

ヘアスタイル・シミュレーション・システムの試作

宮田一乗、黒川雅人

日本アイ・ビー・エム株式会社 東京基礎研究所

入力された人物画像を自動認識し、その結果を用いて、蓄積された髪画像を人物画像へ自動フィッティングさせる、ヘアスタイル・シミュレーションについて報告する。操作の対象となる画像は、すべてカラーの2次元画像であり、ビデオカメラなどの入力装置を用いて、システムに取り込まれる。取り込まれた画像から、まず色領域分割を用いて顔領域を自動抽出する。その後、抽出された顔領域から、色差とエッジ情報を用いて、目や口などの顔部品を抽出する。これらの情報を用いて、蓄積された髪画像の大きさ、傾きなどの補正を行ない、その後画像のワーピング処理を施して、顔画像に髪画像をフィッティングさせる。

A Prototype of Hairstyle Simulation System

Kazunori Miyata and Masato Kurokawa

IBM Research, Tokyo Research Laboratory

We present a prototype of hairstyle simulation system that automatically recognizes an input face image and fits a stored hair image to it. The images, which are all two-dimensional and in color, are input from a video camera. First, the face area is extracted from the face image by means of color segmentation. Next, color reference data and edge information are used to extract the the various parts of the face, such as the eyes and mouth, from the recognized face area. The target hair image is resized and rotated in accordance with these data, and is then transformed by an image-warping operation to fit the face image.

1 はじめに

最近のいわゆるマルチメディア・パソコンには、CD-ROM や音源ボードはもはや標準装備、中にはカメラが装備されているものもある。特にワープロの世界では、年賀状などに家族の写真を取り込むために、CCDカメラが装備されているものが少なくない。本システムは、このようなマルチメディアパソコンを用いて女性の「変身願望」をかなえるための一手段である。

既存のフォト・レタッチ・ソフトを用いて、自分の顔画像に希望のヘアスタイルを合成するには、まず、手作業で髪画像のスケージングや回転、平行移動の操作を行なう。その後、ペインティング・ツールで、顔や首回りの隙間を埋めたりして仕上げを行なう。この手作業にかかる時間は、1パターン当たり2、30分ほど要するものであり、また、操作の複雑さを考えた場合、万人が簡単に操作できるとは考えられない。一方、髪を3次元モデルを作成し、風や重力などの外力の作用を考慮して変形操作を行ない、最終的にレンダリングして目的の画像を得ることも可能ではあるが、データ量の点のみならず、処理時間の長さ、仕上りの点から見ても、あまり有効な解決方法とは考えられない。

本システムの目標は、以上述べた既存の手法の欠点をカバーし、ユーザに負担をかけずに、自動で顔を認識し、希望のヘアスタイルを合成することにある。

2 システム構成および入力条件

図1にシステム構成の概要を示す。処理対象画像はビデオカメラから入力され、キャプチャされたカラー画像である。光源としては、白熱電球(色温度3200K)及び昼光用ライト(5500K)を用いた。

なお撮影条件として以下の条件を与える。

- (1) 一人の人間の正面から見た上半身が写っている。
- (2) 髪が眉や目にかからないように撮影されている。
- (3) 背景は肌色とは明らかに異なる色(青や白)である。

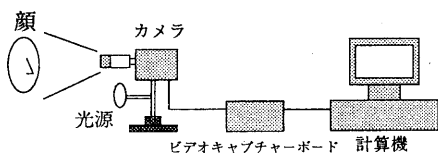


図1: システム構成

3 顔認識手法について

入力された顔画像から顔領域(首を含む)、顔部品(目、口、鼻、顎、頭頂点)を自動的に抽出する手法について述べる。これらの顔特徴は、次段の合成処理において、髪画像を補正する際に用いられる。図2に処理の流れを示す。処理内容は大きく以下の3段階に分かれる。

- (1) 顔領域の抽出処理
- (2) 色差を用いた顔部品候補の抽出
- (3) ルール集合による顔部品の決定

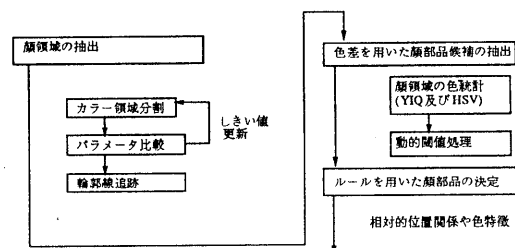


図2: 認識処理の流れ

3.1 顔領域の抽出処理

ここでは顔(首を含む)を肌色のひとかたまりの領域として抽出する処理について述べる。基本となるのは画像の領域分割処理である。ただし、領域分割処理の結果は、必ずしも求めたい顔領域と一致しない。実際に顔に対応する領域を抽出するためには領域分割の結果得られた領域を、もう一度組み合わせ、大局的に判断して顔と思われる領域を作成する必要がある。ここでは、顔を構成する領域の持つ、形状や色に条件を設定し、その条件に見合うまで領域分割を繰り返す方法を用いる。以下にその手順を述べる。

- (1) 領域分割処理(領域成長法)により初期分割を得る。
- (2) 初期分割された領域中で小領域を除き、残りの領域中で色が明らかに肌色と異なる領域を除外する。残った領域は顔を構成する領域の候補となる。それらを組合せた領域を作成し、その形状及び大きさを用い、顔領域となる条件を満足するかどうかを判定する。
- (3) 全ての組合せに関して(2)が成立しなかった場合、領域分割の閾値パラメータを変更し(1)へ戻る。

上記の処理を、顔領域と判定される領域が見つかるか、閾値変更がある回数に達するまで繰り返す。結果が得られない場合は顔領域抽出の失敗をユーザに示す。

ここで、(2)において、顔とみなす色の範囲は、光源の影響を考慮に入れ、従来の方法^{[1],[2]}に比べて広く設定している。本システムでは、色相の値が、-10度～90度のもを肌色候補としている。

また、(3)において顔領域とみなすための条件として、組み合わせられた領域の凸包を作成し、凸包の面積と実際の領域組の持つ面積比、および、凸包の左右対称性が、ある閾値以上のものを用いる。

3.2 色差を用いた顔部品候補の抽出

以上の処理で求められた顔領域の中から目、口の候補となる領域を色差を用いて抽出する。ここでは、HSV表色系における色差とエッジ密度を用いて候補領域を算出する。以下にその手順を述べる。

前節で求められた顔領域の画素に関して

(1) 領域内の画素値を HSV 表色系に変換し、その平均 (μ) と標準偏差 (σ) をとる。

(2) 各画素 (H_i, S_i, V_i) において、

$$\mu - \alpha\sigma < H_i < \mu + \alpha\sigma \text{ かつ } V_i > \beta\mu$$

の画素を肌色成分の画素としてマスクする。 (α と β はあらかじめ設定された定数)

(3) それ以外の色を持つ顔領域内にある画素を部品候補画素とし、連結成分をまとめて部品候補領域とする。

(4) 各部品候補領域内のエッジ密度を求め、エッジ密度がある閾値を越えたもののみを顔部品候補領域として次の処理に送る。ここでエッジ密度とは、原画像に LoG フィルタをかけ、エッジとみなされた画素の候補領域内における面積比である。

3.3 ルール集合を用いた顔部品の決定

以上求められた顔部品候補領域に対し、それらの相互的な配置関係から目、口の組として判断を行なう。以下で使用する記号を図3に定義する。ここでは、右目の候補領域を R_i 、左目の候補を L_i とし、口の候補を M_i とする。また各々の外接矩形についてその面積を R_a, L_a, M_a 、重心を $(R_{gx}, R_{gy}), (L_{gx}, L_{gy}), (M_{gx}, M_{gy})$ とし、右目及び左目の重心を結ぶ線分に対して、口の重心点から下ろした垂線との交点を中心点 (O) とする。

目の候補の選択

正面から見た左右の目に対応しているため、左右の目のペアとして取り出す。そのための条件として顔部品候補領域のペアをそれぞれ作成し、以下の条件に合致しているものを選択する。 (λ, γ は予め設定した定数)

(a) 外接矩形の大きさがほぼ等しい

$$1/\lambda < R_a/L_a < \lambda$$

(b) 外接矩形の形状が横長である。

$$R_h/R_v > 1 \text{ かつ } L_h/L_v > 1$$

(c) 両者の重心を結ぶ直線がほぼ水平である。

$$|R_{gy} - L_{gy}| / \sqrt{|R_{gx} - L_{gx}|^2 + |R_{gy} - L_{gy}|^2} > \gamma$$

口の候補の選択

各部品候補領域の YIQ 表色系における平均値を求め、Q 軸の値が顔領域の平均値より上のものを口候補領域とする。

上記の処理により目の候補となる領域ペア集合と口の候補となる領域集合ができる。これらの各々のペアについて、その外接矩形と重心に関して以下の条件によって、誤った組合せと考えられるものを排除する。

(1) 口は目よりも下に存在する。

(2) 口の両端点は、目の両端点の中にある。

(3) 目の面積は、口よりも小さい。

この結果複数の組が残ることがありえる (眉毛を目と間違えるケースが多い)。それらを以下の値を用いてソートする。

(a) 左右の目の外接矩形の面積比

(b) 左右の目の重心と中心点の距離の比

(a),(b) の値とも 1.0 に近いほど高い評価値を与えるような重み付けをしてソートし、上位のものを認識結果として出力する。

次に目と口の候補から顔の中心線を決定し、鼻及び顎の点を決定する。

鼻の候補 : 顔の中心線上にあり、目と口の間でエッジ強度がピークを持つ点

顎の候補 : 顔の中心線上にあり、口より下で、エッジ強度がピークを持つ点

頭頂点の候補 : 顔の中心線にあり、顔の輪郭線より上でエッジ強度がピークを持つ点

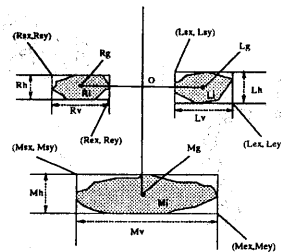


図3: 記号の定義

4 ヘアスタイルの自動合成法

4.1 髪画像の特徴データ

髪画像自体は、図4に示すように、髪領域を切り抜いて、背景を青で塗りつぶしたものを蓄えておく。なお、髪画像も正面から撮影することとする。

髪画像の特徴データは、特徴点の位置データとその属性の組合せで表現する。属性は、以下の3つのいずれかである(図4参照)。これに加えて、髪領域を切り抜く前の顔画像の特徴データ(図5)も合わせ持つこととする。なお、これらのデータは手作業で入力される。

- 属性値E: 顔領域内にかかっている点(■の点、主に頬から顎にかけての点)
- 属性値F: 顔領域内にかかってもよい点(×の点、主に額からこめかみにかけての点)
- 属性値O: 顔領域にかからない外側の点(○の点)

4.2 髪画像の補正

大きさ補正

髪画像の大きさ補正は、顔画像と髪画像の顔の幅(各々WF, WHとする)を求め、髪画像をWF/WH倍する。顔の幅は、顔領域の輪郭線上の各点からHTとJPを結ぶ直線までの距離を各々計算し、顔の左半分・右半分での最大値を求め、その値の和とする。

傾き補正

髪画像の傾き補正は、顔画像と髪画像の頭の傾き角 θ (画像に垂直な軸回りの回転角)を求め、角度の差分だけ髪画像を回転させる。頭の傾き角は、LEとREを結ぶ直線の傾きから求め、垂直方向を0とする。

位置補正

髪画像の位置補正は、顔画像と髪画像の参照点を求め、位置の差分だけ髪画像を平行移動させる。画像の参

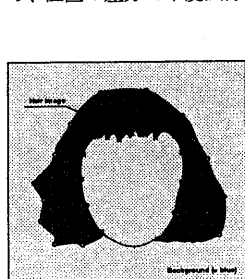


図4: 髪画像の特徴データ

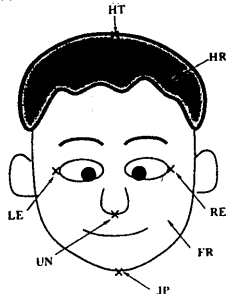


図5: 顔画像の特徴データ

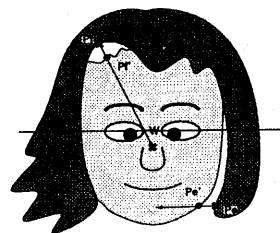


図6: 特徴点の移動

照点は、LE, RE, UNを頂点とする三角形の重心とする。

4.3 髪画像のフィッティング手法

補正を施した髪画像を、顔画像の上に単に張り付けただけでは、顔や首回りに隙間ができてたり、頬の部分に髪がかかり過ぎたりしてしまう。これらの不都合を取り除くために、以下に述べる手法で画像を変形させて、髪画像を顔画像にフィットさせる。

4.3.1 画像の変形法

画像の変形操作には、画像のワーピング操作³⁾を応用している。画像のワーピング変換は、画像上に配置された参照ベクトルの変形・移動に伴い画素を操作することにより行なわれる。

本システムでは、この変形の際に求める、画素の参照ベクトルに対する重み付け weight を、式(1)で求めている。

$$weight = (PreFixWeight / (a + D_{min}))^b \quad (1)$$

但し、a, b は定数。

ここでは、以下のように PreFixWeight を設定した。但し、HEAVY は 4.0、LIGHT は 1.0 に設定した。

- 属性E: HEAVY
- 属性F: 特徴点の移動がある場合 HEAVY、移動のない場合 LIGHT
- 属性O: LIGHT

4.3.2 特徴点の移動

髪画像の変形操作は、髪画像と顔画像の重なり具合を以下のように判定し、髪画像の特徴点を顔画像にフィットするように移動させて行なう。

- 属性Oの特徴点は、顔にかからないので判定外。

- 属性Fの特徴点 P_f に対しては、点 P_f が対象の顔画像の顔領域REの外側に位置した場合、図6に示すように、点 P_f と顔の参照点Wとを結ぶ線分と顔領域との交点 P'_f を求め、点 P_f を点 P'_f に移動させる。点 P_f が対象の顔画像の顔領域の内側に位置した場合は、移動はない。
- 属性Eの特徴点 P_e に対しては、図6に示すように、点 P_e を両目尻を結ぶ線分に平行に伸ばして顔領域との交点 P'_e を求め、点 P_e を点 P'_e に移動させる。

前節で述べた参照ベクトルは、同じ属性値を持つ特徴点を結ぶベクトルをさす。

4.3.3 参照ベクトルのアダプティブ分割

画像のワーピング変換の計算量は、画像の大きさおよび参照ベクトルの数に大きく依存している。髪を顔にフィットさせる場合、参照ベクトルの数が多いほど滑らかに合成できるが、その反面、計算量が多くなってしまふ。そこで、以下に述べるように、参照ベクトルのアダプティブ分割を行ない、この問題点を解決した。なお、以下に述べる処理は、属性Eの特徴点の参照ベクトルに対してのみ施す。

本手法では、髪画像の特徴点は、おおまかに与えておく。そして、図7に示すように、髪のフィッティング操作の際、移動させた後の特徴点を結ぶ参照ベクトルと、顔画像の重なりエラーEを計算する。ここでEは、顔領域の輪郭線と参照ベクトルとの最大距離とする。

そして、このEがある閾値より小さくなるまで、新しい特徴点を増やして参照ベクトルを2分割していく。新しい特徴点は、分割する参照ベクトルの中点に最も近い髪領域の境界線上の点をとる。

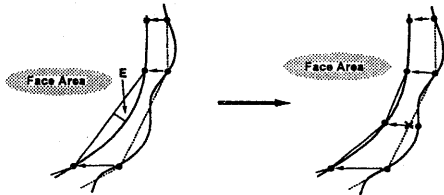


図7: アダプティブ分割

4.4 画像の合成法および結果

髪をフィットさせた仕上がりの画像の合成は、以下に示すように、重ね合わせ時のエイリアジングを除くための α バッファを用いた簡単な処理を行なう。

まず、変形した髪画像から青色画素以外の部分を髪領域として抽出する。切り抜きを行なった後の α バッファの値は、髪の部分が1(不透明)、それ以外の部分は0(透明)の値を持つ。続いて、髪領域のエッジの部分の α 値に対してスムージング・フィルタをかけ^[4]、最後に、髪画像と顔画像を α ブレンドする。

5 実験結果

認識処理の結果を図8に示す。図8(1)が入力された顔画像、図8(2)は抽出された結果である。図8(3)-(6)に大きさや照明条件の異なる認識処理例を挙げる。

合成後の結果を図9に示す。図9(1)が、入力画像、図9(2)はワーピング処理を施さない合成画像、図9(3)はワーピング処理を施した画像である。図9(4)-(6)に、さまざまな髪型を合成した例をあげる。

6 おわりに

本報告で示した手法を用いることにより、短時間で自然な仕上がりのヘアスタイルを、自動的に合成することが可能になった。

今後の課題としては、以下のものが挙げられる。

顔認識部分

影の影響に対する処理

種々の方向を向いた顔に対する対応

ヘアスタイルの合成部分

生え際の処理

顔と髪の照度のマッチング

参考文献

- [1] 佐々木, 赤松, 深町, 末長, 正面顔画像の自動識別法の検討, 信学研資, IE91-50, pp.1-7, 1991.
- [2] T.C.Chang, T.S.Huang, C.Novak, *Facial Feature Extraction from Color images*, 12'th ICPR, Vol.I, pp.39-43, 1994.
- [3] Beier, T. and Neely, S., *Feature-Based Image Metamorphosis*, Computer Graphics, Vol.26, No. 2, pp. 35-40, 1992
- [4] 安居院, 中嶋, コンピュータ画像処理, 産報出版, pp. 15-18, 1979



図 8(1): 入力画像

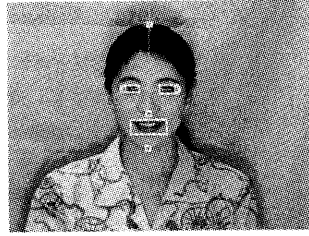


図 8(2): 認識結果 1

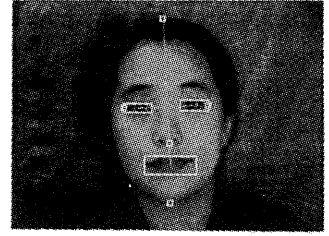


図 8(3): 認識結果 2

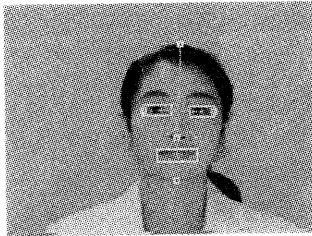


図 8(4): 認識結果 3

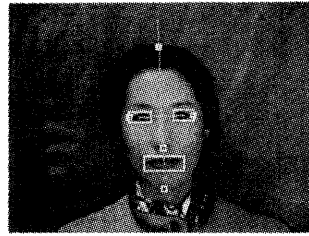


図 8(5): 認識結果 4

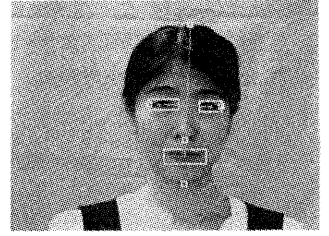


図 8(6): 認識結果 5

図 8: 認識結果

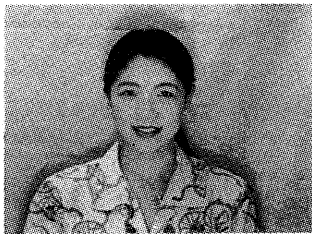


図 9(1): 入力画像

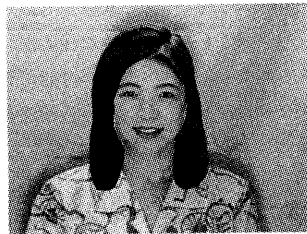


図 9(2): ワーピング処理なし

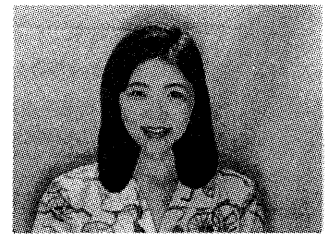


図 9(3): ワーピング処理あり



図 9(4): 合成結果 1

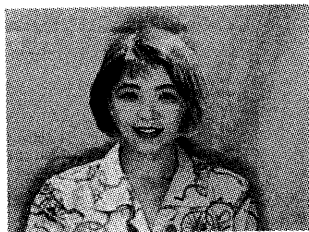


図 9(5): 合成結果 2



図 9(6): 合成結果 3

図 9: 合成結果