Robust Quantization for General Similarity Search

Yuchen Guo [ID], Guiguang Ding, and Jungong Han [ID]

Abstract—The recent years have witnessed the emerging of vector quantization (VQ) techniques for efficient similarity search. VQ partitions the feature space into a set of codewords and encodes data points as integer indices using the codewords. Then the distance between data points can be efficiently approximated by simple memory lookup operations. By the compact quantization, the storage cost, and searching complexity are significantly reduced, thereby facilitating efficient large-scale similarity search. However, the performance of several celebrated VQ approaches degrades significantly when dealing with noisy data. In addition, it can barely facilitate a wide range of applications as the distortion measurement only limits to ℓ_2 norm. To address the shortcomings of the squared Euclidean ($\ell_{2,2}$ norm) loss function employed by the VQ approaches, in this paper, we propose a novel robust and general VQ framework, named RGVQ, to enhance both robustness and generalization of VQ approaches. Specifically, a $\ell_{p,q}$-norm loss function is proposed to conduct the ℓ_p-norm similarity search, rather than the ℓ_2 norm search, and the q-th order loss is used to enhance the robustness. Despite the fact that changing the loss function to $\ell_{p,q}$ norm makes VQ approaches more robust and generic, it brings us a challenge that a non-smooth and non-convex *orthogonality constrained $\ell_{p,q}$-norm function* has to be minimized. To solve this problem, we propose a novel and efficient optimization scheme and specify it to VQ approaches and theoretically prove its convergence. Extensive experiments on benchmark data sets demonstrate that the proposed RGVQ is better than the original VQ for several approaches, especially when searching similarity in noisy data.

Index Terms—Vector quantization, similarity search, efficiency, large scale, robustness, generalization, optimization, experiment.

I. INTRODUCTION

SIMILARITY search, a.k.a., nearest neighbor (NN) search, is of great importance in various applications, such as data mining [1], machine learning [2], information retrieval [3], and etc. Formally, NN search is defined as follows: given a set $\mathcal{S}$ of points in a metric space $\mathcal{M}$ and a query point $q \in \mathcal{M}$, find the closest point in $\mathcal{S}$ to q. One straightforward way is to linearly scan $\mathcal{S}$ and compute the distance $d(q, x_i)$ between q and
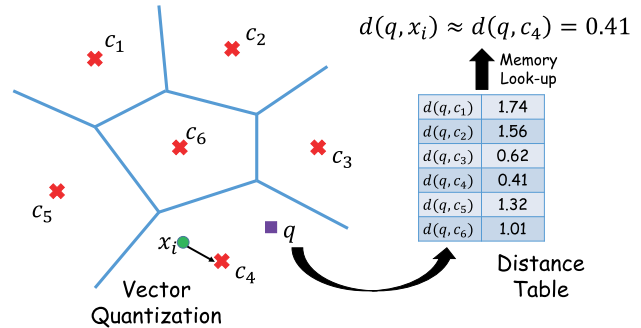
Fig. 1. Illustration of vector quantization based ANN search.

any point $x_i \in \mathcal{S}$. However, when dealing with a large-scale dataset, linear scanning is time consuming due to the expensive distance computing operations. Therefore, how to perform efficient NN search in large-scale dataset is an important and practical problem, which has drawn considerable research interest from both academia and industry in the past decades.

Considering the difficulty of exact NN search for large-scale dataset, approximate NN (ANN) search is regarded as a more practical solution which can simultaneously achieve orders of magnitude speed-ups than exact NN search and near optimal accuracy with proper designs [4]. One paradigm is to utilize the tree structure, such as k-d tree [5]. Theoretically, by recursively bi-partitioning the feature space, tree structure can reduce the frequency of distance computation to $\mathcal{O}(\log n)$. However, because of the *curse of dimensionality*, tree structure may degenerate to sub-linear complexity in high-dimensional spaces since it needs to visit too many branches [6]. Alternatively, vector quantization (VQ) emerges recently which is capable of handling high-dimensional data. Different from tree structure that reduces the number of scanned points, the aim of VQ is to speed up the exhausting distance computing. Specifically, VQ partitions the space into a set of codewords, i.e., a codebook $\mathcal{C}$ ($|\mathcal{C}| \ll |\mathcal{S}|$), and then quantizes each point x_i into the codewords. After the quantization, the feature vector of each point is no longer needed and only an integer index denoting which codeword the point is quantized into is stored. Given a query, its distance $d(q, c_j)$ to all codewords can be pre-computed and stored in a distance table. The distance $d(q, x_i)$ can be approximated by $d(q, c_{I(x_i)})$ where $I(x_i)$ denotes the index for x_i. Obviously, we can obtain $d(q, c_{I(x_i)})$ through a simple memory look-up operation using the pre-computed distance table. Although VQ requires a linearly scan of $\mathcal{S}$, the look-up operation for distance computing is much faster than the floating-point operations such that the
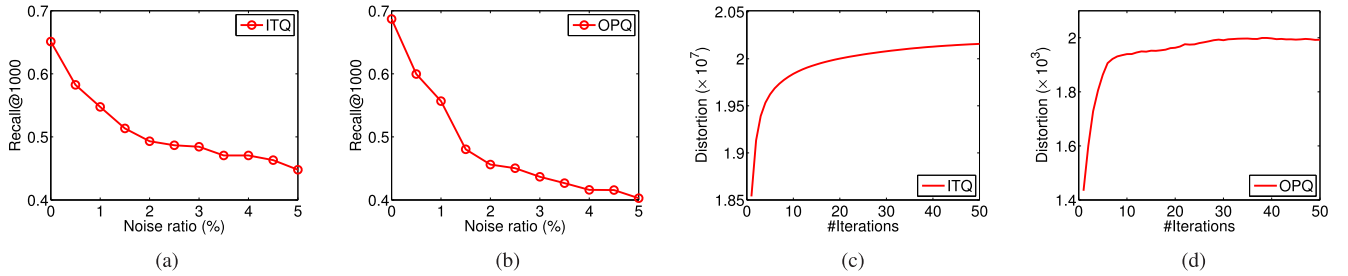
Fig. 2. Traditional VQ approaches perform bad with the noisy data and can not deal with some other similarity measures. (a) SIFT1M, 64 bits. (b) SIFT1M, 32 bits. (c) SIFT1M, 64 bits, $p = 1$. (d) SIFT1M, 32 bits, $p = 1$.

overall searching is quite efficient. Empirically, VQ takes less than 10 seconds to linearly scan a 1-billion-size dataset and it achieves real-time search if proper optimization is further employed [7].

The most representative VQ approaches include hashing-based Iterative Quantization (ITQ) [8], [9] which focuses on binary quantization, Product Quantization (PQ) [10]–[12] which segments a space into several orthogonal subspaces and the quantization is performed in each subspace independently, and Additive Quantization (AQ) [13]–[16] which constructs several independent codebooks and each point is approximated by summing up the selected codewords from each codebook. As VQ approaches can achieve extreme data compression and support efficient ANN search for large-scale dataset, they have been adopted by many applications, such as image retrieval [17]–[20] and many other tasks [21].

A. Problem Statement

The extraordinary performance and widely usage of the VQ approaches motivate as to closely investigate these algorithms. Though specific formulations may have tiny differences, these VQ approaches can be all formulated as a general problem:

$$\min_{\mathbf{R}, Q, \mathcal{C}} \sum_{i=1}^{n} \|\mathbf{x}_i \mathbf{R} - Q(\mathbf{x}_i, \mathcal{C})\|_2^2, \quad \text{s.t. } \mathbf{R}\mathbf{R}' = \mathbf{I} \quad (1)$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the d-dimensional feature vector, Q is the quantization function with the codebook $\mathcal{C}$, $\mathbf{R} \in \mathbb{R}^{d \times d}$ is a rotation matrix which can optimize the quantization [8], [10], [16], and $\mathbf{I}$ is the identity matrix. A close look at the objective function reveals that a squared ℓ_2 loss is applied to measure the distortion. But unfortunately, this sort of distance measurement comes with certain vulnerabilities. For instance, there are noise and outliers in real-world datasets but the squared loss is sensitive to them because their large distortion will dominate the sum of the squared loss [22]–[24], which may markedly degrade the quality of quantization codes. Solving this problem becomes important when we need to search nearest neighbors for data in the wild, such as Flickr images and YouTube videos, as the noises are commonly existed. To verify our observation, we carried out an experiment based on SIFT1M [25] dataset, in which the noise is manually added into the training data and we plot the ANN search performance of two representative VQ approaches, ITQ [8] and OPQ [10],

w.r.t. the noise ratio, which is shown in Fig. 2(a) and 2(b) respectively. Obviously, the performance of VQ degrades significantly in the noisy environment, even with only 1% noisy data. Secondly, existing VQ approaches work well for ℓ_2-norm similarity search, i.e., $d(\mathbf{q}, \mathbf{x}_i) = \|\mathbf{q} - \mathbf{x}_i\|_2$ because they focus on minimizing ℓ_2-norm distortion defined in Eq. (1). However, when other measurements, such as Manhattan distance $d_M(\mathbf{q}, \mathbf{x}_i) = \|\mathbf{q} - \mathbf{x}_i\|_1$ [26], are used, their optimization objective may fail to well preserve the similarity structure. In practice, the preferred measure means may need to be defined by users depending on the specific applications, which indicates that a good similarity search algorithm should be generic enough to deal with different distance measurements. Again, to demonstrate this, we plot the ℓ_1-norm distortion (i.e., $\sum_i \|\mathbf{x}_i \mathbf{R} - Q(\mathbf{x}_i, \mathcal{C})\|_1^1$) w.r.t. the number of iterations of ITQ and OPQ in Fig. 2(c) and 2(d) respectively. It can be observed that the distortion keeps increasing with more iterations, rather than decreasing, because their optimization algorithms are designed for ℓ_2-norm distortion instead of the ℓ_1-norm one. This inevitably leads to less effective quantization function, thereby resulting in worse search performance.

B. Contributions

The two problems mentioned above are important for VQ approaches from both theoretical and practical perspectives, but underestimated by the previous works. This motivates us to develop an improved VQ framework with dual goal to enhance both algorithm robustness and generalization. Recently, several works have demonstrated that the q-th order ($q < 2$, especially $q \leq 1$) of ℓ_2 loss, i.e., $\|\mathbf{x}_i \mathbf{R} - Q(\mathbf{x}_i, \mathcal{C})\|_2^q$, is less susceptible to the noise and outliers in data than the squared loss [27], [28]. In addition, according to the triangle inequality, preserving the ℓ_p-norm distance can be achieved by minimizing the ℓ_p-norm distortion, i.e., $\|\mathbf{x}_i \mathbf{R} - Q(\mathbf{x}_i, \mathcal{C})\|_p$. Therefore, in this paper, we propose a general VQ framework using a $\ell_{p,q}$-norm loss function for learning ℓ_p-norm similarity-preserving quantization function with more robustness, termed as **RGVQ**. In summary, this paper makes the following contributions:

- We put forward a new $\ell_{p,q}$-norm loss function for vector quantization based ANN search. It is robust to noise and outliers by adopting a small q (e.g., $q \leq 1$) and supports ℓ_p-norm ($p \leq 2$) similarity search with the ℓ_p-norm loss.
- To minimize the obtained orthogonality constrained $\ell_{p,q}$-norm function, a novel and efficient iterative

Notation	Description
n	the number of training samples
d	the dimension of training samples
C	the number of codebooks
k	the number of codewords in each codebook
$\mathbf{x}_i$	the i-th sample
$\mathbf{c}_j^m$	the j-th codeword in the m-th codebook
$\mathbf{R}$	a rotation matrix
p, q	positive scalars for matrix norm
I	the indexing function

optimization algorithm is proposed and its convergence property is theoretically investigated. To our best knowledge, it is the first work that provides the theoretical solution to this challenging non-smooth and non-convex problem.

- We specify our framework to several celebrated VQ approaches, including ITQ, OPQ, and AQ. Extensive experiments on benchmark datasets demonstrate the superiority of the improved approaches to the original ones.

In addition, it is worthwhile to highlight two important properties of RGVQ framework from the application perspectives:

- RGVQ is robust to the noise, enabling us to search similarity in wild data. Such an framework is favorably demanded by the applications like Internet image retrieval. Extensive image retrieval experiments on benchmarks collected from the Internet demonstrate the effectiveness.
- Our algorithm is more generic in the sense that multiple distortion measurements are implemented in one framework, allowing us to facilitate a wide range of applications in which various measurements may be requested.

II. PRELIMINARIES AND RELATED WORK

A. Vector Quantization Approaches

In this paper, we focus on three celebrated VQ approaches, Iterative Quantization (ITQ) [8], [9], (Optimized) Product Quantization (PQ) [10]–[12], and Additive Quantization (AQ) [13]–[16]. As was mentioned above, these approaches share a general learning objective presented in Eq. (1) but they have different specific formulations and basic ideas. In this section, we will introduce them in details.

ITQ focuses on binary quantization defined as $Q(\mathbf{x}_i, \mathcal{C}) = \text{sign}(\mathbf{x}_i \mathbf{R})$, where $\text{sign}(x) = 1$ if $x > 0$ or -1 otherwise. Its learning objective is to find the optimal rotation matrix $\mathbf{R}$ to minimize the distortion between the original features and the binary embedding as follows:

$$\min_{\mathbf{R}} \ \mathcal{O}_{\text{ITQ}} = \sum_{i=1}^n \|\mathbf{x}_i \mathbf{R} - \text{sign}(\mathbf{x}_i \mathbf{R})\|_2^2, \quad \text{s.t. } \mathbf{R}\mathbf{R}' = \mathbf{I} \quad (2)$$

After the binarization, the original distance is approximated by the Hamming distance which is defined as the number of different bits between binary codes (a.k.a., hashcodes), and its computation can be accelerated by either memory lookup or the bit operations (like bit XOR), both of which are efficient.

PQ is a k-means clustering like quantization approach. Its basic idea is to cluster the samples into a set of codewords $\mathcal{C} = \{\mathbf{c}_j\}_{j=1}^k$ and the distance $d(\mathbf{q}, \mathbf{x}_i)$ can be approximated by $d(\mathbf{q}, \mathbf{c}_{I(\mathbf{x}_i)})$ which can be pre-computed and stored in a distance table. Obviously, increasing the codebook size (i.e., k) can partition the space more finely, which improves the distance approximation accuracy. In the extreme case where $k = n$, each training sample is quantized to itself such that the distance is precisely approximated. However, when k is large, computing the distance table, i.e., $d(\mathbf{q}, \mathbf{c}_j)$, becomes a time-consuming step. Therefore, it is preferable to construct a large codebook while the extra distance computation is not heavy. To address this issue, PQ proposes to partition the space into C orthogonal subspace and the quantization is performed in each subspace independently. Specifically, after the segmentation, each subspace is $d_s = d/C$ dimension and the final codeword is constructed by the concatenation of the sub-codeword from each subspace, i.e., $\mathbf{c}_{I(\mathbf{x}_i)} = [\mathbf{c}_{I_1(\mathbf{x}_i)}^1, \dots, \mathbf{c}_{I_C(\mathbf{x}_i)}^C]$ where $\mathbf{c}_j^m \in \mathcal{C}_m$ is a codeword from the m-th subspace. Suppose there are k codewords for each subspace, i.e., $|\mathcal{C}_m| = k$, the total number of codewords in the original space is k^C, which is extremely large. In addition, the distance is approximated by $\|\mathbf{x}_i - \mathbf{q}\|_2^2 \approx \sum_{m=1}^C \|\mathbf{q}^m - \mathbf{c}_{I_m(\mathbf{x}_i)}^m\|_2^2$ where $\mathbf{q}_m$ is the component of $\mathbf{q}$ in the m-th subspace. In this way, the distance table, i.e., $d(\mathbf{q}^m, \mathbf{c}_j^m)$, can be computed in each subspace independently, which reduces the total complexity to $\mathcal{O}(C \cdot k \cdot \frac{d}{C}) = \mathcal{O}(kd)$. For good ANN results, PQ minimizes the distortion between the original features and the codewords:

$$\min_{\mathbf{R}, \mathbf{c}_j^m, I_m} \mathcal{O}_{\text{PQ}} = \sum_{i=1}^n \|\mathbf{x}_i \mathbf{R} - [\mathbf{c}_{I_1(\mathbf{x}_i)}^1, \dots, \mathbf{c}_{I_C(\mathbf{x}_i)}^C]\|_2^2,$$
$$\text{s.t. } \mathbf{R}\mathbf{R}' = \mathbf{I} \quad (3)$$

where $\mathbf{R}$ is to optimize the quantization, whose effectiveness has been demonstrated by several works [10], [16]. Suppose $\hat{\mathbf{x}}_i = \mathbf{x}_i \mathbf{R}$ is the rotated data, in PQ, each sub-codebook $\mathcal{C}_m$ is learned by k-means clustering in the m-th subspace over $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ and the quantization function is defined as $Q(\mathbf{x}_i, \mathcal{C}) = [\mathbf{c}_{I_1(\mathbf{x}_i)}^1, \dots, \mathbf{c}_{I_C(\mathbf{x}_i)}^C]$ where $I_m(\mathbf{x}_i) = \text{argmin}_j \|\hat{\mathbf{x}}_i^m - \mathbf{c}_j^m\|_2^2$. At the searching/testing phase, the query $\mathbf{q}$ is also rotated by $\mathbf{R}$.

AQ is motivated by the multi-codebook idea of PQ. Different from PQ which constructs the final codeword by concatenation, AQ constructs the final codeword by the summation of sub-codewords. In addition, the sub-codeword in PQ is d_s dimension while AQ directly constructs sub-codewords in the original space which leads to d-dimensional sub-codeword. Formally, AQ constructs C codebooks $\mathcal{C}_m = \{\mathbf{c}_j^m\}_{j=1}^k$ where $\mathbf{c}_j^m \in \mathbb{R}^d$. The quantization function is defined as $Q(\mathbf{x}_i, \mathcal{C}) = \sum_{m=1}^C \mathbf{c}_{I_m(\mathbf{x}_i)}^m$. Because we have $\|\mathbf{q} - \mathbf{x}_i\|_2^2 = \|\mathbf{q}\|_2^2 + \|\mathbf{x}_i\|_2^2 - 2\langle \mathbf{q}, \mathbf{x}_i \rangle$, it is straightforward to approximate the distance by $\|\mathbf{q} - \mathbf{x}_i\|_2^2 \approx \|\mathbf{q}\|_2^2 + \|\mathbf{x}_i\|_2^2 - 2\sum_{m=1}^C \langle \mathbf{q}, \mathbf{c}_{I_m(\mathbf{x}_i)}^m \rangle$ using the pre-computed table $\langle \mathbf{q}, \mathbf{c}_j^m \rangle_{\forall m, j}$. Analogous to PQ, AQ can also construct k^C codewords in the original space and the

complexity to construct the distance table is only $\mathcal{O}(Ckd)$. Moreover, AQ attempts to minimize the distortion as follows:

$$\min_{\mathbf{R}, \mathbf{c}_j^m, I_m} \mathcal{O}_{AQ} = \sum_{i=1}^{n} \|\mathbf{x}_i \mathbf{R} - \sum_{m=1}^{C} \mathbf{c}_{I_m(\mathbf{x}_i)}^m\|_2^2, \quad \text{s.t.} \ \mathbf{R}\mathbf{R}' = \mathbf{I} \quad (4)$$

Theoretically, AQ can be regarded as the generalization of PQ by removing the orthogonal constraints on the sub-codebooks. Because more flexible codeword combination is given, smaller distortion and more accurate quantization can be achieved such that AQ performs better than PQ to some extent [14], [16].

Based on the VQ approaches, the storage cost is markedly compressed and the distance computation is accelerated. For example, in PQ, if we set $k = 256$ for each sub-codebook, it requires only 1 byte (8 bits) to store the index $I_m(\mathbf{x}_i)$ for a sample. Even if we use 8 sub-codebooks, the memory cost for one sample is only 8 bytes (64 bits) and the storage for all sub-codebooks are independent from the data. Therefore, only 8 gigabytes are required for a 1-billion-size dataset, which can be can be easily handled by only one single machine. During searching, the complexity to compute the distance table is $\mathcal{O}(Ckd)$ at most. Since we have $Ck \ll n$ (in the above example, $Ck = 8 \times 256 = 2,048$ and $n = 1$B), the complexity is almost ignorable. When computing $d(\mathbf{x}_i, \mathbf{q})$, only C memory look-up operations and $C - 1$ addition operations are required which is far fewer than directly computing $d(\mathbf{x}_i, \mathbf{q})$ in an element-wise way, especially for high-dimensional data.

B. Other Related Works

The main focus of this paper is on enhancing the robustness and generalization of existing VQ approaches by introducing the $\ell_{p,q}$-norm loss function to evaluate the quantization distortion. However, we notice several works for some other problems are related to our work [28]–[30]. Therefore, it is necessary to introduce them and discuss their difference. Generally, their difference comes from three folds. Firstly, the tasks are different. We focus on vector quantization for efficient and general similarity search including image retrieval, while the others mainly focus on tasks like dictionary learning [28], representation learning [29], and projection learning [30]. In fact, we are the first to introduce a robust loss function and consider the generalization simultaneously in the field of VQ, which motivates us to propose the $\ell_{p,q}$-norm loss. Moreover, we specify the general framework to three celebrated VQ approaches, ITQ, PQ, and AQ, which is also very useful in practice. Secondly, the formulations are different. Generally, the $\ell_{2,q}$-norm (especially $\ell_{2,1}$-norm) loss is adopted in many robust learning approaches [28]–[30]. But the $\ell_{p,q}$-norm is more general and complicated than them and it seems that it is difficult to directly apply their optimization algorithm to $\ell_{p,q}$-norm loss. Some works also consider the ℓ_p-norm term. For example, the ℓ_1-norm term is considered in sparse coding [31] and the ℓ_p-norm is considered in [28]. But it should be pointed out that our formulation employs ℓ_p-norm to evaluate the reconstruction distortion while their approaches use it only as a sparsity regularization term for coefficients which element-wise decoupled. Obviously, our formulation is more difficult and general especially when coupling with the

q-th order upon the ℓ_p norm and the orthogonality constraint. Thirdly, the solutions are different. As stated, our problem is an orthogonality constrained $\ell_{p,q}$-norm minimization problem. Unlike some works considering parts of the problem, e.g., $\ell_{2,q}$-norm minimization is considered in [28], we systematically solve the general problem and provide the theoretical analysis for the solution. In addition, in Section V, we demonstrate that that the optimization algorithm is consistently effective and efficient under different settings.

III. THE PROPOSED FRAMEWORK

A. Overall Objective Function

The first goal of this paper is to enhance the robustness of VQ approaches. In the current framework, squared Euclidean (i.e., $\ell_{2,2}$-norm) loss is adopted. In fact, because of the square operation, the loss function tends to assign large weight to large-loss samples. However, in practice, the large loss is often caused by noise and outliers. The loss function, in such a situation, will focus on the noise but fail to capture the intrinsic structure of samples, i.e., it is sensitive to noise, which has been empirically demonstrated in Fig. 2(a) and 2(b). To address this issue, we should reduce the weight of large-loss samples. In this paper, we propose to replace the squared loss by the q-th order loss. It has been demonstrated in several literatures [24], [32]–[34] that the loss function is more robust (less sensitive) to the noise and outliers in data in case of $q < 2$, especially $q \leq 1$. Motivated by this idea, we reformulate Eq. (1) from squared loss into q-th order loss as:

$$\min_{\mathbf{R}, Q, \mathcal{C}} \mathcal{O}_{RQ} = \sum_{i=1}^{n} \|\mathbf{x}_i \mathbf{R} - Q(\mathbf{x}_i, \mathcal{C})\|_2^q, \quad \text{s.t.} \ \mathbf{R}\mathbf{R}' = \mathbf{I} \quad (5)$$

where $q < 2$. It is not difficult to observe the following fact. When $q > 1$, the objective prefers to decrease the distortion of large-loss entries because it is obvious that the larger x (the distortion) is, the larger $|x^q - (x - \Delta x)^q|$ (the change in loss) is if Δx is identical, which indicates the loss is encouraged to fit the noisy data. On the other hand, when $q \leq 1$, the situation is different where the loss focuses more on the small-loss entries which are normal data. In this way, we can enhance the robustness of functions by setting $q < 2$, especially $q \leq 1$.

Although the $\ell_{2,q}$-norm loss function is more robust to the noise, it is still questionable whether it works well for the other similarity/distance measurements, like Manhattan distance. In fact, just like the results shown in Fig. 2(c) and 2(d), the ℓ_2-norm loss may fail when dealing with ℓ_1-norm based similarity search. To address this issue, we firstly revisit one important theoretical building block of VQ, i.e., the triangle inequality:

$$\begin{aligned} | \|\mathbf{x} - \mathbf{y}\|_p &- \|Q(\mathbf{x}) - Q(\mathbf{y})\|_p | \\ &\leq K_1 \|\mathbf{x} - \mathbf{y} - Q(\mathbf{x}) + Q(\mathbf{y})\|_p \\ &\leq K_2 (\|\mathbf{x} - Q(\mathbf{x})\|_p + \|\mathbf{y} - Q(\mathbf{y})\|_p) \end{aligned} \quad (6)$$

where $\|\mathbf{x}\|_p = (\sum_j |x_j|^p)^{\frac{1}{p}}$ is the ℓ_p-norm of a vector, $K_1 = K_2 = 1$ for normal vector norm (i.e., $p > 1$) and they are some constants for quasi-norm (i.e., $0 < p \leq 1$), and $Q(\mathbf{x})$

denotes the quantization result of $\mathbf{x}$. The first term denotes the distance approximation error between the original feature based distance ($\|\mathbf{x} - \mathbf{y}\|_p$) and the quantized feature based distance ($\|Q(\mathbf{x}) - Q(\mathbf{y})\|_p$), in which we expect the error to be as small as possible, i.e., the distance approximation using the quantized vectors is more accurate. The last term is exactly the distortion caused by the quantization function ($\|\mathbf{x} - Q(\mathbf{x})\|_p$). Obviously, the distortion provides an upper bound for the distance approximation. Therefore, decreasing the distortion leads to more accurate distance approximation and further results in better ANN performance [8], [10], [15]. Fortunately, based on the triangle inequality, it is straightforward to observe that learning ℓ_p-norm similarity preserving quantization function can be achieved by minimizing the ℓ_p-norm distortion. In the extreme case where the distortion is 0 (i.e., $Q(\mathbf{x}) = \mathbf{x}$), the distance is perfectly approximated (i.e., $\|\mathbf{x} - \mathbf{y}\|_p = \|Q(\mathbf{x}) - Q(\mathbf{y})\|_p$). Theoretically, existing VQ approaches, including ITQ, PQ, and AQ, can be regarded as a special case ($p = 2$) of our scheme. To clarify it, we can rewrite the loss in Eq. (5) from the ℓ_2-norm loss to the ℓ_p-norm loss, leading to the overall objective function of RGVQ:

$$\min_{\mathbf{R},Q,\mathcal{C}} \mathcal{O}_{\text{RG}} = \|\mathbf{XR} - Q(\mathbf{X},\mathcal{C})\|_{p,q}^q = \sum_{i=1}^{n} \|\mathbf{x}_i\mathbf{R} - Q(\mathbf{x}_i,\mathcal{C})\|_p^q$$
$$\text{s.t. } \mathbf{RR}' = \mathbf{I} \tag{7}$$

where $\|\mathbf{A}\|_{p,q}$ denotes the entrywise (in row) matrix $\ell_{p,q}$ norm of matrix $\mathbf{A}$. So far, we derive the $\ell_{p,q}$-norm loss function for RGVQ framework from the original $\ell_{2,2}$-norm loss of VQ.

B. Optimization Algorithm

The motivation of changing the $\ell_{2,2}$-norm loss into the $\ell_{p,q}$-norm loss is clear and reasonable, which makes VQ more robust to noisy data and generalizable for different distance measurements. However, it is challenging to minimize the obtained orthogonality constrained $\ell_{p,q}$-norm function because it becomes a non-smooth and non-convex optimization problem when $p \leq 1$ or $q \leq 1$. Solving this problem is much more difficult than minimizing the $\ell_{2,2}$-norm in the original VQ, for which many solutions are available [8], [10], [14]. To solve it, we propose an efficient optimization algorithm shown below.

It should be noticed that there are a rotation matrix $\mathbf{R}$, quantization function Q and the codebook $\mathcal{C}$ in the objective function, and it is very difficult, if not impossible, to optimize them as a whole. Therefore, following the traditional VQ framework, we adopt an iterative optimization scheme to update any of them while keeping the others fixed as follows.

Update $\mathbf{R}$. This is the most difficult part in the entire solution, which is also an important theoretical contribution of this paper. The $\ell_{p,q}$-norm is neither smooth nor convex, and meanwhile, the orthogonality constraint limits the feasible set, therefore making the problem more difficult. First, we denote $\mathbf{y}_i = Q(\mathbf{x}_i, \mathcal{C})$ as the quantized vector, which is fixed when updating $\mathbf{R}$. Then, to solve the problem, we rewrite the

complicated $\ell_{p,q}$-norm loss into a weighted $\ell_{2,2}$-norm loss as:

$$\min_{\mathbf{RR}'=\mathbf{I}} \mathcal{O} = \sum_{i=1}^{n} \|\mathbf{w}_i \circ (\mathbf{y}_i - \mathbf{x}_i\mathbf{R})\|_2^2 = \|\mathbf{W} \circ (\mathbf{Y} - \mathbf{XR})\|_F^2 \tag{8}$$

where $\mathbf{X} = [\mathbf{x}_1; \ldots; \mathbf{x}_n] \in \mathbb{R}^{n \times d}$ represent the original training vectors, $\mathbf{Y} = [\mathbf{y}_1; \ldots; \mathbf{y}_n] \in \mathbb{R}^{n \times d}$ are the quantized vectors, $\mathbf{W} = [\mathbf{w}_1; \ldots; \mathbf{w}_n] \in \mathbb{R}^{n \times d}$ is the weighting matrix, $\|\cdot\|_F$ is the Frobenius norm of a matrix, and "$\circ$" denotes the element-wise multiplication operation. Specifically, the elements of the weighting matrix in our algorithm are computed as follows:

$$f_i = \|\mathbf{y}_i - \mathbf{x}_i\mathbf{R}\|_p^{q-p}, \quad g_{ij} = |y_{ij} - \mathbf{x}_i\mathbf{R}_{*j}|^{p-2}$$
$$w_{ij} = (f_i g_{ij})^{0.5} \tag{9}$$

Based on the above definition, it is easy to verify that Eq. (8) is numerically equivalent to Eq. (7). Now if we keep $\mathbf{W}$ fixed, the problem is transformed into a weighted $\ell_{2,2}$-norm problem. Fortunately, solving this problem is much easier than solving the original as it is smooth and convex. The only challenge left in this problem is to address the orthogonality constraint which limits the feasible set. In this paper, we adopt the framework proposed by Wen and Yin [35] which is a gradient-descent based algorithm but takes the orthogonality constraint into consideration. In particular, we first compute the derivative of $\mathcal{O}$ w.r.t. the variable $\mathbf{R}$ as:

$$\mathbf{G} = \frac{\partial \mathcal{O}}{\partial \mathbf{R}} = \mathbf{X}'(\mathbf{W} \circ \mathbf{W} \circ (\mathbf{XR} - \mathbf{Y})) \tag{10}$$

In the conventional gradient descent method, we just need to update $\mathbf{R}$ along the direction given by the derivative with a tiny step. However, this strategy will violate the orthogonality constraint which moves $\mathbf{R}$ out of the feasible set. Therefore, more operations on the gradient are required to address the orthogonality constraint. Following the framework [35], a skew-symmetric matrix is constructed based on $\mathbf{G}$ as below:

$$\mathbf{A} = \mathbf{GR}' - \mathbf{RG}' \tag{11}$$

Having obtained $\mathbf{G}$ and $\mathbf{A}$, the following step is to search the next point using the Crank-Nicolson-like scheme [36], [37]:

$$\mathbf{R}_{t+1} = \mathbf{R}_t - \tau\mathbf{A}\left(\frac{\mathbf{R}_{t+1} + \mathbf{R}_t}{2}\right) \tag{12}$$

where τ is a tiny step size. The solution to the problem is:

$$\mathbf{R}_{t+1} = \left(\mathbf{I} + \frac{\tau}{2}\mathbf{A}\right)^{-1}\left(\mathbf{I} - \frac{\tau}{2}\mathbf{A}\right)\mathbf{R}_t \tag{13}$$

The objective function value in Eq. (8) will keep decreasing w.r.t. the updating rule in Eq. (13) until the stationary point is achieved and $\mathbf{R}_{t+1}$ also satisfies the orthogonality constraint. Please refer to [35] for the detailed proof. We update $\mathbf{R}$ by fixing $\mathbf{W}$ as we can see $\mathbf{W}$ depends on $\mathbf{R}$. Therefore, we can update $\mathbf{R}$ and $\mathbf{W}$ in an iterative manner. This strategy can decrease the loss in Eq. (7), whose proof will be given later.

Update Q and $\mathcal{C}$. When the rotation matrix $\mathbf{R}$ is fixed, we can update the quantization function Q and the corresponding codebook $\mathcal{C}$. In this paper, we focus on three celebrated VQ approaches, ITQ, PQ, and AQ, which achieve state-of-the-art ANN performance, and therefore we specify our RGVQ framework into these approaches. As they have

different formulations and codebook construction methods, the updating rules should be different, each being discussed below. For simplicity, we denote $\hat{\mathbf{x}}_i = \mathbf{x}_i \mathbf{R}$ in the following derivation.

ITQ. ITQ focuses on binary quantization and the sign function is adopted, so it does not have a codebook $\mathcal{C}$. Thus, extending it from the $\ell_{2,2}$ normal loss in the original VQ to the $\ell_{p,q}$ norm loss in RGVQ is the easiest one. Moreover, we can observe that the quantization in ITQ is element-wise decoupled even with the $\ell_{p,q}$-norm loss. Therefore, the quantized vector is $y_{ij} = \text{sign}(\hat{x}_{ij})$, which is the quantization function for ITQ.

PQ. In the original PQ with $\ell_{2,2}$-norm loss, it only requires performing k-means clustering in each subspace to learn each sub-codebook $\mathcal{C}_m$ and the corresponding function I_m. In the RGVQ framework with the $\ell_{p,q}$-norm loss, its loss function for this step is more complicated, which is written as follows:

$$\min_{\mathbf{c}_j^m, I_m} \mathcal{O}_{\text{RGPQ}} = \sum_{i=1}^{n} \left(\sum_{m=1}^{C} \sum_{j=1}^{d_s} (\hat{x}_{ij}^m - c_{I_m(\mathbf{x}_i)j}^m)^p \right)^{\frac{q}{p}} \quad (14)$$

If $p = q$ which is the case of original PQ, the problem can be solved in each subspace attributable to decoupled subspaces. However, in the general framework, we have $p \neq q$ in most cases, which makes the problem more complicated because the $(\cdot)^{\frac{1}{p}}$ operation couples each subspace, i.e., the quantization loss in one subspace has influence on the decision in the other subspaces. To simplify the problem, we also adopt the weighting method in Eq. (9) and rewrite the loss function as:

$$\min_{\mathbf{c}_j^m, I_m} \mathcal{O} = \sum_{i=1}^{n} f_i \left(\sum_{m=1}^{C} \|\hat{\mathbf{x}}_i^m - \mathbf{c}_{I_m(\mathbf{x}_i)}^m\|_p^p \right) \quad (15)$$

Obviously, after the transformation, each subspace becomes decoupled. To clarify it, we can rewrite Eq. (15) as follows:

$$\min_{\mathbf{c}_j^m, I_m} \mathcal{O} = \sum_{m=1}^{C} \mathcal{O}_m = \sum_{m=1}^{C} \left(\sum_{i=1}^{n} f_i \|\hat{\mathbf{x}}_i^m - \mathbf{c}_{I_m(\mathbf{x}_i)}^m\|_p^p \right) \quad (16)$$

Therefore, in each subspace we solve the problem below:

$$\min_{\mathbf{c}_j^m, I_m} \mathcal{O}_m = \sum_{i=1}^{n} f_i \|\hat{\mathbf{x}}_i^m - \mathbf{c}_{I_m(\mathbf{x}_i)}^m\|_p^p \quad (17)$$

which leads to a *Minkowski weighted kmeans clustering* [38] problem similar to the original kmeans clustering but with ℓ_p-norm loss and weighted samples. It can be easily solved in the EM framework by iteratively updating the index $I_m(\mathbf{x}_i) = \text{argmin}_j \|\hat{\mathbf{x}}_i^m - \mathbf{c}_j^m\|$ and the centers $\mathbf{c}_j^m$ by the simple gradient descent algorithm. Such a procedure can be performed in each subspace m independently. In this way, the function value in Eq. (15) is decreased until convergence is achieved, which also decreases the function value $\mathcal{O}_{\text{RGPQ}}$ in Eq. (14).

AQ. In the RGVQ framework, the objective function to update Q and C of AQ with $\ell_{p,q}$-norm loss is written as below:

$$\min_{\mathbf{c}_j^m, I_m} \mathcal{O}_{\text{RGAQ}} = \sum_{i=1}^{n} \|\hat{\mathbf{x}}_i - \sum_{m=1}^{C} \mathbf{c}_{I_m(\mathbf{x}_i)}^m\|_p^q \quad (18)$$

Algorithm 1 Optimization Algorithm for RGVQ

Input: Training data $\mathbf{X}$; Parameters $p \leq 2$ and $q \leq p$;
Output: Orthogonal matrix $\mathbf{R}$; Codebooks $\mathcal{C}_m$;
 1: Initialize $\mathbf{R} = \mathbf{I}$, $\hat{\mathbf{x}}_i = \mathbf{x}_i \mathbf{R}$;
 2: **repeat**
 3:     Update weights by Eq. (9);
 4:     For ITQ: update $y_{ij} = \text{sign}(\mathbf{x}_i \mathbf{R}_{*j})$;
 5:     For PQ: solve Eq. (17) for each subspace by Minkowski weighted kmeans clustering [38];
 6:     For AQ: solve Eq. (19) by sequential residual minimization [13], [16] for each sub-codebook;
 7:     Compute quantized vector $\mathbf{y}_i$ with current Q;
 8:     Update weights by Eq. (9);
 9:     Update $\mathbf{R}$ by Eq. (13) and $\hat{\mathbf{x}}_i = \mathbf{x}_i \mathbf{R}$;
10: **until** Convergence.
11: Return $\mathbf{R}$ and $\mathcal{C}_m$;

In order to simplify the problem, we also adopt the weighting method mentioned before, which leads to the following loss:

$$\min_{\mathbf{c}_j^m, I_m} \mathcal{O} = \sum_{i=1}^{n} f_i \|\hat{\mathbf{x}}_i - \sum_{m=1}^{C} \mathbf{c}_{I_m(\mathbf{x}_i)}^m\|_p^p \quad (19)$$

To solve this problem, we adopt the sequential learning scheme [13], [16] which is widely utilized in many optimization problems, such as matching pursuit [39], sparse coding [40], and binary learning [41]. In particular, each sub-codebook $\mathcal{C}_m$ is optimized to minimize the residual sequentially by fixing the other sub-codebooks. Denote the residual vector as $\mathbf{r}_i^m = \hat{\mathbf{x}}_i - \sum_{m' \neq m} \mathbf{c}_{I_{m'}(\mathbf{x}_i)}^{m'}$. When the other sub-codebooks are fixed, the problem w.r.t. $\mathcal{C}_m$ is reduced to:

$$\min_{\mathbf{c}_j^m, I_m} \mathcal{O}_m = \sum_{i=1}^{n} f_i \|\mathbf{r}_i - \mathbf{c}_{I_m(\mathbf{x}_i)}^m\|_p^p \quad (20)$$

which is a Minkowski weighted kmeans clustering, of which the updating rules for $\mathbf{c}_j^m$ and I_m are introduced in PQ. In this way, we can repeat the residual vector computing and sub-codebook updating for each sub-codebooks until convergence.

IV. THEORETICAL ANALYSIS

A. Convergence Analysis

In the above section, we introduce how to optimize the challenging $\ell_{p,q}$-norm loss defined by Eq. (7) in the specific situations of ITQ, PQ, and AQ, which is summarized in Algorithm 1. To simplify the complicated problem, we propose a weighting method shown in Eq. (9) and optimize the transformed problems in Eq. (8), (15), and Eq. (19). From the definition of the weights in Eq. (9), it can be observed that the weights are related to the variables $\mathbf{R}$, $\mathcal{C}_m$, and I_m which are to be optimized. In our algorithm, we iteratively update the weights and the variables by fixing the other one. However, it is not easy to figure out why decreasing the transformed loss can decrease the original loss in Eq. (7) since they are not strictly equivalent. In this section, we will theoretically and rigorously prove that the loss function in Eq. (7) is non-increasing at each iteration of Algorithm 1, which implies that the algorithm can reach a stationary point of Eq. (7) finally.

At the first of the proof, we introduce the following lemma:

Lemma 1: Given any $a > 0$ and $0 < b \leq a$, for $\forall x \geq 0$, we have the inequality: $ax^b - bx^a + b - a \leq 0$.

Proof 1: Denote $c = b/a$ and $f(x) = x^c - cx + c - 1$. Apparently, $f(1) = 0$. Then, we have $f'(x) = cx^{c-1} - c$, leading to $f'(1) = 0$. In addition, $f''(x) = c(c-1)x^{c-2} \leq 0$ when $x \geq 0$ because $0 < c \leq 1$. This implies $f'(x) \geq 0$ $\forall x \in [0, 1]$ and $f'(x) \leq 0$ when $x > 1$. Therefore, $f(x) \leq f(1) = 0$. Finally, we can obtain $af(x^a) = ax^b - bx^a + b - a \leq 0$. $\square$

Based on Lemma 1, we can prove the following theorem:

Theorem 1: The objective function $\mathcal{O}_{\mathrm{RG}}$ in Eq. (7) is non-increasing under the updating rules for $\mathbf{R}$ in Eq. (13), and $\mathcal{C}_m$ and I_m which can minimize Eq. (15) and (19).

Proof 2: Let $\mathbf{S} = \mathbf{Y} - \mathbf{X}\mathbf{R}_t$, $\mathbf{Z} = \mathbf{Y} - \mathbf{X}\mathbf{R}_{t+1}$, we have:

$$\mathcal{O}_{\mathrm{RG}}^t = \sum_{i=1}^n (\sum_{j=1}^d |s_{ij}|^p)^{\frac{q}{p}}, \quad \mathcal{O}_{\mathrm{RG}}^{t+1} = \sum_{i=1}^n (\sum_{j=1}^d |z_{ij}|^p)^{\frac{q}{p}} \quad (21)$$

Based on the proof in [35], we know that the updating rule in Eq. (13) can decrease the value of $\mathcal{O}$ in Eq. (8), i.e., we have

$$\sum_{ij} f_i g_{ij} z_{ij}^2 \leq \sum_{ij} f_i g_{ij} s_{ij}^2. \quad (22)$$

Now if we set $a = 2$, $b = p$, x will be $|z_{ij}|/|s_{ij}|$. Based on the Lemma 1 above, we can obtain the following inequalities

$$2(\frac{|z_{ij}|}{|s_{ij}|})^p - p(\frac{|z_{ij}|}{|s_{ij}|})^2 + p - 2 \leq 0$$

$$\Rightarrow |z_{ij}|^p - \frac{p}{2}|s_{ij}|^{p-2}|z_{ij}|^2 \leq |s_{ij}|^p - \frac{p}{2}|s_{ij}|^{p-2}|s_{ij}|^2$$

$$\Rightarrow \sum_{ij} f_i(|z_{ij}|^p - \frac{p}{2}g_{ij}z_{ij}^2) \leq \sum_{ij} f_i(|s_{ij}|^p - \frac{p}{2}g_{ij}s_{ij}^2) \quad (23)$$

Combining inequalities (22) with (23) will bring us

$$\sum_i f_i \|\mathbf{z}_i\|_p^p = \sum_{ij} f_i |z_{ij}|^p \leq \sum_{ij} f_i |s_{ij}|^p = \sum_i f_i \|\mathbf{s}_i\|_p^p \quad (24)$$

Denote $a = p$, $b = q$, and $x = \|\mathbf{z}_i\|_p / \|\mathbf{s}_i\|_p$, then we get

$$p(\frac{\|\mathbf{z}_i\|_p}{\|\mathbf{s}_i\|_p})^q - q(\frac{\|\mathbf{z}_i\|_p}{\|\mathbf{s}_i\|_p})^p + q - p \leq 0$$

$$\Rightarrow \|\mathbf{z}_i\|_p^q - \frac{q}{p}\|\mathbf{s}_i\|_p^{q-p}\|\mathbf{z}_i\|_p^p \leq \|\mathbf{s}_i\|_p^q - \frac{q}{p}\|\mathbf{s}_i\|_p^{q-p}\|\mathbf{s}_i\|_p^p$$

$$\Rightarrow \sum_i (\|\mathbf{z}_i\|_p^q - \frac{q}{p}f_i\|\mathbf{s}_i\|_p^p) \leq \sum_i (\|\mathbf{s}_i\|_p^q - \frac{q}{p}f_i\|\mathbf{s}_i\|_p^p) \quad (25)$$

Again, if we combine inequalities (24) with (25), we obtain

$$\mathcal{O}_{\mathrm{RG}}^{t+1} = \sum_i \|\mathbf{z}_i\|_p^q \leq \sum_i \|\mathbf{s}_i\|_p^q = \mathcal{O}_{\mathrm{RG}}^t \quad (26)$$

which means $\mathcal{O}_{\mathrm{RG}}$ in Eq. (7) is non-increasing w.r.t. Eq. (13).

Denote $\mathbf{S} = Q^t(\mathbf{X}, \mathcal{C}^t) - \mathbf{X}\mathbf{R}$, $\mathbf{Z} = Q^{t+1}(\mathbf{X}, \mathcal{C}^{t+1}) - \mathbf{X}\mathbf{R}$. We can also have Eq. (21). In addition, by minimizing the loss function value in Eq. (15) and (19), we can obtain Eq. (24) directly. Then together with Eq. (25) we obtain Eq. (26), which indicates that $\mathcal{O}_{\mathrm{RG}}$ in Eq. (7) is non-increasing when we update $\mathcal{C}_m$ and I_m by minimizing Eq. (15) and (19). $\square$

We have the following inequalities with the above proofs:

$$\mathcal{O}_{\mathrm{RG}}(Q_t, \mathbf{R}_t) \geq \mathcal{O}_{\mathrm{RG}}(Q_{t+1}, \mathbf{R}_t) \geq \mathcal{O}_{\mathrm{RG}}(Q_{t+1}, \mathbf{R}_{t+1}) \quad (27)$$

which states that $\mathcal{O}_{\mathrm{RG}}$ is non-increasing with Algorithm 1.

B. Complexity Analysis

Apparently, our optimization is more complicated than that of the original VQ approaches, it is worthwhile to analyze the algorithm complexity. In fact, since VQ approaches are applied to large-scale dataset, we care more about the relationship between the complexity and the training set size n. When updating $\mathbf{R}$, only the gradient computation in Eq. (10) is related to n, whose complexity is $\mathcal{O}(n)$. For ITQ, updating the binary codes requires $\mathcal{O}(n)$ time. For PQ, we need to solve C sub-problems in each subspace given by Eq. (17). In the original $\ell_{2,2}$-norm loss, updating $\mathcal{C}_m$ just needs to compute the average of samples belonging to the same cluster, which can be achieved in only one step. In our method, we have to adopt the gradient descent algorithm to update $\mathbf{c}_j^m$ which needs more steps to reach the optimum. Fortunately, we can adopt the mini-batch based stochastic gradient descent (SGD) where a small batch of training samples (e.g., 256), rather than the whole set, are required to compute the gradient in one single step. Although many steps are required, each step only utilizes a small number of samples such that reaching the optimum needs to traverse the whole set for just a few times. In our experiment, we empirically find out that when trained with 100k samples and 256 mini batch, traversing the training set once (i.e., ≈ 400 steps) can result in good performance. In fact, we can notice that the mini-batch SGD has achieved great success recently for gradient based optimization, such as deep model training [42], [43]. Wen *et al.* [44] also demonstrate that the mini-batch SGD works well for kmeans clustering loss. Therefore, training by mini-batch SGD has a comparable complexity to the original kmeans in our case. For AQ, we can also adopt the mini-batch SGD to solve the sub-problem in Eq. (20) whose complexity is $\mathcal{O}(n)$. In addition, as will be demonstrated in the experiment section, Algorithm 1 can always converge within about 200 iterations. In summary, the increase in complexity due to the use of a more complicated optimization is very limited, meaning that the overall complexity of RGVQ is comparable to that of VQ.

V. EXPERIMENT AND DISCUSSION

A. Datasets

VQ approaches are so general that can be applied to different kinds of features, including features for image [45]–[48], video [49], text [50], [51], or sensing data [52], [53]. To compare our framework with previous VQ approaches, we mainly focus on image features in the experiment below.

To demonstrate the effectiveness of RGVQ for ANN search, we adopt two widely used benchmarks for evaluation. The first benchmark is SIFT1M [25] which consists of 128-dimensional SIFT [54] descriptors. It is made up of 1 million base vectors, 10k query vectors, and 100k vectors for model training. The second dataset is GIST1M [25] containing 960-dimensional GIST [55] descriptors. This dataset contains 1 million vectors as the base, 1k vectors as query set and 100k training vectors.

As introduced in the contributions, RGVQ is robust to noise, which is favorably demanded by the real-world applications like Internet image retrieval. To validate the superiority of

TABLE II

THE STATISTICS OF DATASETS

	#database	#training	#query	#feature
SIFT1M	1m	100k	10k	128
GIST1M	1m	100k	1k	960
CIFAR-10	50k	50k	10k	512
NUS-WIDE	184k	50k	1,866	500

RGVQ to the original VQ for image retrieval, we also conduct experiments on two widely used real-world image retrieval dataset. The first dataset is CIFAR-10 [56] which has 60k images from 10 kinds of objects, such as "dog" and "truck". This dataset is a subset of 80M Tiny Image [57] which is constructed by collecting the images returned by search engines (like Google) using an object name as the query. Each image is represented as a 512-dimensional GIST feature. For this dataset, 50k images are used as the base and the other 10k images form the query set. The second dataset is NUS-WIDE [58] which is collected from the user uploaded images in Flickr. This dataset has $186,577$ images and each images is annotated by at least one of ten concepts from the users. The 500-dimensional bag-of-visual-word feature based on SIFT is utilized for image representation. 1% ($1,866$) images are used as the queries and the other as the database. From the construction methods of these real-world datasets (search engine returned images or user uploaded and annotated images), we can see they are very noisy, making them ideal benchmarks for testing the robustness of various approaches.

The statistics of the datasets are summarized in TABLE II.

B. Settings

Since the primary purpose of this paper is to enhance the robustness and the generalization of existing state-of-the-art VQ approaches, we therefore consider three representative approaches, Iterative Quantization (ITQ) [8], (Optimized) Product Quantization (PQ) [10], [12], and Additive Quantization (AQ) [13]–[15]. Specifically, we extend the original approaches from the $\ell_{2,2}$-norm based VQ framework into the proposed $\ell_{p,q}$-norm based RGVQ framework and then optimize them based on Algorithm 1. When no further statement is given and no ambiguity is triggered, we set $p = 2$ and $q = 1$ for most experiment scenarios and we denote the enhanced versions as ITQ+, OPQ+, and AQ+ respectively.

For each sample, we can adopt VQ or RGVQ approaches to quantize it into a fixed-length codes, whose length is denoted as L. When constructing the codes, the below settings are adopted. For ITQ which focuses on learning binary hashcodes, following [8], the original sample is firstly projected into a L-dimensional space by PCA and the rotation matrix $\mathbf{R} \in \mathbb{R}^{L \times L}$ is learned in the L-dimensional space. Then we use the sign function on the rotated data to get the hashcodes and the distance between a query and a sample is given by the Hamming distance (the number of different bits). For PQ and AQ, following [10], [14] the size of each sub-codebook is set as $k = 256$ such that the integer index $I_m(\mathbf{x}_i)$

needs exact 1 byte (8 bits). Therefore, to learn 64-bit codes, we should construct $C = 64/8 = 8$ sub-codebooks. Moreover, we adopt the asymmetric distance computation for computing the distance as we introduced in the previous part of this paper.

For all approaches, including both VQ and RGVQ, iterative optimization algorithms are adopted for learning quantization models. As suggested by the original literatures [10], [14], [21], their learning procedures can converge with 200 iterations. In the upcoming parts, we will show that RGVQ can also converge fast. Therefore, for all VQ and RGVQ approaches, the maximum number of iterations is consistently set to 200.

C. Robustness Study

We firstly investigate the robustness of RGVQ against the noise and outliers. Specifically, we adopt the ANN search task using the SIFT1M and GIST1M datasets. To better investigate this property we have manually added some noise to the training data. In particular, each dimension of each manually added noisy point is sampled from $100 \times \mathcal{N}(0,1)$ where $\mathcal{N}$ denotes a Gaussian distribution. Obviously, the distribution of noisy data is different from the that of the original data. A robust algorithm should pay more attention to the normal data. To understand the boundary of the algorithm, we continuously change the noise ratio (NR: the ratio between the manually added noise points and the original points), and evaluate the ℓ_2-norm similarity search performance of different approaches.

Following the settings in [10], [15], and [21], we use Recall@R as the metric to evaluate ANN search performance, which reflects the ratio between the number of the true positives in the first R retrieved points given by VQ or RGVQ approaches and the total number of the true positives in the database. More precisely, the true positives for each query are defined as the top l nearest neighbors of the query in the database by running a brute-force linear scan measured by the ℓ_p-norm distance.

The comparison between RGVQ (ITQ+, OPQ+, and AQ+) and VQ (ITQ, OPQ, and AQ) under different code lengths and noise ratios is shown in Fig. 3, 4, and 5. It can be observed that RGVQ is better than VQ at all situations, including different approaches, code lengths, and noise ratios, in terms of the Recall. On average, ITQ+, OPQ+ and AQ+ have improved the recall over ITQ, OPQ, and AQ by **12.2%**, **10.8%**, and **9.85%** when NR = 5%, demonstrating that RGVQ with $\ell_{p,q}$-norm ($q = 1$) loss is indeed more robust to the noise than the original VQ with squared loss. Moreover, it is worthwhile to point out that the results actually reveal the following properties of the proposed RGVQ framework.

Firstly, RGVQ performs observably better than VQ in most cases even when applying to the original dataset where no manual noise is added to the training data (i.e., NR = 0). The major reason is that the data are from the real-world dataset, on which the noises and outliers have existed. Therefore, it turns out that noisy data and outliers in the real-world dataset are indeed influential in the performance of VQ because their large errors may dominate the total distortion due to the
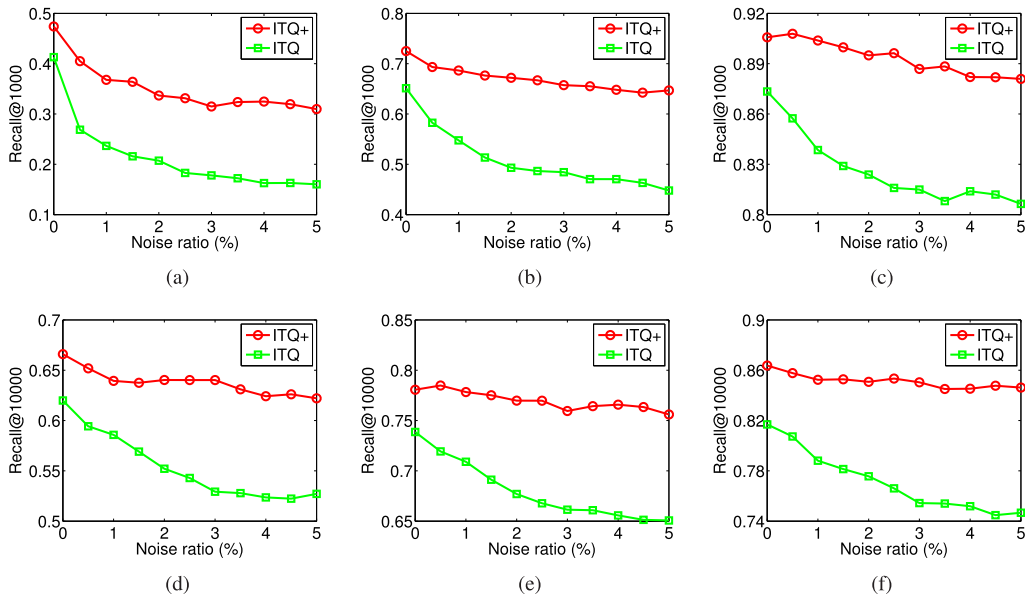
Fig. 3. Performance comparison between ITQ+ and ITQ w.r.t. the noise ratio. We set $l = 10$ for ground truth. (a) SIFT1M, 32 bits. (b) SIFT1M, 64 bits. (c) SIFT1M, 128 bits. (d) GIST1M, 32 bits. (e) GIST1M, 64 bits. (f) GIST1M, 128 bits.
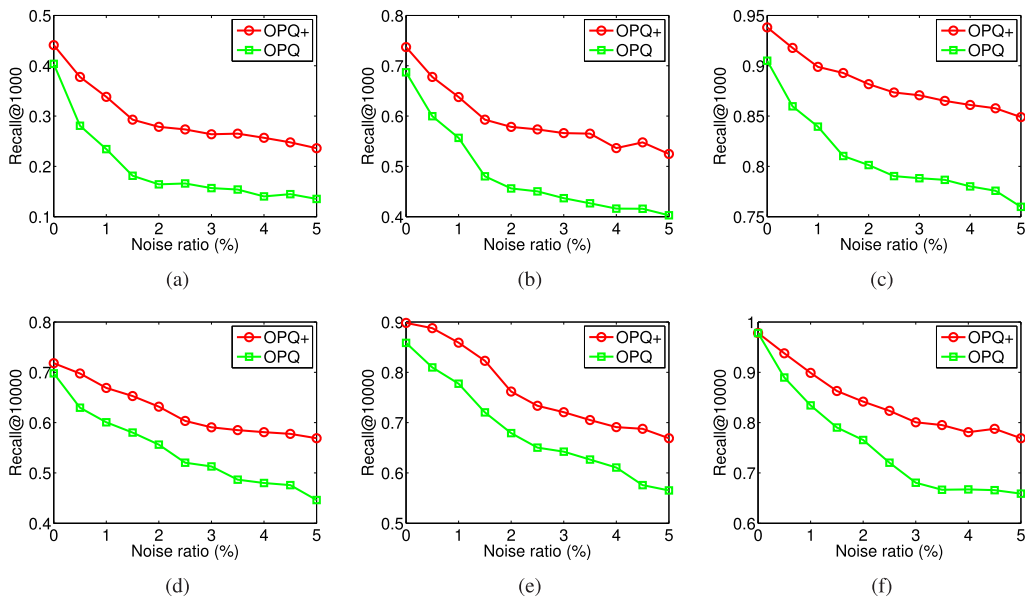

Fig. 4. Performance comparison between OPQ+ and OPQ w.r.t. the noise ratio. We set $l = 100$ for ground truth. (a) SIFT1M, 16 bits. (b) SIFT1M, 32 bits. (c) SIFT1M, 64 bits. (d) GIST1M, 16 bits. (e) GIST1M, 32 bits. (f) GIST1M, 64 bits.

squared loss. In contrast, in RGVQ, we adopt the q-th ($q < 2$) order loss function that can effectively suppress the effect of noisy data and outliers as the learned parameters can better capture the intrinsic information in the dataset. In other words, the proposed RGVQ is better suited to deal with data in the wild.

Secondly, When NR gets increased from 0 to 5%, the similarity search performance of all VQ approaches degrades rapidly. This phenomenon once again demonstrates that VQ is sensitive to noise and outliers in data because of the squared loss, as we have mentioned before. On the contrary, RGVQ approaches show relatively more stable performance in most

cases when we increase NR. More importantly, it can be seen that the performance gap between the corresponding approaches from RGVQ framework and VQ framework becomes even larger when increasing NR. This again demonstrates the superior robustness of the proposed RGVQ against the noise.

Moreover, it is observed that ITQ+ is more robust than OPQ+ and AQ+ since the performance drop of ITQ+ when NR raises from 0 to 5% is less significant. One possible reason is that ITQ+ focuses on binary quantization while OPQ+ and AQ+ adopt real-value quantization. As we will show later, ITQ+ has larger distortion because the binary quantization is
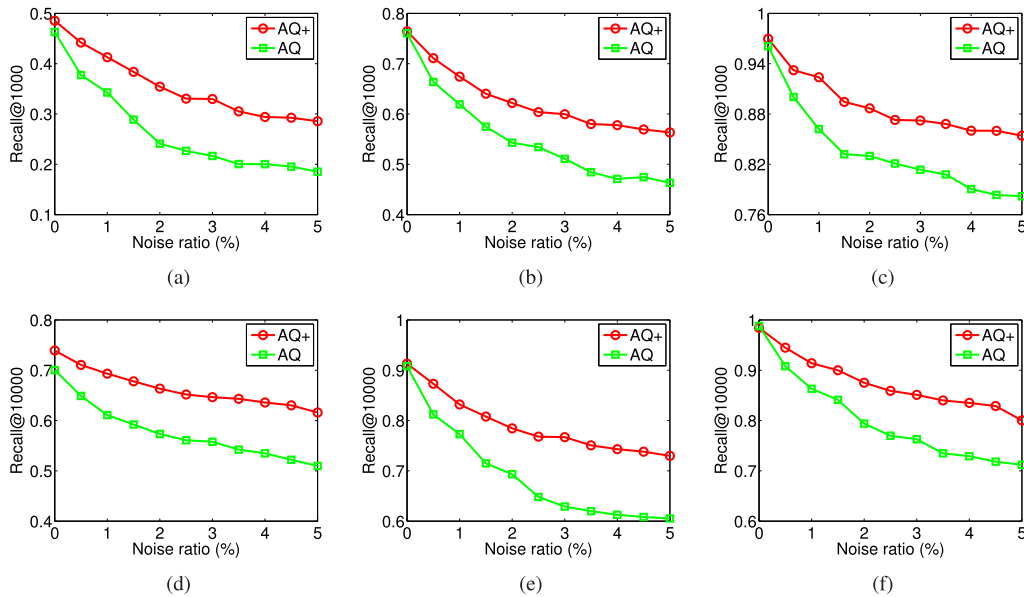
Fig. 5. Performance comparison between AQ+ and AQ w.r.t. the noise ratio. We set $l = 100$ for ground truth. (a) SIFT1M, 16 bits. (b) SIFT1M, 32 bits. (c) SIFT1M, 64 bits. (d) GIST1M, 16 bits. (e) GIST1M, 32 bits. (f) GIST1M, 64 bits.
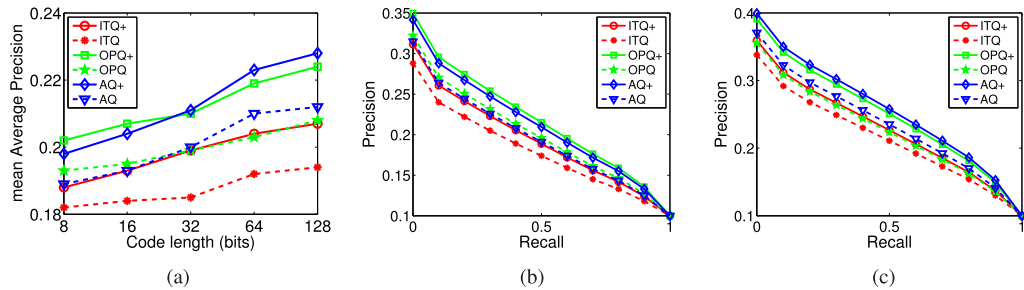

Fig. 6. Performance comparison between RGVQ and VQ on CIFAR-10. (a) Mean Average Precision. (b) Precision-recall curve, 16 bits. (c) Precision-recall curve, 64 bits.
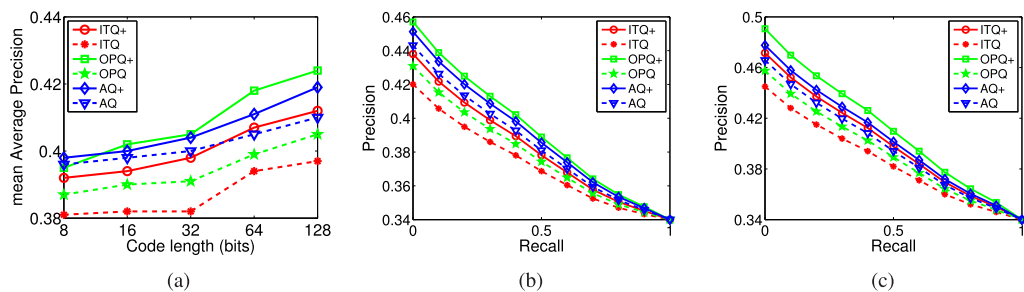

Fig. 7. Performance comparison between RGVQ and VQ on NUS-WIDE. (a) Mean Average Precision. (b) Precision-recall curve, 16 bits. (c) Precision-recall curve, 64 bits.

not that flexible. In this case, the influence of large-distortion entries is relatively smaller in ITQ+ as the majority of entries has large distortion to some extent. Moreover, as OPQ+ and AQ+ have better performance at first, it is more likely that their performance drops more significantly.

D. Image Retrieval Results

From the application perspective, the robustness of RGVQ enables us to search similarity in wild data such as

Internet images. To demonstrate the superiority of RGVQ over VQ, we adopt two widely used image benchmark datasets collected from Web, CIFAR-10 and NUS-WIDE, for the image retrieval task. In particular, in this task, the true positives for each query are defined as the images in the database which share at least one semantic labels/concepts with the query, following [8], [17], [59]. To evaluate the performance, we adopt the Precision-recall curve as the metric, which reflects the precision (the ratio between the number of true
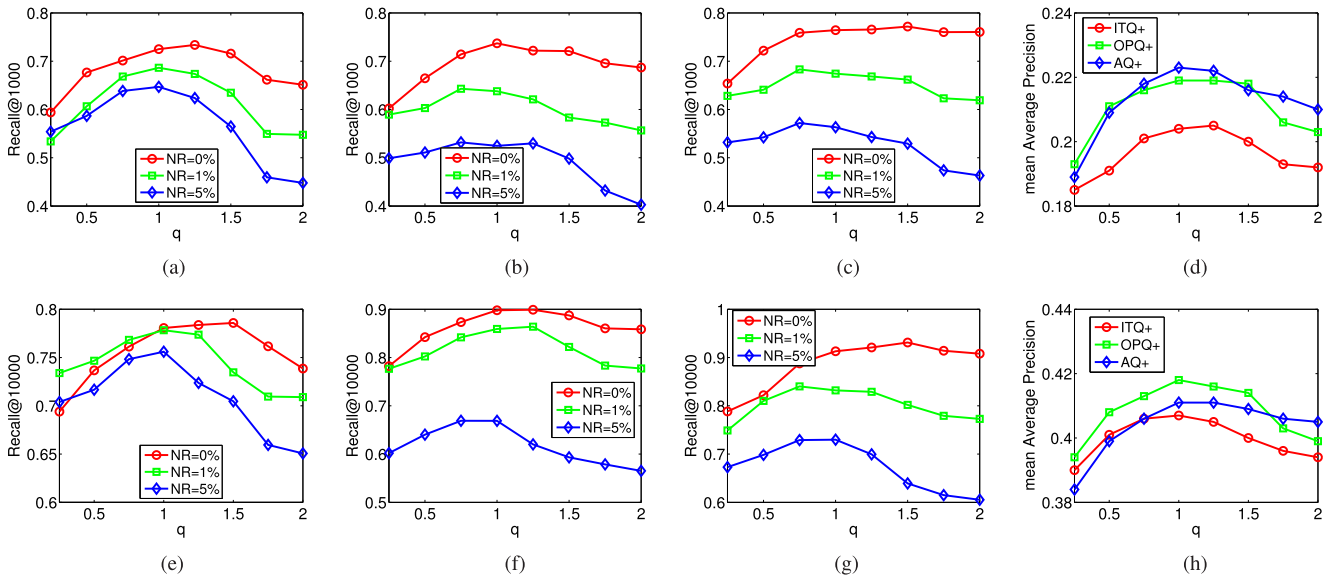
Fig. 8. The effect of q on RGVQ. (a) SIFT1M, 64 bits, ITQ+. (b) SIFT1M, 32 bits, OPQ+. (c) SIFT1M, 32 bits, AQ+. (e) CIFAR-10, 64 bits. (f) GIST1M, 64 bits, ITQ+. (g) GIST1M, 32 bits, OPQ+. (h) GIST1M, 32 bits, AQ+. (i) NUS-WIDE, 64 bits.

positives and that of retrieved images) at different recall levels. Generally, a higher curve indicates that the true positives have higher ranks which is desired for image retrieval task. Moreover, mean Average Precision (mAP) is also utilized as a numeric evaluation metric. It is defined as the area under the Precision-recall curve and a larger value stands for a better performance.

The results of RGVQ approaches and VQ approaches on CIFAR-10 and NUS-WIDE are presented in Fig. 6 and Fig. 7 respectively. It can be seen that RGVQ consistently outperforms VQ with observable margins with different code length on two datasets. In fact, the real-world image sets are always noisy. Unfortunately, existing approaches fail to consider the influence of noise data. As we have analyzed around Eq. (5), when q is large, the learning procedure prefers to decrease the loss of large-distortion entries, while it focuses more on the small-distortion entries when q is small. In the VQ approaches, the squared Euclidean distance is employed to measure the loss to which the noisy samples may contribute significantly since the square operation puts larger weight to the entries with larger distance which are more likely to be noise. Consequently, the models pay too much attention to the noise such that the intrinsic structure of data is not well exploited. On the other hand, by utilizing the q-th order ($q < 2$) of the Euclidean distance, the noisy samples contribute less to the loss function than the squared one. The superior performance of RGVQ again demonstrates that considering the influence of noise, like by setting $q = 1$ in RGVQ framework, is indeed helpful to build effective and efficient ANN search systems in the real-world applications, like image retrieval.

E. Effect of Parameter q

There is one important parameter q in RGVQ which controls the order of the distortion. Here, we investigate how the approaches will behave when varying q. To do so, we change

the value of q and plot the corresponding performance of RGVQ approaches on the benchmark datasets with different binary code length and noise ratios. The results are illustrated in Fig. 8. It is noticed that VQ is a special case of RGVQ when $q = 2$. We have the observations below from the results.

Firstly, in all settings, we can find a Bell-shape curve for all approaches. Basically, the model is affected by both noise and normal data. With a large q (say, $q > 1.5$), RGVQ will increase the weight of those large-distortion entries such that the model will be biased by them. Unfortunately, due to the existence of noisy entries and their large distortions, the learned model will deviate significantly to fit the outliers from the one which best suits to the normal data. Therefore, the performance of all RGVQ approaches degrades significantly when we increase q from 1.5 to 2, especially in more noisy settings, e.g., NR = 5%. On the other hand, if q is too small (say, $q < 0.5$), we cannot obtain good results either. According to the principle, the difference between normal and noisy data becomes smaller in this case, though the effect of outliers is suppressed. In the extreme case where $q = 0$, every entry has the same distortion 1 such that any model is the solution for this case. Thus, it is almost impossible to find the optimal model for normal data. This interprets why RGVQ approaches perform worse when we decrease q from 0.5 to 0.25, especially when there is less noise, e.g., NR = 0. In Fig. 8, we can see that RGVQ approaches perform stably good when $q \in [0.75, 1.25]$ where the effect of outliers on the model is effectively suppressed and that a model which can well fit to the normal data is learned.

Secondly, we can observe that the performance-vs-q curve behaves differently at different noise levels. Specifically, given a small NR, e.g., NR = 0, RGVQ approaches seem more sensitive to q when $q < 1$, because the the performance changes dramatically when varying q in this range. On the other hand, given a large NR, e.g., NR = 5%, they become
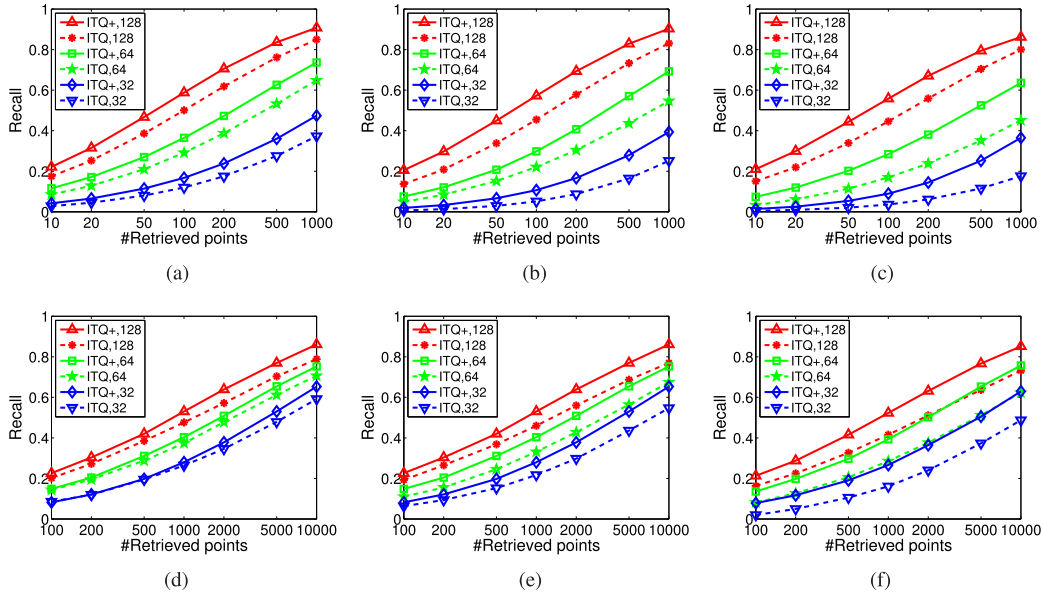
Fig. 9.   Performance comparison between ITQ+ and ITQ for ℓ_p-norm similarity search. (a) SIFT1M, $p = 2$. (b) SIFT1M, $p = 1.5$. (c) SIFT1M, $p = 1$. (d) GIST1M, $p = 2$. (e) GIST1M, $p = 1.5$. (f) GIST1M, $p = 1$.

Fig. 10.   Performance comparison between OPQ+ and OPQ for ℓ_p-norm similarity search. (a) SIFT1M, $p = 2$. (b) SIFT1M, $p = 1.5$. (c) SIFT1M, $p = 1$. (d) GIST1M, $p = 2$. (e) GIST1M, $p = 1.5$. (f) GIST1M, $p = 1$.
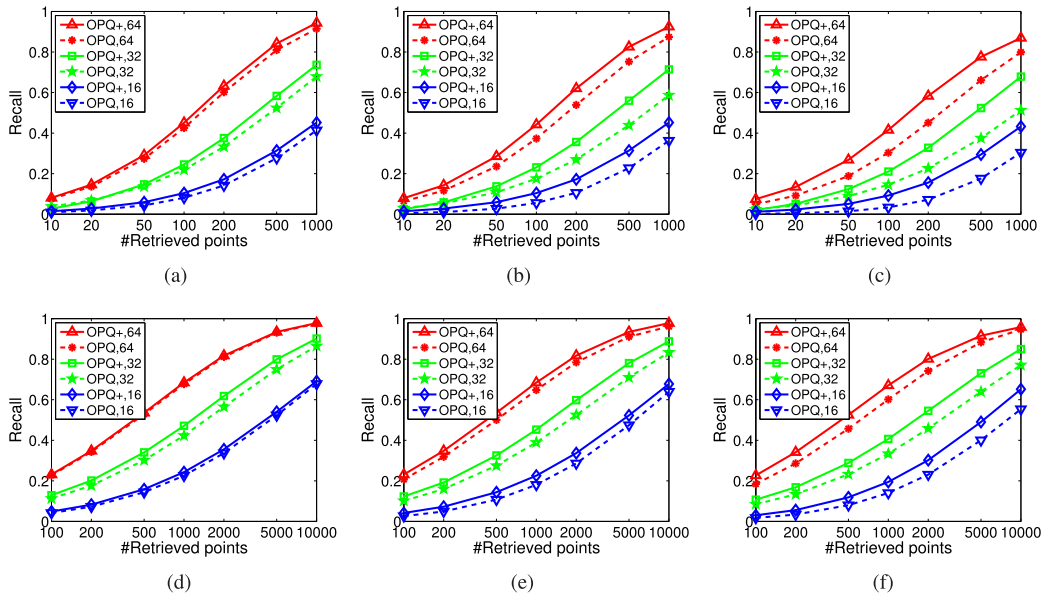
more sensitive when $q > 1$. The reason is analogous to our analysis in the last paragraph. When there is little noise, the primary target of RGVQ is to fit the normal data. In this case, the performance may degrade rapidly if q is too small because the the loss is too indiscriminative. On the other hand, as a result of the increasing noise, the primary target of RGVQ becomes to suppress the influence of noise. Thus, increasing the value of q when $q > 1$ leads to much worse performance.

F. ℓ_p-Norm Similarity Search

As stated as an important property of RGVQ, it can support similarity search with different metrics based on the demand specified by the users while the original VQ only

focuses on the ℓ_2-norm similarity search. In this subsection, we will demonstrate the effectiveness of RGVQ for general ℓ_p-norm similarity search. For RGVQ, we can set the parameter p depending on the specific task and we set $q = 1$ consistently. Specifically, we consider the ℓ_2-norm (Euclidean distance), $\ell_{1.5}$-norm, and ℓ_1-norm (Manhattan distance) similarity search. In the specific task, the ground truth is obtained by running a brute-force linear scan measured by the ℓ_p-norm ($p = 2, 1.5$ and 1) distance in the three tasks respectively.

The recall curves (which reflects the recall level w.r.t. the number of retrieved points) of RGVQ approaches and VQ approaches for three different tasks on two datasets with
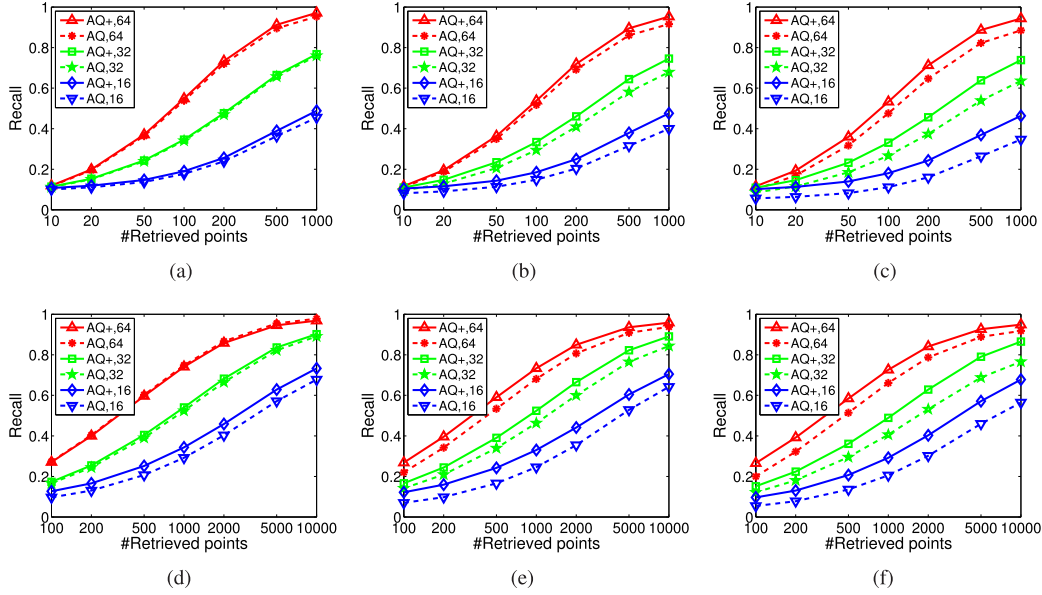
Fig. 11. Performance comparison between AQ+ and AQ for ℓ_p-norm similarity search. (a) SIFT1M, $p = 2$. (b) SIFT1M, $p = 1.5$. (c) SIFT1M, $p = 1$. (d) GIST1M, $p = 2$. (e) GIST1M, $p = 1.5$. (f) GIST1M, $p = 1$.
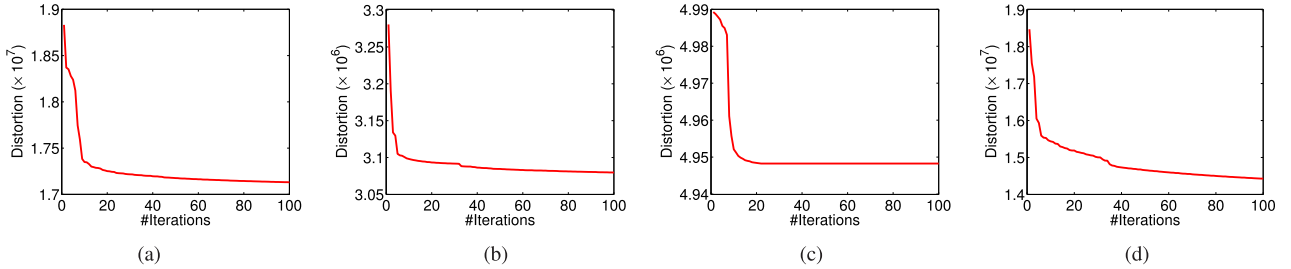

Fig. 12. Convergence study, ITQ+, SIFT1M, 64 bits. (a) $p = 2, q = 1.5$. (b) $p = 2, q = 1$. (c) $p = 1.5, q = 1$. (d) $p = q = 1$.
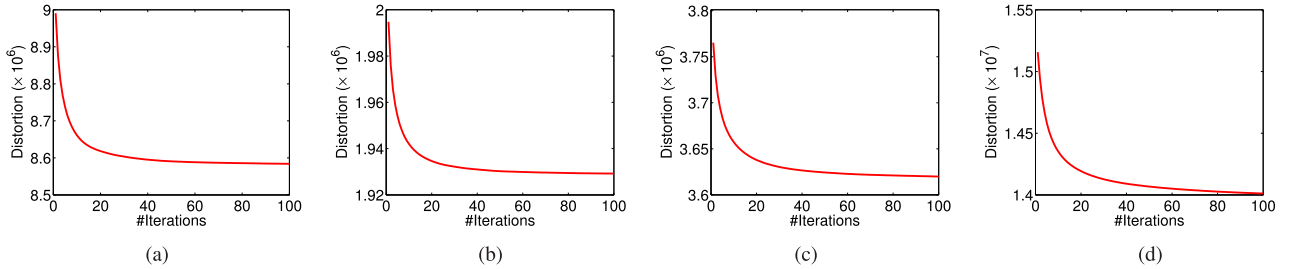

Fig. 13. Convergence study, OPQ+, SIFT1M, 32 bits. (a) $p = 2, q = 1.5$. (b) $p = 2, q = 1$. (c) $p = 1.5, q = 1$. (d) $p = q = 1$.

different code length are summarized in Fig. 9, 10, and 11. Here, we use ℓ_2-norm retrieval performance as the reference as the original VQ approaches are designed for this task. We can observe that RGVQ approaches have relatively more stable performance on different tasks whereas VQ approaches perform much worse on other two tasks than on ℓ_2-norm task. For example, the Recall@1000 of ITQ drops from 0.651 for ℓ_2-norm to 0.474 for ℓ_1-norm on SIFT1M with 64 bits, that of OPQ drops from 0.690 for ℓ_2-norm to 0.527 for ℓ_1-norm, and that of AQ drops from 0.774 for ℓ_2-norm to 0.619 for ℓ_1-norm on SIFT with 32 bits. Consequently, the performance gap between the corresponding RGVQ approaches and VQ approaches becomes much larger when we change p from 2 to 1.5 and 1. In addition, combining with the

results in Fig. 2(c) and 2(d), we can see that the learning algorithms of VQ approaches may unavoidably lead to larger ℓ_p-norm distortion, which is the minimizing objective, with more iterations since it adopts ℓ_2 loss, thus resulting in worse ANN search performance. Fortunately, the RGVQ framework takes the issue into consideration and it is formulated as a more general $\ell_{p,q}$-norm loss function which can be applied to different settings such that it can well support the general ℓ_p-norm similarity search.

G. Convergence Study

As an important theoretical contribution of this paper, we propose an efficient optimization algorithm, Algorithm 1, for optimization the challenging orthogonality constrained
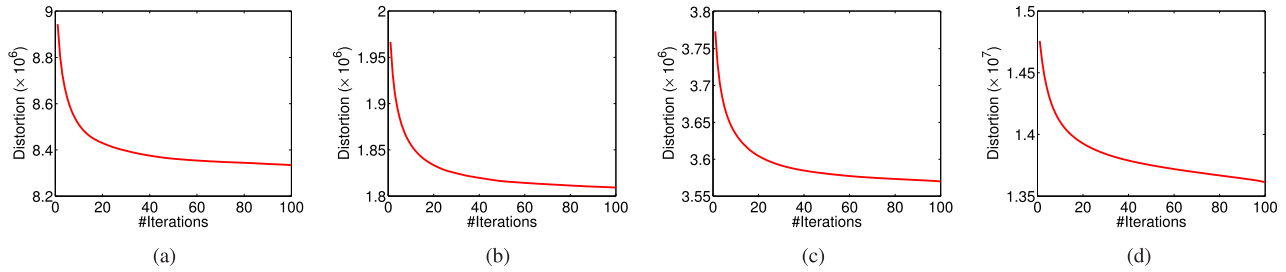
Fig. 14.   Convergence study, AQ+, SIFT1M, 32 bits. (a) $p = 2, q = 1.5$. (b) $p = 2, q = 1$. (c) $p = 1.5, q = 1$. (d) $p = q = 1$.

$\ell_{p,q}$-norm minimization problem. We have rigourously proved that Algorithm 1 leads to nonincreasing objective value. Now, we empirically investigate its convergence property by conducting the experiment on SIFT1M dataset. Because Algorithm 1 is designed for the general $\ell_{p,q}$-norm loss function, we assign different values to p and q and plot the function value of three specific approaches. The objective function value for ITQ+, OPQ+, and AQ+ w.r.t. the number of iterations with different settings are plotted in Fig. 12, 13, and 14 respectively. As can be seen, the objective value decreases steadily with more iterations and can achieve a nearly stable value within less than 100 iterations, which verifies the effectiveness of Algorithm 1.

VI. CONCLUSION

In this paper, we have presented an enhanced VQ framework, termed RGVQ, which changes the $\ell_{2,2}$-norm loss in the original VQ framework to a more general $\ell_{p,q}$-norm loss. The benefits are twofold. On the one hand, the algorithm becomes more robust to the noise, which potentially makes RGVQ better suited to search similarity in the real-world data. On the other hand, promoting to $\ell_{p,q}$-norm loss allows RGVQ to handle various applications, where different distance measurements are requested. The major technical challenge comes from minimizing the new $\ell_{p,q}$-norm loss function, which is a non-smooth and non-convex optimization problem. To solve this orthogonality constrained $\ell_{p,q}$-norm minimization problem, we propose an efficient algorithm and rigorously prove its convergence. We specify the algorithm to three celebrated approaches. Comprehensive experiments on two NN search benchmarks demonstrate that RGVQ performs significantly better than VQ, and validate that RGVQ is robust to noise and works well for ℓ_p-norm similarity search. Moreover, from the application perspective, the extensive results on two image retrieval benchmarks also verify that RGVQ works better than VQ on real-world scenarios as it is more general and robust.

REFERENCES

[1] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statist.*, vol. 46, no. 3, pp. 175–185, 1992.

[2] C. Bishop, *Pattern Recognition and Machine Learning*, vol. 1. New York, NY, USA: Springer, 2006.

[3] G. W. Furnas *et al.*, "Information retrieval using a singular value decomposition model of latent semantic structure," in *Proc. 11th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1988, pp. 465–480.

[4] J. Moraleda, "Gregory Shakhnarovich, Trevor Darrell and Piotr Indyk: Nearest-neighbors methods in learning and vision. theory and practice," *Pattern Anal. Appl.*, vol. 11, no. 2, pp. 221–222, May 2008.

[5] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975.

[6] E. Chávez, G. Navarro, R. A. Baeza-Yates, and J. L. Marroquín, "Searching in metric spaces," *ACM Comput. Surv.*, vol. 33, no. 3, pp. 273–321, Sep. 2001.

[7] Y. Matsui, T. Yamasaki, and K. Aizawa, "PQTable: Fast exact asymmetric distance neighbor search for product quantization using hash tables," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1940–1948.

[8] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

[9] A. Gordo, F. Perronnin, Y. Gong, and S. Lazebnik, "Asymmetric distances for binary embeddings," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 33–47, Jan. 2014.

[10] T. Ge, K. He, Q. Ke, and J. Sun, "Optimized product quantization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 744–755, Apr. 2014.

[11] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.

[12] M. Norouzi and D. J. Fleet, "Cartesian K-means," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3017–3024.

[13] Y. Chen, T. Guan, and C. Wang, "Approximate nearest neighbor search by residual vector quantization," *Sensors*, vol. 12, no. 12, pp. 11259–11273, 2010.

[14] T. Zhang, C. Du, and J. Wang, "Composite quantization for approximate nearest neighbor search," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. II-838–II-846 .

[15] A. Babenko and V. Lempitsky, "Additive quantization for extreme vector compression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 931–938.

[16] J. Wang, J. Wang, J. Song, X.-S. Xu, H. T. Shen, and S. Li, "Optimized Cartesian K-means," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 1, pp. 180–192, Jan. 2015.

[17] X. Lu, X. Zheng, and X. Li, "Latent semantic minimal hashing for image retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 355–368, Jan. 2017.

[18] Y. Guo, G. Ding, L. Liu, J. Han, and L. Shao, "Learning to hash with optimized anchor embedding for scalable retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1344–1354, Mar. 2017.

[19] G. Ding, Y. Guo, J. Zhou, and Y. Gao, "Large-scale cross-modality search via collective matrix factorization hashing," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5427–5440, Nov. 2016.

[20] X. Lu, Y. Chen, and X. Li, "Hierarchical recurrent neural hashing for image retrieval with hierarchical convolutional features," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 106–120, Jan. 2018.

[21] Y. Gong, L. Liu, M. Yang, and L. D. Bourdev, "Compressing deep convolutional networks using vector quantization," *CoRR*, vol. abs/1412.6115, 2014. [Online]. Available: http://arxiv.org/abs/1412.6115

[22] J. Huang, F. Nie, H. Huang, and C. Ding, "Robust manifold nonnegative matrix factorization," *ACM Trans. Knowl. Discovery Data*, vol. 8, no. 3, p. 11, Jun. 2013.

[23] Q. Pan, D. Kong, C. Ding, and B. Luo, "Robust non-negative dictionary learning," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2027–2033.

[24] W. Jiang, F. Nie, and H. Huang, "Robust dictionary learning with capped ℓ_1-norm," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3590–3596.

[25] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. 23rd IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3304–3311.

[26] U. P. Singh and S. Jain, "Content based image retrieval using Euclidean and Manhattan metrics," *J. Math. Sci., Adv. Appl.*, vol. 4, no. 1, pp. 217–226, 2010. [Online]. Available: https://pdfs.semanticscholar.org/77f0/34cbdc74d84d43ec292b402febfbf101a3a0.pdf

[27] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. 23rd Annu. Conf. Neural Inf. Process. Syst.*, 2009, pp. 2080–2088.

[28] H. Wang, F. Nie, W. Cai, and H. Huang, "Semi-supervised robust dictionary learning via efficient $\ell_{2,0+}$-norms minimization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1145–1152.

[29] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 625–632.

[30] H. Wang, F. Nie, and H. Huang, "Learning robust locality preserving projection via p-order minimization," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 3059–3065.

[31] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 801–808.

[32] Y. Guo, G. Ding, J. Zhou, and Q. Liu, "Robust and discriminative concept factorization for image representation," in *Proc. 5th ACM Int. Conf. Multimedia Retr.*, 2015, pp. 115–122.

[33] C. Ding, D. Zhou, X. He, and H. Zha, "R_1-PCA: Rotational invariant L_1-norm principal component analysis for robust subspace factorization," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 281–288.

[34] Y. Guo, G. Ding, J. Han, and X. Jin, "Robust iterative quantization for efficient ℓ_p-norm similarity search," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 3382–3388.

[35] Z. Wen and W. Yin, "A feasible method for optimization with orthogonality constraints," *Math. Program.*, vol. 142, no. 1, pp. 397–434, 2013.

[36] L. A. Vese and S. J. Osher, "Numerical methods for p-harmonic flows and applications to image processing," *SIAM J. Numer. Anal*, vol. 40, no. 6, pp. 2085–2104, 2002.

[37] D. Goldfarb, Z. Wen, and W. Yin, "A curvilinear search method for *p*-harmonic flows on spheres," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 84–109, 2009.

[38] R. C. de Amorim and B. G. Mirkin, "Minkowski metric, feature weighting and anomalous cluster initializing in K-means clustering," *Pattern Recognit.*, vol. 45, no. 3, pp. 1061–1075, Mar. 2012.

[39] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4680–4688, Jul. 2011.

[40] M. Zheng *et al.*, "Graph regularized sparse coding for image representation," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1327–1336, May 2011.

[41] L. Liu, Z. Lin, L. Shao, F. Shen, G. Ding, and J. Han, "Sequential discrete hashing for scalable cross-modality similarity retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 107–118, Jan. 2017.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 26th Annu. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[44] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 499–515.

[45] X. Qian, D. Lu, Y. Wang, L. Zhu, Y. Y. Tang, and M. Wang, "Image re-ranking based on topic diversity," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3734–3747, Aug. 2017.

[46] X. Qian *et al.*, "Image location inference by multisaliency enhancement," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 813–821, Apr. 2017.

[47] S. Zhao, H. Yao, Y. Gao, R. Ji, and G. Ding, "Continuous probability distribution prediction of image emotions via multitask shared sparse regression," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 632–645, Mar. 2017.

[48] Y. Guo, G. Ding, J. Han, and Y. Gao, "Zero-shot learning with transferred samples," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3277–3290, Jul. 2017.

[49] D. Zhang, J. Han, L. Jiang, S. Ye, and X. Chang, "Revealing event saliency in unconstrained video collection," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1746–1758, Apr. 2017.

[50] L. Li, X. Jin, S. J. Pan, and J.-T. Sun, "Multi-domain active learning for text classification," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1086–1094.

[51] M. Chen, X. Jin, and D. Shen, "Short text classification improved by learning multi-granularity topics," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1776–1781.

[52] X. Yao, J. Han, G. Cheng, X. Qian, and L. Guo, "Semantic annotation of high-resolution satellite images via weakly supervised learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3660–3671, Jun. 2016.

[53] X. Lu, X. Zheng, and Y. Yuan, "Remote sensing scene classification by unsupervised representation learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5148–5157, 2017.

[54] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[55] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[56] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

[57] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, Nov. 2008.

[58] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world web image database from National University of Singapore," in *Proc. ACM CIVR*, Jul. 2009.

[59] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.

Yuchen Guo received the B.Sc. degree from the School of Software and the B.Ec. degree from the School of Economics and Management, Tsinghua University, Beijing, China, in 2013, where he is currently pursuing the Ph.D. degree with the School of Software. His research interests include multimedia information retrieval, computer vision, and machine learning.

Guiguang Ding received the Ph.D. degree in electronic engineering from Xidian University, China, in 2014. He was a Post-Doctoral Research Fellow with the Department of Automation, Tsinghua University. He is currently an Associate Professor with the School of Software, Tsinghua University. He has authored 80 papers in major journals and conferences, including the IEEE TIP, TMM, TKDE, SIG IR, AAAI, ICML, IJCAI, CVPR, and ICCV. His current research centers on the area of multimedia information retrieval, computer vision, and machine learning.

Jungong Han was a Faculty Member with the Department of Computer and Information Sciences, Northumbria University, U.K. He is currently a tenured Faculty Member with the School of Computing and Communications, Lancaster University, Lancaster, U.K. His research interests include computer vision, image processing, machine learning, and artificial intelligence.