力時のボトルネックを大幅に改善できているといえる。

試作システムの諸元を表 2 にまとめる。

### 5.2.2 処理の記述例

ビジョンチップシステムを用いた視覚処理の記述例

```
repeat forever
integer AD_convert_result = 0;
integer feature_value;

// A/D Conversion
Reset_PD;
repeat (grayscale_level - 1) times
    Wait_to_adjust_timing;
    AD_convert_result += Read_PD_output;
endrepeat

// Image Processing
Do_some_processing(AD_convert_result);
feature_value = Feature_extract(AD_convert_result);

Synchronize_and_write_result(feature_value);
endrepeat
```

図 9 処理の記述例

Fig.9 Sample description of an algorithm for the vision chip system.

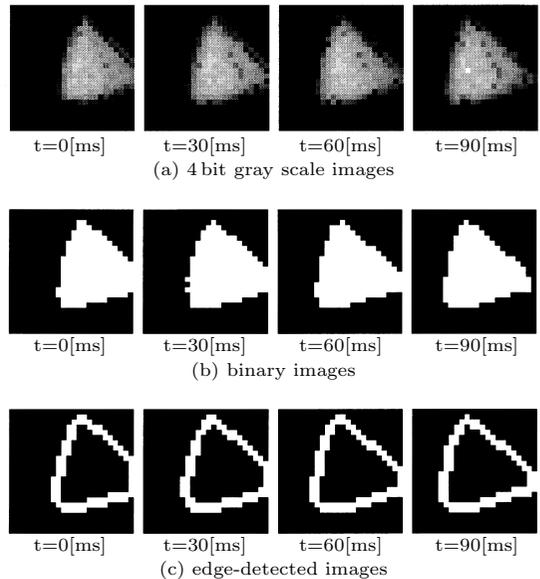


図 10 初期視覚処理実験結果

Fig.10 Image sequences with early visual processing.

を図 9 に示す。内側の repeat ループで A-D 変換を行い、その結果を処理して特徴抽出し、最後に共有メモリを介して外部システムに結果を伝えている例である。例えばこの中で、A-D 変換時のウェイト時間を可変とすることで、目的や環境の変化に応じた PD の動的な制御も実現可能である。

### 5.2.3 視覚処理実験

ビジョンチップシステムに光学系を取り付け、撮像

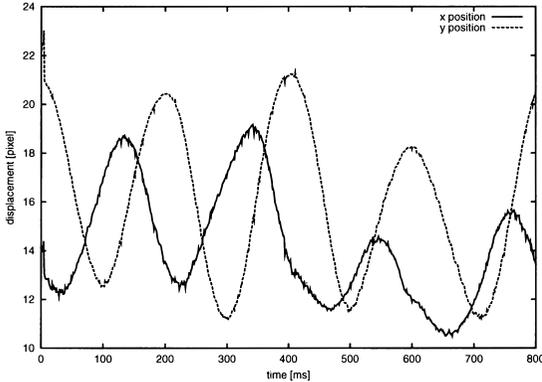


図 11 重心検出実験結果  
Fig. 11 Center of gravity of a moving target.

と視覚処理の実験を行った。現在用いているビジョンチップは感度が低いことがわかっているため、撮影対象は白熱電球の前方を遮光し、その中央に三角形の穴を空けて拡散用の白い紙を貼ったものを用いた。

1 kHz のフレームレートでの撮像と 2 値化、エッジ検出を行った結果を図 10 に示す。処理結果の表示は 30 ms ごとに行っている。また、特徴抽出の例として高速で移動する対象の 1 ms ごとの重心検出結果を図 11 に示す。このように得られる情報を用いて、例えばアクティブビジョンを制御し高速な対象追跡を実現するといったことが可能である [14]。

## 6. むすび

本論文では、汎用ビジョンチップを用いた実時間視覚処理システムのアーキテクチャと試作システムについて述べた。本システムを用いることで、ビジョンチップを様々な分野に応用していくことが可能となり、また接続する外部システムによらずに安定した動作環境をビジョンチップに提供できる。

ビジョンチップの応用面からは、スケールアップモデルを用いて既に様々な研究が進められており [13], [15], [16], その有効性が示されている。今後そのような分野へ実際にビジョンチップを適用していくことが可能となるだけでなく、これまで特にインストラクションレートやその実時間性といった面で実現不可能であった新たな応用がひらけると期待できる。今後、実際の応用からのフィードバックを取り入れながら、より実用的なシステムの実現を目指していく予定である。

## 文 献

- [1] A. Moini, Vision Chips, Kluwer Academic Publishers, Norwell, 2000.
- [2] 小室 孝, 鈴木伸介, 石井 抱, 石川正俊, “汎用プロセッシングエレメントを用いた超並列・超高速ビジョンチップの設計” 信学論 (D-I), vol. J81-D-I, no. 2, pp. 70-76, Feb. 1998.
- [3] Y. Fujita, S. Kyo, N. Yamashita, and S. Okazaki, “A 10 GIPS SIMD processor for PC-based real-time vision applications,” 4th IEEE Int. Workshop on Computer Architecture for Machine Perception (CAMP'97), pp. 22-32, 1997.
- [4] 井上博允, “リアルタイムビジョンとその応用” 日本ロボット学会誌, vol. 13, no. 3, pp. 306-310, 1995.
- [5] R. Cypher and J.L.C. Sanz, The SIMD Model of Parallel Computation, Springer-Verlag, New York, 1994.
- [6] R. Forchheimer and A. Åström, “Near-sensor image processing: A new paradigm,” IEEE Trans. Image Processing, vol. 3, no. 6, pp. 736-746, 1994.
- [7] T.M. Bernard, Y. Zavidovique, and F.J. Devos, “A programmable artificial retina,” IEEE J. Solid-State Circuits, vol. 28, no. 7, pp. 789-798, 1993.
- [8] M. Ishikawa, K. Ogawa, T. Komuro, and I. Ishii, “A CMOS vision chip with SIMD processing element array for 1 ms image processing,” 1999 ISSCC, pp. 206-207, 1999.
- [9] 小川一哉, 小室 孝, 石井 抱, 石川正俊, “S<sup>3</sup>PE アーキテクチャに基づくデジタルビジョンチップとその高集積化” 信学技報, ICD99-4, 1999.
- [10] D.A. Patterson and J.L. Hennessy, コンピュータ・アーキテクチャ, 日経 BP 社, 東京, 1992.
- [11] 川口貴裕, 藤原 崇, 坂本勝人, 天野英晴, “マルチプロセッサ ASCA 用カスタムプロセッサ MAPLE,” 信学技報, CPSY98-49, 1998.
- [12] 小室 孝, 中坊嘉宏, 石井 抱, 石川正俊, “デジタルビジョンチップのためのモーメント抽出アーキテクチャ” 信学技報, PRMU99-51, 1999.
- [13] 中坊嘉宏, 石井 抱, 石川正俊, “超並列・超高速ビジョンを用いた 1 ms ターゲットトラッキングシステム” 日本ロボット学会誌, vol. 15, no. 3, pp. 417-421, 1997.
- [14] 鏡 慎吾, 小室 孝, 石井 抱, 石川正俊, “超並列ビジョンチップシステムを用いた高速ロボットビジョン” 日本機械学会ロボティクス・メカトロニクス講演会'00, pp. 1A1-50-73, 2000.
- [15] 大脇崇史, 中坊嘉宏, 並木明夫, 石井 抱, 石川正俊, “視覚モダリティ変換を用いたリアルタイム実環境仮想接触システム” 信学論 (D-II), vol. J81-D-II, no. 5, pp. 918-924, May 1998.
- [16] 並木明夫, 中坊嘉宏, 石井 抱, 石川正俊, “高速センサフィードバックを用いた感覚運動統合把握システム” 第 4 回ロボティクスシンポジウム, pp. 1-6, 1999.

(平成 12 年 8 月 25 日受付, 12 月 22 日再受付)



鏡 慎吾

平 10 東大・工・計数卒．平 12 同大大学院修士課程了．現在，同大学院工学系研究科計数工学専攻博士課程在学中．ビジョンチップシステム，実時間情報処理の研究に従事．



小室 孝

平 8 東大・工・計数卒．平 13 同大大学院博士課程了．現在，科学技術振興事業団研究員．ビジョンチップのアーキテクチャ，デバイスの研究に従事．工博．平 12 LSI IP デザイン・アワード IP 優秀賞受賞．



石井 抱

平 4 東大・工・計数卒．平 6 同大大学院修士課程了，同大工学系研究科計数工学専攻助手を経て，平 12 東京農工大学工学部講師，現在に至る．工博．ビジョンチップ，センサ並列情報処理，ロボット制御システムの研究に従事．平 10 日本ロボット学会論文賞，平 11 同学術奨励賞，平 12 LSI IP デザイン・アワード IP 優秀賞各受賞．



石川 正俊（正員）

昭 52 東大・工・計数卒．昭 54 同大大学院修士了．同年，通産省工業技術院製品科学研究所に入所．平 1 東大・工・計数助教授，現在，同大大学院情報理工学系研究科システム情報学専攻教授．生体の情報処理機構の回路モデル，超並列・超高速ビジョン，光コンピューティング，センサフュージョン等の研究に従事．工博．昭 59 計測自動制御学会論文賞，昭 63 工業技術院長賞，平 1 応用物理学会光学論文賞，平 10 高度自動化技術振興賞（本賞），平 10 日本ロボット学会論文賞，平 11 日本機械学会ロボティクス・メカトロニクス部門学術業績賞，平 11 櫻井健二郎氏記念賞，平 12 LSI IP デザイン・アワード IP 優秀賞各受賞．