

## NF-Features による画像間の頬点の対応付け

圓城達也<sup>†1</sup> 吳 海元<sup>†1</sup>

本論文では異なる2つの視点から撮影された顔写真からその人の3次元顔モデルを自動的に生成する方法について述べる。顔パーツのコーナ点とはFAST(Features from Accelerated Segment Test)で検出でき、その特徴量はSIFT(Scale Invariant Feature Transform)で求めることができることが一般的に知られている。頬の濃淡変化が少ないので、従来手法で頬の特徴点の検出と対応付けを困難である。本論文では、頬の点とはNF-Featuresというアルゴリズム[7]に基づいて推定し、特徴量を求める方法について述べる。

### Corresponding of cheek points between two images by NF-Features

TATSUYA ENJO<sup>†1</sup> HAIYUAN WU<sup>†1</sup>

This paper describes a method for generating a three-dimensional face model automatically using two images taken from two different viewpoints. It is well known that corner points of a face can be detected by using the FAST (Features from Accelerated Segment Test), and its feature can be described by using the SIFT (Scale Invariant Feature Transform). Because cheeks in an input image have few light and shade changes, it is difficult to detect and correspondence the points of cheeks with conventional techniques. In this paper, we propose a method for detecting and describing the cheek points based on NF-Features algorithm [7].

## 1. はじめに

### 1.1 研究目的と背景

映画やゲームなどの映像製作においては、人間の顔を表現した作品が数多く作られており、3次元顔モデルの需要も高まってきている。もしデジカメで異なる2つの視点から撮影されたある人の顔写真からその人の個人3次元顔モデルを自動的に・簡単に生成することができれば、ユーザが自分で自由に表情・姿勢変化、顔パーツの位置・大きさ調整などの応用が可能になる。また、各個人の顔のデータベース化が実現でき、犯罪者データベースの3次元化、顔の骨格に関する種々の研究、アニメーションの作成など数多くの応用分野への活用が期待できる。

現状では、3次元顔モデルを作るために、2枚画像間の顔特徴点の対応付けは手作業で行い、手間がかかる問題が残っている。また、より正確な顔表面を作るため、顔パーツの特徴点だけではなく、頬あたりにも対応付けた点ができるだけ多く取れる方が望ましい。この作業自身が手作業でも容易ではない。



図1 フォトカルクツビューで顔のポリゴンモデルを作成するための画像セット例  
(黄色の点群は手動で指定されている)

図1に示すような特徴点を自動的に取得できるようにすることが本研究の目的と位置付ける。この目標を達成するため、以下の2種類の特徴点を自動的に抽出することが必要となる

- 1) 顔パーツ(目、鼻、口)の両端の正確なコーナ特徴点 (図8(a)の青点に参照)
- 2) 頬の特徴点 (図8(a)の赤点に参照)

顔パーツのコーナ点とはFAST (Features from Accelerated Segment Test) [1, 5]検出でき、その特徴量はSIFT(Scale Invariant Feature Transform) [6]で求めることができることが一般的に知られている。本論文では、濃淡変化の少ない頬の点とはNF-Features[7]というアルゴリズムに基づいて推定し、特徴量を求める自動抽出と対応付け方法を提案する。

### 1.2 関連研究

顔特徴点や特定部位の抽出や追跡に関する研究は、従来から活発に発表されている。

東芝研究開発の湯浅ら[2]は複数個の顔特徴点を自動的に検出する方法を提案している。湯浅らはあらかじめ定めた14個の顔特徴点について、それぞれの特徴点の性質に基づいた特徴点候補検出、パターンによる検証、近傍特徴点に対する幾何学的な位置条件を利用して特徴点の候補を検出する。さらに、検出された特徴点候補全体に対する幾何学的な整合性は3D標準顔形状モデルを利用して判定することにより、顔らしい配置のものを選択する処理を行なっている。

法政大学の小谷ら[4]は、シーン中から顔を検出する方法として、まず、カラー画像の領域分割処理によって顔面内部に位置する左右の目と唇の候補を多数抽出した後、それら3つの特徴点を基準としたアフィン変換で切り出される

矩形領域の濃淡画像が顔パターンとみなしうるか否かを部分空間類別法を用いて判定する方法をとりあげている. カラー画像処理については画像処理ソフトHALCON を使用して入力画像から顔を自動的に検出するシステムを構築し, さまざまな画像入力条件のもとでのそのシステムによる顔検出の安定性を評価している.

Wuら[3]は眼鏡の有無と顔特徴点の自動検出方法を提案している. 彼らはほぼ正面を向いている顔画像から, 色情報(肌色と髪色)を抽出し, エッジ情報(強度と方向)を抽出する. この情報より, 顔と顔器官を含む領域を決定する. 頬部分の肌らしさの平均値と目尻周辺のそれと比, 頬部分のエッジ強度の平均値と目尻周辺のそれとの比から, 眼鏡の有無を検出する. 検出された顔器官を含む領域から, 口, 鼻, 目の位置を予測し, エッジ情報より確定する. 最後に, SUSAN オペレータを用いて顔特徴点を抽出している.

## 2. 関連アルゴリズム

### 2.1 FAST コーナー検出アルゴリズム

FAST (Features from Accelerated Segment Test) はS.M. Smithらによって開発されたSUSAN (Smallest Univalued Segment Assimilating Nucleus) というアルゴリズムに基づいて開発されたコーナー検出アルゴリズムである.

SUSANコーナー検出法の原理のイメージは図2に示し, 計算式が以下になる.

$$c(i, p) = \begin{cases} 1 & \text{if } |I(i) - I(p)| \leq t \\ 0 & \text{if } |I(i) - I(p)| > t \end{cases}$$

$$n(p) = \sum_{i=1}^N c(i, p)$$

すなわち,  $C(i, p)$ は画像内の画素( $p$ , +で表す)を中心とし, 半径 $r$ とした円内の画素 $i$ ( $N$ 個)と $P$ の明度の差を比較し, 差が閾値 $t$ より小さい場合は1とし, 閾値より大きい場合は0とする.  $n(P)$ が円内の $C(i, p)$ の合計であり,  $3N/4$ より少ない場合, コーナーとして検出する. また,  $n(P)=N/2$ の場合, エッジとして検出できる.

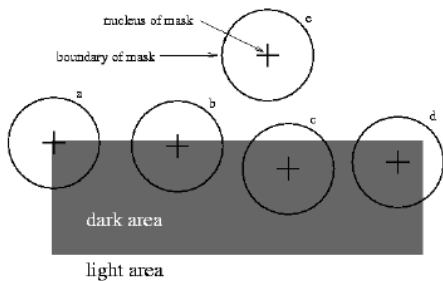


図2 SUSAN コーナー検出アルゴリズムのイメージ

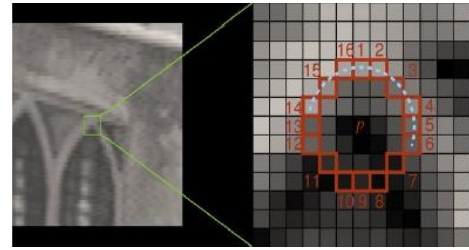


図3 FAST コーナー検出アルゴリズムのイメージ

E. Rosten らはSUSANの高速化を行い, FAST を提案している. FAST の基本原理は図3に示す. 画像内の画素( $P$ )を中心とし, 半径 $r$ とした円周上の画素( $N$ 個)と $P$ の明度の差を比較し, 差が閾値より大きい場合は+1, 合計数が $3N/4$ より多い場合, コーナーとして検出する. さらにスピードアップするために, 図3の右側のように, 1(上)と9(下)の比較, 1(上)と9(下)と5(右)の比較, 1(上)と9(下)と15(左)の比較で判定処理を行う. よって, SUSANより, FAST は画素間の比較数を減らしていることが分かる. 図4にFAST コーナー検出アルゴリズムで検出された画像例を示す.



図4 FAST で検出されたコーナー点の画像例

### 2.2 NF アルゴリズム

NF-Features (No-Features Features)は, Ralf Dragon らによって開発された特徴量記述, 対応付けのアルゴリズムである. 濃淡変化の少ない領域において, 一般的なアルゴリズムでは特徴量記述を行わず, 画像間での点での対応付けは難しい. NF-Features のアルゴリズムでは, 一般的な手法で濃淡変化のあるところから検出された特徴点の情報に基づいて, ボロノイ図とドロネー三角形分割により, 抽出した点をNFの候補点(以降, NF点と呼ぶ)とする. NF点の特徴量記述はまず, 図5のようにNFとして検出された1つの特徴点(青点= $I(0)$ )からその周り $360^\circ$ を $45^\circ$ ずつ等しく8つに分けた線上距離 $L$ の点( $I(1)$ とする)を結ぶ.

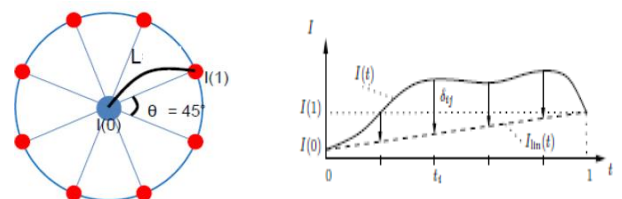


図5 NF の特徴量記述におけるイメージ

I(0)から指定した点(I(1))までを 0 から 1 までのパラメータとし、そのパラメータとし、そのパラメータ間で N 個のサンプル点にてディスクリプタ  $\delta_{ij}$  を B、G、R それぞれを以下の式により算出する。

$$t_i = \frac{i}{N+1} \quad (1)$$

$$\begin{aligned} \delta_{ij} &= I(t_i) - \text{lin}(t_i) \\ &= I(t_i) + (t_i - 1) \cdot I_0 - t_i \cdot I(1) \end{aligned} \quad (2)$$

算出した  $\delta_{ij}$  は、式 (3) で正規化し、その値は  $d_{ij}$  となる。

$$d_{ij} = \frac{\delta_{ij} - \text{mean}\delta}{\sqrt{\text{var}\delta}} \quad (3)$$

またこの処理は 2 画像に存在する全ての NF 点で行うが一つの画像を A 画像、もう一つの画像を B 画像とすると B 画像での NF 点は A 画像での NF 点を幾何学の制約を設けてホモグラフィ行列によって B 画像に射影することで点の位置推定を行う。

各ボロノイ点対の 32 次元の差を B,G,R ごとにそれぞれ式 (4) のように算出し、その差を足して平均値を算出する。

$$e_{ij} = d_{Aij} - d_{Bij} \quad (4)$$

32 次元の差の平均値のうち、1 次元以上で閾値を超える場合は誤対応点対とみなし、逆に 32 次元全てで閾値を下回る場合は正対応点対とみなす。閾値は以下の式のように算出する。

$$b = 2(1-p) + 2p \left( \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right) rN \quad (5)$$

$$rN = \frac{5N+4}{6N+6} \quad (6)$$

$$\frac{1}{\sigma^2} = \frac{\sigma^2}{s1p \cdot \text{var}\delta} \quad (7)$$

NFアルゴリズムで対応付けを行った結果例は図6に示す。

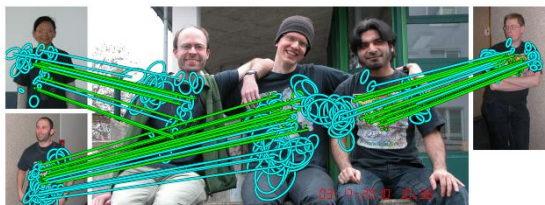


図 6 NFによるマッチング例

### 3. 顔パーツ特徴点の抽出法

#### 3.1 部分領域の明度正規化による FAST の安定化

FAST は円形領域の中心画素の明度と円周上の画素の明度間の差によって中心画素がコーナーか否かを判定する手法である。画素単位でコーナー特徴点を正確に検出できる

メリットがあるので、正確な 3 次元顔モデルを作成するために顔パーツ特徴点の検出方法として採用する。一方、明暗の差を評価する閾値が入力画像の撮影状況によって調整する必要があり、正しく設定できない場合、期待通りの顔パーツ特徴点ができない可能性がある。

本論文では、FAST で顔パーツのコーナー検出処理の安定化と汎用化を図るために、FAST の閾値を入力画像によらず最適な固定値で決める必要があると考え、OpenCV のライブラリで検出される顔矩形内で動的に局所領域の明度の正規化を行う工夫を加える。

予備実験で、顔矩形を 3×3 の 9 ブロック、4×4 の 16 ブロックの各ブロック内で明度の正規化を行ったあと、FAST で顔パーツのコーナー特徴点の検出を行った。図 7 には、検出された顔領域内にブロック分割無し、9 ブロック分割、16 ブロック分割で明るさの正規化を行い、全て同じ閾値を用いて FAST で検出された特徴点の画像例を示す。



図 7 FASTによるブロック分割の処理結果

FAST による顔パーツのコーナー特徴点の検出率を評価する基準として、3 次元顔モデルの作成時重要な、眉毛の両端、目の両端、鼻の穴の下、唇の両端の計 12 個のコーナーパーツ特徴点を定める (図 8(a)の青点)。図 7 に示す 3 種類の方法で検出されたコーナーパーツ点検出数比較グラフは図 8 (b) に示す。図 7 と図 8 から 16 ブロック分割の場合最も良いパフォーマンスが得られることを確認できている。本提案手法では、一番よい結果が得られた 16 ブロック分割の方法を採用する。

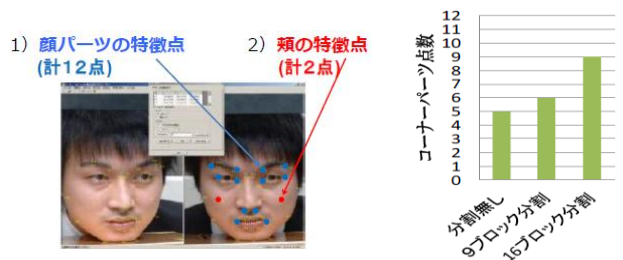


図 8 顔の特徴点の定義と顔コーナーパーツ点比較グラフ

提案手法では、さらに処理の安定化を図るために、この 16 ブロック分割の配置はオーバーラップのある動的な配置法を採用する。具体的に、図 9 に示すイメージ通りに、検出された顔領域 (青線の矩形) によって、16 ブロック矩形サイズの縦横の長さ (N) を求め、顔領域内を縦横ともに

N/2 ずつ計 81 個の矩形をかぶせながらその領域内部（青色の正方形）でノイズを除去しながら明度の線形変換を行った後で FAST 処理を行う。

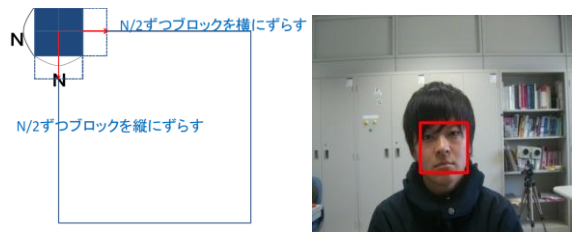
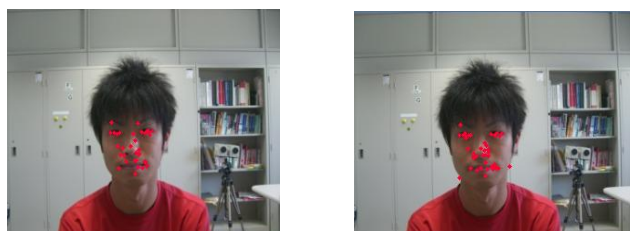


図 9 ブロックを動的に FAST 処理するイメージ



(a) 動的ブロック分割処理 閾値 80 (b) 固定ブロック分割処理 閾値 70

図 10 FAST の動的ブロック分割処理結果の比較

図10(a)に動的に16 ブロック分割で明度正規化をしながらFAST でコーナー検出の結果を示す。図10(b)に固定の16 ブロック分割で明度正規化をしながらFAST でコーナー検出の結果を示す。一般的に、FAST 処理に用いられる明度の閾値は大きく(厳しく)すると抽出できる特徴点の数は少なくなるが、図10 (a)では閾値を80 とし、図10 (b)では閾値を70 としたFAST 処理をしているものの、どちらもほぼ同数の特徴点が検出できている。提案手法の方がより顔パーツのコーナー特徴点を検出できている。

### 3.2 AAM を用いた最近傍探索

3.1 での結果には顔パーツ以外の特徴点が多く含まれており、Bottom Up だけの画像間の対応付けを行う際に誤対応が多くなってしまいう問題が発生する。

AAM(Active Appearance Models)は、画像上で平均的な形状から様々に変形する物体（例えば顔）に関して、その形状と内部の明度分布を同時に低次元で表現することのできる統計モデルである。図 11 に示すように、繰り返し学習処理によって、各個人の AAM を求めることができ、顔パーツの大きな位置情報を得られる。

本論文では、図 12 に示すように、AAM でガイド (Top Down) した最近傍探索で対応付けの精度の向上を試みる。

正面顔、回転顔で FAST 特徴点をそれぞれ検出し、データ点を FAST 特徴点、クエリ点を真値で定義した顔コーナーパーツ点 12 点で定めて、最近傍探索を試みた結果を以下の図 13 に示す。



人物 A(メガネ装着時)



人物 A



人物 B



人物 C

図 11 各個人ごとの AAM 実装結果

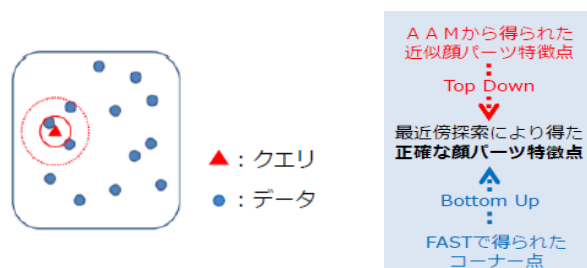
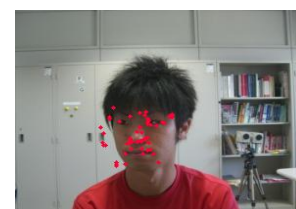


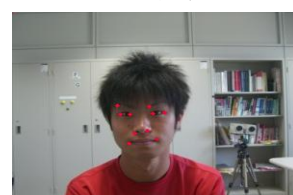
図 12 AAM と FAST を用いた最近傍探索のイメージ



(a1)正面顔の FAST 処理結果



(b1)回転顔の FAST 処理結果



(a2)(a1)の最近傍探索 処理結果



(b2)(b1)の最近傍探索 処理結果

図 13 最近傍探索による特徴点の絞込み結果

### 3.3 頬の点の抽出

頬の候補点は、2.2 で上述したように一般的な特徴点を顔コーナーパーツ候補点の FAST 点をボロノイ点としてボロノイ処理を行う。それにより抽出されたボロノイ母点を NF 点、つまり頬の候補点となる。図 14 に FAST 点(赤点)

とボロノイ点(緑点)の抽出結果を示す。

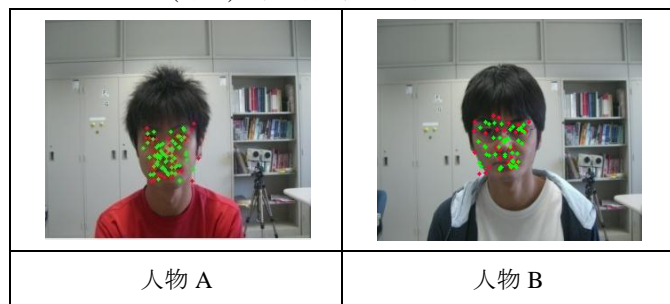


図 14 頬の点の抽出

#### 4. 顔画像間の特徴点の対応付け実験

提案手法の有効性と汎用性を確認するため、複数人の異なる姿勢の顔画像を撮影し、異なるツービューの顔画像を用いて、顔画像間のキーポイント特徴点の対応付けの実験を行った。

提案手法の顔パーツのコーナー検出に特化している FAST で検出された顔コーナーパーツ点周辺の特徴量を SIFT の特徴量記述のアルゴリズムを用いて記述し、また頬の点周辺に対しては NF の特徴量記述のアルゴリズムを用いて、それぞれ画像間の対応付けを行った。

まず、頬の候補点の対応付けは、顔パーツ左目の両端 2 点、鼻の左穴の下 1 点、そして口の左端 1 点の計 4 点を同一平面上に存在すると以下の図 15 のように仮定し、正面顔(A)と回転顔(B)の両画像から定める。これらの点の x,y 座標の 2 次元情報からホモグラフィ行列を算出する。そして正面顔で得られた頬の候補点(ボロノイ点)全点を、この行列を用いてホモグラフィ変換を行い、射影させることで回転顔の頬の候補点の位置推定を行う。その後で頬候補点全てで NF の特徴量記述を行い正面顔と回転顔で 1 対 1 の点对で対応付けを行った。図 16 で L の値が異なる実験結果をそれぞれ示している。

次に SIFT を用いて顔コーナーパーツ点での対応付けを行った。FAST の中の明度の閾値については全て同じパラメータで実験を行った。SIFT で記述される特徴量はアフィン変換に対して不変ではないことを考慮し、今回の実験では、左右の首の回転についての画像は首軸まわりの回転を小さい画像セットを撮影した。実験結果例は以下の図 17 と図 18 に示す。

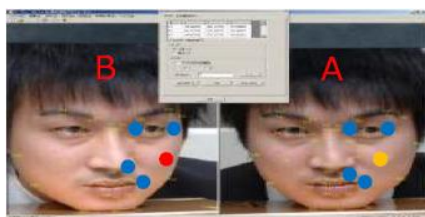
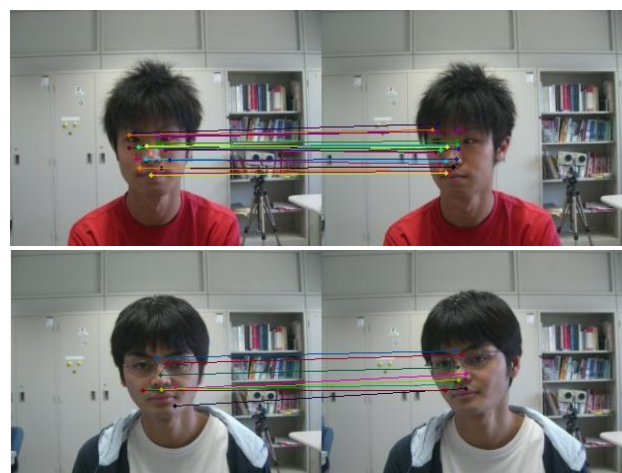
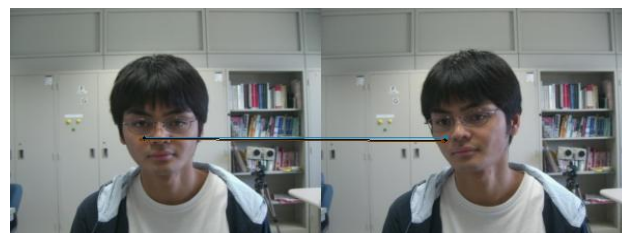
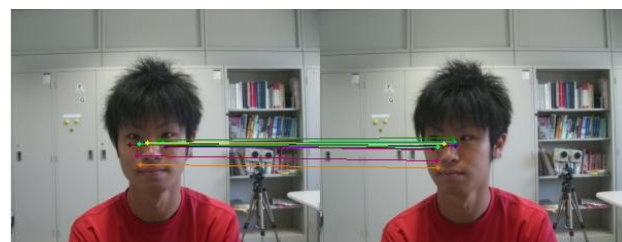


図 15 ホモグラフィ変換のイメージ



L = 5



L = 6

図 16 二人の人物における NF の対応付け結果



図 17 人物 A の SIFT における対応付けの結果



図 18 人物 B の SIFT における対応付けの結果

## 5. 結論

本論文では一つ目に、顔パーツのコーナー検出に特化しているとされているFAST (Features from Accelerated Segment Test) を提案した。また二つ目には、頬の特徴量記述に適したNF(No Features)を提案した。

実画像を用いた実験より、顔パーツのコーナー特徴点、頬の特徴点の検出と対応付けを考察した。まず、特徴点抽出において顔パーツコーナー点については顔の領域を抽出し、ブロック単位の局所的な明度の正規化を行ってからFAST処理を行うことにより正規化前よりも多くの特徴点が得られることも確認できた。そして頬の特徴点抽出では顔パーツ特徴点を元としてボロノイ処理によりボロノイ母点を頬の点とした。顔パーツコーナー点、頬の点それぞれでまだ多くの特徴点が抽出されており、今回AAMと最近傍探索により点の絞り込みを提案した。しかし、現段階ではクエリ点を真値で行っているため、今後AAMの点をクエリ点として最近傍探索を行う予定である。

一方、対応付けにおいて、顔コーナーパーツ点ではSIFTで特徴点の特徴量記述し、一部対応づけを行えることも確認できたがSIFTが奥行方向の回転に対して不変ではないので、正面顔との対応付けは首軸まわりで回転した顔に比べて、顔の傾きによる回転をした顔のほうが対応付けのパフォーマンスが高いことが分かった。今後、アフィン変換に強い新たな特徴量記述の考察をする必要がある。さらに、コーナーパーツ点の情報を元にして頬からの特徴点を検出する方法も考察していかなければならない。

頬の特徴点ではNFで特徴量記述し対応付けをさせたが閾値Lの変化によって誤対応の数と正対応の数の差が見ら

れた。これについては顔サイズ依存の問題やや閾値の調整が上手く行えていないことが一つの原因と考えられるのでまずそちらをこれから考察してみる。

## 謝辞

本研究の一部は、文部科学省科学研究費補助金基盤研究(c) 24500205 の補助を受けている。

## 参考文献

- 1) FAST Corner Detection:  
<http://www.edwardrosten.com/work/fast.html>
- 2) 湯浅真由美、武口智行、小坂谷達夫、山口修:静止画面顔認証のための自動顔特徴点検出、信学技報、PRMU2006-222, 2007
- 3) H. WU, J. INADA, T. SHIOYAMA, Q. CHEN, and T. SHIMADA, 'Automatic Facial Feature Points Detection with SUSAN Operator', 12th Scandinavian Conference on Image Analysis, pp.257-263, 2001.
- 4) 小谷秀輝,関口利彦,真壁弘樹,赤松茂:カラー画像処理と部分空間類別法を用いたシーン中の顔検出、信学技報、PRMU2003-278, 2004
- 5) E. Rosten and T. Drummond, Machine learning for high-speed corner detection, European Conference on Computer Vision. 2006
- 6) Lowe, David G. Object recognition from local scale-invariant features, Proceedings of the International Conference on Computer Vision. 2. pp. 1150-1157, 1999
- 7) Ralf Dragon, Muhammad Shoaib, Bodo Rosenhahn, and Joern Ostermann, NF-Features – No-Feature-Features for Representing Non-textured Regions