

画像処理技術の組み込み分野への応用に関する研究（最終報告）

若生 進一* 富長 博*

1. はじめに

近年、施設のメンテナンスや防犯・セキュリティ等様々な分野で画像処理が注目されている。

一方で、開発に要求される仕様は、処理の高速化、機器の小型化など、これまでの汎用コンピュータによるソフトウェア処理のみでは実現が困難になってきている。さらに、製品開発の多様化は、開発期間の短期化と少量多品種化を促進してきた。

これらの課題に対応するハードウェアとして、内部回路のソフトウェア的な変更が可能であり、並列化による処理の高速化が見込める PLD に着目した。

PLD とは、内部回路を設計者が自由に変更することができる LSI で、これまでは、初期コストの高い専用 LSI の仕様が確定するまでの、試作開発に用いられることが多かった。

現在は価格も下がり、内部回路を自由に変更できるメリットを生かして、そのまま製品に組み込まれることも少なくない。

PLD を用いることで、画像処理を扱う製品に、機器の小型化、開発コスト低減等の付加価値を付与する可能性について研究を行った。

本研究は平成 19 年度～20 年度の 2 年間で特別電源所在県科学技術振興事業の補助金を受けている。

2. 目的

これまで汎用コンピュータや専用ボードを用いる事が多かった画像処理を、より安価なデバイスで実現し小型化するためのハードウェア実装技術及び画像処理アルゴリズムを用いた小型画像処理装置を試作する。

3. ハードウェア実装技術の検討

昨年度の研究により、ハードウェア記述言語 (HDL) による画像処理回路の開発及び PLD へのハードウェア実装が確認できた。

今年度は、よりソフトウェア開発に近い形で、回路の設計および開発を行うために、PLD と汎用のマイコンを組み合わせた構成と、PLD 上にソフトコアプロセッサ CPU および OS の実装を行った 1 チップ構成の実験装置を用い、ハードウェア処理による画像処理とソフトウェア処理による周辺装置の制御を実行させる実験を行った。

3.1 PLD とマイコンの組み合わせによる実装実験

実験装置として、写真 1 のように 50 万ゲート規模の PLD と 16bit マイコン、2Mbyte の RAM を実装した基板に、VGA 出力(640×480 画素)の CMOS カメラ、液晶モニタ(抵抗膜方式タッチパネル搭載)を接続した構成を用いて、実験を行った。

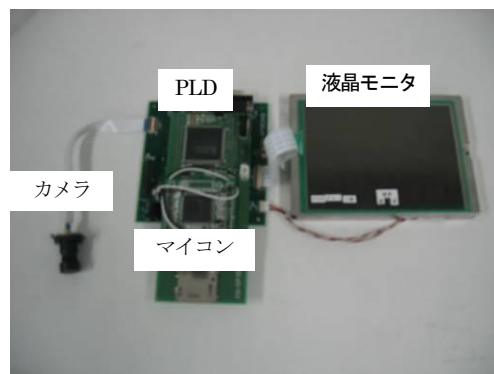


写真 1 実験装置 (PLD+マイコン)

PLD 上で実行するハードウェア処理として、HSV 変換、2 値化処理を実装し、カメラで取得した画像から特定の色の領域を抽出する回路を作成すると共に、マイコンで実行するソフトウェア処理としてカメラ制御及びタッチパネルの入力監視を実装した。

HSV とは、TV やディスプレイなどで色を表すのに良く用いられる RGB が 3 原色 (Red・Green・Blue) の比率で表す形式に対して、色空間と呼ばれる色相 (Hue)、彩度 (Saturation)、明度 (Value) という 3 つの値で表す形式である。色相で色の種類を表し、彩度・明度で色の鮮やかさ、明るさを表現するため、色から画像の特定の領域を抽出するような場合に、色相と彩度で判断でき、光の強さに影響を受けにくいという特徴がある。

RGB から色相 H、彩度 S を求める式を以下に示す。

$$H = \begin{cases} 60 \times (G - B) / (\max - \min) \cdots \cdots (1) \\ 60 \times (B - R) / (\max - \min) + 120 \cdots (2) \\ 60 \times (R - G) / (\max - \min) + 240 \cdots (3) \end{cases}$$

$$S = (\max - \min) / \max$$

※但し、 $\max = \max(R, G, B)$, $\min = \min(R, G, B)$

$$\begin{cases} R = \max \rightarrow (1) \\ G = \max \rightarrow (2) \\ B = \max \rightarrow (3) \end{cases} \text{ で計算}$$

$$H < 0 \rightarrow H = H + 360$$

$$(R, G, B \text{ は } 0 \sim 255, H \text{ は } 0 \sim 360)$$

今回の実験に使用した CMOS カメラはデフォルトでの画像出力方式が RGB ではない。これはカメラとの通信によって変更が可能であるため、マイコンのソフトウェア処理にカメラとの通信制御を実装し、電源入力時に RGB 出力に変更し、HSV 変換回路に入力する形式とした。

HSV 変換の後、2 値化処理を行った画像出力の結果を写真 2 に示す。

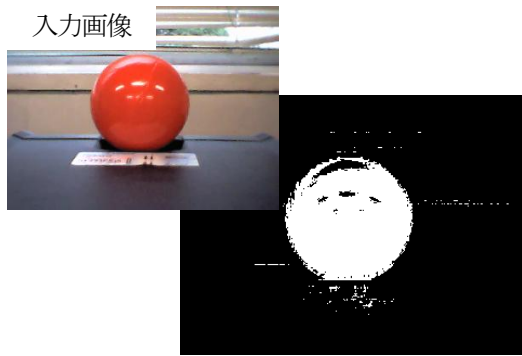


写真2 画像出力結果（2 値化）

入力画像から赤色を表す色相 0～20，彩度 80～255 の範囲にある画素を白く抜き出している。

さらに，この2 値化画像をマスク画像としてメモリに記憶し，入力画像に重ねることで領域抽出を行う回路を追加した。また，領域抽出の閾値を動的に変更するためにタッチパネルをスイッチとして用い，タッチパネルへの入力監視をソフトウェア処理で行うこととした。

領域抽出を行った画像出力結果を写真3 に示す。

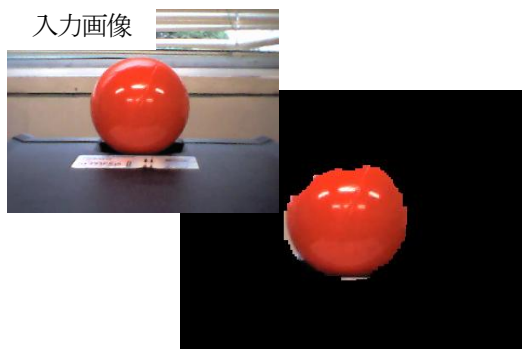


写真3 画像出力結果（領域抽出）

完全な領域抽出とはならなかったが，ノイズ除去回路も含めてほぼリアルタイムでの動作が確認できた。

3.2 PLD 単体による実装実験

3.1 で用いた実験装置からマイコン基板を外し，PLD 単体による画像処理実装実験を行った。

マイコンで行っていたソフトウェア処理を，PLD 上で行うために，MicroBlaze というソフトコアプロセッサを使用した。

ソフトコアプロセッサとは，PLD 内部に回路を構築する際に用いる論理合成によって実現される，PLD 向けのプロセッサである。

メモリや I/O 等の周辺機能を必要な機能のみに絞る等の PLD の特長を生かした開発が可能であり，複数のプロセッサを実装することで，機能分散型のカスタム LSI とすることもできる。

今回の実験では，カメラ制御とタッチパネル入力監視の機能を分散し，2 つのプロセッサで独立制御する構成とした。

結果として，3.1 で行った領域抽出実験と同等の結果が得られた。

4. 画像処理アルゴリズムの検討

監視画像などで動いているものの認識および追跡を行う場合，動きのある領域（動作領域）の認識および各フレーム間での領域の移動認識が必要となる。

今年度は，動作領域の切り出し及び移動認識を行うアルゴリズムについて検討し，汎用コンピュータによる動作確認実験を行った。

4.1 アルゴリズム動作確認実験

動作領域の移動認識アルゴリズムとして，処理フローを下記のように決定し，動作実験を行った。

- (1) カメラ画像を取得する
- (2) 取得画像に，昨年度作成したオプティカルフロー演算を行う
- (3) 抽出される動作領域にラベリング処理を行う
- (4) 複数の動作領域から最大領域を抽出する
- (5) 抽出した領域をテンプレートとし，次フレームで取得した画像にマッチングを行う

ラベリング処理とは，図1 に示すように条件にあった単位領域に対して番号を割り振り，隣接した領域をまとめて同じ番号とすることで分類・識別する処理である。

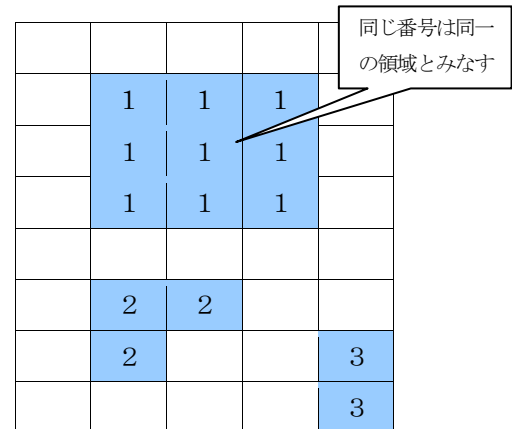


図1 ラベリング処理

実験結果を写真4 に示す。

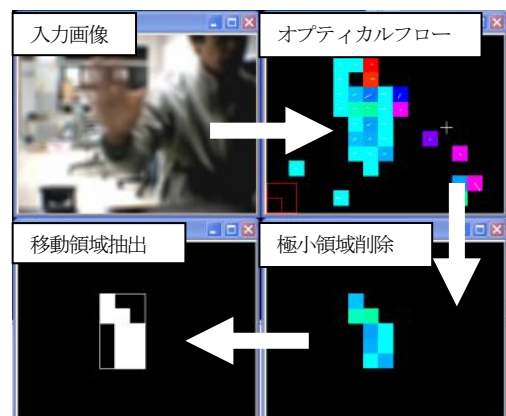


写真4 画像処理アルゴリズム出力結果

結果として、矢印の順に画像の取得、オプティカルフローによる動作領域抽出、ラベリングによる極小領域除去と進み、最後のテンプレート画像により移動領域を認識することが確認できた。

4.2 PLD へのアルゴリズム実装検討

4.1 の画像処理アルゴリズムについて PLD への実装を試みた。まず昨年度に成果として得たブロックマッチング法によるオプティカルフローを実装した。

結果として、数秒おきの更新で実時間とはいかないが動作領域の抽出が可能であることが確認できた。

しかし、ラベリング、テンプレートマッチングの追加実装を試みたところ、PLD の実装領域をオーバーしてしまったため、実験装置での実装は断念した。

5. 小型画像処理装置の試作開発

これまでの実験結果から、PLD 単体での小型画像処理装置の試作を検討し、回路の設計を行った。

5.1 装置仕様の検討

小型画像処理装置として、ネットワーク機能を持った画像検査装置を想定し、LAN 及びカメラインタフェースを有した PLD ボード「SUZAKU」を用いることとした。

写真 5 に SUZAKU ボードの外観を示す。



写真 5 SUZAKU ボード

写真でわかるとおり、SUZAKU ボードは名刺大のサイズで非常にコンパクトながら、PLD、メモリ、LAN コネクタ、シリアルコネクタを搭載し、オプションのボードで A/D コンバータや AV 端子等を追加することができる。

今回の試作では、SUZAKU ボードにオプションボードで AV 端子を追加し、NTSC 方式による画像の入出力ができる構成とした。

開発するシステム構成を図 2 に示す。

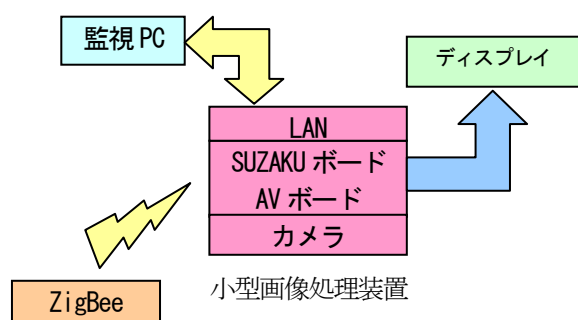


図 2 システム構成図

通信機能として、LAN とシリアルコネクタを用い、LAN は監視 PC と通信し、ブラウザから遠隔により画像を取得できるようにした。また、シリアルコネクタに ZigBee モジュールを接続し、周辺の ZigBee モジュールから通信により各種情報を得られるようにした。

ZigBee とは、近距離無線通信規格の一つであり、最大通信速度が 250kbps、最大伝送距離が見通しの良い所で 30m 程度ではあるが、端末がデータ中継機能も包含する為、最大で 65,535 台のモジュールを繋ぐメッシュネットワークと呼ばれる通信網を構成することができる。

メッシュネットワークは文字通り端末同士を網の目状に繋ぎ、データ中継機能を利用することで、端末同士が直接接続していなくてもパケットリレー式にデータを届けることができる等のメリットがある。

今回の開発には、ZigBee モジュールとして、DigiInternational 社製の「XBee」を用いた。

XBee モジュールは、モジュールのみで A/D コンバータ、I/O 等も備えており、追加回路なしでセンサとして活用できると共に、予め Telec 認証が取得されているため、手続き不要でそのまま装置に組み込むことができる。

大きさも写真 6 にあるように小さく、単 3 電池 2 本で直接駆動することができる。

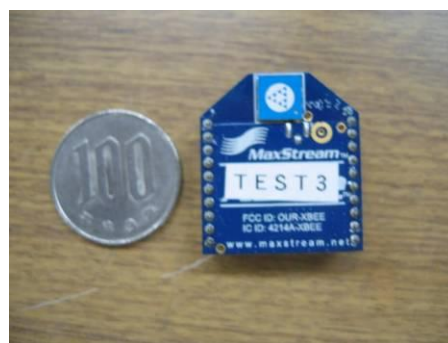


写真 6 XBee モジュール

5.2 PLD 実装回路の設計及び装置の試作

システムの動作フローは下記の通りとした。

- (1) ライン上を流れてくる検査対象物のデータを ZigBee モジュールを介して確認する。
- (2) 検査対象物を特定した後、検査画像の取得に必要なハードウェア処理に変更する。
- (3) 検査対象物がカメラのフレーム内に収まった時点で静止画像を取得し、監視 PC に送信する。
- (4) (1)に戻る。

3.2 での実験装置と同様に、MicroBlaze を実装し、ソフトウェア処理を行わせることとした。但し、今回は MicroBlaze 上で動作する OS (uClinux) の実装も行っている。これはシステム構成上、LAN による通信が必要となるため、TCP/IP プロトコルスタックの開発を省く目的で行っている。

画像を取得するタイミングを計るために、検査対象物の特定領域を監視するマッチング回路を作成した。処理の負荷を軽減するために領域を画像の中心に限定した。

試作に用いている SUZAKU ボードに搭載されている PLD は、120 万ゲート規模であり、全体の実装領域に対して上記の実装回路が占める割合は 30%程度となった。

ハードウェア処理回路の開発ツールとして、Impulse 社の ImpulseC/CoDeveloper（以下 ImpulseC）を用いた。ImpulseC は、ANSI-C 言語からハードウェア記述言語（HDL）を生成するツールである。

生成は ANSI-C 言語におけるプロセス単位で行われるため、ソフトウェア・ハードウェア処理の分割・混在の変更が容易であり、処理の並列化やボトルネックの解消に役立った。但し、ソースを作成する際に、独自の記述規則が有り、これに慣れる必要があった。

写真 7 に試作した装置を示す。

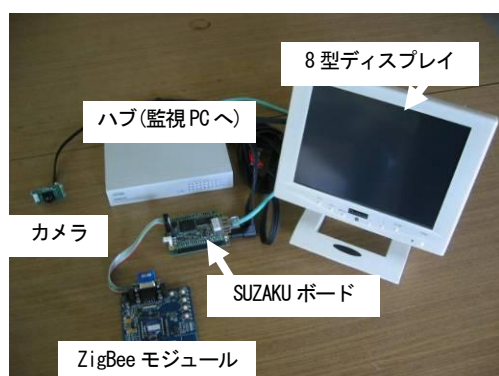


写真 7 試作した小型画像処理装置

5.3 小型画像処理装置のシステム動作実験

工場内における流れ作業を想定し、ZigBee モジュールと共に検査対象物が装置の前を通過する際の動作を確認した。写真 8 に実験の状況を示す。

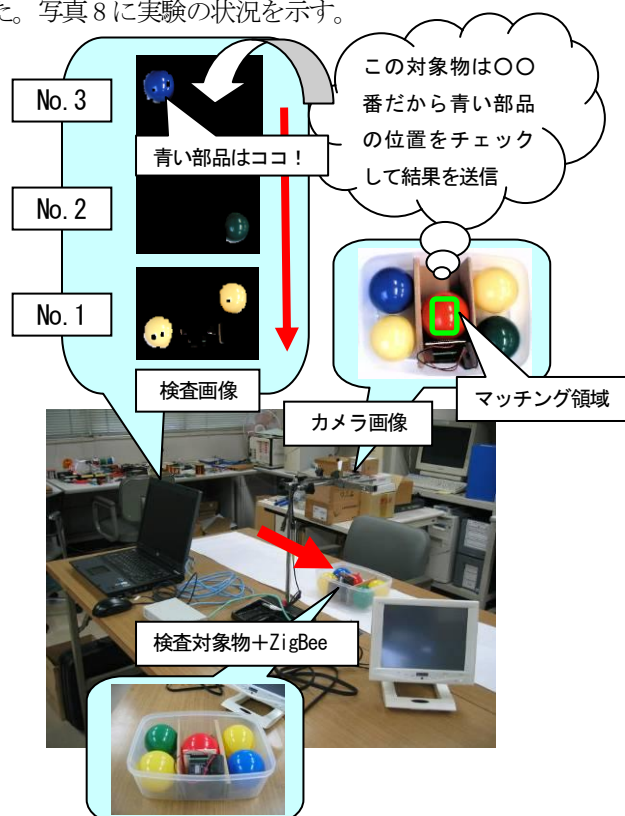


写真 8 システム動作実験

写真 8 において左から右に検査対象物を 5 秒に 1 個の割合で流したところ、ZigBee モジュールからのデータが逐次変化し、検査対象物の特定及び試作装置への接近を確認することができた。また、検査対象物がカメラの直下にきた際に行われる画像処理の変更及び監視 PC へのデータ送信が正しく行われていることが確認できた。

今後の課題としては、静止画像を取得するタイミングにズレが生じることがあったため、光に対するロバスト性の向上やマッチング処理でのノイズ除去等を検討すると共に、ZigBee メッシュネットワークのさらなる活用として、作業の進捗状況や諸注意等を画面に表示する教示機能の追加等を検討する。

6. まとめ

ハードウェア実装技術および画像処理アルゴリズムの検討を経て、小型画像処理装置の試作を行ったことで、以下の結果が得られた。

- ・PLD を用いることでマイコンを含む周辺回路を 1 つにまとめ、処理の並列化、機器の小型化といった付加価値を得られることが確認できた。
- ・PLD への OS の実装により、通信プロトコルの様な標準的ソフトウェアの開発負荷が軽減でき、開発期間の短縮及びコスト低減に繋がることが確認できた。
- ・動作領域の追跡アルゴリズムとして、オプティカルフローとラベリング、テンプレートマッチングの組み合わせが有効であることが確認できた。
- ・組込み機器にオプティカルフローやテンプレートマッチングを実装し、高速で処理するには、処理領域を限定する等の前処理の検討が必要であることがわかった。

今年度の研究を通して、ソフトコアプロセッサおよび OS の PLD への実装技術、動作領域の追跡アルゴリズムについて習得することができた。

次年度以降では、今回試作を行った小型画像処理装置の改善を進め、企業に提案すると共に、研究で得た成果を生かし、PLD を用いた装置開発及び画像処理の研究をさらに進めていく。