

顔領域抽出機能を持つビデオカメラについて

Video camera with face area extraction function

山本 貴彦[†] 川村 尚生^{††} 菅原 一孔^{††}

Takahiko YAMAMOTO[†] Takao KAWAMURA^{††} Kazunori SUGAHARA^{††}

[†] 鳥取大学大学院 工学研究科 ^{††} 鳥取大学 工学部

1 はじめに

現在、顔画像処理技術は様々な分野で利用されつつあり、ヒューマンインターフェースや監視システムなどへの応用を目指し、さらなる高機能化のため活発に研究されている [1]。顔画像処理の最も基本的な処理として、任意の画像中から顔領域を抽出したり、それを追跡することが挙げられる。通常、この顔領域抽出や追跡などの顔画像処理は、すべてソフトウェアにより実現されている。しかし、顔領域特定をソフトウェアで行うと、処理速度が PC の計算能力に依存してしまう。またコンピュータが必須となりシステムの構成が複雑になることが予想される。

したがって、カメラなどのハードウェア側に顔領域抽出や追跡の機能を持たせるということが一つの有効な手段として考えられる。そこで本研究ではビデオカメラからの入力画像から顔領域抽出の処理を行うことができるハードウェア装置の開発を目指している。ハードウェア化することによって、顔画像処理のためのシステムを小型化することが可能になり、システム全体としての処理速度の向上も期待できる。顔領域の抽出方法にはカメラからの入力画像に対して、あらかじめ用意した顔画像のテンプレートと比較し、類似した領域を抽出するテンプレートマッチングを用いている。また探索方法には大域的な探索と局所的な探索が可能な遺伝的アルゴリズムを使用しており、入力画像内の端から全領域を比較するよりも効率よく解を得ることができる。

開発するビデオカメラの利用方法の一例として、図 1 に示すように、室内における TV 会議での利用が考えられる。遠隔地で会場を撮影した映像を見ている側からすれば、発話者の様子だけでなく会議全体の雰囲気を知ることが重要である。顔領域抽出機能を備えたビデオカメラが実現すると、発話者の様子と会場全体の様子という 2 種類の映像をモニタに表示することが可能となり、会場の雰囲気のより詳細な情報を得ることが期待できる。

今まで我々は顔領域抽出処理のためのハードウェア開発について検討を重ねてきた [2][3]。本稿では顔領域抽出アルゴリズムの FPGA 上への実装とその動作について報告する。

2 システム構成

システムの構成を図 2 に示す。本システムは NTSC ビデオカメラ、デコードボード、SSRAM、FPGA ボー

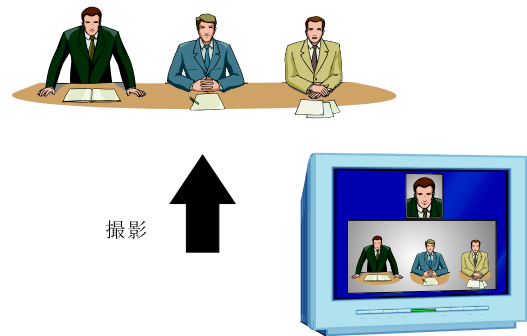


図 1: 開発するハードウェア装置の利用例

ド、エンコードボードから構成されている。NTSC ビデオカメラから送られてきたアナログ映像信号をデコードボードを介することによってデジタルの RGB 各 8bit のデジタル信号に変換し FPGA ボード上の SSRAM に蓄積する。そして FPGA 上の各回路において、入力された画像データ中から顔領域特定の処理を行い、エンコードボードでアナログ信号に変換しモニタに表示する。

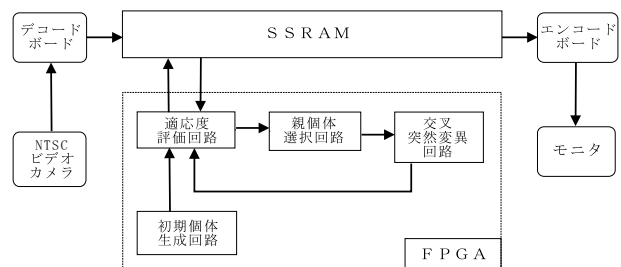


図 2: システム構成

3 顔領域の特定方法と回路構成

本研究における顔領域の特定方法は次のとおりである。あらかじめ 60×74 [pixel] の顔の画像をテンプレート画像として用意しておく。そして、 320×240 [pixel] の入力された画像内からテンプレート画像に類似した領域を比較し特定するテンプレートマッチングを遺伝的アルゴリズムを用いて行った [4]。

FPGA 上には遺伝的アルゴリズムを実現するために初期個体生成回路、適応度評価回路、選択回路、交叉・

突然変異回路をそれぞれ作成した。以下に示した一連の処理を繰り返す。

1. 初期個体生成回路内において乱数を用いて各個体に遺伝子コードとして設定する。
2. 次に適応度評価回路で設定された個体情報をもとに入力画像とテンプレート画像を比較して評価値を各個体に付加する。
3. そして親個体選択回路では交叉に用いる個体の選出を行う。
4. 送られてきた個体を利用して交叉・突然変異回路で交叉および突然変異の遺伝的操作を実行し、次世代の個体を作成する。
5. その後、適応度評価回路において新たに作成された個体に対して評価を付加する。

この時に用いた遺伝的アルゴリズムのオペレーションとしては選択方法はランク戦略、交叉方法は一様交叉、突然変異は一点突然変異である。以下にそれぞれの回路について詳しく説明する。

3.1 初期個体生成回路

入力画像中で探索に用いる個体数は 32 とし、それぞれの個体に乱数を設定する。使用する乱数は FPGA 上の内部メモリに蓄えられており、使用する都度読み込んでいる。各個体の遺伝子コードは入力画像中で探索に指定する領域の左下端の x, y 座標を表しており、上位 9bit が x 座標、下位 8bit が y 座標で構成されている。また入力画像の左下を $(0,0)$ として横方向を x 座標、縦方向を y 座標として定義しており、このとき指定する領域はテンプレート画像と同じサイズである。また指定領域が入力画像からはみ出した場合や入力画像外を探索するような座標値が得られた場合は除外し、再設定している。遺伝子コードの構成を図 3 に示す。

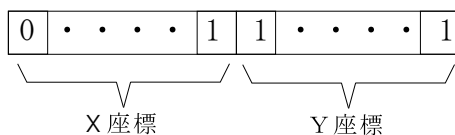


図 3: 遺伝子コード

3.2 適応度評価回路

それぞれの個体で指定した領域において RGB 値を読み取り FPGA の内部メモリに保存されているテンプレート画像と縦横一画素おきに比較を行う。比較方法としては RGB 成分それぞれで差分をとり、その領域における差の絶対値和を評価値として各個体に付加する。このとき評価値の値が小さい方がテンプレート画像により近い領域であると考え、設定された評価値が小さいものほど適応度が高い個体とした。また評価値にしきい値を設定し、差分の加算結果がそのしきい

値を越えると指定した領域はテンプレート画像とは似ていない領域とみなすこととする。そしてその後の比較処理を中断し、評価値には最大の値を設定するようにしている。比較する領域の指定を図 4 に示す。

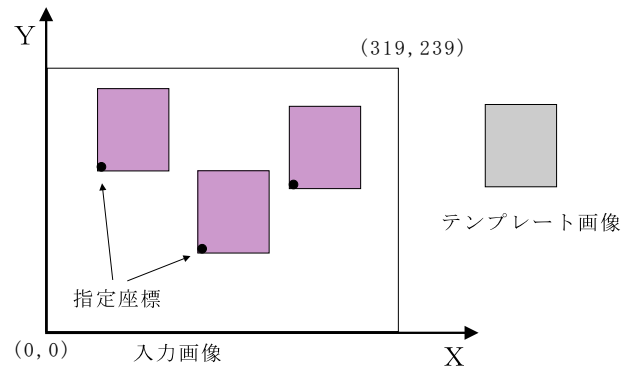


図 4: 比較する領域の指定

3.3 親個体選択回路

適応度評価回路において設定された評価値をもとにして各個体を昇順に並び換える。なおこのときのソート方法にはクイックソートを用いている。そして 1 番目から 16 番目までの上位 16 個体の中から、乱数で交叉に用いる親個体となる 2 個体を選び出す。このとき親個体として選択される確率はランクづけされた順位に依存している。

3.4 交叉・突然変異回路

親個体選択回路から選ばれた 2 個体と乱数の bit 列を用いて一様交叉を行う。乱数はマスクの役割をしており、各 bit をチェックし “1” となっている箇所のみ親個体の値の入れ換えを行い子個体を作成する。

交叉を終えた個体に突然変異を行うかどうかを決定するために各個体に対して乱数を与える。そしてその乱数が突然変異率を満たしていれば、bit 反転を行う遺伝子座をさらに乱数を用いて決定する。その後 32 個体のうち後半の 17 番目以降の個体に新たな個体として上書きする。

以上の作業をくり返すことによって次世代の個体群の作成を行う。適応度が最も高かった個体を特定した顔領域としている。

3.5 回路規模

これまでに述べた遺伝的アルゴリズムを構成するのに要した各回路の logic element 数を表 1 に示す。

4 使用機器

本研究で使用している機器を図 5 に示す。FPGA ボードに搭載している FPGA は Altera 社製の APEX20KC である。この FPGA における使用可能な logic element 数は 8320 であり、これは回路規模 20 万ゲートに相当する。また FPGA ボードには外部メモリとして 36bit 幅の 18MbitSSRAM が 2 つ搭載されており、それぞれバスが独立しているので 2 つのメモリに同時アクセ

表 1: 回路の logic element 数

回路名	logic element 数
初期個体生成回路	226
適応度評価回路	931
選択回路	1207
交叉・突然変異回路	2005
合計	4369(52%)

スが可能である。NTSC ビデオカメラは μtec 社製の IVC-400 を使用しており、有効画素数 $768 \times 494[\text{pixel}]$ である。またビデオカメラからのアナログ信号をデジタルの RGB 信号に変換するデコードボードと、デジタル信号をアナログ信号に復元するエンコードボードを FPGA ボード上に実装している。

論理回路設計は Innoveda 社の Visual Elite を用いて VHDL で記述し、論理合成は Synplicity 社の Synplify Pro 7.3.1 を使用した。配置配線、検証、FPGA への書き込みには Altera 社の QuartusII2.2 を用いた。

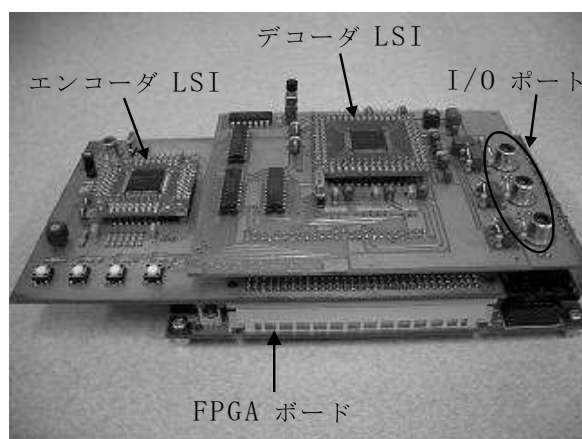


図 5: FPGA ボードとデコードボード及びエンコードボード

5 動作実験

FPGA 上に作成した回路の動作実験としてビデオカメラから転送された入力画像の中から顔領域の抽出を行った。その手順としては、NTSC カメラから入力画像を、PC からはテンプレート画像の RGB 情報を FPGA ボード上の外部メモリである SSRAM に転送し、それぞれ決められた領域に格納する。次にそれらの情報をもとにして FPGA 上の回路を動作させることによって顔領域の探索を行う。そして最後に顔領域が正しく特定できているかを検証するために、画像に特定した領域を線で囲み PC へと転送し、ディスプレイ上に表示する。このとき実験に使用した入力画像のサイズは $320 \times 240[\text{pixel}]$ で、テンプレート画像のサ

イズは $60 \times 74[\text{pixel}]$ である。

5.1 実験 1

実験 1 で用いたテンプレート画像を図 6 に示す。テンプレート画像には入力画像中で被写体として写っている人物と同じ人物の顔の画像を用いている。一般にテンプレートマッチングでは検出対象とテンプレートが異なると認識処理に影響が生じるという問題があるので、前処理としてテンプレート画像にぼかす処理を行っている。このような処理を行った画像を用いることによって、多少の異なりがあったとしても特定に大きく影響しないため、テンプレートマッチングでは有効な手法として用いられている。

このときの実験結果を図 7~図 9 に示す。図 7 は探索の初期の状態である第 1 世代目の結果である。図 8 は途中経過にあたる第 5 世代目の結果である。図 9 は探索がほぼ収束した状態である第 10 世代目の結果である。世代数が進むにつれて収束してゆく様子が確認できる。また回路動作開始から終了までの時間を計測したところ、現段階では約 0.9 秒要することがわかった。



図 6: テンプレート画像

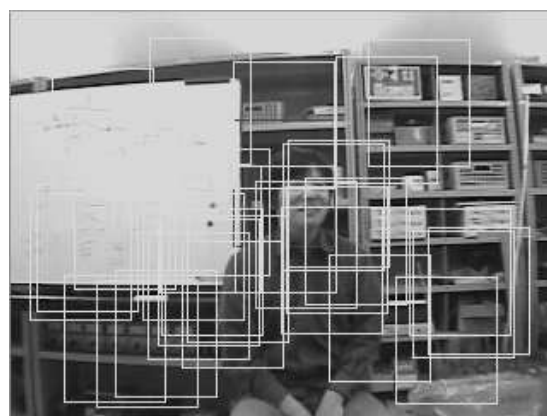


図 7: 初期状態

5.2 実験 2

カメラで撮影されている人物が画像中で左から右へと移動している様子の動画の中から顔領域の探索を行った。また実験 2 のテンプレート画像には、実験 1 で用いたものと同じ人物のテンプレート画像を使用した。実験結果を図 10、図 11 に示す。



図 8: 第 5 世代



図 10: 追跡の様子 1



図 9: 最終状態



図 11: 追跡の様子 2

5.3 考察

実験より回路が正常に動作していることが確認できた。また 10~15 世代で収束するという結果が得られた。このことから画像一枚に対して 15 世代以内で顔領域を抽出することができると考えられる。テンプレートとの比較回数は初期世代では 32 箇所、その後は一世代あたり 16 箇所行っており、15 世代で収束すると考えると 256 回程度比較していることになる。単純に入力画像の端からテンプレートマッチングを行うと 17 万回以上の比較が必要なのに比べて、本手法ではメモリアクセスを少なくし解を得ることが可能である。

6 おわりに

本稿では顔領域抽出アルゴリズムを FPGA 上でワンチップ実現し、その回路動作について検討した。今回の実験ではテンプレート画像に特定の人物の顔画像を使用した。テンプレート画像とは異なる人物を撮影すると、その人の顔領域を抽出することが困難になるのではないかと危惧される。そこで今後はテンプレート画像に平均顔の顔画像を用い、特定の人物でない場合の動作実験もする必要があると考えている。また回

路規模の縮小ならびに処理速度の向上を目指し、回路の最適化を行う。

7 参考文献

- [1] Chengjun Liu, "A Bayesian Discriminating Features Method for Face Detection", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL.25, NO.6, JUNE 2003, pp.725-740.
- [2] 山本貴彦, 佐々木悠介, 川村尚生, 菅原一孔 "顔領域追跡アルゴリズムのハードウェア実現に関する研究", 電気・情報関連学会中国支部第 54 回連合大会講演論文集"
- [3] 山本貴彦, 川村尚生, 菅原一孔 "顔領域追跡アルゴリズムのハードウェア上への実装について", 情報システム研究会資料, 2004
- [4] 長尾智晴, "進化的画像処理," 昭晃堂, 2002.