Gradient Ordinal Signature and Fixed-Point Embedding for Efficient Near-Duplicate Video Detection

Hong Liu, *Member, IEEE,* Hong Lu, *Member, IEEE,* Zhaohui Wen, and Xiangyang Xue, *Member, IEEE*

***Abstract*—In order to meet the requirement of large scale real-time near-duplicate video detection, this paper has achieved two goals. First, this paper proposes a more compact local image descriptor which is termed as gradient ordinal signature (GOS). GOS not only has the advantages of low dimension, simplicity in computation, and high discrimination but also is invariant to mirror reflection, rotation, and scale changes. Second, applying the characteristics of the proposed GOS and combining with the embedding theory of metric spaces, this paper proposes an efficient similarity search method based on the fixed-point embedding (FE). A main advantage of FE is that its parameters have good controllability, and its performance is stable and not sensitive to dataset changes. On the whole, the goal of our approach focuses on the speed rather than the accuracy of near-duplicate video detection. We have evaluated our method on four different settings to verify the two goals. Specifically, the tests include image and video datasets, respectively, to evaluate the performance of GOS. Experimental results demonstrate the effectiveness, efficiency, and lower memory usage of GOS. Furthermore, the third test compares FE with locality sensitivity hashing. FE also shows a speed improvement of about ten times and saves more than 60% in memory usage. The fourth test demonstrates that the combination of GOS and FE for near-duplicate video detection can achieve better overall efficiency than the state-of-the-art methods.**

***Index Terms*—Fixed-point embedding (FE), gradient ordinal signature (GOS), local feature, near-duplicate video detection, similarity search.**

I. Introduction

WITH THE RAPID development and wide application of multimedia hardware and software technologies, the cost of image and video data collection, creation, and storage is becoming increasingly low. Each day tens of thousands of video data are generated and published. Among this huge volume of videos, there exist large numbers of copies or near-duplicate videos. According to the statistics of [1], on average, there are 27% redundant videos (duplicate or nearly duplicate) to the most popular version of a video in the search results from video search engines such as Google Videos, YouTube, and Yahoo! Video. As a consequence, near-duplicate video detection or video copy detection has become a topic of intensive study recently [1]–[18], and has been applied to various emerging applications. In this field, researchers focus on three aspects: 1) extracting a stable and discriminative image feature [1]–[7], [9], [11], [12], [15]–[17]; 2) finding a suitable indexing scheme for efficient image similarity search [10], [17], [19], [20]; 3) determining how to use temporal information of video sequence for video copy detection [1], [3], [4], [6]–[9], [11], [13], [15], [17]. A stable and discriminative image feature is an essential component to enable effective video copy detection. This study shows that local features perform better than global features for video copy detection. Among the local image features, scale invariant feature transform (SIFT) descriptor [21] is an effective descriptor and has been used in a wide range of applications [12], [14], [15], [17], [22], [23]. However, SIFT usually extracts a large number of interest points from an image, and each interest point is described by a 128-D high-dimensional feature, so it is difficult to meet the requirement of large scale real-time near-duplicate video detection. In order to address "the curse of dimensionality" problem in high-dimensional space, many indexing structures have been proposed, such as the tree indexing structure family [24]–[27], vector approximation file (VA-file) [28], locality sensitive hashing (LSH) [29], and ordered vector approximation file (OVA-file) [30]. These indexing methods obtain good results in some applications such as video retrieval, near-duplicate video detection, and others [1], [10], [14], [15], [30].

In practice, our goal is to design a large scale web near-duplicate video detection system, which requires high response speed (i.e., in real time). Thus, in this paper, our study focuses on two aspects: 1) finding a more compact descriptor to effectively reduce the feature dimension; and 2) finding an efficient similarity search method to reduce the search scope. In the use of temporal information of video sequence, our system adopts the temporal grouping method [15], [17] which obtains the best performance in all the submitted runs of TRECVID 2008 [15].

This paper is organized as follows. Section II reviews the related work. Section III describes the proposed gradient

Manuscript received August 3, 2010; revised January 16, 2011; accepted August 10, 2011. Date of publication November 3, 2011; date of current version April 2, 2012. This work was supported in part by the 973 Program, under Project 2010CB327900, in part by the Natural Science Foundation of China, under Grants 60875003 and 60873178, in part by the Shanghai Committee of Science and Technology, China, under Grant 11ZR1403400, and in part by the 211-Project Sponsorship for Young Professors. This paper was recommended by Associate Editor H. Gharavi.

The authors are with the Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai 200433, China (e-mail: liuhong2007@fudan.edu.cn; honglu@fudan.edu.cn; 082024059@fudan.edu.cn; xyxue@fudan.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2168294

1051-8215/$26.00 © 2011 IEEE

ordinal signature (GOS). Section IV describes the fixed-point embedding (FE)-based similarity search method on the proposed GOS. Section V is the evaluation of our proposed methods, and this paper is summarized in Section VI.

II. Related Work

In this section, we first review the work on near-duplicate video detection, local feature descriptor, and similarity search. Specially, we introduce in detail the embedding methods for similarity search that our approach is based on. And then we introduce our approach briefly.

A. Near-Duplicate Video Detection

Most existing content-based near-duplicate video detection methods can be grouped into two types [9]: methods based on global image features and that based on local image features. Hampapur *et al.* [2], [3] compared distance measures and video sequence matching methods for video copy detection. They employed convolution for motion direction feature, L_1 distance for ordinal intensity signature, and histogram intersection for color histogram feature. Comparison results showed that the ordinal intensity signature performed better. Reference [7] took the combination of ordinal intensity signature and color histogram features as a video sequence descriptor. Based on [3], the region intensity rank signature along time sequence was proposed in [4] and [6]. These methods focused on the rapid identification of duplicate videos with global features, which were able to handle almost identical videos. However, duplicates with complex editing can only be reliably detected through the use of more reliable local features. Spatiotemporal interest points were employed to classify human actions and to detect periodic movement [31]. References [14] and [16] used keypoint-based method for near-duplicate image detection and subimage detection. References [9] and [10] used Harris corner points [32] to obtain feature points in video frames. Besides, [9] described the trajectories' or trajectory characteristic of feature points and labels. Similarly, Satoh *et al.* detected duplicate scenes by using the trajectories' or trajectory characteristic based on SIFT feature [11]. These studies show that the performance of the detection methods based on local features performs better than the methods based on global features.

B. Local Feature Descriptor

The local descriptors such as point, line, and shape play an important role in image retrieval. Specifically, the local descriptor based on point has a wide range of applications. These descriptors mainly include SIFT [21], principal component analysis-based SIFT (PCA-SIFT) [22], speeded up robust features (SURF) [33], gradient location and orientation histogram (GLOH) [34], SIFT-rank [35], and others. Reference [34] made a comparative study on various local descriptors, such as SIFT, PCA-SIFT, GLOH, and others. Evaluation results showed that SIFT descriptor has the best performance on object recognition. Specifically, in [21], SIFT used difference of Gaussian (DoG) detector to extract interest points. After DoG detection, each interest point (also called key point) has been assigned with an image location, scale, and orientation. In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated to the main orientation of the interest point. The SIFT descriptor is generated by first computing the gradient magnitude and orientation at each image sample point in a region around the interest point location. These samples are then accumulated into eight-orientation histograms summarizing the contents over 4×4 subregions. Finally, a $8 \times 4 \times 4 = 128$-D descriptor is obtained. PCA-SIFT applied PCA on the gradient image to yield a 36-D descriptor which is fast for matching [22]. SURF focused on rotation and scale attacks in order to obtain a tradeoff between feature complexity and robustness [33]. GLOH was another variant of SIFT which was proved to be more distinctive and robust than the original SIFT. However, its computational cost is more expensive than the original SIFT [34]. SIFT-rank considered SIFT descriptor vector elements in terms of their ranks in an array sorted according to measured values [35]. In spite of the above modified versions, the original SIFT is still the most popular. However, high dimensionality of Lowe's SIFT descriptor makes it difficult to meet the requirements for large-scale real-time near-duplicate video detection.

C. Similarity Search

The simplest similarity search method is sequential scanning, which in turn computes the similarity between the query feature and the reference features in database, and then returns k-nearest neighbors or ϵ-range query results. However, sequential scanning will bring a high computational cost for a large-scale database. In order to resolve the efficient similarity search problem, many high-dimensional indexing structure methods have been proposed. These methods include tree indexing structure family [24]–[27], VA-file [28], LSH [29], OVA-file [30], and others. The performances of R-tree index structure family are known to degrade seriously when the feature dimension increases. Weber *et al.* [36] proposed an approximate version of the VA-file which is about five times faster than the exact version when 20% of the exact kNN is lost. The main advantage of VA-file is that it has high data compression ratio and is profitable for disk storage. The main advantage of LSH is that it is sublinear in database size and it tolerates high dimensionality. However, the quality of the query results can be poor and their accuracy can hardly be controlled [37]. Clustering-based approximate methods have also been proposed to achieve substantial speedups over sequential scan [38]–[40]. Sivic and Zisserman [20] introduced the bag-of-features (BOFs) image representation in the context of image search. Descriptors are quantized into visual words using k-means clustering algorithm. The approach of [40] used the well-known k-means heuristic to generate a large number of small clusters. These search algorithms make use of a fast indexing structure over the set of clusters and achieve speedup of roughly 21 times over sequential search. The main drawback of the k-means clustering algorithm is that it is difficult to determine the number of clusters and could obtain poor-quality query results. The methods of [19], [29], [41], and [42] are based on a binary representation of the vectors

and use a simple Hamming distance for distance comparison. These methods are very fast but the quality of the results can be poor and their accuracy is hard to control.

D. Embedding Methods for Similarity Search

In this paper, we apply embedding metric spaces theories to propose a FE-based similarity search method (see Section IV for detail). In application areas, its objective is to find embeddings of metric spaces into other more simple and structured spaces that have low distortion [43]–[46].

Formally, an embedding of a finite metric space (S, d) into (R^k, d') is a mapping $F : S \rightarrow R^k$, where k is the dimensionality of the embedding space and $d' : R^k \times R^k \rightarrow R^+$ the distance metric of the embedding space. If we denote the norm in R^k with $||\cdot||$, the distance metric d' is defined as $d'(\mathbf{u}, \mathbf{v}) = ||\mathbf{u} - \mathbf{v}||$. Usually, the norm is one of the L_p norms, $||\mathbf{u}||_p = (\sum |u_k|^p)^{1/p}$ $(1 \leq p \leq \infty)$. The most common distance metrics based L_p norms are the Euclidean distance metric (L_2), the City Block distance metric (L_1), and the Chessboard distance metric (L_∞). An embedding is *contractive* if for any $\mathbf{u} \neq \mathbf{v} \in S$, there exists $d'(F(\mathbf{u}), F(\mathbf{v})) \leq d(\mathbf{u}, \mathbf{v})$.

A fundamental class of embedding methods is known as Lipschitz embeddings [47], [48]; the basic theory of embedding into finite metric spaces has been proposed by Bourgain [48].

Theorem 1: For every N-point metric space there exists an embedding into Euclidean space with distortion $O(logN)$.

One of the most important parameters of the embedding into a normed space is the *dimension* of the embedding. For embedding into Euclidean space, the dimension reduction lemma in [47] states that any N-point metric space in L_2 can be embedded in Euclidean space of dimension $O(logN)$ with constant distortion. This reduces the dimension in Bourgain's theorem to $O(logN)$. In [43], it is shown that Bourgain's embedding provides an embedding into L_p with distortion $O(logN)$ and dimension $O(log^2 N)$.

Unfortunately, the embedding of [43] is rather impractical for similarity searching because the dimensionality of the embedding space is relatively large, i.e., $O(log^2 N)$. Even with as few as 64 feature vectors, the number of dimensions is 36, which is too high to index efficiently. To address this problem, Abraham *et al.* [46], with only a small price to pay in distortion, provides an embedding into low dimension.

Theorem 2: For any $1 \leq p \leq \infty$, and $\theta > 0$, every N-point metric space embeds in L_p with distortion $O(log^{1+\theta} N)$ in dimension $O(\theta^{-1} logN / loglogN)$.

For Theorem 2, it is worth noting that with the small loss in the distortion, the dimension is reduced.

How much larger or smaller the distance d' in the embedding space than the corresponding distance d in the original space is important to the quality of similarity search. Hjaltason and Samet [44], [45] introduce some *contractive* embedding methods for similarity searching in metric spaces. They point out that the contractive property of the embedding is a very useful property in similarity search. If the mapping F is contractive, efficient nearest neighbor query algorithms can be implemented that give an exact result. Of course, in order to get a tradeoff between accuracy and efficiency, it is possible at a small price in a small error. Such algorithms usually use a "filter" and "refine" strategy to carry out similarity searching [49], [50]. In particular, in the "filter" step, the embedding space is used as a filter to produce a set of candidates. The satisfaction of the contractive property makes it possible to guarantee that the correct result is among the candidates. In the "refine" step, the actual distance must be computed for all the candidates to determine the actual nearest neighbor.

E. Our Approach

This paper proposes a novel local image descriptor based on SIFT, which is termed as GOS. Its goal is not to get a better description ability than the standard SIFT descriptor, but to improve efficiency on the basis of ensuring a relatively high accuracy. GOS not only has the advantages of low dimension and simplicity in computation but also is invariant to mirror reflection, rotation, and scale changes. Furthermore, because the proposed GOS is actually a fixed dimension rank-ordering feature, we analyze the characteristics of the rank-ordering feature and apply the embedding methods of metric spaces to propose an efficient similarity search method based on the FE. The aim of our approach is not to get a higher accuracy, but to increase the speed. Compared with locality sensitivity hashing (LSH), FE improves about ten times speed and saves more than 60% memory usage. Experimental results show that GOS obtains the accuracy close to other state-of-the-art descriptors, but greatly enhances search speed and cuts memory usage. Significantly, the combination of GOS and FE for near-duplicate video detection can achieve better overall efficiency than the compared state-of-the-art methods.

III. Gradient Ordinal Signature

As described in Section II, the local descriptor has a wide range of applications. In this paper, based on ordinal description, we propose a more compact local image descriptor which is termed GOS. Ordinal description does not take into account the original measurement values themselves, but the rank values of the original measurement values. It can be used as a general strategy for comparing sets of data measurements. Spearman correlation coefficient [51] or the Kendall coefficient [52] studied the similarity of ordinal or rank-ordered data. Furthermore, some applications based on ordinal features are proposed [3], [6], [35], [53]. Among these methods, the ideas of [35] are similar to ours. However, [35] directly converted the measurement value of SIFT descriptors to rank-ordering values, then obtained a 128-D SIFT-rank features. Usually, the standard SIFT algorithm will extract a large number of interest points from an image; each interest point is described using a 128-D high-dimension feature, it is thus difficult to be applied to large-scale real-time near-duplicate video detection. This paper improves upon the standard SIFT descriptor and proposes the GOS. GOS has the advantages of low dimension and simplicity in computation. At the same time, GOS not only retains some properties of original SIFT descriptor (such as rotation and scale invariance) but also has

its own mirror reflection invariance property. Its final goal is not to obtain a better description ability than the SIFT descriptor, but to improve efficiency on the basis of ensuring a relatively high detection accuracy.

A. GOS Representation

This paper directly performs the Harris-Affine interest point detector [54] to detect interest points. The Harris-Affine interest point detector first computes a multiscale representation for the Harris interest point detector [32] and then selects points at which a local measure is maximal over scales. This method provides a set of distinctive points which are invariant to scale and rotation changes. In order to extend this approach to make it affine invariant, the Harris-Affine interest point detector estimates the affine shape of a point neighborhood and then converges to affine invariant points with an iterative algorithm to modify the location, scale, and neighborhood of each point. After interest point detection, each interest point has been assigned an image location, scale, and orientation. This information can also be used to assist feature matching [19], [17] and thus be stored by GOS. For GOS extraction, a square region around the interest point is considered. First, similar to SIFT, in order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated to the interest point's main orientation. Then the square region is divided into M columns, and the gradient magnitude average value of the pixels in each column is computed. Finally, we can obtain a gradient ordinal sequence for each interest point. The extraction process is demonstrated in Fig. 1. The number of columns, M, can be used to vary the dimensionality of GOS. A GOS descriptor with larger M can be be more discriminative in a large database, but it would also be more sensitive to shape distortions and occlusion. Our empirical study shows that dividing the square region into 16 columns can obtain better results.

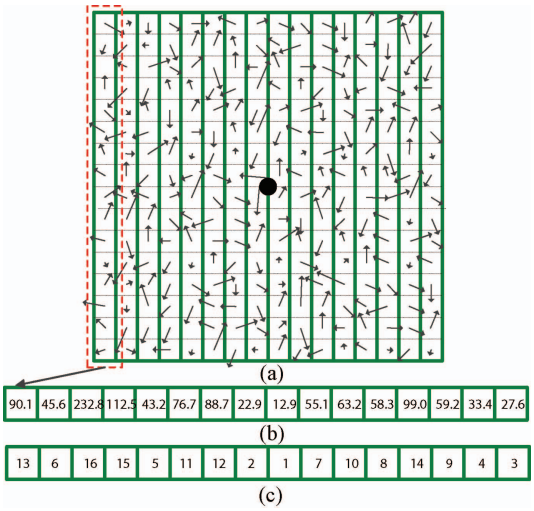


Fig. 1. Extraction of GOS. (a) Compute the gradient of each pixel in a small image patch ($N \times N$ pixels) around an interest point, the small image patch is divided into M columns. (b) Compute the average gradient magnitude in each column. (c) Determine the ordinal position of each column when sorted by increasing gradient magnitude value. Concatenate the order values of M columns in left-to-right manner to obtain the M-D GOS descriptor. In this figure, the GOS feature vector is 16-D feature vector (13, 6, 16, 15, 5, 11, 12, 2, 1, 7, 10, 8, 14, 9, 4, 3).

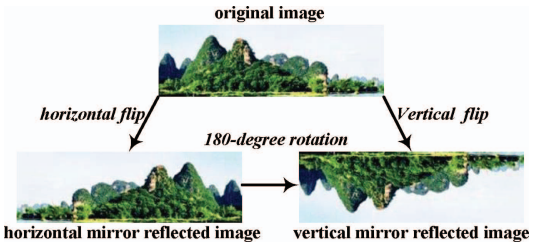


Fig. 2. Relationship between horizontal mirror reflection and vertical mirror reflection.

B. Mirror Reflection Invariance of GOS

In real application, we often find that an image itself is of mirror reflection characteristic or two images are of mirror reflection, such as an image reflected in the water or mirror. Reference [23] obtained a mirror reflection invariant feature by a way which reorganizes the order of cells and restructures the order of orientation bins in each cell. In this paper, we use a simple method to make GOS mirror reflection invariant. Specifically, mirror reflection includes horizontal mirror reflection, vertical mirror reflection, and a combination of these two reflections. Now we analyze the relationship between horizontal mirror reflection and vertical mirror reflection. Given an original image $A(x, y)$, assuming $B(x, y)$ is the horizontal mirror reflected image of A and $C(x, y)$ is the vertical mirror reflected image of A. Denote the image intensities by I, the coordinates of the pixel, (x, y), are relative to the center of the image. Then the intensities of image A, I_A, the intensities of image B, I_B, and the intensities of image C, I_C, are given by $I_B(x, y) = I_A(-x, y)$ and $I_C(x, y) = I_A(x, -y)$. So

$$I_C(x, y) = I_B(-x, -y). \tag{1}$$

By (1), we know that the vertical mirror reflected image can be obtained by rotating its horizontal mirror reflected version at 180 degrees. Because GOS is invariant to image rotation, so for GOS, horizontal mirror reflection is equivalent to vertical mirror reflection (as illustrated in Fig. 2).

Next, we select the vertical mirror reflection for analysis and explain the mirror reflection invariance of GOS. As shown in Fig. 3, the gradient magnitude of each pixel in the vertical mirror reflected image C, $m_C(x, y)$, can be indicated by the gradient magnitude of each pixel in original image A, $m_A(x, y)$, as in (2).

By (2), shown at the bottom of the next page, we can draw that the gradient magnitude of the *first* pixel in the first column in Fig. 3(a) equals to the gradient magnitude of the *last* pixel in the first column in Fig. 3(b), the gradient magnitude of the *second* pixel in the first column in Fig. 3(a) equals to the gradient magnitude of the *penultimate* pixel in the first column in Fig. 3(b), ..., and so on. That means, the original image is vertically flipped, and the gradient magnitude is also flipped along horizontal axis. However, the sum of gradient magnitude of each column remains unchanged, so the gradient magnitude average value of each column in Fig. 3(a) equals

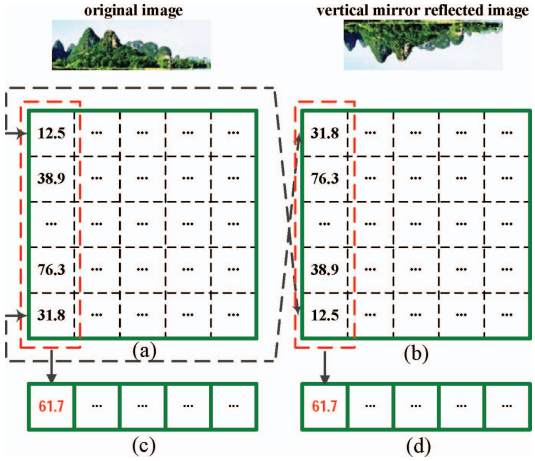


Fig. 3. Mirror reflection invariance of GOS.

to the gradient magnitude average value of the corresponding column in Fig. 3(b); then, the finally extracted GOS descriptor is invariant to mirror reflection.

In conclusion, GOS not only has the advantages of low dimension and simplicity in computation but also is invariant to mirror reflection, rotation, and scale changes. At the same time, GOS is not sensitive to luminance changes because global luminance changes cannot change the ranking result of GOS feature vector. Experimental results show that GOS obtains the accuracy close to other state-of-the-art descriptors, but improves search speed and cuts memory usage (see Section V for detail information).

IV. Fixed-Point Embedding-Based Similarity Search Based on GOS

In order to deal with large image datasets, Sivic and Zisserman introduce the BOFs image representation in the context of image search [20]. Specifically, BOF image search uses descriptor quantization, and descriptors are quantized into visual words using k-means clustering algorithm. In [19], Jegou *et al.* proposed "Hamming embedding" (HE) and "weak geometric consistency" (WGC) constraints based on [20]. HE provides d_b-D binary signature that refines the matching based on visual words. WGC filters the matching descriptors that are not consistent in terms of angle and scale. BOF approach obtains the "centroid" by k-means clustering algorithm, and k is usually obtained by learning on a subset of the data. Since k is sensitive to dataset change, the quality of the similarity search results based on BOF can hardly be controlled. In this paper, we use some characteristics (see Section IV-A for detail) of GOS to propose a FE-based similarity search method. Unlike [19] and [20], FE finds a tuple of suitable quantization functions to map every high-dimensional vector to a tuple of integer values, then each tuple of integer values is encoded to obtain the "centroid" which is called "index ID" in this paper. Finally, we establish an inverted index structure based on FE for efficient similarity search. A main advantage of FE is that its parameters have good controllability, and its performance is stable and not sensitive to dataset change. We will elaborate on its basic principle and the realization of the similarity search as follows.

TABLE I
Notations in Our Paper

Notation	Description
$\mathbf{v}$	Vector (bold text)
v_k	kth component of the vector $\mathbf{v}$
$d_{L_p}(\mathbf{v_i}, \mathbf{v_j})$	Distance metric between $\mathbf{v_i}$ and $\mathbf{v_j}$ with L_p
$q(.)$	Quantization function

A. Principle

In order to facilitate description, Table I summarizes notations in this paper.

According to the representation in Section III, each GOS feature vector is a rank-ordering vector which has two key characteristics.

1) *Characteristic 1:* For any t-D rank-ordering feature vector $\mathbf{v} = (v_1, v_2, \ldots, v_t)$, it satisfies the following equation:

$$\sum_{k=1}^{t} v_k^m = C_m. \tag{3}$$

2) *Characteristic 2:* For any dataset of t-D rank-ordering feature vectors, the upper bound of the number of different feature vectors in this dataset is $t!$. In application, we can use this characteristic to obtain the upper bound of the size of dataset. For example, a 16-D rank-ordering feature vector space, which consists of at most $16! = 20\,922\,789\,888\,000 \approx 2^{44}$ different feature vectors. In this paper, we can apply it to determine the number of selected fixed points.

From Characteristic 1, we can obtain some valuable information as follows.

1) If we denote the L_p norm with $||\mathbf{v}||_p = (\sum |v_k|^p)^{1/p} (1 \leq p \leq \infty)$, then for any t-D rank-ordering feature vector $\mathbf{v} = (v_1, v_2, \ldots, v_t)$, $||\mathbf{v}||_p = (\sum_{k=1}^{t} v_k^p)^{1/p} = (\sum_{k=1}^{t} k^p)^{1/p}$, namely, the p-norms of the rank-ordering feature vector relates only to its dimensionality. Specially, $||\mathbf{v}||_1 = t(t+1)/2$, $||\mathbf{v}||_2 = \sqrt{t(t+1)(2t+1)/6}$, and $||\mathbf{v}||_\infty = t$. Once the dimensionality of GOS feature vector is fixed,

$$\begin{aligned} m_C(x, y) &= \sqrt{(I_C(x+1, y) - I_C(x-1, y))^2 + (I_C(x, y+1) - I_C(x, y-1))^2} \\ &= \sqrt{(I_A(x+1, -y) - I_A(x-1, -y))^2 + (I_A(x, -(y+1)) - I_A(x, -(y-1))^2} \\ &= \sqrt{(I_A(x+1, -y) - I_A(x-1, -y))^2 + (-(I_A(x, -(y-1)) - I_A(x, -(y+1)))^2} \\ &= \sqrt{(I_A(x+1, -y) - I_A(x-1, -y))^2 + (I_A(x, -y+1) - I_A(x, -y-1)^2} \\ &= m_A(x, -y). \end{aligned} \tag{2}$$

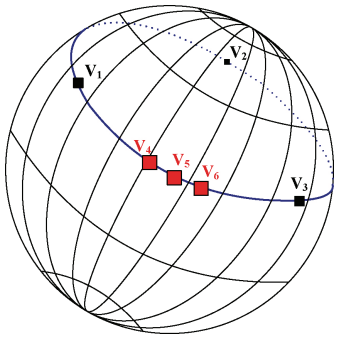


Fig. 4. Visualization of the geometric of GOS. Intuitively, by means of (6), the GOS feature points should be distributed on a hyper dome (blue) in the feature space. $\{\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}\}$ and $\{\mathbf{v_4}, \mathbf{v_5}, \mathbf{v_6}\}$ denote two groups of different fixed points.

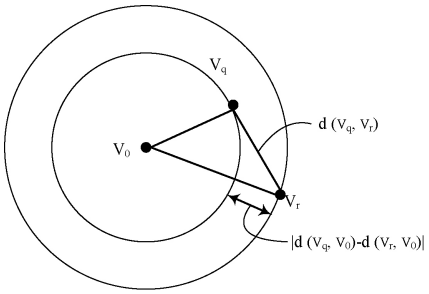


Fig. 5. Demonstration of the triangle inequality $|d(\mathbf{v_q}, \mathbf{v_0}) - d(\mathbf{v_r}, \mathbf{v_0})| \leq d(\mathbf{v_q}, \mathbf{v_r})$.

its L_p norm also becomes a constant. In practical application, it can avoid normalization processing and reduce computational cost.

2) Given any two rank-ordering feature vectors $\mathbf{v_i}$ and $\mathbf{v_j}$, the distance $d_{L_p}(\mathbf{v_i}, \mathbf{v_j})$ exists an upper bound

$$0 \leq d_{L_p}(\mathbf{v_i}, \mathbf{v_j}) \leq d_{L_{p\max}} = \left(\sum_{k=1}^{t} |(t-k+1)-k|^p \right)^{1/p}. \quad (4)$$

3) Characteristic 1 also shows some geometric properties of GOS feature vector. For (3), when m takes 1 and 2, we can obtain the following two equations:

$$v_1 + v_2 + \ldots + v_t = C_1 = t(t+1)/2 \quad (5)$$

$$v_1^2 + v_2^2 + \ldots + v_t^2 = C_2 = t(t+1)(2t+1)/6. \quad (6)$$

Equations (5) and (6) give us some geometric intuition which is shown in Fig. 4. Specifically, (5) indicates that GOS feature points are distributed on a hyper-plane in the feature space. And (6) indicates that GOS feature points are distributed on a hyper-sphere in the feature space. Combining (5) and (6), the GOS feature points should be distributed on a hyper dome in the feature space.

Deriving from the above characteristics of GOS feature vector, intuitively, we can select in advance some fixed points (such as $\{\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}\}$ or $\{\mathbf{v_4}, \mathbf{v_5}, \mathbf{v_6}\}$ in Fig. 4), then in similarity search, we can measure the similarity between query and retrieved-target vector by comparing their distances with these fixed points. For example, $\mathbf{v_0}$ is some fixed point, some information about the distance $d(\mathbf{v_q}, \mathbf{v_r})$ between the query point $\mathbf{v_q}$ and the retrieved-target point $\mathbf{v_r}$ can be obtained by comparing $d(\mathbf{v_q}, \mathbf{v_0})$ and $d(\mathbf{v_r}, \mathbf{v_0})$, i.e., the value $|d(\mathbf{v_q}, \mathbf{v_0}) - d(\mathbf{v_r}, \mathbf{v_0})|$. This is especially true if one of the distances $d(\mathbf{v_q}, \mathbf{v_0})$ and $d(\mathbf{v_r}, \mathbf{v_0})$ tends to zero. Observe that due to the triangle inequality we have $|d(\mathbf{v_q}, \mathbf{v_0}) - d(\mathbf{v_r}, \mathbf{v_0})| \leq d(\mathbf{v_q}, \mathbf{v_r})$, as illustrated in Fig. 5.

Therefore, based on the embedding theory of metric spaces (which has been introduced in Section II), this paper proposes a FE similarity search method for order-ranking feature. FE is based on a simple idea that, if two points are close together, then after an "embedding" operation these two points will remain close together. FE first selects a reference set R which consists of l fixed points $(\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_l})$, a regular Lipschitz embedding with respect to R is defined as a mapping F such that $F(\mathbf{v}) = (d(\mathbf{v}, \mathbf{v_1}), d(\mathbf{v}, \mathbf{v_2}), \ldots, d(\mathbf{v}, \mathbf{v_l})))$. In our application, we make a small change to the mapping F for efficient similarity search. F^q is used to denote the new mapping such that $F^q(\mathbf{v}) = (q(d(\mathbf{v}, \mathbf{v_1})), q(d(\mathbf{v}, \mathbf{v_2})), \ldots, q(d(\mathbf{v}, \mathbf{v_l})))$, where $q(.)$ is a quantization function, which maps the distance d to an integer. Equation (4) shows that the distance $d(\mathbf{v}, \mathbf{v_j}) \in [0, d_{\max}]$. We divide the distance range of $[0, d_{\max}]$ into n bins, so a quantization function $q(.)$ can be defined as follows:

$$q_n(\mathbf{v}, \mathbf{v_j}) = \left\lfloor \frac{n * d(\mathbf{v}, \mathbf{v_j})}{d_{\max}} \right\rfloor \quad (7)$$

where function $\lfloor . \rfloor$ is the floor operation, and d is the L_p metric. In the quantization function $q_n(\mathbf{v}, \mathbf{v_j})$, the subscript n indicates the number of quantization bins, $\mathbf{v}$ indicates some input vector, and $\mathbf{v_j}$ $(1 \leq j \leq l)$ indicates some fixed point (vector).

Obviously, queries performed in the embedding space do not have the same accuracy as queries performed in the original metric space, because the distances measured with the distance function d' in the embedding space can hardly correspond exactly to the distances measured with the original distance function d. Hjaltason and Samet [45] discuss the contractive property of the embedding method, which is a very useful property in similarity search. An embedding based on the mapping F being contractive is sufficient to guarantee 100% recall of queries in the embedding space. Of course, in order to get a tradeoff between accuracy and efficiency, it is possible at a small price in recall.

Now, we discuss the contractive property of the proposed FE. Given arbitrary two rank-ordering feature vector $\mathbf{u}$ and $\mathbf{v}$, the distance $d'(F^q(\mathbf{u}), F^q(\mathbf{v}))$ in the embedding space based on the mapping F^q satisfies

$$d'(F^q(\mathbf{u}), F^q(\mathbf{v})) \leq \frac{n * l^{1/p}}{d_{\max}} d(\mathbf{u}, \mathbf{v}) \quad (8)$$

where d' is an arbitrary L_p metric distance. We can define a contractive factor $c_{L_p} = \frac{n * l^{1/p}}{d_{\max}}$, and use it to demonstrate the contractive property of the proposed FE as follows:

Proof:

$$d'(F^q(\mathbf{u}), F^q(\mathbf{v})) = \left(\sum_{j=1}^{l} |q_n(\mathbf{u}, \mathbf{v_j}) - q_n(\mathbf{v}, \mathbf{v_j})|^p\right)^{1/p}$$

$$= \left(\sum_{j=1}^{l} |\lfloor \frac{n * d(\mathbf{u}, \mathbf{v_j})}{d_{\max}} \rfloor - \lfloor \frac{n * d(\mathbf{v}, \mathbf{v_j})}{d_{\max}} \rfloor|^p\right)^{1/p}$$

$$\approx \left(\sum_{j=1}^{l} |\frac{n * d(\mathbf{u}, \mathbf{v_j})}{d_{\max}} - \frac{n * d(\mathbf{v}, \mathbf{v_j})}{d_{\max}}|^p\right)^{1/p}$$

$$= \frac{n}{d_{\max}} \left(\sum_{j=1}^{l} |d(\mathbf{u}, \mathbf{v_j}) - d(\mathbf{v}, \mathbf{v_j})|^p\right)^{1/p}$$

$$\leq \frac{n}{d_{\max}} \left(\sum_{j=1}^{l} d(\mathbf{u}, \mathbf{v})^p\right)^{1/p} = \frac{n * l^{1/p}}{d_{\max}} d(\mathbf{u}, \mathbf{v}).$$

■

By means of the new mapping $F^q(\mathbf{v}) = (q(d(\mathbf{v}, \mathbf{v_1})), q(d(\mathbf{v}, \mathbf{v_2})), \ldots, q(d(\mathbf{v}, \mathbf{v_l})))$, for each feature vector $\mathbf{v}$ in dataset, we can obtain a l-tuple of values, each coordinate of which varies from 1 to n, so there are n^l different combinations. We encode n^l different combinations to n^l index ID, then construct an inverted file (the structure as described in Fig. 6) to store all GOS descriptors in the dataset. As shown in Fig. 6, the indexing structure is composed of n^l lists of descriptor entries. Each entry indexes the GOS descriptors which are of same quantization index ID. According to the definition about GOS descriptor in Section III, each GOS descriptor includes the original GOS feature vector, the dominant orientation and the scale. The dominant orientation and the scale can be used to assist feature matching [19], [17] and thus be stored in the inverted file. So the information of the GOS descriptor in each indexing entry contains: 1) the interest point ID (vid); 2) the original GOS feature vector GOS; 3) the quantized dominant orientation qo; and 4) the quantized scale qs. When a similarity search is performed for a query descriptor, we first obtain its index ID, then a filter and refine strategy is adopted to search. Specifically, only the descriptors assigned to the same index ID as the query descriptor are checked further, all the retrieved-target descriptors which have different index ID from the query descriptor will be filtered out, then the actual distance, as measured by their original GOS feature vectors, is used to refine the final result.

B. Parameter Selection

Based on the description in Section IV-A, we know that two important parameters will influence the performance of FE. One parameter is the number of fixed points, l. The other parameter is the number of quantization bins, n. On one hand, according to the indexing scheme of FE, the number of index entries is equal to n^l, so the number of index entries in the inverted index file increases with increasing values of l and n. The number of index entries is larger, the number of indexed points in each index ID is less, which, when a "filter" step is carried out, makes the number of candidates less and improves

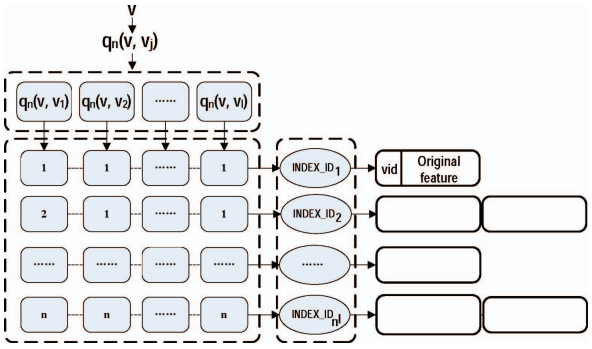


Fig. 6. Inverted file indexing structure of FE.

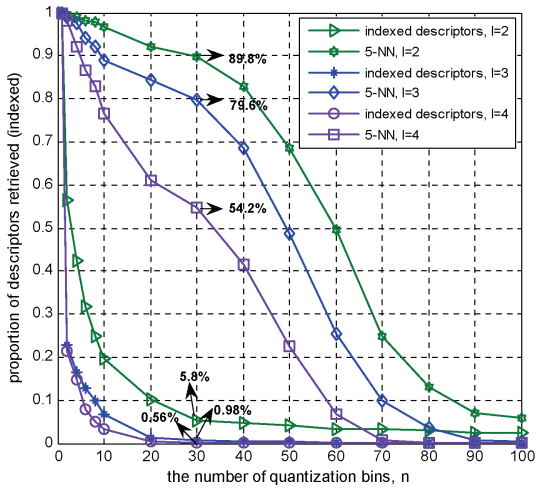


Fig. 7. Influence of parameters (l, n) on performance.

the matching speed. On the other hand, we consider the contractive factor $c_{L_p} = \frac{n * l^{1/p}}{d_{\max}}$ in (8), c_{L_p} increases with increasing values of l and n, which may lead to an noncontractive embedding and decrease the accuracy of similarity search. We can adjust the parameter l, n to obtain the tradeoff between the accuracy and efficiency. Fig. 7 demonstrates the influence of parameters on the performance. The plots have been generated by analyzing a set of 585 158 672 GOS descriptors (other two experiments in this section also used these data). Given a query descriptor x, we can compute its index ID, compare the rate of descriptors that are indexed by this index ID to the rate of 5-NN that are retrieved. Fig. 7 shows that FE can filter out about 99.1% of the descriptors while preserving 79.8% of the 5-NN when $l = 3$ and $n = 30$. In fact, combining Theorem 2 and Characteristic 2 of GOS, we can estimate a suitable value of l. By means of Theorem 2, we know an embedding with distortion $O(log^{1+\theta}N)$ in optimal dimension $O(\theta^{-1}logN/loglogN)$, for any $\theta > 0$. Then according to Characteristic 2 of GOS, we know that the number of different vectors is at most $N = t!$ on a t-D GOS feature vector dataset. In the experiment, the dimensionality of GOS feature vector is 16, we consider the extreme case with $N = 16! \approx 2^{44}$, so the number of fixed points is about $O(\theta^{-1}log2^{44}/loglog2^{44}) \approx O(\theta^{-1}8)$. When θ takes $2 \sim 3$, it reaches the experiment results shown in Fig. 7. Here, we consider the worst case, in real situations, N usually is much smaller than 16!, so the results will be better than the worst case.

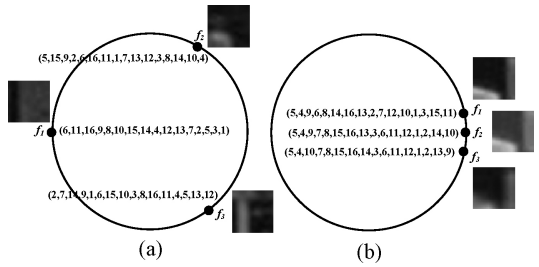


Fig. 8. Example of two different groups of fixed points (three 16-D feature points in each group). (a) Group of dispersive fixed points. (b) Group of concentrated fixed points.

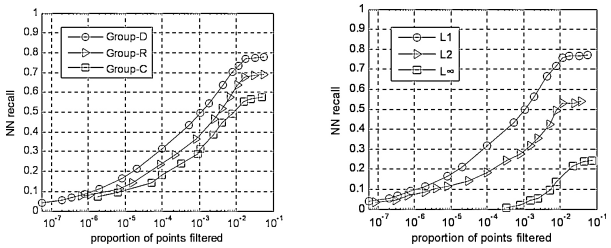


Fig. 9. Comparisons (a) on three ways of fixed points selection for FE and (b) on three distance metrics (L_1, L_2, L_∞) for FE.

After the number of quantization bins (n) and the number of fixed points (l) are determined, another important step of FE is how to select the fixed points. In [43], some points randomly selected are used to construct a reference set R. In our experiment, we evaluate three groups of different fixed points, each group includes three fixed points. The first group of fixed points consist of three dispersive fixed points [such as $\{f_1, f_2, f_3\}$ in Fig. 8(a)]. The distance between them is larger than the second group of concentrated fixed points [such as $\{f_1, f_2, f_3\}$ in Fig. 8(b)]. The third group of fixed points are selected randomly. These three groups of different fixed points are denoted with Group-D, Group-C, and Group-R, respectively. To better illustrate the fixed point selection, we also show in Fig. 8 the reference images of the fixed points (i.e., the 16×16 image patch around the fixed point). It can be observed from Fig. 8, for the three dispersive fixed points, the images are much different, and for the concentrated fixed points, the images are similar.

The experiment result in Fig. 9(a) shows that a group of dispersive fixed points is of better performance than a group of concentrated fixed points. It is seemingly reasonable because l fixed points will degrade into only *one* fixed point (namely, l=1) when the distance between them is small enough (highly concentrated).

In addition, in (7), d is the L_p metric, we compare with the indexing performance of three most common metrics (L_1, L_2, and L_∞). The experimental results show that L_1 metric is of better performance [see Fig. 9(b)]. In the experiment, t=16 and l is fixed to 3, then the plots are generated by changing the value of n. The experimental results can be explained by the contractive property of FE. According to (4), $d_{L_{p\max}} = (\sum_{k=1}^{l} |(t-k+1)-k|^p)^{1/p}$. When t=16, $d_{L_{1\max}} = \sum_{k=1}^{16} |17-2k| = 128$, $d_{L_{2\max}} = (\sum_{k=1}^{16} |17-2k|^2)^{1/2} = \sqrt{1360} \approx 36$, and $d_{L_{\infty\max}} = \max_{1\le k\le 16}|17-2k| = 15$. Then consider the contractive factor in (8) for L_1, L_2, and L_∞ metric: $c_{L_1} = \frac{n*3}{128} \approx \frac{n}{43}$, $c_{L_2} = \frac{n*3^{1/2}}{36} \approx \frac{n}{21}$, and $c_{L_\infty} = \frac{n*3^{1/\infty}}{15} = \frac{n}{15}$. So there exists $c_{L_1} < c_{L_2} < c_{L_\infty}$. Especially, when n=40, the proposed FE based on L_1 metric is contractive and other two cases are noncontractive. So in the next experiments, the parameter combination of FE is $n = 40$ and $l = 3$.

V. Experiments

In this section, we present experiments to show the effectiveness of the proposed methods. First, we evaluate GOS on a public image dataset [55]. Second, we evaluate GOS on Muscle-VCD-2007 video dataset [56], which is the evaluation set used in the video copy evaluation in CIVR 2007. Third, we compare FE with LSH [57]. Finally, we evaluate the overall performance of combination of GOS and FE on TRECVID 2008 video dataset [58].

TABLE II
Computation Time

	GOS	PCA-SIFT	SURF	SIFT
Time (ms)	325	532	655	1976

A. Evaluation of GOS on the Image Dataset

We evaluate the proposed GOS using the experimental methodology, public image dataset, and software of Mikolaczyk *et al.* [55]. The dataset consists of images of eight different scenes, where each scene is imaged six times, with one reference image $Img1$ and five images $Img2, \ldots, Img6$ acquired over successive increments of a particular image deformation, including blur, change of light, zoom, zoom + rotation, JPEG compression noise, and change of viewpoint.

In the experiment, we evaluate four local descriptors: GOS, SIFT, PCA-SIFT, and SURF. The GOS extraction has been described in Section III, and the dimensionality of GOS feature vector is 16-D. From Fig. 10, we can observe that GOS can obtain performance close to the compared descriptors. At the same time, Fig. 10(f) demonstrates the mirror reflection invariance characteristic of GOS that the compared descriptors do not have. In Fig. 10(f), left two images are the original images. Right two images are matching results using GOS, and the lines (green) denotes some right matching between two interest points. Moreover, since the dimensionality of GOS is much lower than the standard SIFT feature, it is fast to compute and match by using GOS (see Table II). Specifically, the computation time in the table is the average processing time (include descriptor extraction and feature matching), tested on the *Graffi* sequence. As mentioned in Section III, the goal of GOS is not to pursue the best description ability compared with other descriptors, but to improve the system response time with a relatively high near-duplicate detection accuracy. The final objective is to apply the proposed GOS to meet the requirement of large scale near-duplicate image and video detection in real time.

It is to be noted that in this paper the experiments are carried out on 4CPU+ 3GHz + DDR 4G computers (Windows).

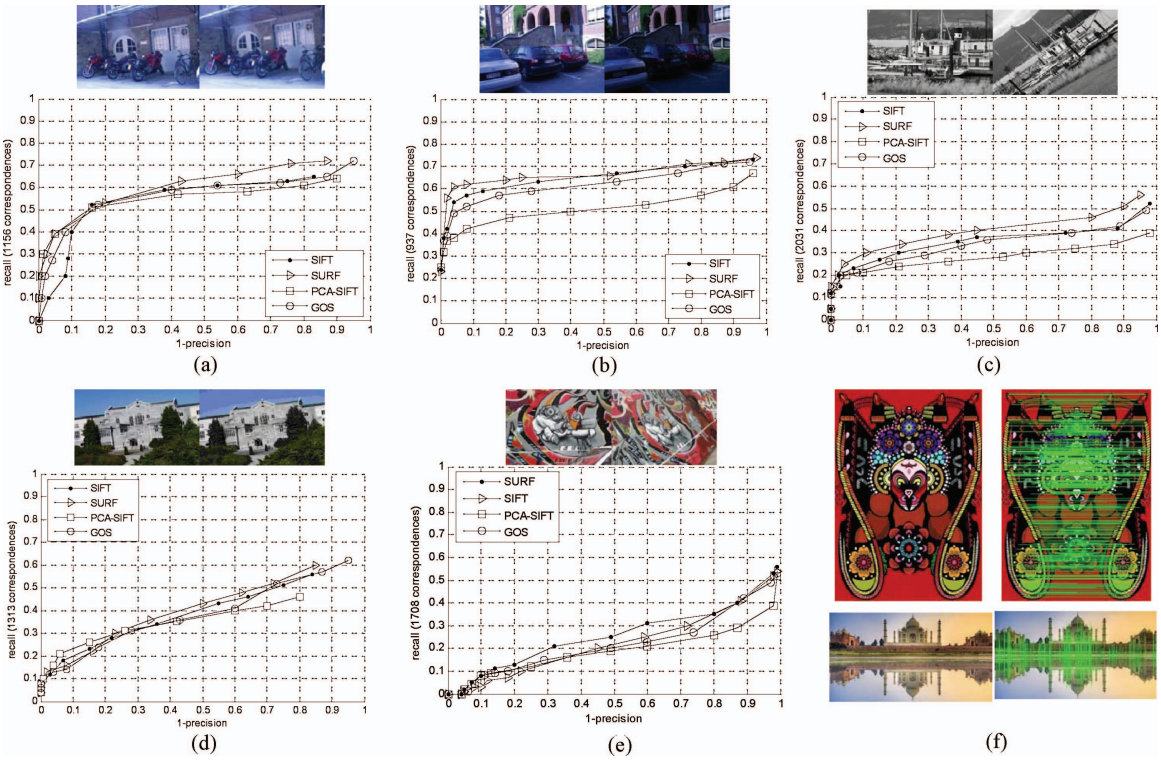


Fig. 10. Recall-(1-Precision) curves. (a) Image blur (*Bikes*). (b) Brightness change (*Leuven*). (c) Zoom + Scale (*Boat*). (d) JPEG compression (*Ubc*). (e) Viewpoint change of 40 degrees (*Graffi*). (f) Demonstration of horizontal and vertical mirror reflection invariance of GOS.

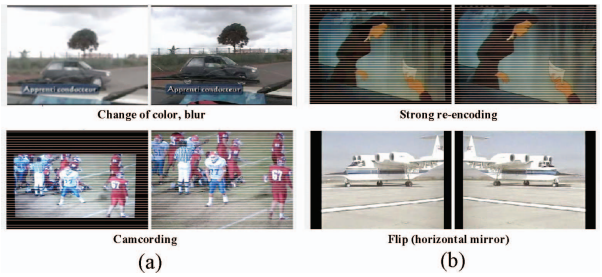


Fig. 11. Four query and reference examples in Muscle-VCD-2007 video dataset. (a) Queries. (b) References.

TABLE III
RESULTS ON THE MUSCLE VCD BENCHMARK DATASET

	ST1 Score	ST2 Score	Runtime (min)
SIFT	0.80	0.76	168
PCA-SIFT	0.66	0.48	65
SURF	0.81	0.75	76
GOS	0.83	0.79	34

B. Evaluation of GOS on the Video Dataset

The final purpose of the proposed GOS is for efficient near-duplicate video detection, so we also evaluated GOS on the Muscle-VCD-2007 video dataset [56], which provides ground truth data for evaluating a system's detection accuracy based on two tasks: finding copies (ST1) and finding extracts (ST2). Given a query video, ST1 task retrieves copies of whole long videos from the database. ST2 task, a much harder task, detects and locates the partial-duplicate segments from the database. Both tasks are challenging because the transformations applied to this benchmark are very diverse. This dataset consists of about 100 h of videos coming from different sources: web video clips, TV archives, and movies. The videos cover a wide range of programs: documentaries, movies, sports events, TV shows, cartoons, etc. Also, the videos have different bit-rates, different resolutions, and different video formats. Some example frames are shown in Fig. 11.

Video copy detection is different from the image copy detection, which also includes key frame extraction, use of time information, feature indexing, and so on. In this evaluation, we only compare the description performance of GOS with other state-of-the-art descriptors. Therefore, besides the descriptors, other same video copy detection methods are used for all compared descriptors. In this experiment, LSH is used for indexing and the temporal grouping method [17] is applied for matching video sequences. Experiment evaluates the detection performance of GOS, SIFT, PCA-SIFT, and SURF. The experimental results are illustrated in Table III.

From the results we can observe that GOS performs better than other compared state-of-the-art descriptors for video copy detection.

C. Evaluation of FE

In this section, we evaluate the performance of our proposed FE. We compare FE with LSH [57] on a dataset of 6 848 870 interest points (from 10 000 images). The interest points are described by GOS. For LSH implementation, we use the E^2LSH package (available at http://www.mit.edu/~andoni/LSH) and apply the approach suggested in the E^2LSH's user manual to find the nearest

TABLE IV
MEMORY CONSUMPTION OF FE VERSUS LSH

	Memory Usage	Parameters	# Keypoints
LSH	2.2G	k=2, m=4, L=6	6 848 870
FE	0.8G	l=3, n=40	6 848 870

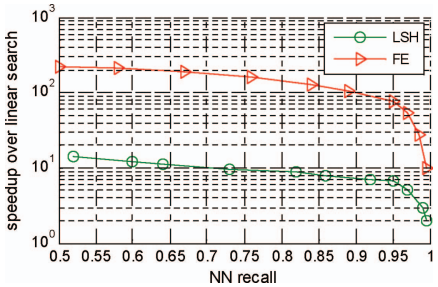


Fig. 12. Search efficiency. Comparison of FE and LSH.

neighbors. Also, the optimal parameters are obtained on the dataset and used to implement LSH. The parameters of LSH and FE are demonstrated in Table IV.

Table IV shows the memory consumption of FE and LSH. LSH required $O(size \times L)$ memory ($size$ is the number of keypoints, L is the number of LSH functions). Also, the value of k is chosen as a function of the dataset to minimize the running time of a query. Since L increases with k, the memory requirement is large for large-scale dataset or moderate-scale dataset, where the optimal time is achieved with higher values of k. Therefore, an upper bound on memory imposes an upper bound on k. After optimization, the memory requirement per keypoint is 12 Bytes. FE requires about $O(size + n)$ memory (n is the number of FE's bins), the memory requirement per keypoint is 16 bits(binID)+16 bits(vid)=4 Bytes.

For the search speed, we compare the speedup of FE and LSH under different NN-recall (the precision achieves as the percentage of the query points for which the nearest neighbors are correctly found). Fig. 12 shows the better performance of the FE compared to the LSH algorithm. For a recall of 0.90, the FE outperforms LSH algorithm by about ten times. Meanwhile, in real application, we need to adjust the parameters in the process of establishing LSH index structure. Also, with the data set changed, we need to rebuild the index structure, which will bring not only high time cost but also technical issues for real application. FE does not have this problem when the data dataset is changed, because the number of the quantizers and bins have been fixed before the index structure is built. Thus, the newly added data will not affect FE's parameters. When a new data point is added, we only need to compute the corresponding index ID, and then add the new data point into this index ID. At the same time, as discussed in Section IV-B, if we consider the worst case, FE is not sensitive to dataset change.

D. GOS and FE for Video Copy Detection

In this section, we demonstrate the overall performance of combination of GOS and FE to detect video copies. We perform experiments on the TRECVID 2008 evaluation video dataset for video copy detection task [5], [58]. This evaluation set includes 438 videos, which last about 200 h. Query videos are generated by implementing the transformations specified in TRECVID 2008, i.e., ten transformations (T1–T10). These ten transformations are described by [59] in detail.

TABLE V
PARAMETERS OF FIVE DIFFERENT METHODS

	# of Frames	# of Descriptors	Descriptor	Parameters
BOF	1 368 000	585 158 672	$SIFT_{128\text{-}D}$	k=20 000
HE	1 368 000	585 158 672	$SIFT_{128\text{-}D}$	k=200 000, d_b=64
HE+WGC	1 368 000	585 158 672	$SIFT_{128\text{-}D}$	k=200 000, d_b=64
FE	1 368 000	585 158 672	$GOS_{16\text{-}D}$	l=3, n=40
FE+WGC	1 368 000	585 158 672	$GOS_{16\text{-}D}$	l=3, n=40

TABLE VI
$F1$ MEASURE OF VIDEO COPY DETECTION

	BOF	HE	HE+WGC	FE	FE+WGC
T1	0.723	0.908	0.928	0.922	0.935
T2	0.695	0.911	0.925	0.901	0.925
T3	0.718	0.920	0.938	0.936	0.947
T4	0.706	0.922	0.935	0.929	0.949
T5	0.672	0.918	0.937	0.922	0.951
T6	0.698	0.922	0.932	0.932	0.942
T7	0.713	0.915	0.936	0.924	0.944
T8	0.654	0.899	0.908	0.922	0.936
T9	0.644	0.901	0.917	0.924	0.935
T10	0.689	0.911	0.923	0.915	0.923

TABLE VII
DETECTION TIME PER QUERY VIDEO (IN SECONDS)

	Compute Descriptors	Preprocessing for Search	Search
BOF	78.68	34.28	246.53
HE	78.68	59.71	103.42
HE+WGC	78.68	59.71	158.83
FE	12.42	5.26	38.25
FE+WGC	12.42	5.26	59.68

We evaluate BOF [20], HE [19], HE+WGC [19], FE, and FE+WGC in our experiment. As demonstrated in Table V, the parameters of BOF, HE, and HE+WGC use default values that are suggested in [19]. Descriptors of BOF, HE, and HE+WGC are obtained by the Hessian-Affine detector and SIFT descriptor. SIFT descriptors are extracted using the software of [55] with the default parameters. Descriptors of FE and FE+WGC are obtained by the Hessian-Affine detector and the proposed GOS. For evaluation, we use F1 measure (F1 is defined as the harmonic mean of precision and recall) and detection time. The results are depicted in Tables VI and VII.

It can be observed from Table VI that our proposed approach can have better performance in recall and precision. BOF, HE, and HE + WGC obtain the parameter k (k is the number of visual words) by performing k-means clustering on a learning set, which makes it sensitive to dataset change. For T8 and T9 including flip transformation, since GOS descriptor can directly deal with this transformation, so FE has better performance. Furthermore, in detection time, Table VII shows FE is faster than BOF and HE. Also, the filtering performance of these methods have great influence on the detection time. According to the data given in [19], when the NN obtains

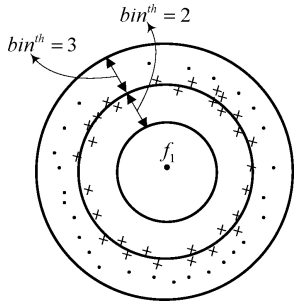


Fig. 13. Hard boundary decision problem of FE.

recall rate of 25%, k=20 000, BOF and HE can filter out about 99.9995% of the descriptors. From data in Fig. 7, FE (l=3, n=40) can obtain same filtering effect in the worst case. In addition, HE and HE + WGC need additional binary signature processing time. GOS also has the advantage of low dimension. Thus the combination of GOS and FE has better overall performance.

Still, we recognize that the proposed FE has its inherent limitations. First, FE is proposed based on some characteristics of GOS, so it is only suitable to the rank-ordering feature. Second, in the FE method, there exists hard boundary between neighboring bins. Since the FE method uses a filter and refine strategy to perform similarity search, only the points which are assigned to the same index ID as the query point are checked further. In fact, for some points near the boundary of the bin, it is hard to determine which bin these points should belong to (as illustrated in Fig. 13). The hard decision of the points can lead to these points been filtered out, and thus will give influence on recall. For a tradeoff consideration, we will assign two bins to these points which lie within the distance range of $[(1-\alpha)d, (1+\alpha)d]$, where d is the distance of the boundary and α is a control parameter (in our application, α takes 0.05). Although, in theory, FE has the hard boundary decision problem, it seldom affects the performance of near-duplicate video detection. The reason being that the number of GOS descriptors extracted from each image (video frame) is normally big; only a tiny part of those points lie near the boundary of the bin. And even if this part of points are filtered out directly, it does not affect the overall results.

VI. Conclusion

This paper presented GOS and FE similarity search method to meet the requirement of large-scale near-duplicate video detection in real time. GOS has the advantages of low dimension and simplicity in computation. Also, GOS posses the inherent characteristics of the standard SIFT, and has mirror reflection invariance. Meanwhile, GOS is a fixed dimension rank-ordering feature. This paper analyzed some characteristics of GOS in detail, and combined with the embedding theory of metric spaces to propose a FE method for efficient similarity search. We evaluated the performance of GOS and FE on different data sets. In description ability, GOS obtained the approximate performance to the state-of-the-art descriptors, but greatly enhanced the speed and cuts memory usage. In similarity search, an major advantage of FE is that its parameters have good controllability. So the performance of FE is stable and not sensitive to dataset change. Compared with LSH, FE improved about ten times speed and saved more than 60% memory usage. Significantly, combination of GOS and FE for near-duplicate video detection can achieve better overall efficiency than the state-of-the-art methods.

References

[1] X. Wu, C.-W. Ngo, A. Hauptmann, and H.-K. Tan, "Real-time near-duplicate elimination for web video search with content and context," *IEEE Trans. Multimedia*, vol. 11, no. 2, pp. 196–207, Feb. 2009.

[2] A. Hampapur and R. Bolle, "Comparison of distance measures for video copy detection," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Aug. 2001, pp. 737–740.

[3] A. Hampapur, K. Hyun, and R. Bolle, "Comparison of sequence matching techniques for video copy detection," in *Proc. SPIE, Storage Retrieval Media Databases*, vol. 4676. Jan. 2002, pp. 194–201.

[4] C. Kim and B. Vasudev, "Spatiotemporal sequence matching for efficient video copy detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 127–132, Jan. 2005.

[5] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid," in *Proc. ACM Int. Workshop Multimedia Inform. Retrieval*, 2006, pp. 321–330.

[6] L. Chen and F. W. M. Stentiford, "Video sequence matching based on temporal ordinal measurement," *Patt. Recog. Lett.*, vol. 29, no. 13, pp. 1824–1831, 2008.

[7] J. Yuan, L.-Y. Duan, Q. Tian, S. Ranganath, and C. Xu, "Fast and robust short video clip search for copy detection," in *Proc. PCM*, 2004, pp. 479–488.

[8] S. C. Cheung and A. Zakhor, "Fast similarity search and clustering of video sequences on the world-wide-web," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 3, pp. 524–537, Jun. 2005.

[9] J. Law-To, B. Olivier, V. Gouet-Brunet, and B. Nozha, "Robust voting algorithm based on labels of behavior for video copy detection," in *Proc. ACM Multimedia*, 2006, pp. 835–844.

[10] A. Joly, O. Buisson, and C. Frelicot, "Content-based copy retrieval using distortion-based probabilistic similarity search," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 293–306, Feb. 2007.

[11] J. S. Satoh, M. Takimoto, and J. Adachi, "Scene duplicate detection from videos based on trajectories of feature points," in *Proc. ACM Int. Workshop MIR*, 2007, pp. 237–244.

[12] X. Wu, A. G. Hauptmann, and C.-W. Ngo, "Practical elimination of near-duplicates from web video search," in *Proc. ACM Multimedia*, 2007, pp. 218–227.

[13] H. Tan, C. Ngo, R. Hong, and T. Chua, "Scalable detection of partial near-duplicate videos by visual-temporal consistency," in *Proc. ACM Multimedia*, 2009, pp. 145–154.

[14] Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," in *Proc. ACM Multimedia*, 2004, pp. 869–876.

[15] M. Douze, A. Gaidon, H. Jegou, M. Marszalek, and C. Schmid. (2008). *TREC Video Retrieval Evaluation Notebook Papers and Slides: INRIA-LEAR's Video Copy Detection System* [Online]. Available: http://www-nlpir.nist.gov/projects/tvpubs/tv8.papers/inria-lear.pdf

[16] W.-L. Zhao and C.-W. Ngo, "Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection," *IEEE Trans. Image Process.*, vol. 18, no. 2, pp. 412–423, Feb. 2009.

[17] M. Douze, H. Jegou, and C. Schmid, "An image-based approach to video copy detection with spatio-temporal post-filtering," *IEEE Trans. Multimedia*, vol. 12, no. 4, pp. 257–266, Jun. 2010.

[18] H. Liu, H. Lu, and X. Xue, "SVD-SIFT for web near-duplicate image detection," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 1445–1448.

[19] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. ECCV*, 2008, pp. 304–317.

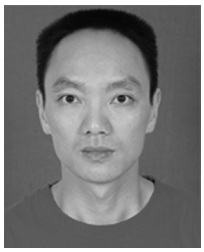
[20] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vision*, Oct. 2003, pp. 1470–1477.

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

[22] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. CVPR*, 2004, pp. 506–513.

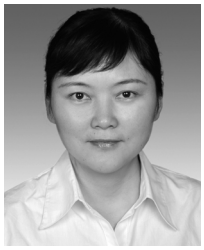
[23] X. Guo, X. Cao, J. Zhang, and X. Li, "MIFT: A mirror reflection invariant feature descriptor," in *Proc. ACCV*, 2009, pp. 536–545.

[24] J. Robinson, "The K-D-B-Tree: A search structure for large multidimensional dynamic indexes," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 1981, pp. 10–18.
[25] N. Katayama and S. Satoh, "The SR-Tree: An index structure for high dimensional nearest neighbor queries," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 1984, pp. 47–57.
[26] A. Guttman, "R-Trees: A dynamic index structure for spatial searching," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 1997, pp. 369–380.
[27] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. VISSAPP*, vol. 1. 2009, pp. 331–340.
[28] R. Weber, H. J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," in *Proc. Int. Conf. Very Large Data Bases*, 1998, pp. 194–205.
[29] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. ACM Symp. Theory Comput.*, 1998, pp. 604–613.
[30] H. Lu, B. C. Ooi, H. T. Shen, and X. Xue, "Hierarchical indexing structure for efficient similarity search in video retrieval," *IEEE Trans. Knowledge Data Eng.*, vol. 18, no. 11, pp. 1544–1559, Nov. 2006.
[31] I. Laptev and T. Lindeberg, "Space-time interest points," in *Proc. Int. Conf. Comput. Vision*, 2003, pp. 432–439.
[32] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.*, 1988, pp. 147–151.
[33] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. ECCV*, 2006, pp. 404–417.
[34] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
[35] M. Toews and W. Wells, "SIFT-Rank: Ordinal description for invariant feature correspondence," in *Proc. CVPR*, 2009, pp. 172–177.
[36] R.Weber and K. Böhm, "Trading quality for time with nearest neighbor search," in *Proc. Conf. Extend. Database Technol.*, 2000, pp. 21–35.
[37] M. E. Houle and J. Sakuma, "Fast approximate similarity search in extremely high-dimensional data sets," in *Proc. Int. Conf. Data Eng.*, 2005, pp. 619–630.
[38] C. Li, E. Chang, M. Garcia-Molina, and G. Wiederhold, "Clustering for approximate similarity search in high-dimensional spaces," *IEEE Trans. Knowledge Data Eng.*, vol. 14, no. 4, pp. 792–808, Jul.–Aug. 2002.
[39] S.-A. Berrani, L. Amsaleg, and P. Gros, "Approximate searches: k-neighbors + precision," in *Proc. Conf. Inform. Knowledge Manag.*, 2003, pp. 24–31.
[40] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A. Abbadi, "Approximate nearest neighbor searching in multimedia databases," in *Proc. Int. Conf. Data Eng.*, 2001, pp. 503–511.
[41] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *Proc. Int. Conf. Vis. Inform. Syst.*, 2002, pp. 117–128.
[42] M. L. Miller, M. A. Rodriguez, and I. J. Cox, "Audio fingerprinting: Nearest neighbor search in high dimensional binary spaces," in *Proc. IEEE Workshop Multimedia Signal Process.*, Dec. 2002, pp. 182–185.
[43] N. Linial, E. London, and Y. Rabinovich, "The geometry of graphs and some of its algorithmic applications," *Combinatorica*, vol. 15, no. 2, pp. 215–245, 1995.
[44] G. Hjaltason and H. Samet, "Contractive embedding methods for similarity searching in metric spaces," Dept. Comput. Sci., Univ. Maryland, College Park, Tech. Rep. TR-4102, 2000.
[45] G. Hjaltason and H. Samet, "Properties of embedding methods for similarity searching in metric spaces," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 25, no. 5, pp. 530–549, May 2003.
[46] I. Abraham, Y. Bartal, and O. Neiman, "Advances in metric embedding theory," in *Proc. 38th Annu. ACM Symp. Theory Comput.*, 2006, pp. 271–286.
[47] W. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Israel J. Math.*, vol. 54, no. 2, pp. 129–138, 1986.
[48] J. Bourgain, "On Lipschitz embedding of finite metric spaces in Hilbert space," *Israel J. Math.*, vol. 52, nos. 1–2, pp. 46–52, 1985.
[49] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas, "Fast nearest neighbor search in medical image databases," in *Proc. Int. Conf. Very Large Data Bases*, 1996, pp. 215–226.
[50] T. Seidl and H.-P. Kriegel, "Optimal multi-step *k*-nearest neighbor search," in *Proc. ACM SIGMOD*, 1998, pp. 154–165.
[51] C. Spearman, "The proof and measurement of association between two things," *Am. J. Psychol.*, vol. 15(1), pp. 72–101, Jan. 1904.
[52] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, nos. 1–2, pp. 81–93, Jun. 1938.
[53] D. N. Bhat and S. K. Nayar, "Ordinal measures for image correspondence," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 20, no. 4, pp. 415–423, Apr. 1998.
[54] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *Int. J. Comput. Vision*, vol. 60, no. 1, pp. 63–86, 2004.
[55] K. Mikolajczyk. (2007). *Binaries for Affine Covariant Region Descriptors* [Online]. Available: http://www.robots.ox.ac.uk/∼vgg/research/affine
[56] J. Law-To, A. Joly, and N. Boujemaa. (2007). *Muscle-VCD-2007: A Live Benchmark for Video Copy Detection* [Online]. Available: http://www-rocq.inria.fr/imedia/civr-bench/data.html
[57] A. Andoni, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. IEEE Symp. Foundations Comput. Sci.*, Oct. 2006, pp. 459–468.
[58] NIST. *TREC Video Retrieval Evaluation* [Online]. Available: http://www-nlpir.nist.gov/projects/t01v
[59] *Final CBCD Video Transformations for TRECVID*. (2008) [Online]. Available: http://www-nlpir.nist.gov/projects/tv2008/final.cbcd.video.transformations.pdf



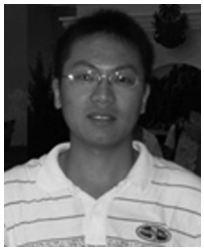
Hong Liu (S'01–M'04) received the B.S. degree from the Department of Physics, Hunan Normal University, Hunan, China, in 1998, and the M.S. degree from the Department of Information Management, Shanghai Branch of Nanjing Political Institute, Shanghai, China, in 2004. Since 2007, he has been pursuing the Ph.D. degree with the School of Computer Science, Fudan University, Shanghai.

From 1998 to 2001, he was a Lecturer and Researcher with the Lianyungang Normal Faculty. Since 2004, he has been an Engineer with the Information Center of Second Military Medical University, Shanghai, where he is currently a Senior Engineer. His current research interests include multimedia information processing and retrieval, pattern recognition, and machine learning.



Hong Lu (S'01–M'04) received the B.Eng. and M.Eng. degrees in computer science and technology from Xidian University, Xi'an, China, in 1993 and 1998, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2005.

From 1993 to 2000, she was a Lecturer and Researcher with the School of Computer Science and Technology, Xidian University. From 2000 to 2003, she was a Research Student with the School of Electrical and Electronic Engineering, Nanyang Technological University. Since 2004, she has been with the School of Computer Science, Fudan University, Shanghai, China, where she is currently an Associate Processor. Her current research interests include image and video processing, computer vision, machine learning, and pattern recognition.



Zhaohui Wen received the B.S. degree from the Software School, Hunan University, Hunan, China, in 2008, and the M.S. degree from the School of Computer Science, Fudan University, Shanghai, China, in 2011.

His current research interests include multimedia information processing and retrieval, pattern recognition, and machine learning.

Xiangyang Xue (M'05) received the B.S., M.S., and Ph.D. degrees in communication engineering from Xidian University, Xi'an, China, in 1989, 1992, and 1995, respectively.

He joined the School of Computer Science, Fudan University, Shanghai, China, in May 1995. Since 2000, he has been a Full Professor. He has published more than 100 research papers in journals or conference proceedings. His current research interests include multimedia information processing and retrieval, pattern recognition, and machine learning.

Dr. Xue is an Associate Editor of the IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT and the *Journal of Computer Research and Development*.