Front. Electr. Electron. Eng. China 2010, 5(4): 419–429
DOI 10.1007/s11460-010-0099-6

RESEARCH ARTICLE

Manni DUAN, Xiuqing WU

Visual polysemy and synonymy: toward near-duplicate image retrieval

Abstract Near-duplicate image retrieval aims to find all images that are duplicate or near duplicate to a query image. One of the most popular and practical methods in near-duplicate image retrieval is based on bag-of-words (BoW) model. However, the fundamental deficiency of current BoW method is the gap between visual word and image's semantic meaning. Similar problem also plagues existing text retrieval. A prevalent method against such issue in text retrieval is to eliminate text synonymy and polysemy and therefore improve the whole performance. Our proposed approach borrows ideas from text retrieval and tries to overcome these deficiencies of BoW model by treating the semantic gap problem as visual synonymy and polysemy issues. We use visual synonymy in a very general sense to describe the fact that there are many different visual words referring to the same visual meaning. By visual polysemy, we refer to the general fact that most visual words have more than one distinct meaning. To eliminate visual synonymy, we present an extended similarity function to implicitly extend query visual words. To eliminate visual polysemy, we use visual pattern and prove that the most efficient way of using visual pattern is merging visual word vector together with visual pattern vector and obtain the similarity score by cosine function. In addition, we observe that there is a high possibility that duplicates visual words occur in an adjacent area. Therefore, we modify traditional Apriori algorithm to mine quantitative pattern that can be defined as patterns containing duplicate items. Experiments prove quantitative patterns improving mean average precision (MAP) significantly.

Keywords near-duplicate image retrieval, bag-of-words (BoW) model, visual synonymy, visual polysemy, extended similarity function, query expansion, visual pattern

Received August 23, 2009; accepted April 26, 2010

Manni DUAN (✉), Xiuqing WU
Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China
E-mail: mnduan@mail.ustc.edu.cn

1 Introduction

Near-duplicate image retrieval aims to find all images that are duplicated or near duplicated to a query image. Detection and retrieval of near-duplicate image is very useful in a variety of real-world applications. For example, it can improve traditional text-based image search by filtering duplicate returning images; it can be used as a bridge to connect two web pages in different languages; and it can provide similarity clues for recognizing visual events and searching news video clips. One of the most popular and practical methods in near-duplicate image retrieval is based on bag-of-words (BoW) model [1] in which one image is represented as a set of high-dimensional local features. These local features are then clustered to "visual words" for efficiency's favor. Based on visual words, which are assumed to be independent text retrieval methods, such as term frequency-inverse document frequency (TF-IDF) [2], cosine similarity can be utilized in image retrieval. BoW method's strength lies in its simplicity and effectiveness.

However, the fundamental deficiency of current BoW methods is the gap between visual word and image's semantic meaning. Actually, there are usually many ways to express a given visual meaning, and most visual words have multiple meanings, so visual words in a user's query will literally match visual words in database images that are not of interest to the user. Similar problem also plagues existing text retrieval. In text retrieval, users want to retrieve on the basis of conceptual content, and individual text words provide unreliable evidence about the conceptual topic or meaning of a document. A prevalent method against such issue in text retrieval is to eliminate text synonymy and polysemy and therefore improve the whole performance. Motivated by the success in text retrieval, in this work, we would like to reduce the

semantic gap by eliminating visual polysemy and visual synonymy.

We use visual synonymy in a very general sense to describe the fact that there are many different visual words referring to the same parts of same objects. Features describing same visual meaning may be clustered to different visual words due to unbalanced distributed feature space, inappropriate clustering parameters, etc. The prevalence of visual synonymy tends to decrease the "recall" performance of image retrieval systems.

By visual polysemy, we refer to the general fact that most visual words have more than one distinct meaning. Generally speaking, one visual word may be composed of several to several hundred visual features. Different image taken condition, unbalanced distributed feature space, inappropriate clustering parameters, etc., will make features presenting totally different appearance to be clustered to one visual word. Thus, using a visual word in a query image does not necessarily mean that an image containing the same visual word is of interest. Visual polysemy is one factor underlying poor "precision".

This "visual synonymy and polysemy" concept is borrowed from text retrieval problem to state our issues in near-duplicate image retrieval more clearly. It differs from text synonymy and polysemy because it depends largely on clustering method. Therefore, we cannot simply adopt all methods from text retrieval to conquer our problem in image retrieval.

In our work, we aim at overcoming the problem of both visual synonymy and polysemy and therefore improve near-duplicate image retrieval performance. To eliminate visual synonymy, we present a new image similarity function, which implicitly extends a query visual word to a set of words. The assumption is that visual words are not totally independent, and a visual word with high probability presents similar object with visual words in nearby feature space. This is also a phenomenon that we have observed in our experiment. Comparing with text retrieval field, we regard this extended similarity function as query expansion technique. To eliminate visual polysemy, visual pattern by which we refer to a meaningful adjacent word set is used in our BoW model. The idea is that a word can indicate different meanings when combining with different adjacent word sets. Moreover, in our experiment, we observe that there exists a large portion of duplicate words within one adjacent area. Therefore, we modify traditional Apriori algorithm to mine quantitative pattern, which can be defined as patterns containing duplicate items (e.g., $\{W_a,W_a\}$), where W_a is a visual word in the context. We also compare different usages of visual pattern in our system:

1) Pre-ranking, which means that we first rank database images according to their similarity score counted by visual pattern vector and then re-rank the top K images according to their similarity score counted by visual word vector.

2) Similarity-merging, which means that we get a total similarity score by adding similarity counted from visual word vector to similarity counted from visual pattern vector, and then, database images are ranked according to this total similarity.

3) Vector-merging, which means that we merge visual word vector and pattern vector into one total vector, similarity score is counted by this total vector.

4) Re-ranking, which means that we first rank database images according to their similarity score counted by visual word vector only and then re-rank the top K images according to their similarity score counted by visual pattern vector.

Briefly speaking, in this work, we 1) propose an extended similarity function to implicitly execute query expansion thus improve image retrieval's recall, 2) use visual pattern in near-duplicate image retrieval and compare different ways of using it, and 3) introduce "quantitative pattern" in image retrieval and prove its effectiveness.

The remainder of this paper is organized as follows. First, related works are introduced in Sect. 2. Section 3 introduces visual synonymy and our proposed extended similarity function. Section 4 introduces visual polysemy and visual pattern. Quantitative pattern mining method is also described in this section. Section 5 introduces the details of our near-duplicate image retrieval system. Section 6 shows the results of experiments and discussion. Finally, conclusions and future work are given in Sect. 7.

2 Related works

Our proposed extended similarity function can be regarded as an implicitly query extension method, which tries to describe the query feature with appropriate visual word. From this aspect, our method is related to generalized vector space model [3] that aims to improving text retrieval's performance. The key idea of this work is to compute word correlations directly from automatic indexing scheme. The counting method for word correlations is based on Boolean algebra, which is effective, however, it is computationally expensive. Similarly, we build a word correlation matrix in our work to find correlation between visual words. Due to scale-invariant feature transform (SIFT) [4] feature's robustness, we use SIFT as local feature and regard SIFT matching as a reliable tool of labeling visual words with similar visual meaning. Therefore, we simply use SIFT feature match instead of computationally expensive statistical method [3] to construct such similarity matrix.

In image retrieval domain, our method is also related to Schindler's work [5], which proposed a greedy n-best paths (GNP) searching algorithm based on hierarchical k-means tree to find the most suitable visuals for features.

This searching algorithm followed multiple branches at each level rather than just the branch whose parent is closest to the query feature. It has been proven to be an effective method to eliminate mismatching. However, Ref. [5] is still based on the assumption that there exists one best visual word for a certain visual meaning, and it cannot deal with an inappropriate vocabulary that intrinsically has a lot of visual synonyms. Instead of finding the most appropriate word for a visual meaning, our method focuses on extending one query visual word to a set of correlated visual words.

The visual pattern concept that is borrowed from text retrieval domain facilitates our work related to Ref. [6]. In this work, Mitra investigated the usefulness of patterns for text retrieval by different weighting methods.

Our work can also be compared with Refs. [7–10], where a similar concept of visual pattern has been introduced. Different with these works, which are toward classification task or localization task, visual pattern is used for image retrieval task in our work. The video mining method [7] proposed by Sivic and Zisserman is a similar work to ours. It uses viewpoint invariant features to form configurations and obtains principal objects, characters, or scenes in a video by mining the reoccurrence configurations. Potential objects in videos are shown as mining examples. In Ref. [8], a method for mining frequently occurring objects from video is presented. Object candidates are detected by finding recurring spatial arrangements of affine covariant regions. A visual phrase lexicon concept is proposed in Ref. [9], where a visual phrase is a meaningful spatial co-occurrence pattern of visual words. The visual phrase lexicon is then used to refine the visual word clustering by tuning the similarity measure through metric learning. Reference [10] presents an approach to automatically find spatial configurations of local features occurring frequently on instances of a given object class. The visual patterns are evaluated by bounding box hit rate, which is to measure how often visual patterns occur in an object's bounding box.

Compared to our method, all the previous works related to visual pattern aim to mine frequent image regions or features but not to improve the performance of classification or retrieval directly. In our work, we investigate the usefulness of patterns by different methods and give evaluation by objective criteria.

In this work, we also propose to use quantitative pattern in image retrieval, which makes our paper related to Ref. [11]. This work introduces the problem of mining associate rules in large relational table containing both quantitative and categorical attributes. For complexity's favor, instead of using the algorithm mentioned in Ref. [11], which can deal with continuous valued attributes, we simply preprocess the transactions so that duplicated words in one transaction are distinguishable by the original Apriori algorithm [12].

3 Visual synonymy

3.1 Origination and definition of visual synonymy

As we know, local features like SIFT are very robust under scale/rotation/transform. However, local feature match is of high complexity if an exhaustive search is taken. To make local feature match more efficient, BoW model clusters local features into visual words and assumes that similar features will definitely be clustered into one visual word. It is obvious that this clustering idea gains efficiency on the sacrifice of accuracy.

If two features in a pair of duplicate image have very small distance, we call them SIFT-match feature pair. We have observed in experiments that a large portion of SIFT-match feature pair will be clustered to two different visual words. It is obvious that these SIFT-match feature pairs are no longer correlated in consequent process of BoW model. Plainly speaking, these SIFT-match feature pairs are lost during clustering. To estimate the quantity of lost feature pairs, we record SIFT-match feature pairs before clustering and compare their visual words after clustering. A word correlation matrix M is shown in Fig. 1, which is obtained by such comparing.

In Fig. 1, we present normalized word correlation matrix. We run SIFT feature match on near-duplicate image pairs from ZuBuD database (http://www.vision.ee.ethz.ch/showroom/zubud/index.en.html) and record the visual words of each SIFT-match feature pair. For example, if feature 1 in an image is clustered to visual word w_i, while feature 2 in another image is clustered to visual word w_j. Moreover, the feature distance in feature space of these two features is very small. We call $\{w_i,w_j\}$ as one pair of correlated visual word. Word correlation matrix M is defined as a matrix recording the number of correlated visual word pair. In addition, we normalize M to make each row's max value equal to one.

Averagely speaking, the possibility that two SIFT-match features being clustered to one visual word is around 0.6, which also means that the possibility of two SIFT-match features being clustered to different visual words is 0.4. We call feature pairs being clustered to different visual words as "lost feature pairs". In Fig. 2, green lines in the left column illustrate all matched SIFT feature pairs in two near duplicated images, while red lines in the right column illustrate SIFT feature pairs that are clustered into different visual word pairs. Figure 2 demonstrates that most of these lost feature pairs describe corresponding parts of these two scenes and therefore should have been clustered to the same visual words.

Motivated by this phenomenon, we define visual synonym as a set of visual words presenting similar features. Features belonging to these words have a high probability to be SIFT-match feature pairs.

Table 1 shows portion of the lost SIFT-match feature

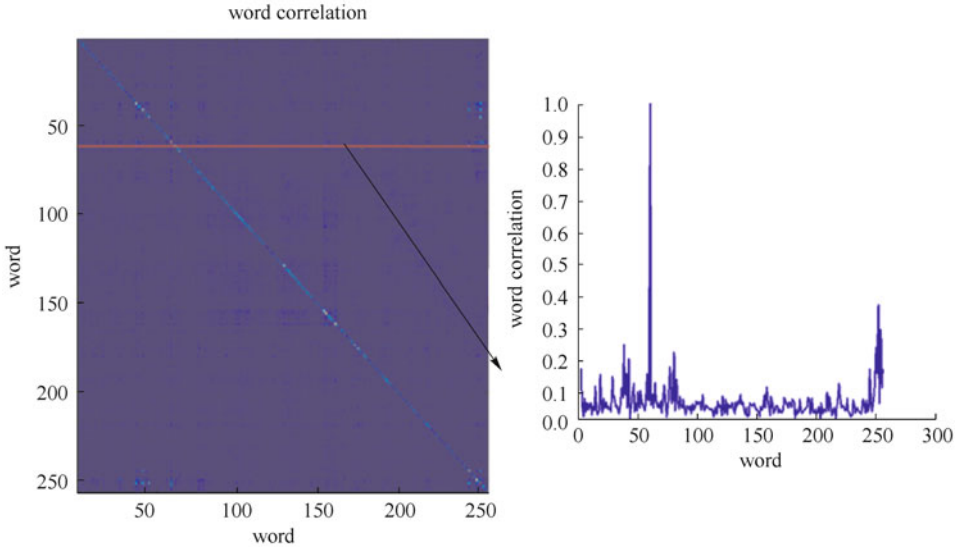

Fig. 1 Word correlation matrix

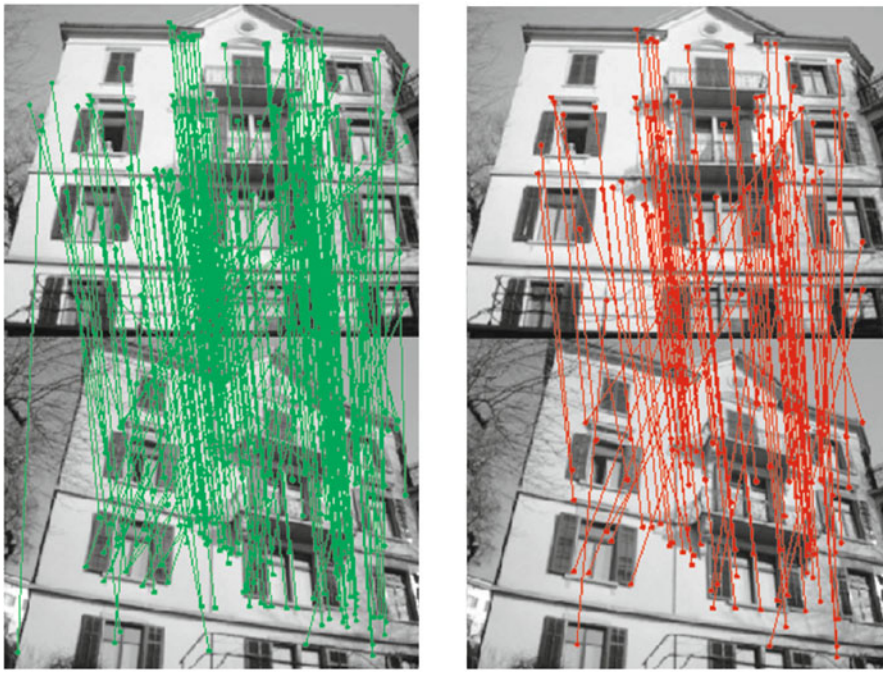

Fig. 2 Case study of synonymy

Table 1 Portion of synonymy in vocabularies of different size

vocabulary size	synonymy portion/%
1296	34
2401	38
4096	41
6561	45

pairs, or let us call them visual synonyms. Synonymy portion equals $\sum_{i \neq j} M(i,j) / \sum_{i,j} M(i,j) \times 100\%$. In Table 1, it is also easy for us to draw the conclusion that the quantity of visual synonyms increases with vocabulary size.

In text retrieval, to deal with the synonymy problem, either one needs to know the explicit representation of the word vectors or one needs some assumptions to account for the correlations between words. Actually, these two methods are all toward capturing the sets of synonym, thus are intrinsically the same thing [3]. In our work, we propose to conquer synonymy by computing word correlations, and we find that the word correlation matrix

M mentioned above is a presentation of words' correlation that will serve our purpose. To be more specific, we use M as a new kernel in similarity function to reduce the effect of visual synonymy. Details will be introduced in the next section.

3.2 Extended similarity function

Recall the similarity function in BoW model: image similarity between images A and B are derived as

$$\mathrm{sim}(\boldsymbol{A},\boldsymbol{B}) = \cos(\boldsymbol{A},\boldsymbol{B}) = \frac{\boldsymbol{A}\cdot\boldsymbol{B}}{\sqrt{\boldsymbol{A}\cdot\boldsymbol{A}}\cdot\sqrt{\boldsymbol{B}\cdot\boldsymbol{B}}}, \tag{1}$$

$$\boldsymbol{A}\cdot\boldsymbol{B} = \sum_{i,j} W_A^i W_B^j S(i,j), \tag{2}$$

where W_A^i is the weight of the ith visual word in image A. Usually, it could be TF-IDF [2]. Actually, $S(i, j)$ should be regarded as correlation matrix. Under the assumption of independence between visual words, the similarity kernel $S(i, j)$ is reduced to the Kronecker delta function, which means that one visual word is only related to itself:

$$S(i,j) = \delta(i,j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \tag{3}$$

However, as what we have observed previously, there exist a lot of visual synonyms in the vocabulary, and it is easy for the words to be synonyms with nearby words. Therefore, we extend the similarity matrix to a diagonal matrix:

$$S(i,j) = M(i,j), \tag{4}$$

where $M(i, j)$ is an element of M mentioned above.

4 Visual polysemy

4.1 Origin and definition of visual polysemy

Same as visual synonymy, visual polysemy is another product of feature clustering. Generally speaking, one visual word may contain several to several hundred visual features. Unbalanced distributed feature space, inappropriate clustering parameters, etc., will make features presenting totally different appearance be clustered to one visual word. In Fig. 3, circles of the same color indicate features belonging to the same visual word. It is obvious that these features do not mean the same visual meaning. We call a word visual polyseme if features belonging to it can present more than one distinct appearance. In text retrieval domain, a popular approach against polysemy is coordination with other words to disambiguate meaning. Success is severely hampered by the users' inability to think of appropriate limiting words if they do exist [13]. Different with text query word, image query inherently have numerous surrounding visual words for each single query visual word; therefore, it is reasonable to take visual words' surrounding visual words as a device against visual polysemy.

An attempt to deal with the visual polysemy problem in our work is relied on visual pattern, which is defined as a meaningful adjacent word set. This visual pattern concept has also been proposed in Refs. [7–9] through different tasks. Similar with these previous works, we use the Apriori algorithm [12] to mine visual patterns from visual vocabulary. In our experiment, we also observed that one certain visual word could occur multiple times in an adjacent area; see Fig. 4. Therefore, we modify Apriori algorithm to mine quantitative pattern, which is defined as pattern containing duplicate visual words, e.g., $\{W_a, W_a\}$.

4.2 Visual pattern

We define transactions over space so that an Apriori-like algorithm can be used. Frequent item sets mined from transaction database are regarded as visual patterns.

Before mining the patterns, we need to define what a transaction for image features is. Transactions over space can be defined by a reference-feature centric method. In this method, transactions are created around one specified feature. References [7–9] define their transactions as a central feature together with its k-nearest neighborhood. Reference [10] uses the scale of the central feature to define

(a) (b)

Fig. 3 Case study of visual polysemy. (a) Case 1; (b) case 2

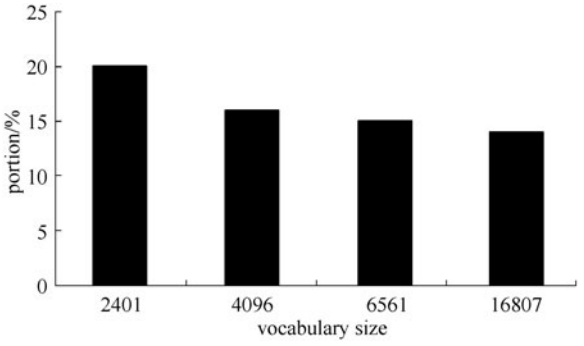

Fig. 4 Portion of transactions containing quantitative patterns on ZuBuD database

the size of adjacent area. All features in this adjacent area are composed of one transaction; hence, it is supposed to be scale invariant. We adopt the transaction definition in Ref. [10].

Let $I = \{w_1,w_2,...,w_k\}$ be a set of visual words. Transaction database $D = \{T_1,T_2,...,T_N\}$ is a set of N transactions with unique identifiers, where T_i is one transaction. Each item in T_i is a feature being clustered to a certain visual word in I. Pattern mining method aims to mine frequently co-occurrent visual word subset from transaction database D.

The support of a word set $A \subseteq I$ is defined as

$$\text{support}(A) = \frac{|\{T \in D | A \subseteq T\}|}{|D|}$$

to describe one set's occurrence frequency. A is called frequent if $\text{support}(A) \geqslant s$, where s is a threshold defined by user.

Another measurement of a word set is described as the confidence of a word set:

$$\begin{aligned}\text{confidence}(A \rightarrow B) &= \frac{\text{support}(A \cup B)}{\text{support}(A)} \\ &= \frac{|T \in D | (A \cap B) \subseteq T|}{|\{T \in D | A \subseteq T\}|},\end{aligned} \quad (5)$$

where A and B are two word sets. The confidence can be seen as a maximum likelihood estimation of the conditional probability that B is true given that A is true.

Finally, we define association hyperedges (AH) [12] of an item set $A = \{w_1,w_2,...,w_N\}$ to give an average estimation of the confidence between words in the set:

$$\text{AH}(A) = \frac{1}{N}\sum_{i=1}^{N} \text{confidence}(A - \{w_i\} \rightarrow w_i). \quad (6)$$

This AH is regarded as a measurement to rank patterns for it can measure how often a set of words co-occur in general. The K largest AH pattern word sets compose visual pattern list in our system.

4.3 Quantitative pattern mining

Conceptually, original Apriori algorithm can be viewed as a question of finding association between "1" values in the relational table where all the attributes are Boolean. The table has an attribute corresponding to each visual word and a record to each transaction. The value of an attribute is "1" if the corresponding visual word is presented in the transaction corresponding to this record, else "0". In our case, the original Apriori algorithm only cares about how often visual words with different indexes co-occur and ignore those co-occurred words with the same indexes. All of these previous works [7–10] simply ignored quantitative pattern and utilized original Apriori algorithm directly. However, in our experiment, we observe that a large portion of transactions contain duplicate items, as Fig. 4 shows. We show some examples of area containing quantitative pattern in Fig. 5. Blue circles in Fig. 5 indicate visual features being clustered to a same visual word. It is reasonable for us to believe that these quantitative patterns that have been ignored in previous visual pattern related works does not contain less information than non-quantitative patterns.

To realize the mining of quantitative visual patterns, i.e., visual patterns with duplicated words, such as $\{w_a,w_a,w_b\}$, we preprocess the transactions so that duplicated words in one transaction are distinguishable by the Apriori algorithm [12].

Algorithm 1 shows the rule how we assign new indexes to duplicate visual word. For example, transaction $\{w_a,w_a,w_b,w_a\}$ will be encoded as $\{w_{a1},w_{a2},w_b,w_{a3}\}$ after preprocessing. Note that the "order" of a visual word in our algorithm corresponds to the appearing order of the visual word among the set of duplicate visual words. Thus, the orders of the first, second, and third w_a in the transaction $\{w_{a1},w_{a2},w_b,w_{a3}\}$ are 1, 2, and 3, respectively.

Algorithm 1 Visual word ID pre-processing

```
For i ← 1; i ⩽ MAXWORD; i ++ do
    Count [i] ← 0
end for
for all transactions t ∈ D do
  for i ← 1; i ⩽ sizeof(t); i ++ do
    w_i ← t.item[i];
    count[w_i] ← count[w_i] + 1;
    order ← count[w_i];
    w_i ← w_i + (order − 1) × MAXWORD;
    for all words w ∈ t do
      count[w] ← 0;
    end for
  end for
end for
```

The preprocessed transactions are then used as the input of the Apriori algorithm to generate frequent visual patterns. A simple post processing step inverse to the

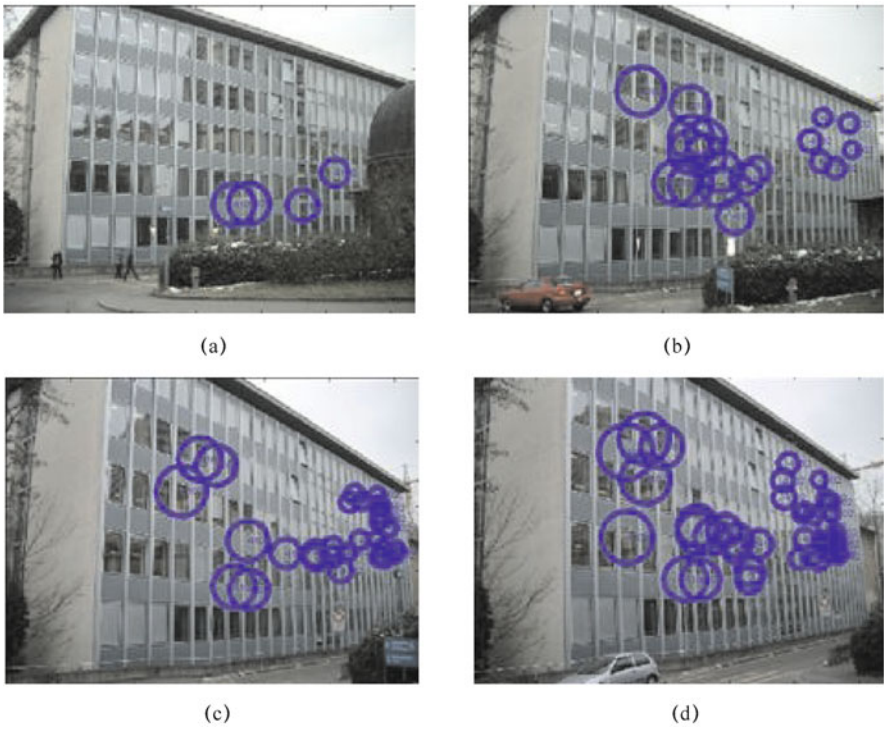

Fig. 5 Case study of quantitative pattern on ZuBuD database. (a) Case 1; (b) case 2; (c) case 3; (d) case 4

preprocessing step decodes the visual patterns from ones like $\{w_a,w_{a2}\}$ to $\{w_a,w_a\}$ as we desired. For those duplicate visual words, their repeating time equals their highest order. For example, $\{w_a,w_{a2}\}$ will be decoded as $\{w_a,w_a\}$, and $\{w_b,w_{a2}\}$ will be decoded as $\{w_b,w_a,w_a\}$.

Finally, we rank all the visual patterns according to their AH. The top K visual patterns consist of visual pattern vocabulary.

4.4 Using pattern in retrieval

During visual pattern mining process, we rank all the patterns according to their AH value. Only the top K patterns will be put into the visual pattern sets. Then, word vectors and pattern vectors are constructed for both database images and the query image. Finally, different combining methods are utilized to obtain the final similarity scores.

4.5 Construct word vector and pattern vector

When visual pattern set is obtained, word vectors and pattern vectors are constructed for both database images and the query image. We introduce the construction method here briefly.

Assume that A is one image, $\boldsymbol{A}_{\mathrm{w}} = \{w_1,w_2,...,w_N\}$ and $\boldsymbol{A}_{\mathrm{p}} = \{p_1,p_2,...,p_M\}$ is word vector and pattern vector of A, respectively; all transactions belonging to A are $\{T_1,T_2,...,T_k\}$. We simply check all these transactions to construct $\boldsymbol{A}_{\mathrm{w}}$ and $\boldsymbol{A}_{\mathrm{p}}$. Take T_i for example, if T_i contains an instance of a certain pattern $p_j = \{w_s,w_t\}$, where w_s and w_t are two different visual words in the dataset, we count one for $\boldsymbol{A}_{\mathrm{p}}(j)$ and update $\boldsymbol{A}_{\mathrm{w}}(j)$ according to all the other words in T_i. By this method, we construct two new vectors for image A.

4.6 Combine word vector and pattern vector

Aiming at using visual pattern to improve image retrieval accuracy, we also want to answer the question which is the best way of combining these two vectors. We compare four different methods here:

1) Pre-rank

First, all images in database are ranked by similarity counted from visual pattern vectors only, and then, visual word vectors are used to re-rank the top K images.

2) Score-merging

A new similarity score is obtained by simply adding the similarity score counted from visual pattern vector to the similarity score obtained by visual word vectors:

$$\mathrm{sim}(\boldsymbol{A},\boldsymbol{B}) = \mathrm{sim}_{\mathrm{w}}(\boldsymbol{A}_{\mathrm{w}},\boldsymbol{B}_{\mathrm{w}}) + \alpha\ \mathrm{sim}_{\mathrm{p}}(\boldsymbol{A}_{\mathrm{p}},\boldsymbol{B}_{\mathrm{p}}),$$

where $\boldsymbol{A}_{\mathrm{w}}$ and $\boldsymbol{A}_{\mathrm{p}}$ are image A's word vector and pattern vector, respectively. $\mathrm{sim}_{\mathrm{w}}()$ is cosine function mentioned in Sect. 3. α is a tunable factor that can change visual pattern's significance in retrieval.

3) Vector-merging

For an image A, word vector $\boldsymbol{A}_{\mathrm{w}} = \{w_1,w_2,...,w_N\}$ and

pattern vector $\boldsymbol{A}_{\mathrm{p}} = \{p_1,p_2,...,p_M\}$ will be merged into one vector $\boldsymbol{A} = \{w_1,w_2,...,w_N,p_1, p_2,...,p_M\}$. Similarity counting method follows the cosine function.

4) Re-rank

This is an inverse method of pre-ranking. First, all images in database are ranked by visual word vector, and then, visual patterns are used to re-rank the top K images.

5 Framework

There are three major parts of our system.

5.1 Database image indexing

First, a visual vocabulary is obtained by clustering detected local features in database images. In this work, difference of Gaussian (DoG) detector and SIFT descriptor [4] are used to generate local features. SIFT descriptors are vector-quantized using hierarchical k-means clustering. All the descriptors in database and query images are assigned to the nearest cluster center to their SIFT descriptors. The use of such a visual vocabulary allows efficient feature matching and captures the variability of a particular feature type. We further mine visual pattern sets based on this visual vocabulary. Modified Apriori algorithm is used as a mining method.

5.2 Query image processing

When a query image enter the system, we detect local features in it and present it as both a word vector and a pattern vector by looking up visual word vocabulary and visual pattern sets, respectively. The query word vector is then convolved with extended similarity function mentioned in Sect. 3.

5.3 Database image ranking

For each image in database, we get word vector and pattern vector, respectively. Four different utilizing methods, as we mentioned in Sect. 4, are tried to combine these two vectors. Finally, top-ranked database images will be returned as near-duplicate image with query image.

6 Experiment

6.1 Dataset and evaluation

We evaluate our image retrieval approach using ZuBuD building image database (http://www.vision.ee.ethz.ch/showroom/zubud/index.en.html) and the University of Kentucky recognition benchmark (UKBench) [14] database. The ZuBuD database consists of color images from 201 buildings in Zurich; each represented by five images acquired at random arbitrary view points under different seasons, weather conditions, and by two different cameras. Besides having variety in illumination conditions as well as scale and orientation changes, these 1005 images also contain partial occlusions and cluttered backgrounds. The authors of the ZuBuD database also created 115 query images of a subset of the same buildings to test the performance of their recognition system. UKBench is a dataset introduced in Ref. [14]. It is recently updated and now contains color images of 2550 classes of four images each. We randomly choose one image from each group as query image and leave other three images as database images. There were 100 queries tested on this database.

To test whether the extended similarity function and visual patterns are helpful in image near-duplicate (IND) retrieval tasks and discuss the insights behind it, we use mean average precision (MAP) for evaluation. MAP emphasizes returning more relevant documents earlier. It is the mean value of the average precisions (APs) computed for each of the queries separately:

$$\mathrm{AP}_q = \frac{\sum_{q=1}^{N_q} \left(P_q(r_q)\mathrm{rel}_q(r_q)\right)}{\#\mathrm{relevent\_image\_of\_Query\_}q}, \tag{7}$$

$$\mathrm{MAP} = \frac{1}{Q}\sum_{q=1}^{Q} \mathrm{AP}_q, \tag{8}$$

where r_q is the rank, N_q is the number of retrieved images for query q, $\mathrm{rel}_q()$ is a binary function on the relevance of a given rank, $P_q()$ is precision at a given cut-off rank, #relevent_image_of_Query_q is the number of relevant images of Query q in dataset, and Q is the total number of queries.

Other evaluation criteria include precision and recall, which are commonly used in information retrieval.

6.2 Results

Some abbreviations are explained as follows:

- BoW: original BoW model
- Q: using patterns including quantitative patterns
- VM: ranking database images using vector-merge method
- ES: with extended similarity function
- NQ: using patterns excluding quantitative patterns
- SM: ranking database images using similarity-merge method

We compare visual pattern's performance on different vocabulary size ranging from 1000 to 10000. Retrieval performance will reach its climax when vocabulary size is large enough; therefore, we show performance on a range including this climax point, i.e., around 4000. All our

experiments use four-layer hierarchical *k*-means, except for experiments in Sect. 6.6.2, which are designed to demonstrate the different result when using variance clustering layers.

Pattern number is about 10% of vocabulary size, which is an experiential setting that will also be demonstrated to be the optimal setting in Sect. 6.6.1.

6.3 Effectiveness of extended similarity function

Figure 6 shows the effectiveness of our proposed extended similarity function. MAP gains an improvement of 2% compared with original BoW model. It is also obvious that MAP gets more improvement on larger vocabulary. This is due to the fact that the quantity of visual synonymy increases with vocabulary size, which has also been shown in Table 1. Same conclusion can be drawn in Fig. 7, which demonstrates that our proposed extended similarity function can further improve retrieval performance even when visual pattern are taken into consideration.

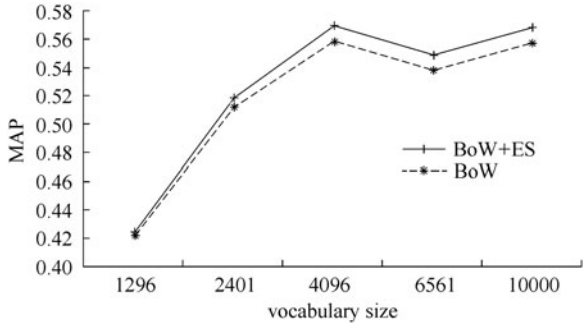

Fig. 6 MAP of using extended similarity function on ZuBuD database

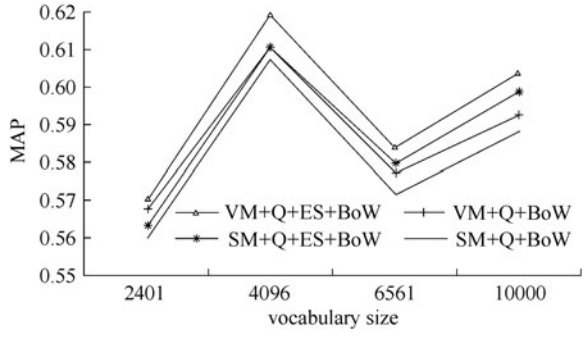

Fig. 7 MAP of using extended similarity function and quantitative patterns on ZuBuD database

6.4 Effectiveness of quantitative pattern

In these experiments, we utilize visual pattern by both similarity-merging method and vector-merging method, which have been described in Sect. 4. Actually, these two merging methods outperform other two methods significantly; therefore, we only show quantitative pattern's effectiveness by these two methods.

To illustrate how the quantitative patterns influence image retrieval performance, we simply compare the results of using quantitative patterns and using non-quantitative patterns in Fig. 8. It is obvious that quantitative pattern means a lot to the performance no matter whether a similarity-merging or vector-merging function is taken.

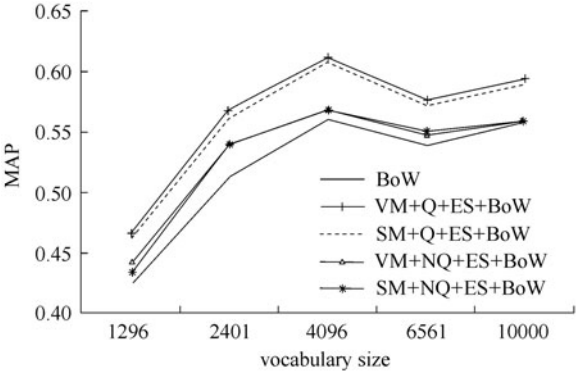

Fig. 8 MAP of using quantitative pattern and using non-quantitative pattern on ZuBuD database

6.5 The best way of using visual pattern

Compared with original BoW model, using pattern in a pre-ranking or re-ranking manner reduces MAP. On the average, MAP will be decreased to around 30%. No matter whether visual pattern similarity is used as a pre-ranking device or a re-ranking device should be based on the assumption that visual word and visual pattern contains comparable information. However, this assumption is definitely not the truth in our experiment. After constructing word vectors and pattern vector, one image could contain visual words 15–20 times of visual pattern; therefore, it is unreasonable to assume that visual pattern vector have equal information with visual word vector. The other two methods of using visual pattern are similarity-merging and vector-merging, which deserve more comparison and discussion.

As we described in Sect. 4, similarity-merging method gives the following similarity function to be used for ranking images:

$$\mathrm{sim}(\boldsymbol{A},\boldsymbol{B}) = \mathrm{sim}_{\mathrm{w}}(\boldsymbol{A}_{\mathrm{w}},\boldsymbol{B}_{\mathrm{w}}) + \alpha\ \mathrm{sim}_{\mathrm{p}}(\boldsymbol{A}_{\mathrm{p}},\boldsymbol{B}_{\mathrm{p}}), \quad (9)$$

where α is a factor weighed with the retrieval performance. Tuning α implies giving visual pattern different weight. When α equals its optimal value 0.1, the similarity-merging method equals can achieve almost same result of vector-merging method, as shown in Fig. 6. The improvement of MAP is around 8%–12%. Compared with similarity-merging method, vector-merging is simpler

because it is parameter-free. It can achieve better results than similarity-merging method without tuning any parameter. In addition, vector-merging method can express the idea that in some sense, visual patterns can be viewed as special visual words. Therefore, we give the conclusion here that vector-merging is the best way among these four methods, not only can it achieve the best performance but also its simplicity. It is also clear in Fig. 8 that VM + Q + ES + BoW achieves the best performance in our experiments. The same result can be obtained from our experiment on the UKBench dataset shown in Fig. 9. On the average, the total improvement is 14.91% on UKBench and 10.8% on ZuBuD.

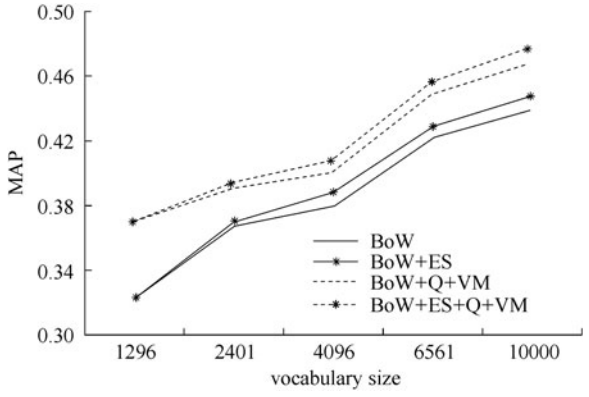

Fig. 9 Results on UKBench database

6.6 Discussion

6.6.1 Pattern number

In the previous experiment, we keep the top K patterns ranked by their AH. Generally, K equals 10% of vocabulary size, which is the optimal setting in our experiment. We demonstrate this in Fig. 10. For the vocabulary size of 6561, performance reaches its climax when the pattern number is 700; while for the vocabulary size of 10000, MAP gets the best performance when the pattern number is 1000.

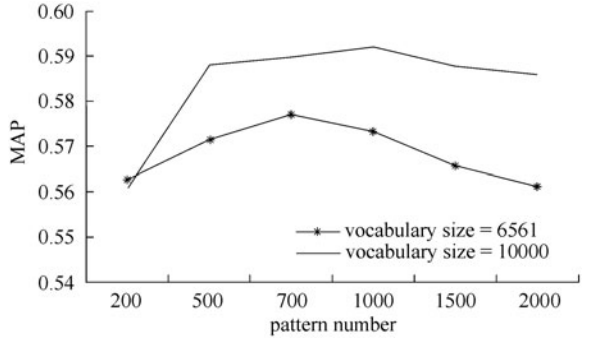

Fig. 10 MAP influenced by pattern number on ZuBuD database

6.6.2 Clustering layer

Visual words are different from text words for they are generated by clustering visual features. We adopt hierarchical k-means [14] to cluster features that has several tunable parameters, e.g., final vocabulary size and layer number. Retrieval performance is not only related to vocabulary size but also cluster layer. In these groups of experiments, we demonstrate our proposed extended similarity function and visual pattern's effectiveness on different cluster layer in Fig. 11. Performances decrease with cluster layer on the whole; however, both extended similarity function and visual pattern's effectiveness increase. For extended similarity function, the improvement ranges from 0.24% (two layers) to 1.76% (six layers); while for visual pattern, it ranges from 4.62 % (two layers) to 11.5% (six layers). Combining these two methods, the improvement could reach 13.36% when six-layer k-means clustering method is employed. It is easy to deduce the reason behind it as more cluster layer generates more visual polysemes and synonyms, thus making our proposed extended similarity function and visual pattern more meaningful in compensating for the loss of performance.

7 Conclusions

BoW model, which is one of the most popular methods in near-duplicate image retrieval, has a fundamental deficiency of semantic gap. Our proposed approach tries to overcome these deficiencies of BoW model by treating the problem as visual synonymy and polysemy issues. We explain the origin of visual synonymy and polysemy and propose different methods to solve these two deficiencies. To solve visual polysemy, we introduce an extended similarity function that is obtained by word correlation matrix. For visual synonymy, we present visual patterns to disambiguate it. We demonstrate that the best way of using visual pattern is vector-merging, which has the strength of effectiveness and simplicity. In addition, we argue that quantitative pattern that has long been ignored in the state-of-art means a lot to image retrieval. We modify original Apriori algorithm to mine quantitative pattern and proved it to be meaningful in our task. On the whole, our proposing method can achieve 12% improvement compared with the already high baseline (around 55% in terms of MAP). This extended similarity function brings around 2% improvement in terms of MAP and 8% improvement in terms of recall. Using visual pattern gives an improvement of around 10% of MAP and 8% of precision.

Future work includes the following:

1) Using other methods, such as statistical method, to construct word correlation matrix and comparing them with our proposed method.

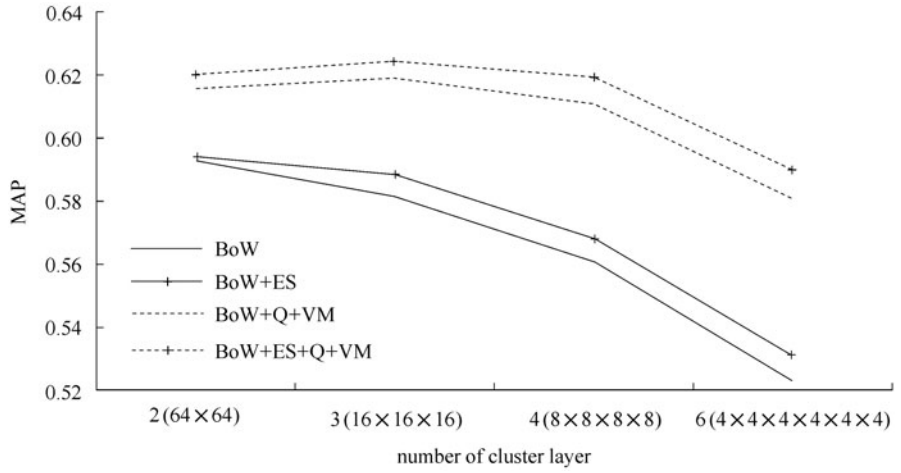

Fig. 11 MAP influenced by cluster layer on ZuBuD database with a vocabulary of 4096 words

2) Explore even better usage of visual pattern in near-duplicate image retrieval and so on.

References

1. Csurka G, Dance C R, Fan L X, Willamowski J, Bray C. Visual categorization with bags of keypoints. In: Proceedings of European Conference on Computer Vision, Workshop on Statistical Learning in Computer Vision. 2004, 1–22
2. Salton G, Wong A, Yang C S. A vector space model for automatic indexing. Communications of the ACM, 1975, 18(11): 613–620
3. Wong S K M, Ziarko W, Wong P C N. Generalized vector space model in information retrieval. In: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval. 1985, 18–25
4. Lowe D G. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004, 60(2): 91–110
5. Schindler G, Brown M, Szelisk R. City-scale location recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2007, 1–7
6. Mitra M, Buckley C, Singhal A, Cardie C. An analysis of statistical and syntactic phrases. In: Proceedings of the 5th International Conference on Recherche d'Information Assistee par Ordinateur. 1997, 200–214
7. Sivic J, Zisserman A. Video data mining using configurations of viewpoint invariant regions. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition. 2004, 1: I-488–I-495
8. Quack T, Ferrari V, Van Gool L. Video mining with frequent itemset configurations. In: Proceedings of the 5th International Conference on Image and Video Retrieval. 2006, 360–369
9. Yuan J, Wu Y, Yang M. Discovery of collocation patterns: from visual words to visual phrases. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2007, 1–8
10. Quack T, Ferrari V, Leibe B, Van Gool L. Efficient mining of frequent and distinctive feature configurations. In: Proceedings of the 11th IEEE International Conference on Computer Vision. 2007, 1–8
11. Srikant R, Agrawal R. Mining quantitative association rules in large relational tables. In: Proceedings of the ACM SIGMOD Conference on Management of Data. 1996, 1–12
12. Agrawal R, Imielinski T, Swami A N. Mining association rules between sets of items in large database. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. 1993, 207–216
13. Deerwester S, Dumais S T, Furnas G W, Landauer T K, Harshman R. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 1990, 41(6): 391–407
14. Nistér D, Stewénius H. Scalable recognition with a vocabulary tree. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2006, 2: 2161–2168