Unsupervised Discovery of Co-occurrence in Sparse High Dimensional Data

Ondřej Chum and Jiří Matas
CMP, Dept. of Cybernetics, Faculty of Elec. Eng., Czech Technical University in Prague
chum@cmp.felk.cvut.cz

Abstract

An efficient min-Hash based algorithm for discovery of dependencies in sparse high-dimensional data is presented. The dependencies are represented by sets of features co-occurring with high probability and are called co-ocsets.

Sparse high dimensional descriptors, such as bag of words, have been proven very effective in the domain of image retrieval. To maintain high efficiency even for very large data collection, features are assumed independent. We show experimentally that co-ocsets are not rare, i.e. *the independence assumption is often violated, and that they may ruin retrieval performance if present in the query image. Two methods for managing co-ocsets in such cases are proposed. Both methods significantly outperform the state-of-the-art in image retrieval, one is also significantly faster.*

1. Introduction

Methods[1] describing images by bags or sets of visual words, i.e. quantized descriptors of image patches, represent the state-of-the-art in image retrieval [19, 14, 16, 10] and related tasks as image clustering or unsupervised object discovery [20, 17, 18]. A bag of words description is compact, and thus suitable for huge databases, and supports fast search via inverted files that list all documents with a given visual word.

One key issue in an image retrieval system exploiting visual word is the definition of image similarity. By far the most common is based on term frequencies (*tf*) and inverse document frequencies (*idf*) of words [19]; visual similarity is defined as a normalized sum, over all visual words, of terms that are a function of *tf* and *idf* in the query and database images.

In retrieval and recognition, the use of a similarity measure is typically justified by its probabilistic interpretation. Any similarity measure employing summing over all visual words implicitly assumes that occurrences of instances of visual words are independent. The assumption of independence in the popular *tf-idf* scheme is not made out of conviction it holds - it is intuitively obvious that it does not - but for computational convenience. Modelling any probabilistic structure for a standard vocabulary sizes of 10^4 to 10^6 is challenging; even exhaustively checking for the simplest of dependencies among pairs of visual words is prohibitive.

As the main contribution of the paper, we present a method capable of detecting significant dependencies[2] within a very large set of binary random variables. The method is especially suitable for rare events. The approach relies on a novel application of the min-Hash algorithm [4], treating the inverted file (not the image) as a document. With the proposed method, we demonstrate that dependencies of visual words are fairly common in large datasets and that ignoring the dependence hurts retrieval performance.

We call groups of non-incidentally co-occurring words *co-ocsets*. We show that modelling of dependencies within co-ocsets leads to improvement in retrieval performance. One such example is visualized in Fig.1. If highly dependent visual words on the water surface are treated as independent, the most similar images are ”full of water”. The bottom results are obtained if co-ocsets are ignored (other

[1] The authors were supported by GAČR project 102/09/P423, by EC project ICT-215078 DIPLECS and by ČVUT SGS10/069/OHK3/1T/13.

[2] The precise meaning of ”significant” is given in Section 4.

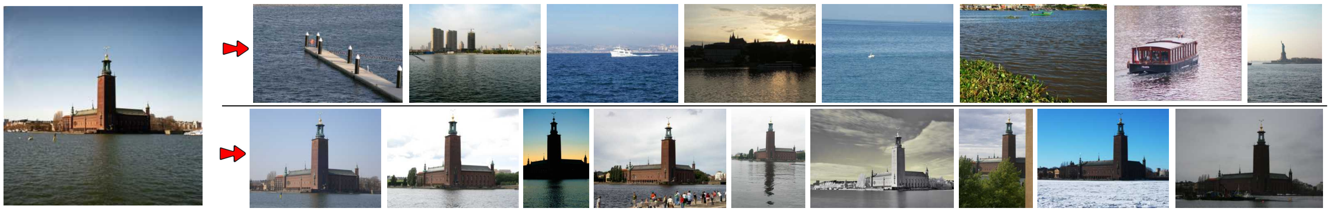


Figure 1. A query image (left) and top ranked images retrieved by two methods from a database of 5 million images: standard (*tf-idf* with spatial verification) retrieval (top) and the same method after automatic detection and removal of co-ocset features (bottom).

possibilities are discussed in section 5.2).

As a second contribution, we present two methods for managing co-ocsets that outperform the state-of-the-art in image retrieval on a standard benchmark. For difficult cases, where the standard methods fail completely, the proposed methods give good results as well as a speed-up.

Image retrieval is not the only application of the proposed method. High dimensional sparse binary features arise in other applications, e.g. document analysis and rare event detection. The proposed method is suitable, as experiments show, for large high dimensional datasets (10^6) with rare (probability < 0.001) of co-occurring events. Standard Monte-Carlo type methods for estimating co-occurrence of such events are inefficient.

1.1. Related work

The standard text or image retrieval with *tf-idf* similarity function [2, 19] weights contributions of different words. Recently, the problem that certain *individual* visual words are observed more times in a single document than the probability model predicts (so called burstiness) has been addressed in [11]. It was shown that burstiness deals well with independent, repeating words by modelling first-order statistics of word occurrences better than *tf-idf*. The co-ocsets represent second order statistics. As shown in the experiments, this becomes critical in the case of many infrequent co-occurring words, see Fig. 8. With large vocabularies, co-occurring infrequent visual words are much more common than frequent individual words.

Approaches that model the document as (mixtures of) topics, such as probabilistic Latent Semantic Analysis (pLSA) [7] and Latent Dirichlet Allocation (LDA) [3] do capture relations between visual words. However, learning such models is prohibitively computationally expensive in very high dimensional spaces and large databases. The problem of co-ocset discovery can be seen as a search for 'topics' that generate a number of features with high conditional probability on the 'topic'. Note that in this task neither all documents need to be explained, nor all features need to be 'generated'.

The closest related work in the field of computer vision is [17] by Quack *et al.*, who developed a method for mining frequent and discriminative feature configurations. The first significant difference is that the approach of [17] is semi-supervised: it is known a priori in which groups of images the co-occurring features might appear, while our approach is fully unsupervised. The second difference is in the efficiency. In [17], a data-mining algorithm APriori [1] is used. This algorithm discovers co-occuring events with frequency higher than a certain (user specified) threshold. If the elementary events that compose the co-occurrences are less frequent than the majority of the elementary events, the threshold on frequency has to be set low. This results in an extremely time–demanding process.

The rest of the paper is structured as follows. In Section 2, a brief overview of the standard min-Hash algorithm is given since min-Hash is the core part of the co-ocset discovery algorithm. The problem of over-counting in (image) retrieval is discussed in Section 3. In Section 4, an algorithm for efficient co-ocset detection is introduced. Experimental validation of the approach is given in Section 5.

2. Overview of the min-Hash Algorithm

The min-Hash algorithm is a Locality Sensitive Hashing method [9] for sets. A brief overview of the min-Hash algorithm follows; for detailed description see [4, 6].

In *standard* min-Hashing, documents (images) are represented as sets of (visual) words. Note that outside this section, the roles of documents and words are interchanged, *i.e.* in the rest of the paper, each word will be represented as a set of documents the word appears in. We will be looking for 'similar' words rather than similar documents.

A min-Hash is a function f that assigns a number to each set of visual words (each image representation). The function has the property that the probability of two sets having the same value of the min-Hash function is equal to their set overlap, *i.e.* the ratio of the intersection and union of their set representations:

$$P\{f(\mathcal{A}) = f(\mathcal{B})\} = \mathrm{ovr}(\mathcal{A}, \mathcal{B}) = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|} \in [0, 1]. \quad (1)$$

To estimate the word overlap of two images, multiple independent min-Hash functions f_i are used. The fraction of the min-Hash functions that assigns an identical value to the two sets is an unbiased estimate of the similarity of the two images. To efficiently retrieve images with high similarity, the values of min-Hash functions f_i are grouped into s-tuples called sketches. Similar images share many values of the min-Hash function and hence have high probability of producing the same sketches. On the other hand, dissimilar images have low chance of forming an identical sketch. Identical sketches are efficiently found by hashing.

The recall of min-Hash is increased by repeating the random selection of s-tuples k times. A pair of images is a potential match when at least one sketch collision is encountered. Potential matches are typically further verified. The probability of a pair of images having at least one sketch out of k in common is a function of the word overlap

$$P\{\text{collision}\} = 1 - (1 - \mathrm{ovr}(\mathcal{A}, \mathcal{B})^s)^k. \quad (2)$$

3. Word Dependence and Similarity Overestimation (Over-counting)

Let $P(A)$ stand for the probability that a visual word A is present in an image, that is $P(A) = |\mathcal{A}|/D$, where $\mathcal{A}$ is a set of images containing word A and D is the total number of images. The self-information weight (often referred to as inverse document frequency *idf*) of a word is defined as

$$\mathit{idf}(A) = -\log P(A).$$

This quantity measures the influence of word A on the similarity of two images. Now let us consider two visual words A and B co-occurring in a document. Under the assumption of independence, their contribution to the similarity function is:

$$\begin{aligned}\mathit{idf}(A,B) &= -\log P(A,B) = -\log P(A) - \log P(B)\\ &= \mathit{idf}(A) + \mathit{idf}(B).\end{aligned}$$

However, if B is dependent on A to a level that $P(A,B) = P(B|A)P(A) \approx P(A)$, the correct joint contribution of the two words is:

$$\begin{aligned}\mathit{idf}(A,B) &= \mathit{idf}(A) + \mathit{idf}(B|A)\\ &= -\log P(A) - \log P(B|A) \approx -\log P(A),\end{aligned}$$

i.e. the contribution of the two words under the independence assumption is almost twice its probabilistically justifiable value. The over-counting gets more prominent when co-ocsets becomes large and contributions from highly dependent words dominate similarity calculation, suppressing other content. Detection of co-ocsets, especially the large ones, therefore significantly influences results. Next, we present an algorithm for co-ocset detection.

4. Detection of Co-ocset via min-Hash

Given $P(A)$, $P(B)$, and $P(A,B)$, it turns out to be convenient to introduce an implicitly defined measure λ of visual word dependence:

$$P(A,B)^{\lambda} = P(A)P(B), \quad 0 \le \lambda \le 2. \tag{3}$$

The measure λ linearly relates to over-estimating the self-information weight of the visual words:

$$\lambda\left(-\log P(A,B)\right) = -\log P(A) - \log P(B).$$

Values of $\lambda \le 1$ mean that the events are uncorrelated or anti-correlated, which is a case that is not interesting here. The value of λ is always smaller or equal to 2 since

$$P(A,B)^2 = P(A)P(B|A)\cdot P(B)P(A|B) \le P(A)P(B).$$

In the range $(1,2)$, λ expresses the dependence of visual words in terms of the "over-counting" factor; $\lambda = 1$ means no over-counting, *i.e.* independence, $\lambda = 2$ a duplication, *i.e.* double counting.

A visual word is represented as a set of images containing that word. Again, note that the roles of words and documents has been swapped in comparison to standard retrieval. The proposed co-ocset detection algorithm exploits the relation between the set overlap ovr of sets representing two visual words and the measure λ. The overlap defines the probability of detection by min-Hash, while λ represents the severity of problem caused by dependence. The value of the set overlap ovr and λ is related as follows:

$$\begin{aligned}\mathrm{ovr} &= \frac{P(A,B)}{P(A)+P(B)-P(A,B)}\\ &= \frac{P(A)^{1/\lambda}P(B)^{1/\lambda}}{P(A)+P(B)-P(A)^{1/\lambda}P(B)^{1/\lambda}}.\end{aligned} \tag{4}$$

For visualization purposes (to avoid 3D plots), we assume that co-occurring features A and B are approximately equally frequent, that is $P(A) \approx P(B)$, which leads to:

$$\mathrm{ovr} \approx \frac{P(A)^{2/\lambda-1}}{2-P(A)^{2/\lambda-1}}. \tag{5}$$

Note that this assumption is not necessary for further derivation. Plots of isocontours of λ as a function of $P(A)$ and ovr, and of ovr as a function of $P(A)$ and λ are shown in Figure 2. In the plots, the range of $P(A)$ has been chosen to correspond to observed values in a real database - see the histogram of word frequencies in Figure 4 (left).

Finally, the conversion from the set overlap to λ is obtained

$$\lambda = \frac{\log\left(P(A)P(B)\right)}{\log(P(A)+P(B)) + \log(\mathrm{ovr}) - \log(\mathrm{ovr}+1)}. \tag{6}$$

Each co-ocset is defined as an ordered pair $(\mathcal{K}_i, \mathcal{F}_i)$ of sets of visual words. The first set $\mathcal{K}_i$, called core, contains highly correlated words. The other set $\mathcal{F}_i$, called fringe, contains words that often occur in images where the core words are present, *i.e.* words with high conditional probability $P(A|\mathcal{K}_i)$. The core is used to detect the presence of words from a co-ocset in an image, the fringe plays a role in similarity adjustment. Each visual word appears in at most one core, but possibly in multiple fringes.

To discover a core of a co-ocset, transitivity is assumed: if word A is highly correlated with B, and B is highly correlated with C, then A is highly correlated with C. Following the assumption, cores are constructed as transitive closures of words with the λ factor exceeding a threshold λ_0. To avoid intractable estimation of λ for every pair of visual words, the min-Hash algorithm is used to efficiently find pairs of visual words with a high value of λ. The construction naturally enforces that each visual word is in at most one co-ocset core.

Given co-ocset cores, an image is defined to contain a co-ocset iff it contains at least $\alpha|\mathcal{K}_i|$ different visual words from the core $\mathcal{K}_i$. The co-ocset fringe is then formed from

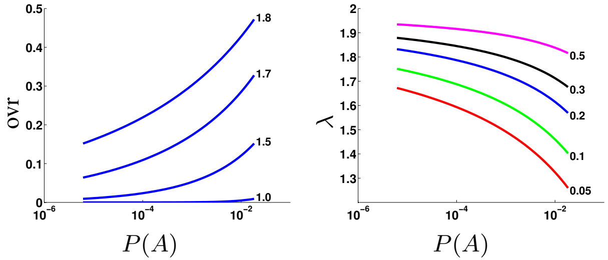


Figure 2. Isocontours of λ given $P(A)$ and the set overlap (ovr) (left) and isocontours of ovr given $P(A)$ and λ (right) according to equation (5). $P(A) \approx P(B)$ is assumed.

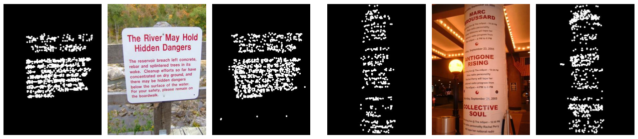


Figure 3. Sample images containing the largest (4426 words) discovered co-ocset 'dark text on light background' and positions of core (left) and fringe (right) features.

words that appear in images containing a co-ocset $\mathcal{K}_i$ significantly more frequently than in random images. The fringe features are efficiently found using 'feature' retrieval, where each list of image features serves as an inverted file. The algorithm for automatic detection of co-ocsets is summarized in algorithm 1.

```
1. For each inverted file, i.e. a list of documents containing a given visual word, compute $ks$ min-Hashes.
2. Record the $s$-tuples of min-Hashes in a hash table.
3. For each pair $A$, $B$ of visual words that encountered a sketch collision, estimate the set overlap ovr and compute $\lambda$ using eqn. (6).
4. Build a graph $G_\lambda$ where words are vertices and edges connect pairs of words with $\lambda > \lambda^*$, ($\lambda^* = 1.5$).
5. Form the co-ocset cores $\mathcal{K}_i$ as connected components in $G_\lambda$.
6. For each core $\mathcal{K}_i$ find a set of images $\mathcal{I}$ containing at least $\alpha|\mathcal{K}_i|$ words from the core $\mathcal{K}_i$, ($\alpha = 0.05$).
7. Form the co-ocset fringes as words $A$ satisfying $\frac{P(A|\mathcal{K}_i)}{P(A)} > r_0$, ($r_0 = 10$).
```

Algorithm 1: Discovery of co-ocsets via min-Hash.

5. Experiments

Like most of the recent work on image retrieval [19, 14, 16, 10], we apply the following approach. First, affine invariant features and descriptors [13] are extracted and images are represented as bags of visual words (vector quantized descriptors) [19]. In particular, we use hessian affine features and the SIFT descriptor [12].

All experiments were conducted with co-ocsets discovered in the Oxford 100k dataset[3] [16] (containing $2.3 \cdot 10^8$ features) for visual vocabulary of 10^6 words.

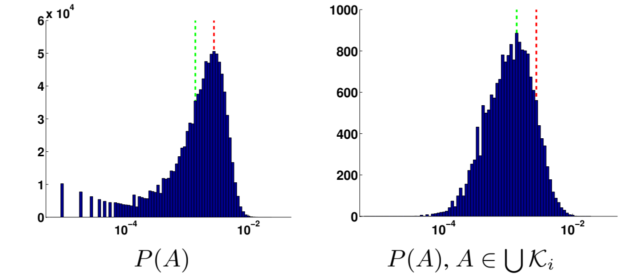


Figure 4. Histograms of visual word frequencies on the Oxford 100k dataset for all words (left) and the core words (right). Note the log scale on x-axis. Red and greed dashed lines denote the location of the modes in the first and the second plot respectively.

Parameters of the discovery process were (see Alg. 1): $k = 85$ sketches of size $s = 3$; 255 independent min-Hash functions were generated. First, selected statistics of the dataset and the co-ocsets are presented. Then, the influence of co-ocsets to image (or rather particular object) retrieval are shown and discussed.

5.1. Statistics

First, consider two histograms of probabilities $P(A)$ (*i.e.* word frequencies) in Figure 4. Note (i) the logarithmic scale on the x-axis and (ii) that the vertical axes of the two plots have different scales. The left plot shows the frequencies of all 1M visual words, while the right plot shows the frequencies of co-ocset core words. Suprisingly, the median of all word frequencies (0.0019) is larger than the median of the core word frequencies (0.0015). This means that the co-occurring sets of visual words are mostly composed of words occurring less frequently than an 'average' word. Highly frequent words are rarely part of a co-ocset core. As a consequence, greedy algorithm trying out combinations of frequent words are unlikely to produce good results in sub-quadratic time.

In the Oxford 100k database, 103 co-ocset cores were discovered. The number of words in the cores ranges from 6 (acceptance threshold) to 4426 words. More than half of the cores are smaller than 15, only 14 large cores with more than 50 words were found. In total, 18071 visual words, *i.e.* close to 2%, are in a co-ocset core. The number of words in a co-ocset fringe ranges from 500-13000, the average being 6063 words. In total, fringes contain 246,782 different words, *i.e.* almost 25% of all words.

Two images containing features from the largest co-ocset and the spatial distributions of core and fringe words are shown in Figure 3. More examples are presented in the colored boxes in Figure 5 and in Figure 6. From the images it can be seen that the co-ocset words are well spatially localized.

5.2. Application: Image retrieval

Managing co-ocsets. We implemented two methods for

[3] Note that Oxford-100k does not include Oxford-5k landmark dataset.

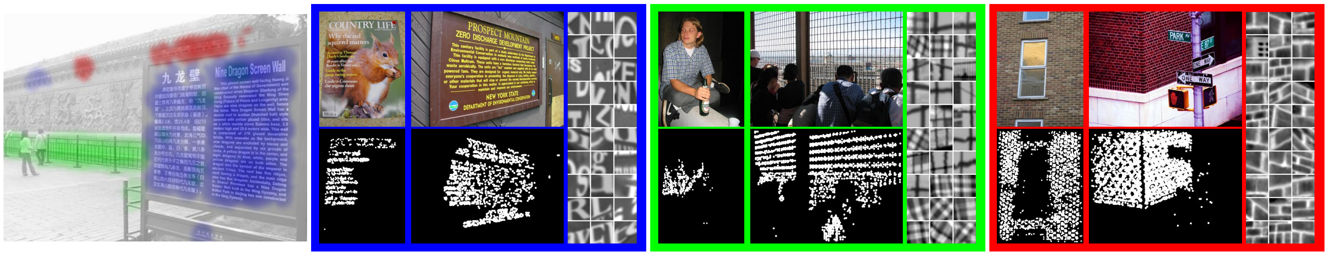


Figure 5. Locations of features belonging to three detected co-ocsets are colour-coded in a query image (left). For each co-ocset ('bricks' in red, 'railings' in green, and 'light text on dark background' in blue), a few selected images (from the Oxford 100k dataset) containing a large number of the co-ocset words, their spatial location, and sample of co-ocset feature patches are shown.

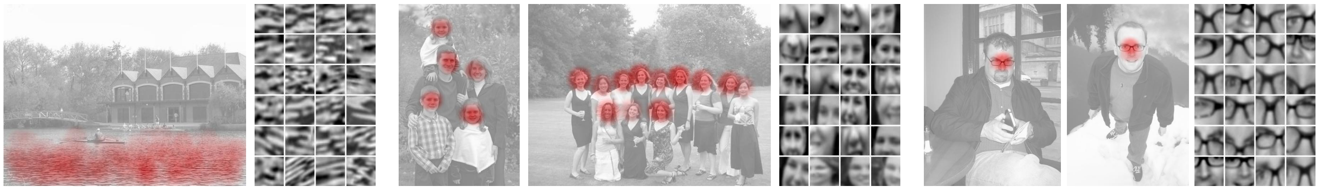


Figure 6. Examples of different co-ocsets with a sample of patches associated with core features. The colour shows the spatial distribution of the co-ocset features.

incorporating information about dependent words into an image retrieval system. Both methods check for the presence of co-ocset cores in the query image. If there is no co-ocset detected in the image, the process is exactly the same as in the case of the baseline algorithm. Otherwise, the first mehod, denoted 'Rmv', removes words belonging to co-ocsets (both core and fringe) from the query. The second method, denoted 'Full', applies an analogy to the burstiness feature reweighting ([11], eq. (4)) in each database-query similarity calculation:

$$s'_f = s_f \sqrt{\frac{s_f}{\sum_{p \in \mathcal{K} \cup \mathcal{F}} s_p}},$$

where $\mathcal{K}$ and $\mathcal{F}$ are the core and fringe of the co-ocset features f belongs to, s_f is the standard *tf-idf* based contribution of f to the matching score. Experiments show that the 'Full' method is slightly more precise (see Tables 1 and 3). The 'Rmv' method outperforms the 'Full' method in certain situations, *e.g.* when the co-ocsets happen to be on occluding or irrelevant structures. In such cases, like for the query 4 in Table 1, the best course of action is co-ocset removal. On the other hand, in rare cases, when the image is composed of co-ocsets features only, the 'Rmv' method fails, as in Figure 7. From a practical point of view, the slightly higher robustness and precision of the 'Full' method is probably more than compensated for by the speed of the 'Rmv' method.

The baseline method follows the architecture described in [15]. First, images are ranked using the *tf-idf* scoring. This procedure is fast, all documents in the database are considered. In the second step, geometric constraints are used to re-rank top ranked images. The spatial re-ranking using RANSAC approach [16] is relatively slow and only top 1000 documents are re-ranked. The spatial verification guarantees a low false-positive rate, if a correct image is ranked high enough in the *tf-idf* scoring, it is usually correctly retrieved. However, if the initial ranking fails to propose correct images, the spatial re-ranking has no chance of improving it.

Efficiency of co-ocset discovery. For the Oxford 100k database, the process requires less than 1 hour for a Matlab implementation run on a single machine. Such a time demand is orders of magnitude lower than the time spent on feature detection, SIFT descriptor extraction, and vector quantization into visual words. The time to pre-process the query is negligible (not measurable) with respect to the query time (for query times see Table 2).

5.2.1 Experiment Q8

With co-ocsets detected on the Oxford 100k database, image retrieval was performed on a database of 5 million Flickr [8] images. Eight queries, each including features from a different co-ocset, were used. The query images are shown above Table 1, selected retrieval resultsfor the baseline and 'Rmv' methods are presented in Figures 9 and 1.

Accuracy of image retrieval for the 8 queries is summarized in Table 1. The average precision of the baseline method is very low, while both 'Full' and 'Rmv' methods have very high average precision, typically about an order of magnitude higher. The result might lead to unjustified optimism. The poor performance of the baseline method is not a surprise as only queries containing co-ocsets, problematic for the *tf-idf* similarity, were selected. The result demonstrates that there are images where the *tf-idf* similarity scheme effectively fails, not how common such images are in real-world retrieval problems. In general, the base-

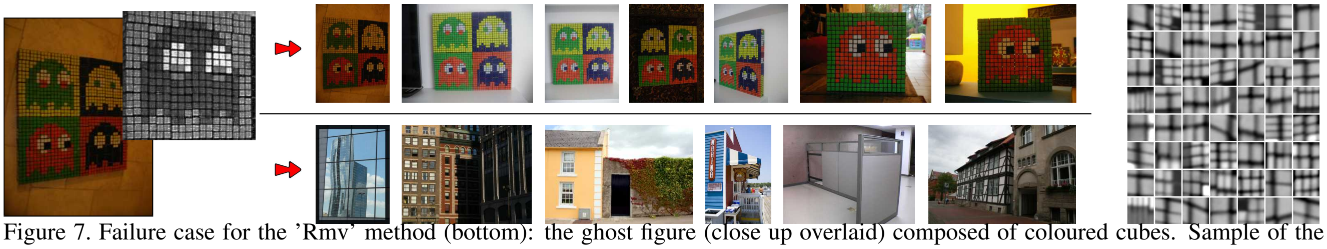
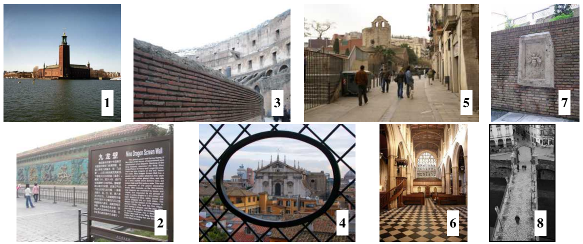


Figure 7. Failure case for the 'Rmv' method (bottom): the ghost figure (close up overlaid) composed of coloured cubes. Sample of the query image patches (right). The 'Full' method is successful (top).



Experiment Q8: the eight queries with different co-ocsets.

	Base	Full	Rmv	Base	Full	Rmv
				QE	QE	QE
1	0.038	**0.340**	0.304	0.038	**0.676**	**0.676**
2	0.073	**0.639**	0.615	0.071	**0.816**	**0.816**
3	0.144	**0.430**	**0.430**	0.143	**1**	**1**
4	0.041	0.073	**0.084**	0.148	0.778	**0.954**
5	0.067	0.468	**0.534**	0.067	**1**	**1**
6	0.031	0.625	**0.787**	0.031	**0.926**	**0.926**
7	0.150	0.403	**0.550**	0.150	**1**	**1**
8	0.200	**0.400**	**0.400**	0.200	**1**	**1**

Table 1. Experiment Q8: Average precision for the baseline[15], 'Full' and 'Rmv' methods (without and with Query Expansion [5]) in a database of 5M Flicker images. Note: the scores are upper bounds as full ground truth is unknown.

	Baseline	Rmv	Baseline+QE	Rmv+QE
1	106	32	519	82
2	465	66	971	200
3	95	76	190	137
4	97	83	199	157
5	127	58	522	101
6	117	26	328	119
7	234	19	686	74
8	309	62	775	105

Table 2. Experiment Q8: Speed (in milliseconds). On average, 'Rmv' is over 4 times faster than the baseline on Q8.

line method weighs high uninformative (for most specific queries) structures that contain a large number of frequently co-occurring words.

The high average precision of the 'Full' and 'Rmv' methods might be an artefact of the way ground truth was obtained. All tested methods were run and all correctly retrieved images among a few hundred top ones were marked. The precision is thus an upper bound; we have no way of checking five million images. The ordering of the average precisions is thus correct, and the comparison of methods is fair, but the absolute value is an upper bound.

Retrieval speed. The speed of the baseline and 'Rmv' methods is compared in Table 2. Since the 'Rmv' method reduces the number of query words, it traverses through less inverted files as well as reduces the number of tentative correspondences in the spatial verification. As a result, the method might be up to over nine times faster. The speed-up depends on the fraction of words in the query image belonging to co-ocsets. The speed of the 'Full' method is comparable to the baseline method.

Comparison with burstiness. It has been observed that high visual word counts in a single image occur much more often than predicted by the *tf-idf* statistical model, Jegou *et al.* [11] proposed a method that deals with such "bursts" of features, caused by repetitive pattern in the image.

For the queries included in the Q8 experiment, the burstiness similarity function of [11] does not significantly improve the retrieval output, see Figure 8 and compare the results with Figure 1 and 9. This is not surprising, *e.g.* the water features repeat only 1.05 times on average. Burstiness works well for a few highly repeating features, co-ocsets allow to deal with many features which are individually not frequent. The two methods can be combined - we adopted the burstiness similarity score in the 'Full' method.

5.2.2 Retrieval benchmark

The two methods for co-ocset management were evaluated on the standard Oxford building retrieval benchmark. The results of all the method are similar, both 'Full' and 'Rmv' methods slightly outperformed the standard (state-of-the-art) method, see Table 3. In the experiment with the Paris vocabulary, co-ocsets discovered in this vocabulary were used. The protocol of the benchmark defines query boxes that do not contain significant distracting co-ocsets. For such queries the standard approach works well and only marginally better results are achieved with the proposed method.

5.3. Application: Image clustering

The benefit in the image retrieval is not achieved by improving the score of correctly matching images, but by suppressing false matches caused by over-counting. This turns

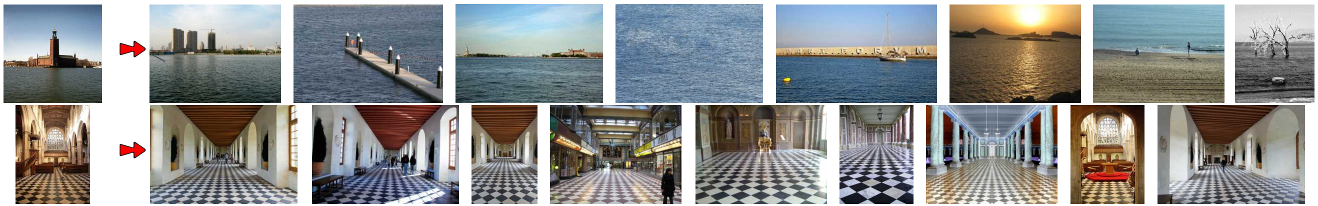


Figure 8. Retrieval results using the burstiness similarity function [11].

	Oxford 5k vocab		Paris	
		QE		QE
Baseline	0.727	0.862	0.574	0.728
Full	**0.731**	**0.864**	**0.579**	**0.732**
Rmv	**0.731**	**0.864**	0.574	0.731

Table 3. Results on the 105k Oxford database, with (QE) and without (blank) query expansion for a vocabulary trained on the Oxford and Paris datasets respectively, following the protocol in [16].

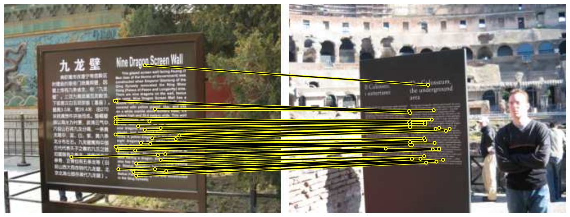


Figure 10. Geometrically consistent set of co-occuring features.

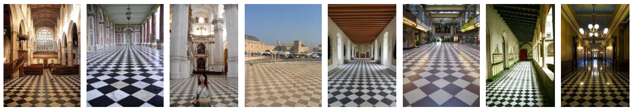


Figure 11. Images from the same cluster: (a) Oxford, UK, (b) Versailles, France, (c) Granada, Spain, (d) Marseilles, France, (e) Chenonceaux, France, (f) Prague, Czech Republic, (g) Barcelona, Spain, (h) Lansing, Michigan.

out to be very important for clustering of spatially related images as it avoids linking *e.g*. different historical spots through the information boards. This phenomenon can be observed in the 'Dragon Wall' query image (Figure 9 top). Three different co-ocsets were detected as shown in Figure 5, the 'text' being by far the strongest. For the 'Dragon Wall', the baseline method retrieves four images of the same information board, followed by a number of (different) information boards of the same type progressively changing into a generic text on a dark background. Figure 10 shows an example of spatially consistent matches on unrelated object. Another example is a cluster of eight different locations sharing the tiled floor, see Figure 11.

6. Conclusions

We have proposed an efficient algorithm, based on min-Hash, for discovery of dependencies in sparse high-dimensional data. The dependencies are represented by co-ocsets, *i.e*. sets of features co-occurring with high probability. We have demonstrated the influence of co-ocsets and the invalid assumption of visual word independence on image retrieval results. We have shown that there exist a large variety of images containing co-occurring words. These structures dominate the computed similarity, completely ruining the results of standard retrieval. Two methods for managing co-ocsets in such case have been proposed. Both methods significantly outperform the state-of-the-art, the 'Rmv' method is also significantly faster.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, 1993.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, ISBN: 020139829, 1999.

[3] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. *J. Machine Learning Research*, 3:993–1022, Jan 2003.

[4] A. Broder. On the resemblance and containment of documents. In *SEQS: Sequences '91*, 1998.

[5] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. ICCV*, 2007.

[6] O. Chum, J. Philbin, and A. Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. In *Proc. BMVC.*, 2008.

[7] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, 1999.

[8] http://www.flickr.com/.

[9] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. of Symposium on Theory of Computing*, 1998.

[10] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. ECCV*, 2008.

[11] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *Proc. CVPR*, 2009.

[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[13] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1/2):43–72, 2005.

[14] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, 2006.

[15] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, 2009.

[16] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007.

[17] T. Quack, V. Ferrari, and L. Van Gool. Video mining with frequent itemset configurations. In *Proc. CIVR*, 2006.

[18] T. Quack, B. Leibe, and L. Van Gool. World-scale mining of objects and events from community photo collections. In *CIVR*, 2008.

[19] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.

[20] J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *Proc. CVPR*, Jun 2004.

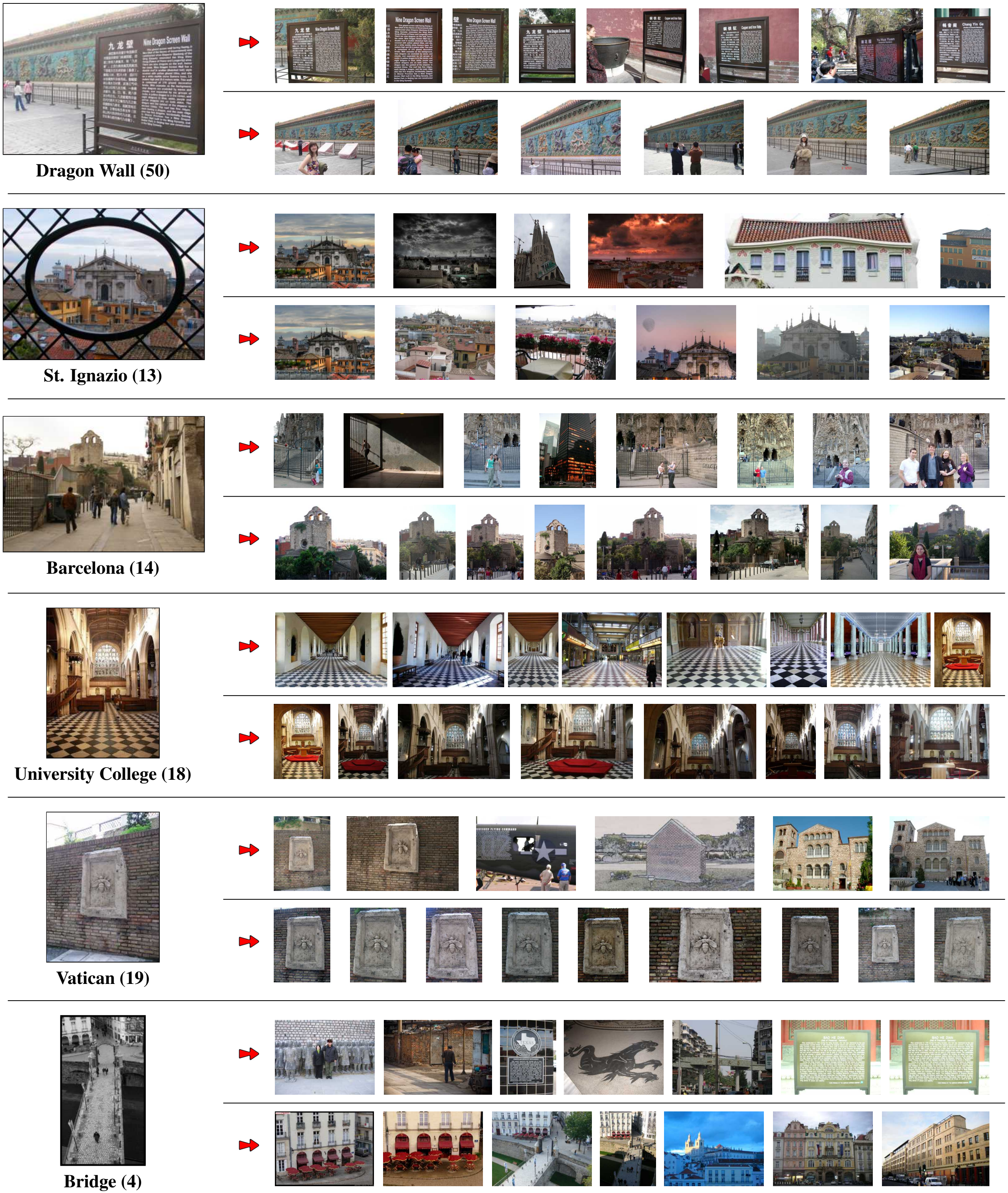


Figure 9. Examples of queries (leftmost images) where standard image retrieval fails to return images of the query object (upper rows of results). The results of the 'Rmv+QE' method are shown in the lower rows. The number of true-positive results prior to first false-positive of the proposed method is shown next to the query image name.