

アバター表情リアルタイム制御のための フェイシャルキャプチャシステム

Facial Capture System for the Real-time Control of Avatar Facial Expression

渡辺 賢悟†
Kengo Watanabe

宮岡 伸一郎‡
Shinichiro Miyaoka

1. はじめに

近年、アニメーション作品や映画作品などのコンテンツ制作において、顔の表情変化を追跡するフェイシャルキャプチャシステムが用いられるようになった。しかしながら環境整備にかかる負担が大きく、コンピュータの一般ユーザへの普及には到っていない。

一方で安価な Web カメラがコンピュータの一般ユーザに普及し、コンテンツへの応用が広がりつつあるが、カメラから得られる動画を十分に活用できているコンテンツは数が少なく、新しい応用方法の登場が期待されている。

そこで Web カメラで撮影した動画から目や口の動きを捉え、リアルタイムにアバターの表情を制御するフェイシャルキャプチャシステムに着目する。システムを一般に普及させ、アバター表情を操作する新しい入力インターフェースとして確立することを目指す。特にオンラインゲームや、ビデオチャットなどのコンテンツでの応用を想定してシステムを構築する。

渡辺ら^[1]は目や口の形状認識に動的輪郭抽出手法 Snakes^[2]を用いて高速に動作するキャプチャシステムを構築した。本研究では認識できる顔面の部位に眉毛を追加し、より多彩なアバター制御を可能にすることを目的とする。眉毛に対しても Snakes 処理を施すことで、眉の形状や位置の変化を抽出する。

本稿では、これまでに構築したシステムの流れを述べた上で、追加要素である眉毛形状の認識に必要な処理について各項で言及する。

2. システムの全体構成・環境条件

フェイシャルキャプチャシステムの全体構成を図 1 に示す。システムは、はじめに Web カメラから取得した動画から顔の位置を検出する。決定した顔領域の中でさらに目、口、眉の位置を検出して切り出す。切り出した画像に動的輪郭抽出手法 Snakes を適用して、目、口、眉の形状を判別し、その結果を用いてアバターの表情を変化させる。

以下に本システムの利用の環境条件を示す。

- ・ PC ディスプレイ 上部中央にカメラを設置
- ・ 横 320x 縦 240 ピクセルの Web カメラを使用
- ・ 蛍光灯下の室内で照明変化無し
- ・ 背景に肌色の近似色がない

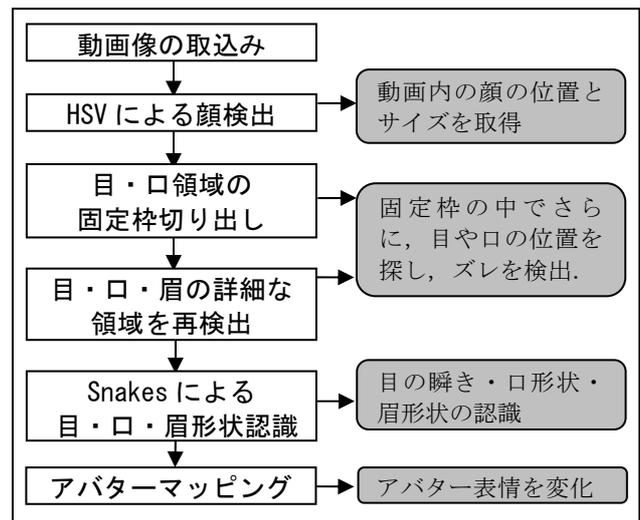


図 1. フェイシャルキャプチャシステムの処理手順

3. 顔の追跡と目・口・眉の領域検出

3.1 色相と彩度による顔領域の検出

Web カメラから取得した動画から肌色領域を抽出するため、肌色のサンプルとなる色をクリックして選択する。クリック位置の色相(H 値)、彩度(S 値)を取得し、取得した値に対して、H,S ともに ± 30 の範囲内である領域を白、範囲外を黒にするように 2 値化する。2 値化の結果画像を用いて、横と縦方向それぞれについて白領域のプロジェクションを行う。プロジェクションを行った結果得られた値が一定閾値以上になる連続領域のうち、一番大きな領域を顔の位置とする。

まず縦方向のプロジェクションの最大領域を抽出し、顔の横幅と横の位置を検出する。次に横方向のプロジェクシ

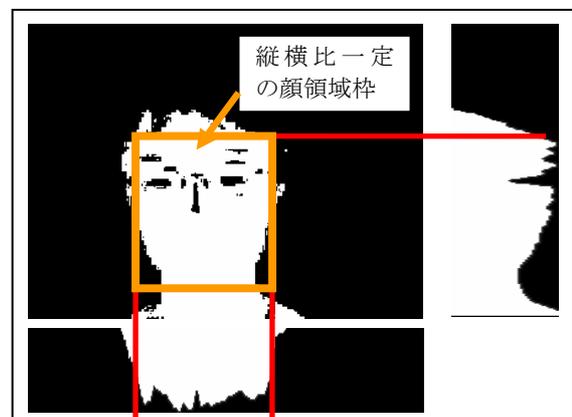


図 2. 縦横プロジェクションによる顔領域検出

† 東京工科大学片柳研究所, Katayanagi Advanced Research Laboratories, Tokyo University of Technology

‡ 東京工科大学メディア学部メディア学科, School of Media Science, Tokyo University of Technology

ジョンから顔領域の上部の位置を取得する。検出する顔領域の縦横比を一定とすることで、求めた横幅から顔領域の縦幅の大きさを求められる。得られた縦幅を検出した顔領域の上部の位置に適用することで、顔領域の位置と大きさがすべて確定する。

3.2 目・口・眉の各領域の切り出し処理

3.1 で検出した顔領域に、顔が正面向きであることを想定した目や口の固定領域(1次領域)を配置する。固定領域は左目、右目、口の3つを用意する。

設定した目の1次領域の中で輝度を用いた2値化処理を行い、3.1と同様に白領域のプロジェクション処理を行う。作成した2値化画像中には、目の他に眉や髪の毛などを含む場合がある。目は眉や髪より下に位置することから、閾値以上にあるプロジェクション領域のうち、最も下に位置する領域を詳細な目領域(2次領域)として採用する。

次に眉の2次領域を設定する。眉は純粋に目の上部に位置するものと考え、目の2次領域の上部に隣接する位置に設定する。

固定枠は顔が正面向きであることを想定しているため、正面顔の時の2次領域は1次領域内のおよそ中央に位置すると考えられる。抽出した2次領域が、1次領域中央より左右にズレを持った位置にある場合、首が横方向に振られていることになる。2次領域の中心座標と1次領域の中心座標の差分を取得し、首振りによるズレの値を求める。

次に取得した両目の2次領域がもつズレの平均値を求め、口領域のズレとして適用する。口の1次領域(固定枠)に求めたズレを適用し、口の2次領域の位置を求める。

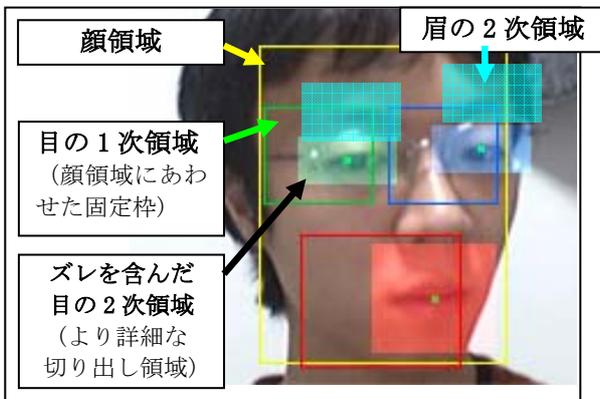


図3. 首振りによる領域のズレ

4. Snakesを用いた形状認識

4.1 2値化の調整(閾値調整・膨張収縮処理)

3.2で切り出した目と口の2次領域に輝度値による2値化処理を施す。このとき照明条件の変化に対応するため、閾値の調整を行う。

4.2以降の処理を行うのに最適な2値化モデル画像をあらかじめ用意し、モデル画像の黒領域の割合を求める。モデル画像は黒領域割合の下限と上限を設定するもの(図4)を1枚ずつ用意し、最適閾値を範囲で設定する。目領域画像を、黒領域の割合がモデル画像で指定された上下限範囲に収まるように閾値を調整しながら2値化処理を行う。両

目の閾値が決定したら、その平均値を用いて口の閾値として適用する(図5)。

眉領域についても目領域と同様にモデル画像を用意し、閾値の調整処理を行う。

システム開始後の照明条件は不変であるものとし、上記の処理を初期50フレームで施して閾値の平均値を求める。51フレーム以降は得られた平均閾値を用いて2値化を行うものとする。

作成した2値化画像はノイズを含むことが多いため、収縮・膨張処理を施してノイズを除去する。目領域で収縮2回、膨張5回を行い、口領域では収縮2回、膨張2回、そして眉領域では収縮2回、膨張2回を行う。

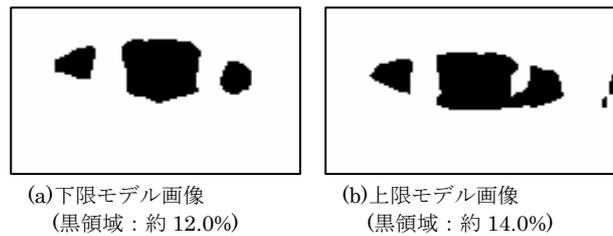


図4. 閾値決定のための左目モデル画像

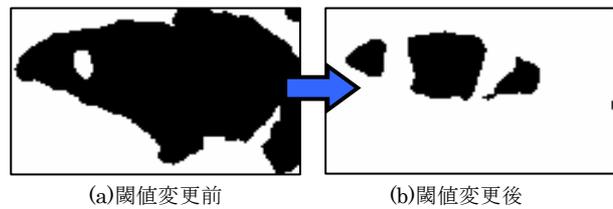


図5. 閾値の変更による2値化画像の変化

4.2 Snakesの適用

4.1で作成した2値化画像に動的輪郭抽出手法 Snakes を適用して、目、口、眉の形状を抽出する。Snakesは設定された初期位置から輪郭線が収縮を繰り返し、画像上のエッジ部分に張り付いていくことで輪郭形状を抽出する手法である。リアルタイム制御を可能にするため、処理が高速な Greedy Algorithm^[3]を採用する。

Snakesはその性質上、ノイズに弱く、制御点の初期位置に強く影響を受ける。安定した輪郭形状の抽出には適した初期位置の設定が重要である。今回のように動画に適用する場合、前回フレームの結果得られた制御点の位置を、次フレームの初期位置とするのが有効であることが示されている^[4]。本システムでも同様の手法を採用し、輪郭抽出の精度を高める。

4.3 Snakes制御点の初期位置膨張処理

Snakesには収縮する性質があるため、4.2の前回フレームの制御点の位置を用いる方法でも、次フレームで黒領域が急激に拡張した場合に対応できない(図6-b)。そこで本システムでは、黒領域が拡張した場合に制御点の初期位置を膨張させる処理を加える。

前回フレームで Snakes を処理した結果、得られた制御点の位置を次フレームの初期位置として適用する。制御点の位置が黒領域内にある場合、白領域に移るまで点を移動させる処理を繰り返す(図7)。この処理により領域が拡張した場合の輪郭抽出精度を向上させる。

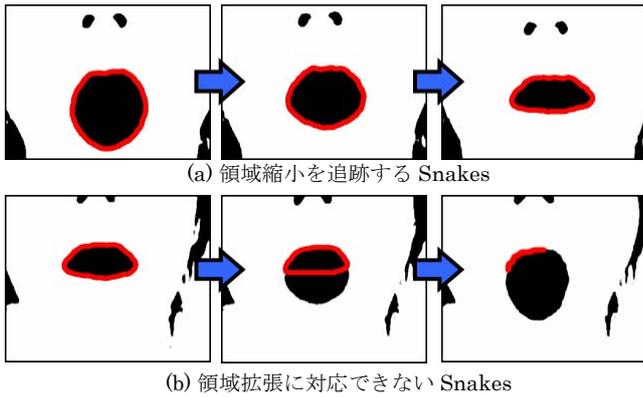


図 6. 領域の縮小・拡張時の Snakes の動作

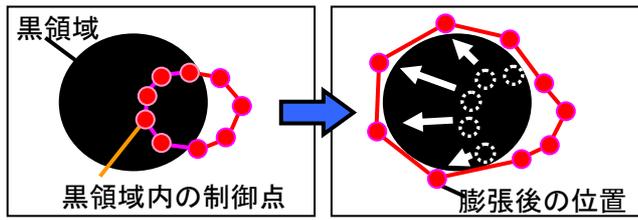


図 7. 制御点初期位置の膨張処理の例

5. アバターマッピング

4.2 で得られた Snakes 制御点の形状情報を用いて目や口の形状変化を捉え、アバターのまばたきや口の動きに適用する。

目は Snakes 制御点が形成する外接矩形の前回フレームとの高さ差分によってまばたきを認識する。

口は Snakes 制御点の外接矩形の面積とアスペクト比から「あいうえお」と「閉じ」状態の6パターンを制御する。

眉毛の変化は、眉頭に位置する Snakes 制御点の高さ情報を用いる。現フレームの眉頭の y 座標と、最新4フレームの y 座標平均値の差分が一定閾値を超えたとき、眉を上下に動作させる。アバターは通常の状態と眉毛をつりあげたパターン2種類の3パターンを用意する。

また 3.2 で得られた首振りによるズレの値を、アバターの首振りの角度として適用する。首振りのパターンは正面と左2パターン、右2パターンの合計5パターンを用意した。いくつかの例を図8に示す。



図 8. アバターマッピングのパターン例

6. 実験・評価

6.1 Snakes 制御点膨張処理の効果検証

4.3 で示した Snakes 制御点の膨張処理の効果を確認するための実験を行った。制御点の初期位置の何割かを黒領域内に配置し(図 9-a)、Snakes 処理を施した。

結果、膨張処理がない場合(図 9-b)は制御点が収縮し、黒領域の輪郭を捉えられなかった。一方で膨張処理がある場合(図 9-c)、黒領域の輪郭周辺に制御点が膨張し、より正確な輪郭形状を抽出できた。以上の結果から、制御点の膨張処理が有効であることがわかった。

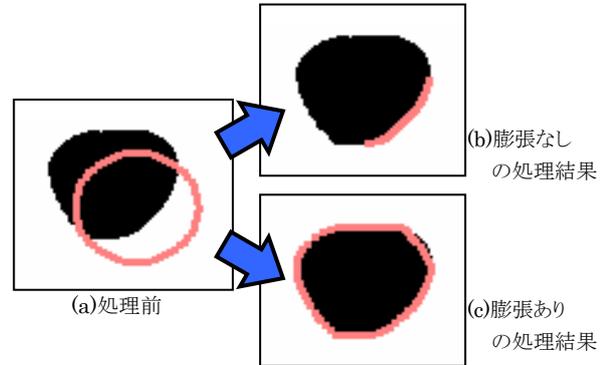


図 9. 膨張処理の有無による Snakes 処理結果

6.2 アバターマッピングの精度検証

システム全体を動作させ、5 で述べたアバターマッピングの精度を検証した(図 10)。

目の瞬きの認識には Snakes の外接矩形の高さの変化量を用いたが、およそ 80% 程度の認識精度であった。高さのみで判別しているため Snakes が縦方向のノイズを捉えた場合、連続して誤認識が発生する現象が見られた。

口形状の認識は、顔が正面向きの場合の「あいうえお」を高い精度で認識することができた。ただし歯が見える場合(「い」の形状)、歯の部分が2値化画像の黒領域にならず、Snakes 制御点が形成する形状が安定しないことがあった。また首を横に向けたとき、口によって得られる2値化画像の形状が変化するため、認識精度の低下が見られた。特に横に幅広く領域を持つ「い」を「え」と誤認識するパターンが多く見られた。このことから、現在は顔の向きにかかわらず一定であるマッピングのパラメータを、顔の向きごとに変化させる必要があることがわかった。

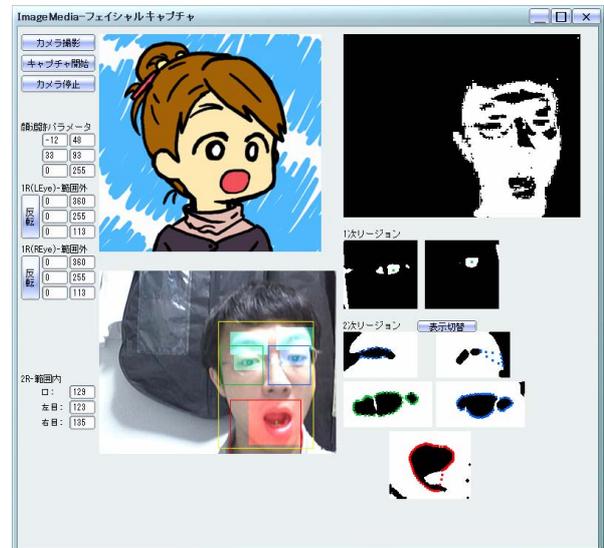


図 10. 試作システムの動作画面

眉の認識については、顔が正面向きで、髪の毛が少量である場合においては、高い精度で認識することが可能であった。ただし、顔が横向きになると顔側面にある髪の毛領域が画面に入るため、眉毛以外の形状に Snakes が追従してしまい、誤認識が増加した。

首振りの認識は、3.2 で得られる 2 次領域が正しい位置にある場合、高い精度の結果が得られた。ただし、2 次領域が誤った位置に移動した場合、大きなノイズが発生して首振りの誤認識が発生した。

6.3 使用したコンピュータ環境と速度の検証

今回試作したシステムの開発環境を表 1 に、実験に使用したマシンの性能を表 2、表 3 に示す。

表 1 システムの開発環境

開発環境	Windows XP + Visual Studio2005
使用言語	Visual C++ (Win32API + GDI)
備考	動画処理 : Direct Show Snakes 処理 : OpenCV

表 2 実験に使用したマシン 1(ノート PC)

CPU	Pentium M 1.1GHz
メモリ	DDR SDRAM 1.2GB
OS	Windows XP SP2
速度計測	10 ~ 13 fps

表 3 実験に使用したマシン 2

CPU	Xeon 3GHz
メモリ	DDR SDRAM 2GB
OS	Windows XP SP2
速度計測	20 ~ 24 fps

結果、表 2 のマシン上で 10~13fps の速度で動作することが確認できた。また、表 3 に示した高性能なマシン上では 20~24fps の計測結果が得られた。

既存研究^[1]に比べ、眉認識の追加による若干の速度低下は見られたが、最適化の余地も多く残っており、さらなる高速化が期待できる。

また表 2 のマシンに比べ 2007 年現在に販売されている一般的なノート PC は、さらに高性能である。よって構築したシステムが十分実用的な速度で実装できていることが確認できた。

7. おわりに

本研究は、動的輪郭抽出手法 Snakes を用いて、安価な Web カメラの撮影画像から顔の表情を認識し、アバターを制御するフェイシャルキャプチャシステムを提案した。一般ユーザのもつ PC の性能でも、高速に処理することが可能で、新しいコンテンツへの応用の有効性を示せた。検証実験の結果から、現状では特定の状況で誤認識率が高まるということが確認されているため、本システムを適用するコンテンツを限定し、精度向上を図る点を明確化する必要があると考えられる。

今後は顔の部位の追跡精度、形状の認識精度を向上させると同時に、本システムの成果を有効に扱えるコンテンツの試作を行っていく予定である。

参考文献

- [1] 渡辺, 岡島, 宮岡: “動的輪郭抽出手法を用いたフェイシャルキャプチャシステム”, 情報処理学会第 69 回全国大会, 第 4 分冊 2G-4, pp.59-60, 2007
- [2] M.KASS et al, “Snakes: Active Contour Models”, International Journal of Computer Vision, 321-331, 1988
- [3] D. J. Williams and M. Shah, “A Fast Algorithm for Active Contours”, IEEE TRANS, pp.592-595, 1990
- [4] 美濃 導彦, 天野 晃: “Snakes: 現在・過去・未来” 信学技報, PRMU97-184 pp.81-88, 1997