C2-05

大規模データベースを用いた画像認識の実験的検討

Experimental Evaluation of Image Recognition Using a Large-Scale Database

氏原 慎弥† 野口 和人‡ 黄瀬 浩一‡ 岩村 雅一‡
Shinya Ujihara Kazuto Noguchi Koichi Kise Masakazu Iwamura

1. はじめに

近年，デジタルカメラやスキャナなどにより，パソコン上での画像を編集，Web 上での写真や絵の公開など，大量の画像を扱う機会が増えてきた．スキャナやデジタルカメラで取り込んだ画像に，関連情報を提示できれば便利である．例えば，飲食店の看板を撮影した時に，その店の場所や，メニューを表示するなどである．これらを受けて，画像を認識するという研究がなされている．

画像認識の手法の一つに，局所特徴量を利用したものがある．局所特徴量は，SIFT(Scale-Invariant Feature Transform)[1] や PCA-SIFT(Principal Component Analysis SIFT)[2] などを用いて抽出し，木構造や表構造などでデータベース化する．認識を行う際には，最近傍探索によりクエリとデータベースの特徴量同士の対応をとり，その結果を投票処理により統合することで回答を得る．具体的には，クエリから得られた各特徴量とデータベース内の特徴量との距離を計算する．距離の値が最小となった特徴量を抽出した画像の ID に，投票処理を行う．最終的に最も得票数の多くなった画像を，認識結果とする．

しかし，これらの手法の問題点は，データベースがより大規模となったときにも有効な手法であるとは必ずしも言えないことである．一般にデータベースが大規模になるとデータ量の増加により，メモリ使用量や認識速度，認識精度の問題が起こる．

そこで，本研究では，大規模データベースでの画像認識の比較実験を行った．実験は，50 万枚の画像をデータベースとして，PCA-SIFT とハッシュを利用した野口らの手法 [3] を用いて行った．実験の結果，認識率 98.2%，処理時間が 376.9[ms] が得られた．また，20 万枚の画像をデータベースとし，野口らの手法と ANN(Approximate Nearest Neighbor)[4] を用いた手法との比較実験も行った．比較実験では，ANN を用いた手法と比べて処理時間が約 1/5 となり，認識率は 0.3%程度低下することがわかった．

以下，2 章では，実験に用いた認識手法の説明を行い，3 章，4 章では実験の条件と結果の考察を詳しく述べ，最後に 5 章でまとめを述べる．

†大阪府立大学知能情報工学科
‡大阪府立大学大学院工学研究科

2. 画像認識の処理手順

以下に認識実験で使用した，野口らの手法と ANN について述べる．

2.1 野口らの手法

野口らの手法では，ハッシュ関数を用いてデータをハッシュ表に登録し，検索に用いる．ハッシュ表にデータを登録，検索する場合には，ハッシュ関数により処理を行うインデックスを知る必要がある．以下にハッシュ関数について説明する．PCA-SIFT で得られる 36 次元の実数値ベクトル $\boldsymbol{x}$ を，第 1 次元から第 d 次元 ($d \leq 36$) までとることにする．そして，特徴量の次元ごとに 2bit に量子化し，ビットベクトル $\boldsymbol{u} = (u_1, ..., u_d)$ を次式により作成する，

$$u_j = \begin{cases} 1 & \text{if } x_j - \mu_j \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

ここで μ_j は認識対象となる画像すべての x_j の平均値である．そして，

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

によってハッシュ値を求める．ここで H_{size} はハッシュ表のサイズである．

その後，特徴量と画像 ID をハッシュ表に登録する．画像 ID とは，画像と特徴量を対応付けるためのものである．データをハッシュに登録する際，衝突が生じると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．削除された特徴量は同じハッシュ値を持っているので，画像の認識にあまり役立たないと考えられる．

検索時にはクエリから，特徴量を抽出し量子化したのちハッシュ関数によって同じハッシュ値をもつ特徴量を求める．しかし，クエリの特徴量が撮影条件によって変動するため，対応する特徴量を得られないことがある．これでは十分な認識結果が得られなくなる．そこで値の変動幅 e を設け，変動に対処す

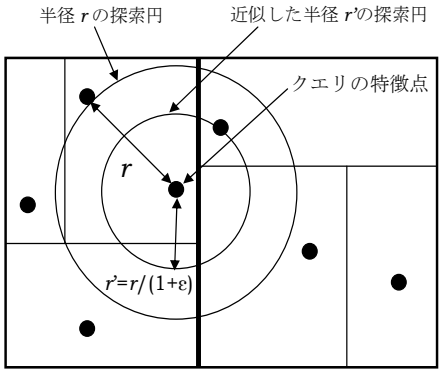

図 1: 各セルには特徴点が 1 つずつ入っている．探索円の半径を近似することで重なるセルが減り，距離計算の回数が減る

る．具体的には，クエリから得た特徴点 $\boldsymbol{q} = (q_1, ..., q_d)$ とするとき $|q_j - \mu_j| \leq e$ を満たす次元 j に対しては，u_j だけでなく $u'_j = (u_j + 1) \bmod 2$(0 ならば 1，1 ならば 0) も用いて，特徴量を検索する．しかし，2 つの処理をすべての特徴量に行うと莫大な計算時間が必要になる．そこで，処理対象となる次元数を b に制限する．$|q_j - \mu_j| \leq e$ を満たす次元の数が b より多いときには，対象とする次元を高次の方から b 個とする．

画像の認識時には，各特徴点ごとに，画像 ID に投票処理を行いう．そして，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．

2.2 ANN を用いた手法

ANN を用いた手法でも，PCA-SIFT で特徴量を記述する．ANN は kd-tree(k-dimensional tree) という木構造で表現される．各特徴点を空間上に配置し，セルと呼ばれる領域にそれぞれ 1 つずつ存在するようになるまで領域を分割する．その様子を図 1 に示す．探索時には二分木として働き，ある 1 つのセルまで下ると，自身とセル内の点との距離を半径 r とする探索円を描く．そして探索円と重なるセル内の点との距離計算を行う．ここで，より高速な探索を行うために，探索円の許容誤差 ε を考える．この ε を使い探索円の半径を次式のように計算しなおす．

$$r' = r/(1 + \varepsilon) \tag{3}$$

式 (3) のように半径を近似することで，探索時に重なるセルを減らし高速な検索を行うことができる．ε を大きくすれば探索時間は減るが，近似しすぎると精度が落ちてしまう．

以上の処理を野口らの手法と同様に，各特徴点ごとに行い，結果を投票処理することで画像の認識を実現する．

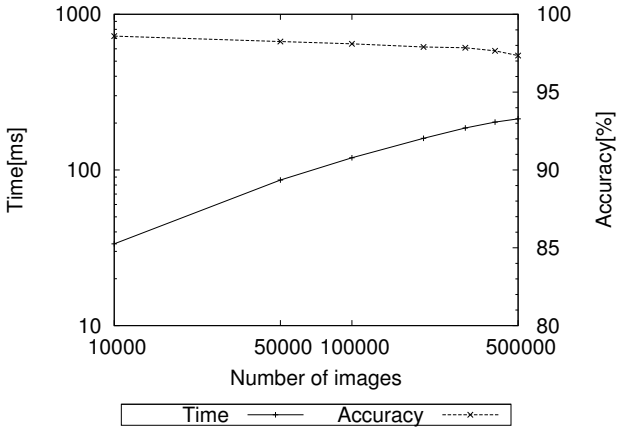

図 2: 登録画像数と認識率，処理時間の関係

3. 実験条件

10 万枚のデータベースで有効であった野口らの手法や ANN が，さらに大規模なデータベースでも有効であるとは限らない．そこで大規模データベースでの認識率を比較するための実験を行った．以下に具体的な実験内容について説明する．

実験には，野口らの手法では最大 50 万枚，ANN ではメモリ容量の都合により，最大 20 万枚の画像を使用した．使用した画像は，写真共有サイト flickr から集めた．収集時の条件は，animal や flower といったタグや，flickr にアップロードされた日時を用いた．図 3 に使用した画像の例を示す．また，サイズが 399×399 画素以下の画像は除外し，残った画像は画像の長辺が 640 画素になるよう縮小した．画像の特徴量を PCA-SIFT で抽出し，特徴点が 200 個以上検出されなかった画像を除外した．残った特徴点からデータベースを作成した．

集めた画像の中から 500 枚選び，それぞれを，正面から，60 度から，75 度から，画像の一部のみを撮影した合計 2000 枚をクエリとして使用した．

使用した計算機は，CPU が AMD Opteron 2.8GHz，メモリ 64GB のものである．

4. 結果・考察

まず野口らの手法の結果を示す．登録画像数による認識率と処理時間を比較するため，文献 [3] で使用していたパラメータ (b=10，c=10，d=28，e=400) を使い実験を行った．登録画像数を 1 万枚から 50 万枚まで変化させた結果を図 2 に示す．画像登録時に，リスト全体がハッシュ表から削除された回数と，画像 1 枚あたりの，リスト内での距離計算の回数を表 1 に示す．グラフと表より，登録枚数が 50 万枚のときと，登録枚数 10 万枚のときの結果を比較すると，処理時間が約 2 倍になっ

図 3: 使用した画像の例

表 1: 登録枚数を変化させたときのリスト削除回数，距離計算の回数，認識率 [%]，処理時間 [ms]

登録枚数	リスト削除回数	距離計算の回数	認識率 [%]	処理時間 [ms]
100000	819,171	341,388	98.1	115.3
200000	2,562,903	514,006	97.9	159.7
300000	5,129,126	630,933	97.9	186.1
400000	8,401,943	709,117	97.7	203.0
500000	12,241,007	759,669	97.4	216.5

ていることがわかる．これは距離計算の回数が，2 倍程度しか増えなかったために，処理時間もそれほど増えなかったのではと考えられる．また，表より登録枚数が増えるほど，リスト削除回数とリスト間での距離計算回数が増えていることがわかる．これは似通った特徴量が多く現れたためだと考えられる．

次に b と e の関係を確認するための実験を 50 万枚のデータベースで行った．文献 [3] で使用していたパラメータ (c=20,d=28 とし，e と b を変化させた) での結果を図 4 に示す．b=12 のところを見れば，e=400 と比較して，e=200 の処理時間が 1/10 以上速くなっており，認識率 96.8%が得られることがわかった．

次にリスト削除の閾値が，結果にどのような影響を及ぼすのか確認するための実験を，50 万枚のデータベースで行った．文献 [3] でのパラメータ (b=10，d=28，e=400 として c を変化させた) でのグラフを図 5 に示す．また，そのときのリスト削除回数と，画像 1 枚あたりのリスト内での距離計算の回数を表 2 に示す．c=50 のとき処理時間が 376.9[ms] となり，認識率は 98.2%を得た．c を大きくし過ぎても，処理時間が増えるばかりで認識率は上がらなかった．表とこの結果から，同じハッシュ値をもつ特徴点は，認識にはあまり役立たないということがわかる．

最後に，野口らの手法と ANN を用いた手法を比較した．20 万枚のデータベースで，パラメータを野口らの手法では b=10，c=10，20，50，d=28，e=200，400 として実験を行った．ANN では ε=1000，100，50，10，5 として実験を行った．図 6 に結果のグラフを示す．b=10，c=50，d=28，e=400 のパラメータで行ったものの処理時間が，ANN での約 1/5 となっている

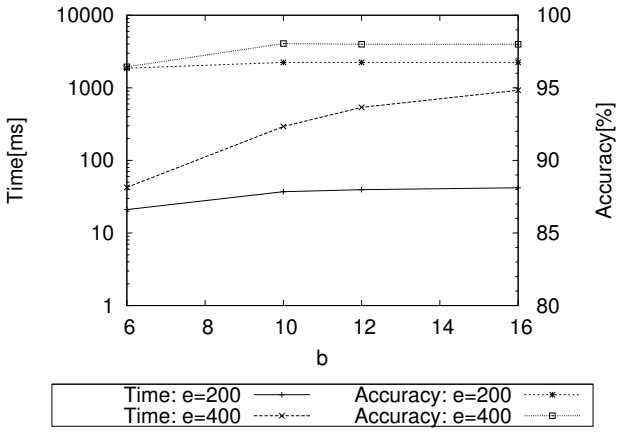

図 4: b,e の認識率，処理時間の関係

ことがわかった．さまざまなパラメータでの実験の結果を表 3 に示す．野口らの手法を使用した結果では，変動幅 e が小さくとも，c が大きければ高い認識率を得られるということがわかった．だがこれは処理時間がかかるということでもある．また，画像枚数が増えれば似通った特徴量が多く出現するということだと考えられる．

5. まとめ

本稿では認識に有効な野口らの手法と ANN について述べ，大規模データベースによる認識実験を行い，結果の比較を行っ

表 2: 閾値の変化による，リスト削除の回数，距離計算の回数，認識率 [%]，処理時間 [ms]

リスト削除の閾値 c	10	20	30	40	50	60	70	80	90	100
リストが削除された回数	12,241,007	3,456,716	1,654,222	991,390	669,340	487,186	372,864	296,033	241,662	201,672
距離計算の回数	759,669	1,152,596	1,360,018	1,497,325	1,599,653	1,680,933	1,748,508	1,806,037	1,855,900	1,900,074
認識率 [%]	97.4	98.1	98.0	98.2	98.2	98.2	98.2	98.2	98.2	98.2
処理時間 [ms]	216.5	293.6	335.9	357.7	376.9	389.9	401.9	415.7	421.5	429.5

表 3: 各手法の認識率 [%] と処理時間 [ms]

手法	パラメータ	60 度		75 度		90 度		一部		平均	
		精度	時間	精度	時間	精度	時間	精度	時間	精度	時間
野口らの手法	b=6,c=10,d=28,e=600	89.0	19.5	96.2	23.3	96.2	23.7	96.4	58.9	94.5	31.3
	b=10,c=20,d=28,e=400	96.4	173.4	98.4	211.6	98.8	216.1	98.6	573.2	98.1	293.6
	b=10,c=50,d=28,e=400	96.6	224.0	98.6	271.9	98.8	733.7	98.8	278.0	98.2	376.9
ANN	ε =50	75.8	12.3	94.6	13.6	95.2	13.7	95.4	28.0	90.3	16.9
	ε =10	96.2	95.4	98.2	103.1	98.4	103.6	98.4	284.6	97.8	146.7
	ε =5	98.0	605.3	98.4	629.2	99.0	628.4	99.0	1882.2	98.6	936.2

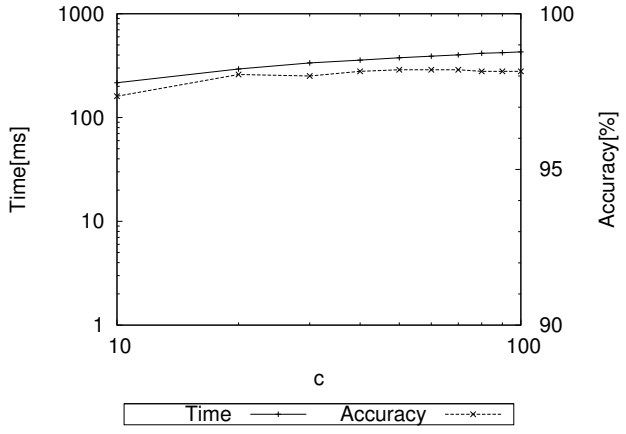

図 5: c の認識率，処理時間の関係

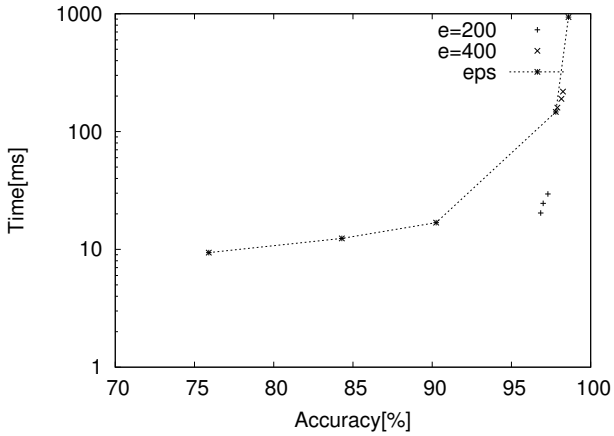

図 6: ANN の認識率，処理時間の関係

た．

野口らの手法では，50 万枚のデータベースで認識率が 98.4% を得られた．そしてある程度の近似を行っても，高い認識率が得られることがわかった．しかし，処理時間が多くかかっていることから，現在の手法をそのまま使用することができないこともわかった．ANN を用いた手法との比較では，処理時間が約 1/5 でかつ，認識率も高精度であることがわかった．また処理時間と，認識率はトレードオフすることもわかった．

参考文献

[1] D. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol.60, no.2, pp.91–110, 2004.

[2] Y. Ke and R. Sukthankar, PCA-SIFT: A more distinctive representation for local image descriptors, Proc. of CVPR2004, Vol. 2, pp.506–513, 2004.

[3] 野口和人，黄瀬浩一，岩村雅一，“局所記述子に基づく物体認識のためのメモリ削減の実験的検討，” 画像の認識・理解シンポジウム (MIRU2008) 論文集，pp.251–258，July, 2008.

[4] S. Arya, D. M. Mount, R. Silverman and A. Y. Wu, “An optimal algorithm for approximate nearest neighbor searching,” Journal of the ACM, vol.45, no.6, pp.891–923, 1998.