(d) 情報

動的輪郭モデルのハードウェア化と唇形状抽出への応用について

佐々木 悠介 † (鳥取大学大学院工学研究知能情報工学専攻)
山本 貴彦 † (鳥取大学大学院工学研究知能情報工学専攻)
川村 尚生 † (鳥取大学工学部知能情報工学科)
菅原 一孔 † (鳥取大学工学部知能情報工学科)
†{sasaki,tyamamot,kawamura,sugahara}@ike.tottori-u.ac.jp

1 はじめに

近年コンピュータの処理速度向上に伴い，コンピュータによる画像処理を用いた様々な試みがなされている．物体の輪郭自動抽出は，コンピュータによる画像認識処理の中でも重要な役割を果たしている．輪郭抽出アルゴリズムの代表的なものの 1 つに動的輪郭モデルがある．

本研究では動画像の処理を目的とした動的輪郭モデルを，ハードウェア記述言語 (HDL - Hardware Description Language) を用いて FPGA 上に実現することを試みる．ハードウェア化により，システムの規模を抑えられることが期待できる．

2 システム構成

図 1 に本システムのシステム構成図を示す．図に示されているよう本システムは，USB1.1 規格のカメラ，USB ホストモジュール，FPGA ボードから構成されている．FPGA ボードには 36bit 幅の 18MbitSSRAM が 2 個搭載されており，それぞれバスが独立しているので，2 個のメモリに同時アクセスが可能である．

本システムの処理の流れとしては，USB カメラより取り込まれた動画像は，USB ホストモジュールを介して FPGA ボードのメモリに保存される．保存された画像データに対し色抽出などの処理を行い，そのデータをもとに動的輪郭モデルによる輪郭抽出を行う．抽出された領域は，原画像中にその色合いを変えるなどして，モニタに表示することで，動作の確認が可能である．

FPGA では USB ホストモジュールの制御，SSRAM への画像取り込み制御，取り込んだ画像の色抽出処理，動的輪郭モデルによる輪郭抽出処理を行う．

ハードウェア化の開発手順としては，まず Innoveda 社の Visual-Elite でフローチャートや状態遷移図で回路記述を行い HDL を生成する．次に論理合成ツール Synplicity 社の Synplify Pro を用いて論理合成を行う．そして合成した論理回路を使用する FPGA デバイスに合わせてパラメータを設定する配置配線を行う．さらにデバイスフィッティングとタイミング検証を行い，最後に FPGA に PC を接続して設計した論理回路を書き込む．使用したツールは ALTERA 社の QuartusII である．

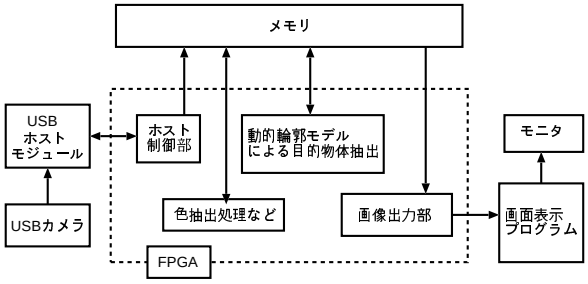

図 1: 本システムの構成図

3 動的輪郭モデル

動的輪郭モデルとは，いくつかの輪郭点で構成される閉ループである．各輪郭点には図 2 に示すような 3 つの働く力が作用し，圧力・引力の合力により目的物体に向かって収縮する．目的物体に接すると，反力により収縮を停止し，目標物体の抽出を行う．このモデルはノイズに強く，ノイズに対しつき抜けたり，すり抜けたりすることができる．以下にそれぞれの力の説明を行う．

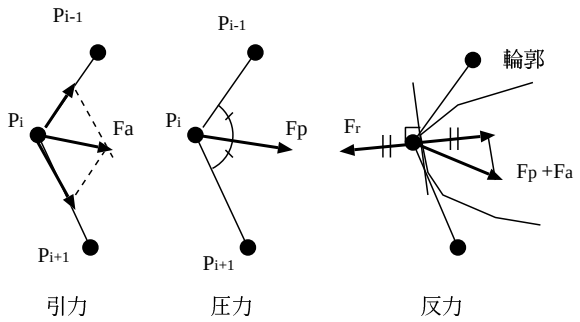

図 2: 動的輪郭モデルに働く力

引力:点 P_i とそれに隣接する点 P_{i-1} との間には，その距離に比例した力が働いていとする．同様に点 P_i と他方の隣接点 P_{i+1} にも力が働いていると考え，その合力がその点に働く引力となる．

圧力:圧力は $\angle P_{i-1}P_iP_{i+1}$ からなる角度を二等分する角度方向に働く力である．圧力が働くことによって，凹

型の目的物体をより正確に抽出することができる．

反力：引力を F_a，圧力を F_p とした場合，輪郭点はその合力 $F_a + F_p$ によって動作する．輪郭点が目的物体に接したとき，反力 F_r が $F_a + F_p$ の目的物体に対する垂直成分を打ち消すように働く．

動的輪郭モデルをハードウェア上で実時間で動作できるように構築する際，距離や角度などの複雑な計算があるため，ハードウェア量が増大し実現が困難となる．そこでハードウェア上での実現のために種々の動的輪郭モデルの簡略化を行った．簡略化を行った箇所を以下に示す．

- 動的輪郭モデルの輪郭点数を 16 点に固定する．
- 動的輪郭モデルに働く力のうち，引力と反力で収縮・停止を行う．(圧力を考慮しない)
- 反力については，輪郭エッジに接触した際に停止するか，しないかの判断のみを行う．

4 処理の流れ

動的輪郭モデル部は，座標値格納レジスタと収縮計算部からなる．座標値格納レジスタには，輪郭点の座標値が格納されている．19bit の幅を持ち，輪郭点 16 点分の x 座標と y 座標を合わせて格納するため，16 個のデータを格納する．

収縮計算部の目標物体に収縮する動作の処理の流れを示す．

1. まず，移動の対象とする輪郭点 P_i を決定し，座標値格納レジスタから P_i と隣り合う 2 点 P_{i-1} と P_{i+1} の計 3 点の座標データを読み出す．
2. 引力計算は x, y 各座標を成分分解して行った．座標点 P_i に引力 F_a が働いたときの移動先を x 成分，y 成分ごとに引力 $F_a(x)$，$F_a(y)$ を導出し，移動点 P_i' を次式により求める．なお，パラメータ k は力と距離の間の比例定数である．

$$P_i' = P_i + k\left((P_{i-1} - P_i) + (P_{i+1} - P_i)\right) \quad (1)$$

3. 移動先の輪郭点の位置に対象物体が存在するかを調べる．引力計算の計算結果により得られたデータから，対応する座標の画像データを SSRAM 部から読み出して，対象物体の存在を確認して輪郭点を更新するかどうか判断する．輪郭点 P_i が移動する新たな候補となる座標に対象物体が存在する場合，この輪郭点での処理を終了し，次の輪郭点の処理を開始する．
4. 対象物体の確認時に座標点が移動すると決定した場合，座標値格納レジスタの座標点 P_i に該当する場所に新しい座標値データを書き込み更新を行う．この処理が，輪郭点の収縮・停止に相当する．

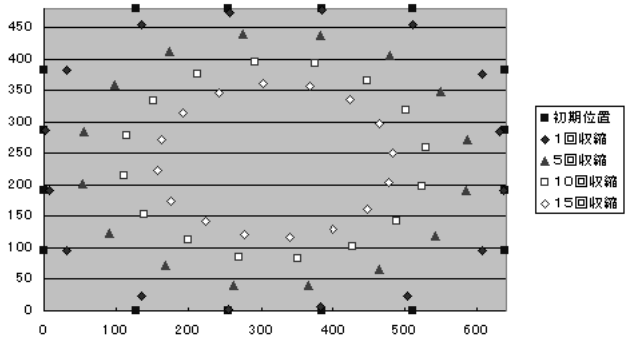

図 3: 収縮結果

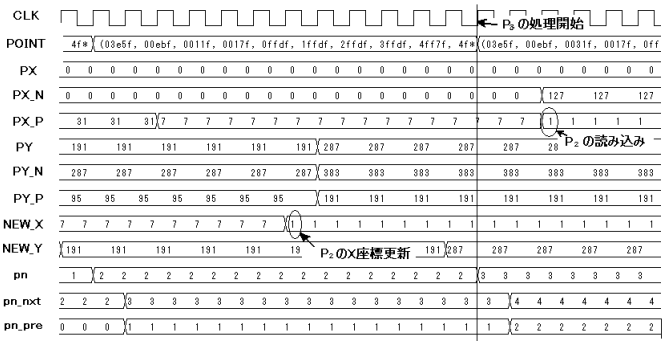

図 4: シミュレーション結果

5 動作検証

今まで述べてきたことをもとに動作検証を行った．初期輪郭は 16 点を画面の周囲を囲むように配置している．画像サイズは 640 × 480[pixel] を想定している．シミュレーションによる動的輪郭モデルの収縮結果を図 3 に示す．図 3 から画面の隅に配置した収縮点が収縮回数が増加するごとに中心に向かって収縮していくことが確認できる．

Visual-Elite で行ったシミュレーション結果を図 4 に示し，収縮動作の様子を確認する．座標値格納レジスタより，該当する 3 点の輪郭点 P, P_{i-1}, P_{i+1} を取り込む．取り込まれた座標データより収縮計算を行い，計算結果を NEW_X と NEW_Y にそれぞれ出力している．この計算結果を POINT に渡すことによって座標値格納レジスタを更新している．今回の手法では更新した座標値をただちに用いる仕様にしているため，次の 3 点の輪郭点の x 座標を PX，PX_P，PX_N(それぞれ P, P_{i-1}, P_{i+1} に対応している) に取り込んだ際に，PX_P が直前の段階で更新した x 座標の値になっていることが確認できる．

6 おわりに

本研究では，動的輪郭モデルのハードウェア実現について，システム構成を示し，簡略化した動的輪郭モデルの手法を述べた．今後の課題としては，今回導入していない動的輪郭モデルの力を加えることにより，ノイズに強いシステムの開発を目指すことを考えている．また，読唇システムの構築などへ本システムを応用することを考えている．