

## FPGA を用いたリアルタイム画像処理に関する研究 The research of real time image data processing by FPGA

佐藤 祥爾† 田中 和明† 安部 憲広†  
Shoji Sato Kazuaki Tanaka Norihiro Abe

### 1. 緒論

様々な分野で自律移動ロボットの研究開発が進められている。ロボットが自律移動を行うには、画像処理による環境認識や障害物回避などが必要である。

従来の画像処理での環境認識は、処理をロボットに搭載されている PC で行なっているが、この PC はロボットの制御も同時に行なわなければならない。そのため、結果的に膨大な情報を処理しなければならず、実時間処理が困難となってしまう。また、カメラと PC の通信の際にアナログ信号を A/D 変換する必要があり、その際に画像データにノイズが入る、フレームレートの低下等の不具合が発生してしまう。そこで、その解決方法として、全てデジタル信号で扱うことを考え、本研究ではデジタル出力カメラと FPGA (Field Programmable Gate Array) を用いる。

本研究の目的は、FPGA を用いて画像処理デバイスを実現することである。ロボットの制御に必要なものは、画像データそのものでなく、画像処理を行なった結果である。必要とする結果のみを PC へ渡すことによって、システムの処理時間が短縮され、実時間処理できると考える。

### 2. ビデオ信号

#### 2.1 ITU-R BT 656

ITU-R BT.656 は、国際電気通信連合 ITU(International Telecommunication Union)によって策定された、ITU-R BT.601 で作動する 525 ライン(NTSC)と 625 ライン(PAL)のテレビジョン方式のデジタルコンポーネントビデオ信号のためのインターフェースである。色の表現には YUV422 方式が用いられている[1]。

#### 2.2 ITU-R BT 601

ITU-R BT.601 とは、ITU によって策定された、アナログビデオ信号をデジタルデータに変換する際のデータ形式を表した規格である。

アナログの RGB 信号( $R_a G_a B_a$ )からデジタルの RGB 信号( $R_d G_d B_d$ )の量子化方法は次のように規定されている。

$$R_d = 219R_a + 16 \quad (2.1)$$

$$G_d = 219G_a + 16 \quad (2.2)$$

$$B_d = 219B_a + 16 \quad (2.3)$$

また、デジタル信号  $Y C_R C_B$  への信号変換式は、

$$Y = 0.301R_d + 0.586G_d + 0.113B_d \quad (2.4)$$

$$C_R = 0.512R_d - 0.430G_d - 0.082B_d + 128 \quad (2.5)$$

$$C_B = -0.172R_d - 0.334G_d - 0.512B_d + 128 \quad (2.6)$$

より表される[2]。

#### 2.3 YUV422

YUV422 方式とは、サンプリングの際に  $2 \times 2$  の 4pixel から、輝度情報 Y を 4 サンプル、輝度と青色成分の差 U を 2

サンプル、輝度と赤色成分の差 V を 2 サンプル採る方式である。8bit 量子化を行なった場合は 4pixel で 64bit、つまり 1pixel あたり 16bit となる。

### 3. システム構成

システム構成の概要を説明する。PC と FPGA 間の通信には USB2.0 を使用する。PC からの画像処理命令が FPGA に入ると、カメラからの出力データを FPGA に取り込み、処理内容に応じて記憶装置に一旦記憶する。そして画像処理後の結果を、PC へ出力する。全体のブロック図を Figure3.1 に示す。

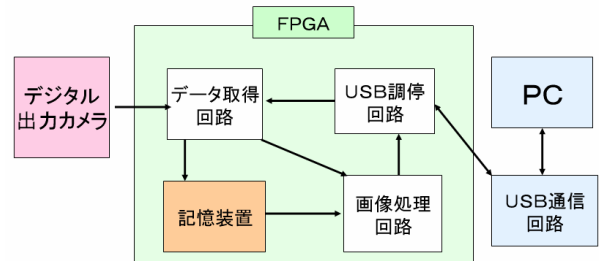


Figure3.1 システム構成

デジタル出力カメラは、MINTRON 社製の CCD カメラ MTV-54K0DN を使用する。このカメラは、27MHz で動作し、ITU-R BT 656 に沿った形式で画像データを出力する。USB 通信ボードには、FTDI 社製の FT245BM を搭載している Xilinx 社製の USB-004-FIFO を使用する。

#### 3.1 デュアルポートブロック RAM

記憶装置には、FPGA 内に含まれているデュアルポートブロック RAM を使用する。

デュアルポートブロック RAM は、ポート A とポート B の 2 つのポートから共通のブロック RAM にアクセス可能である。FPGA に入っているブロック RAM の個数は FPGA の容量によって決まっており、それによって必然的に作れる最大メモリ容量は決まる。基本的な 4 つのデータパスを Figure3.2 に示す。

1. ポート A への書き込み及びポート A からの読み出し
2. ポート B への書き込み及びポート B からの読み出し
3. ポート A からポート B へのデータ転送
4. ポート B からポート A へのデータ転送

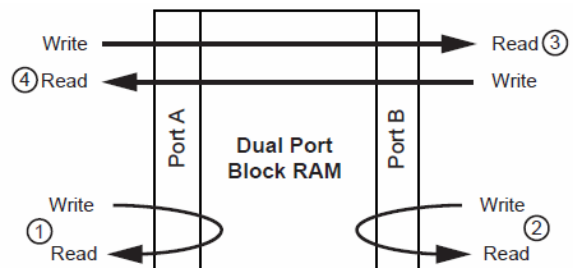


Figure3.2 ブロック RAM のデータパス

4. 実験

フィルタリングは特徴点検出やエッジ処理など、ほとんどの画像処理に用いられる処理である。FPGA で実現することで、より複雑なステレオマッチングなどの処理をFPGA で実現が可能であることが示される。今回はガウシアンフィルタを考え、その実現を目的とする。

FPGA には Xilinx 社の Spartan-3E(xc500e-4fg320)を用いる。ブロック RAM 容量は 368640bit である。

4. 1 ガウシアンフィルタ

ガウシアンフィルタとは、画像のノイズ除去を目的とした平滑化フィルタの一つである。画素の空間的配置を考慮して、対象画素に近い画素には大きな重みを、対象画素から遠い画素には小さい重みを付けた加重平均を取る。2 次元のガウス関数は次のようになる。

$$f(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (4.1)$$

標準偏差  $\sigma$  の値によって平滑化の度合いを変化させることができる。

4. 2 オペレータ

今回は、横方向 1 次元の 5 画素分のガウシアンフィルタを考える。分母を 2 の乗数になるように近似することで、乗法、除法を論理シフト演算で行うことができ、高速な演算が可能となる。式(4.1)において  $\sigma=1$  とすると、ガウシアンフィルタのオペレータは Figure4.1 のようになる。

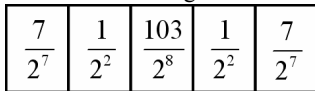


Figure4.1 ガウシアンフィルタのオペレータ

4. 3 パラレル処理

Figure4.1 のオペレータを用いて演算を行うには、連続した 5 画素分のデータを同時に扱う必要がある。そのため、カメラからの画像データを 5 つのブロック RAM に順番に保存していく。同様に演算回路も 5 つ配置し、Figure4.2 のように接続する。

ブロック RAM に 5 画素分のデータを保存した直後にデータを読み出し、演算回路への画像データ送信を開始する。ブロック RAM のタイミングを調整しつつ画像データを 1 つずつシフトさせながら入力し、それぞれの演算回路で処理を行う。このように並列に処理することで、リアルタイムな演算が可能となる。

演算後の結果は、データ処理回路でまとめられた後にバッファリング回路に送られ、USB 調停回路を通し順次 PC に送信される。

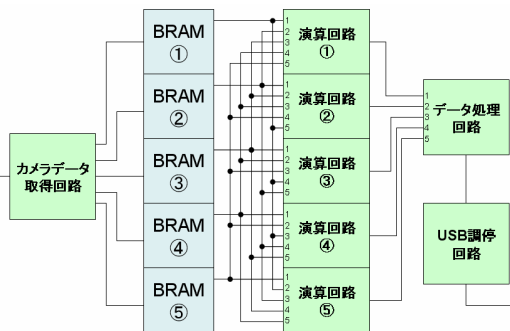


Figure4.2 パラレル処理

5. 結果

画像処理の対象をフィールド 1 の輝度値のみとし、回路を作成した。作成した回路の論理合成結果を Table5.1 に示す。画像 1 枚の処理時間は約 17ms である。

Table5.1 論理合成結果

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	1096	9312	11%
Number of input LUTs	1919	9312	20%
<b>Logic Distribution</b>			
Number of occupied Slices	1318	4656	28%
<b>Total Number 4 input LUTs</b>	1977	9312	21%
Number used as logic	1919		
Number used as a route-thru	58		
Number of bonded IOBs	37	232	15%
IOB Flip Flops	16		
Number of Block RAMs	20	20	100%
Number of GCLKs	1	24	4%
<b>Total equivalent gate count for design</b>	1333806		
Additional JTAG gate count for IOBs	1776		

平滑化ができていないかを確認するために、縦線をカメラに映し画像処理を行った。結果の検証のために元画像と FPGA での処理後の画像の一部をそれぞれ拡大した。元画像を Figure5.1 に、処理後の画像を Figure5.2 に示す。



Figure5.1 元画像



Figure5.2 処理後の画像

Figure5.2 の縦線の境界を見ると平滑化されていることがわかる。

6. 結論・考察

本研究では FPGA を使用し、リアルタイム画像処理を実現することを目標とした。その過程で、画像処理に必要不可欠であるフィルタリング回路を、1 枚の処理時間が約 17ms とリアルタイムに実現することができた。しかし、USB2.0 の通信速度(1Mbyte/s)や、ブロック RAM の記憶容量など、解決しなければならない課題も多く残っている。

今後は画像処理部分の範囲指定や、分割処理、外部記憶装置の使用などを試み、小型で高速に移動できるロボットの開発につなげていきたい。

7. 参考文献

- [1] ITU, "Interfaces for digital component video signals in 525-line and 625-line television systems operating at the 4:2:2 level of Recommendation ITU-R BT.601(Part A)", 1998.
- [2] ITU, "STUDIO ENCONFIG PARAMETERS OF DIGITAL TELEVISION FOR STANDARD 4:3 AND WIDE-SCREEN 16:9 ASPECT RATIOS", 1995.
- [3] インターフェース編集部, "USB ターゲット機器開発のすべて", CQ 出版社, 2005.
- [4] Xilinx, "Spartan-3E FPGA ファミリー: データシート (全モジュール)", 2006.