

新しい美人像の創出 – モルフィングの応用

高橋彩夏 (指導教員: 金子 晃)

1 はじめに

画像処理の方法として、いろいろなアプローチが提案されているが、その一つの方法として、2つの画像から中間画像を生成する、モルフィングという手法がある。

モルフィングには、各画素の色を順次線形補間することで次の画像に移行していく「クロスディゾルブ」という方法と、特徴点を対応付けることにより、画素の色だけでなく、形状全体が対応付けられた場所へと移動していく「ワーピング」という方法がある。

本研究では、ワーピングを採用し、それにおける対応付けの際、OpenCV を用いて自動的に特徴点を検出している。

2 OpenCV を用いた物体の検出

OpenCV は、コンピュータビジョンと呼ばれる、画像認識・解析に関するコンピューティング技術が主な用途として想定されている。OpenCV を用いることによって、例えば物体の認識、パターン認識、動作の認識といった、コンピュータビジョンに関する高度な画像処理機能を容易に利用できるようになる。

特定の物体を検出するには、OpenCV と一緒にインストールされる「haartraining()」「createsample()」というアプリケーションを使い、正画像と負画像を学習させることで、分類器を訓練する。

3 新しい分類器の作成

画像の収集、正解サンプルの生成、分類器の学習の手順を踏むことで新しい分類器が得られる。この得られた分類器を使用することで、OpenCV で標準ではサポートされていない物体の検出が可能になる。

(a) 画像の収集

新しい検出対象の分類器を学習するために、まず対象の映っている正画像と、背景や、異なる物体が映っている負画像を集める。まず、負画像のファイルへのパスを指定した、次のようなテキストファイルを作成する。

```
negative/image.jpg  
negative/image2.jpg  
negative/image3.jpg
```

正画像には、【ファイルのパス】【個数】【左端 X 座標】【上端 Y 座標】【対象幅】【対象高さ】を指定した次のようなテキストファイルを作成する。

```
positive/ps1.jpg 1 24 53 115 47  
positive/ps2.jpg 1 30 28 107 40  
positive/ps3.jpg 1 36 14 75 33
```

(b) 正解サンプルの生成

正解サンプルの生成には、createsample アプリケーションを使用する。このアプリケーションを次のように実行することで、収集した正画像からサンプルを生成する。

```
opencv-createsamples -info positive.txt -vec face.vec  
-num3 -bgcolor 255 -w 25 -h 25
```

この時、positive.txt は、(a) で作成した正画像の情報を示したテキストファイル、face.vec は正解サンプルのファイル名、3 はテキストファイルに書かれた正画像の個数を指す。ここで生成した正解サンプルは、分類器の学習に使用する。

(c) 新しい分類器の教育

(b) で生成した正解サンプルを用いて分類器の教育を行う。この時、haartraining アプリケーションを使用する。実行例は以下の通り。

```
opencv-haartraining -data face  
-vec face.vec -bg negative.txt  
-nstage 3 -npos 3 -nneg 3 -w 25 -h 25
```

この時、face.vec は正解サンプル、face は生成する分類器の名前 (.xml が付加される) を指す。haartraining を実行すると、段階に分けて学習が進む。段階数及び、段階ごとの認識率、誤認識率は引数で指定することができる。

学習は、以下の終了条件に当てはまった時に分類器ファイルを作成して終了する。

指定された段階数まで学習が進む。

認識率 minhitrate 以上かつ誤認識率が maxfalsealarm 以下になる。

4 顔検出

OpenCV によって以下のように顔 (及び目、口) の検出を行う。

(1) 画像を読み込む

関数 cvLoadImage() を用いて、顔を検出する対象となる画像を読み込む。

(2) ブーストされた分類器のカスケードを読み込む
学習によってあらかじめ獲得された、分類器のカスケードが記述された xml ファイルを読み込む。

(3) メモリを確保し、読み込んだ画像のグレースケール化、ヒストグラムの均一化を行う
実際の顔検出の際に利用するメモリを確保、初期化する。また、読み込まれたカラー画像をグレースケール画像に変更し、さらに、関数 cvEqualizeHist() により、そのヒストグラムを均一化する。

(4) 物体 (顔) 検出関数 cvHaarDetectObjects() により、画像から顔を検出する。

(5) 検出された全ての顔位置に、円を描画する
 検出された全ての顔に対して、サイズと中心を求め、その場所に、円を描画する。円の描画には、あらかじめ定めた8色を順次利用する。

5 検出例

検出対象が画像内に1つだけ映っている場合。顔、目、口それぞれを学習させた分類器を用いて、検出した結果は以下ようになる。



図 1: 顔検出

図 2: 目検出



図 5: 口検出

6 モルフィング

モルフィングとは、異なる入力画像 A、B への変形を滑らかに補間するよう画像の合成を行うことである。従来の画像モルフィングでは、2つの入力画像の特徴を対応付けし、線形補間していくことで、中間画像を得ていた。

本研究で用いるのは、複数の画像を合成して、中間画像を生成するという、従来の方法を拡張した、ポリモルフと呼ばれる技法で進める。この方法は、 n 個の入力画像を、それぞれ $n-1$ 次元単体の頂点、中間画像を単体内部の点と策定する。この合成において、各入力画像の混合率を制御できると考え、それゆえ、目、鼻、口、耳を、異なる4つの画像からそれぞれ取り上げ生成することも可能と考えられる。

7 数学的枠組みで見たモルフィング

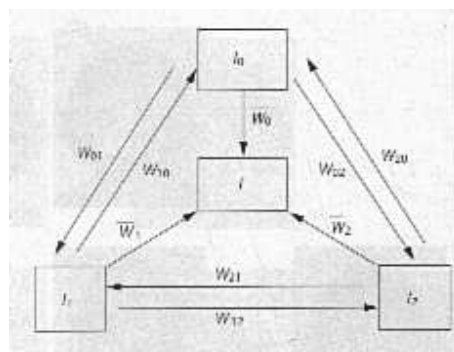
n 個の入力画像、 I_1, I_2, \dots, I_n を、 $n-1$ 次元単体の頂点と考えると、中間画像を生成する場合。従来のモルフィングの考えでは、 $(I_1 \rightarrow I_2 \rightarrow \dots \rightarrow I_n)$ のように、単体の辺全てのパスを通るような、2画像間のアニメーションの連続繰り返ししていくことを意味するが、今回のポリモルフの考え方では、単体内の任意の

パスに対応したアニメーションを生成できる。

$W_{ij} \dots I_i$ から I_j へのワープ関数

$\overline{W}_i \dots I_i$ からのワープ関数 W_{ij} を線形補間した関数

$\overline{I}_i \dots$ 中間生成画像



Polymorph:Morphing Among Multiple Images から引用。

次に、合成ベクトル b から中間画像 I を生成する手順を方程式にまとめる。

$\overline{W}_i \cdot I_i$ は、画像 I_i へのワープ関数 \overline{W}_i の適用を意味し、 p と r はそれぞれ、 I_i と I 内の点で、 $r = \overline{W}_i(p)$ の関係にある。

$$\overline{W}_i(p) = \sum_{j=1}^n b_j W_{ij}(p),$$

$$\overline{I}_i(r) = \overline{W}_i(p) \cdot b_i I_i(p),$$

$$I(r) = \sum_{i=1}^n \overline{I}_i(r).$$

8 まとめと今後の課題

特徴点を対応付けることで、モルフィングにより生成される中間画像は、歪みの少ないものになっている。しかし、既存のモルフィングは特徴付けを手動で行わなければならないため、Opencv の物体検出の機能を応用させ、自動で対応付けできるようにする。また、美人像の創出とあるように、それを目的に各パーツをそれぞれ異なる画像から取り、合成させる。

検出に利用する分類器は、対象をキャッチする意味合いにおいては十分に優れているが、対象物以外誤認識してしまう面にまだ問題があるため、今後の課題として、分類器の認識率をそのままに、誤認識を減らすことを考えている。

参考文献

- [1] Gary Bradski and Adrian Kaehler: Learning OpenCV, O'REILLY, 2008.
- [2] Seungyong Lee, George Wolberg and Sung Yong Shin: Polymorph:Morphing Among Multiple Images, IEEE Computer Graphics and Applications, 1998.
- [3] 奈良先端技術大学院制作チーム: OpenCV プログラミングブック, 2009.