

解像度低下と照明変動に頑健な大規模顔画像認識

北野 豊[†] 内海ゆづ子[†] 岩村 雅一[†] 黄瀬 浩一[†]

[†] 大阪府立大学大学院研究科 〒 599-8531 大阪府堺市中区学園町 1-1

E-mail: kitano@m.cs.osakafu-u.ac.jp, {yuzuko,masa,kise}@cs.osakafu-u.ac.jp

あらまし 数百万枚の顔画像を含む大規模データベースに対して、高速・高精度な顔認識が可能な手法として、坂野らの手法がある。この手法では、局所特徴の検出器を使用して、画像に応じて特徴抽出箇所を決定する。検出器は低解像度顔画像や照明変化に弱いため、坂野らの手法もこれらの画像変化を受けた顔画像の認識に弱いという欠点がある。この問題を解決する方法として、局所特徴の検出方法としてあらかじめ定められた位置から特徴点を密にとる Dense Sampling を用いることが考えられる。しかし、これではメモリ使用量が大幅に増加し、また表情変化に弱くなる。そこで本稿では、Dense Sampling を導入することで生じるメモリ使用量と表情変化の問題を同時に解決する方法を提案する。これにより、低解像度顔画像や照明変化を受けた顔画像の認識性能を向上しつつ、省メモリで表情変化に頑健な認識を実現する。

キーワード 顔認識, 大規模データベース, 局所特徴量, 近似最近傍探索, Dense Sampling, 遺伝的アルゴリズム

1. はじめに

犯罪捜査では、監視カメラの映像から犯罪者を特定するために、顔画像をもとに犯人を特定することが度々求められる。その際に、自動で、データベースに登録されている犯罪者の候補を絞ることができれば犯罪捜査での人手と捜査時間の大幅な削減につながると考えられる。犯罪捜査支援を目的として顔認識を考えた場合、必要なことが三つある。一つ目は、大規模なデータベースで認識できることである。犯罪者の顔画像は大量になることが考えられ、データベースが大規模になった際も認識できる手法であることが望ましい。二つ目は、高速で認識できることである。三つ目は、低品質の画像でも認識可能なことである。監視カメラから得られる顔画像は低解像度であったり、照明変化を受けていたり、低品質の画像である場合が多いため、低品質の画像でも認識できる手法が望ましい。また、認識対象が犯罪者であるため、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り、顔認識システムに登録された画像が1人1枚の場合でも認識できる手法が望ましい。

登録画像が1人1枚の場合でも認識でき、大規模なデータベースでも高速で高精度な認識が可能な手法に、坂野らの手法 [1] がある。坂野らの手法では、特徴点検出によって得られた特徴量を用いて顔認識を行っている。認識の際、近似最近傍探索を用いることで認識速度を向上させ、これにより失われる認識性能を投票処理で補うことで高精度な認識を実現する。

しかし、低解像度の画像や照明変化を受けている画像では特徴点検出精度が低くなり、高い認識率が得られない。そこで、坂野らの手法では、Dense Sampling [2] を用いた手法も提案されている。この手法では、特徴点の検出を特徴点検出で検出するのではなく、Dense Sampling によって等間隔に決定することで一定数の特徴を確保し、低解像度や照明変化のある画像で

も高い精度を得ることができる。しかし、Dense Sampling で、特徴点を大量に取ってしまうため、メモリ使用量が増加し、大規模な問題を扱うことが難しくなる。また、Dense Sampling を用いた場合は顔画像から特徴点を適応的に決める場合よりも表情変化に弱くなる。

そこで本稿では、Dense Sampling を用いることの欠点である、メモリ使用量の増加、表情変化への脆弱性を改善する。表情変化への脆弱性は、認識の際に、周辺の特徴点とも照合することにより改善する。メモリ使用量の増加は、Dense Sampling で決定する特徴点を、認識により有効であるものを選択することにより改善する。Dense Sampling の有効な特徴点の選択には、組み合わせ最適化問題を解く手法である遺伝的アルゴリズムを用いた。このとき、周辺の特徴点との照合を行えるようにするため、特徴点間の相関が高いものを優先して選択するように最適な Dense Sampling 特徴点の組み合わせを得た。

以降、第2章で坂野らの手法を説明する。第3章では、提案手法である特徴点選択手法について述べ、第4章で実験結果を示し、最後に第5章で本論をまとめる。

2. 坂野らの手法

坂野らの手法では、図1に示すように、まず顔画像を正規化をし、顔画像ごとの顔パーツの位置を合わせる。次に、Dense Sampling により特徴点を決定し、決定された特徴点から PCA-SIFT 特徴量 [3] を抽出する。そして、得られた特徴量を用いて、投票処理ベースの認識をする。認識の際、近似最近傍探索を用いることで、高速化を実現している。以降、文献 [1] の手法において局所特徴の検出器を用いた場合の学習処理と、認識処理、ならびに文献 [2] の手法において検出器の代わりに Dense Sampling を用いる場合の処理について述べる。

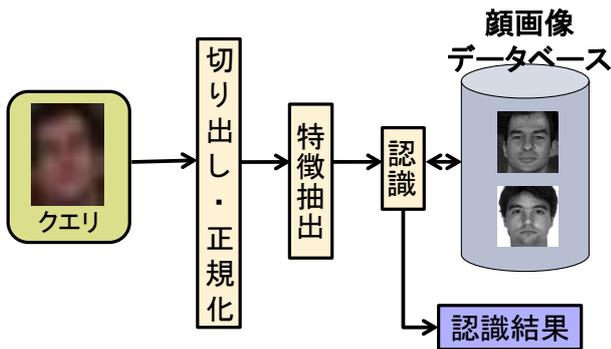


図1 坂野らの手法の流れ

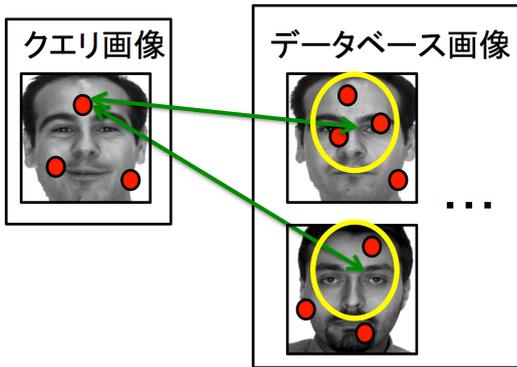


図2 距離によるマッチングの特徴点の限定

2.1 学習処理

坂野らの手法 [1] では、特徴量は PCA-SIFT 特徴量を使用している。PCA-SIFT では、まず局所特徴の検出器によって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。得られた特徴量は近似最近傍探索のためにあらかじめインデクシングして、データベースに登録する。

2.2 認識処理

認識処理は、クエリとデータベースの特徴量のマッチングを行い、投票処理により認識を行う。マッチングの処理を高速化するため、近似最近傍探索を利用する。マッチングの際、誤対応を減らすため、マッチングに用いる特徴点を限定する。特徴量のマッチングの方法、投票による認識、マッチングに用いる特徴点の限定について以下で述べる。

2.2.1 特徴量のマッチング方法

特徴量のマッチングにデータベースの全ての特徴量を用いた場合、特徴量が顔の別の部分と誤対応する可能性がある。そのため、坂野らの手法では、図2のように、クエリの特徴点の位置を中心として一定距離内にある特徴点から得られる特徴量を探索の対象にする。

2.2.2 投票による認識

近似最近傍探索により k 近傍点を見つけ、距離の逆数を重みとして、 k 近傍の画像 ID にそれぞれ投票する。クエリの画像から得られた特徴量すべてで投票を行い、集計したものの得票数が多い順に絞り込みの候補とする。本稿では、正解画像の候補の絞り込みを目的としているため、得票数の多い画像 ID の上位 m 枚の中に正解画像があった場合、認識に成功したとする。

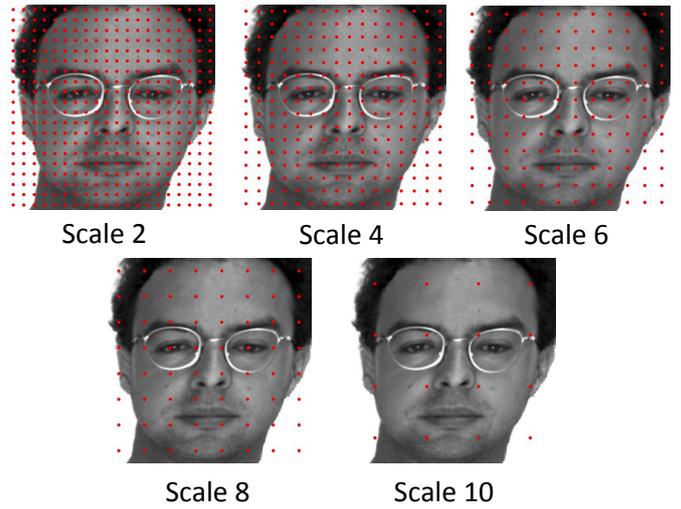


図3 scale ごとの Dense Sampling

坂野らの手法では投票処理においてハッシュベースの近似最近傍探索である Bucket Distance Hashing (BDH) [5] を用いることによって、照合を高速化する。BDH では、あらかじめデータベースの特徴空間を区切り、クエリ特徴量が与えられた場合、まず近い空間の探索を行う。そしてクエリ特徴量に近い空間内の特徴データのみと距離計算を行う。

2.3 Dense Sampling を用いた特徴抽出

低解像度に対応した坂野らの手法 [2] では、Dense Sampling を用いて、低解像度の画像からも認識に必要な一定量の特徴点を確保することで、認識性能の低下を防いでいる。特徴点は図3のように、ある一定の間隔で格子状にとる。特徴点の間隔は、特徴抽出を行う領域のスケールによって決定する。領域が広い場合は、特徴点間の距離を広くし、領域が狭い場合は特徴点間の距離を狭くする。その後、決定された特徴点のまわりの領域から PCA-SIFT 特徴量を抽出をする。決定された特徴量数は、一枚の画像あたり合計で 880 個である。図3の scale は特徴抽出を行う領域の大きさを示していて、scale の値が大きいほど特徴抽出を行う領域が広いことを示す。scale の大きいものは疎に、小さいものは密になるように特徴抽出する領域を決定する。決定された領域から特徴抽出することにより、低解像度の画像でも一定の特徴量を確保し、高精度の認識が行える。Dense Sampling ではデータベースに登録した顔画像とクエリ顔画像の全く同じ座標から特徴が抽出されるので、坂野らの手法 [2] では、同じ座標から得られた特徴とのみマッチングする。

3. 提案手法

前述のように Dense Sampling で特徴点を決定すると、メモリ使用量が膨大になる。そこで本稿では、認識処理において、周辺の局所領域との対応付く度合いを示す局所領域間の対応率を求め、それに基づいた特徴点の削減と、最適解を解くアルゴリズムである遺伝的アルゴリズムに基づく特徴点の削減を組み合わせることによって特徴点を選択し、認識性能をなるべく維持したままメモリ使用量を削減する。特徴量のマッチングの際、

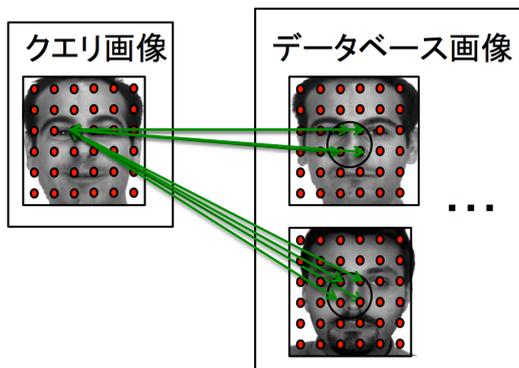


図 4 距離によるマッチングの特徴点の限定

同じ局所領域から得られた特徴量のみを検索の対象とするのではなく、周辺の特徴点から得られた特徴量とも検索をすることにより認識精度を向上させる。

3.1 表情変化に対応する方法

坂野らの手法 [2] で Dense Sampling を用いた場合は、同じ特徴点から抽出された特徴量のみを検索の対象としていた。しかし、提案手法では、2.2.1 で述べたように、同じ特徴点から抽出された特徴量のみではなく、クエリの特徴点を中心として一定距離内にある特徴点から得られる特徴量を全て探索の対象にする。これにより、表情変化に伴う顔パーツの位置ずれに対応する。

3.2 メモリ使用量の削減方法

提案手法では、特徴点間の対応率と遺伝的アルゴリズムを組み合わせることにより、特徴点の組み合わせの最適解を得る。対応率は図 5 のように左上の特徴点の対応率を求める場合、パラメータ調整画像の特徴量の最近傍が、データベース画像のどの特徴点から得られた特徴量になるかを表す。それぞれの位置に対応付いた数を調べて、その値が一定以上の特徴点が左上の特徴点と対応率の高い点となる。対応率を考慮せずに遺伝的アルゴリズムのみで特徴点の選択を行うと、対応率の高い点を選択されない可能性があり、その場合には認識性能の低下を招くと考えられる。そこで、ある特徴点が遺伝的アルゴリズムによって選択されたとき、選択された特徴点と、対応率が一定以上になる点を、追加で選択する。このように対応率の高い点を追加で選択することにより、顔パーツの位置ずれに対する頑健性を保ちつつ、メモリ使用量を削減できる。

以下では、本稿で用いた遺伝的アルゴリズムの詳細について述べる。遺伝的アルゴリズムでは、組み合わせ最適解の候補を遺伝子で表した個体を複数用意し、それぞれの個体を認識率で評価する。そして、評価を元に次世代の個体を決定する。次世代の作成方法は、2つの親個体を選択して遺伝子の組み替えを行う交叉と、交叉によって局所的な解が得られることを避けるために、遺伝子の一部を確率によって変化させる突然変異をする。また、評価の高い親個体を次世代でも使用するエリート戦略も行う。提案手法では、交叉、突然変異によって作成された次世代の個体に対して、対応率を考慮して特徴点を追加する。作成された次世代の個体を評価し、さらに次の世代を作成することを繰り返し、評価が最も大きくなる遺伝子の組み合わせを

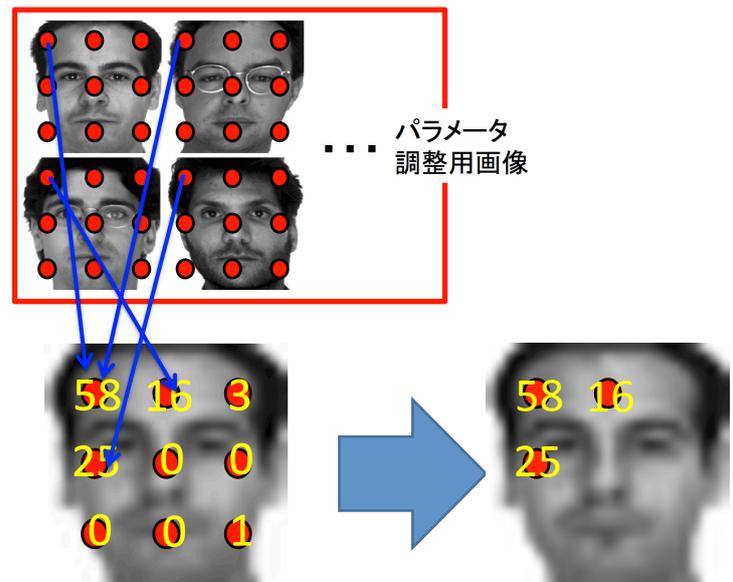


図 5 対応率の計算

見つけ出す。提案手法では、遺伝子の長さを、特徴点の数だけ用意し、遺伝子の要素を、特徴点を使用する場合は 1、使用しない場合は 0、として 0 と 1 で表す。この 0、1 で表した遺伝子を用いて評価、選択、突然変異を繰り返すことで、最も評価の high なる 0、1 の組み合わせを得る。

4. 実験

文献 [1] と [2] に記載されている 2 種類の坂野らの手法と提案手法の比較実験を 1 万枚と 100 万枚の顔画像を含むデータベースで行った。

4.1 実験条件

データベースは文献 [1] で使われているものを使用した。公開された顔画像データベースである AR Face Database [7] の Set 1 の画像と、ウェブ上からダウンロードした顔画像を合わせ、まず 1 万枚の顔画像データベースを作成した。1 万枚の顔画像データベースに Web 上からダウンロードした顔画像を追加し、100 万枚のデータベースを作成した。Web からダウンロードした画像は重複したものを除いており、全く同じ画像は存在しないが、同一人物の画像が複数枚含まれている場合もある。実験に使用した顔画像は、顔の切り出しを行い、目や鼻などの特徴点の自動検出に基づく顔パーツの正規化と顔向きでの正規化を行った [4]。その後、顔画像ごとの顔パーツの位置を合わせた [6]。そのため、正規化に失敗した画像もデータベースに含まれている。(ただし、AR Face Database の画像については、正規化に失敗した 2 枚を除いた) 正規化後の画像はすべてグレースケールで、解像度は 512×512 [pixel] である。実験で用いた、AR Face Database の画像例を図 6 に示す。AR Face Database の Set 1 -Set 7 はそれぞれ、neutral, smile, anger, scream, left side light on, right side light on, all side light on で構成されている。AR Face Database の Set 1 の画像のうち、正規化に失敗した画像を除いた 132 枚の画像をデータベー

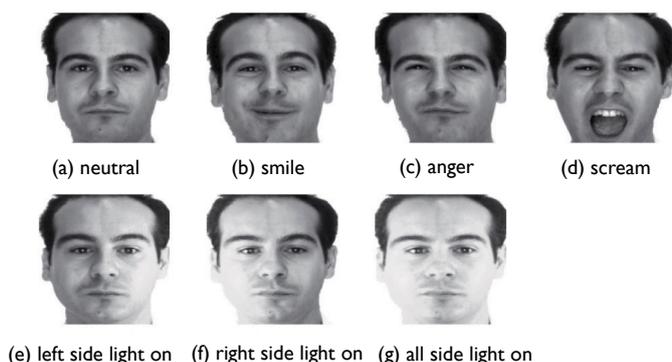


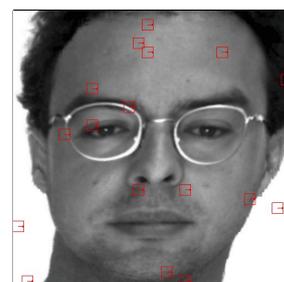
図 6 AR Face Database の画像例.

スに含み、クエリとして、Set 1 に対する Set 2 - Set 7 の画像 792 枚 (132 枚× 6 セット) を用いた。提案手法で学習をする際は、クエリとして用いる画像の半分 (66 枚× 6 セット) を学習画像、残りをクエリとした。その後、学習画像とクエリを入れ替えてもう一度評価を行い、2 つの結果の平均をとることで認識率を求めた。

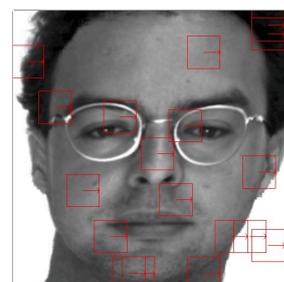
選択する特徴量の候補として、縦横 30 点を等間隔に Dense Sampling して得られた特徴点 900 点から scale の大きさが 2, 6, 8, 10, 20 で、PCA-SIFT を用いて特徴量抽出し、合計 4500 個の特徴点を得る。その後、4500 個の特徴量から提案する特徴量選択手法を用いて特徴量を選択し、PCA-SIFT の検出器を用いる坂野らの手法 [1] と Dense Sampling を用いた坂野らの手法 [2] と比較を行った。坂野らの手法 [1] では、特徴点検出により、一枚あたり平均で約 180 点の特徴点を得た。Dense Sampling を用いた坂野らの手法 [2] では、scale の大きさに 2, 4, 6, 8, 10 を用い、特徴点のサンプル間隔を 40, 80, 120, 160, 200 [pixel] とし、図 3 のように合計 880 点の特徴点を配置している。検索の際は、近似最近傍探索で 30 近傍を探索し、投票に用いた。認識結果は、全ての手法において 100 位累積認識率で求めた。遺伝的アルゴリズムを用いた手法でも同様に、検索の際は、全探索を行うのではなく近似最近傍探索を用いて処理時間を削減し、投票処理は上位 30 近傍に行った。遺伝的アルゴリズムでのパラメーターは、個体数は 10、世代数は 100、交叉率は 90%、突然変異率は 5% で、評価関数は認識率、次世代の選択の方法にルーレット選択を用いて一つの個体は前世代のエリートの個体を引き継いだ。実験に用いた計算機は、CPU が AMD Opteron (tm) Processor 6174 (2.2GHz)、メモリは 256 GB である。

4.2 結 果

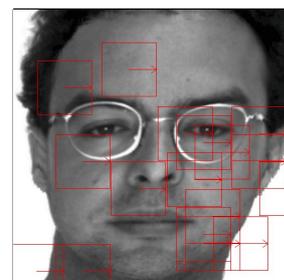
1 万枚データベースにおける、PCA-SIFT の検出器を用いる坂野らの手法 [1] と Dense Sampling を用いた坂野らの手法 [2] と提案手法の比較実験の結果を表 1 に示す。提案手法で特徴点を選択することにより、既存手法に比べて、表情変化、照明変化共に精度が向上した。画像一枚あたりの特徴点数を、坂野らの手法 [1] の平均約 180 点、Dense Sampling を用いた坂野らの手法 [2] の 880 点、と比べて提案手法では 94 点まで削減できた。これにより、Dense Sampling を用いた坂野らの手法 [2]



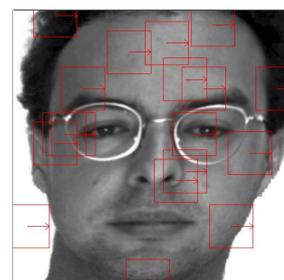
scale2



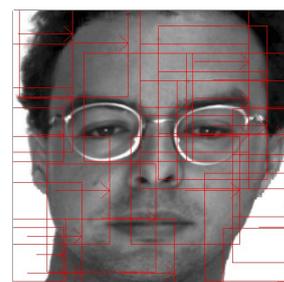
scale6



scale10



scale8



scale20

図 7 scale ごとの特徴点

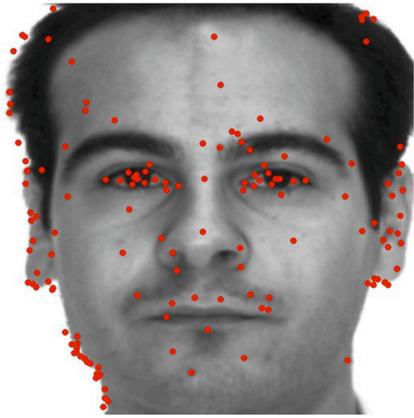


図 8 坂野らの手法 [1] での特徴点の一例

と比べて、メモリ使用量を約 512[GB] 削減することができた。提案手法によって選択された特徴点を中心として各 scale で特徴抽出を行う領域を図 7 に示す。提案手法では、図 8 の特徴点検出を用いた坂野らの手法 [1] での特徴点の一例のように、目や口の周り、顔の輪郭の特徴点が多く選択されている。これにより、提案手法では、少ない特徴点数で高い認識率を得ることができたと考えられる。

次に、100 万枚の大規模なデータベースにおける、坂野らの手法 [2] と提案手法の比較実験の結果を表 2 に示す。100 万枚のデータベースの結果と同様に、坂野らの手法 [2] と比較して、表情変化、照明変化共に提案手法で高い認識率を得られた。また、メモリ使用量も 1302[GB] 削減できた。この結果から、提案手法が大規模なデータベースで省メモリで表情変化、照明変化に頑健な認識ができることが示された。

5. ま と め

本稿では、対応率を考慮して遺伝的アルゴリズムで特徴点を選択することにより、Dense Sampling を導入することで生じるメモリ使用量と表情変化の問題を同時に解決する方法を提案した。これにより、低解像度顔画像や照明変化を受けた顔画像の認識性能が向上した。メモリ使用量も 1 万枚のデータベースで、坂野らの手法 [1] から 46.2[GB]、Dense Sampling を用いた坂野らの手法 [2] から 512[GB] 削減することができた。また、100 万枚の大規模なデータベースでも提案手法により、低解像度顔画像や照明変化を受けた顔画像の認識性能が向上したことから、大規模なデータベースにおける提案手法の有効性を示せた。

謝辞 本研究は、文部科学省の平成 25 年度社会システム改革と研究開発の一体的推進による「安全・安心な社会のための犯罪・テロ対策技術等を実用化するプログラム」の一環として実施された。

文 献

- [1] 内海ゆづ子, 坂野悠司, 前川敬介, 岩村雅一, 黄瀬浩一, “局所特徴量と投票処理を用いた大規模データベースに対する高速顔認識,” 電子情報通信学会論文誌 D vol.J197-D, no. 8, pp.1263–1272, 2014.
- [2] 内海ゆづ子, 坂野悠司, 前川敬介, 岩村雅一, 黄瀬浩一, “局所特

徴量と近似最近傍探索を用いた大規模データベースに対する高速顔認識,” 情報処理学会研究報告 vol.2013-CVIM-186, no. 4, pp.1-7, 2013.

- [3] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.2, pp.506–513, 2004.
- [4] M. Yuasa, T. Kozakawa, and O. Yamaguchi, “An efficient 3D geometrical consistency criterion for detection of a set of facial feature points,” IEICE Transaction on Information and Systems, vol.E91–D, no. 7, pp. 1871–1877, 2008.
- [5] Masakazu Iwamura and Tomokazu Sato and Koichi Kise, “What Is the Most Efficient Way to Select Nearest Neighbor Candidates for Fast Approximate Nearest Neighbor Search?”, Proceedings of 14th International Conference on Computer Vision, pp.3535–3542, 2013.
- [6] T. Kozakaya and O. Yamaguchi, “Face recognition by projection-based 3d normalization and shading subspace orthogonalization,” Proceedings of 7th International Conference on Automatic Face and Gesture Recognition, pp. 163–168, 2006.
- [7] A. Martinez and R. Benavente, “The AR face database,” CVC Technical Report #24, Computer Vision Center, Barcelona, 1998.

	表情変化			照明変化			メモリ使用量 (GB)
	smile (%)	anger (%)	scream (%)	left side light on (%)	right side light on (%)	all side light on(%)	
坂野らの手法 (特徴点検出) 特徴点数: 180 点	96.6	98.5	41.0	84.1	68.9	25.8	7.2
坂野らの手法 (Dense Sampling) 特徴点数: 880 点	96.7	98.5	36.4	100.0	96.7	67.4	53.8
提案手法 特徴点数: 94 点	96.7	98.5	47.7	98.5	97.7	90.9	2.56

表 1 1 万枚 DB での認識率, メモリ使用量の比較

	smile (%)	anger (%)	scream (%)	left side light on (%)	right side light on (%)	all side light on(%)	メモリ使用量 (GB)
坂野らの手法 (特徴点検出) 特徴点数: 180 点	96.2	96.2	33.3	97.7	95.5	24.2	194.3
提案手法 特徴点数: 94 点	96.2	96.2	43.6	96.2	96.5	33.3	64.0

表 2 100 万枚 DB での認識率, メモリ使用量の比較