Large-Scale Discovery of Spatially Related Images

Ondřej Chum and Jiří Matas

Abstract—We propose a randomized data mining method that finds clusters of spatially overlapping images. The core of the method relies on the min-Hash algorithm for fast detection of pairs of images with spatial overlap, the so-called cluster seeds. The seeds are then used as visual queries to obtain clusters which are formed as transitive closures of sets of partially overlapping images that include the seed. We show that the probability of finding a seed for an image cluster rapidly increases with the size of the cluster. The properties and performance of the algorithm are demonstrated on data sets with 10^4, 10^5, and 5×10^6 images. The speed of the method depends on the size of the database and the number of clusters. The first stage of seed generation is close to linear for databases sizes up to approximately $2^{34} \approx 10^{10}$ images. On a single 2.4 GHz PC, the clustering process took only 24 minutes for a standard database of more than 100,000 images, i.e., only 0.014 seconds per image.

Index Terms—minHash, image clustering, image retrieval, bag of words.

1 Introduction

Collections of images of ever growing sizes are becoming common both due to commercial efforts [1] and as a result of photo and video sharing of individual people [2], [3]. Structuring and browsing large image databases is a challenging problem. Developments like Photo Tourism [4] show that access to images based on the 3D acquisition location or on the spatial overlap of the scenes they depict is intuitive and has high user acceptability. Commonly, the sets of relevant spatially related images are obtained using manual annotations. We propose a method for discovering spatial overlaps using image content only via image retrieval techniques.

Recent image and object retrieval systems[1] support visual search even in large databases [5], [6], [7], [8], [9]. Starting from a visual example including an instance of the object of interest, such systems are able to retrieve relevant images with both high precision and recall. A direct application of this methodology to the data mining task is to take each image in the database and query the database with it. The method is quadratic in the size of the database,[2] and hence, not feasible for large databases.

In this paper, a randomized data mining method for finding clusters of images with spatial overlap is proposed. Instead of trying to match each image, in turn, the method relies on the min-Hash algorithm for fast detection of random pairs of images with spatial overlap, the so-called cluster seeds. The seeds are then used as visual queries and clusters are obtained as transitive closures of sets of partially overlapping images that include the seed.

We show that the probability of finding a seed for an image cluster rapidly increases with the size of the cluster and approaches one fast. For practical database sizes, the running time of the seed generation process is close to linear in the size of the database. The cluster completion process requires a number of visual queries proportional to the number of images (or the number of different viewpoints) in all clusters.

The proposed unsupervised clustering method for large (Web scale) image databases has the following desirable properties:

1. It is scalable—expensive operations, such as querying the whole database, are not applied to every single image, but only to a subset with cardinality proportional to the number of images in the clusters.
2. It is incremental—adding new images into the database is possible without recomputing the whole clustering.
3. The probability of discovering a cluster is independent of the database size.
4. It is easy to parallelize.

The rest of the paper is structured as follows: Section 2 reviews the work on unsupervised object and scene discovery. Section 3 describes the use of min-Hash for data mining purposes. In Section 4, the method is experimentally verified on real image databases.

2 Related Work on Unsupervised Object and Scene Discovery

The problem of matching (organization) of an unordered image set was introduced by Schaffalitzky and Zisserman [10]. The objective was to automatically recover geometric relations between images from a spatially related set (of tens of images) and then to perform 3D reconstruction. We are interested in a similar problem, but also in the discovery of multiple such sets in databases with the number of images several orders of magnitude higher.

Recently, the majority of image retrieval systems have adopted the bag-of-words approach [11], which we follow. First, regions of interest are detected [12] and described by an invariant descriptor [13]. The descriptors are then vector quantized into a vocabulary of visual words [11], [5]; here approximate k-means [6] is used.

The approach closest to ours is [14] by Sivic and Zisserman, who aimed at unsupervised discovery of multiple instances of particular objects in feature films. In [14], object hypotheses are instantiated on neighborhoods centered around regions of interest. The neighborhoods include a predefined number of other regions, and the hypothesized object is represented by a fixed number of visual words describing the regions. Each hypothesized object is used to query the database consisting of key frames of the film. To reduce the number of similarity evaluations, which each require counting the number of common visual words, only neighborhoods centered at the same visual word are compared.

The method executes $\sum_{i=1}^{w} d_i^2$ similarity evaluations, where w is the size of vocabulary and d_i is the number of regions assigned to ith visual word. Let D be the number of documents and t the average number of features in an image so that $\sum_{i=1}^{w} d_i = tD$. The lower bound on the complexity of the approach in [14] can be written as

$$\sum_{i=1}^{w} d_i^2 \geq \sum_{i=1}^{w} \left(\frac{tD}{w}\right)^2 = \frac{t^2}{w} D^2. \qquad (1)$$

The asymptotic complexity of [14] is thus $\mathcal{O}(D^2)$. The factor t^2/w is a ratio of two constants independent of the size of the database. The size of the vocabulary commonly used is up to $w = 10^6$, and

[1] By “object retrieval,” we mean retrieval of a particular object (e.g., “my car”), not a category/class of objects (e.g., “a car”). We use the terms “categorization” or “object class recognition” for the latter problem.

[2] For retrieval systems based on inverted files, each query has to touch all images that have at least one visual word in common with the query image. The number of such images is proportional to the size of the database.

- *The authors are with the Faculty of Electrical Engineering, Czech Technical University, Karlovo námestí 13, 121 35 Prague, Czech Republic.*
E-mail: {chum, matas}@cmp.felk.cvut.cz.

Manuscript received 12 Oct. 2008; revised 3 Feb. 2009; accepted 10 Aug. 2009; published online 2 Sept. 2009.
Recommended for acceptance by A. Torralba.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-10-0695.
Digital Object Identifier no. 10.1109/TPAMI.2009.166.

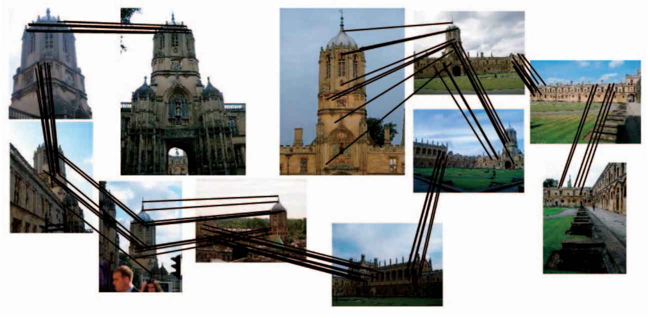

Fig. 1. Visualization of a part of a cluster of spatially related images automatically discovered from a database of 100K images. Overall, there are 113 images in the cluster, all correctly assigned. A sample of geometrically verified correspondences is depicted as links between images. Note that the images show the tower from opposite sides.

the average number of regions in an image for the database used in this paper is slightly over $t = 2{,}800$, leaving the value of the coefficient $t^2/w = 7.84$ in order of units. Hence, the algorithm would behave as quadratic in the number of images even for relatively small databases. The complexity of [14] is thus the same as the complexity of querying the whole database with each image in turn. Another approach that evaluates a complete graph on all images is due to Philbin and Zisserman [15], who report clustering of 37K image database in around 2 hours on a single machine.

Methods for query speedup [16], [7] proceed by preclustering documents into similar groups. For a query, first a set of relevant document clusters is retrieved sublinearly and the query is only evaluated against images in the selected clusters. Such an approach trades off recall for up to a sevenfold speedup [7], but remains quadratic.

Approaches improving the accuracy of image retrieval [7], [9] are relevant to this paper since they improve the second stage of our approach, the crawl over images visually connected to seed pairs. Accuracy improving techniques include learning a local interdocument distance measure based on the density in the document space [7] and selecting the most informative features for the vocabulary [9]. Note that the statistics used in these approaches might be difficult to update when new, either related (changing the density in the document space) or completely unrelated (changing the relevance of the features) images are inserted into the database.

Data mining methods have been applied to video mining of reoccurring objects and scenes in videos (of approximately 1,500 key frames) in [17]. A fast motion segmentation is used as an attention filter.

Large-scale clustering has been recently demonstrated by Quack et al. [18], who use the GPS information to reduce the large-scale task down into a set of smaller tasks.

Li et al. [19] take a large collection of images that are mostly from a single cluster. The collection undergoes an initial clustering of similar views into iconic images using a global image descriptor GIST [20], which avoids image matching within a similar view clique. This task is different from ours, where each cluster typically covers only a small fraction of the database and the aim is to avoid attempts to match unrelated images. However, we find that grouping images into similar views is indeed advantageous for large clusters. We discuss the similar view grouping together with the seed growing step in Section 3.3.

Another class of methods tackling unsupervised object learning is based on topic discovery [21] via generative modeling like probabilistic Latent Semantic Analysis (pLSA) [22] and Latent Dirichlet Allocation (LDA) [23]. Object discovery based on topic analysis method was further developed in [24], where multiple segmentations were used to hypothesize the locations and extent of possible objects. The combination of quadratic preclustering and geometry-aided LDA model has appeared in [25]. The pLSA and LDA models are a favorite choice for (unsupervised) object/image *category* recognition due to their generalization power. However, the ability to generalize to a topic such as "building" is rather a disadvantage when particular objects are sought.

We consider topic analysis approaches not suitable for our problem for the following reasons: 1) Speed: These learning methods are slow, iterative, and sequential (difficult or impossible to parallelize). 2) Topics discovered by pLSA/LDA typically appear in a number of images proportional to the size of the data set, while, in this paper, we aim at finding clusters of certain size independent of the size of the database. 3) When new images are inserted into the database and a new topic should be formed using both old and new data, the methods need to process the original (already processed) data again together with the new ones.

3 Data Mining with min-Hash

In this section, the proposed method for discovery of clusters of spatially overlapping images is described.

We formulate the task of discovery of spatially related images as finding connected components in a graph. Vertices of the graph represent images. Two images are related if they contain the same scene. From the point of view of the fast clustering algorithm, we adopt a pragmatic definition: A pair of images depicts the same scene if they can be matched by some robust matching method.

While the vertices of the graph are known (the image database) the edge structure is not known a priori and has to be discovered by the clustering algorithm. An image retrieval system can be thought of as an efficient method that, given one vertex (an image), returns all edges to related images. In most of current retrieval systems, a query has complexity linear in the number of images in the database, but is many orders of magnitude faster than actually attempting to match every single image to the query image.

The min-Hash is a hashing method for *fast* retrieval[3] of edges. However, the price paid for the efficiency of the method is low recall: Each edge is only discovered with probability P_C. The probability is proportional to the image pair similarity based on the fraction of common visual words shared by the images. Both the similarity and the probability are defined and discussed in detail in Section 3.1. The probability P_C is high (close to one) only for near duplicate images, which is the domain where the min-Hash has been used so far [26], [27].

The approach. We are tackling the problem in a two-step procedure. A subset of edges is detected using the min-Hash algorithm. The complexity of this approach is linear in the number of images in the database. These detected edges are called seeds. Seeds are then completed into connected components by repeated use of image retrieval.

Understanding the procedure requires at least a basic familiarity with min-Hash, and we, therefore, review the algorithm in Section 3.1. Next, the four steps of the cluster discovery algorithm, summarized in Fig. 2, are detailed.

3.1 The min-Hash Algorithm Overview

The min-Hash algorithm is a Locality Sensitive Hashing [28] for sets. It finds highly similar image pairs with probability close to one, unrelated images with probability close to zero, and similar image pairs (with low but nonnegligible similarity, such as images of the same object) with a rather small probability (see Fig. 3) [27], [8]. The low recall prevents min-Hash from being used directly as a general image retrieval method. However, in this paper, we argue that it can be efficiently used for data mining purposes.

[3] Any fast method with sufficient recall can be used in this stage. To our knowledge, min-Hash-based methods are the most suitable for their efficiency and robustness of the representation.

1) **Hashing.** Image descriptors are stored in a hash table. In our experiments, 2^{51} different descriptor values are used. The probability of two images falling into the same bin (*exact* descriptor match) is proportional to their similarity – eqn. (2).
2) **Similarity estimation.** For all $\binom{n}{2}$ pairs of the n images hashed in the same bin, i.e. for n collisions in the bin, similarity is calculated. The process is fast and consists of comparing two vectors and counting the number of identical elements. In our experiments, the number of vector elements is 512. The similarity is then thresholded.
3) **Spatial consistency.** For each image pair that passed the similarity test, spatial consistency is verified. Image pairs that pass the spatial consistency test become *cluster seeds*.
4) **Seed growing.** Seed images are used as visual queries and the query expansion technique is used to 'crawl' the images in the cluster.

Fig. 2. The min-Hash Image Clustering (MHIC) algorithm.

A brief overview of the min-Hash algorithm follows; for a detailed description, see [26], [27]. Images are represented as sets of visual words. This is a weaker representation than a bag of visual words since word frequency information is reduced into a binary information (present or absent).

A min-Hash is a function f that assigns a number to each set of visual words (each image representation). The function has a 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. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be sets of visual words. To simplify the notation and terminology, in connection with min-Hash, we use the term "similarity" for the set overlap:

$$\mathrm{sim}(\mathcal{A}_1, \mathcal{A}_2) = \frac{|\mathcal{A}_1 \cap \mathcal{A}_2|}{|\mathcal{A}_1 \cup \mathcal{A}_2|} \in [0, 1]. \quad (2)$$

The probability of two images having the same min-Hash is then

$$P\{f(\mathcal{A}_1) = f(\mathcal{A}_2)\} = \mathrm{sim}(\mathcal{A}_1, \mathcal{A}_2).$$

In practice, the min-Hash function f is implemented using a hash function π that generates a random number for each visual word in the vocabulary. The function $f(\mathcal{A}_1)$ is then defined as a minimal hash of elements of the set $\mathcal{A}_1$:

$$f(\mathcal{A}_1) = \min_{x \in \mathcal{A}_1} \pi(x).$$

To estimate the similarity of two images, multiple independent min-Hash functions f_i (i.e., independent π_i hash functions) are used. The fraction of the min-Hash functions that assigns an identical value to the two sets gives an unbiased estimate of the similarity of the two images.

Retrieval with min-Hash. So far, a method to estimate a similarity of two images was discussed. To efficiently retrieve images with high similarity, the values of min-Hash function f_i are grouped into s-tuples called sketches. Similar images have many values of the min-Hash function in common (by the definition of similarity), and hence, have high probability of having the same sketches. On the other hand, dissimilar images have low chance of forming an identical sketch. Identical sketches are efficiently found by hashing.

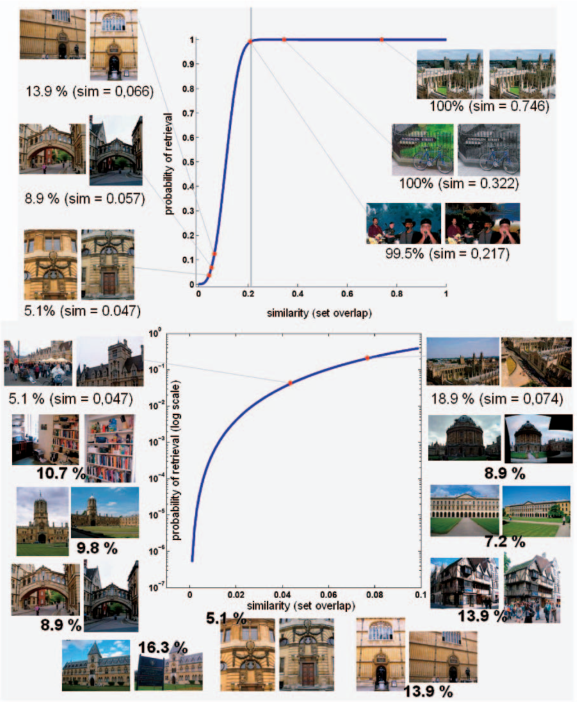

Fig. 3. The probability of at least one sketch collision for two documents plotted against their similarity; with $k = 512$ sketches, $s = 3$ min-Hashes per sketch. Image pairs of different similarities are added to relate to the "visual similarity." The bottom plot shows a close-up of the bottom left corner of the left plot. Note the logarithmic vertical axis.

The probability of two sets having at least one sketch out of k in common is

$$P_C(\mathcal{A}_1, \mathcal{A}_2) = 1 - (1 - \mathrm{sim}(\mathcal{A}_1, \mathcal{A}_2)^s)^k. \quad (3)$$

The probability depends on the similarity of the two images and the two parameters of the method: s the size of the sketch and k the number of (independent) sketches. Fig. 3 visualizes the probability of collision plotted against the similarity of two images for fixed $s = 3$ and $k = 512$. The figure also shows different image pairs and their similarity.

Word weighting. The set similarity measure (2) assumes that all words are equally important. In practice, some visual words are more discriminative than others. An extension proposed in [30] works with a similarity measure allowing different weights for different visual words. Let $d_w \geq 0$ be an importance of a visual word X_w. The weighted set overlap similarity of two sets $\mathcal{A}_1$ and $\mathcal{A}_2$ is

$$\mathrm{sim}_w(\mathcal{A}_1, \mathcal{A}_2) = \frac{\sum_{X_w \in \mathcal{A}_1 \cap \mathcal{A}_2} d_w}{\sum_{X_w \in \mathcal{A}_1 \cup \mathcal{A}_2} d_w}. \quad (4)$$

It was shown that the novel measure has two advantages compared with the original set overlap: It better captures the image similarity and reduces the number of false sketch collisions. For these reasons, we follow [30], using inverse document frequency (*idf*) as word weights.

3.2 Cluster Seed Generation

In this section, a randomized procedure that generates seeds of possible clusters of images is described. Let us first look at the plot of the probability of sketch collision as a function of image similarity depicted in Fig. 3. The sigmoid-like shape of the curve is important for the near duplicate detection task [27]. Image pairs with high similarity are retrieved with a probability close to one. The probability drops rapidly—through similar image pairs

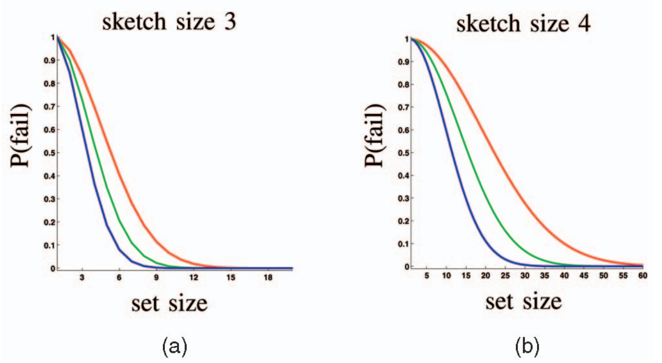

Fig. 4. (a) and (b) Probability of failure to generate a seed in a set of images depicting the same object using min-Hash with 512 sketches of sizes 3 and 4; note the different scales on the horizontal axes. The three curves show the dependence for different "average" similarity equal to 7 percent (lowest curve), 6 percent (middle), and 5 percent (highest).

(typically, images of the same object from a slightly different viewpoint) that are occasionally retrieved to unrelated image pairs (with similarity below 1 percent) that have close to zero probability of being retrieved.

Now, for the purpose of data mining, let us focus on the bottom left corner of the graph. According to (3), an image pair with similarity $\mathrm{sim} = 0.05$ has probability 6.2 percent to be retrieved (using 512 sketches of size 3). Such a poor recall is certainly below acceptable level for a retrieval system. However, we do not aim at retrieving all relevant images from the image clusters in a single step. The task is to quickly detect seeds from the clusters—it is sufficient to retrieve a single seed per cluster, and we are fortunate that the importance of a cluster is related to its size in the database.

The probability that not a single image pair (seed) is found by the min-Hash depends on two factors—the similarity of the images in the cluster and the number of image pairs that actually observe the same object. In the following analysis, which demonstrates an approximate lower bound on this probability, we assume that a particular object or landmark is seen in v views. The probability that none of the pairs $(\mathcal{A}_i, \mathcal{A}_j)$ of v views is retrieved is approximated by

$$P\{\mathrm{fail}\} = \prod_{i \neq j} 1 - P_C(\mathcal{A}_i, \mathcal{A}_j) = (1 - \varepsilon)^{\frac{v(v-1)}{2}}. \tag{5}$$

Here, ε stands for an "average" collision probability. The "average" cluster similarity is then defined by (3). The plot in Fig. 4 shows that, for popular places (i.e., those where photos are often taken from) the probability of failure to retrieve an image pair vanishes. There are three plots for similarities 5, 6, and 7 percent, respectively. Since the similarity is defined as a ratio of the size of the intersection over the size of the union, the difference between similarity 6 and 5 percent is substantial. Going from 6 to 5 percent similarity means removing 17.5 percent of elements that were in the intersection.

It is important to point out that the probability of finding a seed depends on the image similarities and the number of views and is completely *independent* of the size of the database. The v views have the same chance to be discovered in a database of 5,000 images as in a database of several millions of images without any need to change the method parameters or rehash. This is not true for many topic discovery approaches.

Remark. Equation (5) for collision probability resembles the formula for the probability of success of the popular robust estimator RANSAC [29] and there are clear analogies between the two procedures. The probability of discovering a particular edge in the cluster is relatively small. In RANSAC, this corresponds to a small probability of drawing an uncontaminated (correct) data sample. In RANSAC, there are many distinct uncontaminated data samples and any of those enables model parameters to be estimated correctly. Similarly, there are many edges in an cluster of spatially related images. Any single edge from the cluster allows for discovery of the whole cluster (using the image retrieval to complete the cluster).

Time complexity. The method is based on hashing with a fixed number M of bins. The number of bins is based on the size of the vocabulary which cannot be infinitely increased without splitting descriptors of the same physical region. Assuming uniform distribution of the keys, the number C of keys that fall into the same bin is a random variable with a Poisson distribution where the expected number of occurrences is $\lambda = D/M$ (the number of documents divided by the number of bins in the hashing table). The expected number of key pairs that fall into the same bin (summed over all bins) is

$$\sum_{i=1}^{M} \mathbf{E}(C^2) = \sum_{i=1}^{M} \left(\lambda^2 + \lambda\right) = \frac{D^2}{M} + D. \tag{6}$$

The time complexity is $\mathcal{O}(D^2)$ for D, the size of the image database, approaching infinity. However, for finite databases of sizes up to $D \leq M$, the method behaves as linear in the number of documents since $D^2/M + D \leq 2D$. In the min-Hash algorithm, the number of keys depends on the size of the vocabulary w and the size of the sketch s and is proportional to $M = w^s$. In the experiments in this paper, we used $w = 2^{17}$ and $s = 3$ or $s = 4$. This gives the number of different hash keys $M = 2^{51}$ and $M = 2^{68}$. We believe that this number is sufficient to conveniently deal with Web-scale databases.

3.3 Growing the Seed

We build on the query expansion technique [8] to increase the recall. The idea is as follows: an original query is issued and the results are then used to issue new query. Not all results are used, only those that have the same spatial feature layout (for more details on spatial verification, see the following section). The spatial verification prevents the query expansion from so-called topic drift, where an unrelated image is used to expand the query.

In our approach, we combine two types of query expansion methods suggested in [8]—transitive closure and average expansion. In the transitive closure, each previously unseen (spatially verified) result is used to issue a new query. This method is used to "crawl" the scene. To improve the recall, each query is attempted to be expanded by an average expansion: Result images in which a sufficient number of regions are related by a homography (homographies) to the query image are selected. The homography is then used to backproject features from the result image to the query image (only features within a bounding box of the homography support are mapped). A new query is issued using the combination of the original features and the features gathered from the result images. For efficiency, each image is used at most once for an average query expansion.

If our data mining method is used for obtaining images for 3D reconstruction, a (partial) 3D model can be used for query expansion [8]. To retrieve images from unexplored viewpoints, synthetic views (not necessarily pixelwise) could be generated and used as queries. This is beyond the scope of this paper.

Time complexity. Each query is linear in the number of images in the database. Hence, the time complexity of completing the connected components is $\mathcal{O}(DV)$, where D is the size of the database and V is the number of images in all clusters. The worst-case behavior of this step is thus quadratic, when every image is assigned to one of the clusters. In practice, however, we observe that $V \ll D$, which brings immense computational savings.

Further reduction of the time complexity can be achieved by the following observation. The number of images of one object (say the

Colosseum in Rome) will typically grow with the size of the data set, but the number of different viewpoints gets saturated after certain amount of images is exceeded. Grouping images into similar viewpoints (based on a global descriptor) has been used in [19]. In the proposed approach, for very large clusters (over 500 images), we exclude all images with large number of matches (more than 50) from the query expansion step. This does not have a significant impact on the recall since well matching images usually do not carry sufficient amount of new information to be used in the enhanced query. It also reduces the time complexity to $\mathcal{O}(DL)$, where L is the number of clusters rather than the number of images in all clusters.

3.4 Spatial Verification

In spatial verification, we build on the many-to-many RANSAC-like approach from [6]. Tentative correspondences are defined by a common visual word IDs. The geometric constraint is an affine transformation. This choice is convenient since a single ellipse-to-ellipse correspondence (plus a constraint on the gravity vector) is sufficient to instantiate the approximate model, which is then improved using the local optimization step [31]. The model of affine transformation with loose thresholds allows for detection of close-to-planar structures in the scene with no significant perspective distortion. Unlike in [6], we fit multiple such models. The global consistency of those models is then verified by an RANSAC fitting of an epipolar geometry or homography [32], [33]. This final check is rapid—tentative correspondences for this stage are a union of inlier correspondences from the previous stage and a high inlier ratio is expected (only a few samples are drawn in RANSAC). Since we are fitting an exact model now, the geometric thresholds are set tight.

There are two common sources of mismatches: degenerate configurations of points (close to collinear point sets) and repeated structure (many features assigned to a single visual word, typically repeated in a grid-like structure). In our implementation, in order to positively verify a pair of images, there has to be a sufficient number of matches that are not part of a degenerate or repeated structure.

4 Experimental Evaluation

We have conducted two experiments. The first one checks whether the probability of seed generation is sufficiently high on real data as predicted by theoretical estimates presented in Section 3.2. In the second experiment, clusters of spatially related images are discovered in a database of 100K images.

4.1 Seed Generation Success Rate

To evaluate the success rate of the seed generation stage on real data, we use a standard image retrieval benchmark data set (the University of Kentucky data set) introduced in [5]. This database contains 10,200 images; a group of four images depicts the same object/scene, i.e., there are $2{,}550$ clusters of size 4. The data set provides images, detected image features, and SIFT descriptors. The provided features and descriptors were used. The standard experiment on the database is to query the database with each image in turn, trying to retrieve the other three images from the cluster. Success of the retrieval is measured by the average number of correctly returned images in the top four results (the query image is to be retrieved too). The perfect score is thus 4.

We, however, are interested in a different statistic. The objective is to measure for how many clusters (all of size 4), the proposed method generates at least one seed. For this experiment, we have used a visual vocabulary of 2^{17} visual words. For each image, 512 independent random min-Hash functions were evaluated and grouped into 512 sketches of size 3 (individual min-Hashes were used multiple times). With this setting, there are 11,556 pairs of images with at least one common sketch value (a sketch collision) of which 3,553 passed the similarity test at 0.045 (step 2 of the clustering procedure); out of the 3,553 seeds, 3,210 were within a ground-truth-defined group of four images. The number of clusters of four images for which at least one pair was suggested by the hashing is 1,196 (out of 2,550 possible clusters). In other words, a seed for a cluster of size 4 is generated with a probability of 46.9 percent, which is very close to the expected value of failure. The approximately 50 percent probability of detecting a cluster might seem low, but a cluster of four images is much smaller than typical clusters in image collections containing $10^5 – 10^7$ images. The experiment shows the performance of the algorithm for the smallest practical cluster size.

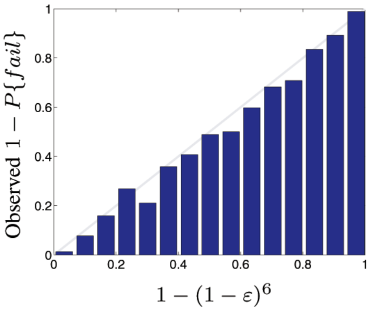

Fig. 5. Histogram of observed success rate plotted against the expected success rate on the Kentucky data set.

In Fig. 5, we compare the predicted success/failure rate (from (5)) and the empirical failure rate. In the experiment, the "average" collision probability ε was computed (exactly) for each cluster by enumerating all image pairs within the cluster. For each cluster, we also observe whether a seed has been generated in the cluster or not. Fig. 5 plots the frequency of observed seed generation success rate for different levels of predicted success rate. The histogram closely follows the gray identity line. We conclude that the prediction given in (5) is precise for the Kentucky data set.

Note that, in this experiment, we are interested only in the false negative rate of the seeding process, not the false positive rate. Potential seeds that are not within a group of four ground-truth images are not necessarily false positives as many objects are presented on the same background. According to the ground truth for the database, such images are in different groups, i.e., declared "spatially unrelated," despite having a significant spatial overlap on the background. Spatial verification filtering was not performed on this data set since we used only data provided by the authors of the database and these do not include information necessary for geometry verification.

If the standard retrieval score was measured, the min-Hash method would reach score of 1.63. This is lower performance than recent retrieval systems that report scores between 3.3 and 3.6. It is important to take into account that min-Hash touches, besides the query image, only 2.27 documents on average. This efficiency (resulting in constant time queries), together with its sufficient recall (46.9 percent success rate for clusters of size 4), proves the min-Hash method suitable for randomized data mining by seed generation.

4.2 Clustering on the 100K Oxford Landmark Database

The experiment was conducted on a large database of images downloaded from Flickr [3]. This database contains 5,062 images from the publicly available Oxford Landmark Database [34] and 99,782 from *Flickr1* data set[4] used in [6]. Both sets are composed of high-resolution images ($1{,}024 \times 768$). The data set consists of images, as well as detected features with SIFT descriptors—these

[4] Courtesy of VGG, University of Oxford.

TABLE 1
Results for Annotated Images in the Oxford Building Data Set

	Good	OK	sketch 3	unrelated	sketch 4
All Souls	24	54	97.44	0	97.44
Ashmolean	12	13	68.00	0	0
Balliol	5	7	33.33	0	0
Bodleian	13	11	95.83	1	95.83
Christ Church	51	27	89.74	0	89.74
Cornmarket	5	4	66.67	0	0
Hertford	35	19	96.30	1	0
Keble	6	1	85.71	0	0
Magdalen	13	41	5.56	0	1.85
Pitt Rivers	3	3	100.00	0	0
Radcliffe Camera	105	116	98.64	0	98.46

The first two columns show the number of ground-truth images labeled as "Good" and "OK," respectively. The column "sketch 3" displays the percentage of labeled images that were clustered into a single cluster using min-Hash with sketches of size 3, "unrelated" gives an absolute number of unrelated images in that cluster. The column "sketch 4" presents results for sketches of size 4.

standard features and descriptors were used in the experiment. Together, there are 104,844 images with 294,105,803 features (2,805 features per image on average). The SIFT descriptors of the features occupy 35 GB. In this data set, images of 11 landmarks were manually labeled. The presence of each landmark in an image is characterized by one of four labels:

1. Good—a nice, clear picture of the object.
2. OK—more than 25 percent of the object is clearly visible.
3. Junk—less than 25 percent of the object is visible or there is a very high level of occlusion or distortion.
4. Absent—the object is not present.

As in the previous experiment, we used a vocabulary with 2^{17} visual words for min-Hash seed generation and with 1M words for seed growing. The Oxford Landmark Database contains clusters with 10^2-10^3 images. To show the potential of the method, we used 512 min-Hashes grouped into 512 sketches of size 3. These settings allow us to discover even small clusters of several images with reasonable probability and are the same as in the University of Kentucky database experiment. On average, the min-Hash generated 38.4 sketch collisions per image. These were reduced to 1.23 potential seeds per image by thresholding the estimated similarity at 0.045—this corresponds to 129,341 seeds. Out of these, 3,103 images were found to have an exact duplicate in the database (the same image was downloaded under different user tags), and 289 images were found to have a near duplicate. Both exact and near duplicates were dropped and the remaining potential seeds were subject to spatial verification, leaving 441 verified seeds. This number is an upper bound on the number of clusters, since typically, there are multiple seeds per cluster. The seed growing by query expansion discovered 354 distinct clusters covering 2,643 images. Cluster examples are shown in Fig. 7 and also in Fig. 1.

Table 1 summarizes the results on objects with the ground-truth information. For each landmark, we found cluster containing the most positive (Good and OK) images of that landmark and computed the fraction of positive ground-truth images in this cluster. Also, the absolute number of unrelated images is reported by eyeballing these clusters. Other buildings that appear in the same cluster are not considered unrelated if images linking these objects exist. For example, images of All Souls and the Radcliffe Camera are all in one cluster—they are right next to each other and appear together on several images.

Clusters corresponding to all ground-truth objects were successfully discovered with the exception of the Magdalen Tower. The percentage of images assigned to the relevant cluster is consistent with the retrieval results in [6], [8] and is related to the "difficulty" of each landmark. This also holds for the "Magdalen"—reported retrieval results were by far the worst for this landmark. In our experiment, three images of the tower were discovered and the method was unable to spatially verify and grow to any other image.

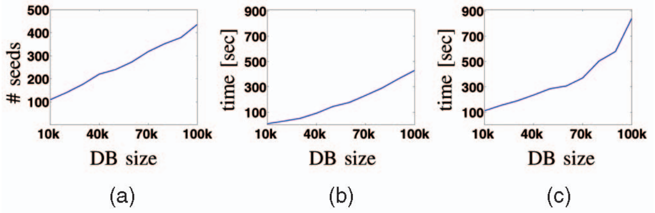

Fig. 6. (a) The number of generated seeds, (b) elapsed time of the seed generation, and (c) elapsed time of the cluster completion as a function of the database size. The values are averaged over 10 executions on the 100K Oxford data set.

Setting sketch size to 3 is suitable for demonstrating the method on a database of 100K images. It allows retrieving even small, perhaps uninterestingly small, clusters. These settings will not be acceptable for Web-scale database size of more 10^7 images or more. To simulate real conditions, we have also used 512 sketches of size 4, which is suitable for very large databases, but returns with acceptable probability only larger clusters. Still, the size of discovered clusters is comparable (or smaller) than the size of clusters used in Photo Tourism [4]. The four largest clusters from the Oxford Landmark ground truth were discovered (together with other larger clusters that are not included in the ground truth). In the case of Magdalen Tower, it is seen on one image of different cluster (Fig. 7 second cluster).

Timing. The seed generation took 7 min 47 sec and the seed growing took 16 min 20 sec on a 2.4 GHz PC using a single processor (MATLAB/MEX implementation). The complete processing of the database took thus slightly more than 24 minutes (the time does not

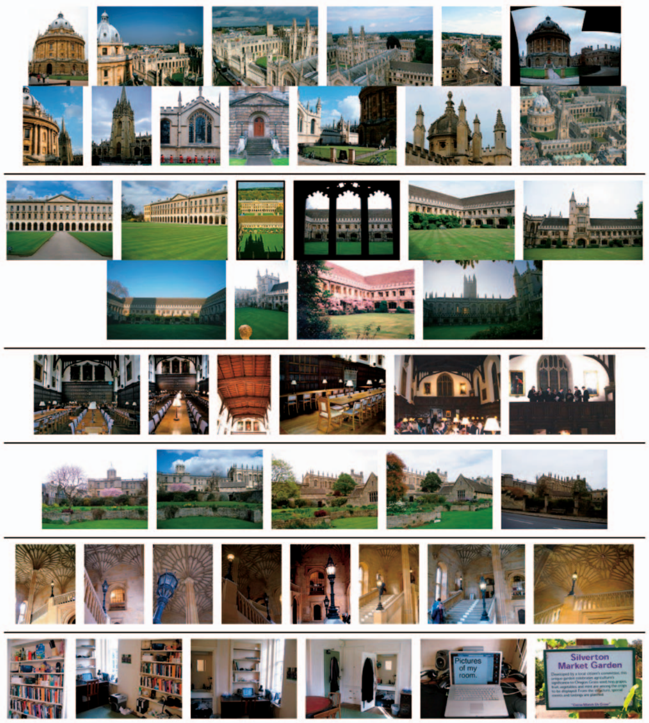

Fig. 7. Selected images from selected clusters discovered in the 100K database including the Oxford Landmark data set. Top: The largest cluster containing the Radcliffe Camera and All Souls (404 images). Below: Discovered clusters of sizes 53, 14, 51, 18, and 13, respectively, not in the ground-truth annotation. The last cluster contains one false positive (the rightmost image), and the other clusters are outlier free. The top four clusters were also discovered in the experiment with sketches of size 4.

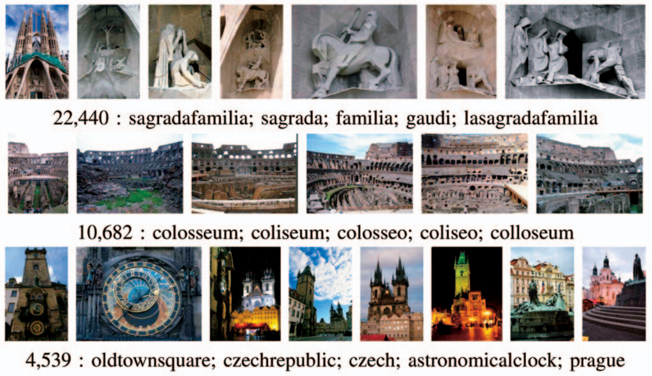
22,440 : sagradafamilia; sagrada; familia; gaudi; lasagradafamilia
10,682 : colosseum; coliseum; colosseo; coliseo; colloseum
4,539 : oldtownsquare; czechrepublic; czech; astronomicalclock; prague

Fig. 8. Sample of large clusters discovered in the 5M database. The size of the cluster and the five most discriminative Flickr tags are shown beneath the images. Note the variety in scale, viewpoint, and illumination conditions.

include the feature extraction, SIFT computation, vector quantization, nor database indexing), which corresponds to 0.014 seconds per processed image. Note that all steps of the proposed method are easy to parallelize.

The influence of the database size on the running time is shown in Fig. 6. The time of seed generation grows approximately linearly (with the slope similar to the number of generated seeds), and the retrieval part grows with both the size of the database and the number of seeds generated. The overall complexity tends toward quadratic. Note that the number of seeds is order of magnitude lower than the size of the database—the randomized clustering is significantly faster than the "query with each image, in turn," approach.

4.3 Large-Scale Clustering of 5 Million Images

We have executed the clustering on a database of 5 million Flicker images. In this experiment, we have used: Hessian affine features [35], a vocabulary of 1M visual words, sketch size $s = 4$, and $k = 512$ sketches. The clustering took slightly under 28 hours on a single machine (3.0 GHz PC, 64 GB memory, using a single core), which is 0.020 seconds per image. Out of the 5M images, 474,434 were assigned to 16,957 clusters. Fig. 8 shows samples of some detected clusters together with the five most discriminative user tags for that particular cluster.

5 Conclusions

We have proposed a method for discovering spatially related images in large-scale image databases. Its speed depends on the size of the database and is very fast in practice and close to linear for database sizes up to approximately $2^{34} \approx 10^{10}$ images. The success rate of cluster discovery is dependent on the cluster size and the average similarity within the cluster and is independent of the size of the database. The properties and performance of the algorithm were demonstrated on data sets with 10^4, 10^5, and 5×10^6 images.

Acknowledgments

The authors would like to thank to Michal Perd'och for discussions and help, and James Philbin for providing the data and his implementation of the spatial verification [6]. Ondřej Chum was supported by Czech Science Foundation Project 102/09/P423, and Jiří Matas was supported by Czech Government grant MSM6840770038 and by EC project ICT-215078 DIPLECS.

References

[1] http://books.google.com/help/maps/streetview/, 2009.

[2] http://www.panoramio.com/, 2009.

[3] http://www.flickr.com/, 2009.

[4] N. Snavely, S. Seitz, and R. Szeliski, "Photo Tourism: Exploring Photo Collections in 3D," *Proc. ACM SIGGRAPH,* pp. 835-846, 2006.

[5] D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2006.

[6] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[7] H. Jegou, H. Harzallah, and C. Schmid, "A Contextual Dissimilarity Measure for Accurate and Efficient Image Search," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[8] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[9] G. Schindler, M. Brown, and R. Szeliski, "City-Scale Location Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[10] F. Schaffalitzky and A. Zisserman, "Multi-View Matching for Unordered Image Sets, or 'How Do I Organize My Holiday Snaps?'" *Proc. European Conf. Computer Vision,* pp. 414-431, 2002.

[11] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *Proc. IEEE Int'l Conf. Computer Vision,* 2003.

[12] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool, "A Comparison of Affine Region Detectors," *Int'l J. Computer Vision,* vol. 65, pp. 43-72, 2005.

[13] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, pp. 91-110, 2004.

[14] J. Sivic and A. Zisserman, "Video Data Mining Using Configurations of Viewpoint Invariant Regions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2004.

[15] J. Philbin and A. Zisserman, "Object Mining Using a Matching Graph on Very Large Image Collections," *Proc. Indian Conf. Computer Vision, Graphics and Image Processing,* 2008.

[16] F. Fraundorfer, H. Stewénius, and D. Nistér, "A Binning Scheme for Fast Hard Drive Based Image Search," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[17] T. Quack, V. Ferrari, and L.V. Gool, "Video Mining with Frequent Itemset Configurations," *Proc. Int'l Conf. Image and Video Retrieval,* 2006.

[18] T. Quack, B. Leibe, and L.V. Gool, "World-Scale Mining of Objects and Events from Community Photo Collections," *Proc. Int'l Conf. Image and Video Retrieval,* 2008.

[19] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.M. Frahm, "Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs," *Proc. European Conf. Computer Vision,* 2008.

[20] A. Oliva and A. Torralba, "Building the Gist of a Scene: The Role of Global Image Features in Recognition," *Visual Perception, Progress in Brain Research,* vol. 155, pp. 23-36, 2006.

[21] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering Object Categories in Image Collections," Technical Report A.I. Memo 2005-005, Massachusetts Inst. of Technology, 2005.

[22] T. Hofmann, "Probabilistic Latent Semantic Indexing," *Proc. ACM SIGIR,* 1999.

[23] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation," *J. Machine Learning Research,* vol. 3, pp. 993-1022, 2003.

[24] B.C. Russell, A.A. Efros, J. Sivic, W.T. Freeman, and A. Zisserman, "Using Multiple Segmentations to Discover Objects and Their Extent in Image Collections," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2006.

[25] J. Philbin, J. Sivic, and A. Zisserman, "Geometric LDA: A Generative Model for Particular Object Discovery," *Proc. British Machine Vision Conf.,* 2008.

[26] A. Broder, "On the Resemblance and Containment of Documents," *Proc. SEQS: Sequences '91,* 1998.

[27] O. Chum, J. Philbin, M. Isard, and A. Zisserman, "Scalable Near Identical Image and Shot Detection," *Proc. Int'l Conf. Image and Video Retrieval,* 2007.

[28] P. Indyk and R. Motwani, "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality," *Proc. Symp. Theory of Computing,* 1998.

[29] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM,* vol. 24, pp. 381-395, 1981.

[30] O. Chum, J. Philbin, and A. Zisserman, "Near Duplicate Image Detection: min-Hash and TF-IDF Weighting," *Proc. British Machine Vision Conf.,* 2008.

[31] O. Chum, J. Matas, and Š. Obdržálek, "Enhancing RANSAC by Generalized Model Optimization," *Proc. Asian Conf. Computer Vision,* 2004.

[32] O. Chum, T. Werner, and J. Matas, "Epipolar Geometry Estimation Unaffected by the Dominant Plane," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2005.

[33] J.M. Frahm and M. Pollefeys, "RANSAC for (Quasi-) Degenerate Data (QDEGSAC)," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2006.

[34] http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/, 2009.

[35] M. Perdoch, O. Chum, and J. Matas, "Efficient Representation of Local Geometry for Large Scale Object Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.