IEICE TRANS. INF. & SYST., VOL.E94–D, NO.10 OCTOBER 2011

PAPER

Scalable Object Discovery: A Hash-Based Approach to Clustering Co-occurring Visual Words

Gibran FUENTES PINEDA[†a)], **Hisashi KOGA**[†], *Nonmembers*, *and* **Toshinori WATANABE**[†], *Member*

SUMMARY We present a scalable approach to automatically discovering particular objects (as opposed to object categories) from a set of images. The basic idea is to search for local image features that consistently appear in the same images under the assumption that such co-occurring features underlie the same object. We first represent each image in the set as a set of visual words (vector quantized local image features) and construct an inverted file to memorize the set of images in which each visual word appears. Then, our object discovery method proceeds by searching the inverted file and extracting visual word sets whose elements tend to appear in the same images; such visual word sets are called co-occurring word sets. Because of unstable and polysemous visual words, a co-occurring word set typically represents only a part of an object. We observe that co-occurring word sets associated with the same object often share many visual words with one another. Hence, to obtain the object models, we further cluster highly overlapping co-occurring word sets in an agglomerative manner. Remarkably, we accelerate both extraction and clustering of co-occurring word sets by Min-Hashing. We show that the models generated by our method can effectively discriminate particular objects. We demonstrate our method on the Oxford buildings dataset. In a quantitative evaluation using a set of ground truth landmarks, our method achieved higher scores than the state-of-the-art methods.

key words: *object discovery, large-scale image mining, bag-of-features, Min-Hashing, agglomerative clustering*

1. Introduction

In most object recognition methods, object models are acquired by some human supervision, e.g. manual object segmentation, image annotations or specifying the number of object kinds. However, even a little human supervision may become extremely expensive, when dealing with large image sets. For this reason, many of the current object recognition methods just can handle small image sets and a limited number of objects, because their performance deteriorates as the number of images and the dimensionality of image representation increases.

Modern feature detectors and descriptors have boosted the development of efficient techniques to represent large sets of images and videos. In particular, the bag-of-features (BoF) approach [1], which represents an image as a set of visual words (vector quantized local image descriptors), has been widely adopted due to its simplicity, flexibility and excellent performance. Furthermore, the BoF representation is robust to occlusion, clutter and changes in scale, illumination and viewpoint. Thanks to these characteristics, several state-of-the-art object/image retrieval and image clustering systems are built upon the BoF approach. However, most of these systems only compute global similarity between images by counting the number of shared visual words. Therefore, their ability to recognize the same objects is limited, especially when the objects do not cover the entire image in a complete form.

The objective of this work is to discover particular objects (as opposed to object categories) from large unordered image sets without supervision by mining visual words that effectively discriminate a particular object. This is a challenging task because the image set consists of an overwhelming number of images, the content is highly diverse and the appearance of the objects varies greatly due to high clutter, occlusion and extreme changes in scale, illumination and viewpoint. Moreover, because of unstable visual words, very similar instances of the same object can have only a few common visual words [2]. The unsupervised discovery of objects can be useful in many applications such as generating summaries from image sets, organizing images based on the objects they contain and improving the efficiency of object/image retrieval systems.

To discover objects, we pay attention to co-occurring local image features. The rationale is that features that belong to the same object tend to appear together much more often than those belonging to different objects. Hence, our method yields object models by extracting visual words that appear together in the same images. In particular, the discovery process consists of two steps. In the first step, we search for co-occurring visual words that consistently appear in the same images on the inverted file of the BoF models; a set of such visual words is called a co-occurring word set. Here, the inverted file of the BoF models is a data structure which, for each visual word, stores a set of images which contains it. Because of unstable visual words and polysemous visual words (visual words associated with multiple objects), co-occurring word sets typically represent only a part of an object. We further observe that co-occurring word sets associated with the same object often share many visual words with one another. Therefore, in the second step, object models are formed by clustering co-occurring word sets sharing many visual words in an agglomerative manner. Remarkably, we accelerate both extraction and clustering of co-occurring word sets by Min-Hashing.

Manuscript received January 17, 2011.
Manuscript revised June 13, 2011.

[†]The authors are with the Graduate School of Information Systems, The University of Electro-Communications, Chofu-shi, 182–8585 Japan.
a) E-mail: gibranfp@sd.is.uec.ac.jp

DOI: 10.1587/transinf.E94.D.2024

Copyright © 2011 The Institute of Electronics, Information and Communication Engineers

1.1 Related Works

Chum and Matas [3] proposed a fast algorithm for discovering related images based on an extension of Min-Hashing [4]. This algorithm hashes images to find similar image pairs and then form clusters of spatially related images. It is easy to see that as the number of common visual words between two images decreases, they are unlikely to be treated as similar. This is a disadvantage that limits the ability to cluster images of the same object, especially when the object occupies a small portion of an image. Our method differs from [3] in that we apply Min-Hashing to the inverted file to extract co-occurring visual words whereas [3] applies it to the BoF models to find similar image pairs. Note that our method utilizes Min-Hashing also to clustering co-occurring word sets.

Motivated by the success of topic discovery from documents, many researchers have relied on latent variable models such as PLSA [5] and LDA [6] to discover particular objects [7], [8] as well as object categories [9] from images. Latent variable models represent each image as a mixture of K topics where each topic corresponds to a single object class. One important limitation of these methods is that the number of topics K must be given a priori. Even slightly different choices of K might lead to quite different results. This limitation becomes worse when the image set is large and diverse because the number of topics can be hard to infer. Furthermore, as it is very time consuming to estimate the model parameters, latent variable models are not easily scalable to large databases.

Similar to our work, Philbin et al. [10], [11] mine objects from large image sets. They first use image retrieval techniques to build a matching graph which divides the image set into groups of spatially related images. Then, [10] performs spectral clustering to partition the groups that contain multiple disjoint objects, whereas [11] employs gLDA (a variant of LDA that takes into account geometric information) on each group to generate object models. An important drawback of these methods is that the construction of the matching graph is very expensive. In addition, applying spectral clustering or gLDA to each group of the matching graph is also very time consuming, especially when there are large groups. In Sect. 4, we compare quantitatively our method with those in [10], [11].

Bhatti and Hanbury [12] exploited the relative co-occurrence of visual words for enhancing the discrimination power of the BoF models. In their work, a new visual vocabulary is constructed by measuring the spatial relation between all possible pairs of visual words. Then, the object models are created in a supervised fashion by using Naive Bayes and SVM. Unfortunately, constructing a new visual vocabulary can be prohibitively expensive for large vocabularies, because the spatial relation must be computed between all possible pairs of visual words. Thus, this method is not suitable for handling large image sets. Our method differs greatly from such method in that our method extracts object models without supervision. In addition, so as to shrink the execution time, our method does not consider the spatial relation between visual words, but exploits the dependency of occurrence of multiple visual words.

1.2 Outline of the Paper

This paper is an extension of our previous paper [13]. The current version presents a more detailed description and analysis of the object discovery method and provides a more extensive experimental evaluation using a benchmark dataset. It also incorporates a mechanism for pruning co-occurring word sets. The content of the paper is organized as follows. Section 2 gives an overview of Min-Hashing. We introduce our object discovery method in Sect. 3. In Sect. 4, we present experimental results on the Oxford buildings dataset. Finally, Sect. 5 gives the concluding remarks.

2. Min-Hashing

Min-Hashing [14] is a randomized algorithm for efficiently computing the *Jaccard similarity* between sets. In this section, we give a brief overview of Min-Hashing. For a more detailed explanation, the reader is referred to the works of Cohen et al. [14] and Broder [15].

Let X_i and X_j be a pair of sets whose elements are chosen from M different items $x_1, x_2, \ldots, x_M$. The Jaccard similarity between X_i and X_j is defined as

$$sim(X_i, X_j) = \frac{| X_i \cap X_j |}{| X_i \cup X_j |} \in [0, 1]. \quad (1)$$

In Min-Hashing, we first select a permutation π of the ordered items $x_1, x_2, \ldots, x_M$ randomly. From the viewpoint of combinatorics, since the number of different items is M, $M!$ permutations of the items are possible. Here, the permutation π is selected randomly from these $M!$ different permutations. After π is determined, the min-hash value for X_i becomes its first element after X_i is permuted according to π. That is,

$$h(X_i) = min(\pi(X_i)), \quad (2)$$

where $\pi(X_i)$ denotes the permutation of X_i under π. For example, suppose that $\pi = \{x_2, x_3, x_1\}$ is a random permutation of the ordered items x_1, x_2, x_3. Now consider two sets $X_i = \{x_1, x_2, x_3\}$ and $X_j = \{x_1, x_3\}$. The first element of X_i permuted according to π is x_2 whereas the first element of X_j permuted according to π is x_3. Therefore, $h(X_i) = x_2$ and $h(X_j) = x_3$. In practice, the random permutation of the items is implemented by assigning a random number to each item. Then, the min-hash value of a set is obtained by finding the minimum of the numbers assigned to its elements.

In Min-Hashing, the probability that X_i and X_j take the same min-hash value is known to be equal to their Jaccard similarity [14]. Namely

$$P[h(X_i) = h(X_j)] = sim(X_i, X_j). \quad (3)$$

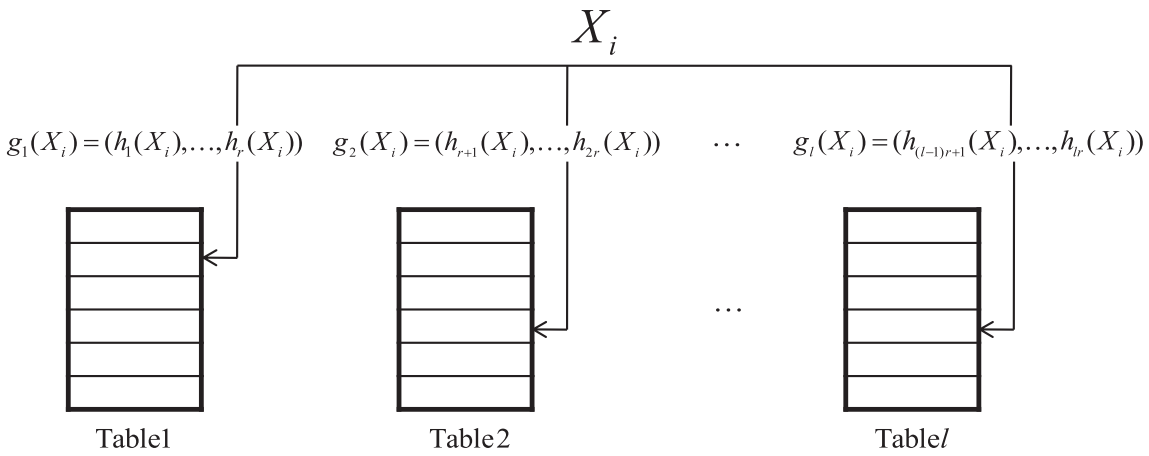

Fig. 1 Construction of hash tables in Min-Hashing.

In the above example, the probability that X_i and X_j take the same min-hash value is 2/3.

Hence, similar sets will have the same min-hash value with high probability. However, because Min-hashing is a probabilistic method, false negatives (similar sets with different min-hash values) and false positives (dissimilar sets with the same min-hash value) are likely to happen. To overcome this problem, multiple min-hash values are computed to judge whether two sets are similar or not, where each min-hash value is obtained under a different permutation selected independently at random from the $M!$ permutations.

In particular, Min-Hashing builds a hash function g which returns the concatenation of r min-hash values as its hash value. Then, l instances $g_1, g_2, \cdots, g_l$ of such g are prepared. The hash values of X_i for the l hash functions $g_1, g_2, \cdots, g_l$ are defined as follows.

$$\begin{aligned} &g_1(X_i) = (h_1(X_i), h_2(X_i), \ldots, h_r(X_i)) \\ &g_2(X_i) = (h_{r+1}(X_i), h_{r+2}(X_i), \ldots, h_{2r}(X_i)) \\ &\cdots \\ &g_l(X_i) = (h_{(l-1)*r+1}(X_i), h_{(l-1)*r+2}(X_i), \ldots, h_{l*r}(X_i)) \end{aligned} \quad . \quad (4)$$

Here $h_j(X_i)$ denotes the j-th min-hash values. Note that $r \cdot l$ min-hash values are used in total, as r min-hash values are necessary to compute each g_i ($1 \le i \le l$). Because one hash table is created for each g_i, l hash tables are constructed in total as shown in Fig. 1. A pair of sets X_i and X_j are stored in the same hash bucket on the k-th hash table, if $g_k(X_i) = g_k(X_j)$.

In the Min-Hashing scheme, highly similar sets are expected to enter the same hash bucket at least on one hash table. The probability that two sets X_i, X_j have the same hash value for g_k is expressed as

$$P[g_k(X_i) = g_k(X_j)] = sim(X_i, X_j)^r, \quad (5)$$

because all of the r min-hash values consisting g_k have to be the same. Because $(1-sim(X_i, X_j)^r)^l$ presents the probability that X_i and X_j take different hash values for all the l hash functions, the probability that X_i and X_j are stored in the same hash bucket at least on one hash table is expressed as Eq. (6).

$$P_{collision}[X_i, X_j] = 1 - (1 - sim(X_i, X_j)^r)^l. \quad (6)$$

By choosing r and l properly, this probability approximates a unit step function such that

$$P_{collision}[X_i, X_j] \approx \begin{cases} 1, & \text{if } sim(X_i, X_j) \ge s* \\ 0, & \text{if } sim(X_i, X_j) < s* \end{cases}. \quad (7)$$

Here $s*$ is a threshold parameter. That is, the probability of collision is close to 1 if $sim(X_i, X_j) \ge s*$ and close to 0 if $sim(X_i, X_j) < s*$. In this way, we can use Min-Hashing to retrieve only a pair of sets whose similarity is greater than $s*$.

3. Object Discovery

This section introduces our method for discovering objects from a given set of images $\Sigma = \{I_1, I_2, \ldots, I_N\}$. The object discovery is realized by executing the next three tasks:

1. We represent each image in Σ with a BoF model and indexing Σ with an inverted file
2. Co-occurring word sets are mined from the inverted file.
3. Object models are derived by clustering co-occurring word sets agglomeratively based on the number of common visual words between co-occurring word sets.

The object discovery process is overviewed in Fig. 2. Remarkably, our method exploits co-occurrence of visual words to generate object models automatically without supervision. In addition, by clustering similar co-occurring word sets agglomeratively in the final task, our method does not demand the number of clusters (kinds of objects) to be specified. In the following, we discuss in detail each of the tasks in our method.

3.1 Bag-of-Features and Inverted File

We follow the BoF approach to represent each image in Σ.

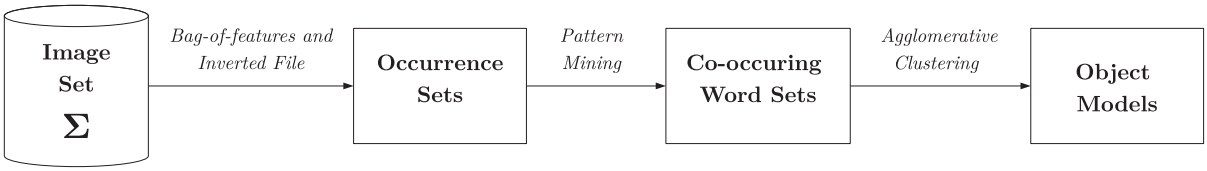

Fig. 2 Overview of the object discovery process.

We further index the set of images with an inverted file. Next, we review the steps to obtain such a representation.

1. Local image features are extracted for each image in Σ by detecting affine covariant regions such as MSER [16] and Hessian Affine [17].
2. Each local image feature is described with a SIFT descriptor [18] and represented as a 128-dimensional vector.
3. A vocabulary of *visual words* $V = \{v_1, \ldots, v_M\}$ is constructed by clustering all the local image features in Σ. Here, each visual word is also represented by a 128-dimensional vector.
4. Each local image feature is assigned the ID of the nearest visual word. In the standard BoF, each image is described as a frequency vector of visual words. However, as our method only analyzes the occurrence pattern of the visual words, we only record their presence or absence. Thus, in our method, each image is described as a binary vector, not as a frequency vector. This results in a more compact representation with a good discrimination power for large vocabularies. In fact, it has been shown [19] that for vocabularies larger than 10000 visual words, the binary BoF slightly outperforms the standard BoF in search quality.
5. We discard very rare and very common visual words by using a stop list.
6. Images are further indexed with an inverted file structure. For each visual word v_i, the inverted file stores the set of the images in which v_i appears. We denote the set of images containing v_i by $\hat{v_i}$ and refer to it as the *occurrence set* of v_i. $\hat{v_i}$ becomes a subset of the set of N images $\{I_1, I_2, \ldots, I_N\}$.

3.2 Co-occurring Word Set Mining

Now that each visual word v_i is associated with the occurrence set $\hat{v_i}$, we can compute the similarity between v_i and v_j by applying Min-Hashing to $\hat{v_i}$ and $\hat{v_j}$. Since $\hat{v_i}$ presents the set of images in which v_i occurs, the Jaccard similarity $sim(\hat{v_i}, \hat{v_j})$ measures how often v_i and v_j co-occur in the identical images. So, for a given visual word v_i, we can exploit Min-Hashing to search other visual words which tend to co-occur together with v_i in the identical images.

The min-hash value of a visual word v_i is defined as

$$h(v_i) = min(\pi(\hat{v_i})). \tag{8}$$

As mentioned in Sect. 2, we rely on multiple min-hash functions chosen independently at random. That is, we construct l hash functions g_i $(1 \leq i \leq l)$ each of which computes its hash value by concatenating r min-hash values. A set of visual words which enter the same hash bucket on one of the hash tables are called a *co-occurring word set* and denoted by ϕ in this paper. Here, one co-occurring word set ϕ is derived from one hash bucket storing multiple visual words. We expect that discriminative visual words that belong to the same object enter the same hash bucket and form a co-occurring word set ϕ, as they should appear in the same images containing the object. By contrast, unrelated visual words from different objects will not be stored in the same hash buckets.

Given a set of d visuals words $\{v^1, v^2, \ldots v^d\}$, the probability that all the d visual words take the same min-hash value for a single min-hash function h, i.e, $P[h(v^1) = h(v^2) = \cdots = h(v^d)]$ is calculated as described in Eq. (9).

$$P[h(v^1) = h(v^2) = \cdots = h(v^d)] = \frac{|\hat{v^1} \cap \hat{v^2} \cap \cdots \cap \hat{v^d}|}{|\hat{v^1} \cup \hat{v^2} \cup \cdots \cup \hat{v^d}|}. \tag{9}$$

In Eq. (9), the numerator becomes the number of the images which contain all the d visual words, whereas the denominator corresponds to the number of the images which include at least one of the d visual words. As the visual words appear in the same images more frequently, the value of Eq. (9) increases, since its numerator becomes larger. This implies that the d visual words are more likely to become a co-occurrence word set, as their appearance patterns in the image set Σ grows more positively correlated.

3.2.1 Pruning

Due to the random nature of Min-Hashing, some co-occurring word sets can contain noisy (unrelated) visual words. To get rid of such visual words, we perform the following pruning step. Given a co-occurring word set denoted by ϕ, we first scan the inverted file to obtain a list of images $Q(\phi)$ that contains at least $\alpha|\phi|$ visual words in ϕ $(0 < \alpha \leq 1)$. Then, the visual words that occur in less than $\beta|Q(\phi)|$ images of $Q(\phi)$ $(0 < \beta \leq 1)$ are discarded from ϕ. Finally, we remove ϕ completely if it contains very few visual words after discarding visual words. We also remove ϕ if $|Q(\phi)|$ is small as it may contain visual words that originate from different objects and that appear together incidentally.

3.3 Agglomerative Clustering

Because of unstable and polysemous visual words, a co-

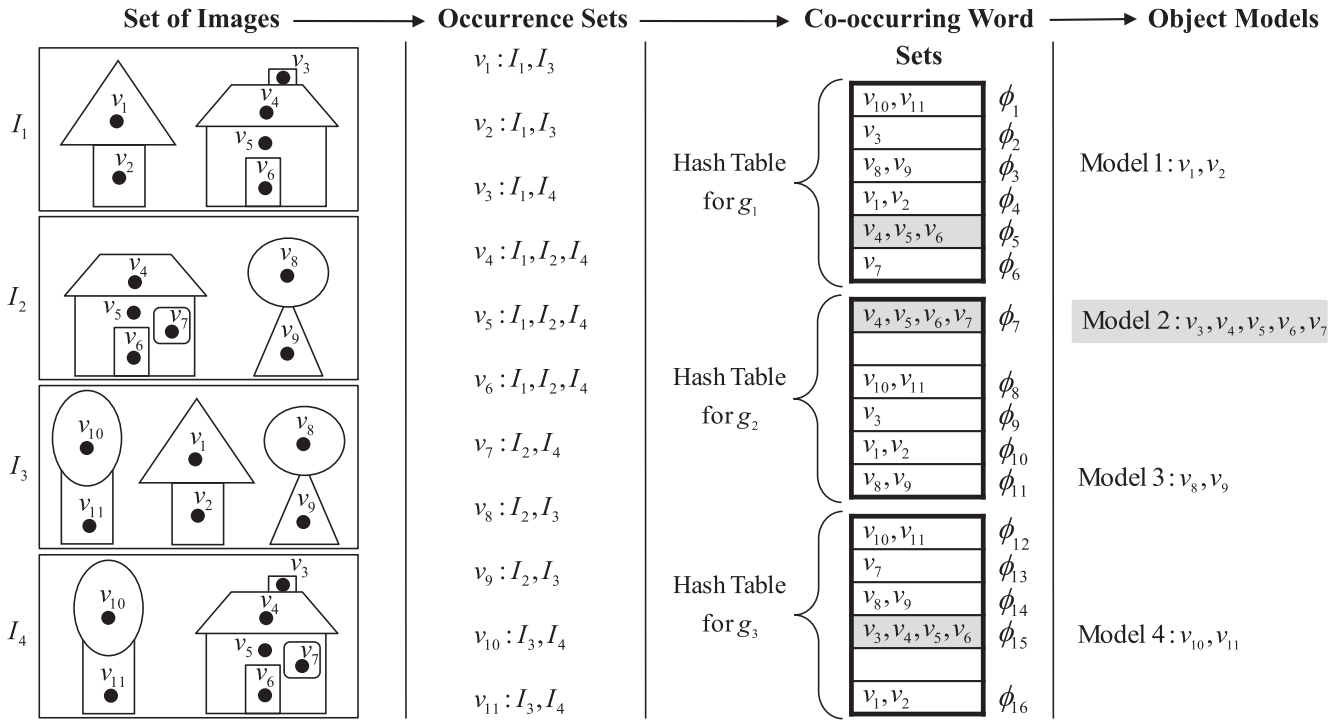

Fig. 3 Toy-example of our object discovery method.

occurring word set will correspond to only a part of the entire object model. By contrast, if a set of visual words contained in the same object are highly stable, they will appear in multiple co-occurring word sets.

Let us illustrate this phenomenon with the toy-example in Fig. 3. This example consists of 4 images and a visual vocabulary of 11 visual words. Each visual word v_i is registered to the 3 hash tables corresponding to the hash functions g_1, g_2 and g_3 by computing the hash value of the occurrence set $\hat{v}_i$. Then, 16 co-occurring word sets from ϕ_1 to ϕ_{16} are extracted from the hash tables. As we can observe, stable visual words in the same object are mapped to the same co-occurring word set often. For example, consider the object "house" composed of the visual words v_3, v_4, v_5, v_6 and v_7. As v_4, v_5 and v_6 always appear together, they are included in the same co-occurring word set three times (ϕ_5, ϕ_7 and ϕ_{15}). On the other hand, unstable visual words are mapped to different co-occurring word sets, even if they belong to the same object. In Fig. 3, v_3 and v_7 are never contained in the same co-occurring word set because they appear together only once in I_4. We can also observe that ϕ_5, ϕ_7 and ϕ_{15} share the stable visual words v_4, v_5 and v_6 and also contain other informative visual words (v_3 and v_7).

Motivated by the above observation, so as to obtain more representative object models, we merge co-occurring word sets that share many common visual words in an agglomerative manner. Because of agglomerative clustering, the number of object kinds need not be specified in our method. Let ϕ_i and ϕ_j be two co-occurring word sets. Note that the elements of the two sets are visual words. We measure the degree of how many visual words are shared between ϕ_i and ϕ_j by their overlap coefficient in Eq. (10).

$$ovr(\phi_i, \phi_j) = \frac{| \phi_i \cap \phi_j |}{min(| \phi_i |, | \phi_j |)} \in [0, 1]. \tag{10}$$

Then, if $ovr(\phi_i, \phi_j) > \epsilon$, we unify ϕ_i and ϕ_j to the same cluster, where ϵ is a parameter of the algorithm.

We can rely on Min-Hashing to find the co-occurring word sets to be merged promptly. Since

$$\begin{aligned} ovr(\phi_i, \phi_j) &= \frac{| \phi_i \cap \phi_j |}{min(| \phi_i |, | \phi_j |)} \geq \frac{| \phi_i \cap \phi_j |}{| \phi_i \cup \phi_j |} \\ &= sim(\phi_i, \phi_j), \end{aligned}$$

a pair of co-occurring word sets whose Jaccard similarity is high will also have a large overlap coefficient. Hence, we may judge whether a pair of co-occurring word sets potentially take a high overlap coefficient from the fact that they enter the same hash bucket in Min-Hashing. This strategy avoids the overhead to compute the overlap coefficient between all the pairs of co-occurring word sets. We remark here that Min-Hashing is applied to the set of visual words in this step, whereas it is applied to the set of images in the co-occurring word set mining in Sect. 3.2. The min-hash value for ϕ_i becomes its first visual word after the order of all the visual words is permuted by the permutation rule π randomly chosen. That is,

$$h(\phi_i) = min(\pi(\phi_i)). \tag{11}$$

Again, we use multiple min-hash values to construct l hash tables. Two co-occurring word sets that share many visual words are expected to enter the same hash bucket at least on

one hash table.

Our algorithm to cluster co-occurring word sets agglomeratively consists of the next 5 steps.

1. Each co-occurring word set is stored into l hash tables.
2. If a pair of co-occurring word sets ϕ_i, ϕ_j are stored in the same hash bucket on some hash table, they are regarded as a candidate pair to be merged.
3. For every candidate pair of co-occurring word set (ϕ_i, ϕ_j), we compute their overlap coefficient as

$$ovr(\phi_i, \phi_j) = \frac{| \phi_i \cap \phi_j |}{min(| \phi_i |, | \phi_j |)} \in [0, 1].$$

4. We construct a graph G such that each co-occurring word set ϕ_i becomes a node and an edge is built between a candidate pair of co-occurring word sets ϕ_i, ϕ_j with $ovr(\phi_i, \phi_j) > \epsilon$.
5. We compute all the connected components in G. Co-occurring word sets (i.e. vertices) belonging to the same connected component are merged into a single cluster and becomes the final object model.

With this algorithm, chains of co-occurring word set pairs with high overlap coefficient are merged into the same cluster. As a result, co-occurring word sets associated with the same object will belong to the same cluster even if they share very few or no visual words, so long as they are members of the chain. For example, consider three co-occurring word sets ϕ_i, ϕ_j and ϕ_k associated with the same object. Even if ϕ_i and ϕ_j do not share visual words at all, they will be merged into the same cluster, in case ϕ_k shares many visual words with both ϕ_i and ϕ_j. In general, for any co-occurring word set in a cluster, there exists at least one co-occurring word set in the same cluster with which it has an overlap coefficient greater than ϵ. Conversely, two co-occurring word sets have an overlap coefficient less than ϵ, if they belong to different clusters.

In the example in Fig. 2, the agglomerative clustering on the co-occurring word sets produces 4 object models (from Model 1 to Model 4). Here, ϕ_5, ϕ_7 and ϕ_{15} are merged into the same cluster to form Model 3, because they share the stable visual words v_4, v_5 and v_6. In this case, the object model consists of the visual words contained in either ϕ_5, ϕ_7 or ϕ_{15}, i.e., v_3, v_4, v_5, v_6 and v_7. Despite v_3 and v_7 are never contained in the same co-occurring word set, they are correctly assigned to the same object model by the agglomerative clustering.

3.4 Retrieval

Because our method generates object models by merging co-occurring word sets, they are represented as a set of visual words. Since images are also represented as a set of visual words in the BOF model, we can determine whether a image contains a specific object from the number of visual words shared between the object model and the image. Especially, we can efficiently identify all the images that share visual words with the object model by searching the occurrence sets of the visual words in the object model. Next, by investigating the number of shared visual words for these images, we retrieve images that share many visual words with the object model and therefore are likely to contain the object. The retrieved images can be further ranked according to the number of shared visual words in order to show the most relevant images first.

3.5 Scalability

In order to achieve scalability with regard to execution speed, we generate object models by simply analyzing the occurrence pattern of visual words. In fact, our method only searches for similar occurrence sets on the inverted file. This contrasts to other methods that adopt expensive learning algorithms. In addition, the most time-consuming tasks in our method, namely mining and clustering co-occurring word sets, are efficiently performed by Min-Hashing, which has proved to be particularly suitable for handling large datasets (see [3], [20], [21]). The time complexity to compute a min-hash value for a set is linear to the number of elements in the set, since we need to find the minimum from the numbers assigned to all the elements. Now, consider the time complexity for the co-occurrence set mining. In the co-occurrence set mining, the time to compute $r \cdot l$ min-hash values for $|V|$ visual words becomes $O(r \cdot l \cdot W \cdot |V|)$, where W is the average number of images in the occurrence sets. In addition, before the computation of Min-Hash values, a time of $O(r \cdot l \cdot |\Sigma|)$ is incurred to generate $r \cdot l$ randomly chosen permutations for the image set Σ. Therefore, the total time complexity for the co-occurrence set mining grows $O(r \cdot l \cdot (W \cdot |V| + |\Sigma|))$. Because $W \ll |\Sigma|$ in general, this time complexity is linear to the number of images, which shows the scalability of our method. On the other hand, as object models are represented as a set of visual words, we can also retrieve the images that contain a particular object quite fast by searching the occurrence of the object model in the inverted file as explained in Sect. 3.4.

As for memory consumption, Min-Hashing is pointed out that it consumes much memory to store all the hash tables. However, both for mining and clustering co-occurring word sets, we only need to store one hash table at a time. Hence, we can avoid the high space complexity often associated with Min-Hashing.

Thus, our method can be applied to both large databases and large visual vocabularies.

4. Experiments

In this section, we demonstrate our method on the Oxford buildings dataset [22]. We first evaluate our results qualitatively by visually examining the discovered objects. In particular, we analyze the meaningfulness and discrimination power of the generated object models. We also carry out a quantitative evaluation using a set of ground truth landmarks and compare our results with the state-of-the-art. Finally, we

Fig. 4 Image samples from the Oxford buildings dataset.

analyze the time and space efficiency of our method.

4.1 Setup

4.1.1 Oxford Buildings Dataset

This dataset consists of 5062 images retrieved from Flickr [23] using particular Oxford landmarks as queries (e.g. "All Souls Oxford"). Image samples from the Oxford buildings dataset are shown in Fig. 4. Note that due to inaccurate annotations, several images unrelated to the Oxford landmarks (which serve as distractors) are also contained in the dataset. For each image, affine covariant hessian regions [17] are detected. Each of the detected regions is represented as a SIFT vector [18]. The total number of the detected regions over all the images are 16,334,970. These 16 million SIFT vectors are classified into 1 million visual words by the approximate k-means clustering of Philbin et al. [24]. The reason why we set the size of visual vocabularies to 1 million is that [24] reported that this value yields the best performance. In the experiment, we relied on the files available at [22] which contain the precomputed visual word IDs and geometries to construct the BoF models and the inverted file. Visual words that occur in more than 30% or less than 0.1% of the images in the dataset were discarded by our stop list.

Manually generated annotations for the occurrence of 11 Oxford landmarks (see Fig. 5) are also provided as the ground truth at [22]. In addition, images with the same landmark annotation are assigned one of the following three labels.

- *Good:* a nice, clear picture of the object.
- *OK:* more than 25% of the object is clearly visible.
- *Junk:* less than 25% of the object is visible, or there is a very high level of occlusion or distortion.

4.1.2 Parameter Tunings

The parameters of our method are set as follows. In co-occurring word set mining, with respect to Min-Hashing, the number of hash table l is 500 and each g_i $(1 \leq i \leq l)$ is built by concatenating $r = 4$ min-hash values.

In pruning co-occurring word sets, $\alpha = 0.7$ and $\beta = 0.8$. Furthermore, co-occurring word sets are removed if they contain less than 3 visual words or appear in fewer than 3 images.

For the agglomerative clustering, as for Min-Hashing, $l = 255$ and $r = 3$. The threshold ϵ for the overlap coefficient is set to 0.6.

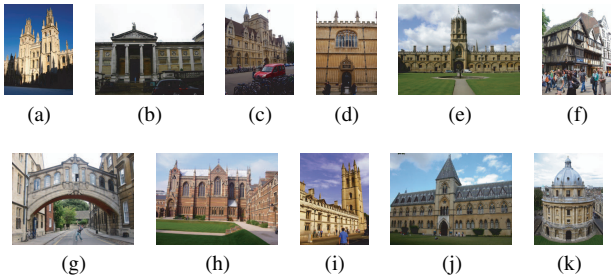

Fig. 5 Ground truth landmarks of the Oxford buildings dataset: (a) All souls, (b) Ashmolean, (c) Balliol, (d) Bodleian, (e) Christ Church, (f) Cornmarket, (g) Hertford, (h) Keble, (i) Magdalen, (j) Pitt Rivers and (k) Radcliffe Camera.

Fig. 6 Top-10 ranked objects. The rank of the object is shown under each image.

4.1.3 Rankings

We define two kinds of rankings to examine and evaluate our results: one over the images that contain the discovered object and another over the discovered objects themselves. For the image ranking, we use each object model (set of visual words) to query the image set through the inverted file. Each query yields a list of images that contains a particular object. Then, the images in the list are ranked based on the number of matched visual words so that more relevant images have a higher rank. For the object ranking, we rank the discovered objects according to the size of their models (that is, the number of visual words) so that more representative objects have a higher rank. Figure 6 illustrates the top-10 objects in the object ranking. Interestingly, the top-5 objects correspond to ground truth landmarks (compare Fig. 5 and Fig. 6).

4.1.4 Methodology

In our experiments, we apply our object discovery method to all the 5,062 images of the Oxford buildings database to

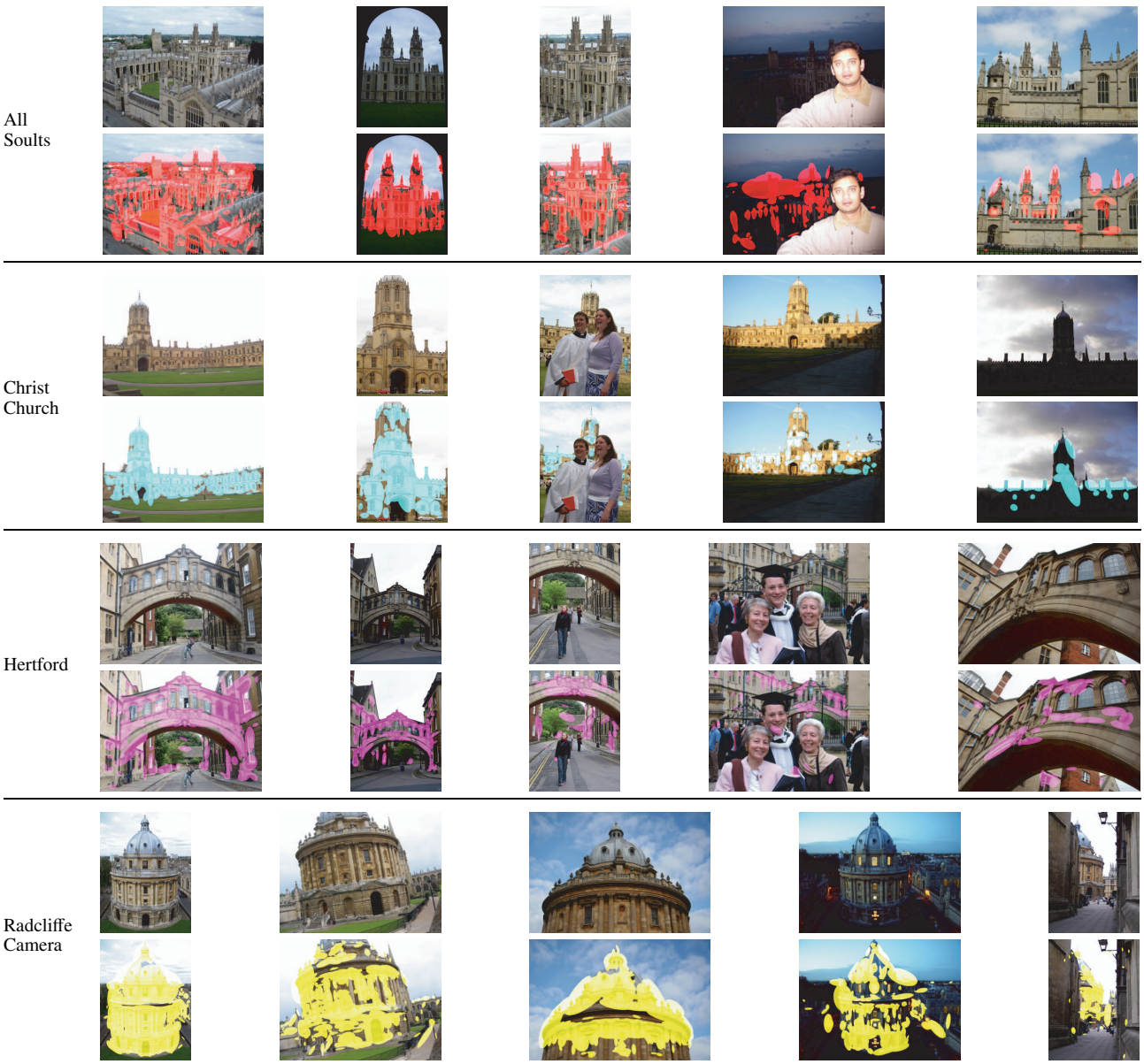

Fig. 7 Sample images of four objects corresponding to ground truth landmarks. The original images (top) and the images with the matched affine covariant regions displayed (bottom) are presented for each object.

extract object models. Then, we use each object model to retrieve the images with the corresponding object. We further rank the retrieved images according to the ranking in Sect. 4.1.3. We will confirm the meaningfulness and robustness of derived object models from the fact that the objects annotated by human are discovered with high accuracy by using the object models. Here, the accuracy is evaluated by the ranking result.

We do not split the dataset in a training and a test set. However, this is also the case for other state-of-the-art methods such as [10] and [11]. Because our primary goal is to extract meaningful object models from a set of images automatically, our experiments focus on automatic object discovery rather than on the ability to recognize unseen new views of the objects.

4.2 Results

4.2.1 Qualitative Evaluation

Several different objects were discovered by our method, including objects corresponding to the 11 ground truth landmarks[†]. Figure 7 shows typical samples of the top ranked images associated with All Souls, Christ Church, Hertford and Radcliffe Camera. The samples displayed in Fig. 7 are presented in descending order of rank: from the top-ranked images (left) to lower-ranked images (right). Although not

[†]Some ground truth landmarks had more than one associated object.

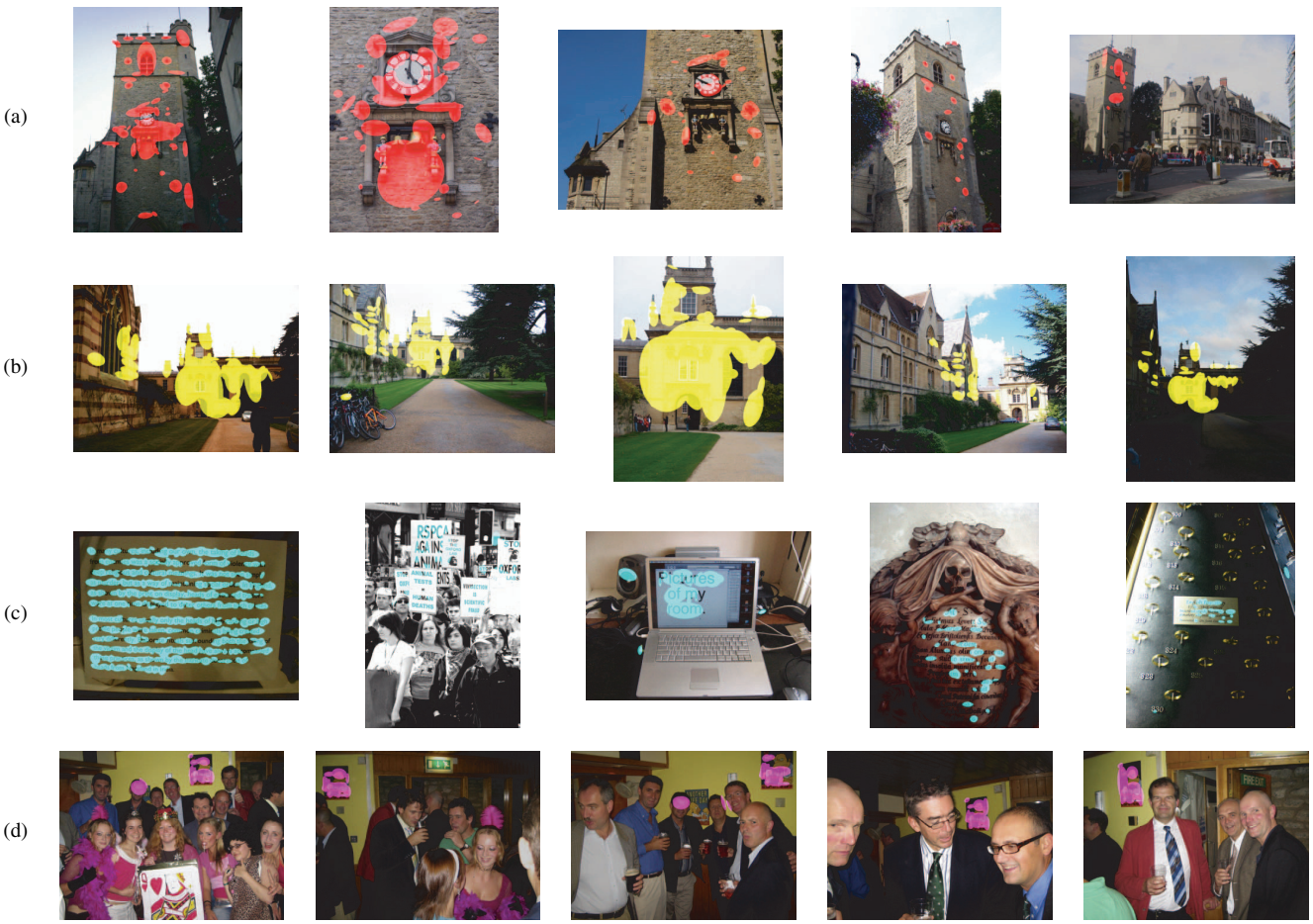

Fig. 8 Sample images of four objects not associated with ground truth landmarks: (a) St Michael at the North Gate, (b) Trinity College, (c) black letters over light background and (c) a cartoon picture on a wall.

presented in this paper, all the high-ranked images are similar to the examples shown here. Note that the matched affine covariant features within each image are correctly localized on the corresponding object (even in the lower-ranked images) despite occlusions, clutter and extreme variations of scale, illumination and viewpoint. These examples demonstrate the meaningfulness and robustness of the object models. A quantitative evaluation using the ground truth landmarks is given in Sect. 4.2.2.

As mentioned before, in our method the number of object kinds is not fixed but rather depends on the correlation of the visual word occurrences. As a consequence, many objects different from the ground truth were also discovered. Four examples of such objects are illustrated in Fig. 8. The rows (a) and (b) correspond to other Oxford landmarks whereas (c) and (d) rows are non-building objects, namely dark letters over light background and a cartoon picture on a wall. Notice that the cartoon picture is quite small relatively to the image size. This shows that our method can discover objects even if they cover only a small portion of the images.

Remarkably, different objects that appear in some images together were correctly discriminated (see Fig. 9). Again, the matched affine covariant features are mostly localized on the corresponding object, which shows that our method generates highly discriminative models.

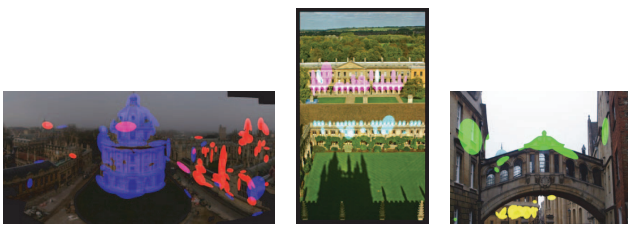

Fig. 9 Sample images containing multiple discovered objects (a) All Souls and Radcliffe Camera, (b) Magdalen Cloisters and New Building and (c) Hertford Bridge and Sheldonian Theater.

4.2.2 Quantitative Evaluation

To evaluate the performance of our method quantitatively, we score the ranked image lists described in Sect. 4.1.3 with the average precision (AP)[†]. The AP ranges from 0 to 1 and is given by the area under the precision-recall curve, where

[†]The AP is typically used for ranked lists because it takes into account the position of the relevant results.

Table 1 Highest APs for LDA, gLDA, spectral clustering (SC) and our method with and without pruning.

Ground truth Landmark	LDA [11]	gLDA [11]	SC [10]	Our Method (Without Pruning)	Our Method	Object Rank
All Souls	0.90	0.95	0.93	0.75	**0.98**	2
Ashmolean	0.49	0.59	0.62	0.84	**0.85**	26
Balliol	0.23	0.23	0.33	**0.64**	0.56	73
Bodleian	0.51	0.64	0.61	0.70	**0.83**	4
Christ Church	0.45	0.60	0.67	0.72	**0.72**	3
Cornmarket	0.41	0.41	0.65	**0.66**	**0.66**	108
Hertford	0.64	0.65	0.70	**0.90**	**0.90**	5
Keble	0.57	0.57	0.93	0.93	**0.95**	43
Magdalen	0.20	0.20	0.20	**0.51**	0.43	56
Pitt Rivers	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	39
Radcliffe Camera	0.82	0.91	0.97	**0.98**	**0.98**	1
Average	0.56	0.61	0.69	0.78	**0.80**	

precision is the ratio of retrieved positive images to the total number of retrieved images and recall is the ratio of retrieved positive images to the total number of positive images. When AP = 1, the precision-recall curve becomes ideal, namely a precision 1 for any recall value. Here, the images labeled as *Good* and *OK* are treated as positive images while images where the landmark is not present are treated as negative images. Images labeled as *Junk* are completely ignored and do not affect the AP.

To further compare our results with other existing methods, we follow the same approach of [10] and [11]. First, for each discovered object model, the AP with respect to the ground truth landmark is computed from the ranked image list. Then, for each ground truth landmark, the discovered object model with the highest AP is selected. Table 1 shows the highest APs for LDA [11], gLDA [11], spectral clustering [10]† and our method. To see the effect of pruning co-occurring word sets in Sect. 3.2.1, this table also includes the result of our method without pruning co-occurring word sets. Note that our method without regard to pruning obtained better results for all the landmarks (except Pitt Rivers for which all the methods obtained a perfect score) and in many cases with a substantial difference than the other three methods. This is clearly reflected on the average of the highest APs, where our method obtained a significantly better result. From Table 1, pruning co-occurring word sets improves the average of highest APs. This is because without pruning co-occurring word sets, meaningless object models can be derived from noisy co-occurring word sets.

Table 1 also shows the object rank of the discovered object models achieving the highest AP for our method. We confirmed visually that the object model achieving the highest AP had the highest object rank among the object models associated with the same landmark for any ground truth landmark. This fact also supports the meaningfulness of our object models.

Finally, we investigate how our method is sensitive to the parameter ϵ, which is the threshold for the overlap coefficient to merge co-occurring word sets. Figure 10 illustrates the average of highest APs for different values of ϵ. Remarkably, our method performs stably for a wide range of ϵ from 0.33 to 0.99. Thus, we can say that our method is insensitive to the choice of ϵ.

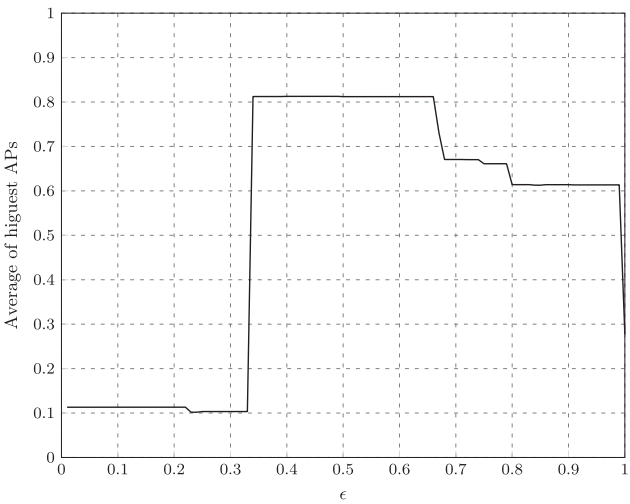

Fig. 10 Average of the highest APs over different ϵ.

4.3 Speed

All the experiments are carried out on a single 2.27 GHz Intel Xeon PC with 4 GB of memory. Table 2 summarizes the execution time for each step of our method with and without pruning. Interestingly, pruning accelerates the speed of the object discovery. This is because pruning removes noisy and uninformative co-occurring word sets, shrinking the time for the agglomerative clustering of co-occurring word sets. Without pruning, while a huge number of objects were discovered, many of them are meaningless and exploring the results may be cumbersome.

To demonstrate the scalability, we apply our method to a bigger dataset of 101,991 images which we call Rome100k. Rome100k was retrieved from Flickr using the keyword "Rome" as a query. We use the same parameter

†The method in [10] does not explicitly generate an object model. It only clusters images of the same object.

Table 3 Processing times of different methods.

Method	Dataset	# of Images	# of Features	Platform	Time
LDA, gLDA [11] (matching graph only)	Rome [25]	1,021,986	1,702,818,841	Cluster of 30 PCs	1 day
LDA, gLDA [11] (matching graph only)	Statue of Liberty [25]	37,034	44,385,173	Single PC	2 hours
SC [26]	Paris500k [27]	501,356	1,564,381,034	Cluster of PCs	61.5 days
Our Method	Rome100k	101,922	460,894,893	Single PC	26.73 minutes
Our Method	Oxford [22]	5,062	16,334,970	Single PC	6.4 minutes

Table 2 Processing time of our method with and without pruning.

	Without pruning	With pruning
# of co-occurring word sets	950,730	287,927
# of discovered objects	649,876	33,102
Time for mining co-occurring word sets (secs)	288.110	288.110
Time for pruning (secs)	0	21.881
Time for clustering (secs)	191.695	75.508
Time for ranking images (secs)	6.092	2.676
Time for ranking objects (secs)	0.020	0.004
Total time (secs)	485.917	388.179

values in Sect. 4.1.2 also for the Rome100k dataset. The time for discovering objects from the Rome100k and the Oxford datasets is presented in Table 3. Because the time for the Rome100k increased only slightly compared to that for the Oxford dataset, our method scales well with the number of images. Table 3 also summarizes the processing time of other object discovery methods reported in literatures which were executed on various datasets and platforms. Though some literatures use PC clusters, it still takes much time to discover object. Because the platforms are different, it is difficult to compare the processing time between different methods. [11] reported that it took 2 hours on a dataset of 37,034 images to construct the matching graph [11] only on a single PC, while our method took 26 minutes on 101,991 images to derive the final object models on a single PC. We interpret this result as that our method is at least comparable to [11].

As for the memory consumption, for the Rome100k dataset, mining and pruning co-occurring word sets consumed at most 806 MB of which the inverted file occupied 774 MB. On the other hand, the agglomerative clustering utilized only 46 MB.

5. Conclusions

We presented an efficient method for automatically discovering particular objects from unordered image sets. Our method pays attention to visual words that appear together in multiple images under the assumption that such co-occurring visual words are associated with the same object. We demonstrated that Min-Hashing can be used to efficiently extract co-occurring visual words from the inverted file and that extracted co-occurring word sets contain discriminative visual words. Furthermore, to deal with unstable visual words, our method obtains object models by clustering co-occurring word sets that share common visual words in an agglomerative manner. We showed that, despite our method not exploring geometric relations between visual words, the generated object models are highly discriminative and robust to occlusion, clutter and large variations of illumination and viewpoint. In a quantitative evaluation, our method achieved higher scores than the other state-of-the-art methods.

Finally, it is important that the proposed method is scalable to huge image sets and large visual vocabularies as it performs the most demanding tasks by Min-Hashing.

Acknowledgments

We would like to acknowledge anonymous reviewers and the associate editor for their helpful comments and suggestions. This research was supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan, Grant-in-Aid for Scientific Research (C), 22500122, 2011.

References

[1] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," Proc. International Conference on Computer Vision, pp.1470–1477, 2003.
[2] O. Chum, J. Philbin, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," Proc. International Conference on Computer Vision, 2007.
[3] O. Chum and J. Matas, "Large-scale discovery of spatially related images," IEEE Trans. Pattern Anal. Mach. Intell., vol.32, no.2, pp.371–377, 2010.
[4] O. Chum, J. Philbin, and A. Zisserman, "Near duplicate image detection: min-hash and tf-idf weighting," Proc. British Machine Vision Conference, 2008.
[5] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," Mach. Learn., vol.42, pp.177–196, 2001.
[6] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent dirichlet allocation," J. Machine Learning Research, vol.3, pp.993–1022, 2003.
[7] X. Wang and E. Grimson, "Spatial latent dirichlet allocation," Proc. Conference on Neural Information Processing Systems, 2007.
[8] J. Tang and P.H. Lewis, "Non-negative matrix factorisation for object class discovery and image auto-annotation," Proc. ACM International Conference on Image and Video Retrieval, pp.105–112, 2008.
[9] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman, "Discovering objects and their location in images," Proc. International Conference on Computer Vision, pp.370–377, 2005.
[10] J. Philbin, J. Sivic, and A. Zisserman, "Object mining using a matching graph on very large image collections," Proc. Indian Conference on Computer Vision, Graphics and Image Processing, pp.738–745, 2008.
[11] J. Philbin, J. Sivic, and A. Zisserman, "Geometric latent Dirichlet allocation on a matching graph for large-scale image datasets," Int. J. Comput. Vis., pp.1–16, 2010.

[12] N.A. Bhatti and A. Hanbury, "Co-occurrence bag of words for object recognition," Proc. Computer Vision Winter Workshop, pp.21–28, 2010.
[13] G. Fuentes Pineda, H. Koga, and T. Watanabe, "Object discovery by clustering correlated visual word sets," Proc. Twentieth International Conference on Pattern Recognition, pp.750–753, 2010.
[14] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J.D. Ullman, and C. Yang, "Finding interesting associations without support pruning," IEEE Trans. Knowl. Data Eng., vol.13, no.1, pp.64–78, 2001.
[15] A.Z. Broder, "On the resemblance and containment of documents," Computer, vol.33, no.11, pp.46–53, 2000.
[16] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," Proc. British Machine Vision Conference, pp.384–393, 2002.
[17] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," Int. J. Comput. Vis., vol.60, no.1, pp.63–86, 2004.
[18] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol.60, no.2, pp.91–110, 2004.
[19] H. Jégou, M. Douze, and C. Schmid, "Packing bag-of-features," Proc. International Conference on Computer Vision, pp.2357–2364, 2009.
[20] T.H. Haveliwala, A. Gionis, and P. Indyk, "Scalable techniques for clustering the web.," Proc. Third International Workshop on the Web and Databases, pp.129–134, 2000.
[21] O. Chum, J. Philbin, M. Isard, and A. Zisserman, "Scalable near identical image and shot detection," Proc. ACM International Conference on Image and Video Retrieval, pp.549–556, 2007.
[22] http://www.robots.ox.ac.uk/˜vgg/data/oxbuildings/
[23] http://www.flickr.com/
[24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1–8, 2007.
[25] N. Snavely, I. Simon, and S.M. Seitz, "Scene summarization for online image collections," Proc. International Conference on Computer Vision, pp.1–8, 2008.
[26] T. Weyand, J. Hosang, and B. Leibe, "An evaluation of two automatic landmark building discovery algorithms for city reconstruction," Proc. Reconstruction and Modeling of Large-Scale 3D Virtual Environments, 2010.
[27] http://www.mmp.rwth-aachen.de/data/paris-dataset

Gibran Fuentes Pineda received the B.E. degree in computer engineering and the M.S. degree in microelectronics engineering from the National Polytechnic Institute of Mexico. He is currently pursuing his Ph.D. degree in information systems at the University of Electro-Communications of Tokyo, Japan. His research interests include object recognition, image data mining and object/image retrieval.

Hisashi Koga received the M.S. and Ph.D. degree in information science in 1995 and 2002, respectively, from the University of Tokyo. From 1995 to 2003, he worked as a researcher at Fujitsu Laboratories Ltd. Since 2003, he has been a faculty member at the University of Electro-Communications, Tokyo (Japan). Currently, he is an associate professor at the Graduate School of Information Systems, University of Electro-Communications. His research interest includes various kinds of algorithms such as clustering algorithms, on-line algorithms, and algorithms in network communications.

Toshinori Watanabe received the B.E. degree in aeronautical engineering in 1971 and the D.E. degree in 1985, both from the University of Tokyo. In 1971, he worked at Hitachi as a researcher in the field of information systems design. His experience includes demand forecasting, inventory and production management, VLSI design automation, knowledge-based nonlinear optimizer, and a case-based evolutionary learning system nicknamed TAMPOPO. He also engaged in FGCS (Fifth Generation Computer System) project of Japan and developed a new hierarchical message-passing parallel cooperative VLSI layout problemsolver that ran on PIM (Parallel Inference Machine) in 1991. Since 1992, he has been a professor at the Graduate School of Information Systems, University of Electro-Communications, Tokyo, Japan. His areas of interest include media analysis, learning intelligence, and the semantics of information systems. He is a member of the IEEE.