IPSJ Transactions on Computer Vision and Applications Vol.5 153–162 (Aug. 2013)

[DOI: 10.2197/ipsjtcva.5.153]

Research Paper

Large-scale Image Retrieval as a Classification Problem

Yusuke Uchida[1,a)] Shigeyuki Sakazawa[1]

Received: October 10, 2012, Accepted: April 30, 2013, Released: August 23, 2013

Abstract: In this paper, we propose a new, effective, and unified scoring method for local feature-based image retrieval. The proposed scoring method is derived by solving the large-scale image retrieval problem as a classification problem with a large number of classes. The resulting proposed score is based on the ratio of the probability density function of an object model to that of a background model, which is efficiently calculated via nearest neighbor density estimation. The proposed method has the following desirable properties: (1) has a sound theoretical basis, (2) is more effective than inverse document frequency-based scoring, (3) is applicable not only to quantized descriptors but also to raw descriptors, and (4) is easy and efficient in terms of calculation and updating. We show the effectiveness of the proposed method empirically by applying it to a standard and improved bag-of-visual words-based framework and a k-nearest neighbor voting framework.

Keywords: specific object recognition, bag-of-visual words, approximate nearest neighbor search, Hamming embedding, product quantization, naive-bayes nearest-neighbor

1. Introduction

With the advancement of both stable interest region detectors [1] and robust and distinctive descriptors [2], local feature-based image or object retrieval has attracted a great deal of attention. Particularly, it has become applicable to large-scale databases with a bag-of-visual words (BoVW) framework [3]. **Figure 1** illustrates the standard framework of a BoVW-based image retrieval system. In the BoVW framework, local feature points or regions are detected from an image, and feature vectors are extracted from them. These feature vectors are quantized into visual words (VWs) using a visual codebook (visual vocabulary), resulting in a histogram representation of VWs. Image similarity is measured by ℓ_1 or ℓ_2 distance between the normalized histograms. As VW histograms are generally sparse, an inverted index data structure and a voting function enable an efficient similarity search. The equivalency between ℓ_2 distances and scores obtained with the voting function is described in Ref. [4] in detail. A weighting scheme based on inverse document frequency (IDF) [3], [5] is integrated with the voting function to improve performance. Finally, geometric verification [6] is performed to refine and re-rank the results obtained with the voting function.

Although the BoVW framework realizes efficient retrieval, there is some room for improvement in terms of accuracy. One significant drawback of VW-based matching is a hard-assignment problem: two features are considered to be matched if and only if they are assigned to the same VW [4]. There are two major extensions of VW-based matching designed to alleviate this problem: post-filtering approaches [4], [7] and multiple assignment (or soft assignment) approaches [4], [8]. In post-filtering approaches, after VW-based matching, unreliable matches are filtered out according to (estimated) distances between query and reference features. In multiple assignment approaches, query features are matched not only with reference features assigned to the nearest VW but also with reference features assigned to the k-nearest VWs.

In this paper, in order to solve these problems, we propose a new, unified scoring method applicable to many conventional frameworks. The optimal scoring method is derived by solving the large-scale image retrieval problem as a classification problem with a large number of classes. The proposed score is based on the ratio of the probability density function of an object model to a background model, which is efficiently calculated in an on-the-fly manner via nearest neighbor density estimation. In experiments, we show the effectiveness and versatility of the proposed scoring method by applying it to a BoVW framework and a k-nearest neighbor voting framework. This paper is the extended version of the paper [9] that appeared in ICPR 2012. Particularly, we conduct detailed experiments in terms of the parameter in the proposed method and discuss the effect of normalization term in the proposed score. Furthermore, the proposed method is thoroughly compared with original classification method and state-of-the-art methods.

2. Improving BoVW-based Image Retrieval

In this section, an overview is provided of related work which improves BoVW-based image retrieval in terms of feature matching and the problems associated with conventional methods are outlined at the end of this section.

There are two major approaches used to improve the performance of BoVW-based image retrieval in terms of feature matching: post-filtering approaches and multiple assignment approaches. Post-filtering and multiple assignment respectively contributes to the precision and recall of feature matching, and

[1] KDDI R&D Laboratories, Inc., Fujimino, Saitama 356–8502, Japan
[a)] ys-uchida@kddilabs.jp

© 2013 Information Processing Society of Japan

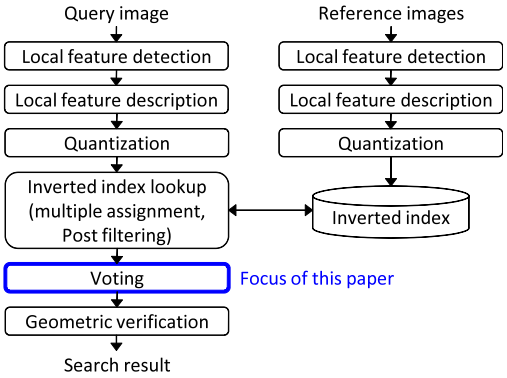


Fig. 1 Standard framework of local feature-based image retrieval system.

they are complementary. Both approaches are reviewed in this section.

Several methods have also been proposed with the intention of improving the performance of BoVW-based image retrieval other than the approaches mentioned above such as the utilization of spatial information [4], [10], [11], [12], [13], query expansion approaches [14], [15], [16], and re-ranking based on k-nearest neighbors [13], [17]. These methods are complementary to the proposed method and not discussed in detail in this paper.

2.1 Post-filtering Approaches

As the naive BoVW framework suffers from many false matches of local features, post-filtering approaches are proposed to suppress unreliable feature matches [4]. In this section, an overview of post-filtering approaches to improve naive VW-based image retrieval is presented. There are two important components of post-filtering approaches: distance estimation and filtering criteria.

2.1.1 Distance Estimation

As mentioned previously, after VW-based matching, distances between a query feature and reference features that are assigned to the same visual word as the query feature are estimated for post-filtering. As distance calculations between an original query feature vector and original reference feature vectors are undesirable in terms of computational cost and memory requirement to store raw reference feature vectors [18], short code based methods are used for this purpose. Feature vectors are encoded into short codes and distances between feature vectors are approximated by distances between the short codes. This approach improves time efficiency and reduces memory requirement in the distance calculation. Two different distance estimation methods are described below.

- Hamming embedding (HE): In Refs. [4], [19], feature vectors extracted from reference images are encoded into binary codes (typically 32–128 bit codes) via random orthogonal projection followed by thresholding for binarizing projected vectors. While all VWs share a single random orthogonal matrix, each VW has individual thresholds so that feature vectors are binarized into 0 or 1 with the same probability. These codes are stored in an inverted index with image identifiers (sometimes with other information on the features [19]). In a search step, after VW-based matching, Hamming distances between codes of query and matched reference features are calculated. Matched features with larger Hamming than a predefined threshold are filtered out, as this considerably improves the precision of matching with only slight degradation of recall.
- Product quantization (PQ): In Ref. [18], a product quantization-based method is proposed and shown to outperform other short codes like spectral hashing (SH) [20] or a transform coding-based method [21] in terms of the trade-off between code length and accuracy in approximate nearest neighbor search. In the PQ method, a reference feature vector is decomposed into low-dimensional subvectors. Subsequently, these subvectors are quantized separately into a short code, which is composed of corresponding centroid indices. The distance between a query vector and a reference vector is approximated by the distance between a query vector and the short code of a reference vector. Distance calculation is efficiently performed with a lookup table. Note that the PQ method directly approximates the Euclidean distance between a query and reference vector, while the Hamming distance obtained by the HE method only reflects their similarity.

2.1.2 Filtering Criteria

Based on the estimated distances described above, unreliable feature matches are filtered out. There is room for discussion on how to utilize the distances. To date, several criteria are used for filtering.

- Distance criterion: The most straightforward way is to filter out reference features with larger (approximated) distances than the predefined threshold [4], [22].
- Rank criterion: The alternative is to use the k-nearest neighbor features in voting and to filter out the others [18]. In this case, for each feature vector in a query image, reference features are sorted according to distances between the query feature and the reference features in ascending order, and corresponding top-k reference features are used in voting.
- Ratio criterion: In the same way as the rank criterion, for each feature vector in a query image, distances between the query feature and reference features in the same VW are sorted in ascending order. Then reference features in the top-p percentile are used in voting [7].

2.2 Multiple Assignment Approaches

While post-filtering approaches try to improve the precision of feature matches with only slight degradation of recall, multiple assignment approaches improve recall at the cost of the precision of feature matches. The basic idea here is, at a search step, to assign a query feature not only to the nearest VW but to the several nearest VWs. This technique alleviates the problem of quantization error; sometimes, similar features are assigned to different VWs. In Ref. [8], each query feature is assigned to the fixed number of the nearest VWs and the influence of a matched feature to image similarity is weighted according to the distance between the query feature and the assigned VWs. In Ref. [4], the distance d_0 to the nearest VW from a query feature is used to determine the number of multiple assignments, where the query feature is assigned to the VWs such that the distance to the VWs

© 2013 Information Processing Society of Japan

is smaller than αd_0 ($\alpha = 1.2$ in Ref. [4]). This approach adaptively changes the number of assigned VWs according to the ambiguity of the feature. As post-filtering approaches and multiple assignment approaches are complementary, it is desirable to use multiple assignment in conjunction with post-filtering.

2.3 Scoring of Feature Matches

After feature matching as described above, scores of feature matches should be determined to vote the scores to the corresponding reference images. The square of the IDF value associated with the corresponding VW is used as a base score [3]. At the same time, weighting terms related to distance metrics and filtering criteria are considered for further improvement [4], [7], [23], [24].

In Ref. [23], the weight is calculated as a Gaussian function of a Hamming distance between the query and reference vector. In Ref. [4], the weight is calculated based on the Hamming distance between the query and reference vector and the probability mass function of the binomial distribution. In Ref. [24], the weight is calculated based on rank information because a rank criterion is used in post-filtering in the literature, while in Ref. [7], the weight is calculated based on ratio information. Thus, as mentioned before, these scoring methods have been specialized to certain frameworks, i.e., they depend on distance metrics and filtering criteria. Hence, they require trial-and-error processes when being applied to different frameworks. In addition, because these scoring methods have little theoretical basis and are not optimal, they result in unsatisfactory performance compared with their potential. Therefore, a comprehensive scoring method is needed that has a theoretical basis and is applicable to any frameworks without having to consider and try many different scoring methods.

3. Proposed Approach

In this section, in order to solve the problems mentioned before, a unified scoring method is proposed. The proposed scoring method is derived by solving the large-scale image retrieval problem as a classification problem with a large number of classes, which can be applied to many existing frameworks. We first present the probabilistic formulation of the proposed scoring method, starting with a classification problem similar to Ref. [25]. Then, in order to make it applicable to large-scale image retrieval, an approximation is introduced. Finally, the detailed formulation of the proposed score is obtained via non-parametric density ratio estimation.

3.1 Probabilistic Formulation

Given a query image Q, the objective is to find a similar image $R_{\hat{j}}$ from a large number of reference images $R_1, \cdots, R_m$. In this paper, we consider images to be similar if they share a same object [26]. Considering it as a classification problem, we start with maximum-a-posteriori estimation: $\hat{j} = \arg\max_j p(R_j|Q)$. Assuming $p(R_j)$ is uniform, the maximum-a-posteriori estimation reduces to a maximum likelihood estimation:

$$\hat{j} = \arg\max_j p(R_j|Q) = \arg\max_j p(Q|R_j). \tag{1}$$

Letting $\mathcal{Q} = \{q_1, \cdots, q_n\}$ denote the descriptors of the query image Q, with the naive Bayes assumption, we get:

$$p(Q|R_j) = p(q_1, \cdots, q_n|R_j) = \prod_{i=1}^{n} p(q_i|R_j). \tag{2}$$

As pointed out in Ref. [25], if we assume all query descriptors are derived from only the object model O_j of R_j, $p(Q|R_j)$ tends to be too small even if Q and R_j share the same object. In Ref. [25], the problem is alleviated by estimating $p(q_i|R_j)$ using a few dozen images representing the same class. As this is not practical for large-scale image or object retrieval, we directly model $p(q_i|R_j)$ by a mixture of the object model O_j of R_j and a background model B distinct from O_j:

$$p(q_i|R_j) = \lambda p(q_i|O_j) + (1-\lambda)p(q_i|B). \tag{3}$$

If we deal with only the quantized version of descriptors (visual words), this is identical to language modeling (LM) [5] in the area of information retrieval (IR). Combining Eqs. (1)–(3), we obtain:

$$\begin{aligned}\hat{j} &= \arg\max_j \prod_{i=1}^{n} p(q_i|R_j) = \arg\max_j \sum_{i=1}^{n} \log p(q_i|R_j) \\ &= \arg\max_j \sum_{i=1}^{n} \log(\lambda p(q_i|O_j) + (1-\lambda)p(q_i|B)) \\ &= \arg\max_j \sum_{i=1}^{n} \log\left(\frac{\lambda}{1-\lambda}\frac{p(q_i|O_j)}{p(q_i|B)} + 1\right).\end{aligned} \tag{4}$$

Finally, we get the voting score s_{ij}:

$$s_{ij} = \log\left(\frac{\lambda}{1-\lambda}\frac{p(q_i|O_j)}{p(q_i|B)} + 1\right). \tag{5}$$

For each q_i, the voting score s_{ij} is assigned to each R_j. The resulting score $s_j = \sum_i s_{ij}$ corresponds to the similarity measure between Q and R_j.

3.2 Approximation with Nearest Neighbors

In the above formulation, it is necessary to calculate s_{ij} for all R_j. Similarly, $\min_{r\in\mathcal{R}_j} \|q_i - r\|^2$ should be calculated for all R_j in Ref. [25], where $\mathcal{R}_j$ denotes a set of descriptors of R_j. These processes involve finding nearest neighbors of q_i from $\mathcal{R}_j$ for each R_j. Letting n_C denote the number of classes and n_D denote the average number of descriptors in an image, the calculation of s_{ij} for all R_j has a time cost of $O(n_C \cdot n_D)$ if brute-force search is adopted. This can be accelerated from $O(n_C \cdot n_D)$ to $O(n_C \cdot \log n_D)$ by using efficient approximate nearest neighbor search algorithms [18], [27], [28] which can find approximate nearest neighbors of q_i from n_D descriptors in $O(\log n_D)$. The computational cost does not become a fatal flaw in classification problems where $n_D \gg n_C$. However, it is intractable in the large-scale image retrieval problem where $n_C \gg n_D$ because n_C corresponds to the number of images or objects in a database.

In order to make it tractable, the following simple approximation is adopted. We assume the set of nearest neighbor descriptors $\mathcal{N}(q_i)$ of q_i (e.g., k-nearest neighbors of q_i) was obtained against all reference descriptors. Then, $p(q_i|O_j)$ is calculated only for R_j at least one of whose descriptors appears in $\mathcal{N}(q_i)$, and otherwise we assume $p(q_i|O_j) = 0$. Because the voting score s_{ij}

© 2013 Information Processing Society of Japan

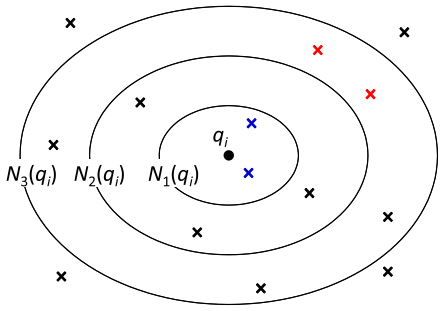


Fig. 2 An example of multiple levels of neighbors $\mathcal{N}_1(q_i), \cdots, \mathcal{N}_k(q_i)$ of q_i in the feature space ($k = 3$).

becomes 0 if $p(q_i|O_j) = 0$, the voting is performed efficiently. With this approximation, the computational cost is reduced from $O(n_C \cdot \log n_D)$ to $O(\log n_C \cdot n_D)$ because approximate nearest neighbors of q_i are searched from $n_C \cdot n_D$ descriptors only once instead of searching nearest neighbors from n_D descriptors and repeating n_C times.

3.3 Non-parametric Density Ratio Estimation

Finally, the voting score s_{ij} is calculated using $\mathcal{N}(q_i)$. As shown in **Fig. 2**, we assume that multiple levels (subsets) of neighbors $\mathcal{N}_1(q_i), \cdots, \mathcal{N}_k(q_i)$ of q_i are defined, which satisfy

$$\mathcal{N}_1(q_i) \subset \mathcal{N}_2(q_i) \subset \cdots \subset \mathcal{N}_k(q_i) = \mathcal{N}(q_i), \tag{6}$$

where $\mathcal{N}_1(q_i)$ is the most fine-grained level of neighbors of q_i and $\mathcal{N}_k(q_i)$ is the most coarse-grained level of neighbors of q_i. An intuitive and practical example of $\mathcal{N}_t(q_i)$ is the t nearest neighbor descriptors of q_i. For each $\mathcal{N}_t(q_i)$ ($1 \le t \le k$), and for each R_j one of whose descriptors appears in $\mathcal{N}_t(q_i)$, the densities $p(q_i|O_j)$ and $p(q_i|B)$ in Eq. (5) are estimated via k-nearest neighbor density estimation:

$$p(q_i|O_j) = \frac{|\mathcal{N}_t(q_i)|_j}{|\mathcal{R}_j| \cdot V_t}, \quad p(q_i|B) = \frac{|\mathcal{N}_t(q_i)|_{\text{all}}}{|\mathcal{R}_{\text{all}}| \cdot V_t}. \tag{7}$$

where $|\mathcal{N}_t(q_i)|_j$ is the number of descriptors of R_j that appear in $\mathcal{N}_t(q_i)$, $\mathcal{R}_{\text{all}}$ is all reference descriptors $\bigcup_j \mathcal{R}_j$, V_t is the volume of a hypersphere with radius $\sqrt{\|q_i - \hat{r}_t\|^2}$, and $\hat{r}_t \in \mathcal{N}_t(q_i)$ is the farthest descriptor from q_i. Combining Eqs. (5) and (7), we obtain the score s^t_{ij} for $\mathcal{N}_t(q_i)$:

$$s^t_{ij} = \log\left(\frac{\lambda}{1-\lambda}\frac{|\mathcal{N}_t(q_i)|_j/|\mathcal{R}_j|}{|\mathcal{N}_t(q_i)|_{\text{all}}/|\mathcal{R}_{\text{all}}|} + 1\right), \tag{8}$$

where the normalization terms about the numbers of descriptors in a query image and reference images are naturally considered as $|\mathcal{R}_j|$ and $|\mathcal{R}_{\text{all}}|$. We adopt t such that it maximizes s^t_{ij} for each q_i and R_j:

$$s_{ij} = \max_{1 \le t \le k} s^t_{ij}. \tag{9}$$

If we adopt fixed $t = k$, the information of the multiple levels of neighbors $\mathcal{N}_1(q_i), \cdots, \mathcal{N}_k(q_i)$ is discarded and only the information of the most coarse-grained level of neighbors $\mathcal{N}_k(q_i)$ is used, resulting in the degradation of accuracy. For instance, in Fig. 2, we assume that the two blue cross marks represent the descriptors of R_1 and the two red cross marks represent the descriptors of R_2. Intuitively, an inequality $p(q_i|O_1) > p(q_i|O_2)$ holds. If we adopt the definition of Eq. (9), $t = 1$ is selected for R_1 and $t = 3$ is selected for R_2, resulting in distinctive scores $s_{i1} > s_{i2}$. However, if fixed $t = 3$ is used for both of R_1 and R_2, s_{i1} becomes equal to s_{i2}. The effect of Eq. (9) is empirically evaluated in Section 5.2.

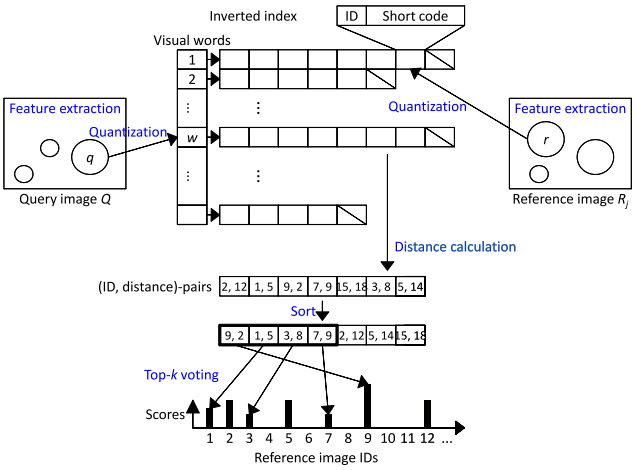


Fig. 3 Overview of the improved BoVW framework based on product quantization.

More concrete examples of the formulation are shown in the following section. One advantage of this scoring method is that an up-to-date score is efficiently calculated in an on-the-fly manner using $\mathcal{N}(q_i)$, even if the database is modified. The only requirement is to store the number of descriptors $|\mathcal{R}_j|$ in each reference image.

4. Implementation Details

Although the proposed scoring method described in Section 3 can be applied to many conventional frameworks by appropriately defining the nearest neighbors $\mathcal{N}_1(q_i), \cdots, \mathcal{N}_k(q_i) = \mathcal{N}(q_i)$ of q_i, we mainly apply the proposed method to one of the state-of-the-art frameworks (described in Section 2) which improves the BoVW framework by having distance estimation based on product quantization [7], [18]. In this section, we describe the framework evaluated in the experiments in detail. **Figure 3** provides an overview of the framework.

4.1 Feature Detection and Description

From query and reference images, a set of feature vectors is extracted. We adopt Hessian-Affine [29] and SIFT [30] as the feature detector and descriptor, respectively. The software [*1] provided by the authors of Ref. [1] is used in the experiments. We denote the i-th feature vector of the query image by $q_i \in Q$ and the h-th feature vector of the j-th reference image by $r_{jh} \in \mathcal{R}_j$.

4.2 Feature Indexing with Product Quantization

We adopt a product quantization-based method [7], [18] to improve the BoVW framework, namely IVFADC. In the indexing (off-line) step in IVFADC, a reference vector r_{jh} with d dimension [*2] is quantized with a coarse quantizer in the same way as the BoVW framework. We refer to the codebook used in coarse quantization as the CQ codebook. This is the same as what is

[*1] http://www.featurespace.org/
[*2] In the case of SIFT vectors, $d = 128$.

© 2013 Information Processing Society of Japan

referred to as visual words or a visual codebook in the context of BoVW-based image retrieval or recognition.

In the indexing step, a reference vector r_{jh} is first quantized into $c_{\hat{a}}$ using the CQ codebook C with k' centroids $c_1, \cdots, c_{k'} \in \mathbb{R}^d$, where

$$\hat{a} = \arg\min_{1 \le a \le k'} \|r_{jh} - c_a\|^2. \tag{10}$$

Subsequently, the residual vector $\bar{r}_{jh}$ from the corresponding centroid $c_{\hat{a}}$ is calculated as

$$\bar{r}_{jh} = r_{jh} - c_{\hat{a}}. \tag{11}$$

Then, the residual vector $\bar{r}_{jh}$ is decomposed into u subvectors $\bar{r}^1_{jh}, \cdots, \bar{r}^u_{jh} \in \mathbb{R}^{d^*}$, where $d^* = d/u$. Subsequently, these subvectors are quantized separately using u codebooks $P_1, \cdots, P_u$, which is referred to as product quantization. In this paper, a codebook used in product quantization is referred to as a PQ codebook. We assume that each PQ codebook P_l has k^* centroids $p_{l1}, \cdots p_{lk^*} \in \mathbb{R}^{d^*}$. Using the l-th PQ codebook, the l-th subvector $\bar{r}^l_{jh}$ is quantized into p_{lb_l}, where

$$b_l = \arg\min_{1 \le b \le k^*} \|\bar{r}^l_{jh} - p_{lb}\|^2. \tag{12}$$

Finally, the short code $(b_1, \cdots, b_u)$ is stored in the $\hat{a}$-th list of the inverted index with the identifier j of the reference image. The size of the short code is represented by $u \log_2 k^*$ bits.

4.3 Distance Calculation in IVFADC

In the search step in IVFADC, a query vector q_i is first quantized using the CQ codebook, and the residual vector $\bar{q}_i$ from the corresponding centroid is calculated in the same manner as the indexing. Subsequently, the distance between the residual vector $\bar{q}_i$ and short codes $(b_1, \cdots, b_u)$ in the corresponding list in the inverted index are calculated. These distances correspond to the approximate distances between the query vector q_i and the reference vectors r_{jh}:

$$d(q_i, r_{jh}) = d(\bar{q}_i, \bar{r}_{jh}) \approx \sqrt{\sum_{l=1}^{u} \|\bar{q}^l_i - p_{lb_l}\|^2}. \tag{13}$$

This distance calculation is performed efficiently using a lookup table T, which is precomputed when a query vector q_i is given:

$$T_{lb} = \|\bar{q}^l_i - p_{lb}\|^2 \ (1 \le l \le u,\ 1 \le b \le k^*). \tag{14}$$

Using the table, Eq. (13) is rewritten as

$$d(q_i, r_{jh}) \approx \sqrt{\sum_{l=1}^{u} T_{lb_l}}. \tag{15}$$

4.4 Voting with the Proposed Scoring Method

After the distance calculation described in Section 4.3, pairs of image identifier and distance are obtained. These pairs are sorted according to the distances and top-k results are used to calculate the proposed score described in Section 3 in voting. In this case, the t-th subset $\mathcal{N}_t(q_i)$ in Eq. (6) is defined as t nearest vectors of q_i. In this paper, we refer to this framework as the k-nearest neighbor (k-NN) voting framework. The voting algorithm is summarized in Algorithm 1.

Algorithm 1 k-NN voting function with the proposed score

Require: $Q = \{q_i\}_{i=1}^n, \{\mathcal{R}_j\}_{j=1}^m$
Ensure: $s_j \leftarrow$ similarity score between Q and $\mathcal{R}_j$ $(1 \le j \le m)$
1: $s_1, \cdots, s_m \leftarrow 0$
2: **for** $i = 1$ **to** n **do**
3: $\quad z_1, \cdots, z_m \leftarrow 0$
4: $\quad c_1, \cdots, c_m \leftarrow 0$
5: $\quad r_1, \cdots, r_k \leftarrow$ (approximate) k nearest vectors of q_i among $\{\mathcal{R}_j\}_{j=1}^m$
6: $\quad$ **for** $t = 1$ **to** k **do**
7: $\quad\quad j_t \leftarrow$ the reference identifier associated with r_t
8: $\quad\quad c_{j_t} \leftarrow c_{j_t} + 1$
9: $\quad\quad$ **if** $v_{j_t} < c_{j_t}/t$ **then**
10: $\quad\quad\quad v_{j_t} \leftarrow c_{j_t}/t$
11: $\quad\quad$ **end if**
12: $\quad$ **end for**
13: $\quad$ **for** $j = 1$ **to** m **do**
14: $\quad\quad s_j \leftarrow s_j + \log(\alpha_j z_j + 1)$
15: $\quad$ **end for**
16: **end for**

Table 1 Summary of different versions of the proposed method.

Section	Voting framework	$\mathcal{N}_t(q_i)$
§5.2	BoVW	A set of reference descriptors that are quantized into one of t nearest VWs of q_i.
§5.3	Improved BoVW	A set of t nearest reference descriptors of q_i.
§5.4	k-NN	A set of t nearest reference descriptors of q_i.

In Algorithm 1, given query vectors Q and reference vectors $\mathcal{R}_1, \cdots, \mathcal{R}_m$, similarity scores $s_1, \cdots, s_m$ between a query image Q and reference images $R_1, \cdots, R_m$ are obtained. For each query vector q_i, k-nearest vectors $r_1, \cdots, r_k$ of q_i are obtained in Line 5. In Lines 6–12, $\max_t |\mathcal{N}_t(q_i)|_j/|\mathcal{N}_t(q_i)|_{\text{all}}$ in Eq. (8) is obtained as z_j, where $|\mathcal{N}_t(q_i)|_j = c_j$ and $|\mathcal{N}_t(q_i)|_{\text{all}} = t$. In Line 14, the proposed score is calculated and voted to the reference image R_j, where α_j is precomputed using constant values in Eq. (8) for efficiency:

$$\alpha_j = \frac{\lambda}{1-\lambda} \frac{1/|\mathcal{R}_j|}{1/|\mathcal{R}_{\text{all}}|}. \tag{16}$$

Algorithm 1 is applicable not only to product quantization-based approximate nearest neighbor search but also applicable to any (approximate) nearest neighbor search algorithms such as locality sensitive hashing [31] as shown in Section 5.4.

In this paper, the parameters recommended in Ref. [18] are used; the size of the CQ and PQ codebooks k' and k^* are set to 20,000 and 256 respectively, and the number of vector decomposition u is set to 8.

5. Experimental Evaluation

In this section, we show the effectiveness and versatility of the proposed scoring method by applying it to the BoVW framework (Section 5.2), the improved BoVW framework (Section 5.3), and the locality sensitive hashing-based k-NN voting framework (Section 5.4). **Table 1** summarizes the different versions of the proposed method evaluated in the experiments.

© 2013 Information Processing Society of Japan

5.1 Experimental Setup

In the experiments, three distinct publically available datasets are used. The details of these datasets are summarized as follows.

- **Kentucky**: The University of Kentucky recognition benchmark dataset [*3] is provided by the authors of Ref. [26]. It includes 2,550 different objects or scenes. Each of these objects is represented by four images taken from four different angles, making 10,200 images in all. These images are used as both reference and query images. This dataset is used in Sections 5.2 and 5.3, focusing on large-scale image retrieval.
- **Stanford MVS**: The Stanford mobile visual search dataset [*4] contains camera-phone images of products, CDs, books, outdoor landmarks, business cards, text documents, museum paintings and video clips. While it includes eight classes of images, we use CD class images in this paper. These images consist of 100 reference images and 400 query images. In the experiments, all images are resized so that the long sides of images are less than 640 pixels, keeping the original aspect ratio. This dataset is used in Section 5.4, focusing on image recognition on mobile phones.
- **MIRFLICKR-1M**: The MIR Flickr collection [*5] is provided by the authors of Ref. [32]. It includes 1 million images downloaded from the social photography site Flickr™. We use this dataset as a distractor set in order to confirm the scalability of the proposed method in Sections 5.3 and 5.4.

Figure 4 shows sample images from the three datasets.

As an indicator of retrieval performance, mean average precision (MAP) [4], [26] is used. For each query, a precision-recall curve is obtained based on the retrieval results. Average precision is calculated as the area under the precision-recall curve. Finally, the MAP score is calculated as the mean of average precisions over all queries. Because the applications of the proposed method only modify the scoring part of the conventional frameworks, the computational cost and database size of the proposed method is theoretically the same as the conventional methods. Therefore, in the experiments, we focus on the evaluations of the proposed scoring method in terms of accuracy.

5.2 Application to the BoVW Framework

In order to show the effectiveness and versatility of the proposed scoring method, first it is applied to the standard BoVW framework [3]. In this case, a set of the nearest neighbor descriptors $\mathcal{N}(q_i)$ of q_i is defined as a set of reference descriptors that are quantized into the same VW as q_i. As subsets of $\mathcal{N}(q_i)$ in Eq. (6), only a single subset ($k = 1$) is defined:

$$\mathcal{N}(q_i) = \mathcal{N}_1(q_i) = \{r \in \mathcal{R}_{\text{all}} \mid \text{q}(r) = \text{q}(q_i)\}, \tag{17}$$

where $\text{q}(r)$ and $\text{q}(q_i)$ denote the identifiers of the nearest neighbor VWs of r and q_i. Then, s_{ij} is calculated using only frequencies of VWs:

$$s_{ij} = \log\left(\frac{\lambda}{1-\lambda}\frac{tf_j^{\text{q}(q_i)}/|\mathcal{R}_j|}{tf_{\text{all}}^{\text{q}(q_i)}/|\mathcal{R}_{\text{all}}|} + 1\right), \tag{18}$$

[*3] http://www.vis.uky.edu/~stewe/ukbench/
[*4] http://www.stanford.edu/~dmchen/mvs.html
[*5] http://press.liacs.nl/mirflickr/

(a) Kentucky dataset. Images in each row represent the same object.

(b) Stanford MVS dataset. Images in the top row are examples of reference images and images in the bottom row are examples of query images.

(c) MIRFLICKR-1M dataset.

Fig. 4 Example images from three datasets used in the experiments.

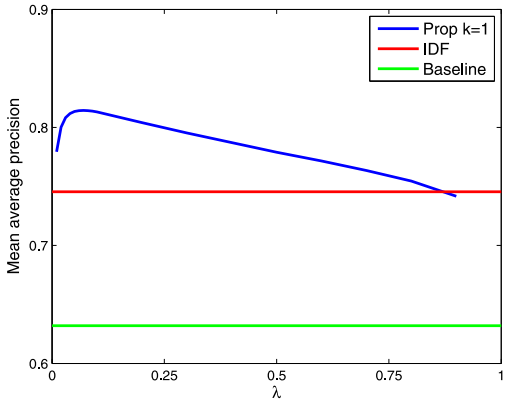


Fig. 5 Comparison of scoring methods in the BoVW framework.

where tf_j^w represents the frequency of the w-th VW in R_j, and tf_{all}^w the frequency of the w-th VW in all reference images.

The above proposed score s_{ij} is evaluated using the Kentucky dataset. **Figure 5** shows the MAP scores obtained with different scoring methods as a function of λ (valid only for the proposed method). The following three scoring methods are compared: (1) Baseline uses normalized frequency $tf_j^{\text{q}(q_i)} / \sqrt{\sum_w (tf_j^w)^2}$ of $\text{q}(q_i)$-th VW, (2) IDF uses IDF weighted frequency $tf_j^{\text{q}(q_i)} \cdot (idf^{\text{q}(q_i)})^2 / \sqrt{\sum_w (tf_j^w \cdot idf^w)^2}$, and (3) Prop uses s_{ij} in Eq. (18) as a voting score, respectively. The IDF weight of the w-th VW is calculated according to the number of all reference images and the number of reference images which have at least one descriptor assigned to the w-th VW:

© 2013 Information Processing Society of Japan

Table 2 Comparisons of the proposed method with the IDF method.

(a) Best 5 objects.

Object ID	Prop	IDF	Diff.	#descriptors
466	0.888	0.484	+0.404	304
631	1.0	0.604	+0.396	1,828
2068	0.942	0.559	+0.383	973
1741	0.954	0.574	+0.380	386
1315	0.869	0.491	+0.378	1,698

(b) Worst 5 objects.

Object ID	Prop	IDF	Diff.	#descriptors
1820	0.523	0.986	−0.463	4,358
2297	0.582	1.0	−0.418	4,238
2331	0.534	0.911	−0.377	4,316
1828	0.585	0.865	−0.280	4,758
2184	0.582	0.818	−0.236	3,708

$$idf^w = \log \frac{|\{R_j\}|}{|\{R_j \mid tf_j^w > 0\}|}. \tag{19}$$

Although the IDF weighting significantly improves the accuracy of the baseline scoring method, the proposed method achieves further improvement. In addition, the accuracy is not so sensitive to the choice of λ. The best MAP score of 0.814 is achieved with relatively small λ ($\lambda = 0.07$), which implies that there are a small number of features useful for object recognition [33]. Object-wise MAPs are also calculated for 2,550 objects. **Table 2** summarizes (a) the best 5 objects and (b) the worst 5 objects of the proposed method compared with the IDF method. In the table, object identifiers, the MAP scores of the proposed method and the IDF method, the differences of them, and the average numbers of descriptors are shown. We found that the proposed method performs better than the IDF method for objects with small numbers of descriptors, while worse for objects with large numbers of descriptors; on average, the proposed method outperforms the IDF method. This is related to the normalization terms associated with the number of descriptors $|\mathcal{R}_j|$ of a reference image: in the proposed method, each score is approximately proportional to $\log(C/|\mathcal{R}_j| + 1)$ according to Eq. (8) or Eq. (18), while, in the case of the IDF method with ℓ_2 normalization, each score is approximately proportional to $|\mathcal{R}_j|^{-1/2}$. **Figure 6** shows the graph of $y = \log(2{,}000/x + 1)/x^{-1/2}$, which represents the ratio of the score in the proposed method to the score in the IDF method as a function of the number of descriptors in a reference image [*6]. The scores of the proposed method are relatively small compared with the IDF method for images with large numbers of descriptors.

In the BoVW framework, the proposed scoring method can be used in conjunction with the multiple assignment approaches described in Section 2.2. For simplicity, we adopt a fixed number k for multiple assignment. In this case, the t-th subset $\mathcal{N}_t(q_i)$ in Eq. (6) is defined as a set of reference descriptors that are assigned to one of the t nearest VWs of q_i:

$$\mathcal{N}_t(q_i) = \bigcup_{1 \le s \le t} \{r \in \mathcal{R}_{\text{all}} \mid \mathrm{q}(r) = \mathrm{q}_s(q_i)\}, \tag{20}$$

where $\mathrm{q}_s(q_i)$ denotes the identifier of the s-th nearest VW of

[*6] We assume that $\lambda = 0.1$, $tf_j^{\mathrm{q}(q_i)} = 1$, and $tf_{\text{all}}^{\mathrm{q}(q_i)}/|\mathcal{R}_{\text{all}}| = 1/20{,}000$ in Eq. (18), resulting $C \simeq 2{,}000$. The term $tf_{\text{all}}^{\mathrm{q}(q_i)}/|\mathcal{R}_{\text{all}}|$ is approximately equal to the probability $1/k'$ with which a descriptor is assigned to a certain visual word, where $k'(= 20{,}000)$ is the number of the visual words.

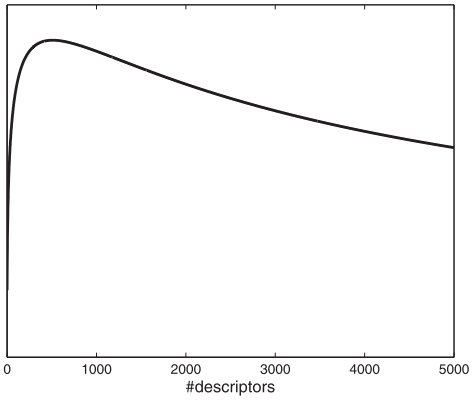


Fig. 6 The ratio of the score in the proposed method to the score in the IDF method as a function of the number of descriptors in a reference image. The average number of descriptors in the Kentucky dataset is about 1,700.

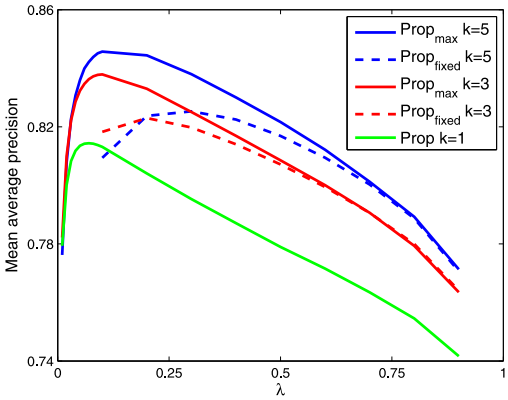


Fig. 7 Evaluation of the proposed scoring method under different settings.

q_i. **Figure 7** shows the MAP scores obtained with the proposed scoring method with different settings as a function of λ. Prop, $\mathsf{Prop}_{\mathsf{max}}k = 3$, and $\mathsf{Prop}_{\mathsf{max}}k = 5$ represent the results of the proposed scoring methods where k is respectively 1, 3, and 5. We can see that the proposed scoring method is effective even if it is used in conjunction with multiple assignment approaches. The best MAP score is improved from 0.814 ($k = 1, \lambda = 0.07$) to 0.838 ($k = 3$, $\lambda = 0.1$) and 0.846 ($k = 5, \lambda = 0.1$).

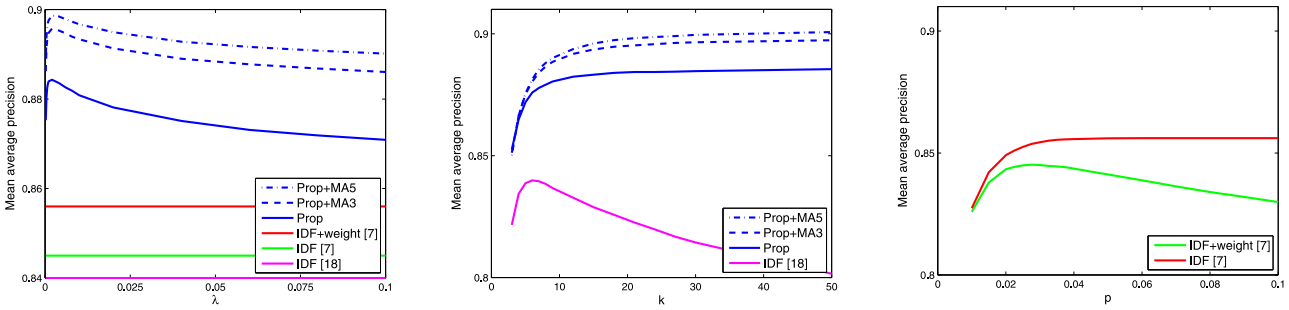
In Fig. 7, the modified version of the proposed scoring method is also compared. $\mathsf{Prop}_{\mathsf{fixed}}$ uses the fixed $t = k$ in the calculation of the proposed score; the score is defined as $s_{ij} = s_{ij}^k$ instead of the original definition $s_{ij} = \max_{1 \le t \le k} s_{ij}^t$ in Eq. (9). However, in this case, the degree of improvement attained by multiple assignment is not significant compared with the original scoring method. The best MAP scores of $\mathsf{Prop}_{\mathsf{fixed}}k = 3$ and $\mathsf{Prop}_{\mathsf{fixed}}k = 5$ are 0.823 ($\lambda = 0.1$) and 0.825 ($\lambda = 0.1$), respectively. This is because, if the fixed $t = k$ is used, the useful information of the multiple subsets $\mathcal{N}_1(q_i), \cdots, \mathcal{N}_k(q_i)$ cannot be exploited.

5.3 Application to the improved BoVW Framework

In this section, the proposed scoring method is applied to the improved BoVW framework described in Section 4, and compared with the conventional methods.

Figure 8 (a) shows a comparison of scoring methods in the approximate k-NN voting framework. These methods differ only

© 2013 Information Processing Society of Japan



(a) Accuracy as a function of λ. (b) Accuracy as a function of k. (c) Accuracy as a function of p.

Fig. 8 Comparison of scoring methods in the approximate k-NN voting framework.

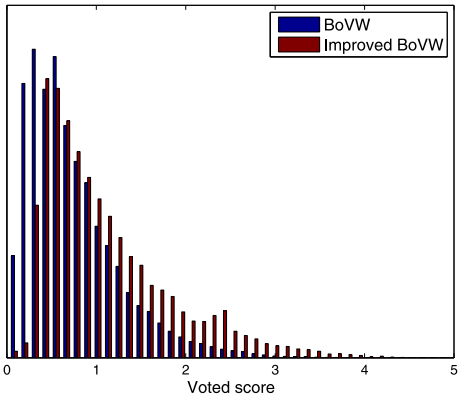


Fig. 9 The distributions of the proposed scores.

in terms of scoring; in other respects, they are the same as the proposed system described in Sections 4.1–4.3. First, the following four methods are considered: (1) IDF [18] votes the IDF scores to the top-k reference images ($k = 6$), (2) IDF [7] votes the IDF scores to the top-p percent reference images instead of the top-k ($p = 0.03$), (3) IDF+weight [7] votes the scores of $idf^2 \cdot \exp(-p^2/\sigma^2)$ to the top-p percent reference images ($p = 0.05$, $\sigma = 0.05$), and (4) Prop votes the proposed scores described in Algorithm 1 to the top-k reference images ($k = 24$). Parameters used in these methods are optimized with a grid search in parameter space. It is shown that the accuracy is significantly improved by using the proposed scoring method instead of IDF scoring, even if the IDF score is corrected with ratio information [7]. The proposed scoring method achieves the best MAP score of 0.884 with $\lambda = 0.002$. Figure 8 (b) shows the accuracy of Prop with $\lambda = 0.002$ and IDF [18] as a function of k, and Fig. 8 (c) shows the accuracy of IDF [7] and IDF+weight [7] as a function of p. We can see that the accuracy of the scoring methods without weighting (IDF [18] and IDF [7]) decreases if k or p is set to a large value, while the accuracy of the scoring methods with weighting (Prop and IDF+weight [7]) monotonically increases as k or p increases. Although optimal λ in Fig. 8 (a) is different from that in Fig. 5 and in Fig. 7, we found that resulting scores are similar. **Figure 9** shows the distributions of the proposed score with the BoVW framework ($\lambda = 0.1$) and that with the improved BoVW framework ($\lambda = 0.002$). Average scores are about 1.0 (0.744 in the BoVW framework and 1.112 in the improved BoVW framework). When λ is small (e.g., $\lambda = 0.002$), it may seem that Eq. (3) reduces to $p(q_i|R_j) = p(q_i|B)$. However, the first term $\lambda p(q_i|O_j)$ in Eq. (3) is not negligible because $p(q_i|O_j)$ is relatively larger

Table 3 Comparisons with state-of-the-art methods.

(a) MAP and KS of the proposed method.

	Prop	Prop+MA3	Prop+MA5	Prop+MA5+RS
MAP	0.884	0.896	0.899	0.912
KS	3.43	3.48	3.49	3.55

(b) KS of state-of-the-art methods.

	Ref. [13]	Ref. [13]+	Ref. [12]	Ref. [11]	Ref. [6]	Ref. [17]	Ref. [23]
KS	3.52	3.56	3.26	3.29	3.45	3.67	3.64

than $p(q_i|B)$; assuming $|\mathcal{N}_t(q_i)|_j = 1$, $|\mathcal{N}_t(q_i)|_{\text{all}} = k = 20$, and $|\mathcal{R}_{\text{all}}|/|\mathcal{R}_j| = 10{,}000$ in Eq. (7), then $p(q_i|O_j)/p(q_i|B) = 500$. Note that $|\mathcal{R}_{\text{all}}|/|\mathcal{R}_j|$ is approximately equal to the number of reference images.

In Fig. 8, the proposed scoring method is also evaluated in conjunction with a fixed number of multiple assignments [8] with the number of assignments being either 3 (Prop+MA3) or 5 (Prop+MA5), where several number of lists corresponding to the nearest VWs of q_i in the inverted index are searched. In this case, multiple assignment simply improves the accuracy (recall) of the k-nearest neighbor search results in product quantization-based nearest neighbor search. The best MAP score of the proposed scoring method is improved from 0.884 to 0.896 (Prop+MA3) and 0.899 (Prop+MA5).

In order to compare the proposed method with state-of-the-art methods, the proposed method is evaluated in terms of Kentucky score (KS), which is the average number of relevant images in the query's top 4 retrieved images as in Ref. [26]. We also evaluate the proposed method with RootSIFT (RS) descriptor [16] as Prop+MA5+RS, where each SIFT descriptor is simply ℓ_1-normalized and each element is square rooted. **Table 3** shows the comparisons of the proposed method with state-of-the-art methods. Although the proposed method does not utilize any geometric information nor depend on re-ranking approach, it achieves comparable or even better results than the state-of-the-art methods. We also evaluated the original naive-bayes nearest-neighbor (NBNN) method [25]. The NBNN method achieved a MAP score of 0.800 even if exact nearest neighbor distances are used. This result comes from the fact that only a single training image is available for each class in image retrieval problems, where the NBNN classifier reduces to a nearest-neighbor-image (NN-image) classifier [25]. In terms of efficiency, the NBNN method with approximate nearest neighbor search [27] requires 0.035 seconds to calculate a distance between a query image and a reference image; the NBNN method takes 357 seconds per query against the Kentucky dataset which consists of 10,200 images,

© 2013 Information Processing Society of Japan

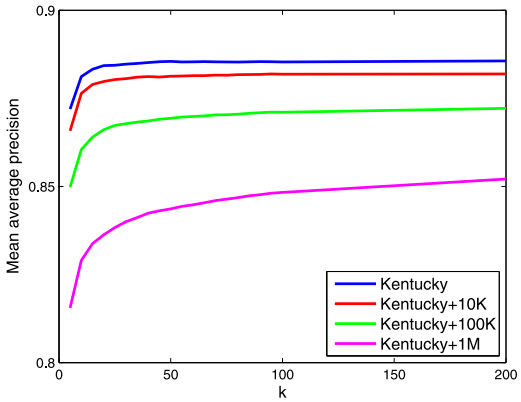


Fig. 10 Comparison of the proposed scoring method using different sizes of the dataset.

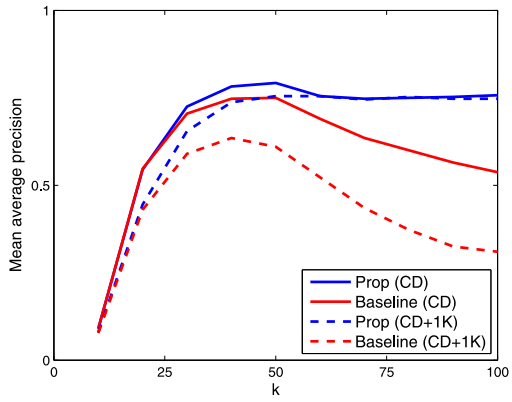


Fig. 11 MAP as a function of the number k of nearest neighbors used in voting.

while the proposed method takes 0.66 seconds per query. Processing times were measured using a program with a single thread on a machine with a Core i7 970 CPU.

Finally, we confirm the scalability of the proposed scoring method by adding the MIRFLICKR-1M dataset as a distractor. **Figure 10** shows the MAP scores obtained by the proposed scoring method with $\lambda = 0.002$ as a function of the number k of nearest neighbors. The following four sizes of the dataset are evaluated; Kentucky corresponds to Prop in Fig. 8 (b), where only the Kentucky dataset is used in the evaluation. Kentucky+10K, Kentucky+100K, and Kentucky+1M represents the results of the proposed method, where 10 K, 100 K, and 1 M images out of the MIRFLICKR-1M dataset are added to the Kentucky dataset as a distractor. We can see that the proposed scoring method requires relatively large k to obtain adequate results when the size of the dataset is increased. This is because the error caused by the approximation introduced in Section 3.2 is increased as the size of the dataset is increased. However, the degradation of the accuracy caused by the increase of dataset is relatively small compared with conventional methods. In the case of the conventional method [7], if the 1 M distractor images are added, the MAP score declines from 0.845 to 0.757 [7], which corresponds to 10.4% degradation. Meanwhile, in the case of the proposed method, the MAP score declines from 0.885 to 0.852 ($k = 200$), which corresponds to only 3.7% degradation. It can be said that the effectiveness of the proposed scoring method becomes more significant when the database size is increased. This is because the background model $p(q_i|B)$ is taken into account in the proposed scoring method, where the accuracy of k-nearest neighbor density estimation is improved as the number of samples is increased.

5.4 Application to a Different Framework

In this section, the proposed scoring method is evaluated under quite different settings from those in Sections 5.2 and 5.3: a different feature extraction method, approximate nearest neighbor search method, and dataset are used. Considering image recognition applications on mobile devices, the following settings are used.

For feature extraction, we adopt ORB [34], which utilizes multi-scale FAST detector [35] and oriented BREIF descriptor [36]. Because ORB realizes very fast detection and description, it even works on mobile phones. The implementation of the OpenCV library [*7] is used in the experiment. Up to 500 ORB features are extracted from three scales. For approximate nearest neighbor search to get top-k descriptors, locality sensitive hashing [31] is used, where the number of hash tables and the size of the hash key is set to 20 and 2^{10}, respectively. For the dataset, we use the CD class images from the Stanford MVS dataset. The Stanford MVS dataset is designed to be used for the evaluation of visual recognition systems on mobile devices.

Figure 11 compares the proposed scoring method in Algorithm 1 with the baseline method, which votes a score of 1.0 to reference images corresponding to the top-k nearest neighbor descriptors of q_i. In the figure, the proposed method achieves better accuracy than the baseline method for all k. The two methods are also evaluated using a larger dataset, which consists of the CD class images and 1,000 images from the MIRFLICKR-1M dataset. If the 1 K images are added as a distractor, the best MAP score of Baseline declines from 0.750 to 0.635 (15.3% degradation), while the best MAP score of Prop declines from 0.793 to 0.755 (4.8% degradation [*8]). The proposed scoring method is more effective in the larger dataset, which is consistent with the results in Section 5.3.

6. Conclusion

In this paper, we have proposed a new, effective, and unified scoring method for local feature-based image retrieval. The proposed scoring method has been derived by solving the large-scale image retrieval problem as a classification problem with a large number of classes. The resulting proposed score is based on the ratio of the probability density function of an object model to that of a background model, which is efficiently calculated via nearest neighbor density estimation. The effectiveness of the proposed method was confirmed by applying it to a standard and improved bag-of-visual words-based framework and a k-nearest neighbor voting framework. The proposed method can also be used in conjunction with many other methods such as hierarchical vocabulary [26] or learned vocabulary [37], where the matched descriptors can be ordered. As a future research topic, we are interested

[*7] http://opencv.org/

[*8] This degradation is relatively larger than that in Fig. 10. It is probably due to the difference in the discriminative power of the SIFT and ORB descriptors.

© 2013 Information Processing Society of Japan

in the application of the proposed method to other modalities such audio and structured data.

References

[1] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T. and Gool, L.V.: A Comparison of Affine Region Detectors, *IJCV*, Vol.60, No.1-2, pp.43–72 (2005).

[2] Mikolajczyk, K. and Schmid, C.: A Performance Evaluation of Local Descriptors, *IEEE Trans. PAMI*, Vol.27, No.10, pp.1615–1630 (2005).

[3] Sivic, J. and Zisserman, A.: Video google: A text retrieval approach to object matching in videos, *Proc. ICCV*, pp.1470–1477 (2003).

[4] Jégou, H., Douze, M. and Schmid, C.: Improving bag-of-features for large scale image search, *IJCV*, Vol.87, No.3, pp.316–336 (2010).

[5] Roelleke, T. and Wang, J.: TF-IDF Uncovered: A Study of Theories and Probabilities, *Proc. SIGIR*, pp.435–442 (2008).

[6] Philbin, J., Chum, O., Isard, M., Sivic, J. and Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching, *Proc. CVPR*, pp.1–8 (2007).

[7] Uchida, Y., Takagi, K. and Sakazawa, S.: Ratio Voting: A New Voting Strategy for Large-Scale Image Retrieval, *Proc. ICME*, pp.759–764 (2012).

[8] Philbin, J., Chum, O., Isard, M., Sivic, J. and Zisserman, A.: Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases, *Proc. CVPR*, pp.1–8 (2008).

[9] Uchida, Y., Takagi, K. and Sakazawa, S.: An Alternative to IDF: Effective Scoring for Accurate Image Retrieval with Non-Parametric Density Ratio Estimation, *Proc. ICPR*, pp.1285–1288 (2012).

[10] Zhao, W.L. and Ngo, C.W.: Scale-Rotation Invariant Pattern Entropy for Keypoint-based Near-Duplicate Detection, *IEEE Trans. Image Processing*, Vol.18, No.2, pp.412–423 (2009).

[11] Lin, Z. and Brandt, J.: A Local Bag-of-Features Model for Large-Scale Object Retrieval, *Proc. ECCV*, pp.294–308 (2010).

[12] Zhang, Y., Jia, Z. and Chen, T.: Image Retrieval with Geometry-Preserving Visual Phrases, *Proc. CVPR*, pp.809–816 (2011).

[13] Shen, X., Lin, Z., Brandt, J., Avidan, S. and Wu, Y.: Object retrieval and localization with spatially-constrained similarity measure and k-NN re-ranking, *Proc. CVPR*, pp.3013–3020 (2012).

[14] Chum, O., Philbin, J., Sivic, J., Isard, M. and Zisserman, A.: Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval, *Proc. ICCV*, pp.1–8 (2007).

[15] Chum, O., Mikulík, A., Perdoch, M. and Matas, J.: Total Recall II: Query Expansion Revisited, *Proc. CVPR*, pp.889–896 (2011).

[16] Arandjelović, R. and Zisserman, A.: Three things everyone should know to improve object retrieval, *Proc. CVPR*, pp.2911–2918 (2012).

[17] Qin, D., Grammeter, S., Bossard, L., Quack, T. and Gool, L.V.: Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors, *Proc. CVPR*, pp.777–784 (2011).

[18] Jégou, H., Douze, M. and Schmid, C.: Product Quantization for Nearest Neighbor Search, *IEEE Trans. PAMI*, Vol.33, No.1, pp.117–128 (2011).

[19] Jégou, H., Douze, M. and Schmid, C.: Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search, *Proc. ECCV*, pp.304–317 (2008).

[20] Weiss, Y., Torralba, A. and Fergus, R.: Spectral Hashing, *Proc. NIPS*, pp.1753–1760 (2008).

[21] Brandt, J.: Transform Coding for Fast Approximate Nearest Neighbor Search in High Dimensions, *Proc. CVPR*, pp.1815–1822 (2010).

[22] Uchida, Y., Agrawal, M. and Sakazawa, S.: Accurate Content-Based Video Copy Detection with Efficient Feature Indexing, *Proc. ICMR* (2011).

[23] Jégou, H., Douze, M. and Schmid, C.: On the burstiness of visual elements, *Proc. CVPR*, pp.1169–1176 (2009).

[24] Jain, M., Benmokhtar, R., Grosand, P. and Jégou, H.: Hamming Embedding Similarity-based Image Classification, *Proc. ICMR* (2012).

[25] Boiman, O., Shechtman, E. and Irani, M.: In Defense of Nearest-Neighbor Based Image Classification, *Proc. CVPR*, pp.1–8 (2008).

[26] Nistér, D. and Stewénius, H.: Scalable Recognition with a Vocabulary Tree, *Proc. CVPR*, pp.2161–2168 (2006).

[27] Muja, M. and Lowe, D.G.: Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, *Proc. VISAPP*, pp.331–340 (2009).

[28] Andoni, A.: Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions, *Proc. FOCS*, pp.459–468 (2006).

[29] Mikolajczyk, K. and Schmid, C.: Scale & Affine Invariant Interest Point Detectors, *IJCV*, Vol.60, No.1, pp.63–86 (2004).

[30] Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints, *IJCV*, Vol.60, No.2, pp.91–110 (2004).

[31] Gionis, A., Indyk, P. and Motwani, R.: Similarity Search in High Dimensions via Hashing, *Proc. VLDB*, pp.518–529 (1999).

[32] Huiskes, M.J., Thomee, B. and Lew, M.S.: New trends and ideas in visual concept detection: The MIR flickr retrieval evaluation initiative, *Proc. MIR*, pp.527–536 (2010).

[33] Turcot, P. and Lowe, D.G.: Better Matching with Fewer Features: The Selection of Useful Features, *Proc. WS-LAVD*, pp.2109–2116 (2009).

[34] Rublee, E., Rabaud, V., Konolige, K. and Bradski, G.: ORB: An efficient alternative to SIFT or SURF, *Proc. ICCV*, pp.2564–2571 (2011).

[35] Rosten, E. and Drummond, T.: Machine learning for high-speed corner detection, *Proc. ECCV*, pp.430–443 (2006).

[36] Calonder, M., Lepetit, V., Strecha, C. and Fua, P.: BRIEF: Binary Robust Independent Elementary Features, *Proc. ECCV*, pp.778–792 (2010).

[37] Mikulík, A., Perdoch, M., Chum, O. and Matas, J.: Learning a Fine Vocabulary, *Proc. ECCV*, pp.1–14 (2010).

Yusuke Uchida received his Bachelor Degree of Integrated Human Studies from Kyoto University, Kyoto, Japan, in 2005. He received a degree of Master of Informatics from Graduate School of Informatics, Kyoto University, in 2007. His research interests include large-scale content-based multimedia retrieval, augmented reality, and image processing. He is currently with KDDI R&D Laboratories, Inc.

Shigeyuki Sakazawa received his B.E., M.E., and Ph.D. degrees from Kobe University, Japan, all in electrical engineering, in 1990, 1992, and 2005 respectively. He joined Kokusai Denshin Denwa (KDD) Co. Ltd. in 1992. Since then he has been with its R&D Division, and now he is a senior manager of the Media and HTML5 Application Laboratory in KDDI R&D Laboratories Inc., Saitama, Japan. His current research interests include video coding, video communication system, image recognition, and CG video generation. He received the Best Paper Award from IEICE in 2003.

(Communicated by *Takayoshi Yamashita*)

© 2013 Information Processing Society of Japan