Computer Vision and Image Understanding 124 (2014) 22–30

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Statistical quantization for similarity search

Qi Wang [a], Guokang Zhu [b], Yuan Yuan [b,*]

[a] Northwestern Polytechnical University, Xi'an 710072, Shaanxi, PR China
[b] Center for OPTical IMagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, PR China

ARTICLE INFO

Article history:
Received 14 August 2013
Accepted 2 March 2014

Keywords:
Machine learning
Computer vision
Similarity search
Hashing
Quantization
Binary code

ABSTRACT

Approximate nearest neighbor search has attracted much attention recently, which allows for fast query with a predictable sacrifice in search quality. Among the related works, *k*-means quantizers are possibly the most adaptive methods, and have shown the superiority on search accuracy than the others. However, a common problem shared by the traditional quantizers is that during the out-of-sample extension process, the naive strategy considers only the similarities in Euclidean space without taking into account the statistical and geometrical properties of the data. To cope with this problem, in this paper a novel approach is proposed by formulating a generalized likelihood ratio analysis. In particular, the proposed method takes a physically meaningful discrimination on the affiliations of the new samples with respect to the obtained Voronoi cells. This discrimination essentially imposes the measure of statistical consistency on out-of-sample extension. The experimental studies on two large data sets show that the proposed method is more effective than the benchmark algorithms.

1. Introduction

With the explosive data growth on the web, fast similarity indexing and search are considered to be one of the most fundamental problems for the multimedia communities [1–3]. This problem is also known as *Nearest Neighbor* (NN) search, which is defined as accurately finding the close samples for a given query within a large database [4,5]. It is of great importance to a wide range of multimedia applications, such as content-based image/video retrieval [6–8], image/video auto-tagging [9–11], image classification [12,13], and scene recognition [14]. Naively searching for the neighbors according to their similarities entails exhaustively comparing the queries with the examples over the entire database. This strategy has linear complexity with respect to the scale of the database, which is infeasible on ever larger databases. Besides, to achieve satisfying performance on such databases, most of the related applications have to rely on the high-dimensional or structured representations, as well as the computationally considerable distance functions [15,16]. Therefore, the naive strategy is prohibitively expensive in practical situations.

To make similarity indexing and search scalable, several *Approximate Nearest Neighbor* (ANN) techniques have been developed, through which a fast query is allowed with the predictable sacrifice in accuracy [17,18]. Instead of performing a completely NN search on a database $\mathcal{X}$ through linear scanning with $O(|\mathcal{X}|)$ query time, ANN techniques are desired to achieve the fast yet accurate indexing with sublinear $o(|\mathcal{X}|)$ [19,20], logarithmic $O(\log|\mathcal{X}|)$ [21,22], or even with constant $O(1)$ complexity. Among these techniques, *hashing*-based ones have attracted more attention and announced laudable performances recently. This type of ANN techniques is preferable for its constant query time and the substantially reduced memory [23]. In this work, we focus on the later aspect that relies on generating compact binary codes for the high dimensional samples in a large database while maintaining the structure of the original database. By constraining the similar data points with close binary codes, similarity search is then accelerated by finding the neighbors within a small Hamming distance from the query.

Broadly, researches toward *hashing* can be divided into two main categories: Hamming-based and lookup-based. Both these two categories involve a quantization process, through which the original feature space is partitioned into some unique cells, and a corresponding strategy for distance computation. For the Hamming-based methods, the quantization is achieved by using hyperplanes [24,25] or kernel hyperplanes [26,27]. These hyperplanes are generally determined by the signs of the employed hashing functions. Each hyperplane is then used to encode a unique bit of the desired compact code. In regard of hashing function, several strategies have been developed to cope with the practical

[*] Corresponding author.
E-mail address: yuany@opt.ac.cn (Y. Yuan).

http://dx.doi.org/10.1016/j.cviu.2014.03.002
1077-3142/© 2014 Elsevier Inc. All rights reserved.

scenarios. For instance, in [28] the post-combination strategy is employed on the linear hash functions of different types of features for content-based image retrieval. In [8] a pre-concatenation strategy is proposed for contend-based video retrieval, which equally concatenates all the employed features as one and then constructs the hashing function. Differently, in [29,30], multiple features are non-linearly concatenated and then projected using linearly combined multiple kernel hyperplanes. As for the lookup-based methods [31–33], they usually partition the feature space through k-means clustering. Such a quantization is considered to be more adaptive than those on the basis of hyperplane construction, and is likely to be more accurate with the same code-length [34].

Though the lookup-based methods have shown success in many large-scale searching scenarios, there is a problem seldom exploited. Given b bits for quantization, k-means quantizers are implemented by mapping the original descriptions to the codebook containing at most 2^b codewords [31–33]. Specifically, the classical k-means quantizers use the cluster centers as codewords, and then assign to any data point a nearest codeword according to the distance measure in Euclidean space. However, in spite of minimal quantization error during off-line training [15,32], assigning a new sample the cluster index through such strategy lacks statistical interpretability, and may fail in many practical cases. For example, as shown in Fig. 1, the new sample x will be assigned the cluster index c_2 if only the Euclidean distances to the cluster centers are consulted, but it may have the property more similar to those contained in the cluster c_1.

In order to cope with the above problem, in this paper a novel quantizer is proposed for effective similarity search. This method improves the classical k-means quantization by taking into account the statistical consistency of a new sample with respect to each partitioned cell. Our purpose is achieved by formulating a *Generalized Likelihood Ratio* (GLR) analysis, through which each sample can be identified as an inlier or not in the examined cell. We claim that the proposed method is physically meaningful and practically preferable in the out-of-sample extension process.

The rest of this paper is organized as follows: Section 2 gives a brief review on the background of quantization methods. Section 3 presents the detailed formulation of the proposed method. Then, the experimental comparison is conducted and analyzed in Section 4 to verify the effectiveness of the proposed method. Finally, the conclusion is given in Section 5 to summarize this paper.

2. Background: quantization via *k*-means

Quantization has been the topic of prolonged and extensive study, and has a large body of literatures in information theory [31,35]. Its purpose is to provide a low-cardinal representation space to a database, which can facilitate further processing especially when the tasks suffer from the *curse of dimensionality*. This section starts with the presentation for the classical concept of quantizing the feature space in a k-means fashion. Then, a brief review on the generalization of this quantization strategy to product space is presented, which improves the practicability of the quantizer when the bit number is large.

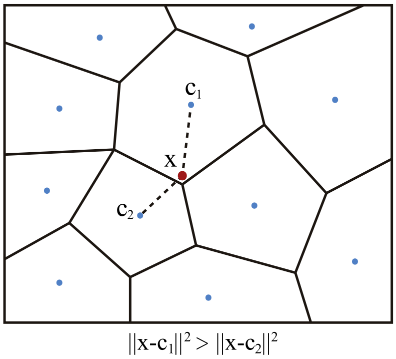


Fig. 1. Failure case of assigning a new sample x to a cluster index according to the similarity in Euclidean space.

2.1. Vector quantization

For the classical vector quantization, the quantizer is a function $q(\cdot)$ mapping a m-dimensional vector $x \in \mathbb{R}^m$ to another vector [31], such that

$$q(x) \in \mathcal{C} = \{c_i | c_i \in \mathbb{R}^m, i \in \mathcal{I}\}, \tag{1}$$

where the set $\mathcal{C}$ is the *codebook* of size k, c_i is the *codeword* usually given by the k-means centers [15,32], and $\mathcal{I} = \{0, 1, \ldots, k-1\}$. Each set of vectors mapped to the same codeword c_i is referred to as a unique Voronoi *cell* $\mathcal{V}_i$, which is defined as

$$\mathcal{V}_i = \{x | x \in \mathbb{R}^m, q(x) = c_i\}. \tag{2}$$

Then, the k cells together characterize the partition that the quantizier induces on the input space $\mathbb{R}^m$. The relationship of these concepts is illustrated in Fig. 2.

By definition, each input vector will be represented by the assigned codeword. The quality of a given quantizer is usually measured in term of the averaged distortion between the original vector x and the mapping $q(x)$ [15,31],

$$D(q) = E_x[d(x, q(x))], \tag{3}$$

where the distortion measure $d(x, y)$ can take various specific forms, and is typically the Euclidean distance between x and y [15,32]. Then, applying the triangle inequality for (3) leads to

$$E_x[|d(x, x') - d(x, q(x'))|] \leqslant D(q). \tag{4}$$

This indicates an upper bound on the expected error for estimating the inter-sample distances, when one sample in a pair is approximated by its quantization result. Therefore, a quantizer that minimizes $D(q)$ for a given codebook of size k can claim its effectiveness for NN search within the database.

In order for a quantizer to be optimal subject to the underlying probability distribution, it has to satisfy the following two properties:

- $q(x) = \{c_i | d(x, c_i) \leqslant d(x, c_j), \forall j \in \mathcal{I}\}$;
- $c_i = \arg\min_{x'} E_x[d(x, x') | x, x' \in \mathcal{V}_i]$.

The first property regularizes that the quantization cells consist of samples no further from its centroid than from any other

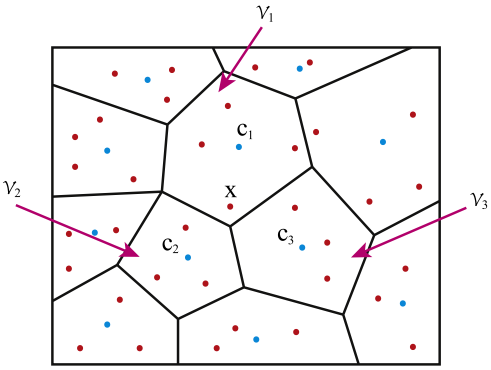


Fig. 2. Illustration of the related concepts. The red and blue dots represent the samples x and the codewords c, respectively. A set of samples in a polygon constitutes a Voronoi cell $\mathcal{V}_i$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

codewords. The second condition is that the codeword for a given cell must be the expectation of the data points lying in the cell. These two properties are also known as the Lloyd optimality conditions [36], which contribute to the theoretical basis of k-means.

2.2. Generalization to product space

Given b bits for vector quantization, the classical k-means quantizers have to contain 2^b codewords. For a large b (e.g., $b = 64$ such that $k = 2^{64}$), it is not practical to directly use the Lloyd optimality conditions due to the expensive computational cost. Actually, it is even impossible to load the $k \times m$ floating-point values into memory. Towards this issue, the product quantization method is an efficient solution, of which the problem is reduced to be the optimization in a set of independent subspaces [15,32].

In the case that the probability distribution of x can be considered to be independent in its components, the product quantization method forms the Cartesian product of $\mathbb{R}^m$ yielding L subspaces. That is, splitting each vector $x \in \mathbb{R}^m$ into L distinct subvectors: $x = [\hat{x}^1, \ldots, \hat{x}^l, \ldots, \hat{x}^L]$, where $\hat{x}^l$ denotes the lth subvector of dimension $\hat{m} = m/L$. For each subspace, a low-complexity sub-quantizer is obtained independently by minimizing the expected distortion for the corresponding subvectors [15,32]. To be specific, the lth quantizer q^l associated with the lth subspace is optimized by minimizing

$$D(q^l) = E_{\hat{x}^l}[d(\hat{x}^l, q^l(\hat{x}^l))]. \tag{5}$$

A sub-quantizer q^l contains a sub-codebook $\hat{\mathcal{C}}^l$ with $\hat{k}$ sub-codewords. Any codeword c of the product quantizer is constructed as a concatenation of L sub-codewords drawn from L sub-codebooks. Correspondingly, the final codebook is also defined as the Cartesian product

$$\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_l \cdots \times \mathcal{C}_L. \tag{6}$$

In such a way, the total number of distinct codewords in $\mathbb{R}^m$reaches $k = (\hat{k})^L$. This property makes a product quantizer to be powerful to produce a large set of codewords from several subsets. But the algorithm only needs to calculate and store $\hat{k} \times L \times \hat{m} = \hat{k} \times m$ floating-point values, instead of the original amount $k \times m$. Thus, the product quantizer turns out to be efficient since $\hat{k}$ can be much less than k.

3. Proposed model

In the classical k-means quantizers, the binary codes assigned to the training samples are determined according to their distances from the codewords in Euclidean space. Then, the same criterion is also applied to tackle the out-of-sample extension problem [15,31–33]. This strategy implies an assumption that including the new samples will not introduce any distortion on the statistical and geometrical properties of the original database. However, this assumption does not hold for many practical cases, particularly the ones where the statistics vary greatly among different clusters.

Toward this problem, in this paper the out-of-sample extension process is formulated by a physically meaningful GLR analysis, through which the statistical consistency of a sample with respect to each partitioned cell is analyzed to determine its affiliation.

3.1. Formulation of the problem

Considering an undesirable signal $\theta \in \mathbb{R}^m$, which will lead to deviations from the distribution of an examined cell $\mathcal{V}$, the two hypotheses to be distinguished for a new sample are given by

$$\begin{cases} H_0 : x = z, \\ H_1 : x = z + s\theta. \end{cases} \tag{7}$$

The hypothesis H_1 represents the presence of an undesirable signal with strength s, and H_0 indicates the opposite case. $z \in \mathbb{R}^m$ is a vector representation of residual “background” or “clutter” signal fitting the desired distribution. Thus, for a data set X including the examined cell $\mathcal{V}$ and the new sample $x, X = \{\mathcal{V} \cup x\}$, the two hypothesis are formulated as

$$\begin{cases} H_0 : X = Z, \\ H_1 : X = Z + S^T\theta, \end{cases} \tag{8}$$

where $S = [s_1, s_2, \ldots, s_N]$ is the strength vector, which also reflects the spatial position of θ across the data set of size N.

3.2. Definition of the likelihood function

Without loss of generality, we impose a mean removal process on Z. Then the residual clutter could be assumed to be multivariate Gaussian and independent from sample to sample according to [37]. As a consequence, X should be also Gaussian distributed. Then we have

$$X \sim \begin{cases} \mathcal{N}(\mathbf{0}, I_N\Sigma), & H_0, \\ \mathcal{N}(S^T\theta, I_N\Sigma), & H_1. \end{cases} \tag{9}$$

where

$$\Sigma = E\{[x - E(x)]^T[x - E(x)]\}, \tag{10}$$

is the covariance of x, and I_N denotes the $N \times N$ identity matrix.

By definition [38], the likelihood function for the H_1 case depending on θ and Σ is in the form of

$$\begin{aligned} \mathcal{L}(\theta, \Sigma) &= [(2\pi)^m|\Sigma|]^{-N/2} \\ &\quad \cdot \exp\left\{-\frac{1}{2}\sum_i[(x - E(x))\Sigma^{-1}(x - E(x))^T]\right\} \\ &= [(2\pi)^m|\Sigma|]^{-N/2} \\ &\quad \cdot \exp\left\{-\frac{1}{2}\mathrm{tr}[(X - S^T\theta)\Sigma^{-1}(X - S^T\theta)^T]\right\}. \end{aligned} \tag{11}$$

Also, a similar expression gives the function $\mathcal{L}(\mathbf{0}, \Sigma)$ for the H_0 hypothesis.

3.3. Determination by GLR

Considering the statistical consistency between the new sample and each partitioned cell, the proposed method aims at constructing a physically meaningful and practically preferable criteria for the out-of-sample extension process. After introducing the formulation of the problem and the definition of the likelihood functions, we now details the GLR analysis to determine the acceptability of a new sample as an inlier in a data set. Given the likelihood functions for the alternative hypotheses, the GLR is calculated by

$$\Lambda(x) = \frac{\max_{\theta,\Sigma}\mathcal{L}(\theta, \Sigma)}{\max_{\Sigma}\mathcal{L}(\mathbf{0}, \Sigma)}. \tag{12}$$

For this definition, a larger value of $\Lambda(x)$ indicates the high possibility of the new sample x being inconsistent with respect to the examined cell.

According to the well-known *Maximum Likelihood Estimator* (MLE) [38], the two items in (12) for the unknown parameters θ and $\Sigma$are given by

$$\max_{\theta,\Sigma}\mathcal{L}(\theta, \Sigma) = [(2\pi)^m|\Sigma_\theta|]^{-N/2}\exp(-mN/2), \tag{13}$$

$$\max_{\Sigma}\mathcal{L}(\mathbf{0}, \Sigma) = [(2\pi)^m|\Sigma_{\mathbf{0}}|]^{-N/2}\exp(-mN/2), \tag{14}$$

where the covariance matrixes under the difference hypotheses H_0 and H_1 are respectively calculated as

$$\Sigma_{\mathbf{0}} = \frac{1}{N} X^T X, \tag{15}$$

and

$$\Sigma_\theta = \frac{1}{N} (X - S^T \hat{\theta})^T (X - S^T \hat{\theta}), \tag{16}$$

with the approximation of the undesirable signal $\hat{\theta}$ given by

$$\hat{\theta} = \frac{S^T X}{S^T S}. \tag{17}$$

Thus, after applying the maximum likelihood estimation results on the definition of GLR, (12) can be further translated to

$$\begin{aligned} \Lambda(x) &= \frac{|\Sigma_{\mathbf{0}}|^{m/2}}{|\Sigma_\theta|^{m/2}} \\ &\propto \frac{|X^T X|}{|X^T X - (S^T X)^T (S^T S)^{-1} (S^T X)|} \\ &\propto \frac{(S^T X)(X^T X)^{-1} (S^T X)^T}{S^T S}. \end{aligned} \tag{18}$$

It should be noted that the vector S essentially reflects the spatial position of the new sample in the constructed data set, i.e., $S = [0, \ldots, 0, 1, 0, \ldots, 0]$ in practice. Besides, the mean of the data set has to be removed to match the zero-mean assumption on the hypothesis H_0. These yield

$$\Lambda(x) = (x - \mu_X) \Sigma_X^{-1} (x - \mu_X)^T, \tag{19}$$

where μ_X and Σ_X are the sample mean and covariance of X, respectively.

Once the statistical consistencies of the new samples with respect to the partitioned cells have been obtained, the out-of-sample extension process can be accomplished by assigning each new sample to the cell associated with the corresponding minimum GLR.

4. Experiments

This section will verify the performance of the proposed method on similarity search. We evaluate our method (termed SQ) and the most related method *Product Quantization* (PQ) [32] on several benchmark data sets. *SQ* and *PQ* share the procedure of codebook construction. But the out-of-sample extension process of *PQ* still relies on the similarity measure in Euclidean space, while *SQ* employs the proposed GLR analysis. Their performance is further compared with 4 state-of-the-art methods: *Compressed Hashing* (CH) [1], *Unsupervised Sequential Projection Learned Hashing* (USPLH) [2], *Spectral Hashing* (SH) [27], and *Locality Sensitive Hashing* (LSH) [23]. The following paragraphs start with the brief description of the employed data sets, and then describe the experimental protocol concerning the evaluation metrics. Finally, the experimental results and the comparisons are presented in detail.

4.1. Data set

The experiments are performed on two publicly available data sets: *SIFT-1M*[1] [32], and *MNIST*[2] [39]. Both these two data sets are originally constituted by a database subset, and a query subset. Each query term in the query subset is associated with its ground truth indices of the nearest neighbors contained in the database subset. Ground truths are defined as Euclidean neighbors for *SIFT-1M*, and by means of the labeled category information for *MNIST*. Additionally, in order to establish the codebooks for *SQ* and *PQ*, as well as to learn the hashing functions for the other benchmark methods, we randomly sample from each database subset with extra 10,000 vectors to constitute the learning subset. Table 1 summaries the properties of the finally employed data sets.

- *SIFT-1M*. This data set contains 110,000 local SIFT descriptors [40] extracted from a large set of images [41]. Each descriptor in the data set is a 128-dimensional vector representing histograms of gradient orientations within a local image structure. In the existing literatures [1,2], there are commonly 1 million samples constituting the database subset, and the remaining 10,000 samples serving as the independent queries. The ground truth nearest neighbors taken into account for a query descriptor are the ones lying in the top 2% positions at the rank of Euclidean similarity.
- *MNIST*. The *MNIST* handwritten digit data set consists of 70,000 images representing the handwritten digit characters from ‘0’ to ‘9’. The overall data set contains examples from approximately 500 different writers. The digits in this data set have been size-normalized and centered in the images of resolution 28 × 28. Following [2], The whole data set is split into a database subset with 69,000 descriptors and a query subset composed of 1000 samples. All the samples in this data set are represented by the raw image information, and the binary encoding is performed in the feature space of dimension 784.

4.2. Experimental protocol

To perform quantitative evaluation, this paper follows the criterion of Hamming ranking commonly adopted in the literatures [1,2,15]. Through Hamming ranking, all the samples in the database subset will be sorted according to their similarities to the query in Hamming distance space. Then, the samples within the top of the ranked list will be retrieved as the desired neighbors. Hamming ranking strategy is an exhaustive search method with linear complexity, but is proved that it can be improved to a very fast speed [42].

Based on the search results, two metrics—*precision* and *recall*, are employed to measure the quantitative performance of the conducted methods. *Precision* is calculated as the proportion of the positive retrieved nearest neighbors among the overall returned samples. *Recall* is defined as the percentage of returned true nearest neighbors in relation to the total number of ground truth nearest neighbors.

4.3. Performance

To validate the effectiveness of the proposed quantizer, the experiments supporting the comparison with the benchmark methods are conducted on the employed data sets. This section starts with the comparison of similarity search quality of different methods. This comparison aims at providing definitive clues concerning the effectiveness. Then, the efficiencies of all the implemented methods are also reported, for which we are interested in

Table 1
Summary of the employed data sets.

	SIFT-1M	MNIST
Dimensionality	128	784
Learning subset size	10,000	10,000
Database subset size	100,000	69,000
Query subset size	10,000	1000
Nearest neighbor size	2000	1380

[1] http://www.irisa.fr/texmex/people/jegou/ann.php.

[2] http://yann.lecun.com/exdb/mnist/.

their performances during the online out-of-sample extension process.

The performance of the proposed quantizer (*SQ*) is compared with 5 competitors. Among these competitors, *PQ* is the most typical case relying on *k*-means clustering and is closest to our method. The other 4 methods are prevalent in the literatures (*SH*, *LSH*) or developed very recently (*CH*, *USPLH*). *SQ* and *PQ* are mainly parameterized by the number L of the subspaces. In this comparison, L is given by $m/4$ yielding $\hat{k} = 2^4$ unique subcodewords of 4 dimensions per subspace, where m is the dimensionality of the original feature space of the data set. For*CH* and *LSH*, the hashing functions are established using the single hash table, and the threshold utilized to find the points to be indexed in the next hash table is fixed at 0.002. As for *USPLH*, the sequential learning rate is 0.125. All these methods are performed to produce the relatively compact codes of 64-bits.

4.3.1. Comparison of search quality

Fig. 3 shows the comparison of the search quality on *SIFT-1 M*, where Fig. 3(a) and (b) respectively give the *recall* versus the number of returned top neighbors and the *precision-recall* curves. From these figures, we can see that *LSH* also performs high-quality search, when the feature space is not compressed with an extremely short code length. Besides, it is surprising to see that *CH*, which is developed in the most recent literature, achieves the worst performance. An additional examination conducted beyond this paper indicates that, *CH* may achieve a better performance when it uses multiple hash tables. However, using multiple hash tables is to some extent equivalent to encoding descriptors with the bits multiple times of the desired length. This strategy will lead to a scenario unfair to the other methods. As the baseline quantizer, *PQ* provides the relatively better but not the best performance among 5 benchmark methods. Differently, *SQ* performs much better since the GLR analysis is introduced in the out-of-sample extension process. In Fig. 3(a), *SQ* achieves the *recall* values dominating most of the competitors and slightly better than *LSH*. Besides, the superiority of *SQ* is more obvious in Fig. 3(b) because it dominates on the *precision-recall* curves. This indicates that the drop in *precision* of *SQ* for larger numbers of returned top neighbors is much less compared with the others.

For the *MNIST* data set, the comparisons of the search qualities are similarly illustrated in Fig. 4. In this case, *CH* with single hash table still entails the worst performance on both *recall* and *precision-recall* curves, and *USPLH* deteriorates to be equivalent to *CH*. As for *LSH*, its excellence does not hold on the *precision-recall* curve in Fig. 4(b), indicating the relative deterioration of the search accuracy compared with the case of *SIFT-1M*. In the meantime, the *k*-means based quantizers perform even better. As shown in Fig. 4(a) and (b), *PQ* lies on top of the *recall* curves among the 5 benchmark methods, and significantly outperforms the others on the *precision-recall* indicator. Further, the proposed quantizer can still improve it to a better performance. Even in the case that *PQ* has announced the high performance, *SQ* can claim to be without loss of its relative superiority. These results also demonstrate the robustness of the *k*-means based quantizers with respect to those relying on the other techniques.

4.3.2. Efficiency during the out-of-sample extension

In order to evaluate the efficiency of the implemented methods, the running times of these methods are also compared. From a practical point of view, we are interested in the efficiency during the online out-of-sample extension process. Timings have been taken on an Intel Core i3-550 3.2 GHz CPU with 4 GB RAM. All these methods are implemented in Matlab platform. Fig. 5(a) and (b) respectively show the comparisons conducted on *SIFT-1M* and *MNIST*, where the average times used for encoding a sample out of the learning subset are presented.

As a result, on both the two data sets, the proposed *SQ* takes moderate computational times among the competitive algorithms. Our methods cannot outperform all the competitors in this aspect, but is computationally acceptable while guaranteeing the superior quantization results. Therefore, from an overall perspective, the proposed method can claim to be outstanding in the practical applications.

4.4. Parameter discussion

The following paragraphs start with the further investigation for *SQ*, concerning the impact of the code length on the search quality. This investigation helps make a trade-off between the computational complexity and quantization effectiveness in practical tasks. Then, the number of subspace L crucial during Cartesian product generalization is analyzed. This analysis is taken as an extension for the research of [32], through which a more generalized observation is given.

4.4.1. Code length

This section focuses on the relationship between the code length and search quality. Figs. 6 and 7 illustrate the comparisons conducted on *SIFT-1M* and *MNIST*, respectively. In both these two figures, it is observed that the search quality of the propose quantizer will improve with the increasing of the code length.

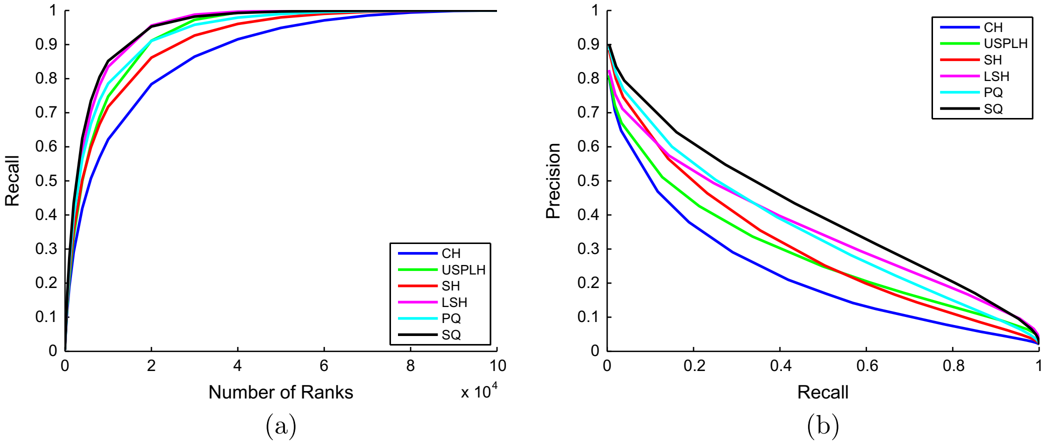
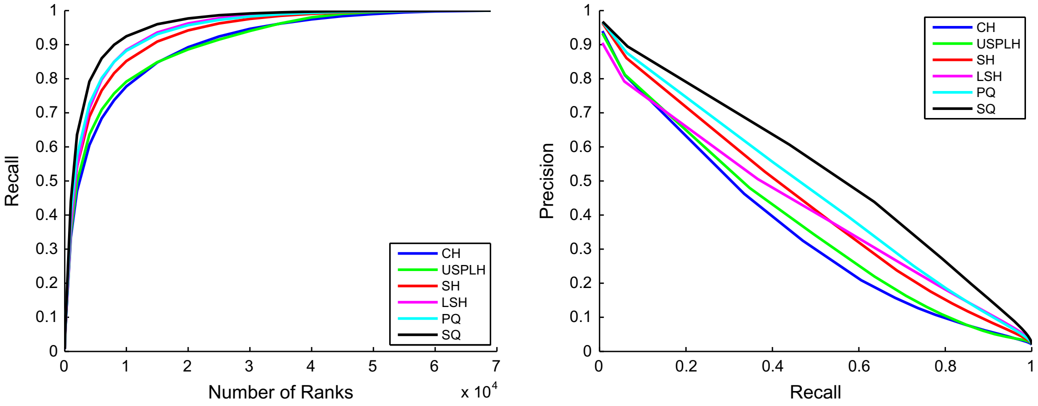
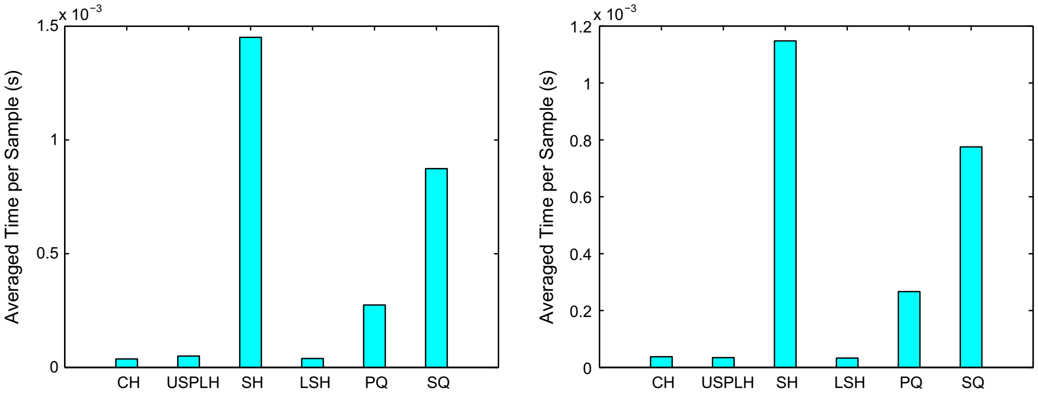


Fig. 3. Comparison of different methods on *SIFT-1M*. (a) *Recall* versus the number of returned top neighbors. (b) *Precision-recall* curves.



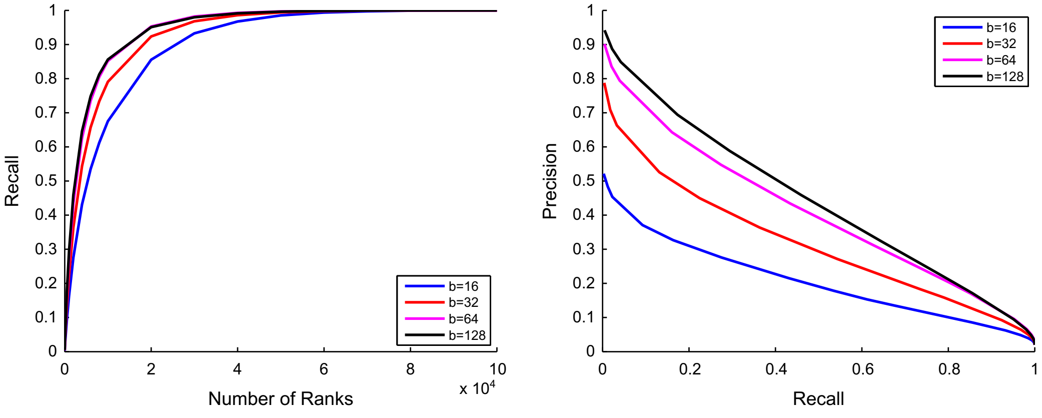
(a) (b)

Fig. 4. Comparison of different methods on *MNIST*. (a) *Recall* versus the number of returned top neighbors. (b) *Precision-recall* curves.



(a) (b)

Fig. 5. Comparison of the average time used for encoding a sample out of the learning subset on (a) *SIFT-1M* and (b) *MNIST*.



(a) (b)

Fig. 6. Comparison between the search qualities of different code lengths on the *SIFT-1M* data set. (a) *Recall* versus the number of returned top neighbors. (b) *Precision-recall* curves.

But the growth cannot remain stable but gradually slow down. As the longer code length is naturally associated with the higher complexity, it is required to make a trade-off between speed and quality in practical tasks. In our experiments, it is found that quantizing the descriptors to be 64-bits is computationally acceptable without a large sacrifice of search quality.

4.4.2. Subspace number

As previously discussed, *SQ* and *PQ* are mainly parameterized by the number L of subspaces. It is reported that for a fixed number of bits, using a small L with many sub-codewords is better than to have large L with few sub-codewords [32]. When having the equivalent number of sub-codewords $\hat{k}$ per subspace, a larger L can

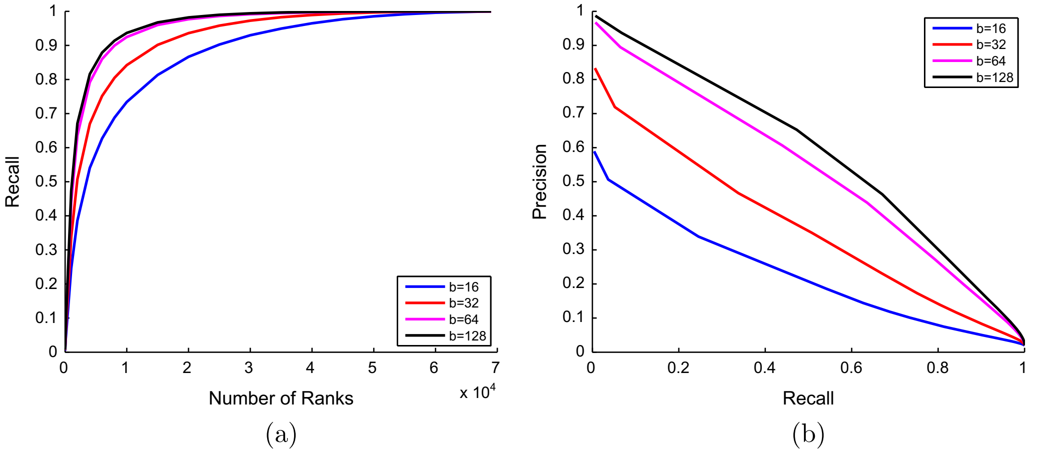


Fig. 7. Comparison between the search qualities of different code lengths on the *MNIST* data set. (a) *Recall* versus the number of returned top neighbors. (b) *Precision-recall* curves.

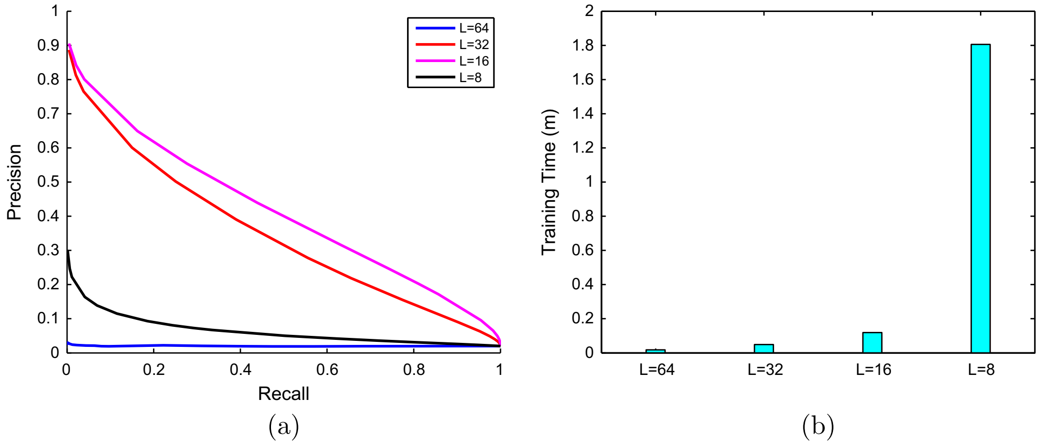


Fig. 8. Comparison between the performances under different settings of *L* on *SIFT-1M*. (a) *Precision-recall* curves. (b) Training times.

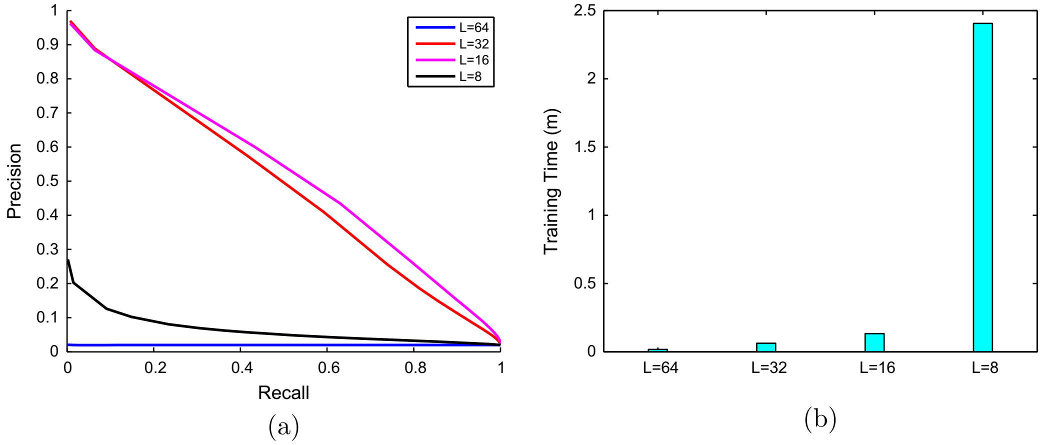


Fig. 9. Comparison between the performances under different settings of *L* on *MNIST*. (a) *Precision-recall* curves. (b) Training times.

produce better performance [32]. This conclusion is achieved by varying the value of L within a small range, i.e., from $L = 1$ to $L = 16$. However, does this tendency hold when L varying within a larger range?

To this end, this paper conducts a further verification by fixing the desired number of bits to be 64, and clustering $\hat{k} = 2^{64/L}$ sub-codewords per subspace. The verification results are shown in Figs. 8(a) and 9(a), representing the comparisons on *SIFT-1M* and *MNIST* for *SQ*, respectively. It is shown that, setting higher L cannot get a better performance when it reaches 16. Instead, the search quality even gets worse with exponential speed. The same verification is conducted for *PQ* and there follows a similar result.

In addition, the off-line training time under different *L* is also recorded for the comparison of efficiency. Timings have been taken on the same device as described in Section 4.3.2. Figs. 8(b) and 9(b) respectively illustrate the comparison on *SIFT-1M* and *MNIST*. As can be observed, the training time drops greatly with the use of larger *L*. This demonstrates that splitting the original feature space into product spaces is indeed an efficient solution. From an overall perspective, it is reasonable to recommend selecting *L* flexibly in practical tasks.

5. Conclusion

Though it has attracted much attention and achieved laudable success in many applications, there is a problem of the traditional *k*-means quantizers that is seldom realized. During the out-of-sample extension process, the traditional *k*-means quantizers assign to new samples the codewords relying only on the similarities in Euclidean space without considering the statistical and geometrical properties of the data. In spite of minimal quantization error during off-line training, this strategy lacks statistical interpretability, and may fail in many practical cases. Toward this problem, in this paper a novel quantizer is proposed, which determines the affiliations of the new samples with respect to the obtained Voronoi cells through the GLR analysis. The formulated out-of-sample extension process is physically meaningful and practically preferable.

Experiments are conducted on two public data sets, *SIFT-1M* and the *MNIST*, through which the proposed method is proved to be computationally acceptable while guaranteeing the superior quantization results. The impact of the code length on the search quality is also analyzed to help make a trade-off between the computational complexity and quantization effectiveness. Furthermore, the most crucial parameter during Cartesian product generalization is extensively investigated. This investigation provides a more generalized observation extending the existing works.

Acknowledgments

This work is supported by the State Key Program of National Natural Science of China (Grant No. 61232010), the National Natural Science Foundation of China (Grant Nos. 61172143, 61105012, and 61379094), and the Natural Science Foundation Research Project of Shaanxi Province (Grant No. 2012JM8024).

References

[1] Y. Lin, R. Jin, D. Cai, S. Yan, X. Li, Compressed hashing, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 446–451.

[2] J. Wang, S. Kumar, S.-F. Chang, Semi-supervised hashing for large-scale search, IEEE Trans. Pattern Anal. Mach. Intell. 34 (12) (2012) 2393–2406.

[3] Y. Yang, Y. Zhuang, F. Wu, Y. Pan, Harmonizing hierarchical manifolds for multimedia document semantics understanding and cross-media retrieval, IEEE Trans. Multim. 10 (3) (2008) 437–446.

[4] K. Kim, M.K. Hasan, J.-P. Heo, Y.-W. Tai, S.-E. Yoon, Probabilistic cost model for nearest neighbor search in image retrieval, Comp. Vis. Image Understand. 116 (9) (2012) 991–998.

[5] D. Gorisse, M. Cord, F. Precioso, Locality-sensitive hashing for chi2 distance, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2) (2012) 402–409.

[6] J. Choi, Z. Wang, S.-C. Lee, W.J. Jeon, A spatio-temporal pyramid matching for video retrieval, Comp. Vis. Image Understand. 117 (6) (2013) 660–669.

[7] P. Li, M. Wang, J. Cheng, C. Xu, H. Lu, Spectral hashing with semantically consistent graph for image indexing, IEEE Trans. Multim. 15 (1) (2013) 141–152.

[8] J. Song, Y. Yang, Z. Huang, H.T. Shen, R. Hong, Multiple feature hashing for real-time large scale near-duplicate video retrieval, in: ACM Multimedia, 2011, pp. 423–432.

[9] M. Guillaumin, T. Mensink, J.J. Verbeek, C. Schmid, Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation, in: IEEE International Conference on Computer Vision, 2009, pp. 309–316.

[10] E. Moxley, T. Mei, B.S. Manjunath, Video annotation through search and graph reinforcement mining, IEEE Trans. Multim. 12 (3) (2010) 184–193.

[11] W. Zhao, X. Wu, C.-W. Ngo, On the annotation of web videos by efficient near-duplicate search, IEEE Trans. Multim. 12 (5) (2010) 448–461.

[12] Z. Liu, Q. Pan, J. Dezert, A new belief-based *k*-nearest neighbor classification method, Patt. Recog. 46 (3) (2013) 834–844.

[13] S. Wang, Q. Huang, S. Jiang, Q. Tian, Nearest-neighbor classification using unlabeled data for real world image application, in: ACM Multimedia, 2010, pp. 1151–1154.

[14] F. Çakir, U. Güdükbay, Ö. Ulusoy, Nearest-neighbor based metric functions for indoor scene recognition, Comp. Vis. Image Understand. 115 (11) (2011) 1483–1492.

[15] J. Brandt, Transform coding for fast approximate nearest neighbor search in high dimensions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1815–1822.

[16] B. Kulis, K. Grauman, Kernelized locality-sensitive hashing, IEEE Trans. Pattern Anal. Mach. Intell. 34 (6) (2012) 1092–1104.

[17] M.M. Esmaeili, R.K. Ward, M. Fatourechi, A fast approximate nearest neighbor search algorithm in the hamming space, IEEE Trans. Pattern Anal. Mach. Intell. 34 (12) (2012) 2481–2488.

[18] S. Har-Peled, P. Indyk, R. Motwani, Approximate nearest neighbor: towards removing the curse of dimensionality, Theory Comput. 8 (1) (2012) 321–350.

[19] R. Weber, H.-J. Schek, S. Blott, A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces, in: International Conference on Very Large Data Bases, 1998, pp. 194–205.

[20] A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in: International Conference on Very Large Data Bases, 1999, pp. 518–529.

[21] J.H. Friedman, J.L. Bentley, R.A. Finkel, An algorithm for finding best matches in logarithmic expected time, ACM Trans. Math. Softw. 3 (3) (1977) 209–226.

[22] C. Silpa-Anan, R. Hartley, Optimised kd-trees for fast image descriptor matching, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[23] G. Shakhnarovich, T. Darrell, P. Indyk, Nearest-Neighbor Methods in Learning and Vision: Theory and Practice, vol. 3, MIT press, 2005.

[24] Y. Gong, S. Lazebnik, Iterative quantization: a procrustean approach to learning binary codes, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 817–824.

[25] B. Kulis, T. Darrell, Learning to hash with binary reconstructive embeddings, in: Conference on Neural Information Processing Systems, 2009, pp. 1042–1050.

[26] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, S.-F. Chang, Supervised hashing with kernels, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2074–2081.

[27] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: Conference on Neural Information Processing Systems, 2008, pp. 1753–1760.

[28] D. Zhang, F. Wang, L. Si, Composite hashing with multiple information sources, in: ACM SIGIR, 2011, pp. 225–234.

[29] X. Liu, J. He, D. Liu, B. Lang, Compact kernel hashing with multiple features, in: ACM Multimedia, 2012, pp. 881–884.

[30] X. Liu, J. He, B. Lang, Multiple feature kernel hashing for large-scale visual search, Patt. Recog. 47 (2) (2014) 748–757.

[31] R. Gray, Vector quantization, ASSP Mag. 1 (2) (1984) 4–29.

[32] H. Jégou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, IEEE Trans. Pattern Anal. Mach. Intell. 33 (1) (2011) 117–128.

[33] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1704–1716.

[34] K. He, F. Wen, 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, pp. 2938–2945.

[35] R.M. Gray, D.L. Neuhoff, Quantization, IEEE Trans. Inform. Theory 44 (6) (1998) 2325–2383.

[36] S.P. Lloyd, Least squares quantization in PCM, IEEE Trans. Inform. Theory 28 (2) (1982) 129–136.

[37] I.S. Reed, X. Yu, Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution, IEEE Trans. Acoust., Speech Signal Process. 38 (10) (1990) 1760–1770.

[38] R.J. Muirhead, Aspects of Multivariate Statistical Theory, vol. 197, Wiley.com, 2009.

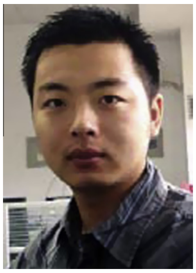
[39] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[40] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comp. Vis. 60 (2) (2004) 91–110.

[41] H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: European Conference on Computer Vision, 2008, pp. 304–317.

[42] M. Norouzi, A. Punjani, D.J. Fleet, Fast search in hamming space with multi-index hashing, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3108–3115.

Qi Wang received the B.E. degree in automation and the Ph.D. degree in pattern recognition and intelligent system from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. He is currently an associate professor with the Northwestern Polytechnical University, Xi'an, China. His current research interests include computer vision and pattern recognition.



Guokang Zhu is currently working toward the Ph.D. degree in the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China. His current research interests include computer vision and machine learning.

Yuan Yuan is a full professor with the Chinese Academy of Sciences (CAS), China. She has published over 150 papers, including about 90 in reputable journals such as IEEE transactions and Pattern Recognition, as well as conferences papers in CVPR, BMVC, ICIP, and ICASSP. Her current research interests include visual information processing and image/video content analysis.