社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

局所特徴量の大規模照合に基づく物体インスタンスの検索

TRECVID2012 Instance Search タスク参加報告

阪田　智大† 　的崎　伸彰† 　黄瀬　浩一† 　岩村　雅一†

† 大阪府立大学大学院工学研究科　〒 599–8531 大阪府堺市中区学園町 1–1
E-mail: {sakata,matozaki}@m.cs.osakafu-u.ac.jp, {kise,masa}@cs.osakafu-u.ac.jp

あらまし　我々が TRECVID 2012 の Instance Search タスクに参加した結果を報告する．我々の研究では，BoF のような手法を用いず局所特徴量を直接マッチングすることで，画像中の物体位置や背景の変化に対応できるという局所特徴量の利点を残し，それにより高い認識率を目指す．この際，データ量が膨大なためマッチングに時間がかかるという問題があるが，ハッシュに基づく近似最近傍探索手法を利用することにより，この問題に対処する．本手法を利用し，最大で 18.2% の Mean Average Precision となり，全 24 チーム中 4 位という結果を得ることができた．本稿では，このような単純な手法で何がどこまで認識できるのかを，実験データに基づいて明らかにするとともに，限界とそれを克服するための方策についても考察する．

キーワード　TRECVID，Instance Search，特定物体認識，近似最近傍探索

1. はじめに

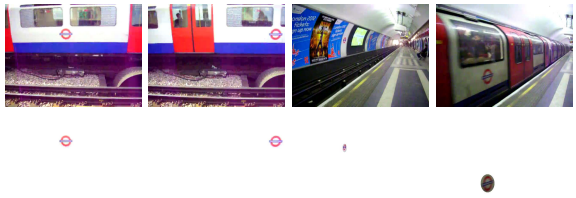
TRECVID(TREC Video Retrieval Evaluation) と呼ばれる，映像コンテンツの内容解析および検索の高度化を目的とした競争型国際プロジェクトが NIST(National Institute of Standards and Technology) 主催で開催されている．このプロジェクトでは，世界中から研究グループの参加を募り，数百時間規模の大規模映像アーカイブに対する同一のタスクに挑戦し，その結果を比較評価することで研究水準の向上を図っている．TRECVID 2012 には 6 つのタスクが存在する．我々は，その中のタスクの一つである Instance Search(INS) タスクに参加した．このタスクでは，数枚の画像で与えられている目標物がどの映像中に含まれているかを検出することを目的としている．これは，映像を対象とした特定物体認識のタスクである．

特定物体認識では，局所特徴量のマッチングによって高い精度で認識できることが知られている．また，局所特徴量のマッチングには，画像中の物体の位置や背景の変化にも強いという利点があり，INS タスクにおいてもこの利点は非常に有利であると言える．しかし，INS タスクでは用意されたデータ量が膨大であり，局所特徴量そのままでのマッチングを行うことを考えたときに，メモリ容量や処理時間などの問題が存在している．そのため，INS タスクにおいては, [1] のように局所特徴量から動画の Bag-of-Features(BoF) 表現を作成することで，データ量と比較回数を減少させることが多い．また，BoF などを使わずに直接マッチングを行う場合でも, [2] のように画像を縮小してから特徴抽出したり，先に検索対象を絞ることで，比較回数を減らすような措置を取る．

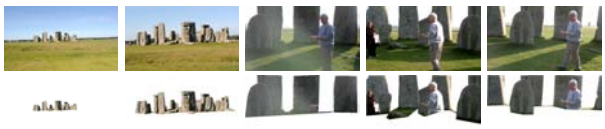
このような従来法により，一定の結果が得られているが，幾つかの問題点が挙げられる．まず，BoF を特定物体認識で用いる場合，Visual Word の数を相当数増やさなければ認識率が落ちてしまうことが知られており [3]，現状用いられている Visual Word の数が十分ではないという問題である．Visual Word の数を増やせばいい話ではあるが，それにより処理時間やメモリ量が増加することから，データ量の多い条件下ではなかなか実現されにくい．次に，BoF に変換することで局所特徴量の物体位置や背景の変化に強いという利点が損なわれるという問題である．最後に，直接マッチングの場合でも，画像縮小などではどうしても情報量が少なくなってしまい，認識率が落ちてしまう．また，先に検索対象を絞ってからマッチングをする場合でも，検索対象を絞る時点では局所特徴量の利点は得られず，この部分が精度の低下を招きやすい．これらの問題点が，従来法の認識率を低下させている．

そこで，我々は局所特徴量の単純なマッチングに基づく手法によって高精度な認識を実現することを考える．前述したその際に生じるメモリ容量や処理時間といった問題には，ハッシュに基づく近似最近傍探索手法 [4] を利用することで対処する．我々の使用する近似最近傍探索手法は，ハッシュを利用することで高速な処理を実現しており，局所特徴量の各次元をスカラ量子化することにより，メモリ削減も可能としている．

このような手法を利用して INS タスクに取り組むことで，我々は最大で 18.2% の Mean Average Precision(MAP) を獲得し，いくつかのクエリにおいて参加チームの中で最高の結果であった．これは全 24 チーム中 4 位の成績であり，局所特徴量のマッチングに基づく手法の INS における有効性を示すことができたと言える．しかし，結果を分析したところ，殆ど認識できないクエリも存在することが分かった．本稿ではその理由についても考察する．その結果，認識できていないクエリは物体領域，背景ともにマッチングがうまくいっておらず，特徴抽



(a)OBJECT(ロンドン地下鉄のロゴ)



(b)LOCATION(ストーンヘンジ)

(c)PERSON(Stephen Colbert)

図 1　クエリの例 (上:全体画像，下:物体領域のみの画像)

出を工夫する必要があることがわかった．

2. Instance Search(INS) タスク

本節では TRECVID2012 での INS タスクの詳細について説明する．

2.1 データベース作成用映像

データベース作成用の映像としては，Flickr video available under Creative Commons licenses for research の映像を使用する．これらの映像を最長で 10 秒の短い映像に分割したものを用い，それらの映像を shot と呼ぶ．クエリの物体が複数の shot の中にある程度の回数出現している．shot は全部で 74,958 件あり，それぞれに対し shot ID が与えられている．このうち，1232 件の shot が正解を含んでいる．また，全映像の合計時間は約 200 時間となっている．

2.2 クエリ

図 1 に与えられたクエリの例を示す．まず，クエリとして 21 種類の事物が用意されており，それぞれのクエリに対し，3 から 6 枚の画像が与えられている．これらの画像には，探したい物体の領域の情報も同時に与えられており，図の上段が全体の画像，下段が物体領域のみを切り出した画像となっている．これらを用いてデータベースからクエリの事物を検索する．クエリトピックは PERSON(人物)，OBJECT(物体)，LOCATION(場所) の 3 つのいずれかに分類されている．これらの分類ごとに別の手法を用いて検索することも可能である．2012 年のタスクでは，PERSON が 1 種，OBJECT が 15 種，LOCATION が 5 種となっている．また，クエリ一つに対しての正解は 5 件から 295 件と様々な数になっており，平均で 58.7 件の正解が存在している．

2.3 タスク

このタスクでは，画像で与えられる視覚的な事例をもとに，人・物体・場所などの具体的な事物を蓄積映像から検出することを目的としており，動画像における特定物体認識と言える．結果は，先に述べたクエリそれぞれに対し，そのクエリが含まれると推定した shot の上位 1000 件を順位付きで求める．そして，全クエリの上位 1000 件を求め，処理時間と一緒にまとめたものを 1 つの Run として提出する．この Run は 1 チームにつき 4 つまで提出することが可能である．これらの Run の評価には，Avarage Precision (AP) と，その AP の算術平均である Mean AP(MAP) を用いて評価を行う．

3. 関連研究

TRECVID2012 の INS タスクにおける上位の他のチームの手法を簡単に紹介する．これらの手法の詳細は，TRECVID のホームページ [5] を参照されたい．

まず，北京郵電大学のチームの手法を述べる．HSV ヒストグラム，SIFT の BoF，Gabor Wavelet，HOG などの 9 種類の特徴量を利用している．これらの特徴量それぞれで類似度を求めた上で，それらを重み付け線形結合することで結果を得る．そして，得られた結果の上位 10 件の動画をクエリとして再検索をすることで結果の更なる改善をする．

続いて，北京大学のチームの手法を述べる．特徴量として，CMG(Color Moment Grid) や LBP(Local Binary Pattern)，そして，SIFT，Color SIFT，Opponent SIFT の BoF 表現を用いる．これらの特徴量のマッチングには multi-bag SVM を利用し，上位 1000 件を得る．得られた上位 1000 件の動画に対して SIFT のマッチングをすることで，順位を決定する．また，得られた順位の中で，半教師あり学習に基づくアルゴリズムにより，リランキングを行う．

次に，香港城市大学のチームの手法を述べる．まず，特徴量には SIFT の BoF 表現のみを用いている．また，Hamming Embedding と Multiple Assignments を利用することで BoF の精度を向上させている．これらの BoF を用い，Delaunay Triangulation に基づいてマッチングを行い結果を得ている．

続いて，NTT-NII チームの手法について述べる．特徴量には，192 次元の Color SIFT を用いる．これらをコサイン尺度でマッチングし，値が一定以上のものを特徴量がマッチングしたものと見なす．これらを用いて，動画毎の特徴量の出現頻度などを求め，投票に重み付けをすることで良い結果を出している．投票の重み付けには，BM 25 アルゴリズムなどの重み付けを用いている．

最後に，チリ大学のチームの手法について述べる．特徴量には，SIFT や Color SIFT を利用する．そのまま利用するには特徴ベクトルの数が多すぎるため，フレーム画像を縦の幅が一定になるように縮小したものから特徴量を抽出する．そして，クエリの特徴ベクトルそれぞれに対し，50 近傍を探索し，それらによる投票を行なっている．また，探索には，Amazon Elastic Compute Cloud を利用した分散コンピューティングを用いる．

傾向として，複数の特徴量を利用しそれらを組み合わせているものが比較的高い結果を残している．

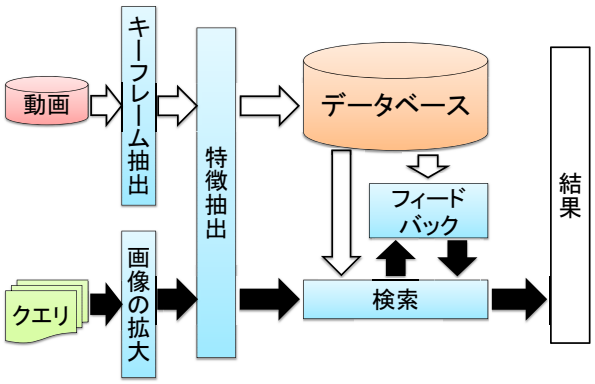


図 2　手法の流れ

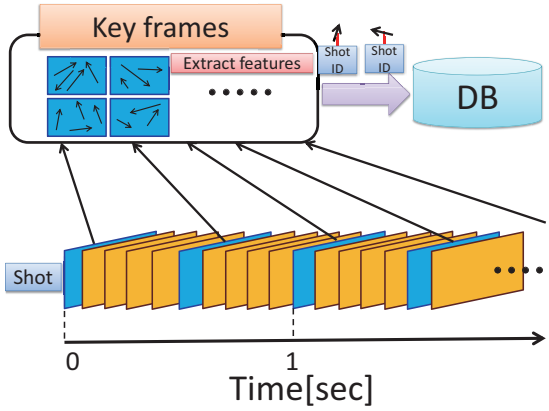


図 3　データベース作成の流れ

4. 手　　法

本節では INS タスクに使用する手法について説明する．

手法の流れを図 2 に示す．白の矢印が参照動画から得た情報の流れ，黒の矢印がクエリから得た情報の流れを示す．まず，動画からキーフレームを抽出し，続いてそこから特徴抽出をしてデータベースに登録する．クエリ画像中の目標物体が小さい場合があるため，物体領域から得られる特徴量の数が認識に十分でないことがある．そこで，物体領域から得られる特徴量を増やすためにクエリ画像を拡大してから特徴量を抽出する．続いて，抽出した特徴量を用いてデータベースにアクセスし，検索を行い，検索結果を得る．しかし，クエリ画像の枚数が少なく，物体に関する情報が十分に得られていないため，それを補う必要がある．そこで，検索結果の上位には正解が含まれているものと考え，検索結果の上位をフィードバックしてもう一度検索をかけることで情報の不足を補い，最終的な結果を得る．

はじめに，キーフレームの抽出について述べ，次にクエリ画像の拡大，検索手法のベースとなるハッシュに基づく近似最近傍探索手法 [4]，最後に，フィードバック法について説明する．

4.1　キーフレーム抽出

キーフレーム抽出から DB 作成までの流れを図 3 に示す．蓄積映像の規模が大きいため，すべてのフレームから特徴抽出を行いデータベースに登録することはメモリ容量等の観点から現実的ではない．また，動画像では，直近のフレームは殆ど同じ画像となることが多いため，ある程度のフレームを捨てても問題ない．そこで，各動画から秒間 2 フレームを取り出しキーフレームとし，そこから特徴抽出を行う．

4.2　クエリ画像の拡大

INS タスクのクエリ画像は解像度が低いことや物体領域が小さいなどの理由で十分な数の特徴ベクトルが得られないことがある．そこで，元のサイズのクエリ画像から抽出した特徴ベクトルだけでなく，クエリ画像を 2 倍，3 倍に拡大し，そこから抽出した特徴ベクトルもそのクエリの特徴量として扱う．画像の拡大には Lanczos 法を使用する．

4.3　ハッシュに基づく近似最近傍探索手法 [4]

ここでは，認識に用いる手法であるハッシュに基づく近似最近傍探索手法について説明する．この手法は，ハッシュを利用して高速に探索を行う近似最近傍探索手法であり，特徴ベクトルの各次元ごとにスカラー量子化を行うことで，メモリ削減を実現している．はじめにデータ登録について述べたあと，検索方法について述べる．

4.3.1　データ登録

まず，各 shot から抽出したキーフレームから d' 次元の特徴ベクトル $\boldsymbol{x} = (x_1, x_2, \cdots, x_{d'})$ を抽出する．その特徴ベクトル $\boldsymbol{x}$ の第 1 次元から第 d 次元 $(d \leqq d^{'})$ までに対して，

$$u_j = \begin{cases} 1 \text{ if } x_j - \theta_j \geqq 0, (0 \leqq j \leqq d) \\ 0 \text{ otherwise,} \end{cases} \tag{1}$$

を適用することで，各次元を 2 値化したビットベクトル $\boldsymbol{u} = (u_1, u_2, \cdots, u_d)$ に変換する．ここで θ_j は，ハッシュ表に登録する特徴ベクトルすべての x_j の平均値である．そして，

$$H_{\text{index}} = \left(\sum_{i=0}^{d-1} u_i 2^i \right) \mod H_{\text{size}} \tag{2}$$

によってハッシュ値を求め，shot ID とスカラー量子化によってデータ量を削減した特徴ベクトルを組にしてハッシュ表に登録する．ここで，H_{size} はハッシュ表のサイズであり，2^d となる．

登録時に衝突が生じる場合，リストとして複数の特徴ベクトルを追加登録する．このとき，多数の衝突が起こると，検索時の距離計算の回数が増加してしまう問題が生じる．また，ID は異なるが同じビットベクトルとなる特徴ベクトルが多い場合，それらの特徴ベクトルは類似しており，認識するのに十分な識別性能を保持していないと考えられる．そこで，これらの問題への対処として，リストとして登録される特徴ベクトルに対応する ID の種類に上限を設け，上限を超える場合はリスト全体をハッシュ表から削除する．そして，以降そのハッシュ値への登録を禁止する．以上の処理をキーフレームから抽出される全ての特徴ベクトルに対して行うことにより，データの登録が完了する．

4.3.2　検　　索

まず，q に対するハッシュ値を計算し，それに基づいてハッシュ表を参照，得られた特徴ベクトルの集合を $\boldsymbol{X}$ とする．次に，$\boldsymbol{q}$ をスカラー量子化したベクトルと $\boldsymbol{X}$ に含まれるベクトルとのユークリッド距離を計算し，上位 k 近傍となる特徴ベク

トルの集合 $\boldsymbol{X}_*$ を求める．そして，$\boldsymbol{X}_*$ に対応する shot ID にそれぞれ投票する．これは，INS では一つのクエリに対して複数の正解が存在するため，一部の正解に票が集まるのを防ぐためである．ここで，shot 毎に抽出される特徴ベクトルの数 C_s は異なるため，特徴ベクトルの数が多い shot には投票される確率が高くなる．また，投票する k 近傍の特徴ベクトルの数が増えてくると，正解以外の shot に投票される確率が高くなる．そこで，第 k 近傍に投票する際に $(0.95)^{k-1}/\sqrt{C_s}$ の重みを付けて投票を行う．さらに，物体領域から抽出した特徴ベクトルに対しては，背景から抽出した特徴ベクトルと比べて，n 倍の重みを付ける．クエリのすべての特徴ベクトルに対してこの処理を行い，最終的に最も得票数の多いものを回答とする．

クエリから得られた特徴ベクトルを用いて探索を行う際，特徴ベクトルの各次元の値が撮影条件によって変動するため，multiprobing によって対応する．具体的には，値の変動幅 e をパラメータとして，変動への次の対処を施す．$\boldsymbol{q} = (q_1, ..., q_d)$ とするとき，$|q_j - \mu_j| \leqq e$ を満たす次元 j に対しては，u_j だけではなく $u'_j = 1 - u_j$ も用いて，特徴ベクトルを検索する．これを b 個の次元に対して行うことで，2^b 個のハッシュ値を用いてハッシュにアクセスすることになる．

4.4 フィードバック法

上記の手法を用いて検索を行った結果，上位の動画には目標物が映っている可能性が高い．よって，そこから対象物に関する特徴ベクトルが新たに得られる．その特徴ベクトルは，元のクエリの特徴ベクトルとは異なるため，それを新たにクエリとして用いると，さらに別の shot が検索できる可能性がある．そこで，検索結果の上位 r 個の動画から抽出した特徴ベクトルもクエリとして用いることで，より高精度な検索を目指す．このとき，上位の動画から抽出した特徴ベクトルはデータベースに登録されている．そのため，検索を行う際，検索に用いる特徴ベクトルと同じ特徴ベクトルが第 1 近傍とされてしまう．それによって，上位の動画に票が集中してしまうという問題が生じる．これを防ぐために，フィードバックの際には，第 1 近傍となる特徴ベクトルに対応する動画 ID には投票しないようにする．また，上位の動画の中には，目標物が映っていない動画も含まれている．そのため，これらの動画から抽出した特徴ベクトルによる投票が集中し，誤検出を引き起こすことがある．この影響を緩和するために，上位の動画から抽出した特徴ベクトルによる投票の際には，$m(< 1)$ 倍の重みを付ける．

4.5 OpponentSIFT 特徴量 [6]

本節では実験に用いる OpponentSIFT 特徴量について述べる．特徴点検出には Harris Laplace detector [7] を用いる．この検出器を用いることでスケール変化に頑健な特徴点を得ることができる．Harris Laplace detector により得られた特徴点から色情報を含む OpponentSIFT 特徴量を抽出する．まず，画像を RGB 色空間から以下の式によって Opponent 色空間 [8] に変換する．

表 1　実 験 条 件

	フィードバック	クエリ拡大		
		1 倍	2 倍	3 倍
*IMP.h_f_e1	✓	✓		
*IMP.h_f_e2	✓	✓	✓	
IMP.h_e1		✓		
*IMP.h_e2		✓	✓	
*IMP.h_e3		✓	✓	✓

*TRECVID2012 に提出したもの

表 2　評 価 結 果

	MAP[%]
*IMP.h_f_e1	18.17
*IMP.h_f_e2	18.10
IMP.h_e1	17.37
*IMP.h_e2	18.23
*IMP.h_e3	17.42

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix} \tag{3}$$

ここで得られたチャンネル O_3 は HSV 色空間の明度に等しく，O_1 と O_2 はそれぞれ赤と緑，黄色と青の反対色の組の情報を保持している．そして，O_1 から O_3 のチャンネルごとに SIFT 記述子を用いて特徴抽出をすることで 128 次元× 3 チャンネル= 384 次元の色情報を持つ特徴量を得る．この特徴量動画中の物体検出に有効とされている特徴量である．この特徴量を主成分分析により次元数を 60 次元に削減して使用する．なお，OpponentSIFT 特徴量の抽出には ColorDescriptor software v3.0 [9] を用いる．

5. 実 験 条 件

実験を行う際の条件を表 1 に示す．フィードバックの有無とクエリ拡大のサイズのみを変更しており，それ以外の条件については，全て同様のパラメータを用いた．以下に各パラメータの値を示す．

- ハッシュに利用する次元数 $d = 32$
- 投票処理で投票する近傍数 $k = 20$
- 物体領域の特徴量の重み $n = 5$
- ビットの反転数 $b = 10$
- フィードバックに上位 $r = 10$shot を利用
- フィードバックの投票の重み $m = 0.1$

6. 実験結果と考察

それぞれの run の結果を表 2 に示す．IMP.h_e2 が 18.23% と最も良い MAP が得られた．この結果は全 79Run 中 10 位，全 24 チーム中 4 位と比較的上位に位置しており，局所特徴量のマッチングによる手法は INS タスクにおいて有効だと言える．また，フィードバックによる MAP の向上は見られなかった．クエリ拡大は 2 倍の時に最も良く，3 倍まで拡大すると MAP が低下した．これは，ある程度の倍率までは拡大することで得

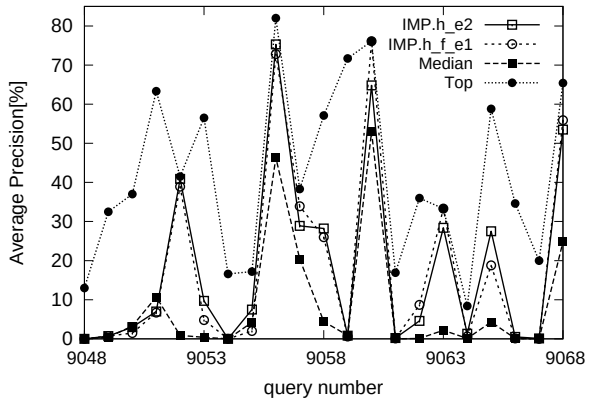
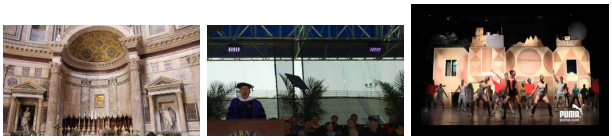


図 4　クエリ毎の結果



(a) 結果が良いクエリ (左から 9056，9060，9068)

(b) 結果が悪いクエリ (左から 9048，9054，9067)

図 5　クエリの例

られる有益な情報がノイズよりも多いが，一定以上の倍率ではノイズの方が多くなってしまうためだと考えられる．

フィードバックの有りと無しのそれぞれの Run で MAP が高かったもののクエリ毎の AP を図 4 に示す．また，クエリ毎の中央値やトップの AP も示している．全体的に中央値よりも高い AP が得られており，9052(ロンドン地下鉄のロゴ) や 9063(プラハ城) では，全 Run 中で最高の AP を獲得した．ここでも，フィードバックの有無による変化はあまり見られないことから，フィードバックがあまり結果に影響を与えていないことがわかる．

図 5 に結果の良かったクエリと悪かったクエリの例を示す．図 5(a) のクエリはクエリ番号 9056(パンテオン内部)，9060(人物)，9068(PUMA のロゴ) であり，高い MAP を得ることができた．パンテオンの建物内部ように，背景が少なく全体が写っているようなクエリは多数の特徴量が得られるため，マッチングの際に正解が得やすく，良い結果が得られたと考えられる．また，他の 2 つでは，同じような背景の動画が正解として存在しており，背景の特徴量のマッチングにより正解が得られていると考えられる．そのため，正解を含む背景の異なる動画が存在する場合，それらを認識することは難しいと考えられる．現状では物体領域が小さく，そこから得られる情報が少なくあまり認識に有効でないため，特徴抽出を工夫することで得られる情報を増やしたり，少ない情報だけでマッチングできるよう改良する必要がある．

図 5(b) のクエリは殆ど認識出来ていなかったものの例であり，

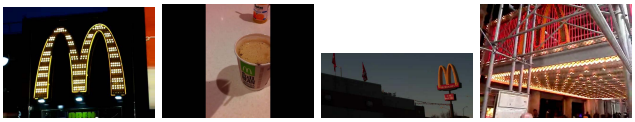


図 6　マクドナルドのマーク

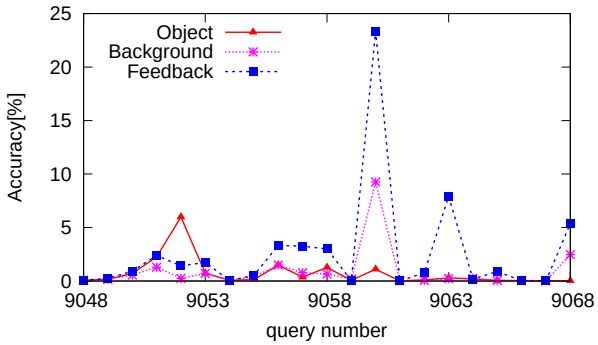


図 7　クエリ毎の投票の正解率

9048(メルセデスのマーク)，9054(ストーンヘンジ)，9067(マクドナルドのマーク) などがある．まず，メルセデスのマークのように物体領域が非常に小さいクエリについては，正解となる動画の背景が全く異なるものが多い．このようなものについては，物体領域のマッチングが難しく，背景の比率も高くなるため誤投票が増えてしまい認識出来ていないと考えられる．また，ストーンヘンジの岩や草原の部分は様々な動画に出現しており，ユニークな特徴量が取れないことから，近傍点を得てもそれが正解でないという問題が起こる．最後に，マクドナルドのロゴに関しては，図 6 のように物体自体の模様が変化しており，このようなものから同等の特徴量を取り出すことは難しい．これらの認識出来なかったクエリに関して同様に言えることは，特徴抽出を改良しなければ認識自体が難しいということである．

6.1　投票の正解率

投票時の投票先の正解率を図 7 に示す．物体領域，背景領域の特徴量，フィードバックの投票で別々に正解率を出している．フィードバックで正解への投票が行われているということは，物体領域と背景領域からの最初の投票で得た結果に正解が含まれているということである．そのため，図 7 からも分かるように，物体領域や背景領域の投票で正解へと投票できていないものに関してはフィードバックの正解率も低い．一方で，物体領域や背景領域からの正解率が数 % あるものは，フィードバックの正解率も高く，正解の動画が得られていることがわかる．フィードバックでの正解率は物体領域や背景領域からの投票の正解率よりも高いものが多いが，フィードバックによる全体での MAP の向上が見られなかったことから，新たな正解への投票が多いのではなく，既に得られている正解への投票が多かったものと考えられる．

また，物体領域からの投票では 9052(ロンドン地下鉄のマーク) が他に比べて正解率が高いことがわかる．9052 は，図 1(a) を見ると，物体領域が小さいクエリであるが，物体領域からの特徴量がユニークであれば，問題なく認識できることがわかる．そのため，他の物体領域が小さいクエリに関しても，ユニーク

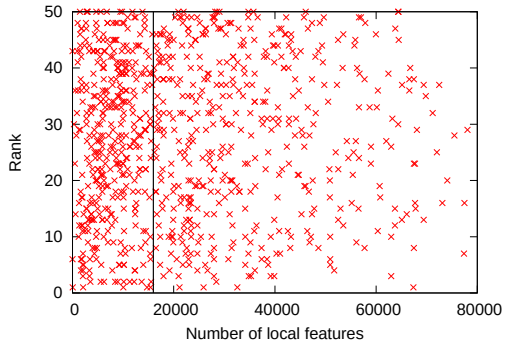


図 8　上位の shot の特徴点数の分布

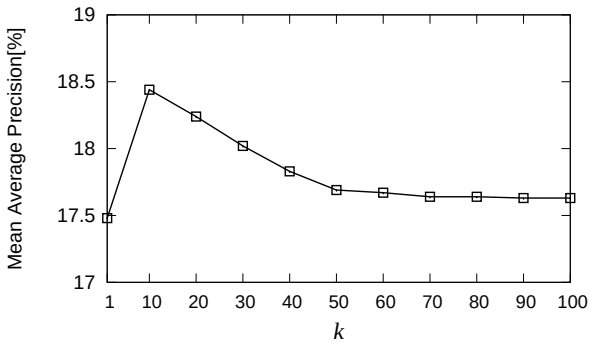


図 9　k による MAP の変化

な特徴量が得られるような特徴抽出法を考える必要がある．

背景領域からの正解率が高いものとして 9060 が挙げられる．これは，正解のうちの殆どが同じ動画を分割したものであったことから背景が似通っており，正解に多く投票できたものである．逆に物体領域から得た特徴量の正解率は低く，背景が異なるクエリが与えられた場合には認識は難しいと考えられる．

6.2　特徴点数による重み付けの影響

図 8 は IMP.h_f_e2 の各クエリにおける上位 50 件の中で不正解の shot の特徴点数の分布を表したものである．縦の線が全 shot の特徴点数の平均値である．この図より，特徴点数の低い shot が 50 位以内に多いことがわかる．これは，投票時の特徴点数による重み付けにより，少ない投票数でも特徴点数の少ない shot が上位にくるためである．このような問題を回避する手段として，重み付けによる格差を緩和することや，ピボット正規化 [10] といった手法が考えられる．

6.3　k の値による MAP の変化

INS タスクでは，1 つのクエリに対して多くの正解となる shot が存在しているため，最近傍のみに投票するだけでは他の正解を捨ててしまうことになる．そのため，1 つの特徴量からどれだけの近傍に投票するかというのは重要である．そこで，投票処理で投票する近傍の数 k の値を変化させたときの認識率の変化を調べた．その結果を図 9 に示す．10 近傍に投票したときが最大の結果となり，それ以降は少しずつ下がっていくという結果となった．20 近傍以上では，少しずつ認識率が下がってしまっている．これは，1 つの特徴量が投票する先の正解は上位 10 件までに多く，それ以上まで増やしてもノイズとなるためである．現状では 10 近傍までの投票で問題がないが，今後用いる特徴量を変化させた場合には，k の値を増加させた方がよい可能性がある．その場合，クエリの正解数が少ないにも関わらず，多数のノイズに投票してしまうことになるため，クエリや特徴量毎に k を変えるような手法を考えなければならない．そのような手法の例としては，投票先の特徴量との距離に閾値を設けることや，投票先の分散率を考慮することが考えられる．

7.　ま　と　め

我々は，TRECVID 2012 の INS タスクに参加した．我々の手法は，ハッシュに基づく近似最近傍探索により，局所特徴量を直接マッチングするものである．この手法を用いた結果，MAP は 18.23% で全 79Run 中 10 位，全 24 チーム中 4 位という成績を達成した．この結果から，我々の手法の映像中の特定物体認識における有効性が確認できたと言える．しかしながら，現状では殆ど認識できていないクエリも存在し，それらは特徴抽出手法を改良しなければ認識することは難しいと考えられる．

今後の課題としては，特徴抽出手法の改良や，投票時の重み付け方法の改良，投票数を自動決定する手法の導入などが挙げられる．

謝辞

本研究の一部は科学研究費補助金基盤研究 (B)(22300062) の補助による．

文　　献

[1] Z. Zhao, Y. Zhao, Y. Hua, W. Wang, D. Wan, G. Jia, Z. Li, F. Su and A. Cai: “Bupt-mcprl at trecvid 2012”, TRECVID 2012 Workshop Notebook (2012).

[2] J. M. Barrios and B. Bustos: “Instance search based on parallel approximate searches”, TRECVID 2012 Workshop Notebook (2012).

[3] D. Nistér and H. Stewénius: “Scalable Recognition with a Vocabulary Tree” (2006).

[4] K. Kise, K. Noguchi and M. Iwamura: “Robust and efficient recognition of low-quality images by cascaded recognizers with massive local features”, Proceedings of the 1st International Workshop on Emergent Issues in Large Amount of Visual Data (WS-LAVD2009), pp. 2125–2132 (2009).

[5] TREC video retrieval evaluation. http://trecvid.nist.gov/

[6] K. E. A. van de Sande, T. Gevers and C. G. M. Snoek: “Color Descriptors for Object Category Recognition”, European Conference on Color in Graphics, Imaging and Vision, pp. 378–381 (2008).

[7] K. Mikolajczyk and C. Schmid: “Scale & Affine Invariant Interest Point Detectors”, Int. J. Comput. Vision, **60**, pp. 63–86 (2004).

[8] J. van de Weijer and T. Gevers: “Boosting saliency in color image features”, Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 1 - Volume 01, CVPR ’05, Washington, DC, USA, IEEE Computer Society, pp. 365–372.

[9] K. E. A. van de Sande, T. Gevers and C. G. M. Snoek: “Evaluating color descriptors for object and scene recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, **32**, 9, pp. 1582–1596 (2010).

[10] A. Singhal, C. Buckley, M. Mitra and A. Mitra: “Pivoted document length normalization”, Proc. SIGIR, ACM Press, pp. 21–29 (1996).