

# Adaboost手法を用いた全方向変化に対応した複数顔検出

## Multiple Face Detection for Omnidirectional Rotation Using Adaboost Method

○本道 雄亮 (岩手大・院), 盧 忻 (岩手大), 西山 清 (岩手大)

○Yusuke Hondou, Xin Lu, Kiyoshi Nishiyama

岩手大学, Iwate University

キーワード : Adaboostアルゴリズム(Adaboost algorithm), 複数顔検出(multiple face detection), 全方向変化(omnidirectional rotation)

連絡先 : 〒020-8551 盛岡市上田4-3-5 岩手大学 工学部 情報システム工学科 西山研究室  
西山清, Tel.: (019)621-6475, Fax.: (019)621-6475, E-mail: nishiyama@cis.iwate-u.ac.jp

### 1. はじめに

画像から人物の顔を検出する技術は、監視システムや人間と計算機のインタフェースなど多岐にわたる用途に置いて必要となる基礎技術である。特に、照明変動やノイズなどの外乱に左右されない頑健性と、携帯電話などの計算能力が低い動作環境でも利用可能な高速性を兼ね備えた方法が求められている。

顔検出の技術は二つのカテゴリに分類できる。一つ目のカテゴリは、視覚特徴を主成分とした特徴ベースアプローチである。特徴ベースアプローチには、一定の顔パターンと候補画像の間の位置の測定値を計算する相関テンプレート法<sup>1)</sup>や、画像の異なる部分とテンプレートとマッチさせ、しきい値によって出力する可変テンプレート法<sup>2)</sup>などがある。

二つ目のカテゴリは、パターン認識理論を利用することによって、一般的な認識問題として顔検出を扱う画像ベースアプローチ<sup>3)</sup>である。特徴ベー

スの方法とは異なり、これらの技術は訓練することによって、システム内に顔知識を組み込んでいる<sup>4)</sup>。画像ベースアプローチにはサンプルから顔パターンを学習して顔検出を行う Sung-Poggio 法<sup>5)</sup>、ニューラルネットワークをベースとしたアルゴリズムで画像の真正面の顔を発見する Rowley 法<sup>6)</sup> などがある。

現在、顔検出において標準的な手法となっているものの1つに、Viola らの提案した AdaBoost ベースの手法がある。

本論文では AdaBoost アルゴリズムについて説明し、顔検出における学習アルゴリズムとして実装し、その性能について考察する。さらに、全方向変化に対応した複数顔検出を製作して評価する。

### 2. 顔の特徴

#### 2.1 積分画像法

積分画像法は画像内の任意の矩形領域内の合計明度値を高速に計算する手法である。Viola らが

提案した矩形特徴による識別法では、その特徴量を積分画像法によって計算することで高速な処理性能が実現された。積分画像法では入力された画像領域

$$I = \{\tau \mid \tau = (x, y), x = 1, \dots, l^I, y = 1, \dots, w^I\} (1)$$

の各点 $\tau = (x, y)$ に対し、

$$\psi(x, y) = \sum_{\tilde{x}=1}^x \sum_{\tilde{y}=1}^y \kappa(\tilde{x}, \tilde{y}), \quad \tau = (x, y) \in I \quad (2)$$

という値が割りふられる。これは矩形領域 $A = \{\tilde{\tau} \mid \tilde{\tau} = (\tilde{x}, \tilde{y}), \tilde{x} = 1, \dots, x, \tilde{y} = 1, \dots, y\}$  上での明度値 $\kappa(\tilde{x}, \tilde{y})$ の総和である。

全画素について $\psi(x, y)$ が計算されると、画像内の任意の矩形領域の平均明度値はその各頂点の積分画像値から計算することが可能になる。図1のように、矩形領域 $R$ の長さ $l$ と幅 $w$ とし、各頂点を $(x, y), (x + l, y), (x, y + w), (x + l, y + w)$ で表すと、この領域の合計明度値 $\phi(x, y, l, w)$ の計算式は下記のようになる。

$$\begin{aligned} \phi(x, y, l, w) = & \psi(x, y) - \psi(x + l, y) \\ & - \psi(x, y + w) + \psi(x + l, y + w) \quad (3) \end{aligned}$$

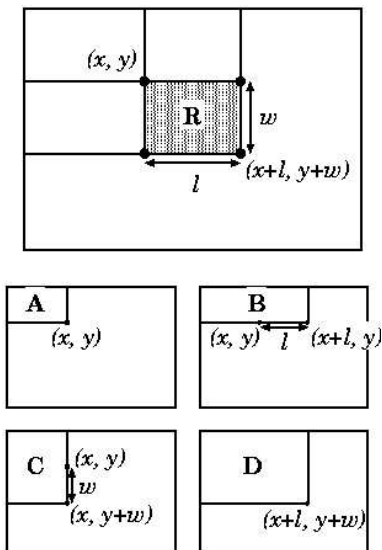


Fig. 1 矩形領域  $A, B, C, D$  の積分画像値を用いて計算した矩形領域 $R$ の合計明度値

## 2.2 矩形特徴

矩形特徴は局所特徴の一種であり、画像の中の特定領域の明るさを利用し、その画像が人の顔であるかどうかを識別する。一つの矩形特徴は互いに隣接する $n$ 個の小矩形から構成される画像が与えられると各小矩形領域ごとに合計明度値が計算され、それらの値の各領域間での差が予め学習しておいたしきい値を超えるか否かによって識別結果が決定される。

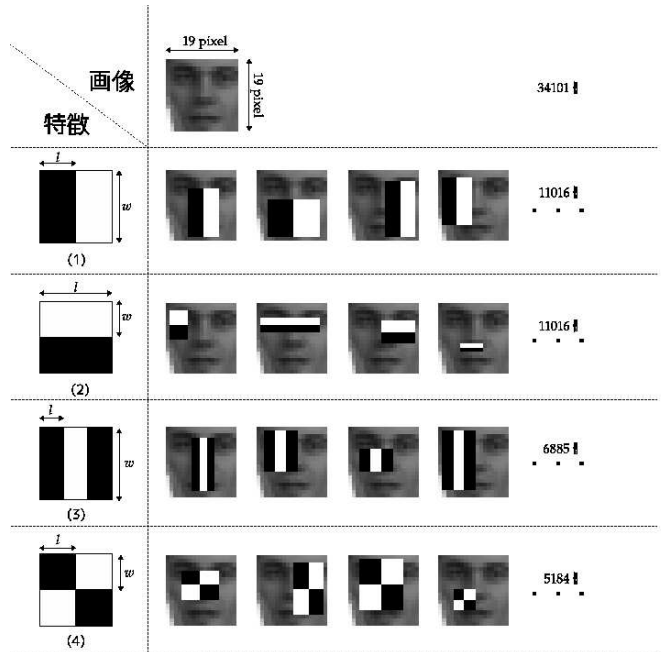


Fig. 2 矩形特徴4パターン

矩形特徴のバリエーション数が多ければ、その中に優秀な矩形特徴が含まれる可能性は高くなる。しかしそれは同時に機械学習における計算量が増加することをも意味する。ゆえにViolaらは小矩形の数および並び方を図2の4パターン( $n = 2, 2, 3, 4$ )に限定し、また1つの矩形特徴において、各小矩形のサイズは同じであるという制限を加えた。4パターンの特徴量は、

$$f^{(1)}(x, y, l, w) = \phi(x + l, y, l, w) - \phi(x, y, l, w) \quad (4)$$

$$f^{(2)}(x, y, l, w) = \phi(x, y, l, w) - \phi(x, y + w, l, w) \quad (5)$$

$$\begin{aligned} f^{(3)}(x, y, l, w) = & \phi(x + l, y, l, w) - \phi(x, y, l, w) \\ & - \phi(x + 2l, y, l, w) \quad (6) \end{aligned}$$

$$f^{(4)}(x, y, l, w) = \phi(x+l, y, l, w) + \phi(x, y+w, l, w) - \phi(x, y, l, w) - \phi(x+l, y+w, l, w) \quad (7)$$

と定義される。すなわち、図3のように $f$ は白色小矩形領域の合計明度値から黒色小矩形領域の合計明度値を引いた値である。

各矩形特徴のバリエーションは以下のように定義される。

$$F^{(1)}(I) = \{f^{(1)}(x, y, l, w) \mid x > 1, x+2l < l^I, y > 1, y+w < w^I\} \quad (8)$$

$$F^{(2)}(I) = \{f^{(2)}(x, y, l, w) \mid x > 1, x+l < l^I, y > 1, y+2w < w^I\} \quad (9)$$

$$F^{(3)}(I) = \{f^{(3)}(x, y, l, w) \mid x > 1, x+3l < l^I, y > 1, y+w < w^I\} \quad (10)$$

$$F^{(4)}(I) = \{f^{(4)}(x, y, l, w) \mid x > 1, x+2l < l^I, y > 1, y+2w < w^I\} \quad (11)$$

例えば、 $19 \times 19$ 画素の画像 $I$ の場合は矩形特徴の合計数  $T = 34101$ である。全ての矩形特徴は列ベクトル $\mathbf{f}(I)$ で表す。

$$\mathbf{f}(I) = \begin{bmatrix} \text{vec}(F^{(1)}(I)) \\ \text{vec}(F^{(2)}(I)) \\ \text{vec}(F^{(3)}(I)) \\ \text{vec}(F^{(4)}(I)) \end{bmatrix} = \begin{bmatrix} f_1(I) \\ f_2(I) \\ \vdots \\ f_T(I) \end{bmatrix} \quad (12)$$

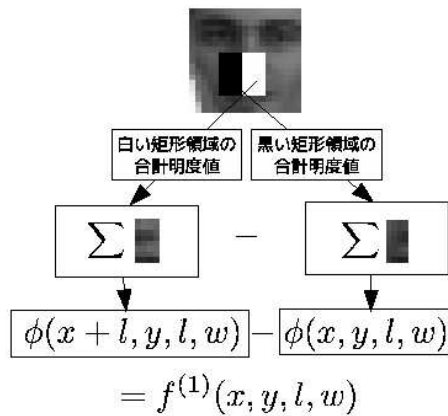


Fig. 3 矩形特徴説明

## 2.3 特徴の識別関数

ある画像の領域に存在する部分画像 $I$ が顔であるかどうかを識別する、という問題を設定する。Violaらは任意の特徴量 $f_t(I)$ による識別関数として、次のしきい値関数

$$h_t(I) = \begin{cases} 1 & (I \text{は顔}), \text{ if } p_t f_t(I) > p_t \theta_t \\ 0 & (I \text{は背景}), \text{ otherwise} \end{cases} \quad (13)$$

を用いた。ここで、 $p_t$ と $\theta_t$ は $t$ 番目の矩形特徴 $f_t(I)$ の符号としきい値である。すなわち、各小矩形の明るさの差 $f_t(I)$ が一定値 $\theta_t$ より大きい ( $p_t = 1$ のとき)あるいは小さい ( $p_t = -1$ のとき)ならば、入力画像は顔であると判断される。

比較的性能が低い識別器を複数組み合わせ、より強力な識別器を構築する学習アルゴリズムをBoostingという。ベースとなる識別器は弱識別器と呼ばれ、ここでは個々の矩形特徴が1つの弱識別器となる。弱識別器を組み合わせ、作られる最終的な識別器を強識別器と呼ぶ。

Violaらの手法では、識別器の学習にBoostingアルゴリズムの1つであるAdaBoostを用いた。AdaBoostは、その実装の容易さや高い識別精度が得られることから広く使われ、現在標準的な手法の1つとなっている。次節でその基本原理を説明する。

## 3. AdaBoost アルゴリズム

AdaBoostは”adaptive(適応的)”にサンプルの重みを更新することにより、サンプルを使いまわすことに成功したパターン識別のための学習アルゴリズムの1つである。図4にAdaBoostによって得られる識別器を示す。入力画像 $I$ に対し、その画像が属するクラスのラベル $y$ を出力する。 $T$ 個の弱識別器 $h_t(I)$ を信頼度 $\alpha_t$ で重み付けて結合することによって強識別器が構成される。

$T$ 個の弱識別器 $h_t(I)$ の重み付き多数決として、未知画像 $I$ のクラスラベル $y \in \{0, 1\}$ を求める強識別器 $H(I)$ の学習手続きについて説明する。

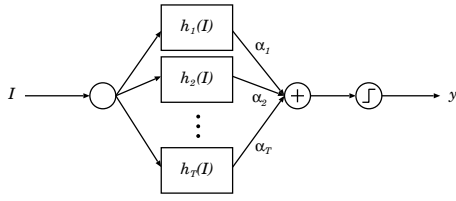


Fig. 4 AdaBoostによって得られる識別器

まず、 $N$ 個の顔画像の学習サンプルと $N$ 個の非顔画像の学習サンプル $(I_1, y_1), \dots, (I_{2N}, y_{2N})$ を用意する。 $I_i$ はサンプルであり、検出対象とそれと区別したい別の対象の画像を意味する。学習サンプルには、あらかじめクラスラベルをつけておく。さらに、各サンプルの重みを評価するための変数 $w_t(I_i)$ も用意しておく。AdaBoostは、各サンプルの重みを更新することによりサンプル分布を変更し、それぞれの分布の下で弱識別器を学習していく。

最終的に得られる強識別器は次式で与えられる。

$$H(I) = \sum_{t=1}^T \alpha_t h_t(I) \quad (14)$$

ここで、 $\alpha_t$ は $t$ 番目の弱識別器 $h_t(I)$ の信頼度である。上式のように、強識別器 $H(I)$ は弱識別器の線形結合で表される。これは、弱識別器の重みつき多数決である。

### 3.1 Adaboost アルゴリズムの手順

$N$ 個の顔画像の学習サンプルと $N$ 個の非顔画像の学習サンプル $(I_1, y_1), \dots, (I_{2N}, y_{2N})$ が与えられているとする。

1. 各サンプルの重みを $w_0(I_i) = \frac{1}{2N}$ で初期化する。
2. for  $t = 1, \dots, T$ 
  - (a)  $k$ 番目の矩形特徴( $k = 1, \dots, T$ )に対してしきい値 $\theta_k$ と符号 $p_k$ を計算し、すべての学習サンプル $I_i$ に対する誤り率

$$\epsilon_k = \sum_{i=1}^{2N} w_{t-1}(I_i) |h_k(I_i) - y_i| \quad (15)$$

$$h_k(I_i) = \begin{cases} 1, & \text{if } p_k f_k(I_i) > p_k \theta_k \\ 0, & \text{otherwise} \end{cases}$$

が最小となる $k$ を $\hat{k}$ とする。

$$\hat{k} = \arg \min_k \epsilon_k \quad (16)$$

誤り率 $\epsilon_{\hat{k}}$ 、符号 $p_{\hat{k}}$ 、しきい値 $\theta_{\hat{k}}$ を得る。

- (b) 誤り率から信頼度 $\alpha_t$ を計算する。

$$\alpha_t = \log \left( \frac{1}{\beta_t} \right), \quad \beta_t = \frac{\epsilon_{\hat{k}}}{1 - \epsilon_{\hat{k}}} \quad (17)$$

- (c) 弱識別器を学習する。

$$h_t(I) = \begin{cases} 1, & \text{if } p_{\hat{k}} f_{\hat{k}}(I) > p_{\hat{k}} \theta_{\hat{k}} \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

- (d) サンプルの重みを更新する。

$$w_t(I_i) = w_{t-1}(I_i) \beta_t^{1 - |h_t(I_i) - y_i|} \quad (19)$$

- (e) サンプルの重みの和が1になるように正規化する。

$$w_t(I_i) = \frac{w_t(I_i)}{z_t}, \quad z_t = \sum_{i=1}^{2N} w_t(I_i) \quad (20)$$

3. 最終的な識別器は、すべての弱識別器を信頼度で重み付けて多数決をとった

$$H(I) = \sum_{t=1}^T \alpha_t h_t(I) \quad (21)$$

となる。

AdaBoostの学習の目的は、 $\alpha_t$ と $h_t(I)$ を決定することである。実際には、 $h_t(I)$ が決まると $\alpha_t$ も決まる。 $h_t(I)$ が満たすべき条件は以下の3点である。

- (i)  $h_t(I)$ は、 $H(I)$ と同様に1もしくは0の2値を出力する関数とする
- (ii)  $h_t(I)$ は、誤り率が0.5より小さくなければならない
- (iii)  $h_t(I)$ は、AdaBoostによって更新されたサンプルの重み分布 $z_t$ のもとで求める

## 4. 実験

### 4.1 識別器の学習

本研究では、1つの顔の検出、複数の顔の検出、異ったサイズの顔の検出、傾きがある顔の検出の4つの実験を行うことにより、この手法の有効性と性質を検証する。

識別器の学習のために、顔と非顔のサンプル画像を収集した。実験に用いる識別器を、サンプル画像全てからランダムに選択した1,000枚ずつのサンプル画像を用いて学習した。また、選択された特徴の総数が500枚を超えた時点で学習を終了し、どの識別器も同数の特徴を用いて識別を行うようにした。

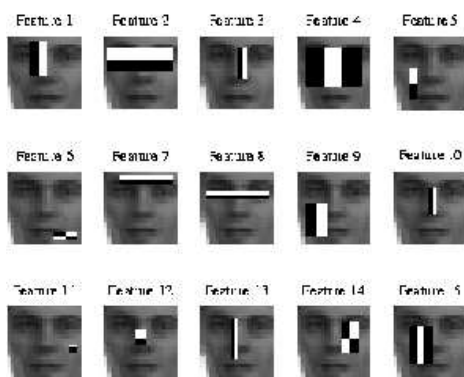


Fig. 5 学習に使用した特徴の一部

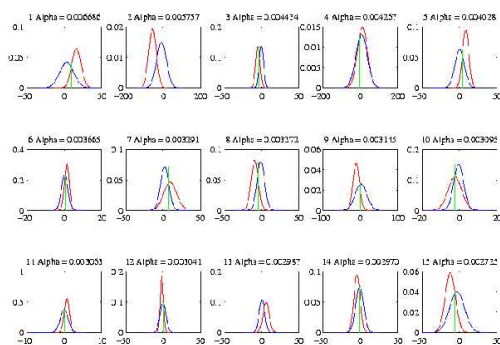


Fig. 6 学習に使用したしきい値の一部

### 4.2 顔の検出

識別器では、対象とする画像を1/1.1倍ずつ縮小してピラミッド画像

$$A^{(\gamma)} = \{ \tau \mid \tau = (x, y), x = 1, \dots, \lfloor \gamma l^A \rfloor, y = 1, \dots, \lfloor \gamma w^A \rfloor \} \quad (22)$$

を作成する。ここで、 $\gamma (= 1/1.1)$ は比例定数である。作成されたすべてのピラミッド画像上の探索領域

$$I^s = \{ \tau \mid \tau = (x, y), x = \underline{x}, \dots, \underline{x} + l^I - 1, y = \underline{y}, \dots, \underline{y} + w^I - 1, x > 1, \underline{x} + l^I - 1 < \lfloor \gamma l^I \rfloor, y > 1, \underline{y} + w^I - 1 < \lfloor \gamma w^I \rfloor \} \quad (23)$$

で顔、非顔の判定を行いながら走査を行う。

以上の条件で実際に顔検出を行った。検出する画像として、顔が1つである画像、顔が複数ある画像、サイズの異なる顔が複数ある画像、斜め向きの顔の画像に対して検出を行った。

#### 4.2.1 1つの顔の検出

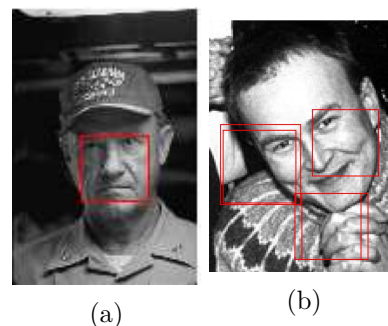


Fig. 7 検出されるべき顔が1つである画像に対する実験結果

検出されるべき顔が1つの場合は、(b)のように顔が傾いている場合には検出に失敗することがあった。

#### 4.2.2 複数の顔の検出

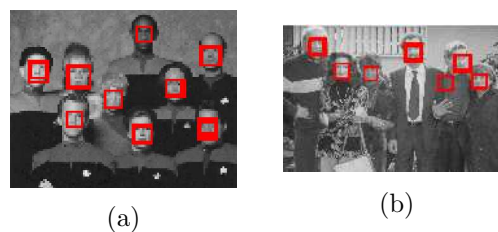


Fig. 8 検出されるべき顔が複数ある画像に対する実験結果

検出されるべき顔が複数の場合でも大体検出できているが、誤検出が見られる。(b)では誤検出が1つある。

#### 4.2.3 異ったサイズの顔の検出

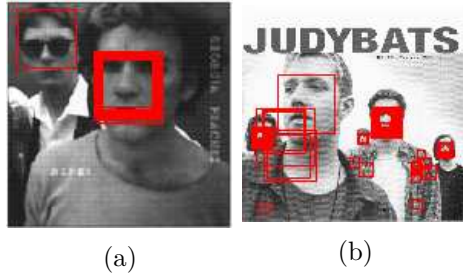


Fig. 9 検出されるべき顔のサイズが異なる画像に対する実験結果

顔のサイズの差が大きい場合には、(b)のように誤検出が多くなる場合がある。

#### 4.2.4 斜め向きの顔の検出

斜め向きの顔の検出を行った。

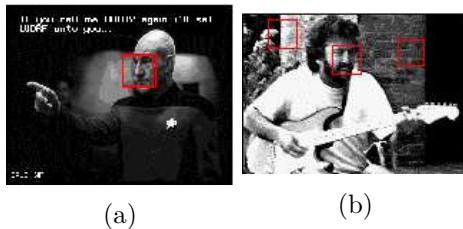


Fig. 10 検出する顔が斜め向きの画像に対する実験結果

画像中の顔が斜め向きである場合でも検出が可能であった。(b)に誤検出が2つある。

#### 4.3 傾きがある顔の検出

図7(b)のように傾きがある顔は検出に失敗することがある。そこで、傾きがある顔の検出方法を考察し、検出を行った。検出の際は、画像を縮小後にそれぞれ10度刻みで360度まで反時計回りで回転させ走査を行った。回転の式は

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x - x^c \\ y - y^c \end{bmatrix} \quad (24)$$

で表される。ここで $(x^c, y^c)$ はそれぞれ画像の中心点の座標を表している。また回転した画像は回転前の画像とサイズが等しくなるように切り取った。さらに回転したイメージの外側のピクセルの値を255(白)とした(図11 (a)→(b))。図11(b)の状態を検索した結果は図11(c)のようになる。検出結果における検出枠 $r'$ は

$$r' = \left\{ \tau'_r \mid \tau'_r = (x'_r, y'_r), \begin{cases} x'_r = x_1 \dots x_2, & \text{if } y'_r = y_1 \text{ or } y_2 \\ y'_r = y_1 \dots y_2, & \text{if } x'_r = x_1 \text{ or } x_2 \end{cases} \right\} \quad (25)$$

と表される。ここで $(x_1, y_1), (x_2, y_2)$ はそれぞれ検出枠の左上、右下の座標を表している。この座標は回転した画像に対応している。回転前の画像に対応している検出枠 $r$ は

$$r = \{ \tau_r \mid \tau_r = (x_r, y_r) \} \quad (26)$$

と表される。これは式(25)に

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}^{-1} \begin{bmatrix} x'_r \\ y'_r \end{bmatrix} + \begin{bmatrix} x^c \\ y^c \end{bmatrix} \quad (27)$$

を適用することによって求められる(図11 (c)→(d))。

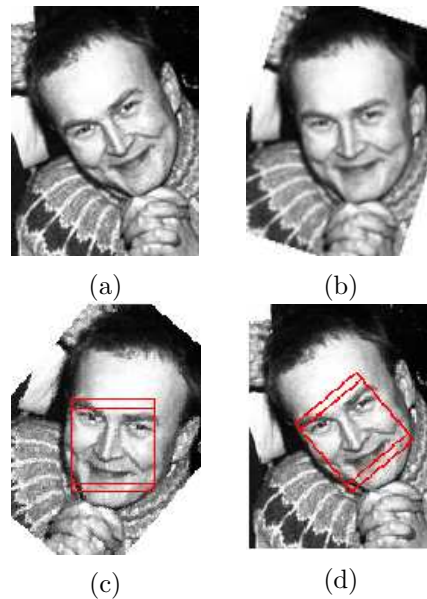


Fig. 11 回転画像の検出過程

### 4.3.1 傾きのある顔の検出

上記の方法を利用して、検出すべき顔が傾いている画像についての検出を行った。

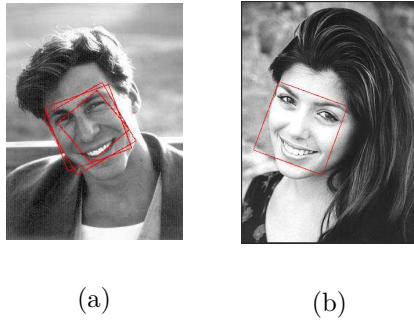


Fig. 12 検出する顔に傾きがある画像に対する実験結果

画像中の顔に傾きがある場合でも検出が可能であった(図12)。

### 4.3.2 画像の大きさによる検出結果の比較

回転する画像に対して、元画像の大きさを変化させて顔検出を行い、結果の違いを検証した。

傾きのある顔に対する検出では画像が大きいほどに検出されたとされる顔が多くなっている。ただし、大きさを3倍した画像には誤検出がいくつかみられる。

### 4.3.3 回転の刻み幅による検出結果の比較

回転する画像に対して回転の刻み幅を変化させて顔検出を行い、結果の違いを検証した。

検出の結果、角度の刻み幅が小さくなるほど検出数が増え、大きくなるほど検出数が減っている。これは刻み幅が小さくなるほど走査する回数が増えるからである。

## 4.4 複数顔検出

4.3章の方法を使用して、傾きがある顔が複数ある画像に対する顔検出を行った。



Fig. 13 画像の大きさによる検出の実験結果



Fig. 14 角度の刻み幅による検出の実験結果



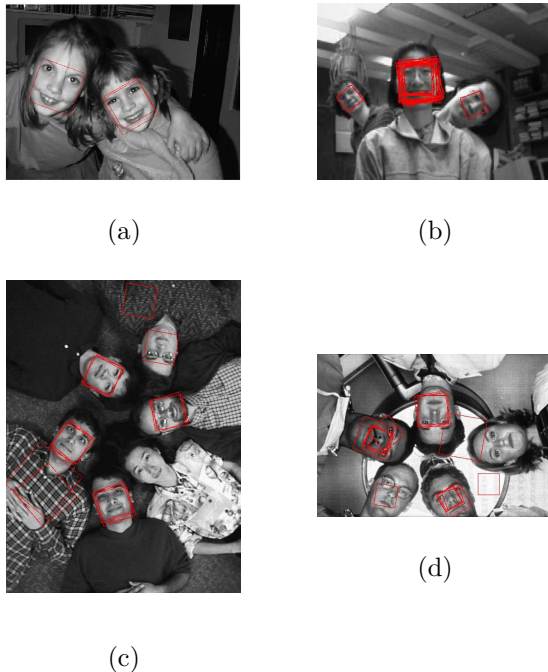


Fig. 15 複数顔検出の実験結果

画像中の顔に傾きがある場合も検出が可能であった。(c),(d)にはそれぞれ一つの未検出といくつかの誤検出がある。

## 5. まとめ

本研究では、代表的な学習アルゴリズムであるAdaBoostを実装し顔検出へ応用した。実験によりその高い識別精度を確認した。しかし、いくつかの結果の中には未検出の顔があり、それらの傾向として以下のことがあげられる。

- 検出されるべき顔に隠れや影がある場合
- 検出されるべき顔に傾きがある場合

さらに、このうち傾きがある顔について検出可能な方法を考察した。その結果、

- 傾きがある顔は画像を回転させることによって検出が可能。
- 顔の傾きが水平に近い場合には検出できない場合がある。

- 傾きがある顔は画像の大きさが大きいほど検出数が多くなる。
- 傾きがある顔は画像は回転させる角度の刻み幅が小さいほど検出数が多くなる。

ということがわかった。今後は未解決の傾向に対する検出の解決が課題となる。

## 参考文献

- 1) R. Brunelli, T. Poggio, "Face Recognition: Features versus Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 10, pp. 1043-1052, 1993.
- 2) A.L. Yuille, P.Hallinan, D.Cohen, "Feature Extraction from Faces Using Deformable Templates," *International Journal of Computer Vision*, Vol. 8, No. 2, pp. 99-111, 1992.
- 3) H. Demiral, T.J. Clarke, P.J.K. Cheung, "Adaptive Automatic Facial Feature Segmentation," *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pp. 277-182, 1996.
- 4) D. Valentin, H. Sbdi, A.J. O'Tool, G. Cottrell, "Connectionist Models of Face Processing: A Survey," *Pattern Recognition*, Vol. 27, pp. 1209-1230, 1994.
- 5) K.K. Sung, T. Poggio, "Example-Based Learning for View-Based Human Face Detection," *Technical Report A.I. 1521, Massachusetts Institute of Technology*, 1994.
- 6) H.A. Rowley, S. Baluja, T. Kanade, "Human Face Detection in Visual Scenes," *Technical Report CMU-CS-95-158, School of Computer Science, Carnegie Mellon University*, 1995.