Asymmetric Distances for Binary Embeddings

Albert Gordo, Florent Perronnin, Yunchao Gong, Svetlana Lazebnik

Abstract—In large-scale query-by-example retrieval, embedding image signatures in a binary space offers two benefits: data compression and search efficiency. While most embedding algorithms binarize both query and database signatures, it has been noted that this is not strictly a requirement. Indeed, asymmetric schemes which binarize the database signatures but not the query still enjoy the same two benefits but may provide superior accuracy. In this work, we propose two general asymmetric distances which are applicable to a wide variety of embedding techniques including Locality Sensitive Hashing (LSH), Locality Sensitive Binary Codes (LSBC), Spectral Hashing (SH), PCA Embedding (PCAE), PCA Embedding with random rotations (PCAE-RR), and PCA Embedding with iterative quantization (PCAE-ITQ). We experiment on four public benchmarks containing up to 1M images and show that the proposed asymmetric distances consistently lead to large improvements over the symmetric Hamming distance for all binary embedding techniques.

Index Terms—Large-scale retrieval, binary codes, asymmetric distances.

1 Introduction

Recently, the computer vision community has witnessed an explosion in the scale of the datasets it has had to handle. While standard image benchmarks such as PASCAL VOC [2] or CalTech 101 [3] used to contain only a few thousand images, resources such as ImageNet [4] (14 million images) and Tiny images [5] (80 million images) are now available. In parallel, more and more sophisticated image descriptors have been proposed including the GIST [6], the bag-of-visual-words (BOV) histogram [7], [8], the Fisher vector (FV) [9], [10] or the Vector of Locally Aggregated Descriptors (VLAD) [11]. Descriptors with thousands or tens of thousands of dimensions have become the norm rather than the exception. Consequently, handling these gigantic quantities of data has become a challenge on its own.

When dealing with large amounts of data, there are two considerations of paramount importance. The first one is the *computational cost*: the computation of the distance between two image signatures should rely on efficient operations. The second one is the *memory cost*: the memory footprint of the objects should be small enough so that all database image signatures fit in RAM. If this is not the case, *i.e.* if a significant portion of the database signatures has to be stored on disk, then the response time of a query collapses because the disk access is much slower than that of RAM access.

These considerations have directly motivated research on image descriptor compression and especially on learning compact binary codes [12], [13], [14], [15], [16], [11], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. A desirable property of such coding schemes is that they should map similar data points (with respect to a given metric such as the Euclidean distance) to similar binary vectors (*i.e.* vectors with a small Hamming distance). Transforming high-dimensional real-valued image descriptors into compact binary codes directly addresses both memory and computational problems. First the compression enables to store a large number of codes in RAM. Second, the Hamming distance is extremely efficient to compute in hardware, which enables the exhaustive computation of millions of distances per second, even on a single CPU.

However, it has been noted that compressing the query signature is not mandatory [31], [32], [17]. Indeed, the additional cost of storing in memory a single non-binarized signature is negligible. Also, the distance between an original signature and a compressed signature can still be computed efficiently through look-up table operations. As the distance is computed between two different spaces, these algorithms are referred to as *asymmetric*. A major benefit of asymmetric algorithms is that they can achieve higher accuracy for a fixed compression rate because they take advantage of the more precise position information of the query. We note however that the asymmetric algorithms presented in [31], [32], [17] are tied to specific compression schemes. Dong *et al*. [31] presented an asymmetric algorithm for compression schemes based on random projections. Jégou *et al*. [32] proposed an asymmetric algorithm for compression schemes based on vector quantization. Brandt [17] subsequently used a similar idea.

In this work, we show that the notion of asym-

- *A. Gordo is with the Computer Vision Center, Universitat Autònoma de Barcelona, Spain. E-mail: agordo@cvc.uab.es.*
- *F. Perronnin is with Textual Visual Pattern Analysis, Xerox Research Centre Europe, France. E-mail: florent.perronnin@xrce.xerox.com.*
- *Y. Gong is with the computer science department, University of North Carolina at Chapel Hill, USA. E-mail: yunchao@cs.unc.edu.*
- *S. Lazebnik is with the comoputer science department, University of Illinois at Urbana-Champaign, USA. E-mail: lazebnik@cs.unc.edu.*
- *A preliminary version of this paper [1] appears in CVPR2011.*

metric distances can be applied very broadly. We first provide an overview of several binary embedding algorithms (including Locality Sensitive Hashing (LSH) [12], [13], Locality-Sensitive Binary Codes (LSBC) [14], Spectral Hashing (SH) [15], PCA Embedding (PCAE) [1], [22], PCAE with random rotations (PCAE-RR), and PCAE with iterative quantization (PCAE-ITQ) [22]) showing that they can be decomposed into two steps: i) the signatures are first embedded in an intermediate real-valued space and ii) thresholding is performed in this space to obtain binary outputs. A key insight that our asymmetric distances will exploit is that the Euclidean distance is a "natural metric" in the intermediate real-valued space, *i.e.* the Euclidean distance in the intermediate space approximates the metric/kernel in the original space.

Building on the previous analysis we propose two asymmetric distances which can be broadly applied to binary embedding algorithms. The first one is an expectation-based technique inspired by Jégou *et al*. [32]. The second one is a lower-bound-based technique inspired by Dong *et al*. [31].

We show experimentally on four datasets of different nature that the proposed asymmetric distances consistently and significantly improve the retrieval accuracy of LSH, LSBC, SH, PCAE, PCAE-RR, and PCAE-ITQ over the symmetric Hamming distance. Although the lower-bound and expectation-based techniques are very different in nature, they are shown to yield very similar improvements.

The remainder of this article is organized as follows. In the next section, we provide an analysis of several binary embedding techniques. In section 3 we build on the previous analysis to propose two asymmetric distance computation algorithms for binary embeddings. In section 4 we provide experimental results. Finally in section 5 we discuss conclusions and directions for future work.

2 Background on Binary Embeddings

We now provide a review of several successful binary embedding techniques: LSH, LSBC, SH, PCAE, PCAE-RR, and PCAE-ITQ. Let us introduce a set of notations. Let x be an image signature in a space Ω and let h_k be a binary embedding function, *i.e.* $h_k : \Omega \rightarrow \{0,1\}$ (some authors prefer the convention $h_k : \Omega \rightarrow \{-1,+1\}$). A set $\mathcal{H} = \{h_k, k = 1 \dots K\}$ of K functions defines a multi-dimensional embedding function $h : \Omega \rightarrow \{0,1\}^K$ with $h(x) = [h_1(x), \dots, h_K(x)]'$ (and the apostrophe denotes the transpose).

We show that for LSH, LSBC, SH, PCAE, PCAE-RR, and PCAE-ITQ, the functions h_k can be decomposed as follows:

$$h_k(x) = q_k[g_k(x)], \tag{1}$$

where $g_k : \Omega \rightarrow \mathbb{R}$ is the real-valued embedding function, and $q_k : \mathbb{R} \rightarrow \{0,1\}$ is the binarization function. We denote $g : \Omega \rightarrow \mathbb{R}^K$ with $g(x) = [g_1(x), \dots, g_K(x)]'$. If we have two image signatures x and y, we also show that the Euclidean distance is the natural metric between $g(x)$ and $g(y)$ and that it approximates the original distance between x and y. Thus, we can write the squared Euclidean distance as $d(x,y) \approx d(g(x), g(y)) = \sum_k d(g_k(x), g_k(y))$.

In the rest of this section, we survey a number of binary embeddings, which can be classified into two types: those based on random projections (LSH and LSBC), and those based on learning the hashing functions (PCAE, PCAE-RR, PCAE-ITQ, or SH).

2.1 Hashing with Random Projections

2.1.1 Locality Sensitive Hashing (LSH)

In LSH, the functions h_k are called hash functions and are selected to approximate a similarity function sim in the original space $\Omega \in \mathbb{R}^D$. Valid hash functions h_k must satisfy the LSH property:

$$Pr\left[h_k(x) = h_k(y)\right] = sim(x,y). \tag{2}$$

Here we focus on the case where sim is the cosine similarity $sim(x,y) = 1 - \frac{\theta(x,y)}{\pi}$, for which a suitable hash function [1] is [13]:

$$h_k(x) = \sigma\left(r'_k x\right), \tag{3}$$

with

$$\sigma(u) = \begin{cases} 0 & \text{if } u < 0, \\ 1 & \text{if } u \geq 0. \end{cases} \tag{4}$$

The vectors $r_k \in \mathbb{R}^D$ are drawn from a multi-dimensional Gaussian distribution p with zero mean and identity covariance matrix I_D. We therefore have $q_k(u) = \sigma(u)$ and $g_k(x) = r'_k x$. In such a case the natural distance between $g(x)$ and $g(y)$ in the intermediate space is the Euclidean distance as random Gaussian projections preserve the Euclidean distance in expectation. This property stems from the equality:

$$\mathbb{E}_{r \sim p}\left[||r'x - r'y||^2\right] = ||x-y||^2. \tag{5}$$

We note that centering the data around the origin can impact LSH very positively, especially when dealing with non-negative data such as GIST vectors [6]. This is because, given a set of points $\{x_i, i = 1 \dots N\}$ with zero-mean and a random direction r_k, we have the guarantee that $\frac{1}{N}\sum_{i=1}^N r'_k x_i = r'_k(\frac{1}{N}\sum_{i=1}^N x_i) = 0$, *i.e.* the distribution of values $r'_k x_i$ is centered around the LSH binarization threshold. If the data is not centered around the origin, the mean of the $r'_k x_i$ values can be very different from zero. We have observed cases where the projections all had the same

[1] In the general LSH case, each hashing function h_k can take more than two values. Here, since we are interested in binary compression, we focus on the binary case. Therefore, in what follows, what we refer to as LSH should be understood as the binary version of LSH.

sign, leading to weakly discriminative embedding functions [2].

The previous analysis can be readily extended to the Kernelized LSH approach of Kulis and Grauman [16] as the Mercer kernel between two objects is just a dot-product in another space using a non-linear mapping $\phi(x)$ of the input vectors. This enables one to extend LSH as well as the asymmetric distance computations beyond vectorial representations.

2.1.2 *Locality-Sensitive Binary Codes (LSBC)*

Consider a Mercer kernel $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ that satisfies the following properties for all points x and y:

1) It is translation invariant, *i.e.* there exists $k : \mathbb{R}^D \to \mathbb{R}$ such that $k(x,y) = k(x-y)$.
2) It is normalized: *i.e.*, $k(x-y) \leq 1$ and $k(0) = 1$.
3) $\forall \alpha \in \mathbb{R} : \alpha \geq 1$, $k(\alpha x - \alpha y) \leq k(x,y)$.

Two well known examples are the Gaussian and Laplacian kernels. Raginsky and Lazebnik [14] showed that for such kernels, $k(x,y)$ can be approximated by $1 - 2\text{Ha}(h(x),h(y))/K$ where $Ha()$ denotes the Hamming distance and $h_k(x) = q_k(g_k(x))$ with:

$$g_k(x) = \cos(r'_k x + b_k), \tag{6}$$

$$q_k(u) = \sigma(u - t_k). \tag{7}$$

b_k and t_k are random values drawn respectively from unif$[0, 2\pi]$ and unif$[-1,+1]$, and K is the number of bits. The vectors r_k are drawn from a distribution p_k which depends on the particular choice of the kernel k. For instance, if k is the Gaussian kernel with bandwidth γ then p_k is a Gaussian distribution with mean zero and covariance matrix γI_D where I_D is the $D \times D$ identity matrix. As the number of bits K increases, $1 - 2\text{Ha}(h(x),h(y))/K$ is guaranteed to converge to $k(x,y)$.

We know from Rahimi and Recht [33] that $g(x)'g(y)$ is guaranteed to converge to $k(x,y)$ since $\mathbb{E}_{w_k \sim p_K} g_k(x) g_k(y) = k(x,y)$ and therefore that the embedding g preserves the dot-product in expectation. Since $k(x,x) = k(x-x) = k(0) = 1$, the norm $||g(x)||^2$ is also guaranteed to converge to 1, $\forall x$. In that case, $||g(x) - g(y)||^2 = ||g(x)||^2 + ||g(y)||^2 - 2g(x)'g(y) = 2(1 - g(x)'g(y))$. Therefore the Euclidean distance is equivalent to the dot-product and the Euclidean distance is preserved in the intermediate space in expectation.

2.2 Learning Hashing Functions

Hashing methods based on random projections such as LSH and LSBC have important properties, such as the guarantee to converge to the target kernel when the number of bits grows to infinity. However, a large number of bits may be necessary to obtain a sufficiently good approximation. When aiming at short codes, it may be more fruitful to learn the hashing functions rather than to resort to randomness.

We will focus on unsupervised code learning techniques, and especially on those based on PCA, since PCA seems to be a core component of the best-performing binary embedding methods: in Product Quantization [32] and Transform Coding [17], PCA is used as a preliminary step before binarizing the data. In [1] and [22], a direct PCA embedding is used. Spectral Hashing [15] can be understood as a way to assign more bits to the PCA dimensions with more energy. In the following, let $\mathcal{S} = \{x_i, i = 1 \dots N\}$, be a set of N signatures in $\Omega \in \mathbb{R}^D$ that are available for training purposes.

2.2.1 *PCA Embedding (PCAE)*

A very simple encoding technique is PCA embedding (PCAE) [22][1]. We can define PCAE as $h_k(x) = q_k(g_k(x))$, with

$$g_k(x) = w'_k(x - \mu), \tag{8}$$

$$q_k(u) = \sigma(u), \tag{9}$$

where μ is the mean of the signatures of $\mathcal{S}$, $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$, and where w_k is the eigenvector associated with the k-th largest eigenvalue of the covariance matrix of the signatures of $\mathcal{S}$. Despite its simplicity, PCAE can obtain very competitive results as will be shown in the experiments of Section 4.

According to [15], when producing binary codes, two desirable properties are: i) that the bits are pairwise uncorrelated, and ii) that the variance of each bit is maximized. As seen in [18], this leads to an NP hard problem that needs to be relaxed to be solved efficiently. The analysis of [18] also shows that projecting the data with PCA is an optimal solution of the relaxed version of this problem, conferring some theoretical soundness to PCAE.

In the case of PCAE, the Euclidean distance is the natural distance in the intermediate space, since PCA projections preserve, approximately, the Euclidean distance (the PCA directions are those that minimize the mean squared reconstruction error). A possible drawback of PCA projections for binarization is that not all the dimensions contain the same energy after the projection. Since, after thresholding, all bits have the same weight, it can be important to balance the variance before quantizing the vectors.

One possible solution is to rotate the projected data. This rotation can be random, such as in PCAE-RR (Section 2.2.2), or learned, such as in PCAE-ITQ (Section 2.2.3). Another option is to assign more bits to the more relevant dimensions. In practice, Spectral Hashing (Section 2.2.4) can be seen as a way to achieve this goal, even though its theoretical foundations are different.

[2] An alternative would be to choose a per-dimension threshold equal to the median of the $r'_k x_i$ values. However, this would not guarantee anymore the convergence to the cosine.

2.2.2 PCAE + Random Rotations (PCAE-RR)

As noted in [32],[22], one simple way to balance the variances is to project the data with the PCA projections and then rotate the result with a random orthogonal matrix $R \in \mathbb{R}^{K \times K}$ (or, equivalently, if we put the column eigenvectors in a matrix $W \in \mathbb{R}^{D \times K}$, to project and rotate the data at the same time with a matrix $\tilde{W} = WR$). One way to generate this random orthogonal matrix is to first create a random matrix drawn from a $\mathcal{N}(0,1)$ distribution and perform a QR decomposition, as done in [34]. Another option is to perform an SVD decomposition of such matrix, as done in [22]. We follow the latter approach.

Therefore, the PCAE-RR embedding can be defined as $h_k(x) = q_k(g_k(x))$, with

$$g_k(x) = \tilde{w}_k'(x - \mu), \quad (10)$$

$$q_k(u) = \sigma(u), \quad (11)$$

where $\tilde{w}_k$ is the k-th column of $\tilde{W}$. Note that rotating the data with an orthogonal matrix after the PCA projection is still an optimal solution of the formulation of [18]. Also, since orthogonal rotations preserve the Euclidean distance – already approximately preserved after PCA – the natural distance in the intermediate space for PCAE-RR is also the Euclidean distance.

2.2.3 PCAE + Iterative Quantization (PCAE-ITQ)

In [22], the idea of rotating the projections to balance the variances after PCA is taken a step further. The goal is to find the optimal orthogonal rotation R that minimizes the quantization loss in a training set:

$$\underset{R}{\text{argmin}} \sum_{x \in \mathcal{S}} ||q(g(x)) - g(x)||^2, \quad (12)$$

with

$$g_k(x) = \tilde{w}_k'(x - \mu), \quad (13)$$

$$q_k(u) = 2\sigma(u) - 1, \quad (14)$$

and, as before, $\tilde{w}_k = (WR)_k$.

Intuitively, the goal is to map the points into the vertices of a binary hypercube. The closer the points are to the vertices, the smaller the quantization error will be. This optimization problem is related to the Orthogonal Procrustes problem [35], in which one tries to find an orthogonal rotation to align one set of points with another. The optimization can be solved iteratively, and involves computing an SVD decomposition of a $K \times K$ matrix at every iteration. A random orthogonal matrix is used as initial values of this matrix. Since we are usually interested in compact codes (*e.g.*, $K \leq 512$), obtaining the orthogonal rotation matrix R is quite fast. Note also that this optimization has to be computed only once, offline.

As in the case of PCAE-RR, we perform an orthogonal rotation after the PCA projection, and so the Euclidean distance is approximately preserved in the intermediate space.

2.2.4 Spectral Hashing (SH)

Given a similarity sim between objects in $\Omega \in \mathbb{R}^D$, and assuming that the distribution of objects in Ω may be described by a probability density function p, SH [15] attempts to minimize the following objective function with respect to h:

$$\int_{x,y} ||h(x) - h(y)||^2 sim(x,y) p(x) p(y) dx dy, \quad (15)$$

subject to a set of constraints. As the constrained problem is NP hard Weiss *et al*. propose to optimize a *relaxed* version of their problem, *i.e.* to remove the constraints and then to binarize the real-valued output at 0. This is equivalent to minimizing:

$$\int_{x,y} ||g(x) - g(y)||^2 sim(x,y) p(x) p(y) dx dy, \quad (16)$$

with respect to g and then writing $h(x) = 2\sigma(g(x)) - 1$, *i.e.* $q(u) = 2\sigma(u) - 1$. The solutions to the relaxed problem are eigenfunctions of the weighted Laplace-Beltrami operators for which there exists a closed-form formula in certain cases, *e.g.* when sim is the Gaussian kernel and p is separable and uniform. To satisfy, at least approximately, the separability condition, PCA is first performed on the input vectors.

When minimizing the SH objective function (16), we learn a function g which enforces points (x,y) which have a large $sim(x,y)$ to have a low Euclidean distance $||g(x) - g(y)||^2$. This shows that the Euclidean distance makes sense in the intermediate space.

3 Asymmetric Distances

In the previous section we decomposed several binary embedding functions h_k into real-valued embedding functions g_k and quantization functions q_k. Let d denote the squared Euclidean distance [3]. We also showed that:

$$d(g(x), g(y)) = \sum_k d(g_k(x), g_k(y)). \quad (17)$$

is a natural distance in the intermediate space. We now propose two approximations of the quantity (17). In the following, x is assumed non-binarized, *i.e.* we have access to the values $g_k(x)$ (and therefore also to $h_k(x)$), while y is binarized, *i.e.* we only have access to the values $h_k(y)$ (but not to $g_k(y)$).

3.1 Expectation-Based Asymmetric Distance

In [32] Jégou *et al*. proposed an asymmetric algorithm for compression schemes based on vector quantization. A codebook is learned through k-means clustering and a database vector is encoded by the index of its closest centroid in the codebook. The distance

[3] We use the following abuse of notation for simplicity: d denotes both the distance between the vectors $g(x)$ and $g(y)$, *i.e.* $d : \mathbb{R}^K \times \mathbb{R}^K \rightarrow \mathbb{R}$, and the distance between the individual dimensions, *i.e.* $d : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$.

between the uncompressed query and a quantized database signature is simply computed as the Euclidean distance between the query and the corresponding centroid.

We now adapt this idea to binary embeddings. We note that in the case of LSH, LSBC, SH, or PCAE we have no notion of centroid. However, we note that the centroid of a given cell in k-means clustering can be interpreted as the expected value of the vectors assigned to this particular cell. Similarly, we propose an asymmetric expectation-based approximation d_E for binary embeddings.

Assuming that the samples in the original space Ω are drawn from a distribution p, we define:

$$d_E(x,y) = \sum_k d(g_k(x), \mathbb{E}_{u\sim p}[g_k(u)|h_k(u)=h_k(y)]). \tag{18}$$

$d(g_k(x), \mathbb{E}_{u\sim p}[g_k(u)|h_k(u)=h_k(y)])$ is the distance in the intermediate space between $g_k(x)$ and the expected value of the samples $g_k(u)$ such that $h_k(u) = h_k(y)$.

Since we generally do not have access to the distribution p, the expectation operator is approximated by a sample average. In practice, we randomly draw a set of signatures $\mathcal{S} = \{x_i, i = 1\dots N\}$ from Ω. For each dimension k of the embedding, we partition $\mathcal{S}$ into two subsets: $\mathcal{S}_k^0$ contains the signatures x_i such that $h_k(x_i) = 0$ and $\mathcal{S}_k^1$ those signatures x_i that satisfy $h_k(x_i) = 1$. We compute *offline* the following *query-independent* values (see Fig. 1):

$$\alpha_k^0 = \frac{1}{|\mathcal{S}_k^0|}\sum_{u\in\mathcal{S}_k^0} g_k(u), \tag{19}$$

$$\alpha_k^1 = \frac{1}{|\mathcal{S}_k^1|}\sum_{u\in\mathcal{S}_k^1} g_k(u). \tag{20}$$

Online, for a given query x, we first pre-compute and store in look-up tables the following *query-dependent* values:

$$\beta_k^0 = d(g_k(x), \alpha_k^0) = (g_k(x) - \alpha_k^0)^2, \tag{21}$$

$$\beta_k^1 = d(g_k(x), \alpha_k^1) = (g_k(x) - \alpha_k^1)^2. \tag{22}$$

By definition, we have:

$$d_E(x,y) = \sum_k \beta_k^{h_k(y)}. \tag{23}$$

The cost of pre-computing the β values is negligible with respect to the cost of computing many $d_E(x,y)$'s for a large number of database signatures y. The sum (23) can be computed very efficently by grouping the dimensions. In our implementation, we subdivide a vector into blocks of 8 dimensions and write (assuming that the number of dimensions K is a multiple of 8, to simplify the notation):

$$d_E(x,y) = \sum_{k=0}^{K/8-1}\sum_{j=1}^{8}\beta_{8k+j}^{h_{8k+j}(y)}. \tag{24}$$

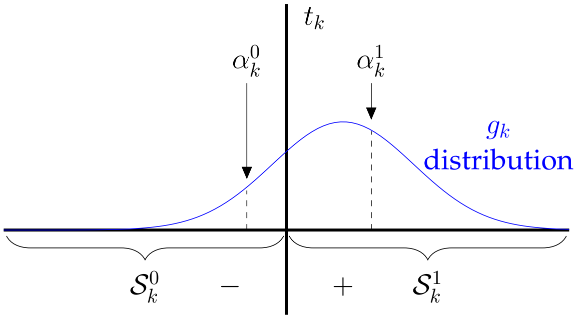


Figure 1. Expectation-based asymmetric distance. Graphical interpretation of the α_k coefficients of equations (19) and (20).

Because the binary subvector $[h_{8k+1}(y),\dots,h_{8k+8}(y)]$ fits in 1 byte, each sum $\sum_{j=1}^{8}\beta_{8k+j}^{h_{8k+j}(y)}$ can only take 256 possible values. We can pre-compute these 256 values and store them in a look-up table. Performing $K/8$ accesses to 256-dimensional look-up tables is faste than performing K accesses to 2-dimensional look-up tables. This also reduces the number of summations performed online by a factor of 8.

3.2 Lower-Bound Based Asymmetric Distance

In [31], Dong *et al.* proposed an asymmetric algorithm for binary embeddings based on random projections. We now show that a similar approach can be applied to a much wider range of binary embedding techniques. For the simplicity of the presentation, we assume that q_k has the form $q_k(u) = \sigma(u - t_k)$ where t_k is a threshold but this can be trivially generalized to other quantization functions.

The idea is to lower-bound the quantity (17) by bounding each of its terms. We note that t_k splits $\mathbb{R}$ into two half-lines and consider two cases:

- If $h_k(x) \neq h_k(y)$, *i.e.* $g_k(x)$ and $g_k(y)$ are on different sides of t_k, then a lower-bound between $g_k(x)$ and $g_k(y)$ is the distance between $g_k(x)$ and the threshold t_k, *i.e.* $d(g_k(x), g_k(y)) \geq d(g_k(x), t_k)$.
- If $h_k(x) = h_k(y)$, *i.e.* $g_k(x)$ and $g_k(y)$ are on the same half-line, then we have the following obvious lower-bound: $d(g_k(x), g_k(y)) \geq 0$ (actually, this bound is always true).

Merging the two cases in a single equation, we have the following lower-bound on $d(g(x), g(y))$:

$$d_{LB}(x,y) = \sum_k \bar{\delta}_{h_k(x),h_k(y)} d(g_k(x), t_k), \tag{25}$$

where $\bar{\delta}_{i,j}$ is the negation of the Kronecker delta, *i.e.* $\bar{\delta}_{i,j} = 0$ if $i = j$ and 1 otherwise. We note that, in the case of LSH, equation (25) is equivalent to the asymmetric LSH distance proposed in [31].

In practice, for a given query signature x, we can

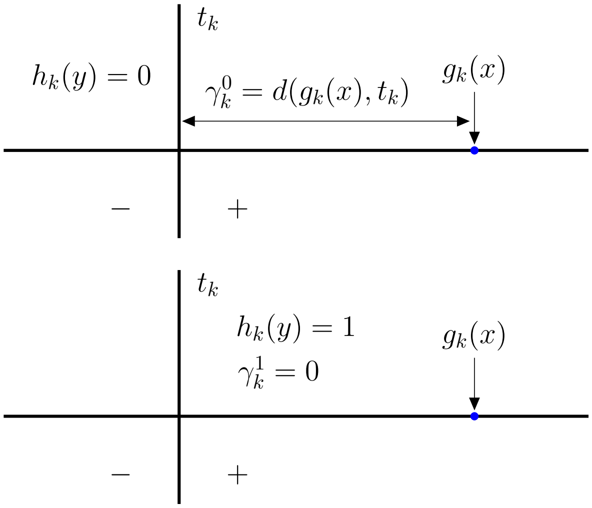


Figure 2. Lower-bound-based asymmetric distance. Top: case where $h_k(x) \neq h_k(y)$, and therefore $d(g_k(x), g_k(y)) \geq d(g_k(x), t_k)$. Bottom: case where $h_k(x) = h_k(y)$, and therefore $d(g_k(x), g_k(y)) \geq 0$.

pre-compute *online* the values (see Fig. 2):

$$\gamma_k^0 = \bar{\delta}_{h_k(x),0} d(g_k(x), t_k) = \bar{\delta}_{h_k(x),0} (g_k(x) - t_k)^2, \quad (26)$$

$$\gamma_k^1 = \bar{\delta}_{h_k(x),1} d(g_k(x), t_k) = \bar{\delta}_{h_k(x),1} (g_k(x) - t_k)^2. \quad (27)$$

We note that one of these two values is guaranteed to be 0 for each dimension k. By definition we have:

$$d_{LB}(x, y) = \sum_k \gamma_k^{h_k(y)}. \quad (28)$$

We can subsequently pack these values by blocks of 8 dimensions for faster computation as was the case of the expectation-based approximation.

A major difference between d_E and d_{LB} is that the former one makes use of the data distribution while the latter one does not. Despite its very crude nature, we will see that d_{LB} leads to excellent results on a variety of binary embedding algorithms.

3.3 Variance Preservation

In what follows, we make the assumption that the distribution of the real-valued embedding in dimension i (*i.e.* the distribution of $g_i(x)$)) is a Gaussian p_i with mean 0 and variance σ_i^2. We assume that the dimensions are ordered such that $\sigma_i > \sigma_{i+1}$. According to information theory, the amount of information carried by dimension i, which can be measured by the entropy of p_i, is proportional to $\log(\sigma_i)$. Therefore, a natural strategy in source coding is to allocate to each dimension a number of bits which is proportional to this number. Two strategies have been proposed for this purpose: either to allocate a variable number of bits per dimension [17] or to equalize the variance [34], [22] and to keep the same number of bits to each dimension (typically 1 bit per dimension). If we do not equalize the variance and allocate a fixed number of bits per dimension, then less informative dimensions are given the same weight as more informative dimensions when computing a Hamming distance which leads to suboptimal results. However, asymmetric distances do not suffer from this problem. Indeed, we now show that the asymmetric distance in each dimension is proportional in expectation to the variance of the data, thus showing that more weight is given to more informative dimensions.

a) d_E case: In a given dimension i, the expectation of the positive samples is the expectation of a half-normal distribution

$$E = \int_0^\infty x p_i(x) dx = \frac{\sigma_i}{\sqrt{2\pi}}$$

and the expectation of the negative samples is $-E = -\frac{\sigma_i}{\sqrt{2\pi}}$. These values correspond to the α's of equations (19) and (20). Consequently, the expectation of the distance to these values (i.e. the expectation of the β's of equations (21) and (22)) is:

$$\int_{-\infty}^{+\infty} (x - (\pm E))^2 p_i(x) dx = \sigma_i^2 (1 + \frac{1}{2\pi}). \quad (29)$$

Therefore *in expectation* the contribution of a given dimension to the asymmetric distance is proportional to the variance in this dimension.

b) d_{LB} case: Let us consider the case where $h_k(x) = 1$ and $h_k(y) = 0$. The other relevant case can be treated analogously. The expectations of the γ's (*cf*. equations (26) and (27)) in dimension i are equal to

$$\int_{-\infty}^{0} 0 \; p_i(x) dx + \int_0^{+\infty} x^2 p_i(x) dx = \frac{\sigma_i^2}{2}.$$

Again, *in expectation*, the contributions of the dimensions to the asymmetric distance decrease with the index i.

As mentioned earlier, this analysis is only valid in the case where the distribution of the real valued embeddings is Gaussian in each dimension and has mean zero. This assumption is reasonable in the case of PCAE since the assumption underlying PCA is that the data distribution is Gaussian and since after projection the data is zero-centered (see Figure 3 top). This may explain, at least partly, the significant improvements of asymmetric distances for PCAE (see, *e.g.*, the results on CIFAR in Figures 4c-6c). On the other hand, the Gaussian assumption does not seem to be valid for LSBC (see Figure 3 bottom) because of the cosine non-linearity.

4 Experiments

We now show the benefits of the asymmetric distances proposed in section 3 on the binary embeddings we reviewed in section 2. We first describe in Section 4.1 the four public benchmarks we experiment on. We then provide in Section 4.2 implementation details for the different embedding algorithms. We finally report and discuss results in Section 4.3.

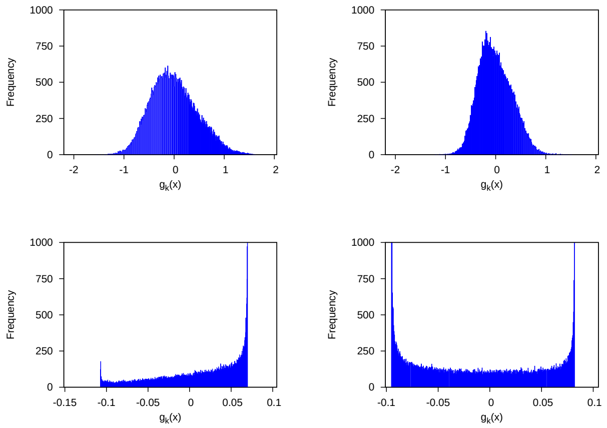


Figure 3. Histograms of the projected values of the 60,000 CIFAR images. Top: projected on the first two PCA dimensions. Bottom: projected with two random LSBC dimensions.

4.1 Datasets and Features

We run experiments on two category-level retrieval benchmarks, CIFAR and Caltech256, and two instance-level retrieval benchmarks, the University of Kentucky Benchmark (UKB) and INRIA Holidays. This diverse selection of datasets allows us to experiment with different setups. On CIFAR, we evaluate both semantic retrieval and retrieval using Euclidean neighbors as ground truth. On Caltech256, we evaluate the effect of different descriptors on the results: GIST, Bag of Words and Fisher Vector. UKB and Holidays are standard instance-level retrieval datasets. As opposed to CIFAR or Caltech256, the number of relevant items per query is much smaller, always 4 for UKB and from 2 to 10 for Holidays. We now describe these datasets as well as the associated standard experimental protocols in detail.

CIFAR. The CIFAR dataset [36] is a subset of the Tiny Images dataset [5]. We use the same version of CIFAR that was used in [22]. It contains 64,184 images of size 32×32 that have been manually grouped into 11 ground truth classes. Images are described using a greyscale GIST descriptors [6] computed at 3 different scales (8, 8, 4) producing a 320-dim vector. 1,000 images are used as queries, 5,000 are used for unsupervised training purposes, and the remaining images are use as database images.

We report results on two different problems.

- Euclidean neighbor retrieval: we discard the class labels and measure the preservation of Euclidean neighbors. We used two types of grountd truth. In the first case, we used as ground truth neighbors for a given point all those other points which are within an ϵ-ball. Following [22], we use as threshold ϵ the average distance to the 50th nearest neighbor. We note that the number of true positives varies widely from one query to another (from 0 to 2,353). We compute the Average Precision (AP) for each query (with a non-zero number of true positives) and report the mean over the 1,000 queries (Mean AP or MAP). In what follows, these experiments are referred to as "Euclidean ϵ-NN". In the second case, we used as ground truth neighbors the k nearest neighbors of each query, *i.e.*, all queries have the same number of true positives. For each query, we compute the Normalized Discounted Cumulative Gain at k (NDCG@k) and report the average over the queries. The relevance of each true neighbor decreases linearly from 1 (the closest neighbor) to $1/k$ (the k-th, less relevant neigbor). We experimented with different values of k and did not observe any significant difference in the shapes of the plots. Therefore, we only report results for $k = 1,000$ neighbors. In what follows, these experiments are referred to as "Euclidean k-NN".
- Semantic retrieval: we use the class labels as ground truth and report the precision at 1.

Caltech256. The Caltech256 dataset [37] contains approximately 30,000 images grouped in 257 classes. Through our experiments, we use only 256 classes and we discard the "clutter" class. As in CIFAR, we split the dataset in three different sets. We select 5 images per class (1,280 images in total) to serve as queries, and 5,000 random images to serve as unsupervised training data. The remaining images are used as the database. We describe the images and report precision at 1 using 3 different descriptors:

- GIST descriptors with 320 dimensions (same configuration as in CIFAR).
- Bag of Visual Words (BOV) [7], [8] with 1,024 visual words on SIFT descriptors [38].
- Fisher Vectors (FV) [9], [10], which were shown to yield excellent results for object and scene retrieval [39], [11]. We compute 4,096-dim FVs on SIFT descriptors as is the case of the BOV.

UKB. The University of Kentucky Benchmark (UKB) [40] contains 10,200 images of 2,550 objects (4 images per object). Each image is used in turn as query to search through the 10,200 images. The accuracy is measured in terms of the number of relevant images retrieved in the top 4, *i.e.* 4 × recall@4. We use the same low-level feature detection and description procedure as in [34]. As in Caltech256, we use 4,096-dim FV representations. For all learning purposes (*e.g.* to learn the visual vocabulary for the FV), we use an additional set of 60,000 images (Flickr60K) made available by the authors of [34].

Holidays. The INRIA Holidays dataset [34] contains 1,491 images of 500 scenes and objects. The first image of each scene is used as query to search through the remaining 1,490 images. We measure the accuracy for each query using AP and report the MAP over the 500 queries. As was the case for UKB, images are described using 4,096-dim FVs, using the same low-level feature detection and description procedure, as well as the same Flickr learning set.

4.2 Implementation Details

To learn the parameters of the embedding functions and to compute offline the α values for d_E we need training data. For CIFAR and Caltech256, we will use the $5,000$ training samples, that are used neither as queries nor as database items. For UKB and Holidays we use the Flickr60K dataset [34].

For CIFAR and Caltech256, where no predefined partitions exist, we repeated the experiments 3 times using different queries and database partitions and averaged the results.

For LSH and LSBC, which perform binarization through random projections, as well as for PCAE-RR and PCAE-ITQ, which use random rotations, experiments are repeated 5 times with 5 different projection matrices and we report the average results.

As discussed in section 2.1.1, mean-centering the data can impact LSH positively. Therefore, we have mean-centered the GIST and BOV descriptors for CIFAR and Caltech256, learning the means on their respective training sets. By definition FVs are already (approximately) mean-centered. We note that centering the data on the origin does not impact PCAE, PCAE-RR, PCAE-ITQ, and SH (which perform PCA of the signatures) or LSBC (which is shift-invariant).

4.3 Results and Analysis

We report results on the four datasets in Figures 4 - 11 with the symmetric Hamming distance as well as with the proposed asymmetric distances d_E and d_{LB}. The following is a detailed discussion of our findings.

Asymmetric vs symmetric. Asymmetric distances consistently improve the results over the symmetric Hamming distance, independently of the dataset, of the descriptor used, and of the binary embedding technique. In general, the gain in accuracy is impressive both in terms of absolute and relative improvement. Here are just two examples: on CIFAR with semantic labels (Figure 6), when using PCAE, we observe an improvement of 8% absolute and 22% relative at 128 bits. On Holidays, when using SH, we can observe an improvement of 8% absolute and 21% relative (Figure 11), also at 128 bits.

The GIST results on Caltech256 (Figure 7) are an exception to this rule, especially for those compression algorithms which involve PCA such as PCAE and SH. Indeed, in this setting the simpler Hamming distance can outperform the proposed asymmetric distances. This seems to indicate that the Euclidean distance we are trying to approximate in the PCA space is suboptimal (at least on this dataset and with these features) and that the Hamming distance in the projected space approximates a better metric.

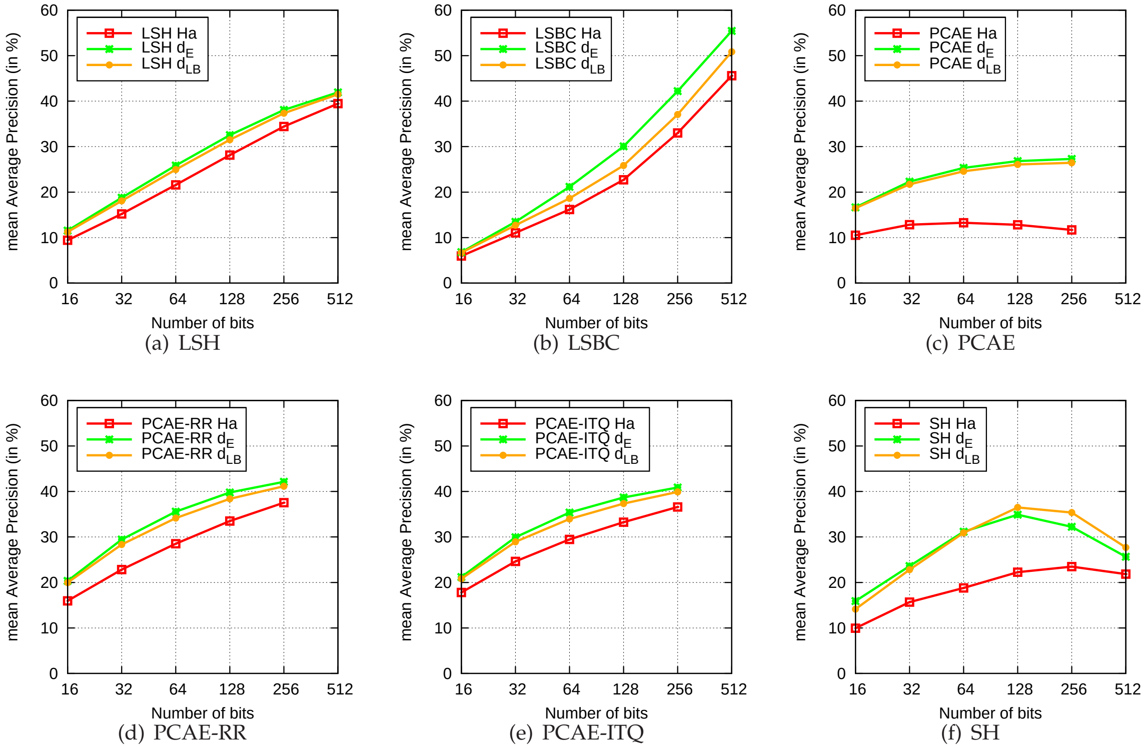
To verify this hypothesis, we experimented with the cosine distance in the PCA space – which is equivalent to the Euclidean distance on ℓ_2-normalized vectors – and obtained better results than with the Euclidean distance. For instance, without any additional binarization, we obtained 12.9% accuracy when using the Euclidean distance in a 256-dim PCA subspace and 14.0% with a cosine distance. This seems to show that ℓ_2-normalizing PCA-projected GIST descriptors leads to improved results. To provide a tentative explanation of the superiotiy of the cosine over the Euclidean distance for PCA-projected GIST descriptors, we refer to some the arguments of [41]. Indeed, Jégou and Chum argue that, when comparing BOV vectors, the presence as well as the absence of a visual word can be informative and that the cosine in the PCA space takes into account the absence of the visual words while the Euclidean distance does not. Similarly, our results seem to show that the absence of a gradient direction in the GIST can be informative. Having shown that, on this dataset, ℓ_2-normalizing PCA-projected GIST descriptors leads to superior results, it is not surprising that the Hamming distance can outperform the asymmetric distances. Indeed, the asymmetric distances attempt to approximate the (inferior) Euclidean distance. On the other hand, the Hamming distance can be related to the Euclidean distance between binary vectors encoded on $\{-1, +1\}$, and since those vectors all have the same ℓ_2-norm, it can also be related to the (superior) cosine distance.

Expectation vs lower-bound. For almost all embedding techniques, d_E and d_{LB} yield very similar results, which is somewhat surprising given that the two approximations are very different in nature. The slight advantage of d_E over d_{LB} comes from the fact that the former approach uses information about the data distribution (through the pre-computed values α) while the latter does not.

We note, however, two exceptions. The first is LSBC, for which in most cases d_E performs significantly better than d_{LB} (see Figures 4b, 5b, 6b, 8b, 9b, 10b, 11b). We are still investigating this difference but we observed that the distributions of the values $g_k(x)$ in the intermediate real-valued space for LSBC are significantly different from those observed for the other embedding methods (typically U- or half-U-shaped for LSBC, as opposed to Gaussian-shaped for the others, particularly PCAE, see Figure 3). The second exception is on the Caltech256 with GIST descriptors on Figure 7, where d_{LB} usually and sometimes very clearly outperforms d_E. As we noted in the previous point, we believe this is because the cosine distance is a better measure of similarity than the Euclidean distance when PCA-projected GIST descriptors.

Finally, preliminary experiments on fusing d_E and d_{LB} yielded only marginal improvements.

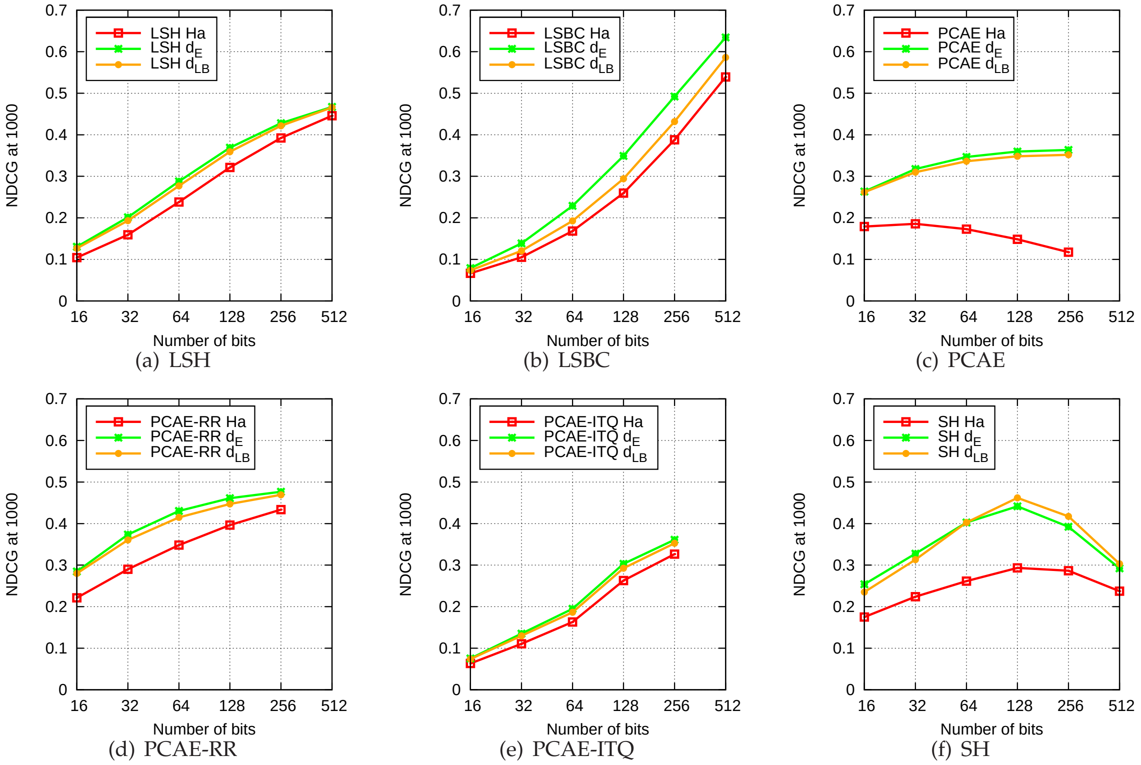
Influence of the retrieval problem. In Figures 4-6 we report results on CIFAR for two different prob-



(a) LSH (b) LSBC (c) PCAE

(d) PCAE-RR (e) PCAE-ITQ (f) SH

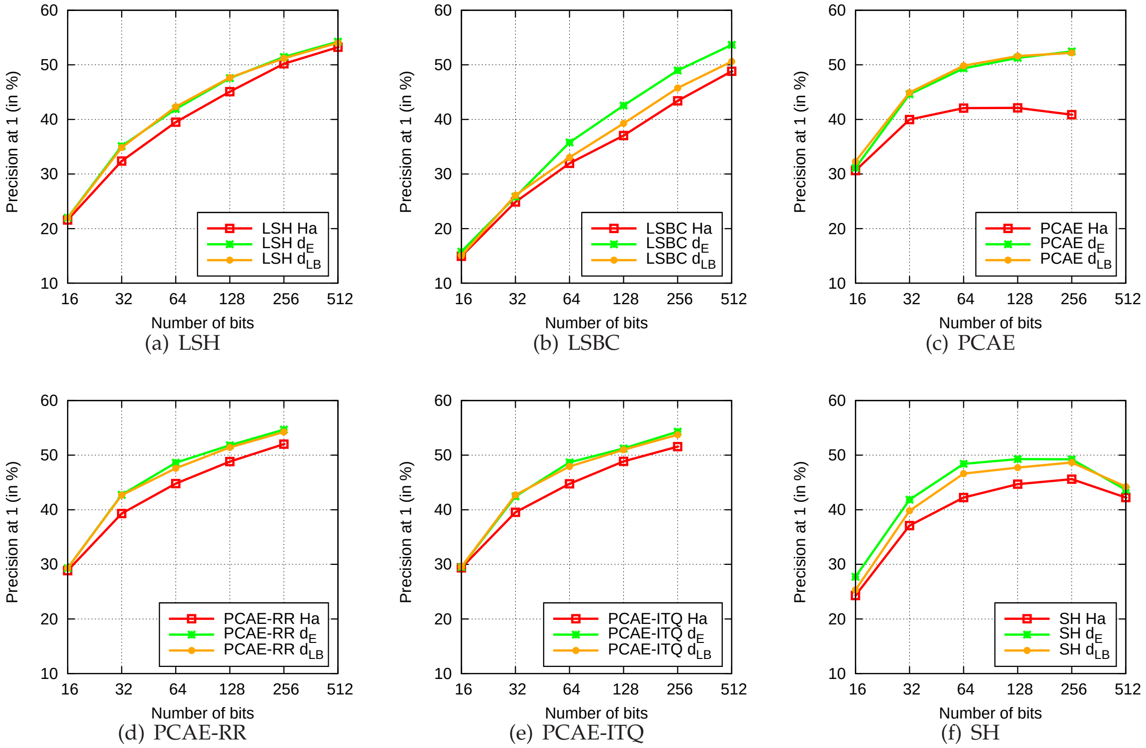
Figure 4. Influence of the asymmetric distances on the CIFAR dataset with Euclidean ϵ-NN. The dimensionality of PCAE, PCAE-RR and PCAE-ITQ is limited by the dimensionality of the original GIST descriptor, 320 dimensions.



(a) LSH (b) LSBC (c) PCAE

(d) PCAE-RR (e) PCAE-ITQ (f) SH

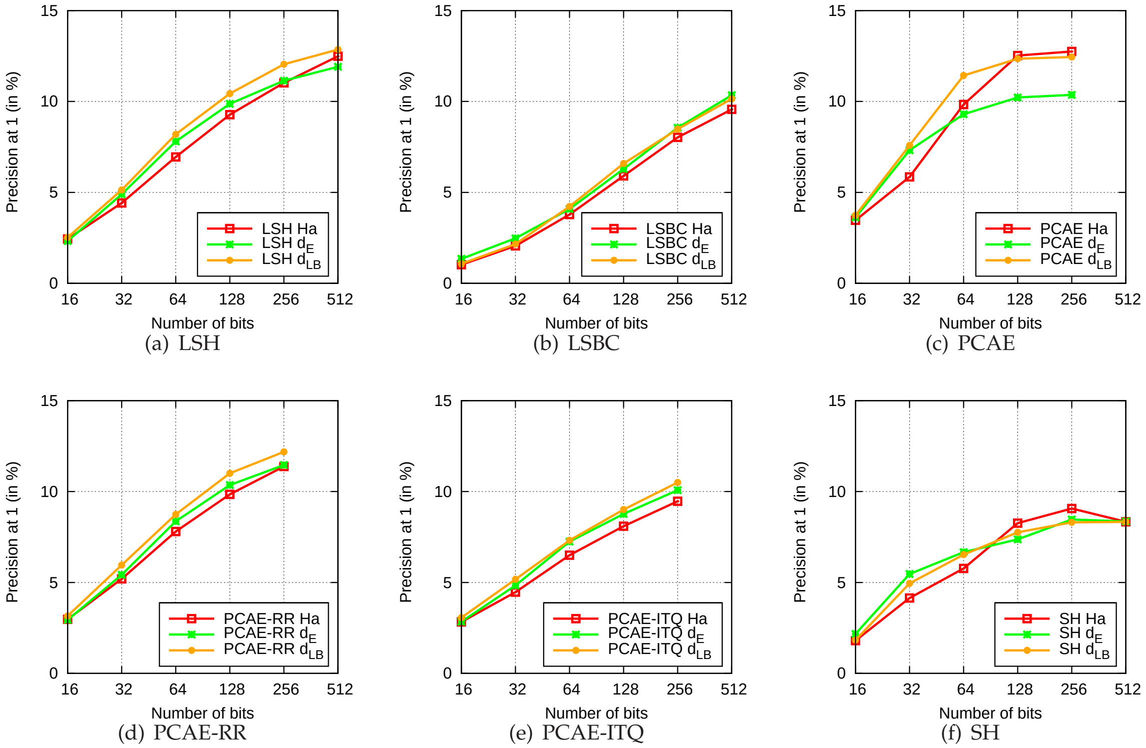
Figure 5. Influence of the asymmetric distances on the CIFAR dataset with Euclidean k-NN with $k = 1,000$. The dimensionality of PCAE, PCAE-RR and PCAE-ITQ is limited by the dimensionality of the original GIST descriptor, 320 dimensions.



(a) LSH (b) LSBC (c) PCAE

(d) PCAE-RR (e) PCAE-ITQ (f) SH

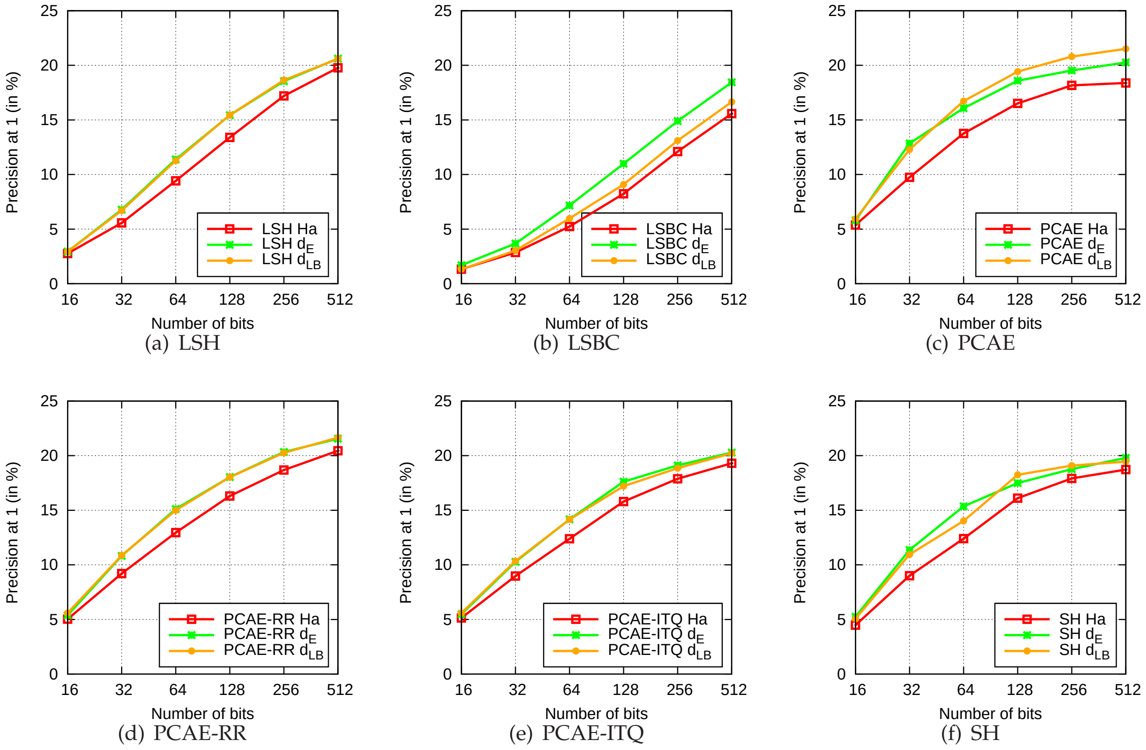
Figure 6. Influence of the asymmetric distances on the CIFAR dataset with semantic labels for 6 different encoding methods: LSH, LSBC, PCAE, PCAE-RR, PCAE-ITQ, and SH. The dimensionality of PCAE, PCAE-RR and PCAE-ITQ is limited by the dimensionality of the original GIST descriptor, 320 dimensions.



(a) LSH (b) LSBC (c) PCAE

(d) PCAE-RR (e) PCAE-ITQ (f) SH

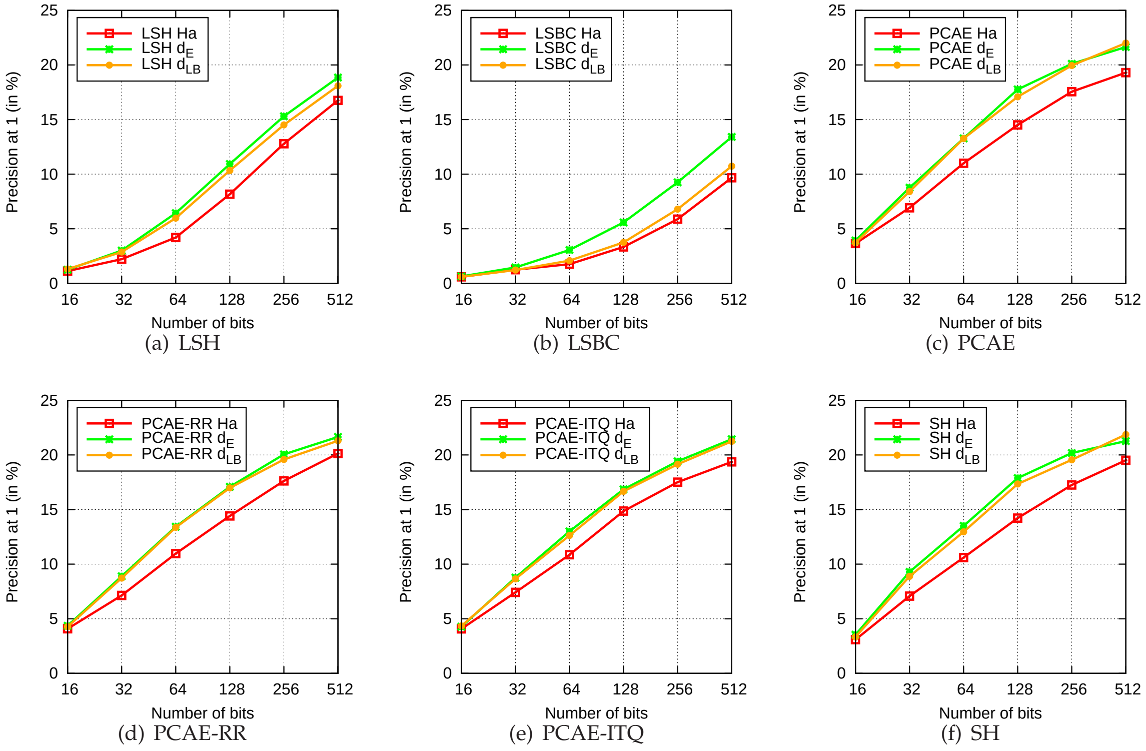
Figure 7. Influence of the asymmetric distances on the CALTECH256 dataset with GIST descriptors.



(a) LSH (b) LSBC (c) PCAE

(d) PCAE-RR (e) PCAE-ITQ (f) SH

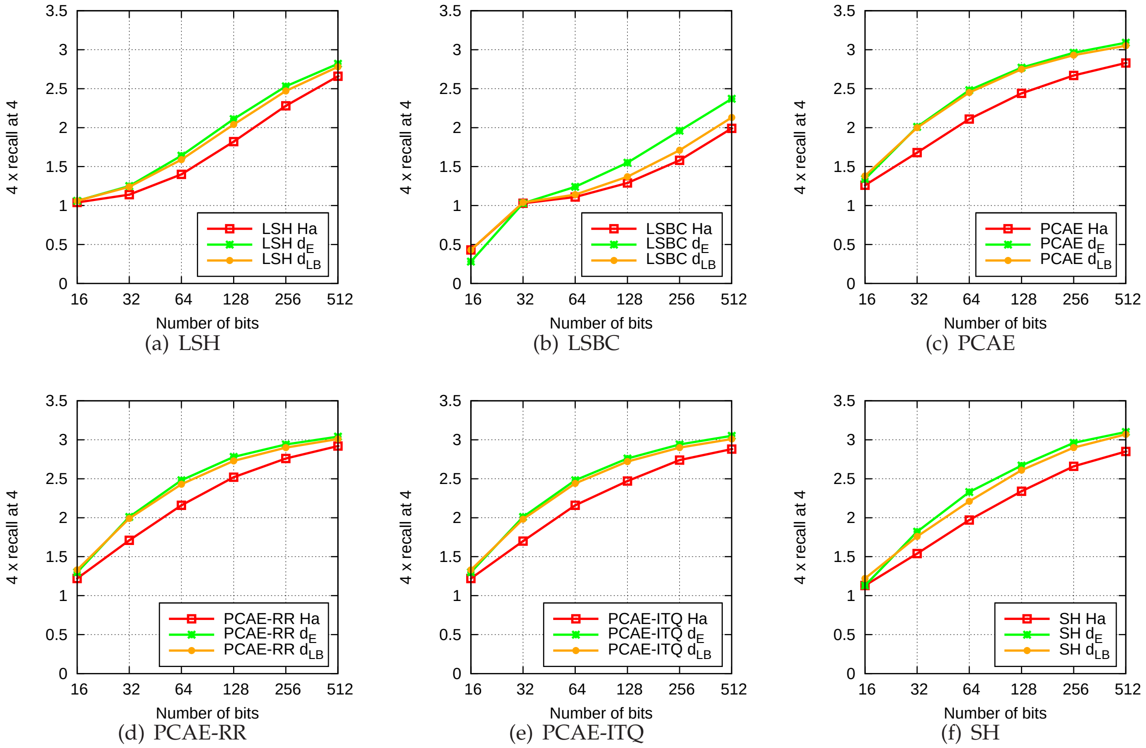
Figure 8. Influence of the asymmetric distances on the CALTECH256 dataset using BOV descriptors.



(a) LSH (b) LSBC (c) PCAE

(d) PCAE-RR (e) PCAE-ITQ (f) SH

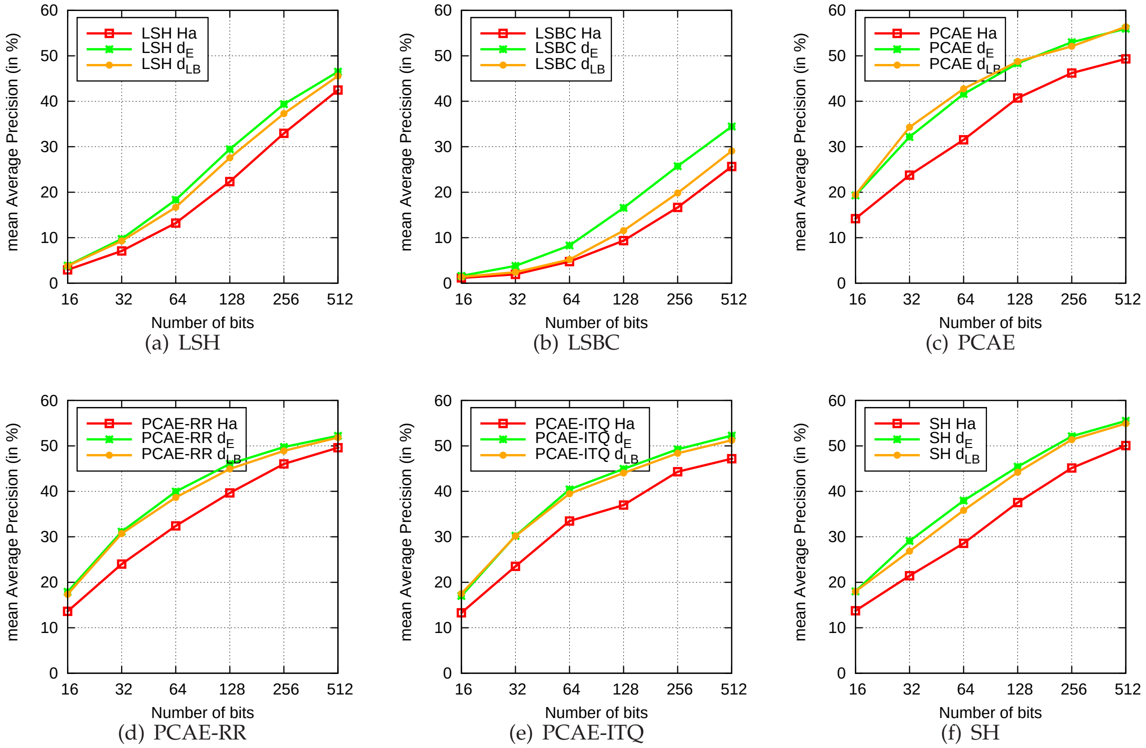
Figure 9. Influence of the asymmetric distances on the CALTECH256 dataset using FV descriptors.



(a) LSH (b) LSBC (c) PCAE

(d) PCAE-RR (e) PCAE-ITQ (f) SH

Figure 10. Influence of the asymmetric distances on the UKB dataset.



(a) LSH (b) LSBC (c) PCAE

(d) PCAE-RR (e) PCAE-ITQ (f) SH

Figure 11. Influence of the asymmetric distances on the Holidays dataset.

lems: retrieval of Euclidean neighbors and semantic retrieval. We can observe how asymmetric distances provide similar improvements for both cases. We can also note how, in the Euclidean problem, PCAE with Hamming distance seems to perform poorly (Fig. 4c) compared to the results of PCAE on other datasets, and also how the asymmetric improvements seem larger in this case. We believe the difference stems not from the problem (semantic vs Euclidean retrieval) but because of the evaluation measure: this is the only experiment that combines, at the same time, a large number of relevant items per query and a "global" measure such as mAP or NDCG@1000. In such a case, balancing the data with PCAE-RR or PCAE-ITQ or using asymmetric distances seems to yield a large benefit. To attest this, we experimented on Caltech256, which has many relevant items per query. Using BOV descriptors, we computed the mAP score instead of the precision at 1 reported in Figure 8c. In that case, PCAE results drastically dropped below those of PCAE-RR and PCAE-ITQ, supporting this idea.

Influence of the descriptor. In Figures 7 to 9 we can observe the influence of the descriptors on Caltech256. In general, the improvements in BOV and FV are larger than the improvements obtained with GIST, showing that the improvement can be dependent on the feature type, particularly if the Euclidean distance was not a good measure in the original space.

We can also observe how, particularly in the PCA-based methods, FV has a slight edge over BOV when aiming at 256 bits or more. However, BOV can obtain better results than the FV when aiming at signatures of 128 bits or less. This is in line with the observations of [11], where they notice that, when producing small codes, it is usually better to start with a smaller image signature. In our case, the BOV has 1,024 dimensions and the FV has 4,096. When we can afford larger codes, the FV usually still outperforms BOV: the uncompressed BOV baseline is 22.11%, while the uncompressed FV baseline is 24.11%.

Influence of the embedding method. All methods benefit significantly from the asymmetric distances. This can be easily understood: since we are not binarizing the query, there is less loss of information on the query side.

PCAE seems to benefit particularly from the asymmetric distances (see, *e.g.*, the results on CIFAR in Figures 4c-6c). This may be explained by the variance-preservation effect of the asymmetric distances (see Section 3.3). The variance problem of the other methods is not so severe: LSH and LSBC use random projections, and their variances are balanced in expectation. PCAE-RR, PCAE-ITQ, and SH all balance the variances, either explicitly as in PCAE-RR and PCAE-ITQ, or implicitly, as in SH, assigning more bits to the more important dimensions. Therefore, the impact of the asymmetric distances on these methods is not as pronounced, yet still sizeable. For example, on Caltech256 with FV (Figure 9), we show improvements on LSBC of about 4% absolute but almost 40% relative.

Asymmetric distances also seem to bridge the gap between the binary encoding methods, particularly between those based on PCA. Figure 12 compares the different encoding methods using Hamming distances and both d_E and d_{LB} on the CIFAR semantic problem with the same data we used for Figure 6. The results suggest that asymmetric distances can be used to compensate for the quality of the embedding method; the difference between the encoding methods is significant when using the Hamming distance (more than 10% absolute at 256 bits), but much less pronounced when using asymmetric distances (less than 5% absolute, again at 256 bits).

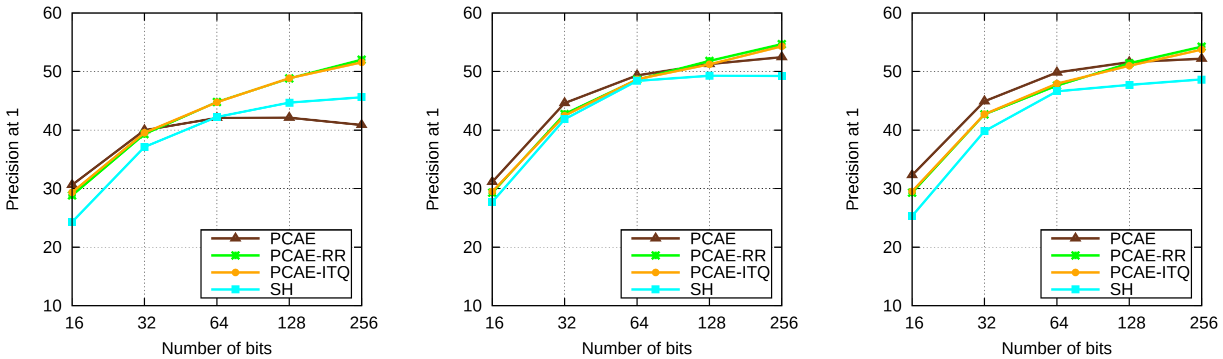
Qualitative results. Finally, Figure 13 shows qualitative results. We show the top 5 ranked images for four random queries of Holidays using PCAE with 128 bits, both for Hamming and for asymmetric distances. The false positives have been framed in red. We can observe how, in general, asymmetric distances obtain better and more consistent results than the Hamming distance.

4.4 Large-Scale Experiments

We now show that the good results achieved with asymmetric distances scale to large datasets. For these experiments we merge Holidays and UKB with a set of 1M Flickr distractors made available by the authors of [34]. We refer to these combined datasets as Holidays+1M and UKB+1M. In both cases we use the original queries, 500 in Holidays and 10,200 in UKB. We experiment with PCAE, since this method obtained results which are competitive with PCAE-RR and PCAE-ITQ while being simpler.

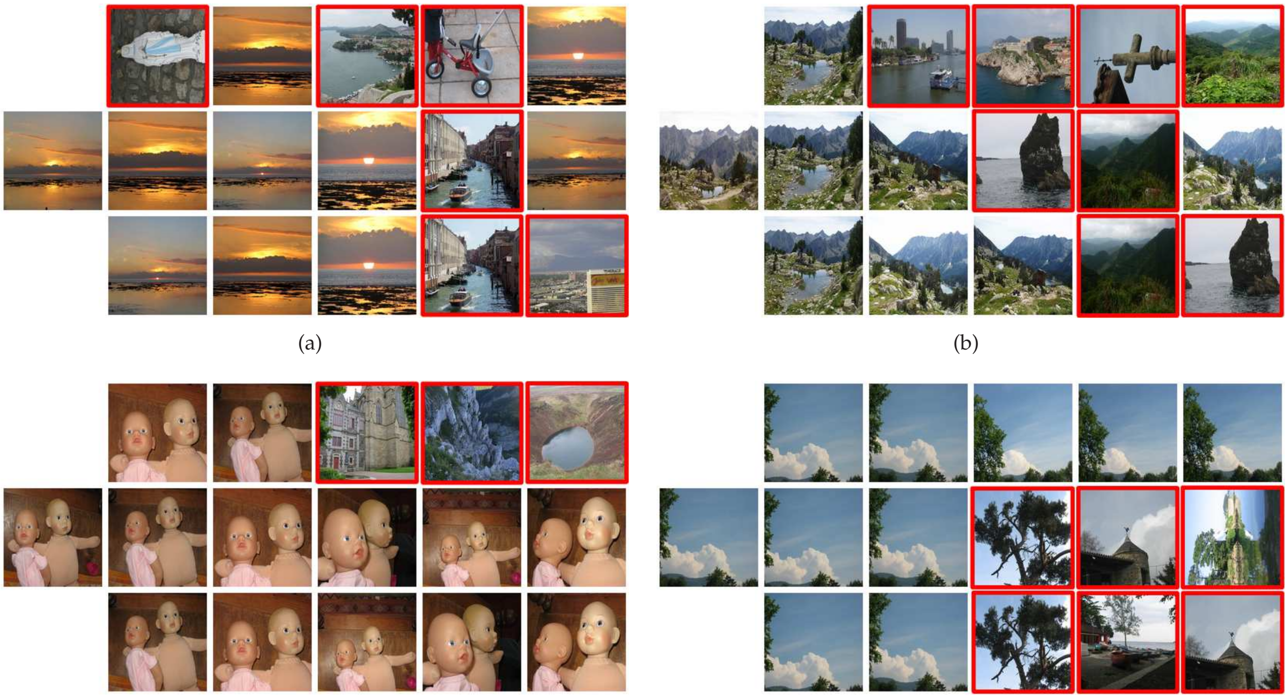
Figure 14 shows the results on both datasets as a function of the number of bits. We compare them with Product quantization (PQ) [32] [11], since, to the best of our knowledge, these are the best results reported on Holidays+1M for very small operating points. For PQ, we employ the same pipeline as [11] which is composed of the following steps: i) PCA compression of the signatures, ii) random orthogonal rotation of the PCA projected signatures, iii) Product quantization and iv) comparison using PQ's asymmetric distances, referred to as ADC in [11].

To fix the number of dimensions D' in the PCA projection step of PQ, the authors minimized the mean square error of the projection and the quantization over a training set. We follow a different heuristic: we set D' to be equal to the number of output bits we are aiming at, and assign 8 dimensions to each subquantizer. We then fix the number of bits per subquantizer to 8, since this seems to be a standard choice that usually offers excellent results. Experimentally, we observed this heuristic to obtain comparable or better



(a) Hamming distance (b) Asymmetric d_E distance (c) Asymmetric d_{LB} distance

Figure 12. Comparison of Hamming and asymmetric distances on CIFAR with semantic labels. Same data as in Figure 6.



(a) (b) (c) (d)

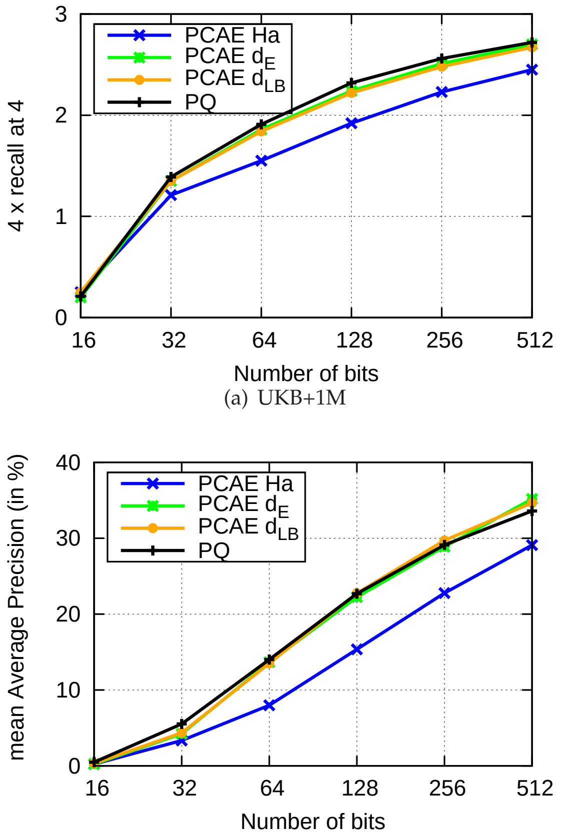
Figure 13. Top five results of four random queries of Holidays using codes of 128 bits. First row: PCAE + Hamming. Middle row: PCAE + asymmetric d_E distance. Bottom row: PCAE + asymmetric d_{LB} distance.

results than minimizing the mean square error, and in most cases was the best possible configuration. As was the case before, experiments are repeated 5 times with different projection matrices and the results are averaged.

We can observe how both asymmetric distances with PCAE perform comparably to PQ on both datasets although PCAE is simpler than PQ both from a conceptual and an engineering standpoint. Furthermore, as opposed to PQ, the lower-bound asymmetric distance does not require any training.

4.5 Timing

We now compare the computational costs of the symmetric Hamming distance and the proposed asymmetric distances. In both cases, we used look-up table implementations. Our experiments were run on 128-bit signatures. Our non-optimized C++ code was run on a single CPU of a machine with a 6-core 8439 SE AMD Opteron processor of 2.8GHz and 64GB of RAM. The cost of computing 1M Hamming distances was approximately 30ms. As for the cost of computing asymmetric distances, for a given query it can be split into the cost of computing a query-dependent look-up table (which is independent of the number of computed distances) plus the cost of computing



(a) UKB+1M

(b) Holidays+1M

Figure 14. Comparison of the proposed asymmetric distances and PQ [11] on UKB+1M (top) and Holidays+1M (bottom). MAP and 4 × recall at 4 as a function of the number of bits.

the distances. For both the expectation-based and lower-bound-based asymmetric distances, the cost of computing the look-up-tables was on the order of 0.06ms while the cost of 1M comparisons was approximately 30ms. Hence, we can conclude from these experiments that, when the query must be compared to 1M dataset items, the look-up table precomputation can be neglected (0.2% of the total time) and both symmetric and asymmetric distances have a similar cost. As the number of comparisons increases, the cost of the look-up table pre-computation becomes even more negligible.

5 Conclusions and Future Work

In this work, we proposed two asymmetric distances for binary embedding techniques, *i.e.* distances between binarized and non-binarized signatures. We showed their applicability to several embedding algorithms: LSH, LSBC, SH, PCAE, PCAE-RR, and PCAE-ITQ. We demonstrated on four datasets with up to 1M images that the proposed asymmetric distances consistently, and often very significantly, improve the retrieval accuracy over the symmetric Hamming distance. We also showed how this asymmetric distances can achieve results comparable to state-of-the-art methods such as PQ, while being conceptually much simpler. The lower-bound asymmetric distance can also be applied on datasets that have already been binarized, with no need to perform any reencoding or extra training.

In future work, we would be interested in investigating coding techniques which would be designed with the asymmetric distances in mind. One possible example which was inspired to us by ITQ would be to learn a rotation matrix which minimizes the quantization error not between the signatures and the binary codes, but between the signatures and the *reconstructed* version of the binary codes using the expected values.

Also, in the asymmetric expectation-based approach, we currently learn the α coefficients (Equations (19) and (20)) which minimize a reconstruction error on the training data. It would be interesting to understand whether we could learn these coefficients with supervised data to optimize directly the retrieval objective function.

Acknowledgements

A. Gordo is partially supported by the Spanish projects TIN2009-14633-C03-03 and CONSOLIDER-INGENIO 2010 (CSD2007-00018). Y. Gong and S. Lazebnik are partially supported by NSF CAREER Award IIS 1228082, Microsoft Research Faculty Fellowship, Xerox, and the DARPA Computer Science Study Group.

References

[1] A. Gordo and F. Perronnin, “Asymmetric distances for binary embeddings,” in *CVPR*, 2011.

[2] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *IJCV*, 2010.

[3] L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,” *IEEE PAMI*, vol. 28, no. 4, 2006.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *CVPR*, 2009.

[5] A. Torralba, R. Fergus, and W. Freeman, “80 million tiny images: a large dataset for non-parametric object and scene recognition,” *IEEE PAMI*, 2008.

[6] A. Oliva and A. Torralba, “Modeling the shape of the scene: a holistic representation of the spatial envelope,” *IJCV*, 2001.

[7] J. Sivic and A. Zisserman, “Video google: A text retrieval approach to object matching in videos,” in *ICCV*, 2003.

[8] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” in *ECCV SLCV Workshop*, 2004.

[9] F. Perronnin and C. Dance, “Fisher kernels on visual vocabularies for image categorization,” in *CVPR*, 2007.

[10] F. Perronnin, J. Sánchez, and T. Mensink, “Improving the Fisher kernel for large-scale image classification,” in *ECCV*, 2010.

[11] H. Jégou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” in *CVPR*, 2010.

[12] P. Indyk and R. Motwani, “Approximate nearest neighbors: towards removing the curse of dimensionality,” in *STOC*, 1998.

[13] M. Charikar, “Similarity estimation techniques from rounding algorithms,” in *STOC*, 2002.

[14] M. Raginsky and S. Lazebnik, “Locality-sensitive binary codes from shift-invariant kernels,” in *NIPS*, 2009.

[15] Y. Weiss, A. Torralba, and R. Fergus, “Spectral hashing,” in *NIPS*, 2008.

[16] B. Kulis and K. Grauman, “Kernelized locality-sensitive hashing for scalable image search,” in *ICCV*, 2009.

[17] J. Brandt, “Transform coding for fast approximate nearest neighbor search in high dimensions,” in *CVPR*, 2010.

[18] J. Wang, S. Kumar, and S.-F. Chang, “Semi-supervised hashing for large scale search,” in *CVPR*, 2010.

[19] L. Torresani, M. Szummer, and A. Fitzgibbon, “Efficient object category recognition using classemes,” in *ECCV*, 2010.

[20] A. Bergamo, L. Torresani, and A. Fitzgibbon, “Picodes: Learning a compact code for novel-category recognition,” in *NIPS*, 2011.

[21] S. Korman and S. Avidan, “Coherency sensitive hashing,” in *ICCV*, 2011.

[22] Y. Gong and S. Lazebnik, “Iterative quantization: A procrustean approach to learning binary codes,” in *CVPR*, 2011.

[23] M. Norouzi and D. Fleet, “Minimal loss hashing for compact binary codes,” in *ICML*, 2011.

[24] J. Wang, S. Kumar, and S. Chang, “Sequential projection learning for hashing with compact codes,” in *ICML*, 2011.

[25] J.-P. Heo, Y. Lee, J. He, S.-F. Chang, and S. eui Yoon., “Spherical hashing,” in *CVPR*, 2012.

[26] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, “LDAHash: Improved matching with smaller descriptors,” *IEEE TPAMI*, 2012.

[27] M. Norouzi, R. Salakhutdinov, and D. Fleet, “Hamming distance metric learning,” in *NIPS*, 2012.

[28] Y. Gong, S. Kumar, V. Verma, and S. Lazebnik, “Angular quantization based binary codes for fast similarity search,” in *NIPS*, 2012.

[29] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, “Supervised hashing with kernels,” in *CVPR*, 2012.

[30] Y. Weiss, R. Fergus, and A. Torralba, “Multidimensional spectral hashing,” in *ECCV*, 2012.

[31] W. Dong, M. Charikar, and K. Li, “Asymmetric distance estimation with sketches for similarity search in high-dimensional spaces,” in *SIGIR*, 2008.

[32] H. Jégou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” *IEEE TPAMI*, 2010.

[33] A. Rahimi and B. Recht, “Random features for large-scale kernel machines,” in *NIPS*, 2007.

[34] H. Jégou, M. Douze, and C. Schmid, “Hamming embedding and weak geometric consistency for large scale image search,” in *ECCV*, 2008. http://lear.inrialpes.fr/˜jegou/data.php.

[35] P. Schonemann, “A generalized solution of the orthogonal procrustres problem,” *Psychometrika*, vol. 31, 1966.

[36] A. Krizhevsky, “Learning multiple layers of features from tiny images,” tech. rep., University of Toronto, 2009.

[37] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” tech. rep., California Institute of Technology, 2007.

[38] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, 2004.

[39] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, “Large-scale image retrieval with compressed fisher vectors,” in *CVPR*, 2010.

[40] D. Nistér and H. Stewénius, “Scalable recognition with a vocabulary tree,” in *CVPR*, 2006.

[41] H. Jégou and O. Chum, “Negative evidences and co-occurrences in image retrieval: the benefit of pca and whitening,” in *ECCV*, 2012.

Albert Gordo received the BSc degree in Computer Engineering and MSc degree in Intelligent Systems from the University Jaume I in Castellón, Spain, in 2007 and 2009, respectively. In 2013 he obtained a Ph.D degree from the Computer Vision Center of Barcelona, Spain in collaboration with the Xerox Research Centre Europe in Grenoble, France. He is currently a postdoctoral student at the LEAR team at INRIA, Grenoble. His main research interests include document image analysis, image classification and retrieval, and object localization, particularly focusing on large-scale problems.

Florent Perronnin holds an Engineering degree from the Ecole Nationale Supérieure des Télécommunications and a Ph.D. degree from the Ecole Polytechnique Fédérale de Lausanne. From 2000 to 2001 he was a Research Engineer with the Panasonic Speech Technology Laboratory working on speech and speaker recognition. In 2005, he joined the Xerox Research Centre Europe in Grenoble where he currently manages the Computer Vision group. His main interests are in the application of machine learning to computer vision tasks such as image classification, retrieval or segmentation.

Yunchao Gong received the BE degree in software engineering from Nanjing University, China, and the MS degree in computer science from the University of North Carolina at Chapel Hill in 2009 and 2012, respectively. He is currently working toward the PhD degree in Department of Computer Science at UNC Chapel Hill. His research interests include object recognition, image retrieval, locality sensitive hashing, and machine learning. He is a student member of the IEEE.

Svetlana Lazebnik received her Ph.D. in 2006 at the University of Illinois at Urbana-Champaign. From 2007 to 2012, she was an assistant professor of computer science at the University of North Carolina at Chapel Hill. As of January 2012, she has moved back to UIUC as an assistant professor. She is the recipient of an NSF CAREER award and a Microsoft Research Faculty Fellowship, a member of the 2011 DARPA Computer Science Study Group, and a member of the editorial board of the International Journal of Computer Vision. Her research interests include computer vision, image understanding, and machine learning techniques for visual data.