

アニメキャラクターの顔における特徴量抽出手法の提案と実装

山田太雅^{†1} 棟方渚^{†1} 小野哲雄^{†1}

近年, gimp に代表されるようなペイントソフトウェアの開発が盛んに行われている. その一方で, 作画技術の向上や, ユーザのモチベーション維持を目的とした支援システムについては, あまり研究が進められていない. そこで, ユーザの模写技術向上を目的とした支援システムに着目した. 支援システムを作成するにあたり, 必要となる特徴量の抽出方法を2つ提案する. 2つの抽出法についてそれぞれ精度実験を行い, アルゴリズムの利点や欠点を見つけることができた.

1. はじめに

近年, gimp に代表されるようなペイントソフトウェアの開発や, ユーザが描いた絵を補完するような技術の研究が盛んに行われている. その一方で, 作画技術の向上や, ユーザのモチベーション維持を目的とした支援システムについては, あまり研究が進められていない. そこで, ユーザの模写技術向上を目的とした支援システムに必要と予想される特徴量の抽出方法を提案する.



(左から)©伏見つかさ/アスキー・メディアワークス/OIP2
©Magica Quartet / Aniplex・Madoka Movie Project Rebellion

図1 アニメキャラクターの顔の例

本システムで対象とする絵の種類として「アニメキャラクターの顔」を選択した. アニメキャラクターとは, 図1に示すような, アニメ内に登場する人物キャラクターと定義する.

図1に示したアニメキャラクターの顔は現実の人間の顔と異なり, 目は大きく, 鼻は短い線や点で描かれている, 眉毛は細い線で表現されている, 顎の先端がおおよそ2本の直線で構成されているといった特徴が見られる.

2. 関連研究

河谷らは, アニメキャラクターが持つ属性(萌え因子)を定義し, 属性ごとにアニメキャラクターを分類する方法についてまとめている[7]. その研究では特徴量の抽出方法や, 分類方法について詳しく述べられている. また, アニメキャラクターを扱う有用性にも触れている.

高山ら[8]は顔が持つ色情報や対称性を利用して, アニメキャラクターの顔検出を行うシステムを構築した. 高山らは

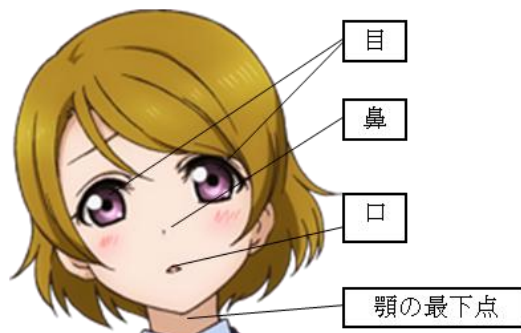
他の顔検出システムとの比較実験を行い, その結果, 提案手法のほうに有用性があることを示した.

これらの先行研究では, システムが顔の特定やパーツの抽出を行うために RGB 値の情報を用いている点で共通している. これに対し, 我々の構築したシステムでは, ユーザが鉛筆などを用いて描いた画像も取り扱うことを前提としているため, グレースケール画像でも情報を抽出できるものとなっている.

3. 特徴量の抽出

3.1 特徴量の定義

アニメキャラクターの顔を扱う際の特徴量を両目, 鼻, 口のパーツの座標と, それらの重心座標および, 顎の最下点の座標とする(図2). ここで, 顎の最下点は, 顎を構成する2本の直線の交点とする. これらは顔を構成する重要なパーツであり, その位置や大きさ形状によって顔の様相が大きく変わる. 「萌えキャラクターの描き方」[1]では, 正面を向いている顔を描く際の顔のパーツの配置について, 「目は顔全体の半分の位置に描き, 鼻は顎と目の中心に, 口は鼻と顎の中心に描くのが良い」と説明しており, これらのパーツの配置が重要であることがわかる. ただし, 眉毛は髪にかかっている場合が多いため抽出は困難と判断し, 本研究では取り扱わないことにした.



©2013 プロジェクトラブライブ!

図2 アニメキャラクターの顔の特徴量

3.2 特徴量の抽出法

特徴量を適切に抽出するためにはまず、顔を特定し、その後抽出処理をする必要がある。人間の顔のパーツには大きな個人差が見られず、左右対称性を持つ。また、肌の色には一定の傾向があることが知られている[2]。これらの特性を利用した顔認識、顔パーツ抽出法がある。同様に、アニメキャラクターの顔認識についての研究[3]もある。これらの先行研究を本研究にも応用出来ると考えられるが、どちらの研究も、RGB 値による色情報を扱っている。我々は、ユーザが鉛筆などで描いたグレースケール画像を扱っていくため、肌の色などといった、RGB 値を扱う処理はできない。以上から既存のものとは別の、RGB 値を扱わないアルゴリズムを考える必要がある。考案した方法は全部で2種類ある。それぞれの詳細を実行結果を示しながら説明していく。

3.3 特徴量抽出法 1

定義した特徴量は全て、顔の内部に存在するパーツである。これより、顔を包含する図形を生成し、その図形の内部ピクセルのみを抽出すれば、特徴量を抽出できると考えた。具体的な手順を次に示す。

- (1) ユーザに画像内にあるアニメキャラクターの顎のあたりをクリックさせる
- (2) 特徴量抽出のために以下の前処理を行う
 - グレースケール化
 - エッジ検出
 - 反転処理
 - 2 値化処理
- (3) 輪郭追跡のアルゴリズム[4]により、顔のパーツを取り囲むような輪郭のみを抽出する
- (4) (3)で生成した輪郭を取り囲むような、外接長方形を生成する
- (5) (2)の前処理によって生成した画像の内、(5)で生成した外接長方形の内側部分のみ抽出する
- (6) 輪郭追跡のアルゴリズムを適用し、輪郭のみ抽出する
- (7) (6)で生成した輪郭の内側を塗りつぶすことでマスクを生成する
- (8) (7)で生成したマスクに収縮処理を加えたあと、(2)で処理した画像から、マスク部分に属するピクセルのみを抽出する
- (9) K-means クラスタリング[5]で、「左目」、「右目」、「左目と右目以外」の3つにクラスタリングする
- (10) 「左目」または「右目」に該当したピクセルのみ残す
- (11) 外れ値を処理する
- (12) (11)で生成した画像に(9)で「左目と右目以外」にクラスタリングされたピクセルを加える
- (13) 目以外のパーツに関して K-means クラスタリングを適用して、クラスタリングする

以上で示したアルゴリズムを適用した結果を図 3 に示す。図 3 の左上の画像を入力とし、番号はアルゴリズムの手順のものと対応している。

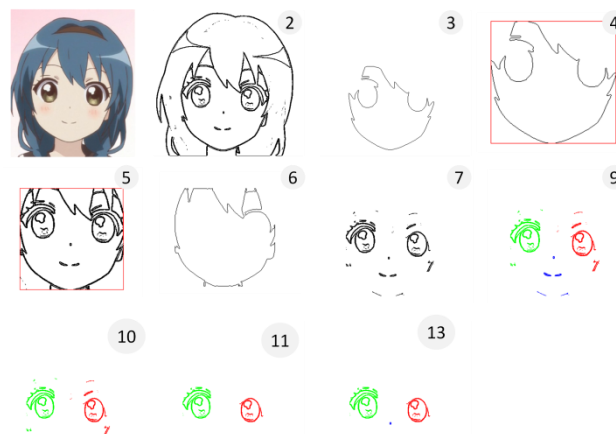


図 2 抽出法 1 の適用結果

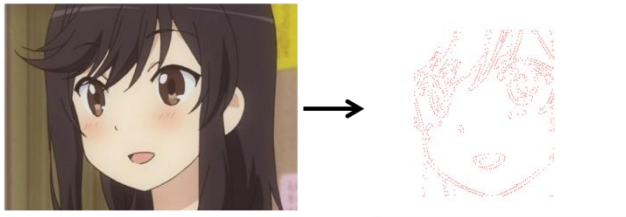
3.4 特徴量抽出法 2

特徴量を抽出する 3 つ目の方法として、機械学習を用いたアルゴリズムを説明する。分類の方法として、識別関数のひとつである、線形回帰モデル[6]を採用した。

入力ベクトルを生成するために、次の前処理を施す。手順 7 の過程までは、抽出法 1 と同じである。

- (1) ユーザに画像内にあるアニメキャラクターの顎のあたりをクリックさせる
- (2) 特徴量抽出のために以下の前処理を行う
 - グレースケール化
 - エッジ検出
 - 反転処理
 - 2 値化処理
- (3) 輪郭追跡のアルゴリズムにより、顔のパーツを取り囲むような輪郭のみを抽出する
- (4) (3)で生成した輪郭を取り囲むような、外接長方形を生成する
- (5) (2)の前処理によって生成した画像の内、(5)で生成した外接長方形の内側部分のみ抽出する
- (6) 輪郭追跡のアルゴリズムを適用し、輪郭のみ抽出する
- (7) (6)で生成した輪郭の内側を塗りつぶすことでマスクを生成する
- (8) (2)で処理した画像から、マスク部分に属するピクセルのみを抽出する
- (9) メディアンフィルタによって、黒から白のピクセルへと変化するピクセルのみ残す

以上の過程で生成された画像を図 4 に示す。



©2013 あつと・メディアファクトリー／旭丘
 分校管理組合

図 4 入力ベクトル生成のための前処理結果

この時、ある対象点 n についての入力ベクトルは、対象点 n の周囲 N ピクセルの状態である (図 5)。図 5 左の画像は前処理後のもので、中央、右の画像は入力ベクトルであり、黒丸で囲んだ点がその入力ベクトルの対象点である。この N の値は定数とする。

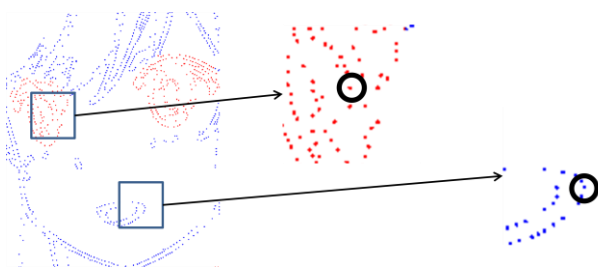


図 5 入力ベクトルの生成

図 5 を見ると、目を構成している対象点は、その周囲に多くの対象点が分布している。また、点のばらつきが大きいことが分かる。そのため、目であるかを判別する識別関数には、これらの要素を取り入れる必要がある。すなわち、点の個数に識別関数は正比例し、点のばらつきが大きいほど、識別関数も大きくなるように入力ベクトルと関数を設定することになる。

識別関数として、次を定義する。ここで、 y_n はある対象点 n の出力値、 w_i は重み、 ϕ_i は基底関数、 $\mathbf{x}_n$ は対象点 n に関する入力ベクトルである。また、 y_n が 0 以上なら、その対象点 n は目を構成する点であり、 y_n が 0 未満なら、その対象点 n は目を構成するピクセルでは無いと判定する。

$$y_n = \sum_i^M w_i \phi_i(\mathbf{x}_n)$$

$$= w_0 + w_1 N(\mathbf{x}_n) + w_2 |\mu_x(\mathbf{x}_n)| + w_3 |\mu_y(\mathbf{x}_n)|$$

$$+ w_4 \sigma_x^2(\mathbf{x}_n) + w_5 \sigma_y^2(\mathbf{x}_n) + w_6 \sigma_{xy}(\mathbf{x}_n)$$

N は入力ベクトル内にある対象点の個数、 μ_x, μ_y はそれぞれ x 軸方向の平均、 y 軸方向の平均を示す。また、 σ_x^2, σ_y^2 はそれぞれ x 軸方向の分散、 y 軸方向の分散、 σ_{xy} は x と y の共分散を示す。

新規のデータを分類する方法を下にまとめた。

- (1) 分類したい画像に対して、前述の入力ベクトル生成用前処理を施す
- (2) 各対象点に対し、入力ベクトルを生成する
- (3) 入力ベクトルを識別関数に代入して分類する
- (4) 全ての対象点の識別が終了したら、 K 近傍法[6]を用いて、対象点以外の点の分類を行う

図 6 の左の画像は、手順(1)で生成した前処理後の結果で、図 6 の右の画像は、手順(3)終了後の結果である。



図 6 分類後の結果

手順(4)で用いられる K 近傍法について説明する。 K 近傍法とは、どのクラスに属するのが未知の点について、近傍 K 個の識別後データを検索し、帰属するサンプル数が最も多いクラスに識別する方法である。これにより、前処理の段階で対象点を減らしても、問題なく全ての点について分類することが出来る。結果を図 7 に示す。左は機械学習後の結果と、未分類の点を重ねて表示してものである。右は K 近傍法を適用した後の結果である。



図 7 手順(4)の結果

以上の方法で、目を抽出することが出来る。残りの口や鼻は、抽出法 1 の手順(13)と同様に、画像を組み合わせる方法で取得することが出来る。

4. 実験

4.1 抽出法 1 の精度実験

抽出法 1 について、抽出精度を実験で確かめた。今回の実験では、特徴量となるパーツが抽出できるのかのみ判断する。従って、クラスタリング処理は行わない。

精度を判定するために正解画像を予め用意し、抽出法 1 によって出力された結果と、正解画像を比較することで精度を算出する。精度 p の定義は次の式とする。ただし、 a は正しく抽出されたピクセルの個数であり、 w は誤って抽出した、もしくは正しく抽出されなかったピクセルの個数である。

$$p = \frac{a}{a + w}$$

10 枚の画像について、抽出法 1 を適用し、精度を計った結果を図 8 に、特徴量を抽出した結果の例を図 9 にまとめた。

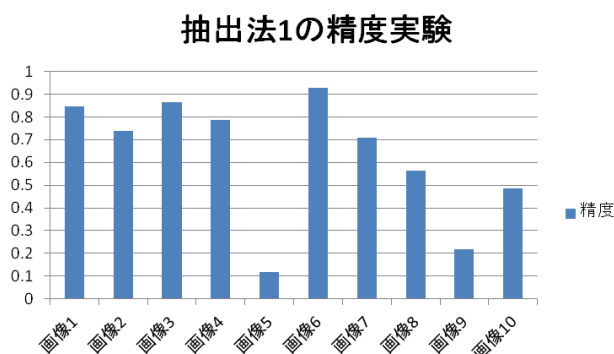


図 8 抽出法 1 の精度



図 9 抽出法 1 の結果例

画像 5 は、輪郭の抽出が適切でなかったため目を全くすることが出来なかった。原因としては、目の上部と接触していた髪の毛の色が黒であったため、目の上部のエッジが取れなかったことにある。抽出法 1 は、目の周りのエッジが検出できることを前提にしているためこのような結果になった。また、画像 9 は外周を輪郭を大きく取り過ぎた

め、余計なピクセルを取得することになった。そのため現状用いている、輪郭追跡のアルゴリズムにさらに条件を追加する必要がある。

画像 5,9 以外の画像については、マスクの収縮処理の不足により、余計なピクセルが残ってしまうパターンと、マスクの収縮処理の過多により、取得すべきピクセルを削ってしまうパターンの 2 つが見られた。この結果から、単純なマスクの収縮処理だけでは、適切にピクセルを抽出できないことがわかった。

4.2 抽出法 2 の精度実験

抽出法 2 については、精度を計る前に適切な、入力ベクトルの大きさ確かめる必要がある。そこで、適切な入力ベクトルの大きさを決定するための前実験を行った。

訓練データは図 10 の画像内に含まれる赤と青の点である。訓練データに用いる画像は 1 つのみであるが、訓練データの個数は点の個数であることに注意されたい。パラメータは入力ベクトルの大きさであり、その種類は 100, 200, ..., 800 の 8 つとした。テストデータとして 10 枚の画像を用意し、それぞれの精度を計測した。この実験における精度は、全対象点の中で正しく分類できた点の割合とする。

各画像について、精度が最大となる入力ベクトルの大きさを調べた結果、図 11 のようになった。

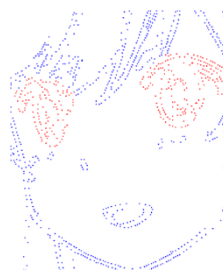


図 10 訓練データとなる点の集合

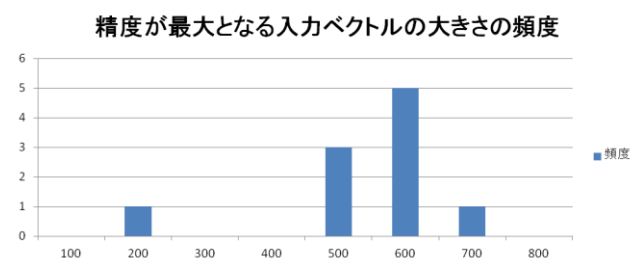


図 11 精度が最大となる入力ベクトルの大きさの頻度

10 枚中 5 枚の画像で、入力ベクトルの大きさは 600 が最適となった。また、500 から 700 の間に最適な大きさが集中している。この結果から、精度実験で扱う入力ベクトルの大きさは 600 とする。

前実験の結果を踏まえて、抽出法 2 の精度実験を行った。訓練データは前実験と同じ画像を使用し、入力ベクトルの

大きさが 600 の状態で精度実験を行った。使用するテストデータの画像は前実験のものとは異なるものにした。精度の結果を図 12 に、抽出法 2 の適用後の結果例を図 13 に示す。

大きさ600の場合の精度

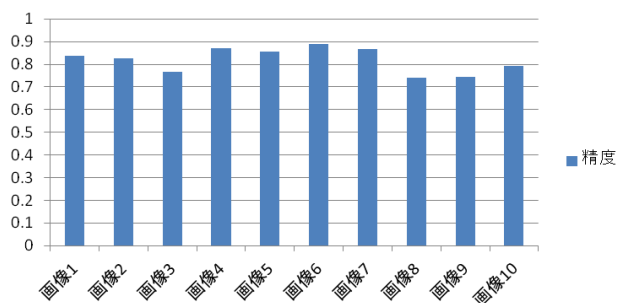
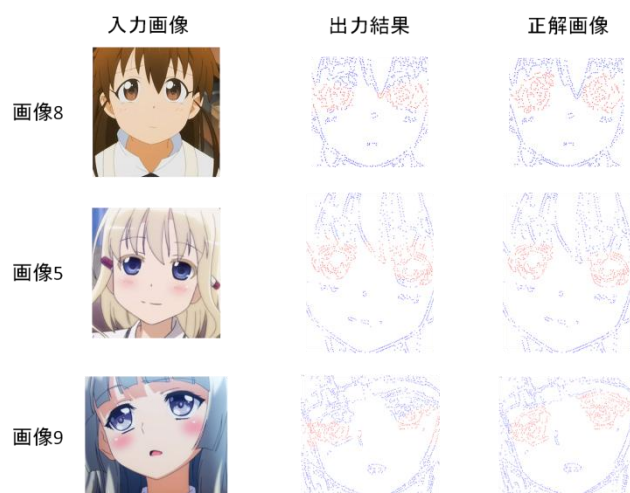


図 12 抽出法 2 の精度



(上から) ©高津カリノ/スクウェア・エニックス・「WORKING!!2」製作委員会
©小林 立/スクウェア・エニックス
©2013 春日部タケル・ユキヲ/角川書店/のうこ製作委員会

図 13 抽出法 2 の実行結果例

全ての画像において、70%の精度で分類することができた。しかし、図 13 の画像 9 のように、目を構成する半分以上のピクセルが正しく分類できなかったケースも見られた。

4.3 考察

抽出法 1 は、ほぼ正確に抽出できるパターンと全く抽出できないパターンの 2 つが見られた。一方、抽出法 2 は、目が全く抽出されないというケースは少ないが、完全に全てのピクセルを正しく分類できた画像はなかった。2 つの抽出法はどちらも、単体での使用では限界があるというのがわかる。そのため、精度の向上には 2 つの抽出法の融合が必要である。抽出法 2 は、目の一部を正しく抽出できることから、目の位置推定に役立つと思われる。そこで、抽出法 2 を応用して目の位置推定を行った後、抽出法 1 を適用するという方法が考えられる。

5. 特徴量の応用例

抽出した特徴量の応用例を図を用いて示す。

図 14 はユーザが描いた模写画像と、模写対象となる画像を比較して、ユーザにアドバイスを提示している状態である。アドバイスの生成には特徴量の位置関係を利用している。

図 15 は目や口といったパーツの位置関係やサイズをユーザに提示している状態である。このような補助機能は、ユーザのモチベーション維持を目的としている。

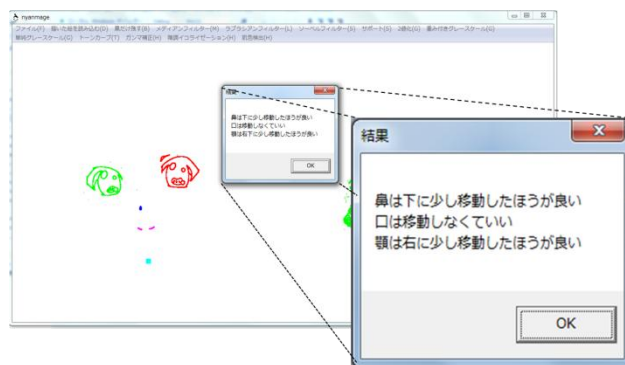
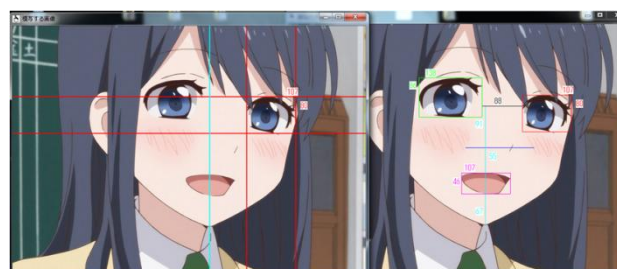


図 14 特徴量の応用例 1 (アドバイスの提示)



©宮原るり/芳文社・藤女生徒会

図 15 特徴量の応用例 2 (パーツ情報の提示)

6. おわりに

アニメキャラクターの模写を支援するシステムを構築するために必要となる特徴量の定義をし、その抽出法を 2 つ考案した。考案した抽出法について、それぞれ精度実験を行い、その結果から問題点を発見することができた。

現段階では、どちらの抽出法も画像内のアニメキャラクターが真横を向いた時の顔 (図 16) にも対応していない。



©2013 あっと・メディアファクトリー／旭丘分
校管理組合

図 16 真横を向いた顔の例

真横は正面を向いた顔と違い、目は片方しか描かれていない上、鼻は輪郭の一部として描かれているため、考案したアルゴリズムでは処理することは出来ない。そのため、真横の場合に対応した処理方法と、処理対象となるアニメキャラクターが正面を向いているか、真横を向いているかの判定処理の2つが必要になる。

今後は、抽出の精度向上を目指しつつ、ユーザの作画技術向上のための支援を構築し、ユーザによる評価実験を行っていきたいと考えている。

参考文献

- 1) 伊達達夫, 角丸つづら. "萌えキャラクターの描き方". (2010).
- 2) 岩井儀雄, 勞世二, 山口修, 平山高嗣: "画像処理による顔検出と顔認識". 情報処理学会研究報告. CVIM, コンピュータビジョンとイメージメディア, pp321-pp322, (2002).
- 3) 高山耕平, HenryJohan, 西田友是: "形状および対称性を用いたアニメキャラクターの顔検出". 情報処理学会全国大会講演論文集, pp443-pp444, (2011).
- 4) 奥富正敏, 大沢慎治, 清水雅夫, 後藤道子, 堀修, 斎藤英雄, 佐藤洋一, 千葉直樹, 江尻正員, 北本朝展, 直井聡, 二宮隆典, 藤吉弘亘, 宮原景泰, 森崎和裕, 山内結子: "デジタル画像処理". (2004).
- 5) C.M.ビショップ: "パターン認識と機械学習 下". (2007).
- 6) C.M.ビショップ: "パターン認識と機械学習 上". (2007).
- 7) 河谷大和, 柏崎礼生, 高井昌彰, 高井那美: "アニメにおける人物顔画像の萌え因子特徴評価と検索分類システムへの応用". 電子情報通信学会技術研究報告. IE, 画像工学, pp113-pp118, (2010).