

ビジョンチップを用いた実時間視覚処理システム VCS-IV

鏡 慎吾^{†a)} 小室 孝[†] 渡辺 義浩[†] 石川 正俊[†]

VCS-IV: A Real-Time Vision System Using a Digital Vision Chip

Shingo KAGAMI^{†a)}, Takashi KOMURO[†], Yoshihiro WATANABE[†],
and Masatoshi ISHIKAWA[†]

あらまし 画素ごとにデジタル処理回路を集積したビジョンチップをベースとした実時間視覚処理システムについて述べる。開発したシステムは、我々の最新のビジョンチップに対応して視覚処理の高速化・高機能化が実現されるとともに、小型かつ拡張性・汎用性の高い実装がなされている。また、画素内 A-D 変換の動作を高時間分解能でソフトウェア制御することによる、プログラマブルな撮像機能を大きな特徴としている。基本的な視覚処理及び計測システムへの適用による性能評価を示すとともに、強化された撮像機能の運用例を示す。

キーワード ロボットビジョン, マシンビジョン, CMOS イメージセンサ, リアルタイムシステム, 画素レベル A-D 変換

1. ま え が き

視覚認識や視覚フィードバック制御に対するニーズの高まりとともに、従来の視覚システムのフレームレートが、必ずしも十分ではないことが認知されつつある。我々はこの問題に対し、イメージセンサの各画素に光検出器 (PD) とともにプログラマブルなデジタル処理要素 (PE) を集積し、汎用かつ高速な視覚処理を実現するビジョンチップの研究を進めてきた。

具体的には、ビジョンチップのための並列処理アーキテクチャの開発と、その CMOS VLSI としての実装を行うとともに [1]、制御機構、周辺インタフェース等を統合した視覚システムの開発を行った [2]。またこれらと並行して、そのアーキテクチャの有効性を示すために、フォトダイオードアレーと並列処理装置を全画素結線したスケールアップシステム [3]、高速 CMOS イメージセンサから並列処理装置への列並列画像転送を用いたスケールアップシステム [4] を構築し、ロボティクスや VR、バイオテクノロジー等への応用を行ってきた [5]。

ビジョンチップ、及びそれをベースとした視覚システムには、視覚処理の高速化・高機能化とともに、様々なアプリケーションシステムにおいて直接利用可能な実装が求められる。また、高フレームレート時にも十分な精度で光検出を行えるよう、撮像機能を強化することも重要な課題である。本論文では、これらの点を考慮して新たに開発を行った実時間視覚処理システムの構成と動作について述べる。

本論文で新たに述べるのは (1) 新しいビジョンチップに対応した制御アーキテクチャの拡張 (2) 小型かつ拡張性の高いシステム実装 (3) ソフトウェアで PD を実時間制御することによる撮像機能の強化の実現、の 3 点である。以下では、2. にて関連研究について述べた後、3. ~ 5. にて上記 3 点についてそれぞれ詳述する。

2. 関連研究

近年、特に CMOS イメージセンサの高性能化により、ビデオレートを上回るフレームレートでの画像出力が可能なイメージセンサが製品として入手可能になり始めた [6] ~ [8]。よってこれらのセンサからの出力を高速転送し実時間処理することで、高速な実時間視覚システムを構成することができる。

大明ら [9] は、Micron Technology 社 [6] のセンサの出力を PC (Xeon 2.4 GHz × 2) で処理し、1000 fps

[†] 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,
University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-
8656 Japan

a) E-mail: Shingo_Kagami@ipc.i.u-tokyo.ac.jp

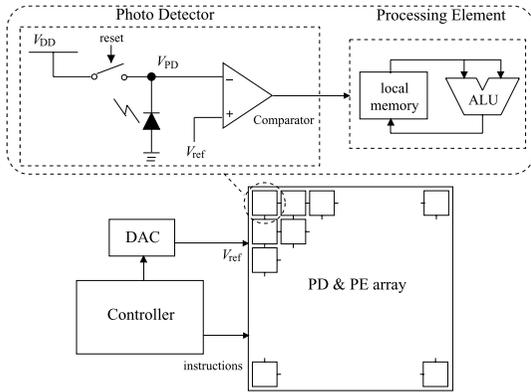


図 1 システムの全体構成
Fig. 1 Structure of the whole system.

での対象追跡を実現している。筆者らのグループでも、専用 CMOS センサの出力を複数の FPGA で構成された並列処理装置で処理する実時間視覚システムを開発している [4]。しかし、大量のデータを転送・処理する必要があることから、システム全体の小型化、低コスト化や低消費電力化は困難である。

CMOS イメージセンサの部分画像読出し機能を利用し、注目する対象の周囲の画像のみを処理することで、高フレームレートを実現することも可能である。石井 [10] は、FillFactory 社 [8] のセンサを採用したカメラを用いて、画素選択機能をもつ高速視覚システムを実現するとともに、独自のセンサの開発も検討している。Muehlmann ら [11] も同様に FillFactory 社のセンサを用い、部分読出した画像を USB を通じて PC で処理し、最大 2500 fps の実時間視覚処理を実現している。これらはアプリケーションによっては極めて有望なアプローチであるが、テクスチャが重要な場合や、新規に現れる対象を瞬時に検出したい場合など、適さない場合も多い。

3. 制御アーキテクチャ

3.1 基本構造

システムの全体構成を図 1 に示す。各画素の PE と PD が、外部に置かれたコントローラからの制御信号に従って SIMD 型で制御される。高フレームレートでの実時間視覚処理、並びに PD の実時間制御をソフトウェアで実現するため、コントローラには高い時間分解能での実時間性を保証することが求められる。

これを実現するため、文献 [2] にて、SIMD 型 PE アレーであるビジョンチップを制御するためのパイプ

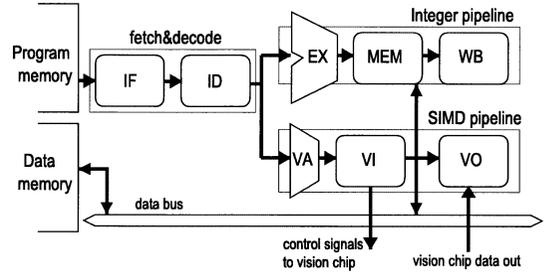


図 2 SIMD アレーのための実時間制御アーキテクチャ
Fig. 2 Real-time control architecture for SIMD arrays.

ライン (SIMD パイプラインと呼ぶ) を、RISC 型マイクロプロセッサに統合したアーキテクチャ SPARSIS を提案した。その構成を図 2 に示す。プログラムは整数命令と SIMD 命令の 2 種類からなり、整数命令によって通常の整数スカラー演算のほか、分岐や関数呼出などのプログラム制御を行うことができる。新たな PE アーキテクチャに対応するため、まず制御アーキテクチャ SPARSIS を一般化して整理した。

SPARSIS の主な特徴は、(1) 整数パイプラインと SIMD パイプラインをどのような組合せで使用しても動的なパイプラインストールが発生せず、命令サイクルのレベルで実時間性を保証できること、(2) 単一の SIMD 命令が複数サイクルの PE アレー制御信号列に展開されることで、全体の動作周波数を抑えながら高いレートでの PE アレー駆動が可能であること、の 2 点である。

この特徴を保持しながら、異なる PE アレーアーキテクチャへの対応を可能とするため、各パイプラインステージの動作タイミングや整数命令との相互作用については仕様として規定した上で、SIMD 命令フォーマットや各ステージの処理内容、VI ステージにおける PE アレー駆動レートの倍率といった詳細を、この仕様の範囲内で設計者の手にゆだねることとした。これにより、制御対象とする PE アレーの構造に最適化された SIMD パイプラインを設計するための自由度が得られるとともに、動作タイミングについては仕様が明確に定められているため、パイプラインストールやディレイスロットの必要性について個別に考慮する必要はなくなる。

3.2 具体設計

以上のアーキテクチャ仕様に基づいた、具体的な SIMD パイプラインの設計について述べる。

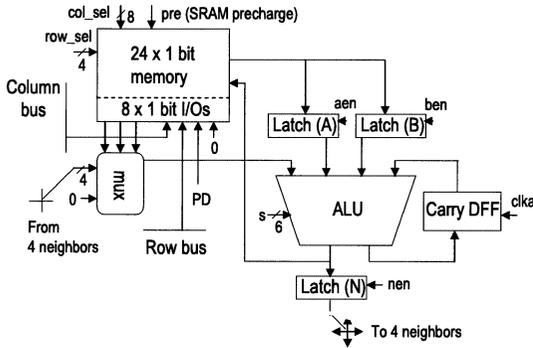


図 3 PE の構成
Fig. 3 Block diagram of the processing element.

制御対象となるビジョンチップの PE の構成を図 3 に示す [12] . I/O 領域を含めて 32 ビット (5 ビット幅) の画素内メモリ空間と, ビットシリアル ALU を基本構成とする点は, 前身のアーキテクチャ [1] と同様である . 新たな特徴として, 隣接する複数 PE の結合による可変解像度処理機能や, 離れた PE 間の効率的な通信機能等が挙げられる . これらにより, 重心計算のような大域的な特徴抽出が高速化されたほか, 並列ブロックマッチングや座標変換など, 単純な画素並列アーキテクチャでは不得手であった処理を効率的に実現できるようになっている .

これを制御する SIMD パイプラインの設計を行った . 設計に際し, ソフトウェア環境の並行開発を可能とするため, また, 画素内メモリを構成する SRAM の制御タイミングマージンを確保するため, 命令セットの部分的変更や, ビジョンチップ制御コードの遷移タイミングの微調整を可能とすることを目指した .

設計されたパイプラインの構成を図 4 に示す . 図の上部に示した命令フォーマットのうち opg フィールドにより, read (画素内メモリから値を読み, A and/or B ラッチに記憶), opw (aluop フィールドで指定された演算を行い, 結果を画素内メモリに格納), opn (同様に, 結果を N ラッチに格納) といった命令グループを指定する . ID ステージに置かれたテーブルにより, opg ごとに汎用引数 arg フィールドと実際の制御信号との対応付けが定められ, 展開結果が VA ステージに引き渡される . VA ステージでは a フィールドの値と汎用レジスタから読み出された値から, 画素内メモリアクセスに用いられる有効アドレスが計算され, SRAM アクセスを行うための行選択信号, 列選択信号に展開される .

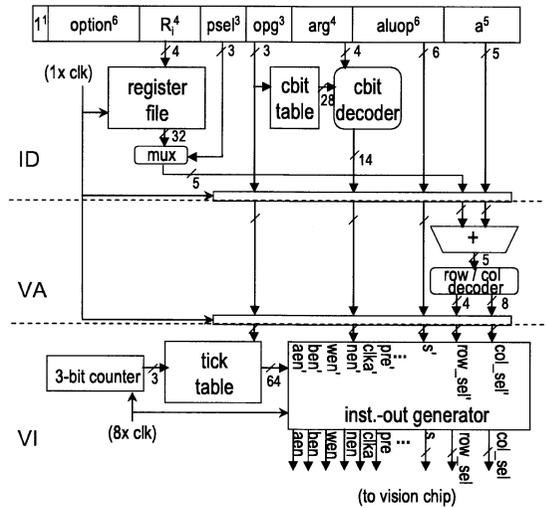


図 4 SIMD パイプラインの詳細構成
Fig. 4 Architecture of the SIMD pipeline.

VA から引き渡された信号は, VI ステージの出力信号生成器を通して, 命令レート (設計値 10 MHz) の 8 倍のレート (設計値 80 MHz) でビジョンチップに送出される . 出力信号生成器は, VI ステージに置かれたテーブルに基づいて, 出力信号の遷移タイミングを制御しており, 試行錯誤的にタイミング調整を行うことが可能となった . 図 5 に, 前述した主な命令グループ (opg) の信号遷移タイミングを例示する . ビジョンチップからの出力は, VO ステージでコントローラに取り込まれ, 整数命令から利用可能になる .

4. システム実装

4.1 構成

実時間視覚処理システムのハードウェアとしての実装に際して, 二つの目標を設定した . 一つは研究用テストベッドとしての拡張性・汎用性であり, 今一つは現実的なアプリケーションへの適用可能性である . この二つの目標は, 特にフォームファクタの決定においては相反しがちなものである .

図 6 に開発したシステム VCS-IV (Vision Chip System, Version 4) の写真を示す . システムは, 図 7 に示すようなスタック接続可能な複数枚の基板に分けて実装されている . 最小構成は, ビジョンチップを搭載する基板 (図 7 左上) と, コントローラを搭載する基板 (同右上) の 2 枚をスタックした形となる . 必要に応じて, 5V TTL との接続のための入出力レベル変

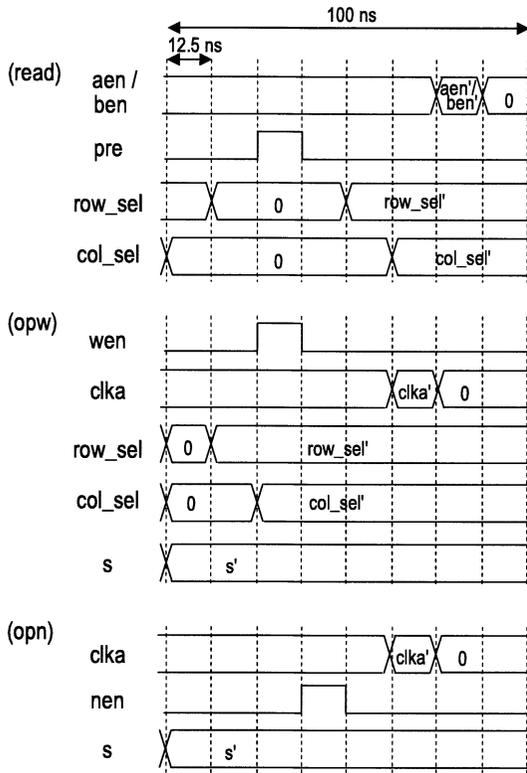


図 5 主な SIMD 命令と対応する制御コードの時間遷移
Fig. 5 Part of the SIMD instruction set.

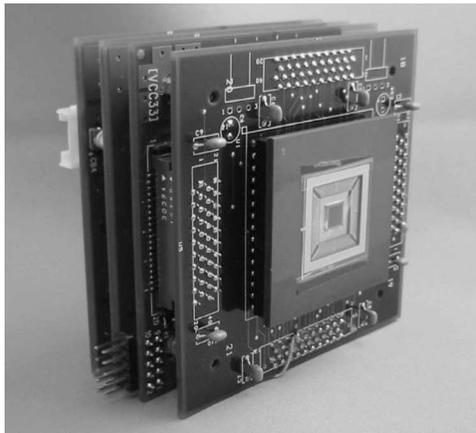


図 6 開発されたシステム VCS-IV
Fig. 6 Photograph of the implemented system, VCS-IV.

換基板（同左下）や電源基板（同右下）を追加して使用する。

すべての基板サイズを 76×76 mm に合わせ、入力レベル変換や電源といったオプションな要素を複

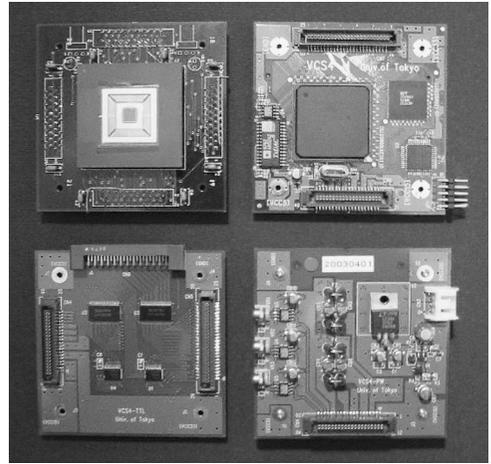


図 7 システムの構成基板
Fig. 7 Photograph of the system boards.

数の基板に分配するとともに、大容量 FPGA を導入して部品数の減少を図った。ユーザは自分の必要な機能のみを選択して使用できるほか、システムとしての基板スタック枚数に制限はないため、必要に応じて新たな追加基板を開発・導入することもできる。結果として、多くのアプリケーションで実用的に用いることのできるサイズを実現した。

図 6 の最前面のセンサ基板には、前章で述べた PE アーキテクチャを $0.35 \mu\text{m}$ CMOS プロセスにて実装したビジョンチップ [12] が搭載されている。 $5.4 \times 5.4 \text{ mm}^2$ のチップ内に 64×64 画素が集積されている。

センサ基板の直後に配置されたコントローラ基板には、前章で述べたコントローラアーキテクチャが実装された FPGA が搭載されている。外部システムとのインタフェースを柔軟にするため、外部との接続は必ず FPGA を経由するように構成されている。これによって様々な外部システムとの接続を、FPGA 内に実装されたインタフェースモジュールの入換えのみで実現することができる。例えば汎用の組込み MPU や DSP との接続、ネットワークインタフェースとの接続などを想定している。

また、市販の高速 D-A 変換器をコントローラ基板上に実装し、コントローラのメモリ空間上にマップした。これは、次章で述べるソフトウェアによる A-D 変換特性の制御に不可欠な機能である。

4.2 動作結果

Linux が動作する PC をホストとし、開発したシステムを市販の平行 I/O ボードを通して接続できる

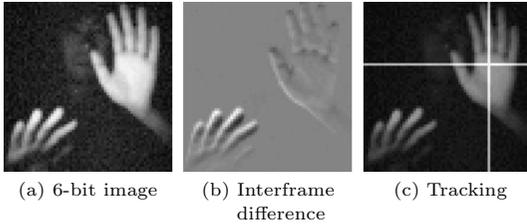


図 8 基本的な視覚処理の実験結果

Fig. 8 Experimental results of basic visual processing.

表 1 各種アルゴリズムの実行時間

Table 1 Execution times of visual processing programs.

program	time [μ s]
local filtering operations:	
dilation, erosion (binary)	2
edge detection (binary)	3
edge detection (6-bit)	20
smoothing (binary)	5
smoothing (6-bit)	16
global operations (binary input):	
global logical-OR	2
0th order moment	12
1st order moment	23
centroid detection	58
object search for connected component labeling (worst case) [13]	54

環境を用意した。システムのプログラムメモリ、データメモリを直接読み書きできる低レベル入出力ライブラリと、実行制御や画像データを取扱う制御ライブラリ、及びコマンドライン上、X11 上で動作するテストツールが用意されている。プログラミングは、アセンブラ及びその生成支援環境を用意し、その上で行った。

機能テストの結果、コントローラの全機能が命令レート 10 MHz で動作し、ビジョンチップ制御コードを出力レート 80 MHz で送出できることを確認した。命令サイクルのレベルの実時間性が保証できることから、実時間性の時間分解能は 100 ns であり、多くのアプリケーションでビジョンチップに求められるフレーム時間である 1 ms の、1/10000 の分解能での制御が可能である。システムの全要素への電源供給源となる 5 V DC 入力の消費電流は、最大で 0.3 A である。

処理結果の出力例を図 8 に示す。両手を交互に上下に動かしているところを撮像し、処理した。6 ビットグレースケールで撮像された原画像を図 8(a) に示す。次章で述べるソフトウェア A-D 変換を用いて比例スケールで撮像されたものである。撮像時間は 1 ms とした。これに対してフレーム間差分を計算したものを同図 (b) に示す。上下の動きが検出されていることが

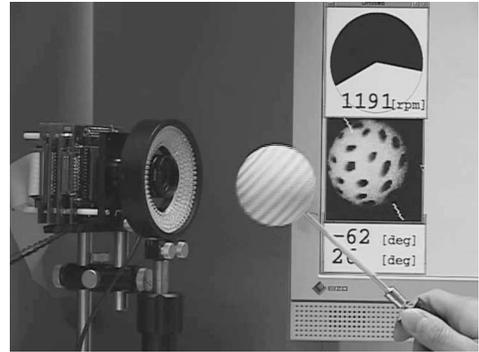


図 9 回転計測実験の様子

Fig. 9 Experiment of real-time rotation measurement.

分かる。また、2 値化とノイズ除去を施した原画像から注目する対象のみを追跡し、その重心を算出・描画した例を同じく (c) に示す。いくつかの代表的な視覚処理に要する実行時間を表 1 にまとめた。

4.3 実時間視覚計測システムへの応用

より実際のアプリケーションとして、複数対象の同時追跡をベースとした視覚計測システムを構築した。ここで用いた複数対象追跡処理は、新規に現れる対象のラベリングと、前フレームまでに追跡中の対象のフレーム間対応付けの 2 フェーズから構成される [13]。前者では画像中の連結領域を 2 分探索を用いて一つずつ抽出する。後者では、前フレームでの対象の像と重なりをもつ領域を抽出する。

この処理に基づいた回転計測実験の様子を図 9 に示す。白球の表面上に多数の黒い点を描き、この球を高速に回転させた。追跡処理によって複数の黒点の軌跡を得て、そのデータから最小二乗法を用いて球の回転軸、回転速度を推定した。図に示すように、推定結果はホスト PC のモニタ上にオンラインで表示される。

文献 [13] で報告した実験では、コントローラとして、メモリに展開したビット列を命令レート (10 MHz) でそのままビジョンチップに送出するプロトタイプ版を使用しており、パフォーマンスが十分でないことが指摘されていた。本論文で述べたシステムを新たに開発した結果、従来は回転速度 550 rpm が限界であったところ、最大 1200 rpm まで計測可能となった。これは主に、追跡処理の高速化による処理時間の短縮と、5.4 で述べる撮像処理と画像処理の並行実行の導入により、フレームレートが 400 fps から 1000 fps に改善されたことによる。

また、視野内を通過する対象の数をカウントする実

験を行った．これは、既にカウントした対象を誤検出しないように追跡し続けながら、新たに出現した対象をカウントするという処理から構成される．フレームレート 500 fps で同時出現個数 15 個まで正常に処理することができ、このとき、移動方向のサイズが約 10 pixel の対象の場合で移動速度 5500 pixel/s まで正常にカウントできた．

5. ソフトウェア A-D 変換の実現

画素内の PE でデジタル処理を行うビジョンチップでは、入射光強度からデジタル値への A-D 変換が画素内で行われなくてはならない．短いフレーム時間内に行える処理量や画素内のメモリ量には限りがあるため、単に高精度の A-D 変換が行われるだけではなく、必要十分な情報を撮像の過程で柔軟かつ効率的に抽出できることが望まれる．

PD と PE が画素内で直結しており、かつそれらが高い時間分解能で制御できることにより、PD の動作をソフトウェアで柔軟に制御することが可能となる．この点に着目し、我々はソフトウェアにより画素内 A-D 変換を制御するための手法を提案した [14]．

PD 回路の構成を図 1 の左上の枠中に示した．PD はフォトダイオード、リセットスイッチ、及び比較器からなる．撮像は以下の手順で行われる．まずフォトダイオードの電位 V_{PD} が V_{DD} にリセットされ、その後スイッチが開放される．フォトダイオードの寄生容量に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 V_{PD} は低下していく．比較器の出力は、その一方の入力である V_{PD} が他方の入力である参照電位 V_{ref} を下回る際に論理値 0 から 1 に変化する．入射光が明るい場合は放電が速いために、比較器の出力は早くに反転し、暗い場合は遅くなる．よってこの出力反転までの時間を計測することで明るさの情報が得られる．リセット時刻、比較器出力の読出し時刻、参照電位 V_{ref} を適切に制御することによって、様々な A-D 変換特性を得ることができる．

n 階調 A-D 変換を実現するソフトウェアの典型的な動作は以下のとおりである．PD リセット命令の実行後、定められた各タイミングでの参照電位 V_{ref} の供給と比較器出力の読出しを、 $n-1$ 回繰り返す．あらかじめ 0 で初期化した変数に対し、繰返しごとに読み出された 1 ビット値を加算していくことで、 n ビットのデジタル値が得られる． V_{ref} の供給は、D-A 変換器がマップされたアドレスへの整数ストア命令で、

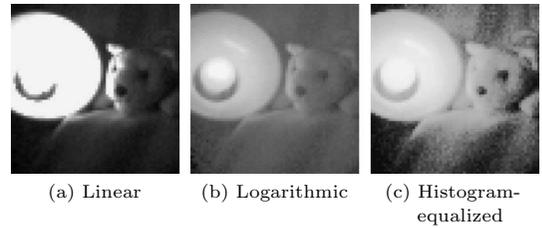


図 10 異なる A-D 変換特性での撮像結果

Fig.10 Images obtained with different conversion scales.

比較器出力の読出しは、PE の入力ポートを読み出す SIMD 命令で、それぞれ行われる．タイミングの制御は、命令スケジューリング及びタイマカウンタの利用により行うことができる．

本章では、開発したシステムにおけるこの「ソフトウェア A-D 変換」の運用例を挙げ、状況に応じた様々な撮像機能が実現できることを示す．

5.1 あらかじめ与えられた変換スケール

A-D 変換スケール、すなわち、フォトダイオードを流れる光電流量を量子化する際の量子化境界値列が与えられた際、そのスケールでの A-D 変換を実現できる PD 読出しタイミング列、及びその各時刻での V_{ref} の値の組合せは一意には定まらない．文献 [14] では、このすべての組合せの中で、ノイズの影響をできるだけ小さくする意味で最適なものを生成するアルゴリズムを提案した．

図 10 に、開発したシステムにこのアルゴリズムを適用して撮像した画像を示す．図 10 (a) には、比例スケールを与えた場合の撮像結果が、同 (b) には、対数応答スケールを与えた場合の撮像結果が示されている．いずれも撮像時間は 8 ms で、A-D 変換の階調数は 64 とした．文献 [14] ではオフラインで合成した画像を示したが、同様の画像が実際に得られていることが分かる．視野内には点灯した白熱電球とぬいぐるみが被写体として収まっており、適切な変換スケールを与えることで、十分な撮像ダイナミックレンジが得られることが分かる．図 11 にそれぞれの撮像を行った際の制御スケジュール、すなわち PD 読出しタイミング列と各時刻での V_{ref} の値の組合せを示す．

5.2 変換スケールの動的制御

照明条件をあらかじめ知ることができない場合や、その変動が激しい場合、環境に適應しながら自動的に撮像条件を変化させる能力が必要となる．そのような適應機能の一例として、ヒストグラム均等化を自動的

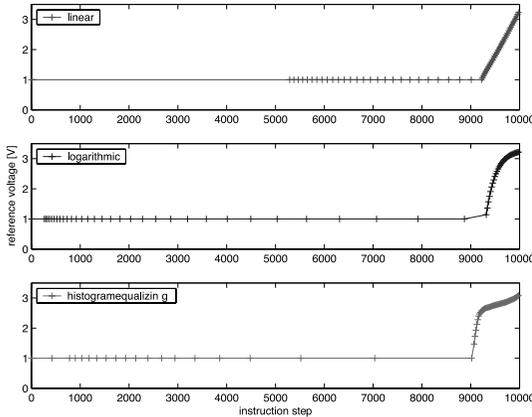


図 11 各変換スケールを実現する制御スケジュール．時間軸 1 ステップは、撮像時間の 1/10000 として正規化した．電圧を示すグラフ上の多数の点が、読み出しタイミングを表す

Fig. 11 A-D conversion schedules for several characteristics.

に行う撮像アルゴリズムを述べる．

このアルゴリズムでは、フレームごとの適応によって、量子化境界値列 $\{i_k\} (i = 0, 1, \dots, n)$ を決定する．ここで境界値列は、光電流 i が $i_k \leq i < i_{k-1}$ を満たす際にデジタル値 $n - k$ に変換されるとして表した． i_0 は ∞ 、 i_n は 0 として扱う．

次のフレームで用いる量子化境界値列 $\{i^{\text{next}}_k\}$ は、現フレームの量子化境界値列 $\{i_k\}$ と得られた画像の累積ヒストグラム $\{H_k\}$ から、以下のように計算する．

$$i^{\text{next}}_k = i_{n-l+1} + \frac{H_n(n-k)/n - H_{l-1}}{H_l - H_{l-1}}(i_{n-l} - i_{n-l+1})$$

ここで、 l は $H_l \geq H_n(n-k)/n$ を満たす最小の自然数とし、 H_k は、画素値が j である画素数を h_j として、 $H_k = \sum_{j=0}^{k-1} h_j$ と定義する．これは、目的とする累積ヒストグラムである $H_k = H_n(n-k)/n$ を達成できる境界値列を、線形補間に基づいて二分探索していくアルゴリズムとなっている．

いったん量子化境界値列 $\{i_k\}$ が与えられれば、PD の制御スケジュールは先の例と同様にして得ることができる．図 10 (c) に、このアルゴリズムを適用して撮像した結果を示す．ただし簡単のため、対象とするシーンは静的なものとし、 $\{i_k\}$ の計算及びスケジュールの生成はホスト PC で行った．図 10 (a) の撮像に用いた比例スケールを初期状態として適応させたところ、数フレームで収束した．図 11 最下段は、収束した時



(a) 1 ms (b) 2 ms (c) 4 ms
図 12 変換スケール固定・異なる撮像時間での撮像結果
Fig. 12 Experimental results of fixed-scale, different-frame-rate imaging.

点での制御スケジュールを示している．

適応プロセスを走らせ続けたところ、得られたスケールがわずかに変動する様子も観察された．動的なシーンに適用した場合の安定性については、更に検討が必要と考えられる．また、今回の実験では発生しなかったが、計算される $\{i_k\}$ によっては、一部の境界値間の間隔が狭すぎるなどの理由により、実行可能なスケジュールが生成できない場合が発生し得る．これが問題となる場合は、最適なスケジュールに近い実行可能なスケジュールを生成するように改変されたアルゴリズムを用いることも可能である [15]．

5.3 フレームレートの制御

以上で述べたものはいずれも、撮像時間が固定のもので、すなわち結果的にフレームレートを一定とする条件のもとで、様々な変換スケールを得るようにスケジューリングを行ったものであった．

一方、同じアルゴリズムを用いて、A-D 変換スケールを固定した上で、様々な撮像時間を実現するようにスケジューリングを行うことも可能である．通常のイメージセンサのように一定時間の露光で蓄積された信号量を測定する方式では、露光時間を長くすることによる画素値の飽和が問題となる．対して我々のアルゴリズムでは、蓄積された電荷が飽和量に達するまでの時間情報が A-D 変換に利用されているため、この問題が生じない．シーンの最大照度が指定された変換スケールの中に収まっている限り、撮像時間をどのように選んでも、常に飽和せずに撮像できる．

A-D 変換スケールを比例スケールに固定し、撮像時間を 1, 2, 4 ms に変えてそれぞれ撮像を行った結果を図 12 に示す．いずれも同じ変換スケールを実現できているが、それぞれノイズレベルが異なる．明らかに、撮像時間が長いものの方がノイズが小さいことが分かる．これらの結果は、フレームレートを高くすることと、画質を高くすることとの間にあるトレード

オフを示すものであり、ソフトウェアによる撮像制御の導入は、ユーザがその目的や状況に応じて適切な撮像特性を設定することを可能とするものであるといえる。フレームレートと画質のトレードオフに関する理論的側面については、文献 [16] にて議論している。

5.4 撮像処理と画像処理の並行実行

開発したシステムにおいては、上記のような様々な制御に基づいて撮像を行った後、画素内メモリに保存されたデジタル画素値に対してフィルタ処理、特徴抽出などの画像処理が施され、得られた最終的な処理結果が出力される。ここで、撮像処理と画像処理の配置の仕方には、大きく二つの方式がある。

第一の方式は、1 フレーム内において撮像処理と画像処理を単純に連続して行うものであり、簡単に実現できる。本章で述べた実験結果はすべてこの方式を用いた。図 13(a) にこれを模式的に表す。この場合、フレーム時間は撮像時間と画像処理時間の和となるため、フレームレートが高いとき、あるいは画像処理の計算量が多いときは撮像時間を十分にとれない場合がある。図 12 に示したようにこの影響は画質の低下となって現れ、照明条件の工夫などが必要となる。

第二の方式は、1 フレーム内において撮像処理と画像処理をオーバーラップさせて実行し、あるフレームにおける撮像結果に対する画像処理を次フレームで施すものである。以上で述べてきたソフトウェア A-D 変換処理では、撮像時間内における有効命令の数は A-D 変換の階調数に対して一意に定まる。撮像時間からこ

の有効命令時間を減じた残りはアイドル時間であり、これを画像処理に割り当てることができる。図 13(b) に 8 階調 A-D 変換を行う場合の例を示した。フレーム時間のうち P_0 から P_7 で表される有効命令ブロックのほかは撮像処理には使用されないで、この時間を用いて、前フレームで撮像された画像の処理を実行する。フレーム時間いっぱいまで利用して撮像できるため長い露光時間をかせぐことができ、画質を確保するためには有利となる。ただし、撮像処理と画像処理を同時実行するために、一般にはそれぞれに必要な画素内メモリを独立して用意しなければならない。

この方式は、A-D 変換スケールが固定で与えられ、画像処理の内容も固定である場合は、静的に命令スケジューリングすることで容易に実装できる。一方、変換スケールや処理内容が動的に変化する場合は一般には困難になるが、最も明るい量子化境界値 i_1 の上限（すなわち、シーンのうち興味のある照度の上限）が既知であるという条件のもとでは、現実的な実装が可能である。この場合、比較器出力の反転がそこ以前では生じ得ないような時刻を求めることができ、図 13(c) のように、リセット処理 (P_0) からその時刻までを画像処理専用で割り当てることができる。撮像処理の P_1 以降は、その時刻以降にしかスケジュールされ得ないからである。またこの場合、 P_0 では PD のリセットのみを行い、画素値変数の初期化を P_1 まで遅らせることによって、撮像処理用と画像処理用のメモリ領域を共有できるため、画素内メモリも無駄なく使用できる。

この条件に当てはまる特殊ケースとして 2 値画像のみが必要な場合は、同図 (d) のようにフレームの開始時と終了時のみに撮像のための命令ブロックを実行し、残りはすべて画像処理に利用できる。前章で述べた視覚計測システムでは、この方式を用いることで十分なフレームレートを確保した。

6. む す び

ビジョンチップをベースとした実時間視覚処理システム VCS-IV を開発した。最新のビジョンチップに対応するための制御アーキテクチャの拡張とともに、研究用テストベッドとしての汎用性・拡張性と、実用性を両立する実装を実現した。また、高時間分解能でセンサの動作タイミングをソフトウェア制御できることにより、撮像機能を柔軟に制御することが可能となった。基本的な視覚処理結果と視覚計測への適用結果、

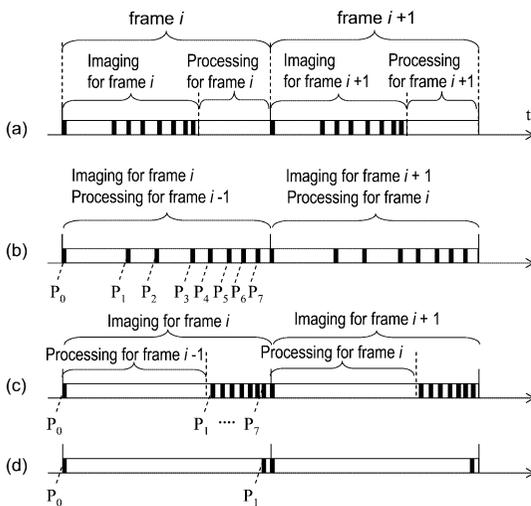


図 13 撮像処理と画像処理の並行実行の様子

Fig. 13 Overlapping of imaging and image processing.

及び強化された撮像機能の具体的な運用例を通して、本システムの有用性を示した。

現在、研究室内外に本システムを評価キットとして配布し、より広範囲の応用における評価を行っている。今後の予定として、本システムに対応した C 言語ベースの開発環境の整備、インタフェース拡張のための追加基板の開発のほか、センサの画素間ばらつきをソフトウェアでキャンセルする機構への対応も進めている。

文 献

- [1] 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.
- [2] 鏡 慎吾, 小室 孝, 石井 抱, 石川正俊, “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” 信学論 (D-II), vol. J84-D-II, no.6, pp.976-984, June 2001.
- [3] 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.
- [4] Y. Nakabo, M. Ishikawa, H. Toyoda, and S. Mizuno, “1 ms column parallel vision system and its application of high speed target tracking,” Proc. 2000 IEEE Int. Conf. Robotics and Automation, pp.650-655, 2000.
- [5] 石川正俊, 小室 孝, 鏡 慎吾, “デジタルビジョンチップの新展開 (特別招待講演)” 信学技報, ICD2002-39, 2002.
- [6] <http://www.micron.com/products/imaging/>
- [7] <http://www.dalsa.com/>
- [8] <http://www.fillfactory.com/>
- [9] 大明準治, 岡田隆三, 山本大介, “一般環境下における高速移動物体トラッキング — 1000 fps アクティブカメラシステムの試作” 第 21 回日本ロボット学会学術講演会, 1K13, 2003.
- [10] 石井 抱, “知的画素選択機能を有する高速メガピクセルビジョンの提案” 日本ロボット学会創立 20 周年記念学術講演会, 3A15, 2002.
- [11] U. Muehlmann, M. Ribo, P. Lang, and A. Pinz, “A new high speed CMOS camera for real-time tracking applications,” Proc. 2004 IEEE Int. Conf. Robotics and Automation, pp.5195-5200, 2004.
- [12] 小室 孝, 鏡 慎吾, 石川正俊, “ビジョンチップのための動的再構成可能な SIMD プロセッサ” 信学論 (D-II), vol. J86-D-II, no.11, pp.1575-1585, Nov. 2003.
- [13] 渡辺義浩, 小室 孝, 鏡 慎吾, 石川正俊, “ビジョンチップのためのマルチターゲットトラッキングとその応用” 信学論 (D-II), vol. J86-D-II, no.10, pp.1411-1419, Oct. 2003.
- [14] 鏡 慎吾, 小室 孝, 藤村英範, 石川正俊, “デジタルビジョンチップのためのソフトウェア A-D 変換手法” 映像学誌, vol.57, no.3, pp.385-390, 2003.
- [15] 鏡 慎吾, 小室 孝, 石川正俊, “デジタルビジョンチップの動的な感度特性制御手法” 日本機械学会ロボティク

ス・メカトロニクス講演会'02, 2P2-G06, 2002.

- [16] 鏡 慎吾, 小室 孝, 石川正俊, “実時間視覚センシングにおけるフレームレートの最適選択” 日本機械学会ロボティクス・メカトロニクス講演会'04, 2P2-L1-51, 2004.

(平成 16 年 5 月 20 日受付)



鏡 慎吾

平 10 東大・工・計数卒。平 15 同大学院博士課程了。現在、同大学院情報理工学系研究科システム情報学専攻助手。実時間センサ情報処理アーキテクチャ, アルゴリズム, システムの研究に従事。博士(工学)。



小室 孝

平 8 東大・工・計数卒。平 13 同大学院博士課程了。現在、同大学院情報理工学系研究科システム情報学専攻助手。ビジョンチップ, 並列プロセッサに関する研究に従事。博士(工学)。



渡辺 義浩

平 14 東大・工・計数卒。平 16 同大学院修士課程了。現在、同大学院情報理工学系研究科システム情報学専攻博士課程在学中。日本学術振興会特別研究員。実時間動画像処理に関する研究に従事。



石川 正俊 (正員)

昭 52 東大・工・計数卒。昭 54 同大学院修士課程了。同年通産省工業技術院製品科学研究所に入所。平元東大・工・計数助教授。現在同大学院情報理工学系研究科システム情報学専攻教授。超並列・超高速ビジョン, センサフュージョン, 光コンピュータ等に関する研究に従事。工博。