

## HwObject を用いた実時間画像処理システム

ソフトウェアからハードウェアをオブジェクトとして扱う, HwObject モデルを用いたロボット向け実時間画像処理システム

明官 佳宏<sup>†</sup> 工藤 健慈<sup>†</sup> ヴィン チャントーン<sup>†</sup> 藤田 政嗣<sup>†</sup> 関根 優年<sup>†</sup>

<sup>†</sup> 東京農工大学, 〒184-8588 東京都小金井市中町 2-24-16

E-mail: †{myo,k2x,wchanthan,fujita,sekine}@sekine-lab.ei.tuat.ac.jp

あらまし ロボット等で必要とされる低消費電力かつ高速実時間な画像処理を, ノイマン型計算機でのソフトウェア処理のみで行うことは困難であり, ハードウェアとソフトウェア (以下, Hw/Sw と呼ぶ) が協調して動作するアーキテクチャが不可欠であると考えられる. 我々は, ハードウェアを動的に生成, 削除するプラットフォームとして, FPGA を用いた再構成可能ハードウェア, HwModule の開発を行っている. またこれを用いた Hw/Sw 協調動作の為のアーキテクチャとして, ハードウェアをカプセル化しソフトウェアからオブジェクトとして扱う HwObject モデルを提案している. 本研究では, ロボットを想定した実時間画像認識システムを HwModule 上に構築することを目指して, 実時間画像処理を行う各種仮想回路 (HwNet) を HDL 言語を用い製作し, これらを HwObject としてカプセル化を行った上で, 実際に HwObject モデルのアプリケーションを製作し, その有用性の検証を行った.

キーワード 再構成可能ハードウェア, ハードウェアオブジェクト, ウェーブレット変換, テンプレートマッチング, オブジェクト指向設計

## Real-Time Image Processing System Using HwObject

Yoshihiro MYOKAN<sup>†</sup>, Kenji KUDO<sup>†</sup>, Winh CHANTHAN<sup>†</sup>, Masatsugu FUJITA<sup>†</sup>, and Masatoshi SEKINE<sup>†</sup>

<sup>†</sup> Tokyo University of Agriculture and Technology.

E-mail: †{myo,k2x,wchanthan,fujita,sekine}@sekine-lab.ei.tuat.ac.jp

**Abstract** It is difficult to make a robot which requires low power consumptions and real time high speed image processings only by the software implementation on a normal computer - von Neumann-type architecture. So the architecture which makes the cooperation of the hardware and the software(Hw/Sw) would be essential. In our previous studies, We proposed and developed the HwModule with three FPGAs as a platform to create and delete hardwares dynamically. And, for Hw/Sw cooperative executions, we proposed the HwObject model to use the hardware as an object from the software by the capsulation of the hardware. In this research, we aim to construct a real time image processing system for the robot. First we designed some virtual circuits(HwNets) and capsulate them as HwObjects, then we confirm their usefulness.

**Key words** reconfigurable computing system, hardware object model, wavelet transformation, template matching, object-oriented design

### 1. はじめに

#### 1.1 Hw/Sw 協調動作の必要性

ロボットでは, 大量の入力デバイス (カメラ, センサー) から入ってくる情報を高速に処理しなくてはならない. このよう

なシステムの構築において, 単一プロセッサによるソフトウェアのみでの実装では十分なパフォーマンスを得ることは難しい. しかしながら, これらを DSP 等の専用ハードウェアの組み合わせによって構築した場合, ハードウェア資源の無駄が多く, 柔軟性が乏しいシステムとなる. そこでハードウェアをソフト

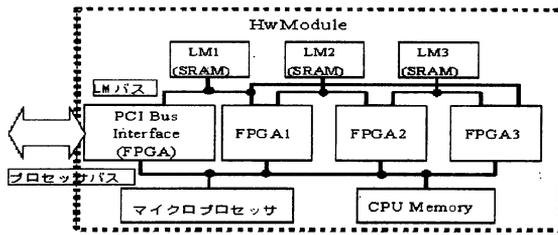


図 1 HwModule の内部構成図

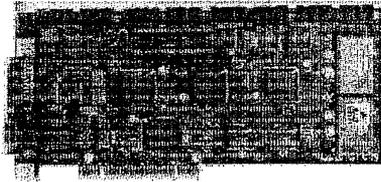


図 2 HwModule(TUAT-Ver1.0) の写真  
開発費：STARC，製造：来栖川電工

ウェアのようにオブジェクトとして扱うことができるアーキテクチャが必要であると考えられる。

## 1.2 本研究の目的

本研究では、ハードウェアをソフトウェアからオブジェクトとして扱う「HwObject モデル[2]」のアプリケーションとして、ロボットを想定した、2眼カメラによる実時間対象物追跡制御システムの構築を実際に行いその有用性の検証を行った。

## 2. Hw/Sw 混載システム

### 2.1 HwModule

Hw/Sw 混載システムは図1のようにホストコンピュータ及び PCI バス上に配置される HwModule[1] で構成される。HwModule は、図で示すように複数の FPGA, LM(SRAM), MPU, PCI BUS コントローラから構成されるメモリ型 PCI デバイスである。図2に、HwModule の写真を示す。Hw/Sw 混載システムでは、オブジェクト指向に適合するように、ハードウェア部分である仮想回路を隠蔽したオブジェクトを考えて、ソフトウェア上からハードウェアを意識しない形で利用可能である。このハードウェアのオブジェクトを HwObject と呼ぶ。

### 2.2 HwObject モデル

次の図3に HwObject モデルの概要を示す。HwObject は仮想回路 (HwNet)、入出力データ及び入出力関数からなり、それぞれ HwModule 上の FPGA, LM, ホスト PC のメインメモリ上に生成され、動的な生成、削除が可能である。図3における SwObject は従来のソフトウェアのオブジェクトであり、HwObject により SwObject がハードウェアと協調して動作しやすくなっている。

FPGA で実行される仮想回路 (HwNet) は、汎用の Verilog, VHDL 等のハードウェア設計言語を用いてデザインする。

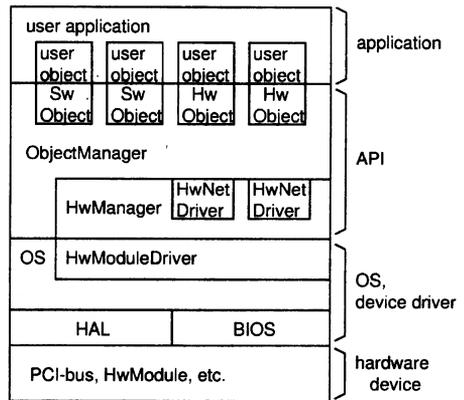


図 3 HwObject モデル

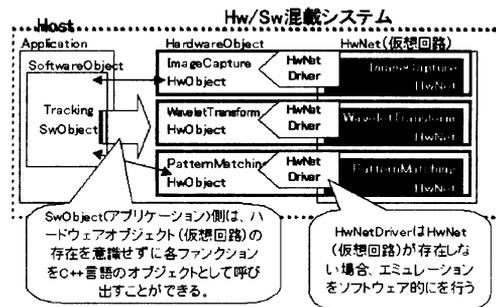


図 4 Hw/Sw 混載システムを用いた対象物追跡システムにおける各オブジェクトの概要

HwObject の基底クラスから派生させるハードウェアを含むオブジェクトはアプリケーションを作成するための C++言語のライブラリとして提供される。

### 2.3 HwObject モデルにおける各オブジェクトの振る舞い

ここでは、Hw/Sw 混載システムにおけるアプリケーション、SoftwareObject、HardwareObject の各オブジェクト間の関係について、本研究で構築する「対象物追跡システム」を例にとり説明を行う。図4に Hw/Sw 混載システムを用いた対象物追跡システムの概要を示す。Hw/Sw 混載システムが提供する各ファンクションは、SwObject 側からは HwObject (C++言語のオブジェクトのクラス) として提供される。HwNetDriver は基本的には HwObject に対して要求された処理を HwNet (ハードウェア記述言語により記述された仮想回路) に対し送り、処理結果を SwObject に返す動作を行うものであるが、状況に応じて HwNet が本来行うべき処理のエミュレーション等も行う。このようにして、HwNet を呼び出すことが出来ない環境であっても、HwObject アプリケーションは問題なく処理を続けることが出来る。次に、本研究のために製作した ImageCaptureObject を例にとり、オブジェクトの振る舞いの概要を説明する (図5)。ImageCaptureHwObject は、ImageCaputreHwNet(仮想回路)によって HwModule に外部接続された画像取り込み装置から

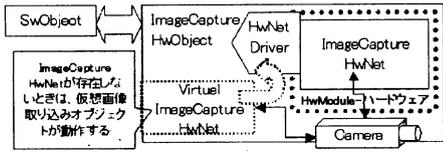


図5 画像取込オブジェクト

の画像取り込み動作の制御、画像データの取得等を担当するオブジェクトである。ここで、もし仮に ImageCaptureHwNet が存在しないとしても、HwNetDriver によってソフトウェア的に実現される VirtualImageCaptureHwNet が ImageCaptureHwNet の動作をエミュレーションするので、SwObject からアプリケーション側では特に変更を加えることなくシステムを稼働させることができる。

### 2.4 HwObject モデルを用いたアプリケーションの開発フロー

次の図6に、HwObject モデルのアプリケーションの開発フローを示す。通常のアプリケーションの開発手法と同様に①要求仕様の整理を行い、②必要な HwObject が何であるかを整理する。たとえば、対象物画像追跡のアプリケーションであれば、カメラから画像を入力する「画像取込」、入力画像を認識処理がしやすいような大きさにするウェーブレット変換などの「画像圧縮」、対象物の認識に用いる「パターンマッチング」、対象物を追従するように制御を行う「制御」のように、それぞれ HwObject による抽象化を行うレベルでの機能の切り出しを行う。それぞれの HwObject がライブラリに存在している場合には、直ぐにそれらのオブジェクトを用いて③-④アプリケーションの開発に取り掛かることができるが、必要とされる HwObject がライブラリに揃っていない場合は、新たに HwObject を製作しなくてはならない。HwObject の作成にあたっては、最初に機能モデルを作成し、これに基づく HwNet を汎用 HDL 言語等を用いて開発する。HwNet は、汎用論理合成・配置配線ツール等を用い、HwModule の FPGA 内に JTAG によりロード可能な形式であるビットストリームファイルの形で作成する。作成された HwNet (ビットストリームファイル) は、HwNetDebugger を用いて、制御 I/O やデータポートが正しく動作しているかどうかをデバックする。最後に、作成した HwNet を HwObject 機能モデルに従ってオブジェクトとして HwNet を呼び出し扱うにあたって必要なコード、ヘッダファイル等を作成し、HwNetDriver としてライブラリに登録を行う。

### 3. 画像処理 HwNet

本研究では、入力画像の取り込みから、対象物の認識、追跡動作までを全てハードウェアを用いて行うために、HwObject の実体である HwNet (仮想回路) として、以下のようなハードウェアを HDL 言語を用いて製作した。

#### 3.1 画像取込 HwNet

入力対象画像をリアルタイムで処理する為には、HwModule へ直接画像データを入力する必要がある。そこで、汎用小型

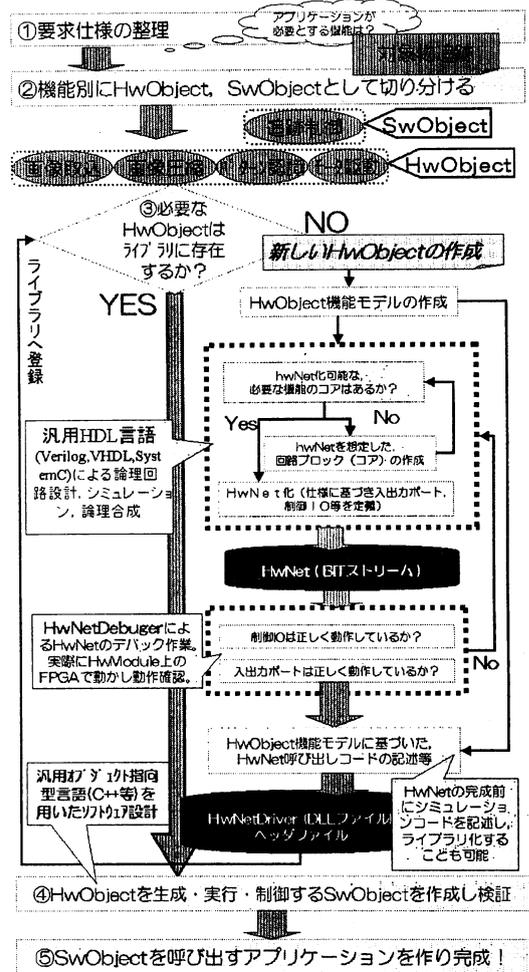


図6 HwObject モデルを用いたアプリケーションの開発フロー

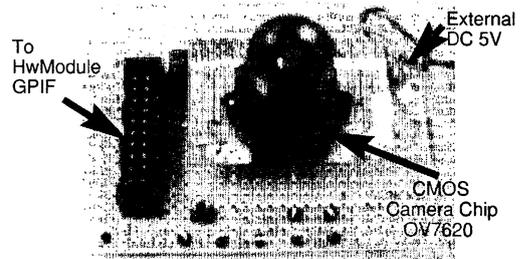


図7 HwModule 用画像取込装置

CMOS イメージセンサ(OmniVision, OV7620) から、実時間で HwModule の SRAM へ直接画像を入力する為の HwNet とハードウェア(図7)を製作した。ホストとの接続には、HwModule 上の各 FPGA の汎用外部入出力ポート(GPIF)を用いた。なお、OV7620 の制御には、i2c インターフェースが用いら

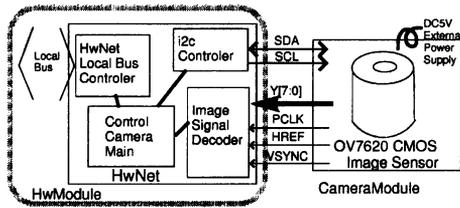


図 8 画像取込 HwNet の構成.

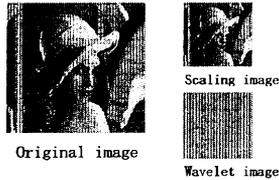


図 9 ウェーブレット変換

れており、オープンソースとして公開されている回路 IP<sup>(注1)</sup>を組み込んだ。図 7 に画像取込 HwNet の構造を示す。オブジェクト (ホスト) からの要求に応じて、任意のポート (LM) に対して画像を出力することが可能である。

### 3.2 画像圧縮-ウェーブレット変換 HwNet

本 HwNet は、画像取り込みオブジェクトと同一の FPGA 上に生成され、メモリを介さずカメラから取り込んだ画像に対して直接ウェーブレット変換を行うことが可能である。入力画像に対して図 9 に示すようなウェーブレット変換 [3] を行うことで、画像の情報量を減らし大きな画像から小さな画像へと段階的に処理を行う、多重解像度表現を用いた coarse-to-fine テンプレートマッチングなど、高速で高度な認識システムへの応用が期待できる。

### 3.3 画像認識-マッチング HwNet

画像認識処理における対象物認識用に、テンプレートマッチング (差分絶対値和演算) 回路を作成した。今回作成したテンプレートマッチング回路は、指定されたテンプレート画像と入力画像の一致度が最小になる場所の位置及び一致度を  $48 \times 48$  のサーチウィンドウ単位で出力する。今回ハードウェアによる並列演算の高速性を最大限に生かす為、差分絶対値和演算を行う PE (Processing Element) をストリクアレイ状 [4] に複数個並べテンプレートマッチングを行う。PE は、32 個単位で、グループ化がされ、搭載する FPGA のゲート規模などを考慮し、これらのグループを組み合わせることで、最大 256 個での構成が実現可能である。今回は、HwModule に搭載されている FPGA (Spartan2-2S200FG456) のゲート規模、BlockRAM のサイズを考慮して、2 グループの PE32 による 64 個構成のものを作成した。次の式 (1) に、マッチング回路内部の各 PE が演算を行う内容を示す。

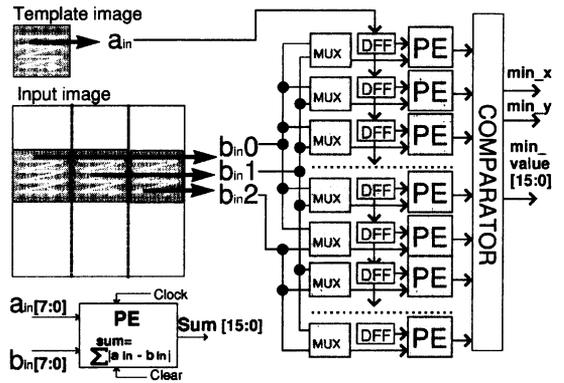


図 10 テンプレートマッチング回路の内部構成 (PE32).

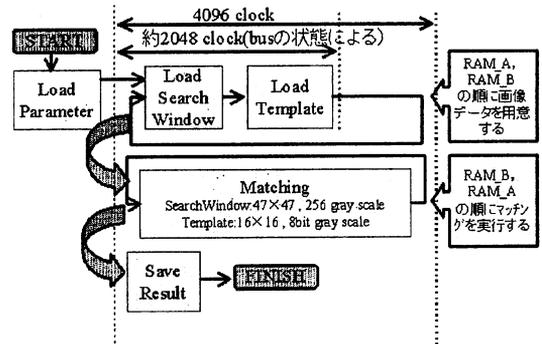


図 11 マッチングのシーケンス

$$M(x, y) = \sum_{i=0, j=0}^{15, 15} |A(i, j) - B(i + N_{pe} + x, j + y)| \quad (1)$$

PE は、 $M(m_x, m_y)$  の最小値と、その時の  $(m_x, m_y)$  ( $0 \leq m_x, m_y \leq 31$ ) を出力する。ここで、 $A(a_x, a_y)$  ( $0 \leq a_x, a_y \leq 15$ ) はテンプレート画像、 $B(b_x, b_y)$  は入力画像を示し、 $N_{pe}$  は、PE グループ (PE32) 内でのエレメント番号であり、 $x$  方向の演算位置 ( $0 \leq N_{pe} \leq 31$ ) となる。差分絶対値和演算回路を HwNet として実装を行った MatchingHwNet における処理の流れを図 11 に示す。今回製作した回路では、64 個の PE により構成されており、 $((48 - 16) \times (48 - 16) \times (16 \times 16) \div 64) = 4096$  クロック周期でマッチング処理を行い、結果を出力する。なお、マッチング処理のためのサーチウィンドウ、テンプレートデータを格納する一時 RAM は、HwModule 搭載 FPGA (Xilinx, Spartan2, 2S200FG456) 内の BlockRAM として、裏表 2 面生成してあり、データの準備と読み出しを各面で交互に行うことで、PE は一時も動作を止めず連続してマッチング処理を行うことが可能である。

(注 1) : <http://www.opencores.org/>

```
viewObject = new KViewObject;/--① SwObject 生成
```

図 12 アプリケーションによる SwObject の生成

```
ViewObject::ViewObject( void ) {
    LHNv hnv; //HwModule デバイス情報
    matchingObj = new MatchingObject; //②マッチングオブジェクト生成
    hnv.yHwModule = 1; hnv.yProgDev = 2;
    matchingObj->RefHwNetLink().dVector = hnv;
    camObj = new CamObject( matchingObj );//③画像取込オブジェクト生成
    hnv.yHwModule = 1; hnv.yProgDev = 1;
    camObj ->RefHwNetLink().dVector = hnv;
}
```

図 13 SwObject における HwObject 生成記述

```
void ViewObject::Operate( uint nSeq){
    switch ( nSeq ){
    case 0://④ camObject の実行
        camObj->ReqCapture();
        camObj->PostSignal( mSigThisP1( Operate, 1 ) );
        break;
    case 1://⑤ camObject からの画像取込
        camObj->Output( kaySrc );
        ExpandImage( 0 );//⑥ viewObject による表示
        matchingObj->Input( kaySrc );
        matchingObj->ReqMatching();//⑦マッチングの実行
        matchingObj->PostSignal( mSigThisP1( Operate, 2 ) );
        break;
    case 2://⑧マッチング結果の表示
        matchingObj->Output(mat);
        showMatchingResult(mat);break;
    default: ;
    } }
```

図 14 SwObject における HwObject の制御記述

## 4. ソフトウェアからのハードウェアの呼び出し

### 4.1 オブジェクトの呼び出しの記述

HwNet は、図 16 に示すように、SwObject へ HwObject として ObjectManager[2] を介して、ソフトウェア上のオブジェクトとしてカプセル化が施され提供される。アプリケーションが、SwObject を生成する記述を図 12 に示す。また、図 13 に各 HwObject 生成を行う記述例を、図 14 に SwObject においてイベント処理を用いて HwObject 制御を行う記述例をそれぞれ示す。

### 4.2 画像認識システムにおける HwObject の呼び出し

ここでは、簡単に呼び出しの様子の解説を行う。まず、①アプリケーションが、全体を統括する SwObject である ViewObject

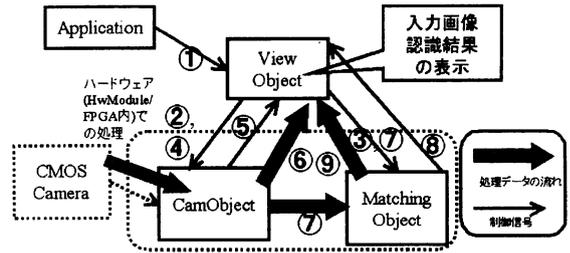


図 15 HwObject 呼び出しの様子

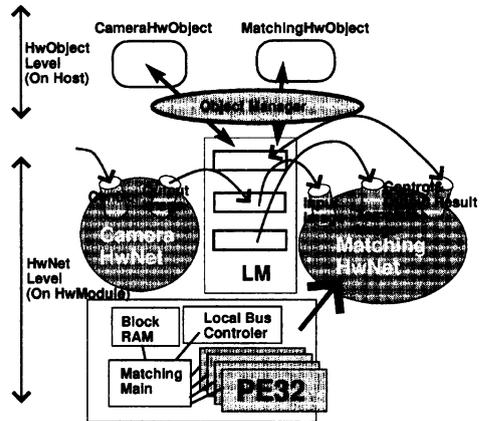


図 16 テンプレートマッチング回路の構成 (左下)、テンプレートマッチング回路が HwObject としてホストから呼び出されるイメージ

を生成する。以下、ViewObject による処理となり、② CMOS カメラから画像を入力する CamObject、③テンプレートマッチングを行う MatchingHwObject の各オブジェクトの生成が行われる。次に、④ CamObject を実行し、⑤画像取り込み完了の通知がきたら、⑥ ViewObject で表示を行い、⑦マッチング処理を実行する。最後に、⑧マッチング処理が完了したら、⑨ ViewObject でその結果を表示する。

## 5. 実時間画像認識実験

HwModule を用いた Hw/Sw 混載、HwObject モデルの実験アプリケーションとして、テンプレートマッチングを用いて対象物追跡動作を行う、実時間画像処理システムの構築を行った。

### 5.1 実験システムの構成

対象物追跡実験を行うにあたっては、CMOS イメージセンサ、小型 DC デジタルサーボモータから構成される二軸二眼カメラ「ステカム君」(図 17) を用いた。次の図 18 に、システム全体のハードウェア構成を示す。

### 5.2 対象物追跡アルゴリズム

対象物追跡システムの動作アルゴリズムを以下に述べる。①「ステカム君」の左右カメラから HwModule の LM(ローカルメモリ) へ画像を取り込む。②取り込まれた画像に対してマッ

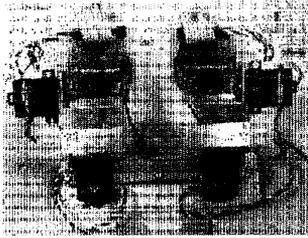


図 17 2軸2眼カメラモジュール「ステカム君」

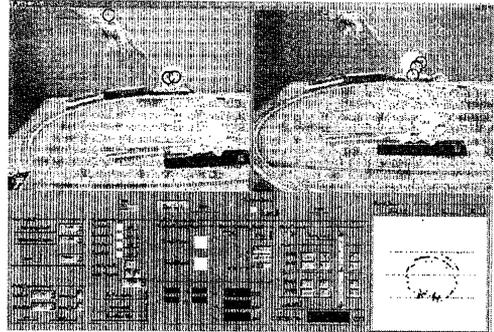


図 19 2軸2眼カメラによる追跡アプリの実行画面

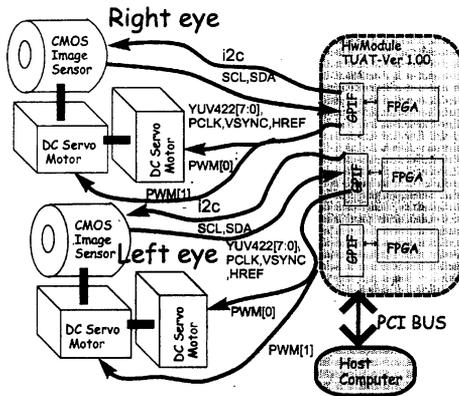


図 18 HwModule を用いた 2 軸 2 眼画像取込システムのハードウェア構成

チング仮想回路 (TemplateMatchingHwNet) が差分絶対値と演算を行い、対象物として与えるテンプレート画像が一番近い場所及び一致度を、サーチウィンドウ毎に出力する。③マッチング結果を用いて、対象物位置をソフトウェア (C++による記述) により演算する。④対象物が画面の中央に来るように、サーボモータへ位置決めパルス信号を発生する PWM 発生仮想回路 (PwmObject) に対し Duty 値を出力する。以上のようなアルゴリズムにより、入力された画像内の対象物を認識し、それを追跡する動作を行う。

### 5.3 対象物追跡システム動作評価

次の図 19 にテンプレートマッチングを用いて、実時間対象物追跡動作を行っている様子を示す。MatchingObject のテンプレートイメージとして「ボール」の断片画像を指定し、毎秒 10 フレーム程度で走行している鉄道模型の上に取り付けられた「移動するボール」を認識し、リアルタイムに追跡することができた。ホスト PC では、カメラによって取り込んだ画像のリアルタイム表示以外は殆ど処理資源を用いる事無く、画像取込、対象物認識、対象物追跡制御の一連の動作を行うことができる事が確かめられた。

## 6. まとめ

実時間対象物追跡の動作を、ホスト PC の処理資源を殆ど用いる事無く行うことができたことから、HwObject モデルがロ

ット等を想定したリアルタイム高速画像認識システムの開発等に有効であることが分かった。今後は、実際にウェーブレット変換オブジェクトなどを活用し、多重解像度表現を用いた coarse-to-fine テンプレートマッチングなど、高度な画像認識システムへと発展させていく予定である。今回は、画像認識が必要となった各種 HwObject の開発も同時に行った為、かなりの作業量・期間となった。本モデルを有効に活用する為には、再利用可能な HwObject ライブラリの充実が不可欠であると考えられる。

## 7. 付 録

本研究で製作した、各モジュールの回路規模、動作速度等を次に示す。

モジュール名	ゲート数	最大動作速度
カメラ制御	105,699	39.189MHz
マッチング (PE64)	278,266	41.869MHz
ウェーブレット変換	42,566	62.962MHz

## 文 献

- [1] 仮想回路をカプセル化した HwObject を実行する HwModule. 今中晴記, 志賀裕介, 工藤健慈, 上野貴史, 関根優年, 第 15 回路とシステム (軽井沢) ワークショップ論文集 pp.423-428 2002/4
- [2] Hw/Sw 混載システムに於ける HwObject モデルとその制御手法, 工藤健慈, 今中晴記, 志賀裕介, 関根優年, 情報科学技術フォーラム一般講演論文集第 1 分冊 F'IT2002,C-3 pp.193-194 2002/9
- [3] ウェーブレット変換の多重解像度表現を利用した画像認識システム, 伊藤光, 金丸隆志, 仁木雅, 工藤健慈, 関根優年, 信学技術報 (ITS) Vol.101 pp.71-76 2002/1
- [4] VLSI とデジタル信号処理, 谷萩隆嗣, コロナ社 (1997) pp.162-175,p202-208