Distortion Minimization Hashing

Tongtong Yuan, and Weihong Deng

***Abstract*—Hashing method applied to large-scale image retrieval has drawn much attention due to its high efficiency and favorable accuracy. Its related research generally involves two basic problems: similarity-preserving projection and information-preserving quantization. Most previous works focus on learning projection approaches ~~but ignore~~ the importance of quantization strategies. Although several hashing quantization models have recently been proposed to improve the retrieval performance by assigning multiple bits to projected directions, these methods still suffer from suboptimal results, because the critical information loss in quantization procedure has not been considered. In this paper, to construct an effective quantization model, we ~~transfer~~ rate-distortion theory to hashing quantization procedure, and minimize the distortion to reduce the information loss. Furthermore, combining Principal Component Analysis (PCA) with our quantization strategy, we present a quantization-based hashing method named Distortion Minimization Hashing (DMH). Extensive experiments on one synthetic dataset and three image datasets demonstrate the superior performance of our proposed methods over several quantization techniques and state-of-the-art hashing methods.**

***Index Terms*—Machine Learning, Image Retrieval, Hashing Quantization, Rate-distortion Theory, Iterative Optimization.**

I. Introduction

Content-Based Image Retrieval (CBIR), which has ~~attracted~~ much attention [1]–[6] due to the explosive increase of multimedia information, ~~puts forward high requirement to~~ nearest neighbor (NN) search. However, NN search for large-scale image retrieval in Euclidean space will occupy large storage space and influence the retrieval efficiency [7]–[10]. Subsequently, there emerge many works [11]–[14] to simplify NN search by returning approximate nearest neighbor (ANN) rather than exact nearest neighbor. A representative technique of ANN search is hashing, which learns compact binary codes as image representation and searches the similar images in Hamming space. The original high-dimensional and real-valued data is mapped into low-dimensional and binary data by hashing techniques, which ~~largely~~ decreases storage consumption. Due to the fast speed and high accuracy, hashing methods have been popular in computer vision to solve image retrieval tasks [15]–[18]. Directly learning compact binary codes from the original features is NP hard, hence majority of existing hashing methods have a two-step learning paradigm to preserve the original similarity. The first procedure is learning a similarity-preserving projection and the second is quantizing real data into binary codes.

Previous studies have concentrated on ~~learning~~ the projection procedure rather than the quantization procedure. Among these various hashing techniques, the most popular hashing approach is Locality-Sensitive Hashing (LSH) [15], which generalizes random projections to map the original data into a new subspace. Furthermore, many variants of LSH have proposed to accommodate p-norm distance [19] and kernel similarity [16]. However, LSH-based methods have a poor performance at short code length due to the data-independent learning strategies. Different from LSH-based methods, many researches have explored the data structure to learn data-dependent projections, including Spectral Hashing (SH) [18], Iterative Quantization (ITQ) [20], K-means Hashing [21], *etc*. These projection-based hashing methods adopt single-bit quantization while this quantization approach causes much information loss and influences the retrieval performance.

Many researchers have realized the importance of quantization strategy in hashing methods. They have explored multi-bit quantization strategies outperforming the traditional quantization strategy. These multi-bit quantization approaches can be roughly categorized into equal-bit quantization methods [22], [23] and variant-bit quantization methods [24], [25]. Equal-bit quantization allocates the same bits to each projected dimension without considering of the diversity of each dimension, causing much information loss. On the contrary, variant-bit quantization, learning a bit allocation scheme, allocates different bits to dimensions by taking the data distribution into account. Generally, projections with more information should obtain more bits, accordingly adaptive variant-bit quantization methods have more promising performance. Actually, compared with equal-bit quantization, variant-bit quantization reduces the information loss. Since information loss will bring negative effect to quantization performance, learning an optimal variant-bit quantization is supposed to consider the information loss. However, existing quantization strategies have not attached much importance to the information loss. Besides, some unreasonable settings also influence the retrieval performance. Moreover, minimizing the information loss for these strategies is an NP-hard problem due to both the inherent discreteness in quantization results and the uncertainty in the measurement of information loss. Therefore, the exploration of hashing quantization procedure is worthwhile in both theory and practice.

In this paper, we propose a variant-bit quantization model on the basis of rate-distortion theory to minimize the information loss in hashing quantization procedure. In the aspect of information theory, the quantization step in hashing can be regarded as lossy source coding applied to data compression. Actually, the variant-bit quantization strategy in hashing aims to preserve the informativeness under given projected data and code length, thus the critical problem is learning a bit allocation scheme in the quantization stage to minimize information loss, which is generally reflected by distortion in the encoding theory [26]. Therefore, we build a simple quantization model

Pattern Recognition and Intelligent System Laboratory, Beijing University of Posts and Telecommunications, Beijing 100876, China.

named Distortion Minimization Quantization (DMQ) by rate-distortion theory in lossy source coding [27]. In this model, we utilize distortion to measure the information loss, which avoids handling the discreteness problem in quantization results. Meanwhile, we develop an iterative optimization algorithm to jointly minimize the distortion and learn a bit allocation scheme. To the best of our knowledge, this is the first time to use the distortion from rate-theory to represent information loss in hashing quantization procedure. Furthermore, utilizing the combination of PCA projection and our DMQ model, we propose an unsupervised hashing method named Distortion Minimization Hashing (DMH). To evaluate the effectiveness of our model, we test the performance of our method on one synthetic dataset and three real-world datasets. All the results indicate our quantization strategy outperforms these quantization strategies and our DMH has superiority to state-of-the-art hashing methods.

The main contributions of this paper are three-fold. First, our model utilizes distortion to measure the information loss in quantization, which is the first attempt to transfer rate-distortion theory to hashing quantization procedure. Moreover, we construct our objective function based on minimizing the distortion and explore an iterative algorithm to learn an optimal bit allocation scheme. Second, by combining PCA projection with our quantization strategy, we develop a quantization-based hashing method outperforming projection-based hashing methods. Lastly, extensive experiments on one synthetic data and three image datasets, including CIFAR10, LabelMe and SIFT-1M, demonstrate our superiority to both several quantization strategies and state-of-the-art hashing methods.

The remainder in this paper is organized as follows. Section II briefly introduces the related works. Section III starts with rate-distortion theory and describes our proposed model. In Section IV, comparing with some state-of-the-art methods, we report and analyze the experimental performances of our model. Finally, conclusions are drawn in Section V.

II. Related works

Image hashing has paid much attention to projection learning for a long time and many researchers have proposed various projection approaches to improve the retrieval precision. The typical data-independent hashing method is Locality-Sensitive Hashing (LSH), which learns random projections without using data. Due to the popularity of LSH [15], there emerge many variants of LSH, like Locality-Sensitive Binary Codes from Shift-invariant Kernels (SKLSH) [16], Kernelized Locality-Sensitive Hashing (KLSH) [28] and so forth. However, data-dependent methods, such as Spectral Hashing (SH) [18] and Iterative Quantization (ITQ) [20], have presented better performance utilizing the data property. In particular, ITQ proposes a novel rotation transformation to reduce the quantization error and it is recognized as a promising hashing method due to the accuracy and efficiency. The corresponding quantization strategy of these hashing methods is single-bit quantization (SBQ), which uses '0' as the threshold for zero-centered data. The general form of hash function is

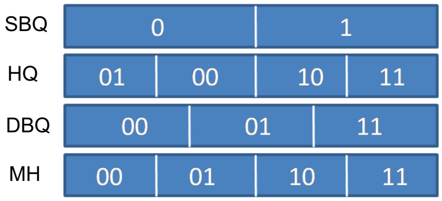


Fig. 1: Binary codes generated by different quantization methods. Except SBQ, other methods produce codes by assigning double bits to each dimension.

$$\mathbf{B} = sgn(\mathbf{XW}), \quad sgn(v) = \begin{cases} 1 & v > 0 \\ 0 & v \leq 0 \end{cases}, \tag{1}$$

where $\mathbf{X}$ is the zero-centered data and $\mathbf{W}$ is the projection matrix. The threshold of single-bit quantization (SBQ) usually lies in the region of high point density, which will destroy the neighbor structure [22]. Besides, the information loss in SBQ is large by assigning only one bit to each dimension. Due to the limitations of single-bit quantization, it requires much effort to improve the performance of hashing. Gradually, many researchers have realized the importance of quantization strategies in hashing, since multi-bit quantization methods have achieved significant improvement in retrieval precision.

In the early stage, multi-bit quantization methods use equal-bit strategy, which allocates the same number of bits to each dimension. For example, instead of using single bit for each direction, Hierarchical Quantization (HQ) for Hashing With Graphs [29] assigns two bits to encode each projection dimension by dividing the dimension into four parts. However, HQ influences the consistency of Hamming distance with Euclidean distance due to the destruction of neighbor structure. Departing form HQ, Kong *et al.* [22] propose Double-Bit Quantization (DBQ), which generates three double-bit codes to preserve the neighbor structure. However, this encoding approach will reduce the encoding efficiency because assigning n bits to each dimension can only generate $n-1$ codes rather than 2^n codes. To further improve the performance, Kong *et al.* [23] also present a more flexible quantization approach named Manhattan Hashing (MH) that is based on Manhattan distance and natural binary codes (NBC). Compared with DBQ, MH effectively preserves neighborhood structure of data and avoids the decline of encoding efficiency, bringing a significant improvement to image retrieval. Notably, the distance measurement in SBQ, HQ and DBQ is Hamming distance while the measurement in MH is Manhattan distance. As shown in Fig. 1, Hamming distance d_h in HQ is not consistent with Euclidean distance d_e, where $d_h(01,10) > d_h(01,11)$ but $d_e(01,10) < d_e(01,11)$. To avoid this problem, DBQ uses three binary codes rather than four codes, where $d_h = 1$ for neighbor codes and $d_h = 2$ otherwise. Differently, MH use decimal distance d_d to measure the distance of corresponding NBC codes, like $d_d(00,01) = 1$, $d_d(00,10) = 2$ and $d_d(00,11) = 3$. When computing Manhattan distance d_m

between two points, we need to obtain the sum of decimal distance on their dimensions. Given two binary codes a and b generated by MH, the Manhattan distance between them is:

$$d_m(a,b) = \sum_i d_d(a_i, b_i), \tag{2}$$

where a_i and b_i correspond to the i_{th} projected dimension which contains q_i bits. If $a = \{000110\}$, $b = \{110000\}$ and $q_i = 2$ for all i, then

$$\begin{aligned} d_m(000110, 110000) &= d_d(00,11) + d_d(01,00) + d_d(10,00) \\ &= 3+1+2 = 6. \end{aligned} \tag{3}$$

However, equal-bit quantization inevitably increases the information loss by reason that it assigns the same bits to each dimension but dimensions with more informativeness should be assigned more bits actually. To adapt the data distribution, Xiong *et al.* [24] and Deng *et al.* [25] propose variant-bit quantization which allocates different bits for each dimension according to the variance. Adaptive Quantization for Hashing (AQH) [24] is based on MH while Adaptive Multi-bit Quantization (AMBQ) [25] regards DBQ as the fundamental strategy and constructs neighbor-preserving binary codes. Jointly utilizing Manhattan distance and variant-bit quantization, AQH performs better retrieval results than AMBQ. However, AMBQ and AQH have ignored to specify the information loss, hence the research for quantization in hashing remains to be further explored.

All the multi-bit quantization methods have shown better performance than single-bit quantization methods. Equal-bit quantization and variant-bit quantization have no obvious difference when each dimension has similar informativeness. While each dimension has different informativeness, equal-bit quantization methods are powerless and variant-bit quantization methods perform superiority. Therefore, variant-bit quantization approach is a leap from the equal-bit allocation by considering the different properties in different dimensions. Actually, the existing variant-bit quantization methods also have weaknesses. These methods have not considered the information loss in the quantization, which brings negative effect to the retrieval performance. In addition, some unreasonable settings in existing strategies (e.g. limiting the maximum bits to 4 in AQH) also influence the retrieval performance. Directly reducing the information loss is difficult because of the discreteness of quantization results. Meanwhile, the representation of information loss in quantization procedure is uncertain, thus the theoretical basis of bit allocation strategy is pending further study. To address these problems, we propose a novel and adaptive quantization approach based on rate-distortion theory.

III. Distortion minimization model

This section describes our proposed model in detail. Firstly, we introduce the rate-distortion theory, which is the basis of our method. Then we propose our Distortion Minimization Quantization (DMQ) to learn a bit allocation strategy for Gaussian source. Finally, we develop an unsupervised hashing method named Distortion Minimization Hashing (DMH) by deploying DMQ on PCA-projected data.

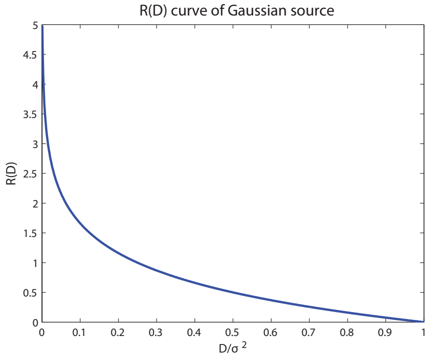


Fig. 2: Rate-distortion curve of Gaussian source. $R(D)$ stands for the minimum information rate to make sure that the average distortion of σ^2- variance Gaussian source is less than or equal to D.

A. Rate-distortion Theory

Given an input source $\mathbf{x} \in \mathbb{R}^n$, the quantization procedure aims to learn the output source $\mathbf{y}$. Assuming D stands for the distortion in quantization procedure, we consider the square error measurement, i.e., $D = (\mathbf{x} - \mathbf{y})^2$. To measure the information loss in quantization, we introduce the rate-distortion theory.

Rate-distortion theory [26], [30], [31] shows learning an explicit rate-distortion function of general source is difficult except Gaussian source. Hence, we focus on the Gaussian source to construct a unified expression for providing convenient extension in the following quantization model. We denote $\mathbf{R}$ and σ^2 as the information rate and the variance of the given Gaussian source $\mathbf{x}$, respectively. Then the rate-distortion function is:

$$R(D) = \frac{1}{2} max(0, log\frac{\sigma^2}{D}) = \begin{cases} \frac{1}{2} log\frac{\sigma^2}{D} & 0 < D \leq \sigma^2 \\ 0 & D > \sigma^2 \end{cases}. \tag{4}$$

The rate-distortion curve is shown in Fig. 2, which obviously presents that higher rate leads to lower distortion. Furthermore, we can get the following equation (3), which is the inverse function of (2). This equation gives the formula of $D(R)$ under the rate of lossy compression R:

$$D(R) = \sigma^2 2^{-2R}, \ R \geq 0, \tag{5}$$

where the computation is based on single Gaussian source that can be regarded as one dimension of data.

Considering the data generally contains multiple dimensions, we extend the single Gaussian source into parallel Gaussian source. If a multi-dimensional discrete independent Gaussian source is $\mathbf{X}^k = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}$, where $\mathbf{x}_i$ is a single Gaussian source with zero mean value and σ_i^2 variance, then this source is named as independent parallel Gaussian source [26]. Given the information rate $\mathbf{R} = \{R_1, R_2, \cdots, R_k\}, R_i \geq 0$, the total minimum distortion is:

$$D = \sum_{i=1}^{k} D_i = \sum_{i=1}^{k} \sigma_i^2 2^{-2R_i}, \tag{6}$$

which is proposed in information theory. Actually, compared with the general sources, Gaussian source is the most difficult to be compressed [26]. In other words, the distortion of Gaussian source is the upper limit of other sources, thus minimizing the distortion of general source can be relaxed by minimizing the distortion of Gaussian source. Moreover, independent parallel Gaussian source has brief distortion function, which provides the quantization procedure with convenience.

Making a quantization for projected data in hashing methods is similar to lossy source coding in information theory. The information loss in quantization procedure can be effectively reflected by distortion with considering the discreteness of quantization results. In addition, the assigned bits in quantization is equivalent to information rate while the data before quantization is the source pending to be encoded. Therefore, utilizing these similarities in quantization and source coding procedure, we can construct our model based on rate-distortion theory in information theory. Particularly, via minimizing the distortion of independent parallel Gaussian source, we can learn a bit allocation strategy for hashing quantization.

B. Distortion Minimization Quantization

In this section, we focus on learning a quantization strategy named Distortion Minimization Quantization for projected data, which can be regarded as the second step in hashing.

Suppose zero-centered Gaussian source is denoted as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ represents the column of projected data $\mathbf{X}$. The corresponding variance vector is $\sigma^2 = \{\sigma_1^2, \sigma_2^2, \cdots, \sigma_k^2\}$. Assuming the total information rate is C and information rate of each dimension is R_i, the distortion in quantization procedure can be formulated as:

$$\begin{gathered} D(R) = \sum_{i=1}^{k} D_i = \sum_{i=1}^{k} \sigma_i^2 2^{-2R_i} \\ s.t. \sum_i R_i = C,\ R_i \in \mathbb{N}, \end{gathered} \tag{7}$$

where $\mathbb{N}$ stands for natural number.

Due to the integer constraint, it is NP-hard to directly minimize the objective. Thus, we explore a simple optimization algorithm to learn a global optimal $\mathbf{R}$. The bit allocation will be iteratively updated by adjusting the rate values of minimum distortion and maximum distortion until it has been optimal. Assume information rate corresponding to maximum distortion is R_p and the non-zero information rate corresponding to minimum distortion is R_q. Notably, here exists natural number constraint, i.e. the values of R_p and R_q are nonnegative integers. However, if the value of information rate is zero, this value cannot be reduced any further. Thus, the information rate R_q corresponding to minimum distortion should satisfy: $R_q \neq 0$. The relative distortions D_p and D_q can be computed by:

$$D_p = \sigma_p^2 2^{-2R_p},\ D_q = \sigma_q^2 2^{-2R_q}\ . \tag{8}$$

The decision condition to update R_p and R_q by $R_p = R_p + 1$ and $R_q = R_q - 1$ is depended on the values of distortions. We only consider the relative distortions D_p and D_q, and denote the changed distortions as D'_p and D'_q. The decision condition can be derived via comparing the value:

$$\begin{aligned} &(D_p + D_q) - (D'_p + D'_q) \\ &= \sigma_p^2 2^{-2R_p} + \sigma_q^2 2^{-2R_q} - \sigma_p^2 2^{-2(R_p+1)} - \sigma_q^2 2^{-2(R_q-1)} \\ &= \frac{3}{4}\sigma_p^2 2^{-2R_p} - 3\sigma_q^2 2^{-2R_q} \\ &= 3(\frac{1}{4}D_p - D_q) > 0 \Rightarrow \frac{1}{4}D_p - D_q > 0\ . \end{aligned} \tag{9}$$

The information rate will be updated only if $\frac{1}{4}D_p - D_q > 0$ is true. Otherwise, the information rate has been optimal. The detail optimization procedure to find optimal R is shown in Algorithm 1. Information rates R in this scenario is equivalent to the number of assigned bits.

After learning optimal R, we divide the i_{th} column into 2^{R_i} parts by k-means clustering and obtain the natural binary codes which are similar to MH [23] and AQH [24]. The measurement of similarity in image retrieval is achieved by Manhattan distance. For example, two binary codes $a = \{010110110\}$, $b = \{110101011\}$ have four dimensions, and the rate for each dimension is $R_1 = R_2 = 3, R_3 = 2, R_4 = 1$, then the distance between a and b is computed as follows:

$$\begin{aligned} &d_m(a,b) = d_d(010,110) + d_d(110,101) + d_d(11,01) + \\ &d_d(0,1) = 4 + 1 + 2 + 1 = 8. \end{aligned} \tag{10}$$

Thus, we finish the exploration of our quantization procedure and relative measurement for retrieval.

By analysing the objective function, we can note that the information rates to minimizing the distortion are determined by the variances of projected dimensions. Obviously, if the variances are balanced, the learned information rate $\mathbf{R}$ will be consisted of similar numbers and the variant-bit allocation strategy will have similar results with equal-bit quantization.

Algorithm 1 Optimization Procedure

```
Input:
    training data $\mathbf{X} \in \mathbb{R}^{n \times k}$;
    final code length $C$;
Output:
    information rate $\mathbf{R}$;
1: sort variances: compute the variances $\sigma_i^2$ of each column
   $\mathbf{x}_i$, then sort the columns in descending order by the
   variance;
2: initialize $\mathbf{R}$: assign one bit to each column in order, then
   loop 2 until the sum of bits is up to $C$;
3: find $D_p$ and $D_q$: compute the respective distortion $D_i$,
   then find maximum distortion $D_p$ with information rate
   $R_p$ $(0 < p \leq k)$ and minimum distortion $D_q$ with non-
   zero information rate $R_q$ $(R_q \neq 0,\ 0 < q \leq k)$;
4: minimize the distortion by iteratively updating $\mathbf{R}$:
5: while $\frac{1}{4}D_p > D_q$ do
6:     update $R$ by $R_p \leftarrow R_p + 1$, $R_q \leftarrow R_q - 1$;
7:     recompute $D_p$, $D_q$ by (8);
8:     find new $D_p$, $D_q$ by step 3 and update the index $p$, $q$;
9: end while
```

During optimization procedure, the complexity of computing variance is $\mathcal{O}(kn)$. In the worse case, sorting the columns or distortions will cost $\mathcal{O}(k^2)$ time. Initializing information rate yields a complexity of $\mathcal{O}(C)$. Iteratively updating R mainly involves searching for the maximum and minimum distortions, costing $\mathcal{O}(k)$ at a time. The maximum iterations is C, thus the total time complexity of iteratively updating R is $\mathcal{O}(Ck)$. Given $C \ll n$ and $k \ll n$, $\mathcal{O}(kn + k^2 + C + Ck)$ is equivalent to $\mathcal{O}(n)$. Therefore, the total complexity of this optimization is described as $\mathcal{O}(n)$.

C. Distortion Minimization Hashing

Hashing methods aim to generate compact binary codes by learning a projection and making a quantization. The common strategy to reduce the dimensionality in the projection procedure is PCA, which is used in [17], [18], [20]. PCA extracts the uncorrelated principle components by using the eigenvectors from the covariance data. The presupposition of PCA is that data follows Gaussian distribution. In addition, PCA-projected data is composed of several uncorrelated directions, hence we can assume this data as independent parallel Gaussian source. Simultaneously, PCA-projected data has unbalanced variance distribution and the range of variances is large. These properties are exactly compatible with our model.

Suppose our zero-centered data is denoted as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}$, where $\mathbf{x}_i \in \mathbb{R}^n$ represents the column of $\mathbf{X}$. We denote the projected data as $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k\}$, whose variance is $\sigma^2 = \{\sigma_1^2, \sigma_2^2, \cdots, \sigma_k^2\}$. Then PCA projection has the following formula:

$$\mathbf{X}^T\mathbf{X} = \mathbf{W}\mathbf{\Sigma}\mathbf{\Sigma}^T\mathbf{W}^T, \tag{11}$$

where projection matrix $\mathbf{W} \in \mathbb{R}^{k\times k}$ and eigenvalue matrix $\mathbf{\Sigma} \in \mathbb{R}^{k\times k}$ are learned by the eigenvalue decomposition of $\mathbf{X}^T\mathbf{X}$. PCA-projected data $\mathbf{Y}$ is generated by $\mathbf{Y} = \mathbf{X}\mathbf{W}$.

In order to capture more variance information, there are some parameter settings different from DMQ. Notably, in our hashing model, we project the original data without reducing the dimensionality and set up the quantization model as:

$$\begin{aligned} D(R) &= \sum_{i=1}^{k} \sigma_i^2 2^{-2R_i} \\ s.t. &\sum_i R_i = k, R_i \in \mathbb{N}, \end{aligned} \tag{12}$$

where the sum of information rates is k rather than the given final code length C. Generally, hashing methods generate binary codes with a shorter code length than the dimensionality of original data to reduce the dimension, accordingly the final code length C is not larger than k. To satisfy the final code length, we construct a constraint to select several projections with allocated bits in descending order by information rates:

$$\begin{gathered} \sum_{i=1}^{t-1} R_i + R_t - \alpha = C \\ 1 \le t \le k,\ 0 \le \alpha < R_t,\ \ R_{(t+1):k} = 0, \end{gathered} \tag{13}$$

in which the information rate of the final selected dimension will be modified by an integer α and the rates of unselected dimensions will be zero. Thus, the sum of this truncated information rates will be same as C. After learning optimal $\mathbf{R}$, the final step to generate binary codes is the same as Distortion Minimization Quantization. The whole procedure of our DMH is described in Algorithm 2.

Our special settings for this combination of PCA and DMQ are based on the consideration of preserving comprehensive information. PCA-projected data satisfies all the demands for distortion minimization quantization. Meanwhile, the variance distribution will determine the result of bit allocation. In order to learn an adaptive quantization approach and preserve the total informativeness after PCA projection, we preserve all of the dimensions of projected data without truncating the eigenvectors. This is because the diversity of variances will enlarge the diversity of rates while the larger diversity of rates will lead to a more precise bit allocation. When minimizing the distortion, we set $\sum R_i = k$ to make sure that the code length is the same as the projections and each dimension has one bit initially. In this case, each projection can make contributions to the assignment of the information rate. This quantization strategy can jointly minimize the distortion and maximize the assigned bits of dimensions associated with large variance when the total information rate is k. Otherwise, the variance distribution will have a low diversity and the assigned bits for major dimensions will be decreased. Moreover, compared with directly setting $\sum R_i = C$, our settings need not to update the whole information rate with the changed code length. Obviously, we only need to select the suitable dimensions and relative information rates to meet the final code length.

Algorithm 2 Distortion Minimization Hashing

```
Input:
    training data $\mathbf{X} \in \mathbb{R}^{n\times k}$;
    final code length $C, C \le k$;
Output:
    information rate $\mathbf{R}$;
    binary code set $\mathbf{B}$;
learn projected data $\mathbf{Y} \in \mathbb{R}^{n\times k}$ by PCA;
compute the variances $\sigma_i^2$ of each projection $\mathbf{y}_i$;
minimize the objective function (12) by Algorithm 1;
select and modify the rate $\mathbf{R}$ to satisfy constraint (13);
generate $2^{R_i}$ centers for each column $\mathbf{y}_i$ then assign the
    natural binary codes to obtain $\mathbf{B}$. If $R_i = 0$, the relative
    dimension will be discarded.
```

IV. Experiments and evaluations

A. Datasets And Experiment Settings

We firstly run simulations on one synthetic projected dataset and compare our DMQ with DBQ, MH under the Gaussian model. We randomly generate $50,000$ zero-mean data following the distribution of independent parallel Gaussian source. The feature dimensionality of synthetic data is 50 and the variance of each dimension is ranging from 0.1 to 1000. We

TABLE I: Comparisons of the quantization strategies on three datasets. Mean average precision (MAP) is used to measure the whole retrieval performance.

	MAP on CIFAR10				MAP on LabelMe-22K				MAP on SIFT-1M			
# bit	16	32	64	128	16	32	64	128	16	32	64	96
SBQ	0.043	0.037	0.031	0.026	0.058	0.046	0.035	0.027	0.042	0.110	0.171	0.163
DBQ	0.080	0.130	0.192	0.247	0.101	0.147	0.171	0.156	0.034	0.163	0.321	0.423
MH	0.101	0.171	0.256	0.311	**0.124**	0.179	0.218	0.194	0.073	0.229	0.463	0.571
AMBQ	0.091	0.115	0.218	0.253	0.096	0.131	0.178	0.189	0.042	0.181	0.325	0.431
AQH	0.102	0.180	0.323	0.348	0.112	0.186	0.281	0.465	0.075	0.243	0.473	0.578
DMH	**0.130**	**0.182**	**0.397**	**0.576**	0.086	**0.191**	**0.365**	**0.553**	**0.076**	**0.262**	**0.527**	**0.598**

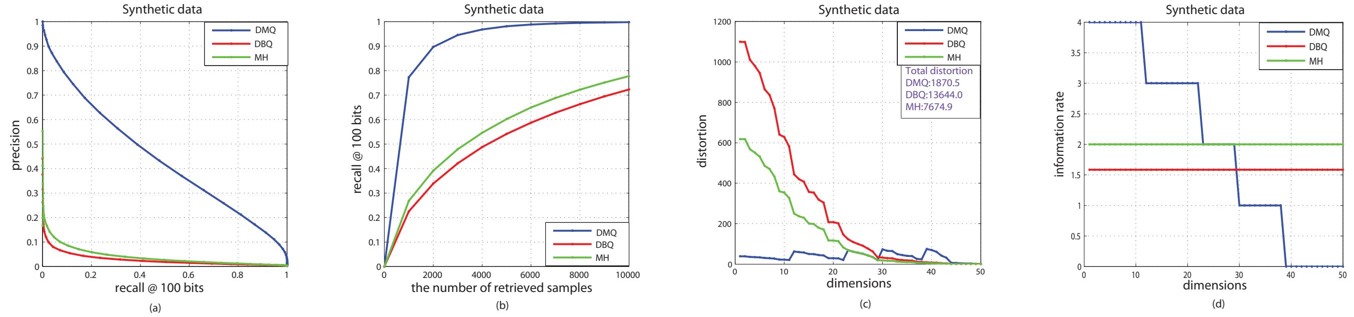


Fig. 3: (a). Precision-recall curves on synthetic data. (b). Recall curves on synthetic data. (c). Distortion curves and total distortion results on synthetic data. (d). Information rate curves on synthetic data.

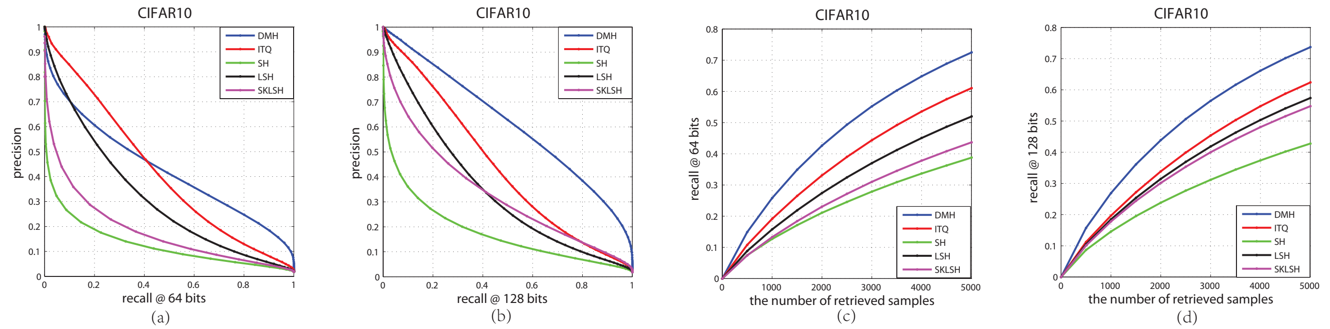


Fig. 4: (a). Precision-Recall curves at 64 bits on CIFAR10. (b). Precision-Recall curves at 128 bits on CIFAR10. (c). Recall curves at 64 bits on CIFAR10. (d). Recall curves at 128 bits on CIFAR10.

use $1,000$ samples as testing data and the remaining samples make up training data.

Furthermore, we compare our DMH with quantization-based methods and projection-based methods on three real-world image sets:

- CIFAR10 used in [20], [24], [32] contains $60,000$ 32×32 images which are categorized into 10 classes. Each image is represented by 320-dimensional grayscale GIST feature [33] as in ITQ [20].
- LabelMe-22K [25], [34] is consisted of $2,2019$ images sampled from the large LabelMe dataset. Each image is described by 512-dimensional GIST feature [33].
- SIFT-1M [35], [36] is composed of one million 128-dimensional SIFT features.

Since all of the involved methods are unsupervised, we use the Euclidean-based ground truth and set the distance within 50 neighbors as nominal threshold to distinguish positive or negative images as in ITQ [20]. All the experiments are conducted under the same final code length. Precision-recall curve, mean average precision (MAP) and the recall curve for different numbers of retrieved points are the evaluation criteria in our experiments. We randomly choose $1,000$ images from CIFAR10 and SIFT-1M as testing set. For LabelMe-22K, we choose $2,000$ samples as testing set. The remaining images form the training set. All the experimental results are averaged over 5 random train/test partitions.

B. Results On Synthetic Data

We expand the original synthetic data into 100 dimensions by several quantization measures, including DBQ, MH and our DMQ. The retrieval performance has been shown in Fig. 3(a) and Fig. 3(b). Our method obtain the best performance in both precision-recall curves and recall curves corresponding to the number of retrieved samples. MH has better performance than DBQ due to minor distortion. Fig. 3(c) describes the distortion curves corresponding to the dimensions and the results of total

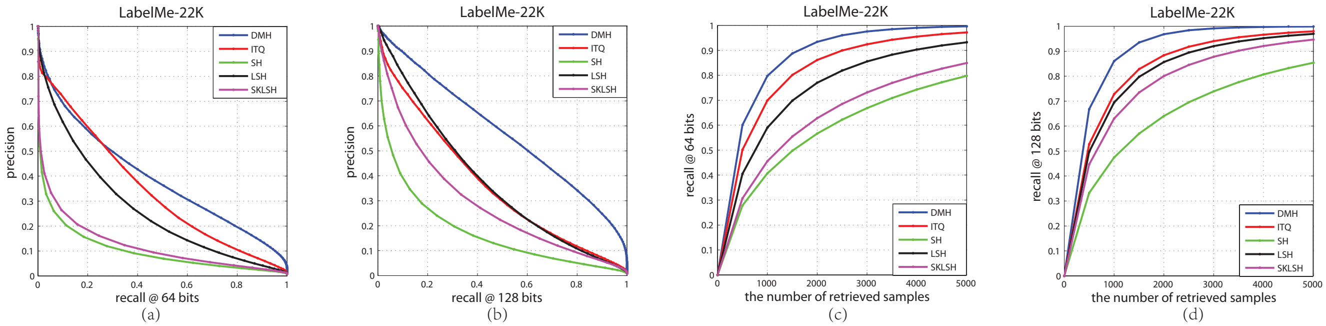


Fig. 5: (a). Precision-Recall curves at 64 bits on LabelMe-22K. (b). Precision-Recall curves at 128 bits on LabelMe-22K. (c). Recall curves at 64 bits on LabelMe-22K. (d). Recall curves at 128 bits on LabelMe-22K.

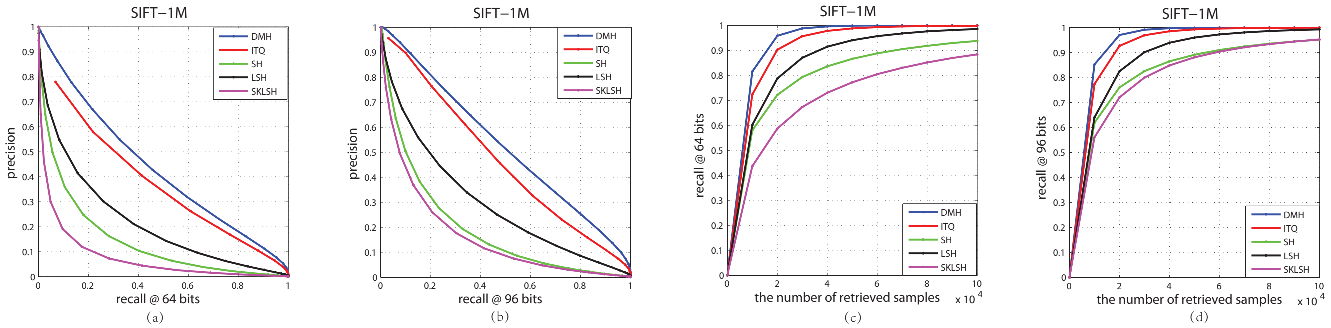


Fig. 6: (a). Precision-Recall curves at 64 bits on SIFT-1M. (b). Precision-Recall curves at 96 bits on SIFT-1M. (c). Recall curves at 64 bits on SIFT-1M. (d). Recall curves at 96 bits on SIFT-1M.

distortion. The trend in distortion curve indicates the difference of bit allocation strategies. Since DBQ uses the neighbor-preserving code, the information rate of each dimension is $log_2 3$. While MH uses the natural binary codes, the information rate is $log_2 4 = 2$. Based on the smaller information rate, DBQ has higher distortion than MH. Our method significantly decreases the distortion by variant-bit quantization and shows the lowest total distortion to benefit the informativeness preservation, therefore our method achieves the best retrieval performance. Moreover, we record the information rate of each dimension and report them in Fig. 3(d). Our DMQ has variant information rates ranging from 0 to 4, which is more adaptive to the Gaussian projected data than other quantization methods. The preliminary experimental results suggest variant-bit quantization has apparent advantages than equal-bit quantization when handling the data with unbalanced variance.

C. Results On Image Datasets

In this section, we compare our DMH with quantization-based methods and projection-based methods on real-world image sets to evaluate the retrieval performance.

1) Comparisons With Quantization Methods: Variant-bit allocation strategy is depended on the variances of projected data, thus we only combine all the involved quantization strategies with PCA-projected data, which has unbalanced variances. Once combining with other projection approaches, these variant-bit quantization strategies, including AMBQ, AQH and our DMQ, will have no obvious advantages than equal-bit quantization methods. Table I shows the results of mean average precision (MAP) on three mentioned datasets. SBQ with PCA is exactly PCAH [17], apparently it shows the worst results due to the simplest quantization approach. DBQ [22], which uses the double-bit quantization and develops neighbor-preserving codes, has better performance than SBQ. MH [23] deploys Manhattan distance and double-bit natural binary codes achieving higher precision than DBQ. As for variant-bit quantization methods, AMBQ[8] utilizes the neighbor-preserving code and Hamming distance as DBQ while AQH [24] and our DMH method use natural binary codes and Manhattan distance as MH. Consequently, the MAP of AMBQ is higher than DBQ, but lower than MH, AQH and our DMH. Our method has the highest MAP in almost all of the comparisons on three datasets, owing to our optimal bit allocation strategy. Notably, the performance of our DMH is slightly lower than MH at the 16 bits on LabelMe-22K, because our method only utilizes few dimensions ($5-6$ dimensions) to generate binary codes at short code length. The superiority of our method is more apparent at longer code code due to enough information provided by more dimensions. The results further indicate minimizing the distortion is an effective strategy to reduce the information loss and improve the quantization performance.

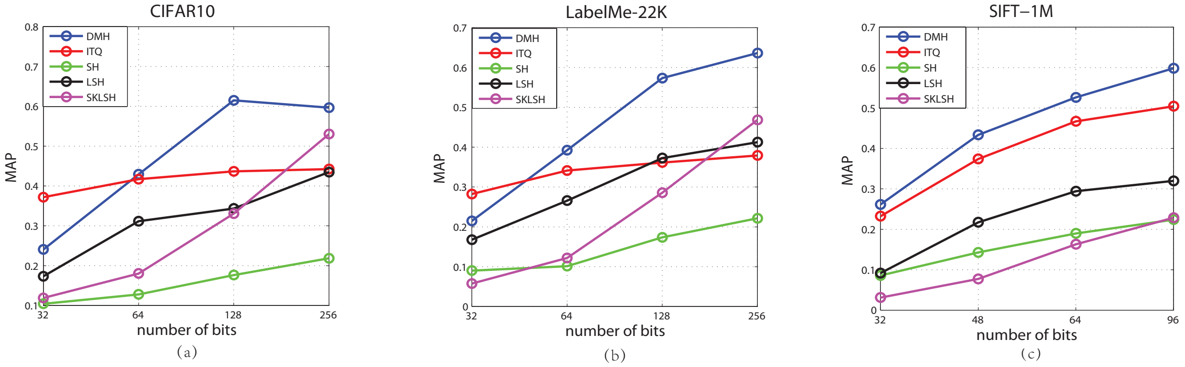
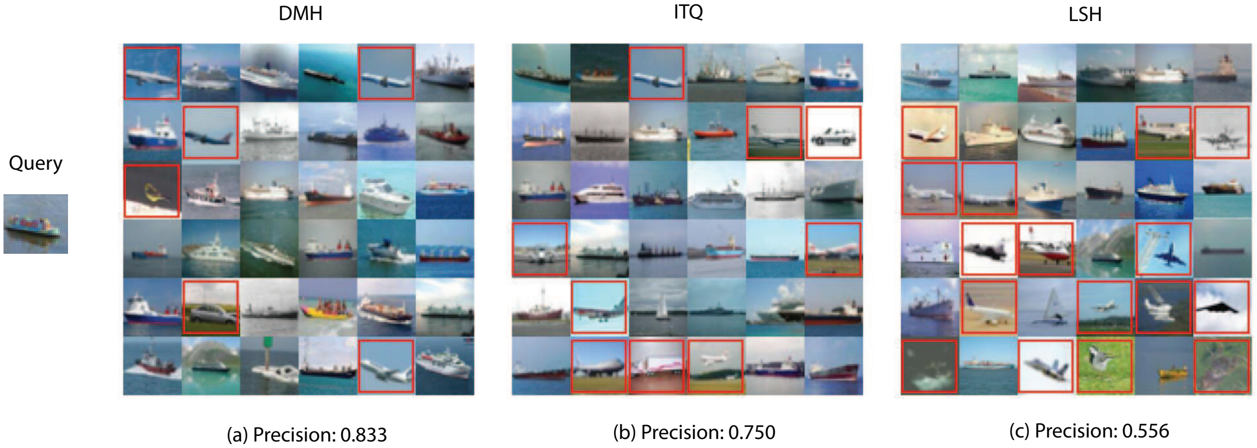


Fig. 7: (a). MAP curves on CIFAR10. (b). MAP curves on LabelMe-22K. (c). MAP curves on SIFT-1M.



(a) Precision: 0.833

(b) Precision: 0.750

(c) Precision: 0.556

Fig. 8: Visualization of top 36 retrieved images for query (ship) at 64 bits on CIFAR10. Red rectangle represents false positive image. The retrieval precision is obtained by computing the proportion of true positive images.

2) Comparisons With Hashing Methods: We compare our DMH method with four state-of-the-art hashing methods, including ITQ [20], SH [18], LSH [15] and SKLSH [16]. These projection-learning methods are based on single-bit quantization while our DMH belongs to multi-bit quantization. The precision-recall results and recall results at 64 and 128 bits on three datsets are shown in Fig. 4, Fig. 5 and Fig. 6, respectively. Our method DMH achieves better results than other hashing methods in all of the precision-recall curves and recall curves, while SH and SKLSH have poor performances. The precision-recall curve of our DMH is similar to ITQ in Fig. 4(a). When comparing the recall curves, our method has absolute advantages than others. For further comparison, we report the MAP results on these datasets in Fig. 7. Mean average precision (MAP) shows the whole performance of these hashing methods at a wide range of code length. Except that the MAP of our DMH is only second to ITQ at 32 bits on CIFAR10 and LabelMe-22K, our DMH shows the highest precisions in other comparisons. As the code length is longer, our method produces results significantly better than others. Fig. 7 illustrates our DMH outperforms other methods by a large margin at most bits. Simultaneously, these experimental results demonstrate the importance of quantization strategy in improving the retrieval performance.

Fig. 8 presents the top retrieved 36 samples by DMH, ITQ and LSH on CIFAR10 when query image is ship. The false positive samples are labelled by red rectangles. In the visualized comparison at 64 bits, our DMH performs the highest retrieval precision.

D. Discussions

We notice that the retrieval performance of our method is slightly lower than ITQ when code length is short. Actually, as shown in Table I, almost all the multi-bit quantization methods have weak performances at short code length. The reason is that multi-bit quantization methods use fewer dimensions than projection-based method to satisfy the uniformity of final code length. We take our approach as an example to discuss this phenomenon. When the code length is short, the selected dimensions by our method are too few to preserve the enough information, resulting in a lower retrieval precision. However, our method can perform the best results in all the comparisons when code length is longer, due to the sufficient information and our adaptive bit allocation strategy. The good performance of our method proves the effectiveness of transferring the rate-distortion theory to hashing quantization. In all, the comparisons with projection-learning hashing methods demonstrate that the retrieval improvement produced by quantization strategies is comparable to the improvement from projection strategies. Despite the limitation in code length, these experimental results can clearly indicate the superiority of our proposed method and the importance of quantization strategy in hashing.

V. Conclusions

Benefiting from the theoretical support of rate-distortion theory, we explore a simple but effective quantization strategy via minimizing the distortion to improve performance of the quantization in hashing. Furthermore, we develop an unsupervised hashing method by combining our adaptive quantization strategy with untruncated PCA projection to improve the performance of image retrieval. Compared with other quantization strategies and some state-of-the-art hashing methods on synthetic data and real-world image sets, our method has revealed the effectiveness in both hashing quantization and image retrieval. Moreover, the experimental results demonstrate the importance of quantization strategy in hashing and the connection between distortion in rate-distortion theory and information loss in quantization.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (No. 61573068, No. 61471048, No. 61375031 and No. 61532006), Beijing Nova Program under Grant No. $Z161100004916088$, the Fundamental Research Funds for the Central Universities under Grant No. $2014ZD03-01$, and the Program for New Century Excellent Talents in University (NCET-13-0683).

References

[1] J. Wang, W. Liu, S. Kumar, and S. Chang, "Learning to hash for indexing big datała survey," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 34–57, 2015.

[2] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval:ideas, influences, and trends of the new age," *Acm Computing Surveys*, vol. 40, no. 2, pp. 1–60, 2008.

[3] L. Yu, L. Feng, C. Chen, T. Qiu, L. Li, and J. Wu, "A novel multi-feature representation of images for heterogeneous iots," *IEEE Access*, vol. 4, no. 99, pp. 6204–6215, 2016.

[4] J. P. Heo, Y. Lee, J. He, and S. F. Chang, "Spherical hashing," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[5] W. Kong and W. J. Li, "Isotropic hashing," in *Advances in Neural Information Processing Systems*, 2012.

[6] W. Lu, A. L. Varna, and M. Wu, "Confidentiality-preserving image search: A comparative study between homomorphic encryption and distance-preserving randomization," *IEEE Access*, vol. 2, pp. 125–141, 2014.

[7] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2017.

[8] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Thiry-Fourth ACM Symposium on Theory of Computing*, 2002.

[9] T. Lee, T. Moon, S. J. Kim, and S. Yoon, "Regularization and kernelization of the maximin correlation approach," *IEEE Access*, vol. 4, pp. 1385–1392, 2016.

[10] Z. Feng and Y. Zhu, "A survey on trajectory data mining: Techniques and applications," *IEEE Access*, vol. 4, pp. 2056–2067, 2016.

[11] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *International Conference on Very Large Data Bases*, 1999.

[12] W. Liu, J. Wang, Y. Mu, S. Kumar, and S. F. Chang, "Compact hyperplane hashing with bilinear functions," in *International Conference on Machine Learning*, 2012.

[13] D. Zhang, F. Wang, and L. Si, "Composite hashing with multiple information sources," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011.

[14] Y. Zhen and D. Y. Yeung, "Active hashing and its application to image and text retrieval," *Data Mining & Knowledge Discovery*, vol. 26, no. 2, pp. 255–274, 2013.

[15] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions." in *IEEE Symposium on Foundations of Computer Science*, 2006.

[16] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Conference on Neural Information Processing Systems*, 2009.

[17] X. J. Wang, L. Zhang, F. Jing, and W. Y. Ma, "Annosearch: Image auto-annotation by search," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[18] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Conference on Neural Information Processing Systems*, 2008.

[19] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Twentieth Symposium on Computational Geometry*, 2004.

[20] Y. Gong, S. Lazebnika, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.

[21] K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[22] W. Kong and W. J. Li, "Double-bit quantization for hashing," in *AAAI Conference on Artificial Intelligence*, 2012.

[23] W. Kong, W. J. Li, and M. Guo, "Manhattan hashing for large-scale image retrieval," in *ACM SIGIR conference on Research and development in information retrieval*, 2012.

[24] C. Xiong, W. Chen, G. Chen, D. Johnson, and J. J. Corso, "Adaptive quantization for hashing: An information-based approach to learning binary codes," in *Proceedings of the 2014 SIAM International Conference on Data Mining*, 2014.

[25] C. Deng, H. Deng, X. Liu, and Y. Yuan, "Adaptive multi-bit quantization for hashing," *Neurocomputing*, vol. 151, pp. 319–326, 2015.

[26] T. Han, *Information-Spectrum Methods in Information Theory*. Springer, 2003.

[27] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criteria," *Ire National Convention Record Part*, vol. 4, pp. 93–126, 1959.

[28] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 34, no. 6, pp. 1092–104, 2012.

[29] W. Liu, J. Wang, S. Kumar, and S. F. S. F. Chang, "Hashing with graphs," in *International Conference on Machine Learning*, 2011.

[30] L. Davisson, "Rate distortion theory: A mathematical basis for data compression," *IEEE Transactions on Communications*, vol. 20, no. 6, pp. 1202–1202, 2003.

[31] Berger and Toby, "Rate-distortion theory," *Encyclopedia of Telecommunications*, vol. 42, no. 1, pp. 63–86, 2003.

[32] M. J. Shafiee, P. Siva, and A. Wong, "Stochasticnet: Forming deep neural networks via stochastic connectivity," *IEEE Access*, vol. 4, pp. 1915–1924, 2016.

[33] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[34] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[35] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European Conference on Computer Vision*, 2008.

[36] J. Wang, S. Kumar, and S. F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 34, no. 12, pp. 2393–2406, 2012.