

# FPGA を用いたカメラ画像からの移動物体抽出の研究

## Extraction of Moving Objects from Camera Images Using FPGA

榑原 良太<sup>†</sup>, 市古 洋平<sup>††</sup>, 小林 鷹謙<sup>†††</sup>, 澤田泰幸<sup>†††</sup>, 江口 一彦<sup>†††</sup>

Ryota SAKAKIBARA, Youhei ICHIGO, Takanori KOBAYASHI, Yasuyuki SAWADA, Kazuhiko EGUCHI

Abstract: Image recognition technology is widely used in various application such as driveassist, pedestrianrecognition etc. However, real time processing is required in the most case of application. We employed FPGA(Field Programmable Gate Array) to implement image recognition algorithms for the purpose of faster operation compared to software approach. A motorcycle running on 60 km/h is captured and recognized in real time processing.

### 1. はじめに

人間は極めて優れた画像認識能力を有している。たとえば、近づいてくる小動物、それが犬であるのか、猫であるのか、あるいはその他の動物なのか、瞬時に判別できる。未知の小動物の特徴をつかみ、過去から現在までの記憶の中に蓄積された膨大なデータや経験を参照し、一瞬にして対象を識別し、認識を完了する。この人間の目と頭脳の持つ優れた機能・能力をコンピュータによって代行しようというのが画像認識である。それは、入力した画像から未知の対象物を認識し、数値や符号のデータとして、認識結果をわれわれに提供してくれるものである。

画像認識技術<sup>[1]</sup>は、画像処理技術の中の一分野である。コンピュータを使った画像認識処理により文字、記号、物体、その他を認識しようという試みは、文字の認識という分野で研究・実用化が真先に始まった。現在ではあらゆる分野に普及しており、交通事故防止、ドライバアシスト、防犯、リモートセンシング、医療画像、製品検査、ロボットの資格等盛んに行われていることがわかる。

<sup>†</sup>愛知工業大学工学研究科電気電子工学専攻(豊田市)  
<sup>††</sup>名三工業株式会社(名古屋市)  
<sup>†††</sup>愛知工業大学 工学部電気学科(豊田市)  
<sup>†††</sup>愛知工業大学 工学部電気学科(豊田市)  
<sup>†††</sup>愛知工業大学 工学部電気学科(豊田市)

これらの技術はリアルタイム性が求められる。そのようなシステムは専用のハードウェアを設計したほうが効率的であると考えられる。

以上のような考えから、本研究では FPGA(Field Programmable Gate Array)を用い、カメラ画像の取り込みと、画像中の移動物体を抽出するためのアルゴリズムを検証する。

### 2. 画像処理システムの設計

#### 2.1 システム構成

今回作成したシステム構成図を図 2-1 に示す。

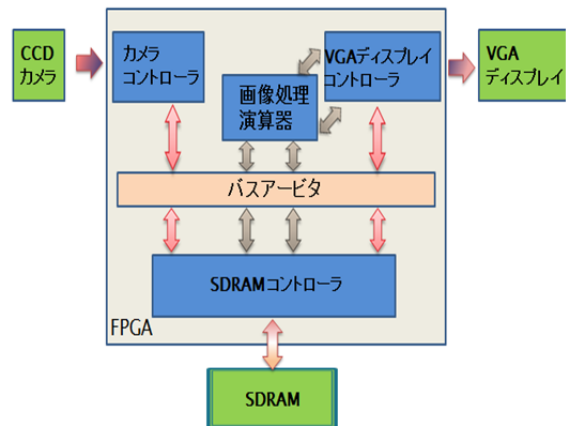


図 2.1 システム構成

2・2 カメラコントローラ

カメラコントローラでは CCD カメラから送られてくるデータを取得する。本研究で使用したカメラは ITU656 フォーマット<sup>[4][5]</sup>、YUV422 規格の 8bit 幅のデータが 27MHz の同期クロックに合わせて送られてくる。送られてくるデータはフレームの開始、終了を表すタイミング符号やブランキングデータも含まれている。この符号を取得し、画像データを構成する。送られてくる全データを画像とした時の図を図 2-2 に示す。



図 2-2 全データ画像

2・2・1カメラコントローラ構成

カメラから転送される画像情報取得するために必要な構成を図 2-3 に示す。

①カメラから入力されたデータを基にタイミング信号を取得する

図 2-2 のように全ての行に画像データの始まりを表す SAV(Start of Active Video)と終わりを表す EAV(End of Active Video)を検出する。検出方法は、カメラからのデータをバッファリングし、タイミング符号と比較し続ける。FF 00 00 の後に来るデータによってフィールド 1、フィールド 2 の画像を取得する。

② VGAサイズに切り出す

研究で使用したカメラは有効画素数が横 720 ピクセル、縦 468 ピクセルであるが、今回は PC で一般的に扱われる画像サイズである横 640 ピクセル、縦 480 ピクセルの VGA サイズに変更する。YUV データは図 2-4 のように U,Y,V,Y,U,Y…の形式で送られてくる。Y が 1 画素を表しているので Y を検出したらカウンタを動作させ VGA サイズにしている。

③ 書き込み

カメラからの 2 ピクセル分 32bit データを 8 つバッファリングしておく。アービタからの書き込み信号がきたら 8 つのデータを同時にアービタへ介して SDR SDRAM へ書き込んでいる。

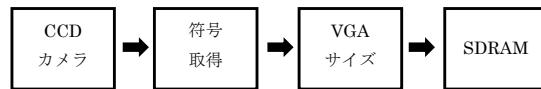


図 2-3 カメラコントローラ構成



図 2-4YUV データ

2・3 SDRAMコントローラ

カメラ画像の記録素子として、開発ボードに付属している SDR SDRAM を使用した。

SDR SDRAM の初期化タイミングチャートを図 2-5 に示す。

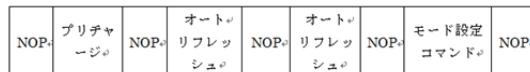


図 2-5 初期化タイミングチャート<sup>[2]</sup>

- ①プリチャージコマンドを発行する
- ②オートプリチャージコマンドを 2 回発行する
- ③バースト長設定のモード設定を行う

SDR SDRAM の書き込みタイミングチャートを図 2-6 に示す。

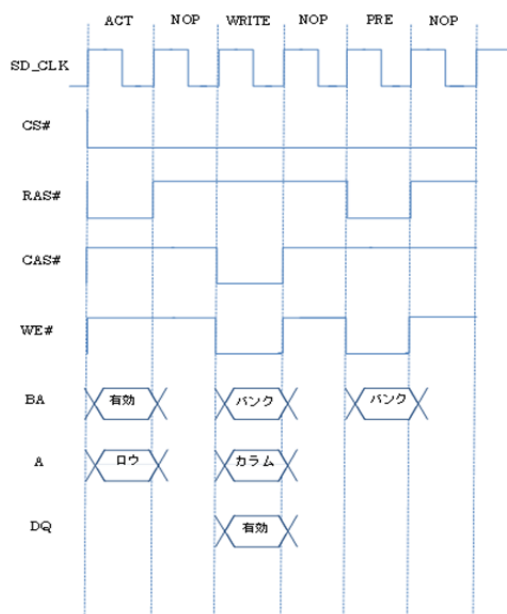
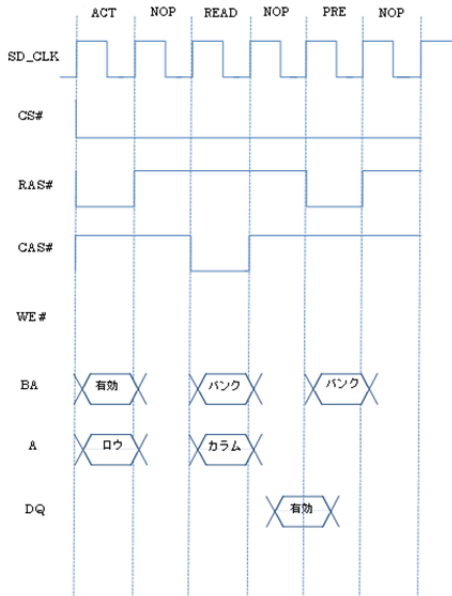


図 2-6 書き込みタイミングチャート<sup>[2]</sup>

- ① アクティブコマンドを発行してロウアドレスを SDRAM に与える
  - ② 1 クロック(15ns)後ライトコマンドと共に、カラムアドレスとデータを SDRAM に与える。
  - ③ ライトの 1 クロック(15ns)後にプリチャージする
- SDR SDRAM の書き込みタイミングチャートを図 2-7 に示す。

図 2-7 読み込みタイミングチャート<sup>[2]</sup>

- ① アクティブコマンドを発行して、ロウアドレスを SDRAM に与える
- ② 1 クロック(15ns)後、リードコマンドとともにカラムアドレスを SDRAM に与える
- ③ CL 後 SDRAM から出てくるデータを読み込む
- ④ リードコマンドの 1 クロック(15ns)後に、プリチャージをする

#### 2.4 VGAディスプレイコントローラ

取り込んだカメラ画像の出力用に VGA ディスプレイを制御するコントローラを作成した。ディスプレイとの接続は開発ボードに付属している VGA 端子を使用した。このボードの仕様では、RGB にそれぞれ 4 ピンずつ割り当て D/A 変換しているの、 $16 \times 16 \times 16$  の 4,096 色表示することができる。ディスプレイコントローラの動作周波数は  $800 \times 521 \times 60\text{Hz}$  の 25MHz である。

VGA 信号のタイミングは VISA(Video Electronics Standards Association)<sup>[3]</sup>により規定されている。タイミングに合わせて HSYNC、VSYNC を出力している。

SDRAM から読み込んだデータはUYまたはVYであるが、ディスプレイに表示するにはRGBに変換する必要がある。使用した変換式は式(2-1)である。

VGA ディスプレイコントローラの構成を図 2-8 に示す。

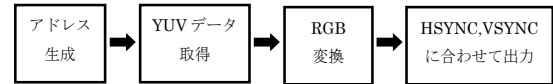


図 2-8 VGA ディスプレイコントローラ構成

$$R=Y+(V-128) \mu$$

$$G=Y-(U-128) \times 0.186-(V-128) \times 0.508$$

$$B=Y+(U-128) \mu$$

式 2-1 YUV-RGB 変換式

- ① 画像データを読み出すためにアドレスをSDRAMに渡す

アドレスは、バスアービタからの読み込み許可信号が入るとカウントアップして、バスアービタ経由で SDRAM に送られる。

- ② 渡したアドレスのYUVデータを受ける

データを受けたら、水平ピクセルカウンタと垂直ピクセルカウンタをカウントする。データはU,Y,V,Y,U,Y,V,Y...と送られてくる。この時 16bit のUY で 1 ピクセル、VY で 1 ピクセルなので、偶数ピクセルと奇数ピクセル、さらには上位 8bit か下位 8bit かを判断して、Y,U,V を判別する。

- ③ 式 4-1 の変換式に基づいてYUVをRGBに変換する

判別されたY,U,Vは、式 4-1 の変換式に基づいてRGBに変換される。

- ④ ディスプレイの制御タイミングに基づいて出力する

ディスプレイの制御タイミングは水平ピクセルカウンタと垂直ピクセルカウンタの値によって表示時間、フロントポーチ、バックポーチであるのかを判断する。

#### 2.5 バスアービタ

カメラコントローラ、VGA ディスプレイコントローラと画像処理演算器とで、SDRAM にアクセスする際に問題となるのが書き込み、読み込みの調停である。それを解消するためバスアービタを設計した。

#### 2.6 画像処理演算器

画像処理演算器では、二値化部(2.6.1)、背景差分部(2.6.2)、検出部(2.6.3)で構成される。

2・6・1 二値化部

SDRAM から VGA ディスプレイコントローラへ読みだされた YUV データから Y(輝度)を抽出し、閾値よりも大きければその Y(輝度)を 1 にし、閾値よりも小さければ Y(輝度)を 0 にした。二値化の処理の構成を図 2-9 に示す

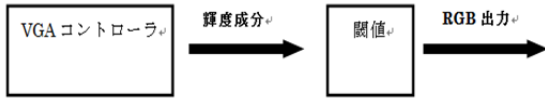


図 2-9 二値化部の構成

閾値は FPGA 付属のロータリースイッチによって変化させることができ、輝度成分は丸め処理で 8 ビットから 4 ビットに圧縮し RGB 出力している。そのため 0~15 までの 16 段階で設定できる。

2・6・2 背景差分部

二値化したデータを SDRAM へ送る。送られた二値化データを新しい二値化のデータと比較し差分を取る。差分を取るとき XOR 演算をして 0 の時は変化なし、1 の時は変化ありとして、0 なら黒、1 なら白とした。背景差分の構成を図 2-10 に示す。



図 2-10 背景差分部の構成

2・6・3 検出部

640×480 の画面を 100 等分する。1 マス(64×48)ずつ差分をかけて、閾値以上なら検出として、マスの外枠を赤色にすることで視覚化した。

検出部の構成を図 2-11 に示す。この閾値は 1 マス(64×48)のうち半分以上の画素が黒であった場合、検出としている。

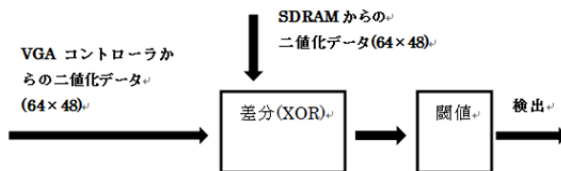


図 2-11 検出部の構成

3. 実験結果

3・1 論理合成結果

論理合成結果に関して結果を表 3-1 に示す。

表 3-1 論理合成結果

| ロジック     | 使用数  | 総数    | 使用率 |
|----------|------|-------|-----|
| LUT      | 5875 | 17264 | 34% |
| REGISTER | 4812 | 17264 | 28% |

3・2 カメラ画像取り込み結果

カメラ画像取り込み結果に関して、カラー画像を図 3-1、モノクロ画像を図 3-2、対象物が鮮明に映る閾値 8 の時の二値化画像を図 3-3、検出画像を図 3-4 に示す。



図 3-1 カラー画像



図 3-2 モノクロ画像



図 3-3 二値化画像(閾値 8)



図 3-4 検出画像(閾値 8)

### 3・3実験方法

実験に用いる動画撮影には画像検出できるFPGAボードを屋外に固定して設置できなかったため、市販のカメラ Sony Handycam HDR-SR1 を用いた。そのため、撮影した映像を PC で再生し、モニター越しでの検出を行った。検出対象は自動二輪車で、場所は愛知工業大学学生第二駐車場、撮影時間は 17:22 で天気は曇りである。

実験は図 3-5 の風景の中を自動二輪車が 10km/h～60km/h まで 10km/h ずつ速度を上げて横切る。動画のフレームレートは 30fps、解像度 340×280 のものを使用した。図 3-6 では背景画像に対してこの場合は自動二輪車が入ってきた場合、移動物体として 64×48 の赤枠を表示させて検出としている。



図 3-5 風景画像



図 3-6 検出画像

今回閾値が 8 の時が一番自動二輪車の検出率が高かった。二値化の閾値が 8 の時を例に自動二輪車に対する検出率のグラフを図 3-7 に示す。検出率は検出フレーム数/自動二輪車出現フレーム数×100 で定義する。

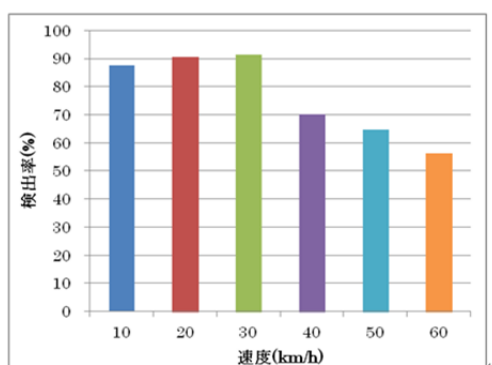


図 3-7 閾値 8 の時の自動二輪車に対する検出率

このグラフでは、速度は、10km/h、20km/h、30km/h の場合、90%近くの検出率が出ている。しかし速度が上がると、検出率は全体的に低下し、50km/h で 60%、60km/h では 56%の検出率であった。これは、リアルタイムでバイクを検出できているが、ビデオカメラとバイクの距離が近いためにフレームの移動距離が長くなり、バイクとして認識しづらくなっているからだと考える。

### 4. 結論

FPGA を使用して ITU656 フォーマットのカメラ画像の取り込みのための各コントローラ的设计を行い、処理演算として取り込み画像の二値化処理、差分法、移動物体の抽出まで行うことができた。

### 5. 今後の課題

今回は、背景差分法を用いて移動物体を検出している。実験では駐車場を背景に自動二輪車を移動物体として検出を行っている。変化している物体が自動二輪車のみであったが、雲が時間的に変化してしまい自動二輪車以外でも反応してしまうことがあった。この場合、雲は常に映像上部に位置しており、自動二輪車は地面を移動するため常に映像下部に位置している。そのため映像下部だけ検出するように検出範囲を狭めることで正確な移動物体検出が可能になると考える。

今回、背景差分法を用いるとき、差分する画像は二値化されている。検出対象を検出しようとする時背景と検出対象物が同じ色相をもっている場合、背景と同化してしまい検出されない。また背景差分法は背景全体が変わってしまうと、全て検出対象と見なしてしまう。この場合背景差分法ではなく、パターン認識を用いることで背景に関係なく認識できると考える。当研究室では、PC 上でのパターン認識の研究も行っている。その知識とノウハウを FPGA に移植することが出来れば FPGA を用いた画像認識の幅が広がると考える。

## 参考文献

- [1] 荒井寿光「画像処理-JPO」  
URL:[http://www.jpo.go.jp/shiryoku/s\\_sonota/map/denki02/frame.html](http://www.jpo.go.jp/shiryoku/s_sonota/map/denki02/frame.html)
- [2] デジタル・デザイン・テクノロジー No.10 CQ 出版社
- [3] XILINX 社「Spartan-3A/3AN スタータキットボードユーザーガイド」  
URL:[http://japan.xilinx.com/support/documentation/boards\\_and\\_kits/j\\_ug334.pdf](http://japan.xilinx.com/support/documentation/boards_and_kits/j_ug334.pdf)
- [4] 「ITU656 規格の信号を FPGA で撮り込む」  
URL:<http://bitcraft.web.fc2.com/embedded/itu656/itu656.html>
- [5] 「ITU-656 規格のカメラを使う」  
URL:<http://arx.ee.utsunomiya-u.ac.jp/azlab/members/sinsaku/old/itu656.html>

(受理 平成 24 年 3 月 19 日)