

100万顔画像データベースに対する34msでの照合の実現

近似最近傍探索を用いた大規模高速顔画像検索

前川 敬介[†] 内海ゆづ子[†] 岩村 雅一[†] 黄瀬 浩一[†]

[†] 大阪府立大学 大学院工学研究科 〒599-8531 堺市中区学園町 1-1

E-mail: [†]maekawa@m.cs.osakafu-u.ac.jp, ^{††}{yuzuko,masa,kise}@cs.osakafu-u.ac.jp

あらまし 本稿では大規模データベースに対する高速顔認識手法を提案する．防犯や犯罪捜査を目的として顔認識を利用する場合，多数の人物を認識する必要がある．そのため，データベースの規模が拡大した場合に高速に認識を行うことは重要な課題である．従来の顔認識では数百枚程度のデータベースに対して認識性能の評価が行われているため，計算時間が考慮されていない．全探索などの手法は，データベース内の特徴量すべてと距離計算を行う必要があり，計算時間はデータベースの規模に比例するため，データベースの規模が拡大すると認識に膨大な処理時間がかかる．本稿では，顔認識処理時間の削減のために近似最近傍探索を用いて特徴量の検索を行う．ハッシュを利用し，対応点の探索範囲を限定することで，距離計算の回数を削減し処理の高速化を行う．Web上から収集した100万枚の顔画像データベースに対して認識を行った結果，特徴点検出を用いる方法ではハッシュを用いない場合と比較して13,252倍高速に認識した．また，Dense Samplingを用いることで最速で34msで認識が可能となった．

キーワード 顔認識，PCA-SIFT，近似最近傍探索，高速化，大規模データベース

1. はじめに

近年，犯罪やテロへの対策として，個人認証を用いたセキュリティが強化されており，指紋，虹彩，静脈，顔画像などの生体特徴を用いた個人認証技術に対する関心が高まっている．これらの中で代表的なものとして顔認識が挙げられる．顔認識は，カメラで撮影した顔画像を用いて認識を行うため，機器への接触を必要とせず認識を実現できる．そのため，データ計測に機器への接触を必要とする指紋などを用いた方式と比べて，離れた場所からも認識対象を計測し，認識ができるといった利点がある．この利点から，監視カメラに顔認識技術を組み込み，特定の人物を見つけ出すといった用途にも利用できる．さらに，認識に顔画像を用いるため，指紋，虹彩といった人が見ても誰か判別できない生体情報とは違い，撮影された顔画像と認識結果の顔画像を記録することで，認識結果を人間の目で確認することも可能である．このような利点から，顔認識は，コンピュータビジョンの分野において盛んに研究されてきた．

顔認識の利用法の1つとして，監視カメラの映像から，犯罪者などを見つけ出すといった犯罪捜査を目的としたものが挙げられる．本研究では，犯罪捜査支援のための顔認識システムの構築を目標とする．犯罪者を見つけ出すためには，あらゆる犯罪者を認識する必要があり，登録人物のデータベースは大規模になる．また，監視カメラの映像中における画像に対して顔認識を行うためには，実時間で処理が必要とされる．そのため，顔認識手法は大規模なデータベースに対して高速な処理が可能でなければならない．加えて，認識対象が犯罪者である場合，

データベース画像として数枚の画像しか与えられないこともあるため，登録画像が一人一枚の場合でも認識できる手法が望ましい．

これまでの顔認識では様々な手法が提案されてきた．代表的なものとして，学習データに対して最も識別能力の高い線形写像を求める線形判別分析(LDA)[16]が挙げられる．線形判別分析は実装が簡単な上，認識結果が良好なため顔認識によく用いられてきた．また，カーネルを用いて非線形の高次元空間に写像された特徴量に対して線形判別分析を行うことで，非線形なデータの分布に対応したものも提案されている[17]．

カーネルを用いて特徴量を高次元空間に写像し，識別を行うSupport Vector Machine (SVM)[18]も顔認識に用いられる認識手法の一つである．SVMは分類境界と学習サンプルの距離であるマージンが最大となる平面を分類境界とする．このことにより，汎化誤差が低く抑えられ，ほかの識別器と比較して良い性能を示している[19]．非線形のカーネル関数を用いれば，非線形に分布する特徴量の分類が可能となる．

ほかに，各人物の顔画像特徴量の分布を部分空間や多様体(Manifold)で表し，認識する手法が盛んに行われている[20],[21]．これらの手法では，顔の向きや照明の異なった複数の画像を用いて学習を行う．そのため，入力される顔画像の向きや照明条件が任意の場合でも高精度に認識が可能となる．

ここまでで挙げて来た手法では，高い認識精度で顔認識が行えるが，各クラスごとに学習に多くのサンプルを必要とするため，1つのクラスに対する学習サンプルが少ない場合は適応できない．また，従来の顔認識の研究では，CAS-PEAL[1]や

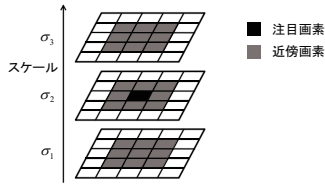


図 1 極値の検出

FERET [2] といった数百から数千枚程度の小さなデータベースに対して認識性能の評価がされており、大規模なデータベースを用いて高速に認識できる手法については、ほとんど研究がされていない．そこで、本稿では大規模データベースを用いた高速顔認識を実現するため、野口らにより提案されているハッシュを用いた近似最近傍探索 [3] を顔認識に適用し、認識の高速化を図る．野口らの手法は、データベースに登録されている特徴量をハッシュテーブルに格納し、クエリと一致するハッシュ値に登録されている特徴量のみを探索に用いる．このように探索範囲を限定することで、距離計算が大幅に削減され、高速な検索が可能となる．100 万枚顔画像データベースを用いて評価実験を行った結果、特徴点検出を用いる方法では全探索と比較して 13,252 倍高速に認識を行った．また、Dense Sampling を用いることで最速で 34ms で認識が可能となった．

2. 特徴抽出

本研究では特徴量に野口らによる物体認識手法 [3] で用いられている PCA-SIFT [4] を使用する．PCA-SIFT は主成分分析 (PCA) と SIFT (Scale Invariant Feature Transform) [5] を組み合わせた特徴抽出手法である．SIFT は、特徴点の検出と特徴量の記述を行うアルゴリズムである．画像の回転・スケール変化・照明変化などに頑健な特徴量を記述でき、認識性能が良かったため、物体認識で広く用いられている [6]．また、従来研究において、顔認識に SIFT を用いた場合、良好な認識結果が得られることが報告されている [7]．

PCA-SIFT では、まずキーポイントを検出し、検出したキーポイント周辺の勾配情報を用いて特徴量を記述する．本研究では、PCA-SIFT を抽出する際に、キーポイント検出を用いた手法と、Dense Sampling により特徴点を決定する 2 つの手法を用いた．以下、従来の PCA-SIFT と Dense Sampling による特徴点の決定について詳しく述べる．

2.1 PCA-SIFT

PCA-SIFT では、キーポイント周辺の勾配情報に対して PCA を行う．抽出した特徴量をあらかじめ主成分分析により作成した部分空間に射影することで、特徴次元数を削減する．SIFT の特徴次元数は 128 次元であるのに対し、PCA-SIFT では 36 次元に圧縮するのでマッチングを高速に行えるという利点がある．PCA-SIFT の処理はキーポイントの検出と特徴量の記述の 2 段階からなる．

2.1.1 キーポイント検出

PCA-SIFT におけるキーポイント検出は SIFT と同様のものである．DoG (Difference-of-Gaussian) 処理を用いてスケ-

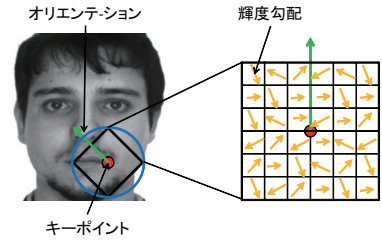


図 2 キーポイントからの特徴抽出

ルスペースにおける極値探索をすることで、キーポイントの位置とスケールを決定する．キーポイント候補点は、スケールの異なるガウス関数 $G(x, y, \sigma)$ と入力画像 $I(u, v)$ を畳み込んだ平滑化画像 $L(u, v, \sigma)$ の差分 (DoG 画像) から求める．ここで、 σ はガウシアン関数のスケール、 u, v は注目画素の座標であり、 x, y はそれぞれ注目画素からの垂直方向、水平方向の距離である．平滑化画像 $L(u, v, \sigma)$ とガウス関数 $G(x, y, \sigma)$ は、以下の式により求める．

$$L(u, v, \sigma) = G(x, y, \sigma) * I(u, v) \quad (1)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (2)$$

DoG の結果の画像を $D(u, v, \sigma)$ とすると、DoG 画像は次式で表される．

$$\begin{aligned} D(u, v, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(u, v) \\ &= L(u, v, k\sigma) - L(u, v, \sigma) \end{aligned} \quad (3)$$

この処理を、 σ の初期値 σ_0 から k 倍ずつ大きくした異なるスケール間で行い、複数の DoG 画像を求める．得られた DoG 画像から極値を検出し、キーポイントとスケールを決定する．図 1 に示すように、DoG 画像の注目画素を、近傍点と、上下スケールの画素 26 点と比較し、極値であった場合、その画素をキーポイント候補点として検出する．そして、そのときの DoG 画像のスケールをキーポイントのスケールとする．

2.1.2 特徴量の記述

特徴量の記述は、まず、検出された各キーポイントのオリエンテーションを求める．オリエンテーションは、キーポイント周辺の勾配情報から算出され、キーポイントにおける方向を表す．オリエンテーションは、キーポイント周辺の勾配情報から作成される重み付き方向ヒストグラム h_{θ} により決まる．ここで、 h_{θ} は、全方向を 36 方向に量子化したヒストグラムである．この 36 方向のヒストグラムの最大値の 80% 以上となるピークをキーポイントのオリエンテーションとして割り当てる．図 2 に特徴量記述の例を示す．まず、回転に不変な特徴量を抽出するため、検出したキーポイントを中心とした領域を、オリエンテーションの方向に正規化する．その後、キーポイントの周辺領域から水平方向、垂直方向の勾配を算出し、特徴量を記述する．特徴量記述に用いられる領域の大きさは、キーポイントのスケールを σ とすると、元画像において $\sigma \times 20$ [pixels] に相当する．この処理により記述される特徴量は 3042 次元である．大量の一般画像から抽出された SIFT 特徴量を用いて PCA で 36 次元の部分空間を求め、得られた特徴量をこの部分空間に

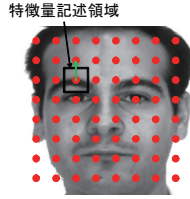
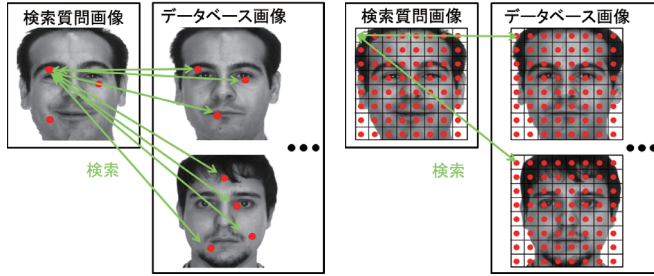


図 3 Dense Sampling を用いたときの特徴点



(a) キーポイント検出 (b) Dense Sampling

図 4 認識処理

射影することで 36 次元の PCA-SIFT 特徴量を抽出する。

2.2 Dense Sampling

照明などの影響により特徴点が少ししか検出されない場合、認識率は低下する。そのため、画像によらず一定量の特徴量を抽出できれば認識に有効であると考えられる。そこで、特徴点の検出アルゴリズムを使用せず、特徴量を抽出する領域を固定し、Dense Sampling により特徴量を抽出する方法を検討する。本研究では、画像に特徴点を格子状に均等に配置し、各特徴点の周辺領域から特徴抽出を行う。特徴抽出に用いるスケールとして $\sigma = 2, 4, 6, 8, 10$ を用いる。また、オリエンテーションは常に 0 とし、特徴量記述領域の回転は行わない。 $\sigma = 4$ のときの特徴点の位置と特徴量記述領域を図 3 に示す。スケールが小さいときは特徴点の間隔を狭めて配置する。特徴点の数は、 $\sigma = 2$ のとき $20 \times 20 = 400$ 点、 $\sigma = 10$ のとき $4 \times 4 = 16$ 点のようになり、合計 880 領域から PCA-SIFT 特徴を抽出する。

3. 認識処理

画像の認識時には、特徴点ごとに、画像 ID に投票処理を行う。データベース画像の特徴量のうち、検索質問画像 (クエリ) から得た特徴量と距離の値が最小となる特徴量を求め、その特徴量が抽出された画像 ID に投票する。距離尺度にはユークリッド距離を用いる。キーポイント検出を行う場合は、図 4(a) に示すようにすべての特徴量を検索の対象とする。一方、Dense Sampling の場合は、図 4(b) に示すように検索質問画像から得た各特徴量に対して、同じ局所領域から抽出された特徴量のみを検索の対象とする。これにより、特徴点の誤対応を減らし、かつ、計算コストを低減することができる。特徴量の検索処理を検索質問画像のすべての特徴量に対して行い、最終的な得票数が最も多い画像 ID を認識結果として出力する。

4. 高速化

特徴量の検索処理では、データベース内の特徴量すべてと距

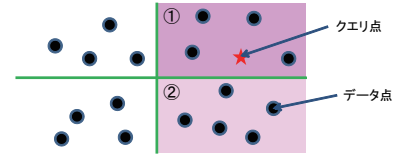


図 5 検索の多段階化

離計算を行った場合、データベースの規模に比例して処理時間が増加する。そのため、データベースが大規模になると、処理時間が膨大になる。そこで、計算時間を短縮するために近似最近傍探索を用いて検索を行う。本章では、本研究で用いた野口らの手法について述べる。

4.1 ハッシュ関数

野口らの手法では、ハッシュ関数を用いて近似最近傍探索を行う。ハッシュ関数は、データベース画像の特徴量をハッシュテーブルに登録する際や、クエリの特徴量から最近傍の特徴量を検索する際に用いられる。PCA-SIFT で得られる 36 次元の実数値ベクトル x を、第 1 次元から第 d 次元 ($d \leq 36$) までとり、 $\hat{x} = (x_1, x_2, \dots, x_d)$ とする。そして、特徴量の次元ごとに次式を用いて二値化を行い、ビットベクトル $u = (u_1, \dots, u_d)$ を作成する。

$$u_j = \begin{cases} 1 & \text{if } x_j - \mu_j \geq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

本研究では、しきい値を $\mu_j = 0$ とした。そして、以下のようにしてハッシュ値 H_{index} を求める。

$$H_{\text{index}} = \left(\sum_{i=1}^d u_i 2^{i-1} \right) \bmod H_{\text{size}} \quad (5)$$

ここで H_{size} はハッシュ表のサイズである。(5) 式からハッシュ値を求め、特徴量と画像 ID をハッシュ表に登録する。画像 ID は、登録された特徴量がどの画像から抽出されたものであるかを示す。データをハッシュに登録する際に衝突が生じた場合、複数の特徴量をリストとして登録する。しかし、リストが長すぎると検索時の距離計算のコストがかかりすぎるという問題が生じる。そこで、リスト長 n に対する閾値 c を設け、 $n > c$ を満たすとリスト全体をハッシュ表から削除する。同じハッシュ値を持つ特徴量が多いということは、画像の特定性が低く認識にあまり役立たない特徴量であるため、削除しても影響は少ないと考えられる。

4.2 検索の多段階化

検索時には検索質問画像から得た各特徴量 $q = (q_1, \dots, q_d)$ に対してハッシュ関数を適用し、ハッシュ表から最近傍となる特徴量 x_* を求める。そして、 x_* に対応する画像 ID に投票する。このとき、クエリの特徴量が撮影条件によって変動するため、データベース画像の特徴量とクエリの特徴量から算出したハッシュ値が異なり、対応する特徴量を得られないことがある。そこで、値の変動幅 e を設け、変動に対処する。具体的には、クエリから得た特徴量を $q = (q_1, \dots, q_d)$ 、 q のビットベクトルを $u = (u_1, \dots, u_d)$ とするとき、 $|q_j - \mu_j| \leq e$ を満たす次元

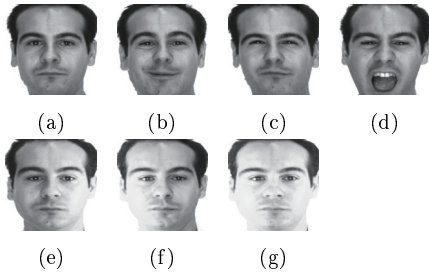


図 6 AR Face Database の画像例 . (a) neutral , (b) smile , (c) anger , (d) scream , (e) left side light on , (f) right side light on , (g) all side light on

j に対しては, u_j の値を反転させたビットベクトルも用いて特徴量を検索する . しかし, $|q_j - \mu_j| \leq e$ を満たす次元が多数存在すると, 探索範囲が増え, 莫大な計算時間が必要となる . そこで, 処理対象となる次元数を b に制限する . ビットの反転条件を満たす次元の数が b より多いときには, 対象とする次元を $|x_j - \mu_j|$ の値が小さいものから b 個とする . 図 5 の例では, まずクエリの特徴量が入ったピン①が探索範囲となる . 次に, しい値 μ_j に近い次元のビットを反転させた②が探索範囲として追加される . ビット反転により, 探索範囲を順次拡大しながら認識を行い, 認識結果に対する十分な信頼性が得られた段階で処理を打ち切る . 1 位の得票数を V_1 , 2 位の得票数を V_2 とすると, $V_1 > t, rV_1 > V_2$ を同時に満たすとき, 処理を打ち切って 1 位の画像を回答とする . ここで, t は得票数のしい値, r は 1 位と 2 位の得票数の比のしい値である . 検索を多段階化することで, 簡単に認識できる画像は高速に, 誤認識しやすい画像は時間をかけて精密に認識を行うことができ効率的な認識が可能となる .

5. 実験

提案手法の認識の高速化に関する性能を評価するために実験を行った . データベースとして, 公開顔画像データベースとウェブ上からダウンロードした画像を合わせ, 100 万枚の顔画像データベースを作成した . 使用した公開顔画像データベースは, AR Face Database [8], CAS-PEAL [1], FERET [2], The ORL Database of Faces [22], Georgia Tech Face Database, [23], Surveillance Cameras Face Database [9], Indian Face Database [10] である . 実験に使用した顔画像は, 目や鼻などの顔特徴点を 14 点自動で検出し, 得られた特徴点に基づいて顔の切り出しと顔向き正規化を行っている . [11] [12] [13] . データベース中には, 顔の切り出しや正規化に失敗した画像も含まれている . 正規化後の画像は 8bit グレースケールで, 解像度は 512×512 [pixels] である . データベースには AR Face Database の Set 1 の画像のうち, 正規化に失敗した 2 枚の画像を除いた 132 人 132 枚の画像が含まれている . クエリとして, AR Face Database の Set 2 ~ Set 7 の画像 792 枚 (132 枚 \times 6 セット) を用いた . AR Face Database の Set 1 ~ Set 7 は図 6 に示すように, neutral, smile, anger, scream, left side light on, right side light on, all side light on の画像で構成されて

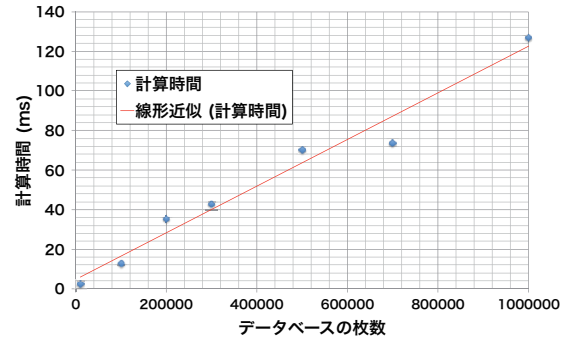
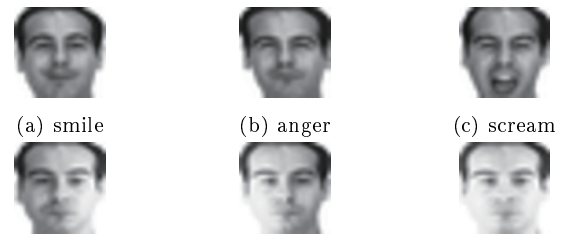


図 9 データベースの枚数と計算時間



(d) left side light on (e) right side light on (f) all side light on

図 10 低解像度画像例

いる . 処理時間は特徴量の検索にかかった時間のみを測定し, 画像の正規化や特徴抽出の時間は含まない . 実験に使用した計算機は, CPU が AMD Opteron(tm) Processor 6174(2.2GHz), メモリは 256GB である .

5.1 認識率と処理時間の評価

PCA-SIFT 特徴量と近似最近傍探索を用いた場合の認識率の評価と計算時間の評価を行った . はじめに, 近似最近傍探索のパラメータを 1 つに固定し, 累積分類率の変化を調べた . 特徴点検出を用いた方法 (Detector) と Dense Sampling を用いた方法の累積分類率の結果を図 7 に示す . 多くのクエリにおいて上位 100 位で認識率がほぼ横ばいになることがわかる . そこで, 以降の実験では 100 位累積分類率を認識率の評価値として用いることとする .

続いて, 近似最近傍探索のパラメータ e, b, c, d, t, r を様々に変化させ, 各クエリに対する 100 位累積分類率と処理時間を調べた . 各クエリに対する結果を図 8 に示す . 実験の結果, scream, all side light on 以外のクエリでは 96% 以上の高い認識率が得られた . smile に対する, Detector の全探索の認識率は 97.7% であり, 処理時間は 1683 s であった . 近似最近傍探索の結果, 図 8(a) より Detector では認識率 96.2% において処理時間 125 ms で認識ができており, 全探索と比べて 1.5% の認識率の低下で, 13,464 倍の高速化を達成した . また, Detector は, smile, anger, left side light on では 100 ms 前後の処理時間で認識ができており, Dense Sampling では right side light on に対して最速で 34 ms で認識ができており, Detector と Dense Sampling を比較すると, smile, anger, left side light on, right side light on では最高の認識率にはほとんど差は見られない . scream に対する認識率はどちらも低かった . この原

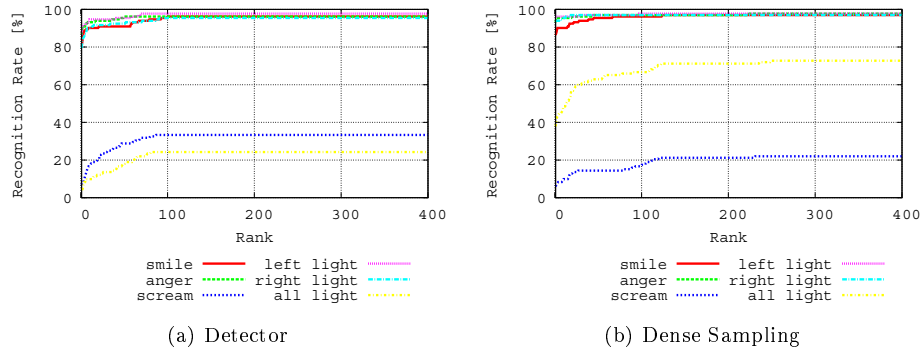


図 7 認識結果の累積分類率

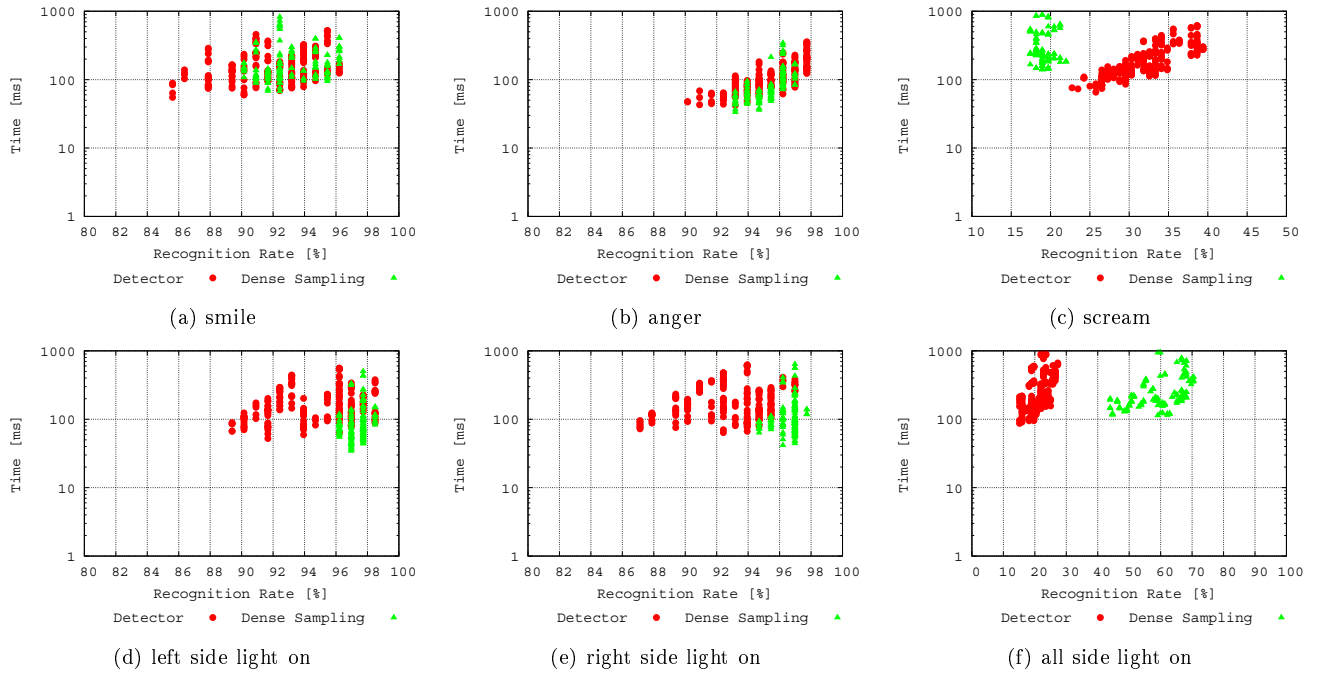


図 8 認識率と処理時間

因として、クエリの表情変化が大きく、得られた特徴量が対応するデータベース画像の特徴量から大きく変化してしまったことが挙げられる。照明変化のあるクエリに関して、all side light on では特に Detector の認識率が低く、また、left side light on, right side light on でも Dense Sampling の方が良い結果であった。図 8(e) から right side light on では、認識率 97.0%において、Dense Sampling は Detector よりも約 200 ms も高速であることがわかる。Detector で計算時間が大幅にかかったのは、照明により画像が白飛びし、検出できる特徴点の数が減ったことで、認識の多段階化の打ち切りがうまくできなかったことが原因であると推定される。一方で Dense Sampling は画像によらず同じ位置から一定量の特徴を抽出できるため、照明の変動があった場合でも高速化手法が有効に働いたと思われる。表情変化のあるクエリ smile, anger, scream では Detector の方が良い結果であった。この理由として、表情の変化により顔の部位にずれが生じるが、Detector では顔の同じ部分を検出でき、同じ特徴量を抽出できたことが考えられる。

5.2 データベースの枚数と計算時間

データベースの枚数が増加したときの計算時間を予測するた

め、データベースの枚数と計算時間の関係を調べた。データベースの枚数を 100 万枚までで変化させ認識を行った。Detector を用いた PCA-SIFT を使用し、Smile に対して高い認識率を得られた時の処理時間を測定した。結果を図 9 に示す。実験の結果から、データベースの枚数に比例して処理時間が増加していることがわかった。

5.3 低解像度画像の認識

監視カメラで撮影される実際の画像は、カメラと撮影対象の人物の顔の距離が離れているため、一般的に低解像度の画像であることが多い。そこで、低解像度の画像に対する認識性能を評価するための実験を行った。データベース画像の解像度は 512×512 [pixels]、クエリ画像は 512×512 [pixels] の画像を 25×25 [pixels] に縮小後、再度 512×512 [pixels] に拡大したものをを用いた。画像のリサイズにはバイキュービック法を使用した。5.1 節の実験で smile に対して最高の認識率を得たパラメータを用いて近似最近傍探索を行い、100 位累積分類率で評価した。結果を表 1 に示す。表 1 より、Detector では smile で 33.9%, right side light on では 80.9% 認識率が低下した。元の画像からは平均約 180 点の特徴点が発見されたのに対し、低解

表 1 低解像度画像での認識率 [%]

	size	Query set					
		smile	anger	scream	left light	right light	all light
Detector	512×512	96.2	96.2	33.3	97.7	95.5	24.2
	25×25	63.6	83.3	6.1	41.0	18.2	0
Dense sampling	512×512	96.2	97.0	17.4	97.7	97.0	66.7
	25×25	79.5	92.4	9.1	88.6	81.1	20.5

像度画像からは特徴点が平均 25 点しか検出されなかった。検出した特徴点が大幅に減少したため、Detector では認識率が大きく低下してしまったと考えられる。Dense Sampling では、Detector よりも認識率の低下が小さく、低解像度画像ではすべてのクエリセットで Dense Sampling の方が Detector より高い認識率が得られた。これは、Detector ではほとんど特徴点が検出されなかったのに対し、Dense Sampling では常に一定量の特徴量が得られるためであると考えられる。

6. まとめ

本稿では、大規模データベースを用いた高速顔認識を実現するため、近似最近傍探索手法である野口らの手法を用いて顔認識を行った。100 万枚の顔画像データベースを用いた実験の結果、特徴検出を自動で行う Detector では、AR Face Database の smile に対して 125 ms で認識ができ、全探索に比べて 13,464 倍高速に認識した。また、特徴点の位置を固定した Dense Sampling では right side light on に対して 34 ms で認識が可能となった。両側に照明が当たっている場合と大きく表情変化した場合を除いて、約 100 ms の処理時間で 96% 以上の認識率が得られた。照明が強い画像や低解像度の画像では Detector による特徴点検出が困難なため、常に一定量の特徴点が得られる Dense Sampling が有効であることがわかった。今後の課題として、データベースの規模を拡大し評価を行うことや、更なる認識率の向上のため、Gabor Wavelet 特徴量 [14] や Haar like 特徴量 [15] などの特徴量を用いることがあげられる。また、Dense Sampling では画像の端などの認識にあまり有効でない点が特徴点として含まれていた。Dense Sampling で用いた特徴点からブースティングアルゴリズムを用いて特徴点の選択を行い認識性能の向上を図ることも今後の課題である。

文 献

- [1] W. Gao, B. Cao, S. Shan, D. Zhou, X. Zhang, D. Zhao, and S.S.E. Al, "The CAS-PEAL large-scale Chinese face database and evaluation protocols," Technical report, Joint Research & Development Laboratory, CAS, 2004.
- [2] J.P. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss, "The feret evaluation methodology for face-recognition algorithms," IEEE Trans. on PAMI, vol.22, no.10, pp.1090–1104, 2000.
- [3] 野口和人, 黄瀬浩一, 岩村雅一, "近似最近傍探索の近似と多段階化による物体の高速認識", 画像の認識・理解シンポジウム (MIRU2007) 論文集, pp.111–118, 2007.
- [4] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," CVPR'04, vol.2, pp.506–513, 2004.
- [5] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," IJCV, vol.60, pp.91–110, 2004.
- [6] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Trans. on PAMI, vol.27, no.10, pp.1615–1630, 2005.
- [7] J. Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade, and B.L. Lu, "Person-specific sift features for face recognition," ICASSP 2007, vol.2, pp.593–596, 2007.
- [8] A. Martínez and R. Benavente, "The AR face database," Technical Report 24, Computer Vision Center, Bellaterra, 1998.
- [9] M. Grgic, K. Delac, and S. Grgic, "Scface - surveillance cameras face database," Multimedia Tools and Applications, pp.1–17, 2009.
- [10] V. Jain and A. Mukherjee, "The indian face database," 2002. <http://vis-www.cs.umass.edu/~vidit/IndianFaceDatabase/>
- [11] T. Mita, T. Kaneko, B. Stenger, and O. Hori, "Discriminative feature co-occurrence selection for object detection," IEEE Trans. on PAMI, vol.30, pp.1257–1269, 2008.
- [12] M. Yuasa, T. Kozakaya, and O. Yamaguchi, "An efficient 3D geometrical consistency criterion for detection of a set of facial feature points," IEICE - Trans. Inf. Syst., vol.E91-D, pp.1871–1877, 2008.
- [13] T. Kozakaya and O. Yamaguchi, "Face recognition by projection-based 3d normalization and shading subspace orthogonalization," FGR '06, pp.163–168, 2006.
- [14] L. Wiskott, J.-M. Fellous, N. Kruger, and C. von derMalsburg, "Face recognition by elastic bunch graph matching," IEEE Trans. on PAMI, vol.19, pp.775–779, 1997.
- [15] Y. Pang, X. Li, Y. Yuan, D. Tao, and J. Pan, "Fast haar transform based feature extraction for face representation and recognition," Trans. Info. For. Sec., vol.4, pp.441–450, 2009.
- [16] T.-K. Kim, B. Stenger, J. Kittler, and R. Cipolla, "Incremental linear discriminant analysis using sufficient spanning sets and its applications," IJCV, vol.91, pp.216–232, 2010.
- [17] J. guoWang, Y. shengLin, W. kouYang, and J. yuYang, "Kernel maximum scatter difference based feature extraction and its application to face recognition," Pattern Recognition Letters, vol.29, pp.18320–1835, 2008.
- [18] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machines: Global versus component-based approach," ICCV '01, vol.2, pp.688–694, 2001.
- [19] P.J. Phillips, "Support vector machines applied to face recognition," Advances in Neural Information Processing Systems, vol.11, pp.803–809, 1998.
- [20] Arandjelović, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell, "Face recognition with image sets using manifold divergence," CVPR '05, vol.1, pp.581–588, 2005.
- [21] T.-K. Kim, J. Kittler, and R. Cipolla, "On-line learning of mutually orthogonal subspace for face recognition by image sets," IEEE Trans. on PAMI, vol.19, no.4, pp.1067–1074, 2010.
- [22] "ORL Face Database," <http://www.c1.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
- [23] "Georgea Tech Face Database," http://www.anefian.com/research/face_reco.htm