Aggregating Local Image Descriptors into Compact Codes

Hervé Jégou, *Member*, *IEEE*, Florent Perronnin, *Member*, *IEEE*, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid, *Fellow*, *IEEE*

Abstract—This paper addresses the problem of large-scale image search. Three constraints have to be taken into account: search accuracy, efficiency, and memory usage. We first present and evaluate different ways of aggregating local image descriptors into a vector and show that the Fisher kernel achieves better performance than the reference bag-of-visual words approach for any given vector dimension. We then jointly optimize dimensionality reduction and indexing in order to obtain a precise vector comparison as well as a compact representation. The evaluation shows that the image representation can be reduced to a few dozen bytes while preserving high accuracy. Searching a 100 million image data set takes about 250 ms on one processor core.

Index Terms—Image search, image retrieval, indexing.

1 Introduction

This paper proposes an image indexing scheme for very large databases. Image search consists here of finding in a large set of images the ones that represent the same object or location possibly viewed from different viewpoints and in the presence of occlusions and clutter. Typical applications include finding locations [1] and particular objects [2], as well as detecting partial image duplicates on the web [3] and deformed copies [4]. We do not consider specific visual recognition tasks, such as face detection and recognition, which require the use of specific descriptors and detectors. We also do not consider semantic queries such as "retrieve all images containing cars," which are usually addressed by supervised classification techniques such as those evaluated in the PASCAL visual object recognition competition [5].

The bag-of-words (BOW) representation is the most widely adopted method for this purpose [6]. There are three reasons for the success of the BOW representation. First, this representation benefits from powerful local descriptors, such as the SIFT descriptor [7] or more recent ones [8], [9], [10]. Second, the BOW vector representation can be compared with standard distances. Third, the BOW vector can be high dimensional [2], [1], in which case the vectors are sparse and inverted lists can be used to implement efficient search [6]. However, two factors limit the number of images that can be indexed in practice: the search time itself, which becomes prohibitive when considering more than 10 million images, and the memory usage per image.

In this paper, we address the problem of searching for the most similar images to a given query in a very large image database (a hundred million images). We put an emphasis on the joint optimization of three constraints: the search accuracy, its efficiency, and the memory usage. The last two are related [11], as search efficiency can be approximated by the amount of memory visited during the search. Most similar to our work is the approach of Jégou et al. [12], which proposes an approximate nearest neighbor search for BOW features. However, this method is limited to small vocabularies and thus obtains a relatively low search accuracy. Moreover, an image needs to be encoded by more than a hundred bytes to approximate the search accuracy of the initial representation.

The efficiency problem was partially addressed by the min-Hash approach [13], [14] and the prefiltering method of Torresani et al. [15]. However, these techniques still require a significant amount of memory per image and are more relevant in the context of near-duplicate image detection as their search accuracy is significantly lower than BOW. Some authors address the efficiency and memory constraints by using GIST descriptors [16] and by converting them to compact binary vectors [17], [11], [18]. These approaches suffer from the limited robustness of the global GIST descriptor to geometric transformations [19], and none jointly fulfills the three aforementioned constraints: [11], [18] require an exhaustive search, while [17] is memory consuming due to the redundancy of the LSH algorithm. In contrast, our approach obtains a higher accuracy with a very compact representation, i.e., codes of 20 bytes. This performance is achieved by optimizing:

1. the representation, i.e., how to aggregate local image descriptors into a vector representation,
2. the dimensionality reduction of these vectors,
3. the indexing algorithm.

- *H. Jégou is with INRIA, Campus de Beaulieu, 263 avenue du Général Leclerc, 35042 Rennes, France. E-mail: herve.jegou@inria.fr.*
- *F. Perronnin is with Xerox, 6 chemin de Maupertuis, 38240 Meylan, France. E-mail: florent.perronnin@xrce.xerox.com.*
- *M. Douze and C. Schmid are with INRIA, 655 avenue de l 'Europe, 38334 Saint Ismier Cedex, France.*
  E-mail: {matthijs.douze, cordelia.schmid}@inria.fr.
- *J. Sánchez is with the Research Centre in Informatics for Engineering, Maestro López esq Cruz Roja Argentina, UTN, FRC, Ed. Ing., Salcedo, Córdoba X5016ZAA, Argentina. E-mail: jsanchez@scdt.frc.utn.edu.ar.*
- *P. Pérez is with Technicolor, 1, avenue de la Belle Fontaine, 35576 Cesson Sevigne, France. E-mail: Patrick.Perez@technicolor.com.*

Manuscript received 3 Dec. 2010; revised 12 July 2011; accepted 7 Oct. 2011; published online 7 Dec. 2011.
Recommended for acceptance by T. Darrell, D. Hogg, and D. Jacobs.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMISI-2010-12-0923.
Digital Object Identifier no. 10.1109/TPAMI.2011.235.

0162-8828/12/$31.00 © 2012 IEEE

These steps are closely dependent: A high-dimensional image representation usually provides better exhaustive search results than a low-dimensional one. However, high-dimensional vectors are more difficult to index efficiently. A low-dimensional vector can be indexed efficiently, but its discriminative power may be too low.

Our first contribution is an image representation of moderate dimensionality that provides an excellent search accuracy. Our representation is based on Fisher kernels [20], a powerful tool to transform an incoming variable-size set of independent samples into a fixed-size vector representation. We assume that the samples are distributed according to a parametric generative model, in this case a Gaussian Mixture Model (GMM) estimated on a training set. Perronnin and Dance [21] applied Fisher kernels in the context of image classification and large scale image search [22]. Experimental results demonstrate that this choice significantly outperforms BOW for the same size on most data sets. Moreover, it is cheaper to compute because fewer visual words are required. We also evaluate a simplified version of Fishers kernels, the Vector of Locally Aggregated Descriptors (VLAD) [23].

The description vectors are compared using the L2 distance. Their dimensionality is reduced to a few hundred components by principal component analysis (PCA). To index the resulting vectors, we show the advantage of jointly optimizing the tradeoff between the dimensionality reduction and the indexing algorithm. We use the recent approximate nearest neighbor search method of Jégou et al. [24], which performs similarity search with a technique derived from source coding, i.e., the encoded vectors can be approximately reconstructed. As a result, the error induced by PCA can be directly compared with the error resulting from the vector compression.

The paper is organized as follows: Section 2 introduces the data sets used to evaluate the components of our approach. The different strategies used to aggregate the descriptors, BOW, the Fisher Kernel, and the VLAD representations, are presented in Section 3 and compared in Section 4. We introduce the subsequent indexing stage in Section 5, where a joint optimization of dimensionality reduction and indexing is proposed. Experimental results demonstrate the performance of our approach in Section 6: The performance of BOW is obtained with an image representation of about 20 bytes. In terms of memory usage, search accuracy, and efficiency, this is a significant improvement over the state of the art [12], as well as over our two prior conference publications [22], [23]. This paper is built upon these two prior publications and includes the following additions: 1) an analysis of the dimensionality reduction for local descriptors as well as for the final aggregated representation, 2) a theoretical and experimental comparison of the Fisher Kernel [21] and the VLAD representation [23] which allows improving our image search system, and 3) additional experiments, in particular on a data set with 100 million images.

2 Data Sets and Evaluation Protocol

To extract local features, we have used the experimental setup of Jégou et al. [12] and the feature extraction software available online.[1] The regions of interest are extracted using the Hessian rotation and affine-invariant region detector [25] and described by the SIFT descriptor [7]. We have used an independent image set for all the learning stages. The evaluation is performed on the data sets described below, and conducted by submitting each query image of the data set to the evaluated system, which returns a list of images sorted by decreasing relevance.

The **INRIA Holidays** data set [26] is a collection of 1,491 holiday images, 500 of them being used as queries. The accuracy is measured by the mean Average Precision (mAP), as defined in [1].

The **University of Kentucky Recognition Benchmark** (UKB [2]) contains images of 2,550 objects, each of which is represented by four images. The most common evaluation metric for this data set counts the average number of relevant images (including the query itself) that are ranked in the first four positions when searching the 10,200 images. This corresponds to 4 times the recall@4 measure, i.e., the best performance is 4.

The **Oxford5K building** data set [1] contains 5,062 photos gathered from Flickr with tags related to buildings in Oxford. Each of the 55 queries is defined by a rectangular region delimiting a building on an image. The correct results for a query are images of this building. The accuracy is measured by mAP.

The **Flickr10M** data set allows us to evaluate the performance of our method on a large scale with high-quality pictures. It was obtained by downloading 10 million random images from Flickr. This image set is merged with the other data sets to evaluate the accuracy and efficiency on a large scale [12].

The **Exalead100M** data set is used for large scale evaluation on up to 100M images and used in conjunction with the Copydays data set described below. The images have been extracted from crawled webpages by Exalead, and were downscaled uniformly to 150 pixels in their larger dimension and compressed with a JPEG quality of 75.

INRIA Copydays was designed to evaluate near-duplicate detection [19]. The data set contains 157 original images. To obtain query images relevant in a copy detection scenario, each image of the data set has been transformed with three different types of transformation:

- 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). For each compression factor, 157 images are generated (one per database image).
- Cropping ranging from 5 to 80 percent of the image surface. Here, we use only the queries with the cropping parameter fixed to 50 percent (one per database image, i.e., 157 query images in total).
- Strong transformations: print and scan, occlusion, change in contrast, perspective effect, blur, very strong cropping, etc. There are, in total, 229 transformed images, each of which has only a single matching image in the database.

The accuracy is measured by mAP. We merge the database with 10,000 distractor images from Flickr10M for most experiments. It is merged with images from Exalead100M for our very large scale evaluation. In this case, in order to

[1] http://lear.inrialpes.fr/people/jegou/data.php.

obtain an image quality consistent with that of Exalead100M, the same down-scaling factor and JPEG compression are applied to the Copydays images.

For experiments on large image sets, we will use the recall@R metric, i.e., the rate of relevant images that are ranked in the top R positions. This measure is relevant for two reasons. First, evaluating recall@R with respect to R indicates the empirical distribution of the rank of the true positives. Second, if the rank R is fixed to a small value (e.g., $R = 100$), this indicates if a method provides a relevant shortlist for a subsequent precise (and more costly) direct image comparison based on geometric verification [1], [7].

3 Image Vector Representation

In this section, we introduce the different vector aggregation methods compared in this paper, namely, the BOW representation, the Fisher Vector (FV) representation, and the VLAD vectors, which are shown to be an extremal case of FV. All these methods aim at aggregating d-dimensional local descriptors into a single vector representation.

3.1 Bag-of-Features

The BOW representation groups together local descriptors. It requires the definition of a codebook of K centroids ("visual words") usually obtained by k-means clustering. Each local descriptor of dimension d from an image is assigned to the closest centroid. The BOW representation is the histogram of the number of image descriptors assigned to each visual word. Therefore, it produces a K-dimensional vector, which is subsequently normalized. There are several variations on how to normalize the histogram. When seen as an empirical distribution, the BOW vector is normalized using the Manhattan distance. Another common choice consists of using euclidean normalization. The vector components are then weighted by *idf* (inverse document frequency) terms [2], [6]. In the following, we perform L_2 normalization of histograms and use the *idf* calculation of Sivic and Zisserman [6]. Several variations have been proposed to improve the quality of this representation. One among the most popular [27], [28] consists of using soft quantization techniques instead of a k-means.

3.2 Fisher Vector

By counting the number of occurrences of visual words, BOW encodes the 0-order statistics of the distribution of descriptors. The Fisher vector extends the BOW by encoding high-order statistics (first and, optionally, second order). This description vector is the gradient of the sample's likelihood with respect to the parameters of this distribution, scaled by the inverse square root of the Fisher information matrix. As a result, it gives the direction, in parameter space, into which the learned distribution should be modified to better fit the observed data. In other terms, FV describes how the set of descriptors deviates from an average distribution of descriptors, modeled by a parametric generative model. It has been shown [20] that discriminative classifiers can be learned in this new representation space. In our image search framework, the FV is seen as a method to capture the information conveyed by a set of descriptors into a fixed-length signature.

The Fisher kernel framework. Let $X = \{x_t, t = 1, \dots, T\}$ denote the set of T local descriptors extracted from an image. We assume for now that the generation process of X can be modeled by an image-independent probability density function u_λ with parameters λ.[2] Jaakkola and Haussler [20] proposed to describe X by the vector

$$G_\lambda^X = \frac{1}{T} \nabla_\lambda \log u_\lambda(X). \qquad (1)$$

The gradient of the log likelihood describes the contribution of the parameters to the generation process. Its dimensionality only depends on the number of parameters in λ. A natural kernel on these gradients is the "Fisher kernel" [20]:

$$K(X,Y) = G_\lambda^{X\top} F_\lambda^{-1} G_\lambda^Y, \qquad (2)$$

where F_λ is the *Fisher information matrix* of u_λ:

$$F_\lambda = E_{x\sim u_\lambda}\left[\nabla_\lambda \log u_\lambda(x) \nabla_\lambda \log u_\lambda(x)^\top\right]. \qquad (3)$$

As F_λ^{-1} is symmetric and positive definite, it has a Cholesky decomposition $F_\lambda^{-1} = L_\lambda^\top L_\lambda$. Therefore, the kernel $K(X,Y)$ is advantageously rewritten as a dot product between normalized vectors $\mathcal{G}_\lambda$, obtained as

$$\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X. \qquad (4)$$

We refer to $\mathcal{G}_\lambda^X$ as the *Fisher vector* of X.

The FV image representation. We follow Perronnin and Dance [21] and choose u_λ to be a GMM: $u_\lambda(x) = \sum_{i=1}^K w_i u_i(x)$. We denote $\lambda = \{w_i, \mu_i, \sigma_i, i = 1, \dots, K\}$, where w_i, μ_i, and σ_i are, respectively, the mixture weight, mean vector, and variance matrix (assumed diagonal) of Gaussian u_i. The model u_λ can be understood as a probabilistic visual vocabulary. It is trained offline on a large number of images using Maximum Likelihood (ML) estimation. In this work, we focus on the gradient with respect to the mean. We use the diagonal closed-form approximation of the Fisher information matrix of Perronnin and Dance [21], in which case the normalization of the gradient by $L_\lambda = F_\lambda^{-1/2}$ is simply a whitening of the dimensions. Let $\gamma_t(i)$ be the soft assignment of descriptor x_t to the ith Gaussian:

$$\gamma_t(i) = \frac{w_i u_i(x_t)}{\sum_{j=1}^K w_j u_j(x_t)}. \qquad (5)$$

Let $\mathcal{G}_i^X$ be the d-dimensional gradient with respect to the mean μ_i of Gaussian i. Assuming that the x_ts were generated independently by u_λ, we obtain, after standard mathematical derivations,

$$\mathcal{G}_i^X = \frac{1}{T\sqrt{w_i}} \sum_{t=1}^T \gamma_t(i) \sigma_i^{-1}(x_t - \mu_i). \qquad (6)$$

The final vector $\mathcal{G}_\lambda^X$ is the concatenation of the $\mathcal{G}_i^X$ vectors for $i = 1, \dots, K$ and is therefore Kd-dimensional.[3] In comparison

[2] We abuse the notation to simplify the presentation: λ denotes both the set of parameters of u as well as the estimate of these parameters.

[3] Taking the gradient with respect to the mean and variance leads to vectors of size $2Kd$. We can obtain vectors of the same size by taking the gradient with respect to the mean only on a GMM with $2K$ Gaussians. In a set of preliminary experiments, we observed that for a fixed FV size the two options led to very similar results.

with BOW, d times fewer visual words are required to obtain a signature of the same length. Experimental results show that excellent results can be obtained even with a relatively small number of visual words K: We typically consider values ranging from $K = 16$ to $K = 256$.

FV Normalization. The FV undergoes two normalization steps. First, a *power normalization* is performed by applying the following operator independently on each component:

$$f(z) = \operatorname{sign}(z)|z|^{\alpha}, \tag{7}$$

with $0 \leq \alpha \leq 1$. We empirically observe that this step consistently improves the quality of the representation. Several complementary interpretations justify this transform. First, it reduces the influence of bursty visual elements, which were shown to corrupt the image similarity in [29]. Second, assuming the compound Poisson distribution as a good generative model of FVs, the power normalization can be interpreted as a variance stabilizing transform which corrects the dependence between the variance and the mean. We provide an analysis of this interpretation in the Appendix provided as supplemental material, which can be found in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.235. Variance stabilizing transforms have previously been applied to BOW histograms, e.g., [30] for GMM modeling and in [31], [32] for better separability of linear classifiers.

We found experimentally that the optimal value for α depends on the number of Gaussians K. However, setting $\alpha = 0.5$ consistently leads to near-optimal results on the range of K values we consider.

The power normalized FV is then L2-normalized, which amounts to using the cosine similarity between FVs. It guarantees that if we query a database with one of its images, the first result will be the image itself if the inner product is used as a similarity measure—a desirable property.

Properties of the FV. We now assume that the descriptors $X = \{x_t, t = 1, \dots, T\}$ of a given image are *iid* and follow a distribution p which departs from the generic distribution u_λ. According to the law of large numbers (convergence of the sample average to the expected value when T increases), we have

$$G_\lambda^X = \frac{1}{T}\sum_{t=1}^{T} \nabla_\lambda \log u_\lambda(x_t) \tag{8}$$

$$\approx \nabla_\lambda E_{x\sim p} \log u_\lambda(x) \tag{9}$$

$$= \nabla_\lambda \int_x p(x) \log u_\lambda(x) dx. \tag{10}$$

Now let us assume that we can decompose p into a mixture of two parts: a background image-independent part which follows u_λ and an image-specific part which follows an image-specific distribution q. Let $0 \leq \omega \leq 1$ be the proportion of image-specific information contained in the image:

$$p(x) = \omega q(x) + (1 - \omega) u_\lambda(x). \tag{11}$$

We can rewrite

$$\begin{aligned} G_\lambda^X \approx{} & \omega \nabla_\lambda \int_x q(x) \log u_\lambda(x) dx \\ & + (1-\omega)\nabla_\lambda \int_x u_\lambda(x) \log u_\lambda(x) dx. \end{aligned} \tag{12}$$

If the values of the parameters λ were estimated with an ML process—i.e., to maximize (at least locally and approximately) $E_{x\sim u_\lambda} \log u_\lambda(x)$—then:

$$\nabla_\lambda \int_x u_\lambda(x) \log u_\lambda(x) dx = \nabla_\lambda E_{x\sim u_\lambda} \log u_\lambda(x) \approx 0. \tag{13}$$

Consequently, we have

$$G_\lambda^X \approx \omega \nabla_\lambda \int_x q(x) \log u_\lambda(x) dx = \omega \nabla_\lambda E_{x\sim q} \log u_\lambda(x). \tag{14}$$

This shows that the image-independent information is approximately discarded from the FV signature and that an image is described by what makes it different from other images (on average). Such a decomposition of images into background and image-specific information has also been employed in the BOW approach by Zhang et al. [33]. However, while the decomposition is *explicit* in [33], it is *implicit* in the FV case.

Relationship with BOW. We introduce

$$w_i^X = \frac{1}{T}\sum_{t=1}^{T} \gamma_t(i), \tag{15}$$

$$\mu_i^X = \frac{\sum_{t=1}^{T} \gamma_t(i) x_t}{\sum_{t=1}^{T} \gamma_t(i)}, \tag{16}$$

where w_i^X is the proportion of descriptors of X soft-assigned to visual word i, i.e., this is the soft-BOW representation. The vector μ_i^X is the average of the descriptors of X weighted by their probability of being assigned to Gaussian i, i.e., loosely speaking, the average of the descriptors of X assigned to Gaussian i). We can rewrite the gradient (6) as follows:

$$\mathcal{G}_i^X = \frac{w_i^X}{\sqrt{w_i}} \sigma_i^{-1}\left(\mu_i^X - \mu_i\right). \tag{17}$$

This clearly shows that the FV extends the BOW representation. Not only does the FV count the number of descriptors assigned to each region in the space, it also encodes the approximate location of the descriptors in each region (relative to the mean μ_i and the variance σ_i). The division by $\sqrt{w_i}$ in (17) can be interpreted as a BOW *idf* term: The descriptors which occur frequently are automatically discounted.

3.3 VLAD: Nonprobabilistic Fisher Kernel

In [23], Jégou et al. proposed the VLAD representation. As for the BOW, a codebook $\{\mu_1, \dots, \mu_K\}$ is first learned using k-means. Each local descriptor x_t is associated with its nearest visual word $\mathrm{NN}(x_t)$ in the codebook. For each codeword μ_i, the differences $x_t - \mu_i$ of the vectors x_t assigned to μ_i are accumulated:

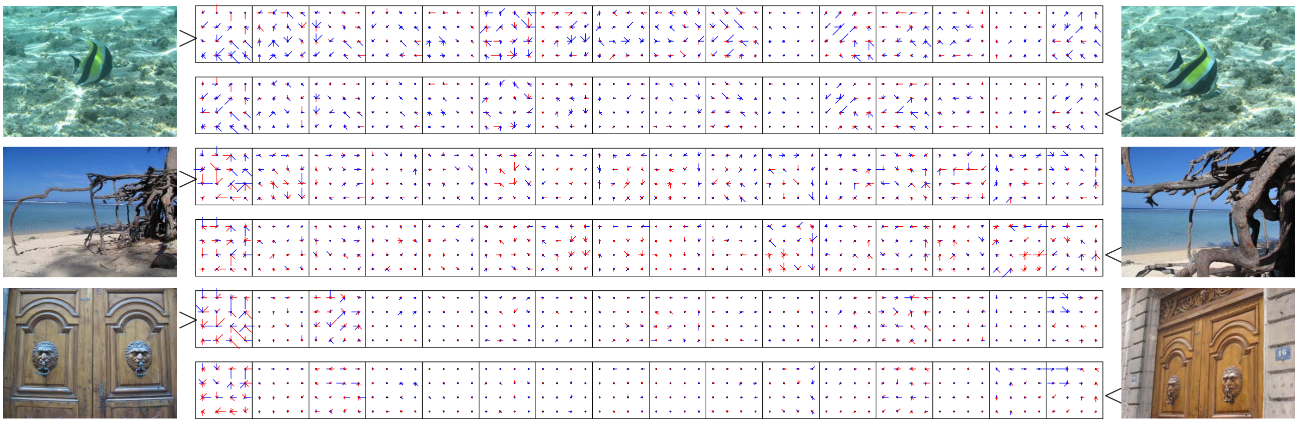

Fig. 1. Images and corresponding VLAD descriptors for $K = 16$ centroids. The components of the descriptor are represented like SIFT, with negative components in red.

$$v_i = \sum_{x_t : \mathrm{NN}(x_t) = i} x_t - \mu_i. \tag{18}$$

The VLAD is the concatenation of the d-dimensional vectors v_i and is therefore Kd-dimensional. Algorithm 1 sums up the resulting algorithm. As for FV, the VLAD can then be power and L2-normalized.

Algorithm 1 Computation of the VLAD descriptor V from a set of descriptors $x_1, \dots, x_T$. The set $\mu_1, \dots, \mu_K$ of centroids is learned on a training set using k-means.

```
For $i = 1, \dots, K$
    $v_i := 0_d$
% accumulate descriptor
For $t = 1, \dots, T$
    $i = \arg\min_j ||x_t - \mu_j||$
    $v_i := v_i + x_t - \mu_i$
$V = [v_1^\top \dots v_K^\top]$
% apply power normalization
For $u = 1, \dots, Kd$
    $V_u := \mathrm{sign}(V_u)\ |V_u|^\alpha$
% apply L2 Normalization
$V := \frac{V}{||V||_2}$
```

We now show that the VLAD is a simplified nonprobabilistic version of the FV: The VLAD is to the FV what k-means is to GMM clustering. The k-means clustering can be viewed as a nonprobabilistic limit case of GMM clustering when:

1. the mixture weights are equal, i.e., $w_i = 1/K, \forall i$,
2. covariance matrices are isotropic, i.e., $\sigma_i = \epsilon \mathbb{I}$ with $\epsilon \geq 0$,
3. the value ϵ converges to zero.

If we enforce hypotheses 1 and 2, (6) simplifies to

$$\mathcal{G}_i^X \propto \sum_{t=1}^{T} \gamma_t(i)(x_t - \mu_i), \tag{19}$$

up to a factor independent of ϵ after normalization.

As $\epsilon \rightarrow 0$, the Gaussians converge to Dirac distributions and hypothesis 3 makes the assignments $\gamma_t(i)$ binary: $\gamma_t(i) = 1$ if $\mathrm{NN}(x_t) = i$ and $\gamma_t(i) = 0$ otherwise. Therefore, as $\epsilon \rightarrow 0$, (19) becomes

$$\mathcal{G}_i^X \propto \sum_{x_t : \mathrm{NN}(x_t) = i} x_t - \mu_i, \tag{20}$$

where we recognize the VLAD representation.

Fig. 1 depicts the VLAD representations associated with a few images when aggregating 128D SIFT descriptors. The components of our descriptor map to components of SIFT descriptors. Therefore, we adopt the usual 4×4 spatial grid with oriented gradients for each v_i, $i = 1, \dots, K$, with $K = 16$. In contrast to SIFT descriptors, a component may be positive or negative, due to the subtraction in (18).

One can observe that the descriptors are relatively sparse (few values have a significant energy) and very structured: Most high descriptor values are located in the same cluster and the geometrical structure of SIFT descriptors is observable. Intuitively, and as shown later, a principal component analysis is likely to capture this structure. For resembling images, the similarity of the descriptors is obvious.

3.4 Dimensionality Reduction on Local Descriptors

Principal component analysis is a standard tool [34] for dimensionality reduction: The eigenvectors associated with the most energetic eigenvalues of the empirical vector covariance matrix are used to define a matrix M mapping a vector $x \in \mathbb{R}^{128}$ to a vector $x' = Mx$ in a lower dimensional space.

We will show in the next section that applying the Fisher Kernel framework directly on local descriptors leads to suboptimal results. Therefore, we apply a PCA on the SIFT descriptors to reduce them from 128 to $d = 64$ dimensions. Two reasons may explain the positive impact of this PCA:

1. Decorrelated data can be fitted more accurately by a GMM with diagonal covariance matrices.
2. The GMM estimation is noisy for the less energetic components.

To confirm that these hypotheses are reasonable, we have performed two experiments on descriptor transformations. First, we applied a random rotation matrix after the PCA that cancels its decorrelating effect. This significantly degrades the search quality. The second transformation is the PCA rotation without dimensionality reduction. This

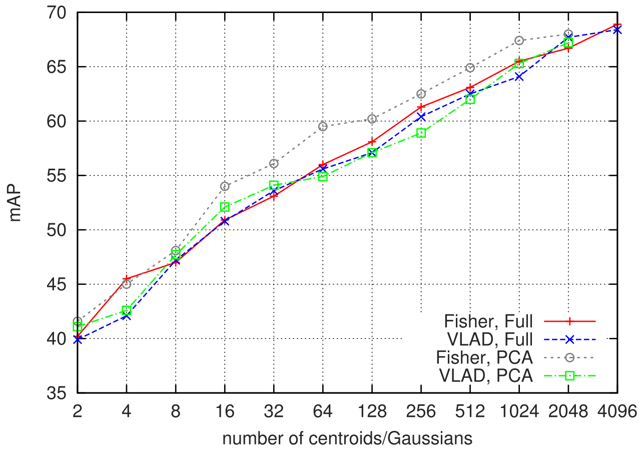

Fig. 2. Comparison of the Fisher and VLAD representations with and without reduction of the local descriptors with PCA (mAP on Holidays).

also degrades the retrieval performance, which means that the least energetic components are detrimental.

4 Evaluation of the Aggregation Methods

In this section, we evaluate and compare the different local aggregation methods described in Section 3, BOW, FV, and VLAD, as well as the global GIST descriptor [16]. We analyze the impact of the number K of centroids/mixture components and study the impact of dimensionality reduction at two stages of the algorithms:

- the local SIFT descriptors are reduced from 128 to 64 components using PCA (see Section 3.4);
- the final VLAD, Fisher, and BOW vectors are reduced from $D = Kd$ to D' components using PCA.

In both cases, the PCA rotation matrices are learned on an independent image data set. Note that in this section the evaluation is performed without the subsequent indexing stage.

Dimensionality reduction of local descriptors. Fig. 2 compares the FV and VLAD representations on the Holidays data set as a function of K. The respective results are similar if these representations are learned and computed on the plain SIFT descriptors, as observed in [23]. However, applying PCA to local descriptors (reduced to 64 dimensions) consistently improves the results of the FV; see Section 3.4 for possible reasons. This dimensionality reduction does not improve the results for VLAD, probably because it does not benefit from the decorrelating effect of the PCA. As a result, $\text{FV} + \text{PCA}$ outperforms VLAD by a few points of mAP.

In the rest of the paper, the local descriptors are reduced to 64 dimensions using PCA in all experiments.

Impact of the codebook size. Fig. 2 clearly shows that the larger the number of centroids, the better the performance. For $K = 4{,}096$ we obtain a $\text{mAP} = 68.9\%$, which outperforms any result reported for standard BOW on this data set ([35] reports $\text{mAP} = 57.2\%$ with a 200k vocabulary using the same descriptors). Interestingly, the slope of the curve does not exhibit a saturation for the values of K considered. However, for high values of K, the dimensionality of the vector becomes very large and is not compatible with the dimensionality reduction strategy proposed in this paper.

Comparison of BOW/VLAD/FV. The results are presented in Tables 1 and 2. Overall, one can observe that Fisher and VLAD provide competitive results. On Holidays and UKB, both Fisher and VLAD with only $K = 64$ outperform the BOW representation. However, BOW is comparatively better when the variability of the objects/images is limited, as for the Oxford Building data set and the Copydays data set. In this case, very large vocabularies are necessary and Philbin et al. [1] used up to 1 million visual words, thereby obtaining a $\text{mAP} = 49.3\%$ on Oxford5K when learning the 1M vocabulary on the Paris data set[4] and using soft quantization. FV outperforms these results by increasing K: Using the setup of [1] (same descriptors and same learning set), we obtained the following mAP values:

$K =$	512	1,024	2,048	4,096
mAP(%)	49.2	52.8	54.5	55.3

Dimensionality reduction on BOW/VLAD/FV. Our objective is to obtain a very compact representation. We, therefore, evaluate the performance of each representation when reduced to $D' = 128$ dimensions. Tables 1 and 2 show that the conclusions for the plain representations remain valid. The main difference appears for the Oxford data set where the PCA has a strong negative impact on the performance, especially in the case of BOW.

Dimension reduction does not necessarily reduce the accuracy, as already observed in [12]. Indeed, Table 1 shows that a limited reduction tends to improve the accuracy for BOW, Fisher, and VLAD. It is also worth noticing that higher dimensional representations, which usually provide better accuracy, suffer more from the dimensionality reduction. This is especially true for Fisher kernels and VLAD: for $D' = 64, 128, 512$, using only 64 centroids/mixtures is better than using larger values of K. Using only $D' = 128$ dimensions, i.e., the dimensionality of a single SIFT descriptor, Fisher attains the excellent accuracy of $\text{mAP} = 56.5\%$ on Holidays, which is comparable to the result obtained with BOW when using a vocabulary of 200k visual words.

Comparison with GIST. The global GIST descriptor is very popular in some recognition systems [11]. It was compared with BOW in [19], where it was shown to provide lower results than BOW for image search. However, its performance is competitive in a near duplicate detection context. Table 2 confirms these observations. It appears that the full 960-dimensional GIST descriptor is significantly outperformed by all other representations except for the JPEG5 transformation, which does not change the image geometry. This is true even when BOW/Fisher/VLAD are reduced to a small number of dimensions. Overall, this global descriptor results in a poor performance for image search.

Conclusion. In this section, we have evaluated the performance of VLAD and FV and measured the impact of the different dimensionality reduction steps. In particular, our study explains most of the difference between the results reported for FV in [23] and [22]: The better

[4] Results obtained when learning the vocabulary on the Oxford Building set itself do not correspond to a realistic setup, as in that case the vocabulary is very specific and not likely to be reused for other types of images or larger data sets.

TABLE 1
Comparison of BOW, Fisher, and VLAD representations, Before and After Dimension Reduction:
The Performance Is Given for the Full D-Dimensional Descriptor and After a Reduction to D' Components with PCA

Descriptor	K	D	Holidays (mAP)					
			$D' = D$	$\rightarrow D'$=2048	$\rightarrow D'$=512	$\rightarrow D'$=128	$\rightarrow D'$=64	$\rightarrow D'$=32
BOW	1 000	1 000	40.1		43.5	44.4	43.4	40.8
	20 000	20 000	43.7	41.8	44.9	45.2	44.4	41.8
Fisher (μ)	16	1 024	54.0		54.6	52.3	49.9	46.6
	64	4 096	59.5	60.7	61.0	56.5	52.0	48.0
	256	16 384	62.5	62.6	57.0	53.8	50.6	48.6
VLAD	16	1 024	52.0		52.7	52.6	50.5	47.7
	64	4 096	55.6	57.6	59.8	55.7	52.3	48.4
	256	16 384	58.7	62.1	56.7	54.2	51.3	48.1

The number K stands for the number of centroids for BOW and VLAD and for the number of Gaussians for Fisher.

TABLE 2
Performance of the Raw Descriptors as well as Descriptors Compressed to $D' = 128$ on Several Data Sets,
Measured by mAP for All Data Sets Except for UKB

Descriptor	GIST	BOW 200k	BOW 20k		Fisher 64		VLAD 64	
PCA to $D' =$				128		128		128
UKB ($4 \times R@4$)	1.62	2.81	2.87	2.95	3.35	3.33	3.28	3.35
Holidays	36.5	54.0	43.7	45.2	59.5	56.5	55.6	55.7
Oxford		36.4	31.9	15.9	31.7	24.3	30.4	25.7
Oxford (learned on Paris)			35.4	19.4	41.8	30.1	37.8	28.7
Copydays JPEG 5 (+10k images)	100.0	90.6	100.0	100.0	93.7	63.7	94.1	72.5
Copydays Crop 50% (+10k images)	67.8	97.3	100.0	100.0	98.7	92.7	97.7	94.2
Copydays Strong (+10k images)	27.7	70.7	54.3	26.9	59.6	41.2	59.2	42.7

performance of FV in [22] is mainly due to the power normalization, which consistently improves the results, and to the dimensionality reduction of the local descriptors by PCA, which is highly beneficial for FV. In this setup, the results of FV are better than those of VLAD on most databases, see Table 2. In the following, we adopt the choices that are, on average, the best:

- SIFT descriptors reduced to 64 dimensions by PCA;
- FV with power normalization of the components ($\alpha = 0.5$) followed by a L2 normalization.

5 From Vectors to Codes

This section addresses the general problem of coding an image descriptor. Given a D-dimensional input vector, we want to produce a code of B bits encoding the image representation such that the nearest neighbors of a (nonencoded) query vector can be efficiently searched in a set of n encoded database vectors.

We handle this problem in two steps that must be optimized jointly: 1) a projection that reduces the dimensionality of the vector (see previous section) and 2) a quantization used to index the resulting vectors. We consider the recent approximate nearest neighbor search method of Jégou et al. [24], which is briefly described in the next section. We will show the importance of the joint optimization by measuring the mean squared euclidean error generated by each step.

5.1 Approximate Nearest Neighbor

Approximate nearest neighbors search methods [36], [17], [37], [11], [18] are required to handle large databases in computer vision applications [38]. One of the most popular techniques is Euclidean Locality-Sensitive Hashing [36], which has been extended in [17] to arbitrary metrics. However, these approaches and the one of Muja and Lowe [37] are memory consuming, as multiple hash tables or trees are required. The method of Weiss et al. [18], which embeds the vector into a binary space, better satisfies the memory constraint. It is, however, significantly outperformed in terms of the tradeoff between memory and accuracy by the product quantization-based approximate search method of Jégou et al. [24]. In the following, we use this method as it offers better accuracy and because the search algorithm provides an explicit approximation of the indexed vectors. This allows us to compare the vector approximations introduced by the dimensionality reduction and the quantization, respectively. We use the asymmetric distance computation (ADC) variant of this approach, which only encodes the vectors of the database, but not the query vector. This method is summarized in the following.

5.1.1 ADC Approach

Let $x \in \mathbb{R}^D$ be a query vector and $\mathcal{Y} = \{y_1, \dots, y_n\}$ a set of vectors in which we want to find the nearest neighbor $\mathrm{NN}(x)$ of x. The ADC approach consists in encoding each vector y_i by a quantized version $c_i = q(y_i) \in \mathbb{R}^D$. For a quantizer $q(.)$ with k centroids, the vector is encoded by $B = \log_2(k)$ bits, k being a power of 2. Finding the a nearest neighbors $\mathrm{NN}_a(x)$ of x simply consists of computing

$$\mathrm{NN}_a(x) = a\text{-} \arg\min_i \|x - q(y_i)\|^2. \tag{21}$$

Note that, in contrast with the embedding method of Weiss et al. [18], the query x is not converted to a code: There is no approximation error on the query side.

To get a good vector approximation, k should be large ($k = 2^{64}$ for a 64 bit code). For such large values of k, learning a k-means codebook as well as assigning to the centroids is not tractable. Our solution is to use a product quantization method which defines the quantizer without explicitly enumerating its centroids. A vector x is first split into m subvectors $x^1, \ldots, x^m$ of equal length D/m. A product quantizer is then defined as a function

$$q(x) = \big(q_1(x^1), \ldots, q_m(x^m)\big), \tag{22}$$

which maps the input vector x to a tuple of indices by separately quantizing the subvectors. Each individual quantizer $q_j(.)$ has k_s reproduction values learned by k-means. To limit the assignment complexity $\mathcal{O}(m \times k_\mathrm{s})$, k_s is a small value (e.g., $k_\mathrm{s} = 256$). However, the set k of centroids induced by the product quantizer $q(.)$ is large: $k = (k_\mathrm{s})^m$.

The square distances in (21) are computed using the decomposition

$$\|x - q(y_i)\|^2 = \sum_{j=1,\ldots,m} \big\|x^j - q_j(y_i^j)\big\|^2, \tag{23}$$

where y_i^j is the jth subvector of y_i. The square distances in this summation are read from look-up tables computed, prior to the search, between each subvector x^j and the k_s centroids associated with the corresponding quantizer q_j. The generation of the tables is of complexity $\mathcal{O}(D \times k_\mathrm{s})$. When $k_\mathrm{s} \ll n$, this complexity is negligible compared with the summation cost of $\mathcal{O}(D \times n)$ in (21).

This quantization method offers an explicit vector approximation: A database vector y_i can be decomposed as

$$y_i = q(y_i) + \varepsilon_\mathrm{q}(y_i), \tag{24}$$

where $q(y_i)$ is the centroid associated with y_i and $\varepsilon_\mathrm{q}(y_i)$ the error vector generated by the quantizer.

Notation: ADC $m \times b_\mathrm{s}$ refers to the method when using m subvectors and b_s bits to encode each subvector ($b_\mathrm{s} = \log_2 k_\mathrm{s}$). The total number of bits B used to encode a vector is then given by $B = m\, b_\mathrm{s}$.

5.2 Indexation-Aware Dimensionality Reduction

Dimensionality reduction is an important step in approximate nearest neighbor search as it impacts the subsequent indexing. In this section, for the ADC approach we express the tradeoff between this operation and the indexing scheme using a single quality measure: the approximation error. For the sake of presentation, we assume that the mean of each vector component is 0. By construction, this is approximately the case for Fisher[5] and VLAD vectors.

The $D' \times D$ PCA matrix M maps descriptor $x \in \mathbb{R}^D$ to the transformed descriptor $x' = Mx \in \mathbb{R}^{D'}$. It is the upper part of an orthogonal matrix. This dimensionality reduction can also be interpreted in the initial space as a projection. In that case, x is approximated by

$$x_\mathrm{p} = x - \varepsilon_\mathrm{p}(x), \tag{25}$$

where the error vector $\varepsilon_\mathrm{p}(x)$ lies in the null space of M. The vector x_p is related to x' by the pseudo-inverse of M, which is the transpose of M in this case. Therefore, the projection is $x_\mathrm{p} = M^\top M x$. For the purpose of indexing, the vector x' is subsequently encoded as $q(x')$ using the ADC approach, which can also be interpreted in the original D-dimensional space as the approximation[6]

$$q(x_\mathrm{p}) = x - \varepsilon_\mathrm{p}(x) - \varepsilon_\mathrm{q}(x_\mathrm{p}), \tag{26}$$

where $\varepsilon_\mathrm{p}(x) \in \mathrm{Null}(M)$ and $\varepsilon_\mathrm{q}(x_\mathrm{p}) \in \mathrm{Null}(M)^\perp$ (because the ADC quantizer is learned in the principal subspace) are orthogonal. At this point, we make two observations:

1. Due to the PCA, the variance of the different components of x' is not balanced. Therefore, the ADC structure, which allocates a fixed number of bits per subvector, quantizes the first principal components more coarsely in comparison with the last components, leading to a high quantization error on the first components. In order to address this problem, it is possible to balance the components' variance by applying an orthogonal transformation after the PCA. In [23], two strategies are compared. First, a Householder matrix is learned to perfectly balance the energy on the components. The second strategy simply consists of applying a random rotation matrix after the PCA. Both approaches improve the results, and the random matrix is shown to be as effective as the learned one. We therefore adopt this choice in the following.
2. There is a tradeoff in the number of dimensions D' to be retained by the PCA. If D' is large, the projection error vector $\varepsilon_\mathrm{p}(x)$ is of limited magnitude, but a large quantization error $\varepsilon_\mathrm{q}(x_\mathrm{p})$ is introduced. On the other hand, keeping a small number of components leads to a high projection error and a low quantization error.

Joint optimization of reduction/indexing. Let us now consider the second problem, i.e., optimizing the dimension D', having fixed a constraint on the number of bits B used to represent the D-dimensional vector x, for instance $B = 128$ (16 bytes). The squared euclidean distance between the reproduction value and x is the sum of the errors $\|\varepsilon_\mathrm{p}(x)\|^2$ and $\|\varepsilon_\mathrm{q}(x_\mathrm{p})\|^2$, both of which depend on the selected D'. The mean square error $e(D')$ is empirically measured on a learning vector set $\mathcal{L}$ as

$$e(D') = e_p(D') + e_q(D') \tag{27}$$

$$= \frac{1}{\mathrm{card}(\mathcal{L})} \sum_{x \in \mathcal{L}} \|\varepsilon_\mathrm{p}(x)\|^2 + \|\varepsilon_\mathrm{q}(x_\mathrm{p})\|^2. \tag{28}$$

[5] This comes from the property $E_{x\sim u_\lambda} \nabla_\lambda \log u_\lambda(x) = \nabla_\lambda E_{x\sim u_\lambda} \log u_\lambda(x) \approx 0$ if u_λ is estimated with MLE.

[6] For the sake of conciseness, the quantities $M^T q(x')$ and $M^T \varepsilon_\mathrm{q}(x')$ are simplified to $q(x_\mathrm{p})$ and $\varepsilon_\mathrm{q}(x_\mathrm{p})$, respectively.

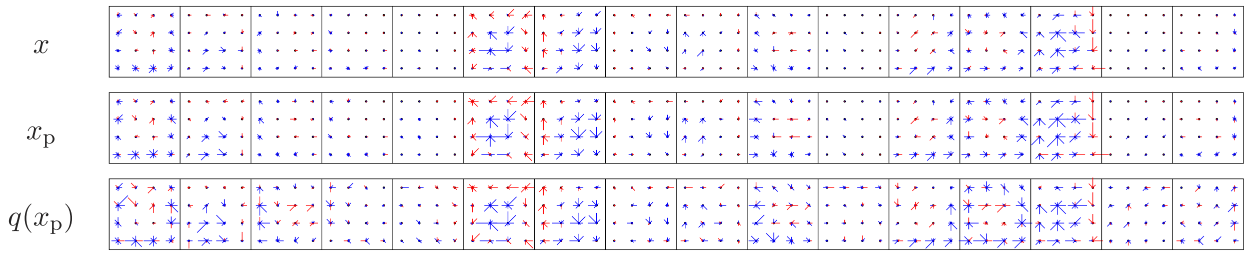

Fig. 3. Effect of the encoding steps on the descriptor. *Top*: VLAD vector x for $K = 16$ ($D = 2{,}048$). *Middle*: Vector x_{p} altered by the projection onto the PCA subspace ($D' = 128$). *Bottom*: Vector $q(x_{\mathrm{p}})$ after indexing by ADC 16×8 (16-bytes code).

This gives us an objective criterion to directly optimize the dimensionality, which is obtained by finding on the learning set the value of D' minimizing this criterion.

Remarks.

- The selection of D' using this mean square error minimization is not optimized with respect to an image search criterion. Note, however, that minimizing this error appears to be a reasonable choice, as empirically observed in [23].
- In the ADC indexing scheme, D' must be a multiple of m. For instance, by keeping $D' = 64$ eigenvalues, the valid set of values for m is $\{1, 2, 4, 8, 16, 32, 64\}$.

The impact of dimensionality reduction and indexation based on ADC is illustrated by the VLAD pictorial representation introduced in Section 3. We can present the projected and quantized VLAD in this form, as both the PCA projection and the quantization provide a way of reconstructing the projected/quantized vector. Fig. 3 illustrates how each of these operations impacts our representation. One can see that the vector is only slightly altered, even for a compact representation of $B = 16$ bytes.

6 Experiments

In this section, we evaluate the performance of the Fisher vector when used with the joint dimensionality reduction/indexing approach of Section 5. This evaluation uses the improved FV representation of Perronnin et al. [22] in conjunction with the indexing scheme of Jégou et al. [23]. Our comparison focuses on the UKB and Holidays data sets. Large scale experiments on Holidays+Flickr10M were used to measure search accuracy and efficiency on a large scale of 10 million images. As we focus on the intrinsic quality of the large scale system, we do not apply the reranking stage, which is usually performed on a short list to filter out geometrically inconsistent results [1], [35].

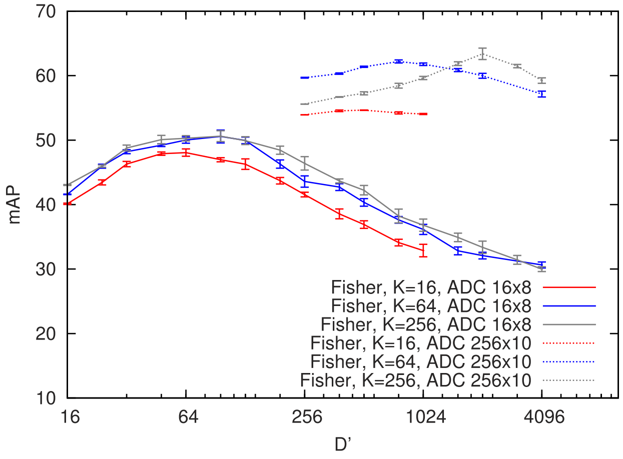

Fig. 4. Search on Holidays with ADC 16×8 and ADC 256×10 representations, for reduction to varying dimensions D'. Experiments are averaged over five learning runs (error bars = standard deviations over the runs).

6.1 Dimensionality Reduction and Indexation

Given an image representation with a vector of length D and a fixed number B of bits to encode this vector, Fig. 4 confirms the analysis of Section 5: There is an important variation with respect to D'. A method to fix D' consists in minimizing the total square error introduced by the projection and the quantization steps, as suggested in Section 5.2. This choice may not be optimal with respect to image search quality measured by, e.g., mAP. However, as the performance is stable around the optimal value of D' (see Fig. 4), in practice the proposed optimization procedure provides close-to-optimal results.

6.2 Comparison with the State of the Art

Table 3 and Fig. 5 compare the performance obtained by our indexing scheme to the state of the art on the Holidays and UKB data sets. Compared to the miniBOF approach of Jégou et al. [12], the proposed approach is significantly more precise at all operating points. Compared to BOW, our

TABLE 3
Comparison of Our Approach with the State of the Art on UKB (Score/4) and Holidays (mAP)

Method	bytes	UKB	Holidays
BOW, K=20,000	10 364	2.87	43.7
BOW, K=200,000	12 886	2.81	54.0
miniBOF [12]	20	2.07	25.5
	80	2.72	40.3
	160	2.83	42.6
FV K=64, spectral hashing 128 bits	16	2.57	39.4
VLAD, K=16, ADC 16×8 [23]	16	2.88	46.0
VLAD, K=64, ADC 32×10 [23]	40	3.10	49.5
FV K=8, binarized [22]	65	2.79	46.0
FV K=64, binarized [22]	520	3.21	57.4
FV K=64, ADC 16×8 (D'=96)	**16**	**3.10**	**50.6**
FV K=256, ADC 256×10 (D'=2048)	**320**	**3.47**	**63.4**

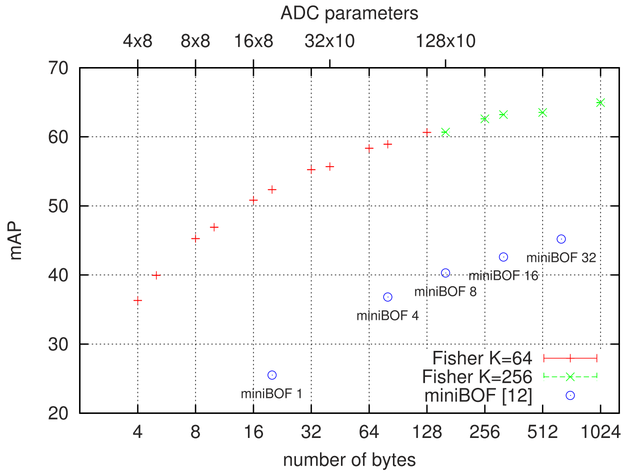

Fig. 5. mAP for a search on Holidays with varying ADC quantization parameters (number of bytes). Results of [12] are reported with circles for reference.

approach obtains a comparable search quality with about two orders of magnitude less memory. With respect to the approaches [22] and [23], a significant improvement is obtained by using the improved FV of Perronnin et al. [22] jointly with the indexing method of Jégou et al. [23].

Fig. 5 also illustrates the tradeoff between search quality and memory usage. Interestingly, the best choice for the number K of Gaussians depends on the number of bits B chosen to represent the image. Compared to BOW, which gives $\mathrm{mAP} = 54\%$ for a 200k vocabulary, a competitive accuracy of $\mathrm{mAP} = 55.2\%$ is obtained with only 32 bytes. Note that small (respectively, large) values of K should be associated with small (respectively, large) values of B, as they are more impacted by dimensionality reduction. In this figure, the variant $K = 16$ is never selected as the best, which explains why only $K = 64$ and $K = 256$ appear in the plot.

Table 3 also compares our indexing scheme to the spectral hashing [18] coding scheme. For a memory usage of 16 bytes, ADC outperforms spectral hashing by more than 10 points of mAP on Holidays. Similarly, it is possible to use significantly less memory to attain the same performance: Using 5 bytes per image in ADC achieves the same mAP ($\approx 40\%$) as SH with 16 bytes, see Fig. 5.

6.3 Large-Scale Experiments

6.3.1 Experiments on Holidays and Flickr10M

Fig. 6 shows the behavior of our approach on a large scale. We have constructed a data set by combining the images from Holidays with a subset of Flickr10M of increasing size. Recall that, in such a large data set, the number of outliers is very important. The challenge is to rank the relevant images well enough to ensure that geometrical verification will validate them in a reranking stage, which can typically be performed for a few hundred images only.

For this experiment, we have used the nonexhaustive search variant of ADC called IVFADC. IVFADC combines ADC with an inverted file to restrict the search to a subset of vectors: In our case, only 64 lists of coded descriptors are visited out of 8,192 lists in total. Consequently, it stores the image identifiers explicitly, which requires an extra 4 bytes of memory per image (see [24] for details). Compared to ADC, on large data sets IVFADC is one or two orders of magnitude faster and gives slightly better results.

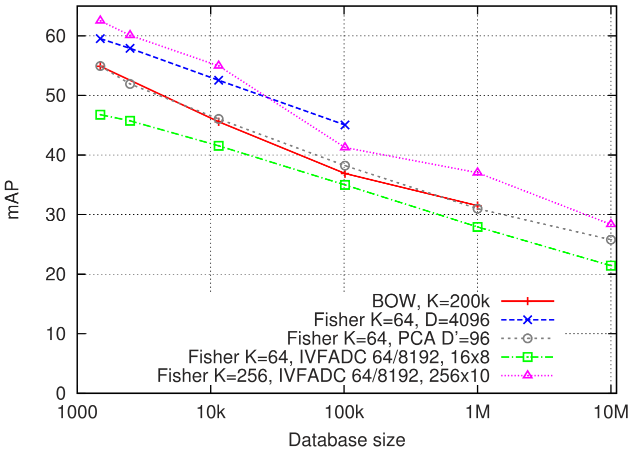

Fig. 6. Search accuracy as a function of the database size.

The mAP performance is displayed as a function of the data set size. We report the results for

- the plain Fisher vector ($K = 64$, $D = 4{,}096$),
- the same Fisher vector reduced to $D' = 96$ dimensions by PCA,
- these PCA-reduced FV indexed by IVFADC with 16×18 codes, i.e., $16 + 4 = 20$ bytes per indexed image. We also present a more expensive operating point, for which $K = 256$, $D' = 2{,}048$, and 256×10 codes have been used, leading to a representation of 324 bytes in memory.

The results are significantly better than those reported in [12], where a mAP of 6.6 percent is reported for 1 million images and a 20-bytes representation, against $\mathrm{mAP} = 27.9\%$ for the same memory usage with the proposed method, and $\mathrm{mAP} = 37.0\%$ when using 324 bytes. Note that this is also an improvement over [23], where we obtained $\mathrm{mAP} = 24.1\%$ using the VLAD vector. Interestingly, the 96-dimensional vector obtained by PCA offers results comparable to those of BOW with a large vocabulary.

In order to measure how our system would benefit from being used in conjunction with a postverification scheme, Fig. 7 gives the recall@R as a function of the number R. It can be interpreted as the rate of relevant images that will be geometrically verified if we consider that a verification is applied on a shortlist of R images (typically, R is limited to 100). The experiment is limited to $\mathrm{Holidays{+}Flickr1M}$ because BOW does not scale to 10 million images. With a representation of 20 bytes, the proposed approach is almost as accurate as BOW and becomes significantly better when increasing the size of the codes.

6.3.2 Experiments on Copydays and Exalead100M

Given the compactness of our image descriptor encoding, our method can scale up to 1 billion images with a reasonable amount of RAM (for instance, 20 GB when using only 20 bytes per image). Only a few systems are able to work on such a scale, an example is the GISTIS approach of Douze et al. [19]. This approach combines the

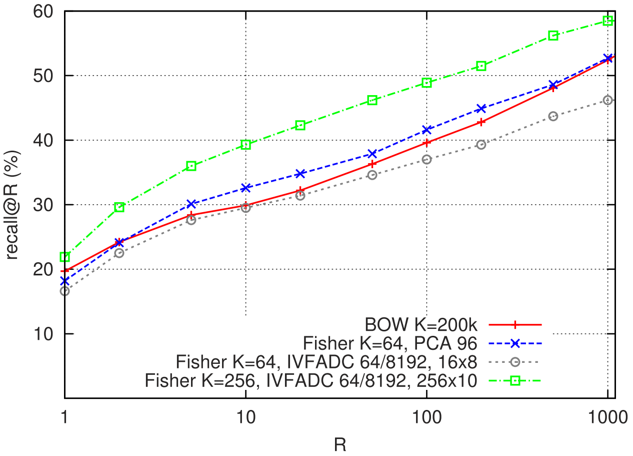

Fig. 7. Quality of the shortlist (of varying size R): recall@R when searching in 1 million images.

GIST descriptor [16] with an indexing method derived from [26]. Results are reported on 110 million tiny images (32×32 pixels). In the following, we report a comparison of our approach with the GISTIS method for Copydays merged with the Exalead100M data set (see Section 2 for a description). Since the images are smaller in this setup (150 pixels in their larger dimension), the local descriptor's threshold (on cornerness) is reduced to increase the number of descriptors input to the Fisher computation.

Fig. 8 shows the results for two subsets of Copydays: crop 50 percent (157 transformed queries, one per database image) and strong transformations (229 query images, each of which has only one matching image in the database). The number of distracting images from Exalead100M is increased up to the full data set (100M images). We can observe that the uncompressed Fisher descriptor ($K = 64$, $D = 4{,}096$) is clearly more discriminant than the uncompressed color GIST descriptor ($D = 960$). The IVFADC configuration (IVF 64/8192, 64×8) was selected to allocate the same size per descriptor as the GISTIS indexing method: 68 bytes per image. The results after indexing are better than for GISTIS. Moreover, the slope of the curves shows that our approach is less sensitive to distractors: The comparative performance is much better for our scheme as the database grows.

Timings. All timing experiments have been performed on a single processor core. On average, searching our 100 million data set with the IVFADC indexing structure (64×8 codes, 64 lists visited out of 8,192) takes 245 ms. This efficiency is at least two orders of magnitude above the BOW: Jégou et al. [35] report a query time of 620 ms on a quad-core processor to search in 1 million images given a vocabulary of $K = 200k$ visual words. The time search is 50 percent higher than the one of GISTIS (160 ms), but for a significantly better search quality.

7 Conclusion

Many state-of-the-art large-scale image search systems follow the same paradigm: Statistics computed from local invariant features are aggregated into an image-level vector signature which is subsequently compressed and indexed for computational and memory efficiency. The BOW histogram has become a standard for the aggregation part. For the compression part, most approaches use binary embeddings. This paper departs from this paradigm in two ways.

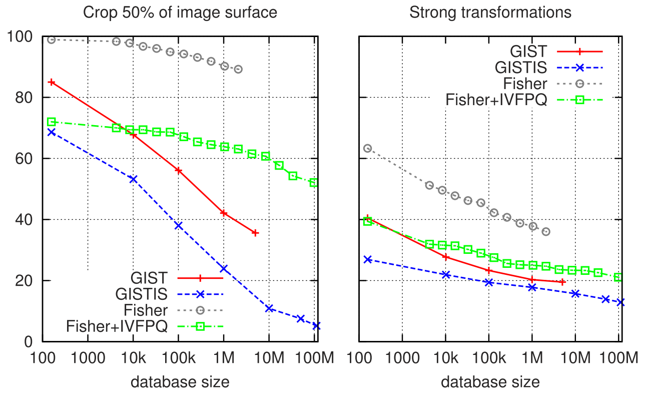

Fig. 8. Copydays retrieval results (mAP), for two types of transformations and a varying number of distractors.

We first propose using the Fisher kernel framework for local feature aggregation. This representation is shown to yield high performance and its accuracy remains competitive even after a significant dimensionality reduction to 128 dimensions, i.e., of the same size as a single SIFT vector.

Second, we employ an asymmetric product quantization scheme for the vector compression part and jointly optimize the dimensionality reduction and compression. Impressive search results are achieved with a tiny binary code, e.g., a mAP of 50.6 percent on Holidays with 128 bits signatures. With such a small memory footprint, 1 billion images fit in the RAM of a 20 GB server. We achieve a response time of 250 ms on a 100 million image data set on a single processor core.

Acknowledgments

The authors thank the company Exalead for providing the corpus of 100M images. This work was partially funded by the European project PINVIEW, by the QUAERO project (supported by OSEO, the French State agency for innovation), by the European integrated project AXES, and by the ANR project GAIA.

References

[1] 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*, June 2007.

[2] D. Nistér and H. Stewénius, "Scalable Recognition with a Vocabulary Tree," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 2161-2168, June 2006.

[3] Z. Wu, Q. Ke, M. Isard, and J. Sun, "Bundling Features for Large Scale Partial-Duplicate Web Image Search," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 25-32, 2009.

[4] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford, "Video Copy Detection: A Comparative Study," *Proc. ACM Int'l Conf. Image and Video Retrieval*, pp. 371-378, 2007.

[5] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int'l J. Computer Vision*, vol. 88, pp. 303-338, June 2010.

[6] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *Proc. Ninth IEEE Int'l Conf. Computer Vision*, pp. 1470-1477, Oct. 2003.

[7] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[8] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, Oct. 2005.

[9] S. Winder and M. Brown, "Learning Local Image Descriptors," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2007.

[10] S. Winder, G. Hua, and M. Brown, "Picking the Best Daisy," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2009.

[11] A. Torralba, R. Fergus, and Y. Weiss, "Small Codes and Large Databases for Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2008.

[12] H. Jégou, M. Douze, and C. Schmid, "Packing Bag-of-Features," *Proc. 12th IEEE Int'l Conf. Computer Vision*, Sept. 2009.

[13] O. Chum, M. Perdoch, and J. Matas, "Geometric Min-Hashing: Finding a (Thick) Needle in a Haystack," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, June 2009.

[14] O. Chum, J. Philbin, and A. Zisserman, "Near Duplicate Image Detection: Min-Hash and tf-idf Weighting," *Proc. 19th British Machine Vision Conf.*, Sept. 2008.

[15] L. Torresani, M. Szummer, and A. Fitzgibbon, "Learning Query-Dependent Prefilters for Scalable Image Retrieval," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, June 2009.

[16] A. Oliva and A. Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope," *Int'l J. Computer Vision*, vol. 42, no. 3, pp. 145-175, 2001.

[17] B. Kulis and K. Grauman, "Kernelized Locality-Sensitive Hashing for Scalable Image Search," *Proc. 12th IEEE Int'l Conf. Computer Vision*, Oct. 2009.

[18] Y. Weiss, A. Torralba, and R. Fergus, "Spectral Hashing," *Proc. Conf. Neural Information Processing Systems*, 2008.

[19] M. Douze, H. Jégou, H. Singh, L. Amsaleg, and C. Schmid, "Evaluation of GIST Descriptors for Web-Scale Image Search," *Proc. ACM Int'l Conf. Image and Video Retrieval*, July 2009.

[20] T. Jaakkola and D. Haussler, "Exploiting Generative Models in Discriminative Classifiers," *Proc. Conf. Neural Information Processing Systems*, 1998.

[21] F. Perronnin and C.R. Dance, "Fisher Kernels on Visual Vocabularies for Image Categorization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2007.

[22] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-Scale Image Retrieval with Compressed Fisher Vectors," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2010.

[23] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating Local Descriptors into a Compact Image Representation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2010.

[24] H. Jégou, M. Douze, and C. Schmid, "Product Quantization for Nearest Neighbor Search," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117-128, Jan. 2011.

[25] 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, nos. 1/2, pp. 43-72, 2005.

[26] H. Jégou, M. Douze, and C. Schmid, "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search," *Proc. 10th European Conf. Computer Vision*, Oct. 2008.

[27] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, June 2008.

[28] J. van Gemert, C. Veenman, A. Smeulders, and J. Geusebroek, "Visual Word Ambiguity," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1271-1283, July 2010.

[29] H. Jégou, M. Douze, and C. Schmid, "On the Burstiness of Visual Elements," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2009.

[30] J. Winn, A. Criminisi, and T. Minka, "Object Categorization by Learned Universal Visual Dictionary," *Proc. 10th IEEE Int'l Conf. Computer Vision*, 2005.

[31] F. Perronnin, J. Sánchez, and Y. Liu, "Large-Scale Image Categorization with Explicit Data Embedding," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010.

[32] A. Vedaldi and A. Zisserman, "Efficient Additive Kernels via Explicit Feature Maps," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2010.

[33] X. Zhang, Z. Li, L. Zhang, W. Ma, and H.-Y. Shum, "Efficient Indexing for Large-Scale Visual Search," *Proc. 12th IEEE Int'l Conf. Computer Vision*, Oct. 2009.

[34] C.M. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2007.

[35] H. Jégou, M. Douze, and C. Schmid, "Improving Bag-of-Features for Large Scale Image Search," *Int'l J. Computer Vision*, vol. 87, pp. 316-336, Feb. 2010.

[36] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, "Locality-Sensitive Hashing Scheme Based on P-Stable Distributions," *Proc. Symp. Computational Geometry*, pp. 253-262, 2004.

[37] M. Muja and D.G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," *Proc. Int'l Conf. Computer Vision Theory and Applications*, Feb. 2009.

[38] G. Shakhnarovich, T. Darrell, and P. Indyk, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*, chapter 3. MIT Press, Mar. 2006.

Hervé Jégou received the PhD degree in computer science from the University of Rennes, defended in 2005, on joint source channel coding. He is a former student of the Ecole Normale Supérieure de Cachan. He joined INRIA as a permanent researcher in 2006, on the LEAR team, and moved to INRIA Rennes in 2009. His research interests include large scale indexing (image, audio, and video). He is a member of the IEEE.

Florent Perronnin received the engineering degree in 2000 from the ENST (Paris) and the PhD degree in 2004 from the EPFL (Lausanne, Switzerland). From 2000 to 2001 he was a research engineer with the Panasonic Speech Technology Laboratory, Santa Barbara, California working on speech and speaker recognition. In 2005, he joined the Xerox Research Center Europe, Grenoble, France. His main interests are in the practical application of machine learning to computer vision tasks such as image classification, retrieval, or segmentation. He is a member of the IEEE.

Matthijs Douze received the MS degree in computer science and the PhD degree from the University of Toulouse, France. He has been an engineer on the LEAR project-team at INRIA Rhône-Alpes since 2005. He works on local image descriptors and image and video indexing.

Jorge Sánchez received the engineering degree from the National Technological University, Córdoba Regional Faculty, in 2006. He is currently working toward the PhD degree from the National University of Córdoba with a grant from CONICET, Argentine. His research interests include robot navigation, image classification, object recognition, and large-scale problems in computer vision.

Patrick Pérez received the PhD degree from the University of Rennes in 1993 and joined INRIA in 1994 as a full time researcher. From 2000 to 2004, he was with Microsoft Research Cambridge. In 2009, he joined Technicolor Research & Innovation as a senior researcher. He is currently an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and a member of the editorial board of the *International Journal of Computer Vision*.

Cordelia Schmid received the MS degree in computer science from the University of Karlsruhe and the doctorate from the Institut National Polytechnique de Grenoble. She is a research director at INRIA Grenoble and directs the project-team called LEAR, for LEArning and Recognition in Vision. She is the author of more than 100 technical publications. In 2006, she was awarded the Longuet-Higgins prize for fundamental contributions in computer vision that have withstood the test of time. She is a fellow of the IEEE.

▹ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.