Fast Computation of min-Hash Signatures for Image Collections

Ondřej Chum Jiří Matas

Centre for Machine Perception
Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague
chum@cmp.felk.cvut.cz

Abstract

A new method for highly efficient min-Hash generation for document collections is proposed. It exploits the inverted file structure which is available in many applications based on a bag or a set of words. Fast min-Hash generation is important in applications such as image clustering where good recall and precision requires a large number of min-Hash signatures.

Using the set of words represenation, the novel exact min-Hash generation algorithm achieves approximately a 50-fold speed-up on two dataset with 10^5 and 10^6 images respectively. We also propose an approximate min-Hash assignment process which reaches a more than 200-fold speed-up at the cost of missing about 2-3% of matches.

We also experimentally show that the method generalizes to other modalities with significantly different statistics.

1. Introduction

In the last decade, very large collections of images have become readily available. Discovering groups of images of the same object or landmark in a very large collection is a challenging problem with a number of applications like city-size reconstruction [6], image clustering [3] and the discovery of canonical views [11]. Importantly, sets of images of objects or surfaces acquired in a range of viewing and illumination conditions provide input, when correspondences are established, for machine learning techniques applied to computer vision problems. Data-driven approaches have become very popular and often significantly outperform manually designed solutions. Examples of recent developments in this domain include descriptor learning [12, 13], learning of descriptor distances [9], and improved feature space quantization [10].

In problems where generic solutions are sought, the diversity of the data, *e.g.* of the locations and scenes, is an important factor that prevents a bias of all subsequent results. The required amount and precision of such low-level (correspondence) annotation by far exceeds the capabilities of human annotation, including cheap internet-based efforts. The training data collections should simulate the situation that all possible data have been seen during the training phase.

The min-Hash has been shown in [3] to scale well to web-size collections of images and to support establishing correspondence and finding spatially-related images in such collections. The clustering method presented in [3] consists of two stages: first, min-Hash matching is used to generate the so called cluster seeds; in the second stage, the clusters are formed around the seeds.

In the paper, we address the first step – generating the min-Hash signatures. We propose two variants, one approximate and one exact, of a novel min-Hash construction method. The exact variant results in approximately a 50-fold speed-up over the standard min-Hash, while delivering identical results. The approximate variant offers a user-controlled trade off between speed and accuracy. For instance, at the recall reduced by about 2- 3% the min-Hash generation speed is improved more then 200 times.

Fast min-Hash generation is important in applications such as clustering and matching where low overlaps of visual words are encountered[1]. In such cases, good recall and precision of the constant-time hashing-based matching method requires a large number (hundreds to thousands) of min-Hash sketches and thus signatures [3]. If a single min-Hash generation for all images in a collection takes tens of seconds, a speed-up of two orders of magnitude has a strong practical impact. Further more, for difficult image pairs (with low similarity) increasing the number of min-Hash signatures increases the chance of match discovery close to linearly, see Figure 1(c).

The rest of the paper is structured as follows. First, the relevant background on min-Hash is reviewed in Section 2. The proposed method for min-Hash speed-up, both the ex-

[$]The authors were supported by the following projects: GACR P103/12/2310 and EC FP7-ICT-247022 MASH.

[1]Unlike in near duplicate detection

act and approximate variant, is described in Section 3. Experimental evaluation follows in Section 4.

2. Background Review : min-Hash

Before presenting the highly efficient method for computation of min-Hash for a collection of images, the essentials necessary for understanding min-Hash techniques are reviewed. A detailed description is given in *e.g.* [2, 5].

The min-Hash algorithm is a Locality Sensitive Hashing [7] for sets. In the min-Hash method, 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). However, it was shown that for large vocabularies the set of words and bag of words representations are almost identical [4].

There is a number of equivalent definitions of the min-Hash. It will be convenient to use the definition exploiting ordering of the vocabulary by a random permutation π. Let N be the size of the vocabulary and

$$\pi(i) : \{1 \dots N\} \rightarrow \{1 \dots N\}$$

a permutation of N elements. Let $p(i)$ be an inverse function to $\pi(i)$. That is, $\pi(i)$ gives the rank of visual words w_i, while $p(i)$ is the index of i-th smallest visual word in the ordering induced by π.

A min-Hash signature of a set $\mathcal{A}$ is defined as $h(\mathcal{A})$ where

$$h(\mathcal{A}) = \min_{i:w_i \in \mathcal{A}} \pi(i). \tag{1}$$

Such a function has the property that the probability of two sets having the same value of the min-Hash signature is equal to their set overlap, *i.e.* the ratio of the cardinalities of the intersection and union of the two sets. 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' and the set overlap interchangeably:

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

For a random permutation π the probability of two images to have the same min-Hash signature is then

$$P\{h(\mathcal{A}_1) = h(\mathcal{A}_2)\} = \text{sim}(\mathcal{A}_1, \mathcal{A}_2).$$

To estimate the similarity of two images, multiple independent min-Hash functions h_j (*i.e.* independent random permutations π_j of the vocabulary) are used. The fraction of the min-Hash functions that assign an identical min-Hash signature to the two sets is an unbiased estimate of the similarity of the two images.

Retrieving similar images. 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 functions h_i are grouped into s-tuples called sketches. Similar images have identical values of the min-Hash signature for many random permutations π_i (by the definition of similarity) and hence have a 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.

The probability of two sets having at least one sketch (of size s) out of r in common, *i.e.* the probability of at least one collision, is

$$P_C(\mathcal{A}_1, \mathcal{A}_2) = 1 - (1 - \text{sim}(\mathcal{A}_1, \mathcal{A}_2)^s)^r. \tag{3}$$

The probability depends on the similarity of the two images and on the two parameters of the method: the size of the sketch s and the number of (independent) sketches r. Figure 1 (a) and (b) visualizes the probability of collision plotted against the similarity of two images for fixed $s = 3$ and $r = 512$. Figure 9 shows examples of image pairs and their similarity.

Word weighting. It has been shown that different features carry different amount of information and that weighing visual words by their relative importance improves retrieval quality [1]. An extension to min-Hash proposed in [5] introduces a method of vocabulary permutation generation that allows to assign different weights to different features. Let $d_i \geq 0$ be the importance of a visual word w_i. The weighted set overlap similarity of two sets $\mathcal{A}_1$ and $\mathcal{A}_2$ is

$$\text{sim}_w(\mathcal{A}_1, \mathcal{A}_2) = \frac{\sum_{w_i \in \mathcal{A}_1 \cap \mathcal{A}_2} d_i}{\sum_{w_i \in \mathcal{A}_1 \cup \mathcal{A}_2} d_i}. \tag{4}$$

It was shown that the weighted measure (with *idf* in [5]) has two advantages compared with the original set overlap: it better captures the image similarity, and reduces the number of false sketch collisions. All plots in the paper were generated using *idf* weighted min-Hash, plots for standard min-Hash are indistinguishable.

3. Fast generation of min-Hash signatures for image collections

In this section, we present a novel efficient construction of min-Hash representation for large image collections. The process is explained for a single min-Hash function h. To generate multiple min-Hash signatures, the procedure is repeated with different hash function, *i.e.* with the different random permutations of the vocabulary. Let N be the size of the vocabulary. Let a permutation of the vocabulary be π and its inverse p be defined as in Section 2.

In the standard min-Hash, the signature is generated for each image separately by selecting a visual word present

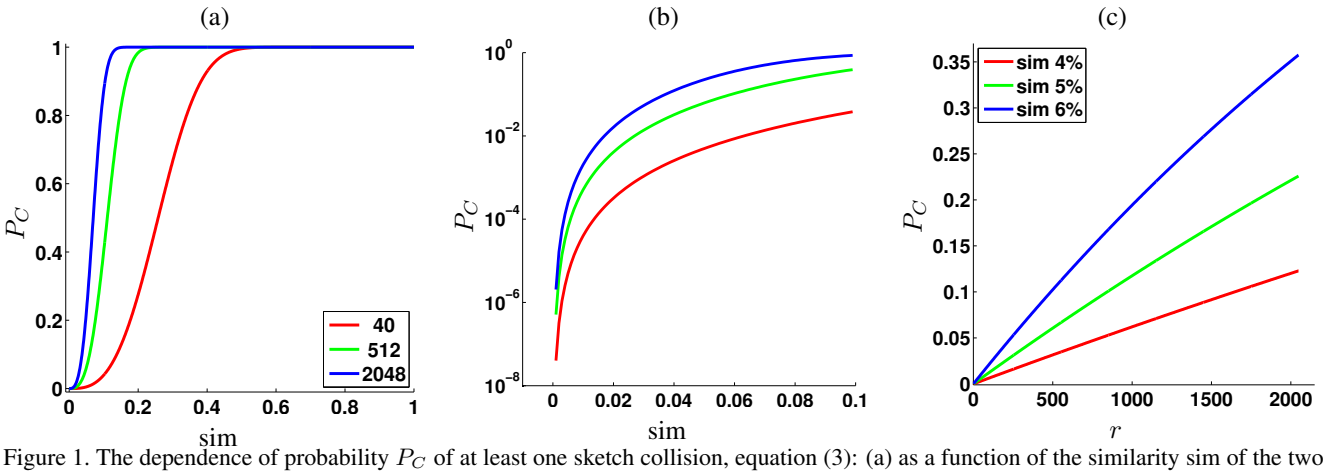

Figure 1. The dependence of probability P_C of at least one sketch collision, equation (3): (a) as a function of the similarity sim of the two images for for $r = 40$, 512 and 2048 ; (b) a close up of the same plot – note the logarithmic scale on the vertical axis; (c) as a function of the number of sketches r, the probability P_C increases almost linearly with the number of sketches r for similarities sim $= 0.04$, 0.05, and 0.06.

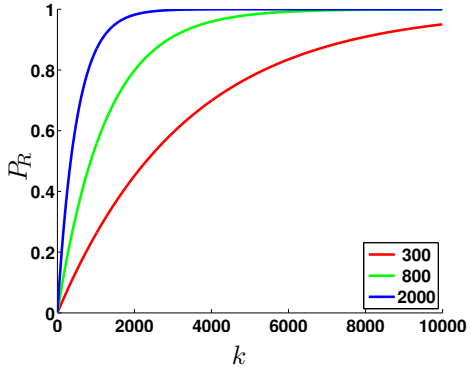

Figure 2. The dependence of the probability P_R of assigning a single min-Hash for an image with 300, 800, and 2000 visual words respectively after processing k visual words. The size of the vocabulary is 1M visual words. Note that at $k = 10000$ only 1% of the inverted lists has been processed.

Input: $p(i)$ - ordering of the vocabulary, k - the number of visual words to be used
Output: $M[1:D]$ array of min-Hash signatures for all D images

initialize $M[1:D] =$ **NDef**
for $i = 1:k$
  for every image j containing visual word $w_{p(i)}$ **do**
    if $M[j] ==$ **NDef then**
      $M[j] = i$
    end
  end
end

Algorithm 1. Partial min-Hash signature generation using the inverted file.

in the image with the smallest value of h. In the proposed method, documents with the same min-Hash are processed in a single step of the assignment process. To do so efficiently, the inverted file structure used in the image retrieval is exploited.

The assignment procedure proceeds as follows. From the definition, visual word $w_{p(1)}$ is the smallest element from the whole vocabulary with respect to ordering induced by π. Thus, any document containing $w_{p(1)}$ will have this the visual word as its min-Hash signature (for hash function h induced by π). Id-s of all such documents are stored in a list of the inverted file associated with $w_{p(1)}$. Similarly, visual word $w_{p(2)}$ will be a min-Hash signature of all documents that contain $w_{p(2)}$ and *do not* contain $w_{p(1)}$. In general, $w_{p(i)}$ is a min-Hash signature for exactly those documents that contain visual word $w_{p(i)}$ and do not contain any visual word $w_{p(j)}$, where $j < i$. This leads to the following simple algorithm (summarized in Algorithm 1).

At the beginning, all images are marked as not having a min-Hash defined. Then, starting with $w_{p(1)}$, the inverted lists are scanned in the order given by $p(i)$. All images in a list corresponding to $w_{p(i)}$ with undefined min-Hash are assigned the min-Hash value $\pi(p(i)) = i$. When the algorithm passes through all lists of the inverted file, all non-empty images are guaranteed to have a min-Hash signature assigned. However, the complexity of such an approach is the same as the complexity of the standard min-Hash construction – each feature in each document is touched once.

Clearly, the procedure is very efficient at the beginning of the assignment process when almost none of images have a min-Hash assigned. Gradually, the inverted lists include higher and higher proportion of images with min-Hash assigned from previous iterations.

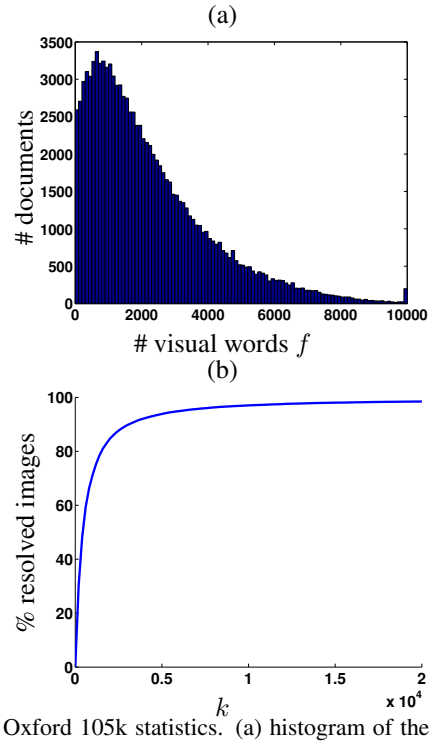

Figure 3. Oxford 105k statistics. (a) histogram of the number of unique visual words per image, (b) the fraction of resolved images during processing top k visual words, note that $k = 2 \cdot 10^4$ represents only 2% of visual words.

3.1. Partial min-Hash assignment with inverted lists

In the following paragraphs, we study what happens if the procedure does not consider all visual words, but is terminated after traversing only first k lists corresponding to visual words $w_{p(1)}$ to $w_{p(k)}$.

Let us consider the probability that a particular image has a min-Hash signature assigned after processing the first k visual words by traversing their list in the inverted file. The probability P_R that a document will be resolved (will have a min-Hash signature defined) depends not only on the number of processed lists k but and also on the number of different visual words f in that image. The probability that an image with f visual words, uniformly and independently drawn from the vocabulary of N words, includes a visual word is equal to f/N. The probability of resolving such an image after k visual words is approximated by

$$P_R(f,k) = 1 - \left(1 - \frac{f}{N}\right)^k. \qquad (5)$$

The relation would be exact if the k visual words were selected with replacement, in our case they are selected by the permutation without replacement. Equation (5) provides a lower bound and it is a close approximation for $k \ll N$. Figure 2 shows the probability P_R for a vocabulary of size $N = 1$M and different values of k and f. The number of

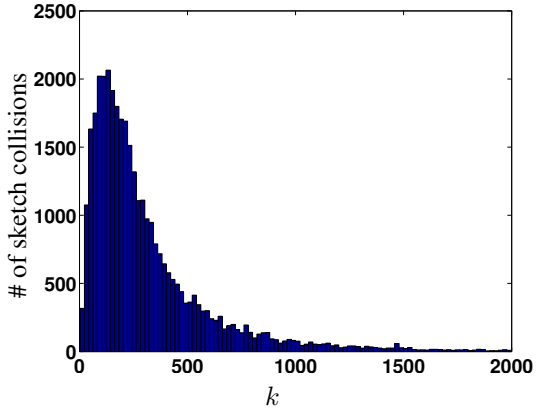

Figure 4. Histogram of the number of sketch collisions involving sketches resolved after considering k inverted lists for Oxford 105k dataset. Only 2.0% of collisions lie beyond $k = 2000$.

resolved images depends on the distribution of the number f of unique visual words in images

$$E[P_R(k)] = \sum_f P(f) P_R(f,k). \qquad (6)$$

To show the behaviour on a realistic database we use the Oxford 105k as a running example. Figure 3(a) shows the relative frequencies, used as an estimate of $P(f)$. The distribution peaks at about 1000 visual words. The dependence in eqn. (6) is made explicit in Figure 3(b). Virtually all images have a min-Hash assigned for k=10 000, that is after considering about 1% of all inverted files.

An empirical dependence of the success rate of the partial min-Hash on the number of used lists of the inverted file is shown in Figure 4. A brief mathematical analysis of the dependence is presented in appendix A.

3.2. Fast exact min-Hash calculation

For some applications it might be necessary, or convenient, to generate the min-Hash signature for all images. To exploit the efficient procedure using the inverted file structure, we propose a hybrid algorithm which is exact in the sense that every image is assigned the same min-Hash as with the standard algorithm. For the top k inverted lists, the algorithm described in the previous subsection is used. Then, the min-Hash signature of the unresolved images is obtained as in the standard min-Hash – minimal element with respect to the permutation π will be selected from each unresolved image.

The complexity of this second step is given by the product of the number of unresolved images and by the average number of visual words in those images. The number of unresolved images is derived from equation (6) as $D(1 - E[P_R(k)])$, where D is the total number of images, see Figure 5(a). The average number of visual words in un-

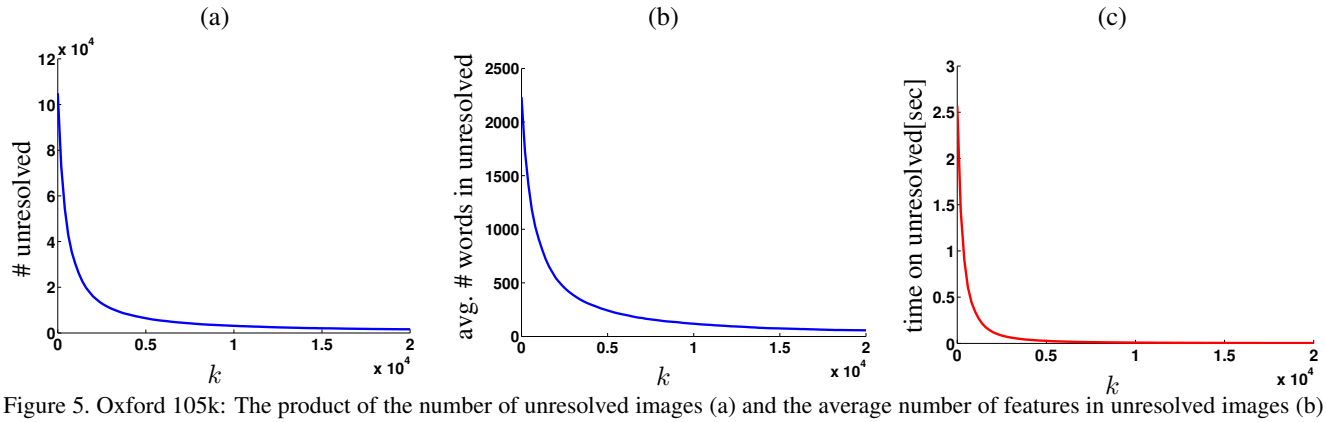

Figure 5. Oxford 105k: The product of the number of unresolved images (a) and the average number of features in unresolved images (b) determines the time (c) required to resolve all unresolved images after processing k inverted lists.

resolved images after processing first k visual words is

$$\mathrm{E}[f_N(k)] = \frac{\sum_f fP(f)(1-P_R(f,k))}{\sum_f P(f)(1-P_R(f,k))}. \qquad (7)$$

Since the probability P_R increases with the number of visual words in an image, more images with a large number of visual words will be resolved with every processed list of the inverted file. Hence, the expected number of visual words in unresolved images after k iterations is a decreasing function of k, as shown in Figure 5(b) for the Oxford 105k dataset. The time complexity of the hybrid method as a function of k is plotted in Figure 6.

Note on the weighted min-Hash. The *idf* weighted min-Hash algorithm preferably assigns low ranks in the ordering π to visual words from shorter lists of the inverted file (corresponding to visual words with a higher weight). This renders the statistical analysis intractable. However, our experiments empirically show that the efficiency of the proposed algorithm is the same for both weighted and the standard similarity measure.

3.3. Sketches with the NDef symbol

In this section, a "lazy" version of the exact algorithm is proposed. Instead of using a standard min-Hash algorithm to generate the min-Hash signature for all unresolved images, a special symbol **NDef** is used as a min-Hash signature for such documents. Evaluating the min-Hash function is postponed and executed only for images that have a matching sketch that includes the **NDef** symbol. The following table shows different examples of sketches of size 2 of two images.

	Image 1	Image 2	sketch collision
1	(1, 2)	(1, 2)	matching
2	(1, 3)	(2,3)	non-matching
3	(1,**NDef**)	(2, **NDef**)	non-matching
4	(1,2)	(1, **NDef**)	non-matching
5	(1, **NDef**)	(1, **NDef**)	could match

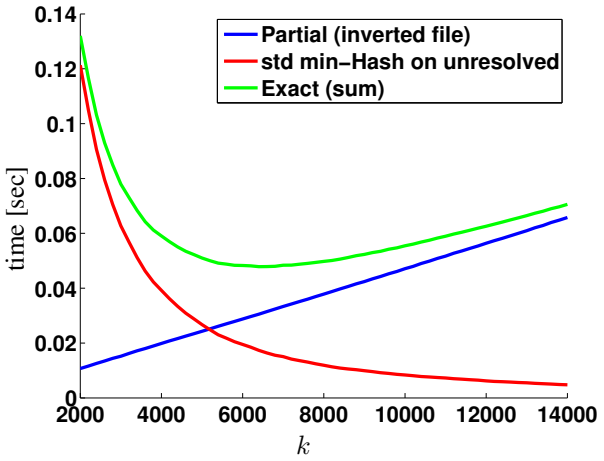

Figure 6. Time complexity of the exact algorithm on Oxford 105k for a range of k values (green curve). The optimum is reached for $k^* \approx 6400$. First, the inverted file is used to generate min-Hash signatures up to the k-th smallest visual word in the vocabulary (blue line). The unresolved images are subject to standard min-Hash signature generation procedure (red curve). The exact method achieves more than 50-fold speed-up taking 48 milliseconds compared to the standard min-Hash applied to all images taking 2.57 sec, which is the value of the red curve at k=0. Note that the horizontal axis starts at the value of $k = 2000$. The red curve (std min-Hash on unresolved) corresponds to the curve in Figure 5(c).

The **NDef** symbol acts as a new element of the vocabulary. In order to define a sketch collision, all entries of the sketch must be identical. Therefore, the examples in rows 2 and 3 are not matching, because the first entry of the sketch is different. The example in row 4 cannot be matching either: even though the second image has an the second min-Hash signature unresolved, it cannot be equal to 2 which is the min-Hash signature of the first image. This is due to the fact that the list of the inverted file associated with w_2 has been processed (is has generated the min-Hash signature for

the first image) and the second image is certainly not containing w_2, otherwise the min-Hash signature would have been resolved.

The sketch hashing follows the same procedure as in the standard. If a collision of a sketch containing a **NDef** symbol was encountered, the unresolved min-Hash signatures are obtained in the standard manner. The sketches are consequently compared again to verify whether the collision is valid or not. This way, some images do not have some of their min-Hash signatures (without the **NDef** symbol) generated at all, since these are not necessary.

The lazy evaluation can reduce the number of evaluated unresolved min-Hashes to approximately one third, depending on the number k of the inverted lists processed (see Figure 11). This may be useful for applications where the exact min-Hash is required, but only the inverted file resides in the memory, while the access to the data for standard min-Hash is slower.

4. Experiments

In this section, we experimentally measure the speed-up achieved by the proposed approach. To demonstrate the wide applicability two different modalities are considered: images represented by sets of visual words and binary images represented as a set of pixels.

4.1. Visual words

Besides the Oxford 105k dataset, the speed of the exact and partial min-Hash methods was validated on a collection of 5 million image downloaded from Flickr. Let us first consider the partial algorithm.

The speed-up of the partial method depends on the loss of collisions that can be tolerated. Fortunately, a min-Hash signature with a high value is unlikely to be generated and, as analysed in Appendix A, even less likely to lead to a collision. This dependence is visualised in Figures 4 and 8. The full speed-up vs. loss of collisions curve is presented in Figure 7. For instance, at $k = 2000$ a speed-up of 240 (Oxford 105) and 215 (Flickr 5M) is achieved at the loss of 2.1% (Oxford 105k) and 3.1%(Flickr 5M) collisions respectively.

Figure 9 shows samples of image pairs retrieved by a sketch collisions in Oxford 105k. The image pairs are ordered by the largest value of the min-Hash signature in the colliding sketch. If a lower number k of lists of the inverted file were processed, the sketch would have been undefined. Note the relation between the number of features in the images and the value of k. Images with low number of features may require high values of k to be matched by a sketch collision, even if the images are exact duplicates.

The results of the exact variant for the Oxford 105k image collection shown in Figure 6 are qualitatively the same as the results on the Flickr 5M dataset, see Figure 10; note

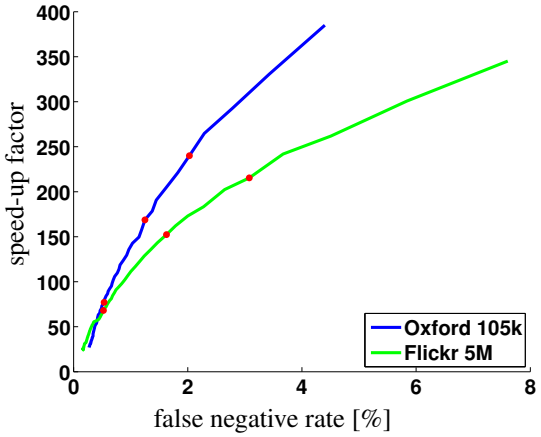

Figure 7. The trade-off between the accuracy and the speed of the partial algorithm for Oxford 105k and Flickr 5M datasets. Points representing $k = 2000$, 3000, and 7000 are marked on the curves (the lower k the higher speed-up).

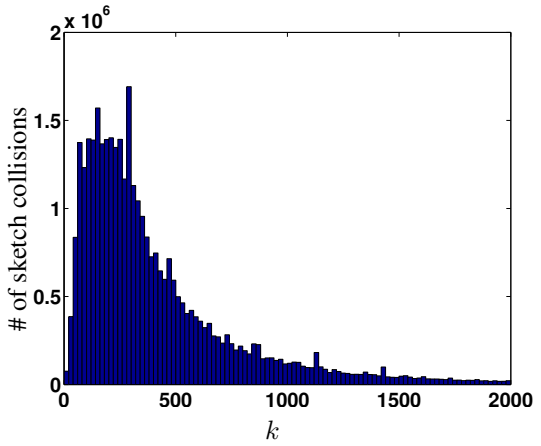

Figure 8. Histogram of the number of sketch collisions of sketches resolved after k inverted lists for Flicker 5M dataset. Only 3.1% of collisions lie beyond $k = 2000$.

the different time scales. The optimal values of k, *i.e.* the optimal number of inverted lists scanned, is equal to 6400 (Oxford 105k) and to 7000 (Flickr 5M) respectively[2]. The speed-up achieved at the optimal k is 50 (Oxford 105k) and 45 (Flickr 5M) times respectively. The performance is insensitive to the exact value of k, both Figures 6 and 10 show a broad valley with near-optimal performance between 6000 to 8000 visual words that are assigned using inverted files.

Finally, we study the impact of the lazy min-Hash generation. Figure 11 demonstrates the gain of the method evaluated on Oxford 105k dataset. For low values of k, there are many unresolved min-Hash signatures in the sketches. This causes a large number of sketches to match on the **NDef** symbols, which requires high fraction of the min-Hash sig-

[2]Performance was evaluated only for values of k that are multiples of 200

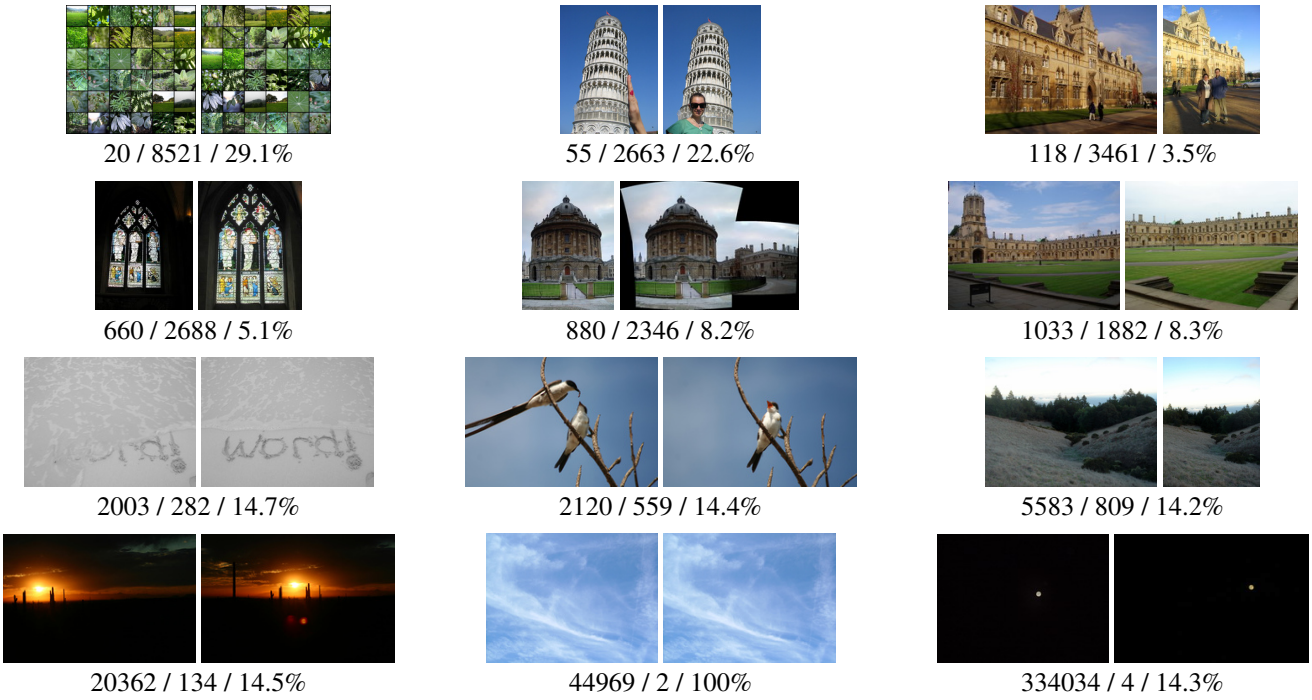

Figure 9. Sample image pairs retrieved from the Oxford 105k dataset by min-Hash with r=512 sketches of size s=3. Three characteristics are shown for each pair: the minimal k (out of 1M vocabulary) required to resolve a whole sketch / the average of the numbers of features in the two images / similarity. Note that the pairs are not selected uniformly from the pairs with sketch collisions – the top two rows represent 98% of the colliding pairs, while the bottom two rows represent only 2%.

natures to be resolved. For values of $k > 5000$, the method reduces the number of required min-Hashes to be resolved to approximately one third.

4.2. Binary images

To show the applicability in other domains than a set of visual words retrieval, the gain in performance of the proposed method on binary image matching was measured . A database of 70k binarized images of hand-written digits MNIST [8] was used. Each image is 28×28 pixels, yielding a 784 dimensional binary descriptor (compared with 1M dimensions used in the previous experiments). Each image is represented by a set of pixels that are black. Images on average contain 103 elements, which corresponds to the sparsity of 13% of non-zero (compare with 2% in the the set of visual words). Despite significantly different nature of this dataset, the proposed method still brings a significant 15 fold speed-up, see fig. 12.

5. Conclusions

We have presented a method for efficient exact and partial assignment of min-Hash signatures to a large collection of images that exploits the inverted file structure. A fast min-Hash generation is important in applications since a good recall and precision of this constant-time matching methods requires a fairly large number of min-Hash sketches and thus signatures.

We have shown that an approximately 50 times speed-up was achieved on two datasets with 10^5 and 10^6 images respectively for the fast exact min-Hash algorithm. An approximate min-Hash assignment process reached more than 200-fold speed-up at the cost of missing about 2-3% of matches.

Experimentally, it was shown that the method generalizes to other modalities with significantly different statistics.

References

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

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

[3] O. Chum and J. Matas. Large-scale discovery of spatially related images. *IEEE PAMI*, 32:371–377, 2010.

[4] O. Chum, M. Perdoch, and J. Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In *CVPR*, 2009.

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

[6] J.-M. Frahm, P. F. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, and S. Lazebnik.

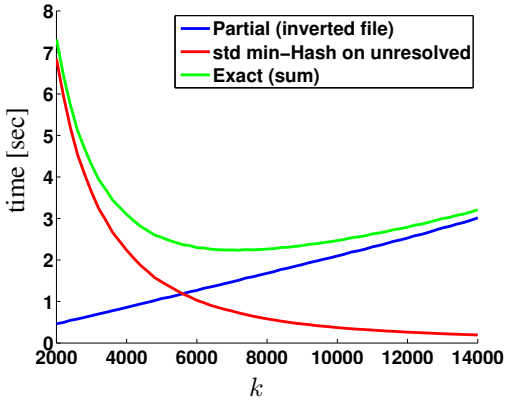

Figure 10. Time complexity of the exact algorithm on Flickr 5M for a range of k values (green curve). The optimum is reached for $k^* \approx 7000$. Inverted file min-Hash signature generation time up to the k-th visual word (blue line). Time of the standard min-Hash signature generation procedure (red curve). The exact method achieves more than 45-fold speed-up taking 2.2 seconds compared to the standard min-Hash applied to all images taking 100.2 sec.

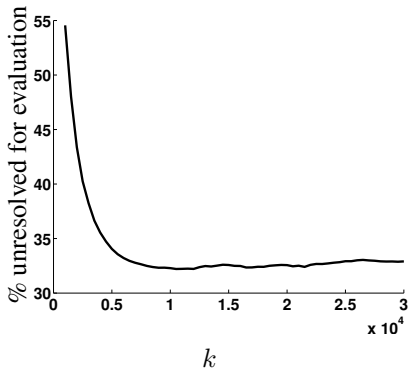

Figure 11. Fraction of the unresolved min-Hash signatures to be evaluated after the collision in the lazy min-Hash generation scheme.

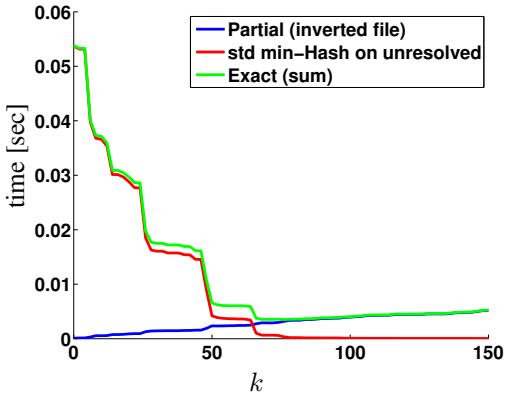

Figure 12. Time complexity of the exact algorithm on binarized MNIST 70k for a range of k values (green curve). The optimum is reached for $k^* \approx 75$. A 15 fold speed-up was achieved over the standard min-Hash.

Building rome on a cloudless day. In *Proc. ECCV*, pages 368–381, 2010.

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

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998.

[9] K. Mikolajczyk and J. Matas. Improving sift for fast tree matching by optimal linear projection. In *Proc. ICCV*, 2007.

[10] A. Mikulik, M. Perdoch, O. Chum, and J. Matas. Learning a fine vocabulary. In *Proc. ECCV*, pages 1–14. Springer, 2010.

[11] T. Weyand and B. Leibe. Discovering favorite views of popular places with iconoid shift. In *ICCV*, 2011.

[12] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *Proc. CVPR*, 2009.

[13] S. A. J. Winder and M. Brown. Learning local image descriptors. In *Proc. CVPR*, 2007.

A. Efficiency of the partial min-Hash

The relative success rate of the partial variant w.r.t. to the exact algorithm is given by the fraction of sketch collisions that are missed, that is collision of sketches containing a min-Hash signature with value exceeding the maximal number k^* of processed lists of the inverted file. The probability that a min-Hash matches with the value k of the signature is given by two factors. First, the probability that a min-Hash will take the value k and that the min-Hash will be matching given its value is k. The first probability is (up to the normalizing factor) shown in Figure 5 (a).

Now we analyse the probability that a min-Hash is matching given its value is k. Let the min-Hash of a set $\mathcal{A}_1$ be k, that is

$$\forall w_i \in \mathcal{A}_1 : \pi(i) \geq k. \tag{8}$$

A set $\mathcal{A}_2$ will have identical value of the min-Hash iff $w_i \in \mathcal{A}_2$ and

$$\forall w_i \in \mathcal{A}_2 : \pi(i) \geq k.$$

From equation (8), the above condition is satisfied for $w_i \in \mathcal{A}_1 \cap \mathcal{A}_2$, and the probability that it will be also satisfied by the elements of $\mathcal{A}_2 \setminus \mathcal{A}_1$ is closely approximated by

$$\left(1 - \frac{k-1}{N - |\mathcal{A}_1|}\right)^{|\mathcal{A}_2 \setminus \mathcal{A}_1|}. \tag{9}$$

In other words, the expression (9) gives a probability that none of the visual words in $\mathcal{A}_2$ that are not present in $\mathcal{A}_1$ will have smaller hash value than k, given the min-Hash of $\mathcal{A}_1$ is $\pi(w_i) = k$ (in which case the sets $\mathcal{A}_1$ and $\mathcal{A}_2$ would have different min-Hash signatures). The probability is a polynomial of degree $|\mathcal{A}_2 \setminus \mathcal{A}_1|$ in k. For larger values of $|\mathcal{A}_2 \setminus \mathcal{A}_1|$ approaches (9) zero fast.