Evaluation of GIST descriptors for web-scale image search

Matthijs Douze, Hervé Jégou, Sandhawalia Harsimrat, Laurent Amsaleg, Cordelia Schmid

Evaluation of GIST descriptors for web-scale image search

Matthijs Douze Hervé Jégou Harsimrat Sandhawalia
Laurent Amsaleg Cordelia Schmidt

2009

Abstract

The GIST descriptor has recently received increasing attention in the context of scene recognition. In this paper we evaluate the search accuracy and complexity of the global GIST descriptor for two applications, for which a local description is usually preferred: same location/object recognition and copy detection. We identify the cases in which a global description can reasonably be used.

The comparison is performed against a state-of-the-art bag-of-features representation. To evaluate the impact of GIST's spatial grid, we compare GIST with a bag-of-features restricted to the same spatial grid as in GIST.

Finally, we propose an indexing strategy for global descriptors that optimizes the trade-off between memory usage and precision. Our scheme provides a reasonable accuracy in some widespread application cases together with very high efficiency: In our experiments, querying an image database of 110 million images takes 0.18 second per image on a single machine. For common copyright attacks, this efficiency is obtained without noticeably sacrificing the search accuracy compared with state-of-the-art approaches.

1 Introduction

Web-scale image indexing requires the description and storage of billions of images. It is, therefore, important to describe an image as compactly as possible and to develop efficient indexing strategies. There exists a trade-off between the precision of an image description and its size. Storing all the information contained in an image efficiently is impossible for a large image collection. On the other hand, storing just a few bits is not sufficient to distinguish between a large number of images.

Currently, two types of approaches are popular for web-scale image indexing. The first one uses global descriptors, in particular GIST [16], and the second

one is based on a bag-of-features (BOF) [20]. In the following we briefly review these two approaches and discuss their advantages and drawbacks.

The GIST descriptor was initially proposed in [16]. The idea is to develop a low dimensional representation of the scene, which does not require any form of segmentation. The authors propose a set of perceptual dimensions (naturalness, openness, roughness, expansion, ruggedness) that represent the dominant spatial structure of a scene. They show that these dimensions may be reliably estimated using spectral and coarsely localized information. The image is divided into a 4-by-4 grid for which orientation histograms are extracted. Note that the descriptor is similar in spirit to the local SIFT descriptor [11].

The GIST descriptor has recently shown good results for image search. In Li et al. [10] GIST is used to retrieve an initial set of images of the same landmarks, for example the statue of liberty, and then image point based matching is used to refine the results and to build a 3D model of the landmark. In Hayes and Efros [4] it is used for image completion. Given a huge database of photographs gathered from the web the algorithm patches up holes in images by finding similar image regions in the database based on the GIST descriptor. Torralba et al. [23, 24] developed different strategies to compress the GIST descriptor.

BOF image search systems [20] first extract a set of local descriptors for each image, such as the popular SIFT descriptor [11]. Combined with appropriate region detectors [12, 14], these descriptors are very discriminant and invariant to local transformations. Furthermore, image comparison based on local description is robust to cropping, clutter, change in viewpoint, illumination change, etc. The BOF representation is built from such a set of local descriptors. The key idea of using this representation is to mimic state-of-the-art text retrieval systems, and in particular to exploit the *inverted file* indexing structure [25]. This structure efficiently computes any Minkowski distance [15] between sparse vectors, which is especially of interest for a document representation based on term frequencies and its variants [19]. For this purpose, a visual vocabulary is required to transform the continuous feature space into a discrete word space. This step consists in learning a vector quantizer, typically by k-means clustering, and in using it to map the descriptors into *visual words* (forming a *visual vocabulary*): descriptors are quantized by finding their nearest centroid.

Different strategies have been proposed to improve BOF-based image search. For instance [15] introduces a hierarchical visual vocabulary that improves the search efficiency. Re-ranking based on the estimation of a geometrical transform [17], as well as query expansion [1] –inspired by text retrieval–, improves the accuracy. However, these methods can be used for a limited number of images only, because of the high cost of estimating complex geometrical transformations with individual database images. Using a richer descriptor representation and geometrical consistency [5] improves search accuracy while maintaining similar query times.

The BOF representation of images was proved be very discriminant and efficient for image search on millions of images [5, 15]. However, web-scale

indexing based on this approach suffers from two limitations: complexity and memory usage. Different strategies have been proposed to overcome the first one. The vocabulary size can be increased [15], but only to some extent, as the accuracy decreases for very large vocabularies [17]. Another strategy is to use approximate search, as done in [7] by using a two-level inverted file structure or in [2] using min-Hash. However, it is not clear how to use these schemes on scales ranging from 100 million to 10 billion images. Memory usage is the other problem, as the BOF indexing structure does not fit into memory for web-scale datasets. Using hard-drives instead of RAM would severely damage efficiency, so only a few approaches have addressed image indexing in this context [9].

For very large datasets, it is therefore appropriate to consider a global image description, which is much faster and compact. Global descriptions suffer from well-known limitations, in particular they are not invariant to significant transformations (crops,...). However, for some applications, such as copy detection, most of the illegal copies are very similar to the original: they have only suffered from compression, scaling or limited cropping.

In this paper, we compare the global GIST descriptor with the BOF image representations in different application scenarios. To our knowledge, these descriptions have not been compared in a similar setup. Clearly, one would not expect a global descriptor to outperform BOF representations. One of the problems of GIST description being the fixed spatial layout, we evaluate the impact on the accuracy resulting from this fixed spatial image partitioning. Finally, we propose an indexing strategy for GIST that improves the efficiency without significantly penalizing search accuracy. The advantage over the binary codes proposed in [23] is that only a small fraction of the database has to be visited. The idea is to first apply the Hamming Embedding technique proposed in [5] to the GIST descriptor. This selects most of the potentially correct images. Then we apply filtering and re-ranking steps to further improve the quality of the ranking.

The paper is organized as follows. Section 2 introduces the different image representations evaluated in this paper. Our efficient image search system is introduced in Section 3. The datasets representing the application cases and the evaluation protocol are introduced in Section 4. Finally, Section 5 provides experimental results comparing the performance of GIST descriptors and of the BOF representation.

2 Image representation

In this section, we briefly present the image descriptions that are compared. Each method is represented by an acronym given in the subsection title.

2.1 GIST global description GIST

To compute the color GIST description the image is segmented by a 4 by 4 grid for which orientation histograms are extracted. Our implementation[1] takes as input a square image of fixed size and produces a vector of dimension 960. Most of the works using the GIST descriptor [10, 4] resize the image in a preliminary stage, producing a small square image. Its width typically ranges from 32 to 128 pixels. This is sufficient due to the low dimensionality of the descriptor, i.e., it does not represent the details of an image. We choose a size of 32×32 pixels. The images are rescaled to that size irrespective of their aspect ratio. GISTs are compared using the L2 distance. In the following, image search consisting in exhaustively computing the L2 distances between the GIST representation of a query and of a set of GIST descriptors is simply called "GIST".

2.2 Bag-of-features representation BOF

The BOF framework [20] is based on local invariant descriptors [14, 11] extracted at invariant regions of interest. It matches small parts of images and can cope with global transformations, such as scaling, local changes in illumination or combined transformations.

The feature extraction is performed in two steps: detecting regions of interest with the Hessian-Affine detector [14], and computing SIFT descriptors for these regions [11]. These steps are performed using the software available at [13].

The fingerprint of an image is obtained by quantizing the local descriptors using a nearest-neighbor quantizer. The image is then represented by a histogram of visual word occurrences, which is normalized, here with the L2 norm.

The visual vocabulary of the quantizer is produced using k-means. It contains a large number of visual words (in this paper, $k = 200,000$, as in [5]). Therefore, the fingerprint histograms are very sparse, making queries in the inverted file efficient.

2.3 Hamming embedding HE

We also compare GIST with the state-of-the-art image indexing system of [5]. This technique was successfully used [3] in the video copy detection task of the TRECVID'08 evaluation campaign [21]. This work, shows that a richer representation of the images is obtained by adding a short signature that refines the representation of each descriptor within its quantizer cell. The signature is obtained by a function locally mapping the Euclidean space associated with a particular Voronoi cell to the space of binary sequences, or Hamming space (hence the name "Hamming embedding"). More precisely, a descriptor x is represented by a tuple $(q(x), b(x))$, where $q(x)$ is the visual word resulting from the

[1] A re-implementation in C of the Matlab code available on Antonio Torralba's web page.

k-means quantizer $q(.)$, and $b(.)$ is the HE mapping function. Two descriptors are assumed to match if

$$\begin{cases} q(x) = q(y) \\ h\left(b(x), b(y)\right) \leq h_t \end{cases}, \tag{1}$$

where $h(b_0, b_1)$ is the Hamming distance between binary vectors b_0 and b_1, and h_t is a fixed threshold. The binary vectors are typically of length 64 and $h_t = 24$. The score of an image is first obtained as the weighted sum [6] of the distances of the matches satisfying (1), then normalized.

2.4 BOF with a spatial grid GBOF/GHE

One of the drawbacks of GIST compared to BOF is the spatial grid partitioning the images. This arbitrary segmentation does not allow the recognition of images that have suffered strong cropping, or images of the same objects shot from different viewpoints, etc. In order to evaluate the impact of this segmentation on the search quality, we define a BOF representation that partitions the images as in GIST, i.e., using a 4×4 regular spatial grid (see Fig. 1). Only interest points from the same grid cell may be matched.

Note that using image partitioning with BOF is not new and may be useful for some applications, such as object class recognition [8]. For object/location recognition and copy detection, such a partitioning is expected to decrease the accuracy. Besides, using a grid improves the efficiency, as the grid artificially creates a larger visual vocabulary: the lists in the inverted file are indexed by tuples of the form $(q(x), r(x))$ (instead of only $q(x)$ in standard BOF), where $r(x)$ is the region associated with descriptor x. For a 4×4 grid and a $200,000$-visual word vocabulary, the resulting number of lists in the inverted file becomes $3,200,000$. This increased size of the vocabulary (by a factor 16) results in a reduction of the average length of the lists and of the number of entries to be visited during a query. This spatial grid can be used with standard BOF (then denoted GBOF), or with the Hamming Embedding described in section 2.3 (GHE).

2.5 Spatial verification SV

We also provide results obtained with a full spatial geometrical verification (SV) between images. We use the same method as in [11], i.e., we estimate the affine 2D transformation in two steps. First, a Hough scheme estimates a similarity transformation with 4 degrees of freedom. Each pair of matching regions generates a set of parameters that "votes" in a 4D histogram. In the second step, the sets of matches from the largest bins are used to estimate a finer 2D affine transformation. Images are similar if such a transformation can be estimated and results in a number of inliers, i.e., matching points which are consistent with the transformation.

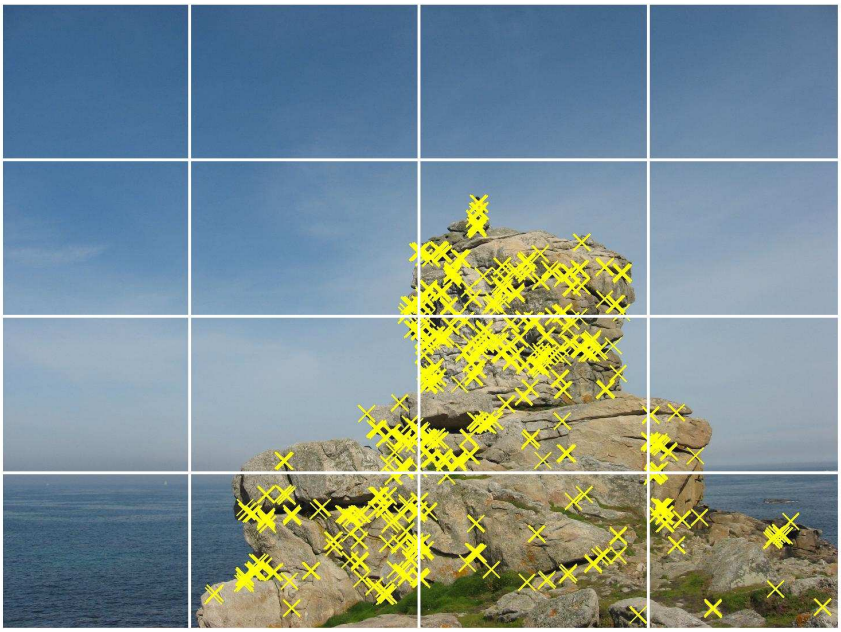


Figure 1: BOF with grid. The regions of interest, here localized by the crosses, are represented by both their visual words (texture) and the grid region where they are localized.

3 Image indexing strategy

In this section, we propose an image search system that handles image collections on a very large scale. Our scheme, illustrated in Fig. 2, operates in three steps that rank and filter the images. The role of each step is to filter most of the images so that the next and more accurate image ranking method becomes tractable.

3.1 GIST indexing structure GISTIS

The first filtering step of our image search system, (Fig. 2, left) relies on an efficient GIST indexing structure: GISTIS. It is derived from [5], which employs an inverted file [25]. In this structure, a quantized index identifies the descriptor lists that are likely to contain the nearest descriptors. GISTIS is constructed as follows (notations are the same as in section 2.3):

- A k-means algorithm is run on a set of GIST descriptors, to produce a codebook $\{c_1, \ldots, c_k\}$ of k centroids. Here $k = 20,000$. This clustering is performed only once on an independent image dataset, i.e., not using the GIST image descriptors to be indexed.

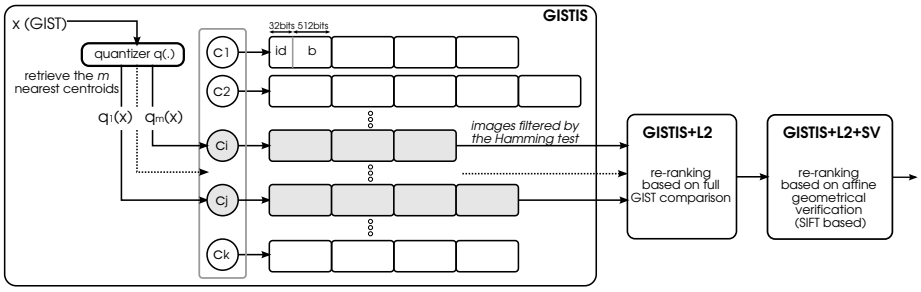


Figure 2: Overview of the image search system. *Left to right:* filtering based on GIST using GISTIS, filtering based on GIST with exact distance computation, final filtering based on SIFT-based geometrical verification.

- Each database descriptor x is assigned to the nearest centroid $q(x)$ of the quantizer codebook.
- A binary signature is computed using Hamming Embedding, i.e., the signature generation procedure of [5]. Because of the high dimensionality of GIST descriptors, the length of the signature is set to 512 bits.
- The image identifier and its signature $b(x)$ are stored in an entry of GISTIS. Like in section 2.3, the entries are indexed by $q(x)$.

Retrieving similar signatures from the structure is performed by

- finding the m nearest centroids of the query descriptor x, producing quantized indexes $q_1(x), \ldots, q_m(x)$. We set $m = 200$ for the experimental results. Note that the fingerprint generation is not symmetric, as on the query side we produce several quantized indexes. Multiple assignment to centroids on the database side would use too much memory.
- computing binary signatures $b_1(x), \ldots, b_m(x)$ associated with descriptor x for all the quantized indexes.
- visiting the entries associated with the quantized indexes. Images which binary signatures are below a pre-defined threshold $h_t = 220$ of the query signature are kept.

Memory usage: an image is represented by 68 bytes using our structure: 4 bytes for the image identifier and 64 bytes for the binary signature. Note that the quantized index is implicitly encoded by the list where the considered image entry is stored.

The structure filters most of the images. Only 200 inverted lists are visited out of $20,000$. As a first approximation, we can expect to visit $200/20000 = 1\%$ of the image entries. It turns out that, because the inverted lists are not

balanced, on average about 2.3% of the entries are visited. Second, the Hamming distance test further filters 94% of the remaining images (for a threshold $h_t = 220$). Finally, the structure typically returns 0.13% of the images (only 0.01% for $h_t = 200$), which are ranked according the Hamming distance between binary signatures.

3.2 Two-stage re-ranking

The indexing structure proposed above dramatically reduces the number of images assumed to be relevant. For a billion image dataset, it would typically return one million images. Having filtered the majority of the images using GISTIS, we can now apply more precise image comparison methods. Here, we propose two strategies that are used either independently or jointly to refine the ranking provided by our efficient structure.

- **GISTIS+L2**: The first consists in re-ranking the images based on the comparison of the full GIST descriptors. producing a list of images ranked according to the Euclidean distance.
- **GISTIS+L2+SV:** In addition to the two pre-filtering stages, a full geometrical verification is performed to keep only the images that are consistent in terms of an affine transformation. For this purpose, we use the spatial verification described in the subsection 2.5. This scheme is illustrated by Fig. 2.

For both these re-rankings steps, the image representation (i.e., either full GIST or the set of local descriptors used in SV) is read from a mechanical hard drive. That is why we only re-rank the top 200 images returned by GISTIS. Using solid-state drives would certainly increase the efficiency, as the bottleneck in our case is the disk latency.

4 Datasets and measures

This section introduces the datasets used in our experiments, as well as the measures of accuracy used to evaluate the different methods.

4.1 Datasets

We have used two evaluation datasets, namely the INRIA Holidays dataset and the INRIA Copydays dataset[2]. In addition, we have used a set of "distracting images", referred to as *distractors*. These images are merged with those of the evaluation datasets to evaluate the impact of the large scale on complexity and accuracy.

Holidays: object/location recognition. This dataset is mainly composed of personal holiday photos. The remaining ones were taken on purpose to test

[2]Both datasets are available at `http://lear.inrialpes.fr/people/jegou/data.php`

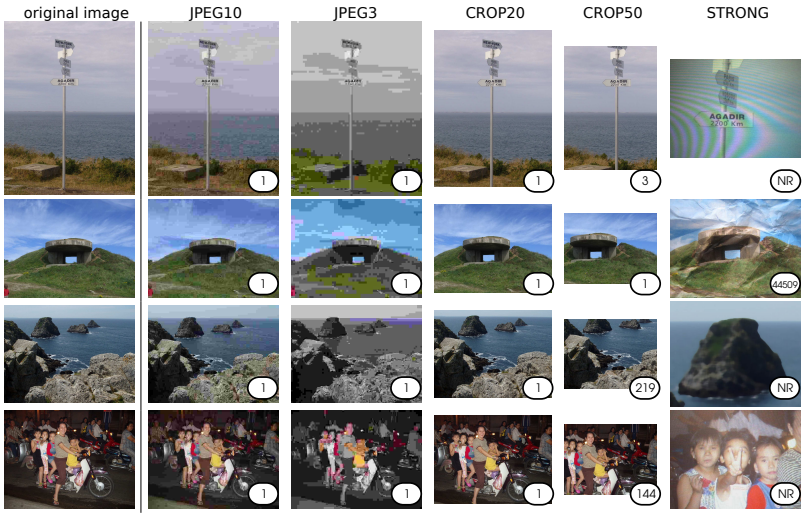


Figure 3: Sample images from *INRIA Copydays* and corresponding transformed images. The number is the rank of the original image when submitting the attacked image to a database of 110 million images using the GISTIS method introduced in Section 3. NR means that the image is not returned by GISTIS.

the robustness to various transformations: rotations, viewpoint and illumination changes, blurring, etc. The dataset includes a large variety of scene types (natural, man-made, water and fire effects, etc) and images are of high resolution. The dataset contains 1491 images partitioned into 500 groups, each of which represents a distinct scene, location or object. The first image of each group is the query image and the correct retrieval results are the other images of the group. Because of the significant variations in viewpoint, one would expect global description to perform poorly on this dataset.

Copydays: near-duplicate detection. We have created this dataset to evaluate our system for images that have been synthetically attacked. The dataset contains 157 original images. To represent typical transformations performed on images in a copy detection application, each image of the dataset has been transformed with three kinds of transformations:

- image resizing (by a factor of 4 in dimension = 16 in surface), followed by JPEG compression ranging from JPEG3 (very low quality) to JPEG75 (typical web quality).
- cropping ranging from 5% to 80% of the image surface.
- strong transformations: print and scan, paint, change in contrast, perspective effect, blur, very strong crop, etc. We have produced 229 transformed images using such strong transformations.

The transformations are illustrated in Fig. 3. The first two types of transformations are easy, but of practical interest: most copyright violations occur with these transformations, producing near-duplicate images. The transformations from the last category strongly degrade the images, compromising their commercial value. For this reason, this class of transformations can be seen as the worst case of copy detection.

Distractors. We have retrieved 32.7 million high resolution images from Flickr. The subsample of one million images built for [5], Flickr1M, is used for most experiments. In addition to these images, we use the "tiny image" dataset of Torralba and Fergus [22]. This dataset contains 79 million images of size 32×32. Due to their small size, they have only been used to evaluate the behavior of GIST and of GISTIS on a large scale. Note, however, that this size is consistent with the pre-processing we apply to all images when computing GIST, see subsection 2.1.

4.2 Evaluation protocol

In order to evaluate the methods described in section 2 and the image search system presented in section 3, we have used two standard evaluation measures, namely the mean average precision (mAP) and the recall at particular ranks.
mAP. For each query image we obtain a precision/recall curve, and compute its average precision (the area under the curve). The mAP is then the mean for a set of queries [17].

recall@R. Measuring the recall at a particular rank R, i.e., the ratio of relevant images ranked in top R positions, is a very good measure of the filtering capability of an image search system. For a system involving several filtering steps, such as ours or the one proposed in [18], curves for varying value of R allow to optimize short-list sizes.

5 Experiments

In this section, we evaluate the different image matching methods introduced in Section 2 and the efficient image search system introduced in Section 3. For local description, we only report results for up to one million images, due to the large storage volume of local descriptors.

5.1 Complexity analysis

Table 1 summarizes the memory usage and the query times we measured by making 500 queries on the Flickr1M dataset. As a general observation, more accurate representations yield higher memory usages and query times.

Memory usage. The spatial verification SV typically requires 500KB per image, which does not fit into memory. Considering a powerful machine with 64GB of main memory, the BOF approach and the method of [5] can typically index

method	bytes (RAM) per image	time per query image *fingerprint*	*search*
SV [11]	501,816	440 ms	13 h
HE [5]	35,844	780 ms	96 ms
BOF	11,948	775 ms	353 ms
GHE	35,844	780 ms	47 ms
GBOF	11,948	775 ms	67 ms
GIST	3840	35 ms	1.26 s
GISTIS	68	36 ms	2 ms
GISTIS+L2	68	36 ms	6/192 ms

Table 1: Memory usage and computing speed for the various methods on a 64-bit 8-core computer. Timings were measured on the Flick1M dataset (1 million images) by performing 500 queries, except for SV, where the timing is extrapolated from queries in the Holidays dataset (1491 images). The two query times for GISTIS+L2 are for full GISTs stored in main memory and on disk, respectively.

5 and 1.5 million images respectively, a bit more if using less local descriptors. GISTIS allows the storage of about 850 million images, a large improvement over the 15 million raw GIST descriptors that can be stored in memory. GISTIS is therefore able to store two orders of magnitude more images than local description-based approaches.

Efficiency. One can see in Table 1 that global GIST descriptors are one to two orders of magnitude more efficient than approaches based on local description. The time measurements are given separately for 1) the fingerprint extraction, which does not depend on the dataset size, and 2) search in the indexing structure. Note that in all the schemes we consider, the query time of the structure is linear in the number of images. The fingerprint extraction time is slightly higher for GISTIS compared with GIST, as the descriptor quantization and the computation of the corresponding binary signature take about 1 ms. Here again, the results are appealing: the schemes based on global description are at least one order of magnitude more efficient that those using local descriptors.

The re-ranking methods would highly benefit from the recent solid-state drives, which are not penalized by the latency of disks (here, the limiting factor is the storage's seek time, not its throughput). Depending on whether GISTs are read from memory or from a hard disk, the query time of GIST+L2 is of 6 ms and 192 ms, respectively. As expected (see subsection 2.4), the GBOF and GHE variants are faster than the corresponding BOF and HE methods.

Complexity on a large scale. The average query time per image measured on our largest dataset of 110 million images is 0.18 s (36 ms for the fingerprint and

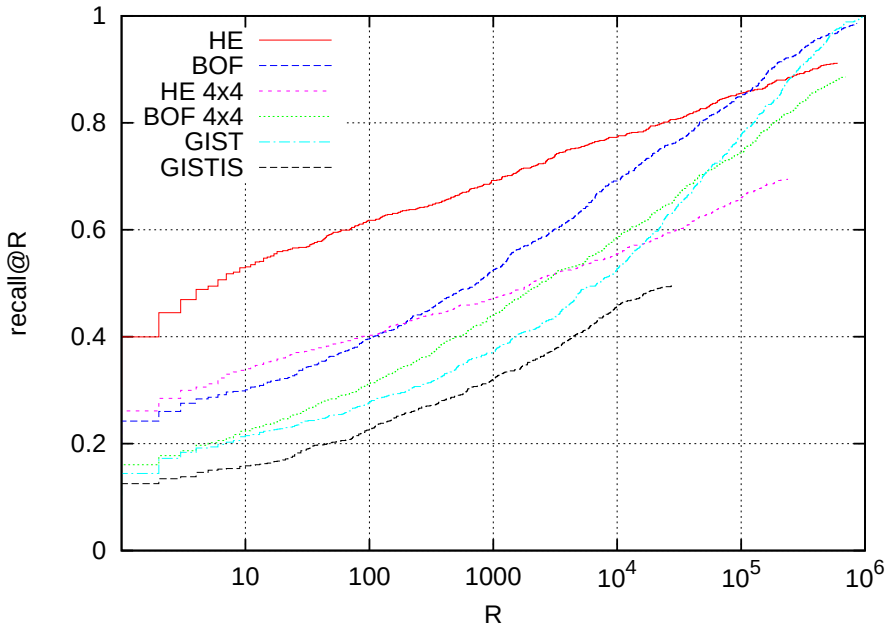


Figure 4: Holidays+Flickr1M: rate of relevant images found in the top R images.

143 ms for the search using GISTIS). This is a large improvement over the result given in [23], where a query time of 0.75 s is reported for querying a dataset of 12.9 million images. The memory usage is also very reasonable: about 7GB to index all the images. To our knowledge, this is the best result reported on that scale on a single computer.

5.2 Search quality

Object/location recognition. Fig. 4 shows how the different systems rank the relevant images in a scene or object recognition setup. One can observe the better accuracy obtained by local approaches. Fig. 6(a) shows the mAP measure as a function of the number of distractors. Our GISTIS structure only slightly reduces the accuracy compared with the GIST exhaustive search, so given its much higher efficiency, it is worth using it in this context. Moreover, by re-ranking the images (GISTIS+L2), the accuracy is nearly the same as for the GIST exhaustive search, which shows the relevance of the two-stage algorithm in this context.

Near-duplicate detection. The tolerance of the methods to near-duplicate attacks is illustrated in Fig. 5, which gives the mAP values for the Copydays dataset merged with the one million images of Flickr1M. One can first observe the excellent behavior of GIST descriptors for the SCALE+JPEG attack, which

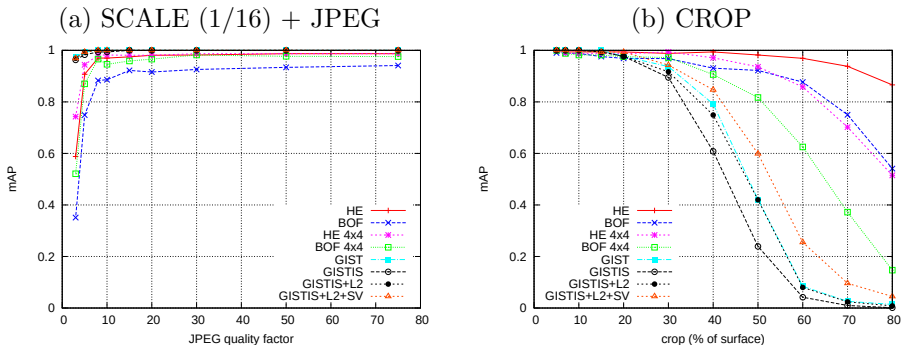


Figure 5: Attacks on Copydays + Flickr1M.

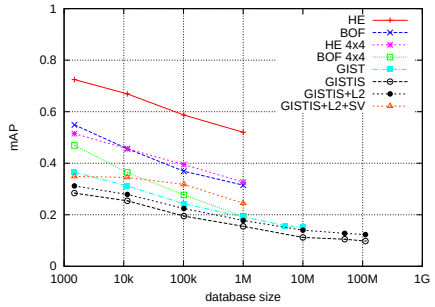
is better than local descriptors for all quality factors. The conclusion is reversed for the severe image cropping transformation. However, there is no clear advantage of using local descriptors for moderate crops, i.e., if less than 20% of the image surface is removed.

The GISTIS structure slightly reduces the quality compared with exhaustive GIST search. However, by re-ranking the results (GISTIS+L2) the performance we obtain is similar to that of exhaustive GIST distance computation. Re-ranking based on spatial verification (GISTIS+L2+SV) further improves the results, providing better results than those of GIST. This is because the SV technique is complementary with global descriptors.

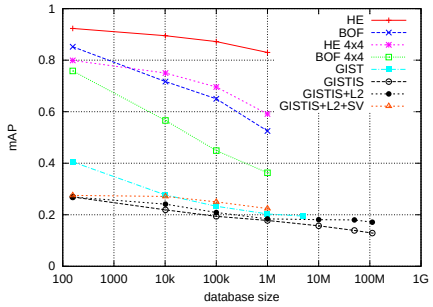
Impact of the spatial segmentation. Fig 4 shows that the GBOF approach is only slightly better than GIST, which reflects the penalization resulting from the spatial grid in the context of object and scene recognition. Interestingly, in Fig. 5, depending on the transformation, the spatial grid used in GBOF and GHE has different effects on the mAP accuracy values. The accuracy decreases for the cropping, but this segmentation improves the results for the SCALE+JPEG transformation.

Search quality on a large scale. Fig. 6 shows the mAP values we obtained by increasing the number of images up to 110 millions. One can observe that for the strong attacks (Holidays and Copydays-STRONG) depicted in Fig. 6(a) and Fig. 6(b), it is worth using local descriptors if this choice is tractable. Besides, for usual transformations such as scaling and compression, see Fig. 6(c), or limited cropping, see Fig. 6(d), the results are comparable with those of the state-of-the-art approach of [5]. On 110 million images, the ranking obtained by GISTIS is perfect without re-ranking even for a JPEG quality factor of 15. Fig. 3 shows typical results with their rank when querying the largest dataset.

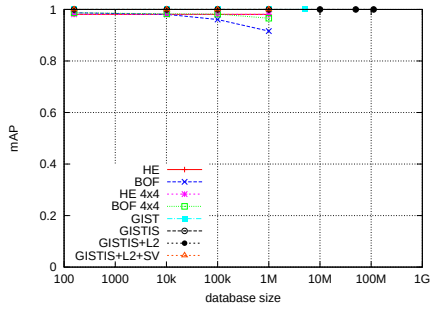
(a) Holidays: scene or object recognition



(b) Copydays: STRONG



(c) Copydays: SCALE (1/16) + JPEG20



(d) Copydays: CROP20

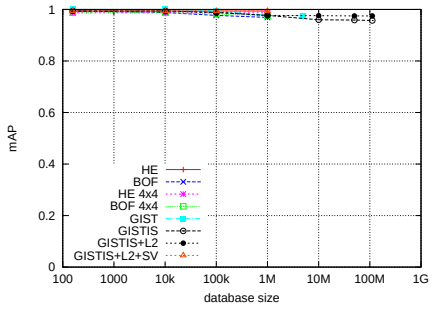


Figure 6: Accuracy (mAP) as a function of the number of database images (up to 110 million images).

6 Conclusions

We have evaluated the GIST descriptor for two different applications and compared it to state-of-the-art methods based on local descriptors. Local representations obtain significantly better results for object and location recognition. However, the global GIST descriptor is shown to find part of the relevant images even in large datasets.

For near-duplicate detection the GIST descriptor provides very high accuracy, in some cases outperforming the state-of-the-art local approaches, namely for transformations such as scaling, JPEG compression and limited cropping. Overall, the results obtained with GIST are compelling given its much higher efficiency and smaller memory usage, allowing to scale up to very large datasets.

We have also introduced an efficient indexing strategy for the GIST descriptor. It provides results similar to those of exhaustive search, while providing very high efficiency.

7 Acknowledgments

We would like to thank Rob Fergus, Antonio Torralba and William Freeman for kindly providing the entire tiny image dataset of 80 million images. This work was supported by the French multimedia engine project QUAERO and the ANR project GAIA.

References

[1] 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 *ICCV*, 2007.

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

[3] M. Douze, A. Gaidon, H. Jégou, M. Marszałek, and C. Schmid. INRIA-LEAR's video copy detection system. In *TRECVID Workshop*, November 2008.

[4] J. Hayes and A. Efros. Scene completion using millions of photographs. In *SIGGRAPH*, 2007.

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

[6] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometry consistency for large scale image search - extended version. Technical report, INRIA, RR 6709, October 2008.

[7] H. Jégou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *CVPR*, 2007.

[8] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[9] H. Lejsek, F. Ásmundsson, B. Jónsson, and L. Amsaleg. Scalability of local image descriptors: a comparative study. In *ACM Multimedia*, pages 589–598, 2006.

[10] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV*, October 2008.

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

[12] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, pages 384–393, 2002.

[13] K. Mikolajczyk. Binaries for affine covariant region descriptors, 2007.

[14] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[15] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006.

[16] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

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

[18] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[19] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[20] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, 2003.

[21] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, 2006.

[22] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large database for non-parametric object and scene recognition. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, November 2008.

[23] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large databases for recognition. In *CVPR*, 2008.

[24] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, 2009.

[25] J. Zobel, A. Moffat, and K. Ramamohanarao. Inverted files versus signature files for text indexing. *ACM Transactions on Database Systems*, 23(4):453–490, 1998.