

# The Kernel Trick for Content-Based Media Retrieval in Online Social Networks

Guang-Ho Cha\*

## Abstract

Nowadays, online or mobile social network services (SNS) are very popular and widely spread in our society and daily lives to instantly share, disseminate, and search information. In particular, SNS such as YouTube, Flickr, Facebook, and Amazon allow users to upload billions of images or videos and also provide a number of multimedia information to users. Information retrieval in multimedia-rich SNS is very useful but challenging task. Content-based media retrieval (CBMR) is the process of obtaining the relevant image or video objects for a given query from a collection of information sources. However, CBMR suffers from the dimensionality curse due to inherent high dimensionality features of media data. This paper investigates the effectiveness of the kernel trick in CBMR, specifically, the kernel principal component analysis (KPCA) for dimensionality reduction. KPCA is a nonlinear extension of linear principal component analysis (LPCA) to discovering nonlinear embeddings using the kernel trick. The fundamental idea of KPCA is mapping the input data into a high-dimensional feature space through a nonlinear kernel function and then computing the principal components on that mapped space. This paper investigates the potential of KPCA in CBMR for feature extraction or dimensionality reduction. Using the Gaussian kernel in our experiments, we compute the principal components of an image dataset in the transformed space and then we use them as new feature dimensions for the image dataset. Moreover, KPCA can be applied to other many domains including CBMR, where LPCA has been used to extract features and where the nonlinear extension would be effective. Our results from extensive experiments demonstrate that the potential of KPCA is very encouraging compared with LPCA in CBMR.

## Keywords

Content-Based Retrieval, Dimensionality Curse, Nearest Neighbor Query, Online Social Network, Kernel Method, Kernel Principal Component Analysis, Similarity Search, Social Network Service

## 1. Introduction

Ahmad and Ali [1] categorizes SNS into three categories: (1) services based on social interaction such as Facebook, MySpace, LinkedIn, Twitter, etc.; (2) services based on multimedia such as YouTube, Flickr, etc.; and (3) services based on modern question answering such as Yahoo! Answers, Stack Overflow, Quora, etc. The information based on multimedia including images, audios, and videos is shared by individuals on Instagram, Facebook, Flickr, YouTube, and so on.

The explosion of the amount of digital multimedia in online social networks has brought about the need for the content-based media retrieval (CBMR) services that find images, audios, and videos required by users. This demand in recent years has made CBMR a very active research area. In CBMR image and

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received June 30, 2020; first revision September 24, 2020; accepted December 1, 2020.

Corresponding Author: Guang-Ho Cha ([ghcha@seoultech.ac.kr](mailto:ghcha@seoultech.ac.kr))

\* Dept. of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul, Korea ([ghcha@seoultech.ac.kr](mailto:ghcha@seoultech.ac.kr))

video are usually searched using the visual features such as shape, color, texture, and so on. The visual features are extracted and stored as  $n$ -dimensional feature vectors. During the search, the feature vector of the query image is extracted and compared with feature vectors in the database. The images returned by query processing should be similar to the images given by the query. In this case, if feature vectors have moderate dimensionalities, say below 10, this similarity search problem can be efficiently solved using the conventional multidimensional indexing structures such as the R\*-tree [2], the VP-tree [3,4], and the space filling curve-based indexing method such as the Hilbert-R tree [5] and the HG-tree [6]. But, unfortunately, there have been no efficient solutions so far to solve this problem in high dimensions, for example over 100.

Thus, the major issue in this area is to overcome the problem of dimensionality curse—a phenomenon that the indexing and retrieval performance degrades drastically with dimensionality. In order to overcome the problem caused by the dimensionality curse, the indexing techniques based on approximation have been developed. For example, the vector approximation-based techniques such as the VA-file [7] and the LPC-file [8] and the approximated hashing-based methods such as the locality-sensitive hashing [9,10]. However, in high dimensions, in theory or practice, the indexing methods based on the approximation technique have inherent weakness, for example, their performance is affected by the size of the dataset.

In this paper, we try to solve the problem in view of the reduction of the dimensionality. The approaches to reduce the dimensionality of feature vectors have been attempted by using the techniques such as the principal component analysis (PCA) [11,12] and the singular value decomposition (SVD) [13,14].

PCA is a well-known method to identify patterns from a dataset and to express the dataset with these patterns. In CBMR, PCA is a powerful tool to extract fewer number of dimensions from the original dataset and to represent the dataset on these reduced dimensions. But it is very difficult that this PCA always detects distinguishing patterns in a given dataset. With the use of suitable nonlinear technique, we can extract more outstanding patterns. In this work, we extend the conventional linear PCA (LPCA) to discovering nonlinear embeddings using the kernel method and investigate the effectiveness of the kernel PCA (KPCA) in reducing the dimensionality of dataset in CBMR.

From the next section, we introduce the KPCA that is a nonlinear extension of LPCA. In Section 3, we provide the process of reducing the dimensionality based on KPCA. In Section 4, we discuss the measures to evaluate and analyze the performance of KPCA. Section 5 provides the result of experiments to compare the performances between KPCA and LPCA. Finally, we conclude with discussions of the significance of the work in Section 6.

## 2. Kernel Principal Component Analysis

The kernel trick is a method to extend algorithms into a nonlinearly mapped feature space. The kernel makes an algorithm work in the kernel transformed space. If  $F(x)$  is a transformation of a point  $x$  in the original data space to the feature space then the kernel  $f$  calculates the pairwise scalar product (or similarity) in the feature space of two data points from the original space.

$$f(x, y) = \langle F(x), F(y) \rangle.$$

This paper investigates the efficacy of the kernel trick, specifically KPCA, for feature extraction and dimensionality reduction in CBMR. PCA is a method that conducts an orthogonal basis transformation of the coordinate system where the original dataset is described. The *principal components* are the newly computed orthogonal variables by which the original dataset is represented. The smaller number of principal components than the dimensionality of the data space is usually sufficient to describe the major features in the dataset. We find the principal components of dataset that are nonlinearly related to the structure inherent to the dataset. Among these are feature variables taken by applying arbitrary higher order correlations between sample feature vectors.

The new principal components are drawn by diagonalizing the covariance matrix  $C$  of a given dataset  $\{x_i \in R^N \mid i = 1, \dots, n\}$ , which are centered. It is defined as follows

$$C = \frac{1}{n} \sum_{i=1}^n x_i x_i^t, \quad \sum_{i=1}^n x_i = 0$$

To find the principal components, one has to solve the eigenvector problem.

$$\lambda v = Cv,$$

for eigenvectors  $v \in R^N - \{0\}$  and eigenvalues  $\lambda \geq 0$ . The principal components are the coordinates in the eigenvectors  $v$  and they are represented by the linear combination of variables standardized in the original data space. The size of an eigenvalue  $\lambda$  associated with an eigenvector  $v$  is equal to the size of variance in the direction of  $v$ . Moreover, the directions of the first  $p$  eigenvectors associated with the largest  $p$  eigenvalues show the sum of variances covered by  $p$  orthogonal directions of the dataset. In many real applications, the most interesting information is represented in those directions. For example, in data compression or in data denoising, one projects the original data onto the directions with largest variances to retain as much information as possible and drops deliberately the directions with small variances.

In most cases, it cannot be asserted that LPCA detects most structures in a given dataset. Furthermore, LPCA may be affected seriously by the existence of outliers (or wild data). By applying the suitable *nonlinear* transform, we can draw effective information from the dataset. KPCA is very appropriate to draw informative nonlinear embeddings in a dataset [15].

Therefore, we adopt KPCA for dimensionality reduction in our work and investigate the efficacy of KPCA. In KPCA, all data points are first mapped into a feature space  $P$  using a nonlinear function  $F$  and then all computations are performed on the transformed data. In fact, KPCA adopts *Mercer kernels* instead of directly performing the mapping  $F$  because the mapped feature space  $P$  might be very high dimensional. A Mercer kernel is a function  $f(x, y)$  that constitutes a positive matrix  $G_{ij} = f(x_i, x_j)$  for all datasets  $\{x_i\}$  [15,16]. Using the kernel function  $f$  instead of applying a scalar product in input space corresponds to transforming the data with some transformation function  $F$  to a feature space  $P$ , i.e.,  $f(x, y) = (F(x) \cdot F(y))$ , that allows us to calculate the value of scalar product in  $P$  without having to conduct the mapping  $F$  directly.

After transforming the original data into the high dimensional feature space  $P$  via  $F$ , we perform LPCA, just as performing PCA in the original input space. To perform PCA in the feature space  $P$ , we have to find eigenvalues  $\lambda > 0$  and eigenvectors  $v \in P - \{0\}$  that satisfy  $\lambda v = Cv$  with the covariance matrix  $C$  in  $P$ , defined as follows:

$$C = \frac{1}{n} \sum_{i=1}^n F(x_i) \cdot F(x_i)^t, \quad \sum_{i=1}^n F(x_i) = 0$$

Every solution eigenvector  $v$  must be lain in the span of  $F$ -image of the original data. Thus, we can take into account the following equivalent equation as follows:

$$\lambda(F(x_i) \cdot v) = (F(x_i) \cdot Cv) \quad \text{for all } i = 1, \dots, n \quad (1)$$

There exist coefficients  $c_1, \dots, c_n$  such that

$$v = \sum_{i=1}^n c_i F(x_i) \quad (2)$$

We can get a problem that is given with regards to scalar product when we substitute  $C$  and (2) into (1) and define  $n \times n$  Gram matrix  $G_{ij} = f(F(x_i), F(y_j)) = f(x_i, y_j)$ . Solve the eigenvector problem for non-zero eigenvalues  $\lambda$  and eigenvectors  $c = (c_1, \dots, c_n)^t$

$$n\lambda c = Gc \quad (3)$$

with  $\lambda_p$  being the last non-zero eigenvalue subject to normalization condition  $\lambda_m(c^m \cdot c^m) = 1$  for all  $m = 1, \dots, p$ . We calculate the projection onto the  $m$ -th component by the following equation in order to compute nonlinear principal components for  $F$ -image of the input point  $x$ :

$$v^m \cdot F(x) = \sum_{i=1}^n c_i^m k(x, x_i) = \sum_{i=1}^n c_i^m F(x) \cdot F(x_i) \quad (4)$$

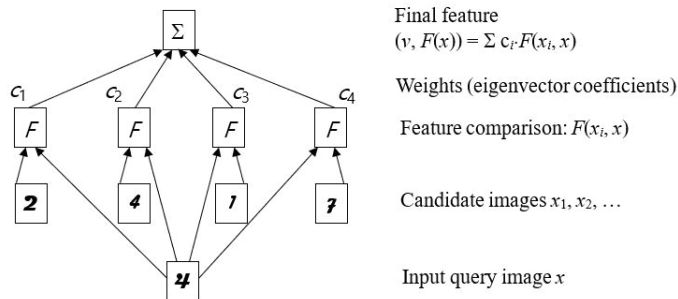
In fact, KPCA corresponds to LPCA in high dimensional feature space. As a result, all statistical and mathematical properties of LPCA are applicable to KPCA, with the modification that they become equations on a set of points  $F(x_i)$ ,  $i = 1, \dots, n$ , in the feature space rather than in the original data space.

Moreover, it can be shown that most forms of nonlinear embeddings appear as large eigenvectors of similarity matrix and are therefore special cases of KPCA. With the following properties, KPCA is actually the orthogonal basis transformation in  $F$  [15,16]: (1) the first  $p$  ( $p \in \{1, \dots, n\}$ ) principal components have more variance than any other orthogonal directions, (2) in representing input vectors, the approximation error caused by the first  $p$  eigenvectors is minimal, (3) the eigenvectors extracted are uncorrelated, and (4) the first  $p$  eigenvectors have maximal information with regard to the input vectors.

### 3. Image Feature Extraction and Dimensionality Reduction

Fig. 1 illustrates the procedural architecture of KPCA for image feature extraction and feature vector comparison that consists of three layers which take different actions. The query image is given at the bottom layer with its feature vector  $x$  computed. The input (query) feature vector  $x$  and the feature vectors (sample patterns)  $x_i$  are nonlinearly mapped via the function  $F$  into the high dimensional feature space  $P$  where scalar products (or similarities) are calculated. Using the kernel function  $F$ , these two layers are actually calculated in one step instead of knowing each feature value. The comparison results are then

combined linearly using the weight  $c_l^i$  computed by solving an eigenvalue problem and the result in the  $l$ -th nonlinear principal component corresponding to  $F$ . Therefore, when we assume that the eigenvectors are sorted in descending order of their corresponding size of eigenvalues, the first  $p$  principal components constitute the  $p$ -dimensional feature vector for an image. By selecting the suitable kernel function  $f$ , various transformation functions  $F$  can be induced indirectly. In this paper, we employ the Gaussian radial basis kernel  $f(x, x') = \exp(-(\|x-x'\|^2/2\sigma^2))$  as our kernel because it is commonly used in pattern recognition and CBMR [17,18].



**Fig. 1.** Image feature extraction and comparison architecture with KPCA. At the bottom layer, the input query feature vector is given. At the upper layer the query input feature vector is compared to sample feature vector using the chosen kernel function (Gaussian radial basis function in our work). Each output is then linearly combined with weight computed by solving the eigenvector problem. The functions of the network can be considered as projections onto the eigenvectors of the similarity matrix in high dimensional feature spaces.

To perform KPCA, we carry out the following steps. We calculate first the Gram (or similarity) matrix  $G_{ij} = f(x_i, y_j)$ . And then, we diagonalize the Gram matrix  $G$  and solve the Eq. (3). We normalize the eigenvector coefficients  $c^n$  such that  $\lambda_n(c^n \cdot c^n) = 1$ . For the  $F$ -image of an input image  $x$ , in order to draw nonlinear principal components that correspond to the kernel function  $f$ , we make projections onto the eigenvectors by Eq. (4).

Unlike LPCA, KPCA permits to extract the amount of principal components that exceeds the dimensionality of input data because it diagonalizes the  $n \times n$  Gram matrix  $G$ ,  $G_{ij} = f(x_i, y_j)$ , instead of the covariance matrix  $C$  of the original data. For example, let us suppose that the number of inputs  $n$  exceeds the input dimensionality  $N$ . Even when LPCA is based on the  $n \times n$  scalar product matrix, it can compute at most  $N$  non-zero eigenvalues and they are identical to the non-zero eigenvalues computed from  $N \times N$  covariance matrix. Contrary to LPCA, KPCA can find up to  $n$  non-zero eigenvalues. It describes that KPCA cannot be directly performed with an  $N \times N$  covariance matrix.

## 4. Performance Analysis

### 4.1 Dimensionality Reduction

In order to provide the insight into how KPCA in feature space  $P$  behaves in input space  $I$ , we provide our experimental results with an image dataset consisting of 13,000 images of US postage stamps with

256 colors as shown in Fig. 2. The images in the dataset are usually not randomly distributed but have very skew distributions in high-dimensional space. Postage stamps usually come in series (for example, flowers, states, persons, birds, etc.) with related designs and similar colors, and the US Postal Service (USPS) has usually used common colors or designs for many long-running postage stamps. Consequently, the image dataset used in our experiments reveals highly clustered and skewed distributions. In other words, the dimensionality of the transformed feature space may be much smaller than that of the input data space, and thus it is appropriate to reduce the dimensionality of the original image dataset. In our work, we retain a sufficient number of principal components so that we can account for at least 80% of the variance in each original variable.

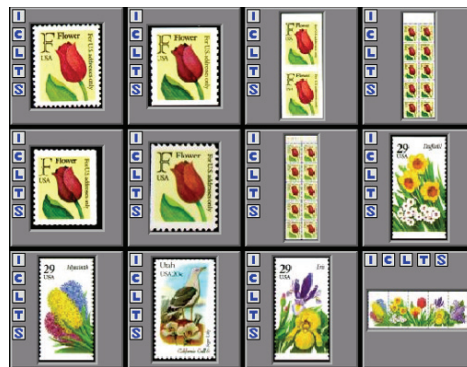


Fig. 2. Image dataset of US postage stamps.

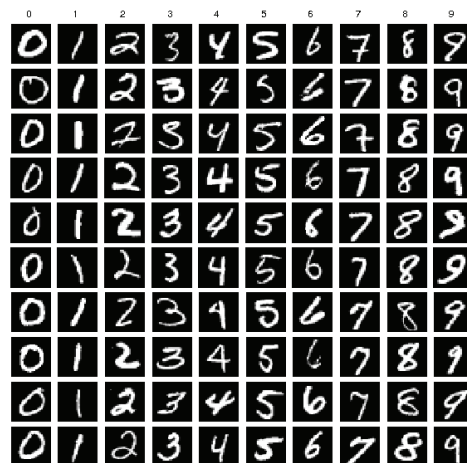


Fig. 3. MNIST dataset of handwritten digit images.

## 4.2 Recognition of Handwritten Digit Characters

In our experiments, with the use of KPCA, we extract nonlinear principal components from a handwritten digit character dataset. We choose the MNIST dataset that contains 120,000 handwritten digit images of  $28 \times 28$  black-and-white pixels as shown in Fig. 3. The Modified National Institute of Standards and Technology (MNIST) database is a large image dataset of handwritten digit characters. It has been widely used for training in many computer vision systems and also commonly used for and

testing and training in machine learning fields. The MNIST dataset is also widely used for classifier benchmark and many techniques have been evaluated using this dataset. The feature of each digit image in the MNIST dataset is represented by a visual feature vector with 784 ( $=28 \times 28$ ) dimensions. For computational reasons, our experiments use only the first 5,000 images from the MNIST dataset. To assess the effectiveness of KPCA, we perform the  $k$ -nearest neighbor ( $k$ -NN) search to find the most similar  $k$  objects for the given query objects after transforming the image dataset based on LPCA and KPCA.

### 4.3 Performance Analysis

In order to evaluate the performance of KPCA, the measures to assess performance should be considered. In traditional information retrieval for documents, to assess its performance, the recall and precision measures are commonly used. Recall is the fraction of the relevant objects that has been retrieved, that is, it evaluates the ability to retrieve useful objects. Precision is the fraction of the retrieved objects that is relevant, that is, it evaluates the ability to reject useless objects.

Consider an example query for explanation. Let  $R$  be the number of relevant objects in a dataset,  $R_r$  be the number of relevant objects that have been retrieved, and  $R_s$  be the number of objects that have been retrieved. Then recall is defined by  $R_r/R$  and precision is defined by  $R_r/R_s$ .

In CBMR, recall and precision also can be used as suitable measures for performance evaluation. Normalized recall and precision have been suggested in IBM QBIC [19] that performs similarity search (or  $k$ -NN search) instead of exact search. These reflect the positions in the retrieved sequence where the relevant objects appear. Assume that a case of ideal retrieval, if there are  $R$  relevant objects in a dataset, then whole  $R$  relevant objects should appear as the first  $R$  retrievals (in any order). In [20], this is defined as the ideal average rank of relevant objects (IAVRR). IAVRR has the highest value if all relevant objects are retrieved and ranked at the top, in this case, IAVRR is computed by  $(0 + 1 + \dots + (R-1))/R$ , where the first (or topmost) position starts at the 0th place. The ratio of average rank of relevant objects (AVRR) to IAVRR ( $AVRR/IAVRR$ ) presents a measure to evaluate the average retrieval accuracy on the number of experimental trials. In the case of ideal retrieval, that is, the perfect performance would generate this ratio to be 1 (i.e.,  $AVRR/IAVRR = 1$ ).

As an example, we assume that the relevant objects for the query object  $Q$  are defined as follows:

$$Q_{46}, Q_{18}, Q_{101}, Q_{52}, Q_{35}, Q_{120}$$

where  $R = 6$ , and the system returns the query result as the following order of

$$Q_{85}, \mathbf{Q_{120}}, Q_{109}, Q_{50}, \mathbf{Q_{18}}, Q_{74}, \mathbf{Q_{46}}, \mathbf{Q_{52}}, Q_{57}, Q_{17}, \mathbf{Q_{35}}, Q_{63}, Q_{16}, Q_{58}, \mathbf{Q_{101}}$$

then the relevant objects appear at positions 1, 4, 6, 7, 10 and 14. The AVRR for this is therefore  $(1 + 4 + 6 + 7 + 10 + 14)/6 = 7$ . The IAVRR is computed by  $(0 + 1 + 2 + 3 + 4 + 5)/6 = 2.5$ . Therefore,  $AVRR/IAVRR = 7/2.5 = 2.8$ .

When the retrieval order is significant, we can use the measure of Kendall's tau to represent the correlation between two datasets [21]. Kendall's tau indicates the disorder in the sequence of retrieved objects. As an example, let us examine two rankings as follows, where both retrievals select the same five objects but position them with different orders:

1 2 3 4 5 (user's choice)  
 2 4 1 3 5 (system's choice)

For explanation, the user's first choice in the first row is ranked second in the second row chosen by the CBMR system. Kendall's tau is computed as follows:

$$(\text{no. of concordant pairs} - \text{no. of not concordant pairs}) / (\text{total no. of pair combinations})$$

In the above example, 2 in the second row (system's choice) is followed by 4, 1, 3, and 5. Here, 2-4 is in order, thus it scores +1, 2-1 is out of order, thus it scores -1, and 2-3, 2-5 are in order, thus they score +1 each. Likely, 4 is followed by 1 and 3. Both are out of order, thus they score -1 each. 4-5 is in order, it scores +1. 1 is followed by 3 and 5. Both are in order, thus they score +1 each. Finally, 3 is followed by 5, it scores +1. The number of in-order pairs is seven, and out-of-order pairs is three, thus the total  $+7 - 3 = +4$ . And the total number of possible pairs,  ${}^N C_2$  is 10 where  $N = 5$ . Thus, the value of tau is  $4/10 = 0.4$ . In other words, Kendall's tau presents the measure of the difference or dissimilarity between two query result rankings. It is in the range  $[-1, +1]$ : the value of -1 denotes the complete disagreement, the value 0 represents there is no correlation between two rankings, and the value +1 means the complete agreement.

In our experimental evaluation, we conduct 100  $k$ -NN queries, average the results, and evaluate the performance based on the above performance measures. In the case of  $k$ -NN queries, recall and precision are the same since  $R = R_r$ . It means that we cannot actually distinguish between the relevant objects and the irrelevant objects in  $k$ -NN queries because the judgment is not based on the exactness but the similarity. Therefore, as a representative of two, we calculate only the precision measure. The ratio AVRR/IAVRR takes into account ranks of relevant objects and presents the measure how much the ranking results are close to top ranks. Kendall's tau presents the measure of order (or disorder) of query results for the  $k$ -NN search.

## 5. Performance Evaluation

With the use of two datasets (a set of 13,000 images with 256 colors of US postage stamps and a set of 6,000 handwritten digit images from MNIST dataset), we conduct extensive experiments for LPCA and KPCA, evaluate their performances base on the measures aforementioned, and compare their performances.

### 5.1 Dimensionality Reduction

In the experiments, we employ 4 MPEG-7 [22] visual feature descriptors to extract feature vectors from the dataset of the US stamps. These visual features are general descriptors widely used in CBMR. They are: (1) 256-dimensional color structure: it is based on color histograms and the localized color distribution, (2) 30-dimensional homogeneous texture: it features the regional texture using the mean and the standard deviation of the input image, (3) 80-dimensional edge histogram: it presents local edge distribution of sub-images in an image, and (4) 35-dimensional region-based shape: it considers all pixels



constituting the shape in the image, that is, both the interior and boundary pixels.

To reduce the dimensionalities of the above four feature vector datasets consisting of four MPEG-7 visual descriptors extracted from the database of the US stamps, we apply KPCA and LPCA to those four datasets, respectively. We conduct  $k$ -NN queries in two categories of datasets, i.e., (1) the datasets where their dimensionalities are reduced by KPCA and (2) the datasets where their dimensionalities are reduced by LPCA. We choose the number  $k$  in  $k$ -NN queries as 20, 40, 60, 80 and 100 in all experiments. In each experiment, we process 100 random  $k$ -NN queries and average their results.

The results of experiments using the four MPEG-7 visual feature datasets are shown in Tables 1–4. In each table, the original dimensionality and the reduced dimensionalities of the datasets are presented in the first column. Additionally, we also present the rate of the variance retained in each original variable in the first column of each table.

As shown in Tables 1–4, with regard to the three performance measures, i.e., Kendall’s tau, AVRR/IAVRR, and precision, the performance of KPCA is superior over that of LPCA. In terms of Kendall’s tau, the performance of KPCA is better than that of LPCA more than 50%. With respect to AVRR/IAVRR and precision, the performance of KPCA is around 10%–20% better than that of LPCA. These results of experiments demonstrate that it is desirable to extend LPCA so as to discover nonlinear embeddings in a dataset and KPCA can be adopted as an efficient nonlinear extension of LPCA in CBMR.

**Table 1.** Experiments on homogeneous texture features (d: dimensionality)

Original, d = 30	LPCA			KPCA		
	Tau	AVRR/IAVRR	Precision	Tau	AVRR/IAVRR	Precision
95%, d = 14	0.41	3.50	0.60	0.69	3.15	0.71
90%, d = 10	0.37	3.75	0.56	0.64	3.30	0.66
85%, d = 9	0.36	3.80	0.51	0.61	3.45	0.62

**Table 2.** Experiments on edge histogram features (d: dimensionality)

Original, d = 80	LPCA			KPCA		
	Tau	AVRR/IAVRR	Precision	Tau	AVRR/IAVRR	Precision
95%, d = 30	0.45	3.48	0.59	0.61	3.27	0.69
90%, d = 25	0.42	3.45	0.54	0.60	3.27	0.64
85%, d = 22	0.39	3.68	0.50	0.58	3.35	0.61

**Table 3.** Experiments on region-based shape features (d: dimensionality)

Original, d = 35	LPCA			KPCA		
	Tau	AVRR/IAVRR	Precision	Tau	AVRR/IAVRR	Precision
95%, d = 15	0.42	3.63	0.57	0.62	3.35	0.68
90%, d = 11	0.40	3.73	0.51	0.60	3.47	0.62
85%, d = 10	0.37	3.85	0.50	0.59	3.45	0.61

**Table 4.** Experiments on color structure features (d: dimensionality)

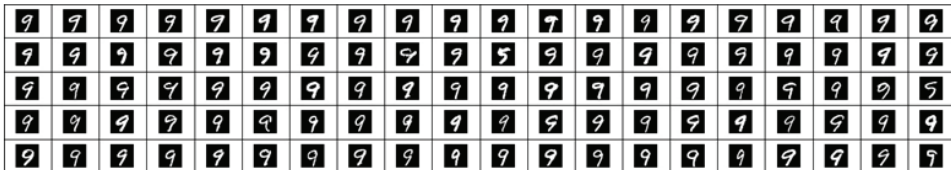
Original, d = 256	LPCA			KPCA		
	Tau	AVRR/IAVRR	Precision	Tau	AVRR/IAVRR	Precision
95%, d = 164	0.42	3.63	0.57	0.62	3.35	0.68
90%, d = 113	0.40	3.73	0.51	0.60	3.47	0.62
85%, d = 77	0.37	3.85	0.50	0.59	3.45	0.61

## 5.2 Recognition of Handwritten Digit Characters

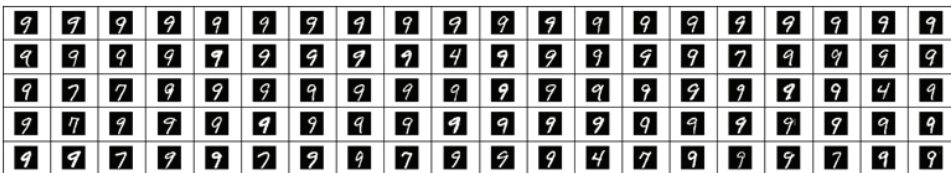
To estimate objectively the performance of KPCA, we regard the concept of query as the category of image sets in which the digit character image is included, that is, one of the labels from “0” to “9” are assigned to each category of digit images.

We also assess the performance of precision of  $k$ -NN queries, where  $k$  is from 10 to 100, and calculate the precision by the ratio of the number of objects returned that are included in the query object category to the returned  $k$  objects.

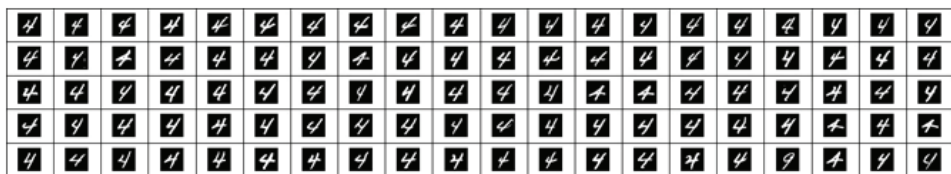
We conduct  $k$ -NN queries 100 times and average the results. From MNIST database, the query objects (images) are selected randomly. We show results of 100-NN queries in Figs. 4–9 to assess the performance of KPCA. The experimental results using single query images were presented in Figs. 4 and 5. The query image is shown in the top-left corner. As shown in Fig. 5, there are actually many digit images not “9” when we used the LPCA algorithm. As shown in Fig. 4 where KPCA is applied, there are only two result images other than the query image (digit “9”). This result of experiments demonstrates the supremacy of KPCA. The results of multi-object queries are shown from Fig. 6 to Fig. 9 where we use two-digit images as query objects. Two query objects are shown in the top-left. The first query uses digit “4” as an input query image. In the second query, two images with digits “0” and “6” are used as query images. For this kind of multi-object queries, we adopt the aggregate similarity metric used in FALCON [23] in which the  $\alpha$  constant is chosen as -3. From Fig. 6 to Fig. 9, it is shown that there are several digit images other than the query objects when we adopt the LPCA algorithm. However, the KPCA algorithm returns the uniform result and the precision is 99%. In this multi-object query, KPCA also demonstrates a better result.



**Fig. 4.** The 100 images returned using the KPCA algorithm (precision = 98%). The query object is in the top-left corner.



**Fig. 5.** The 100 images returned using the LPCA algorithm (precision = 88%).



**Fig. 6.** The 100 images returned using the KPCA algorithm (precision = 99%). The query objects are the two images in the top-left corner.

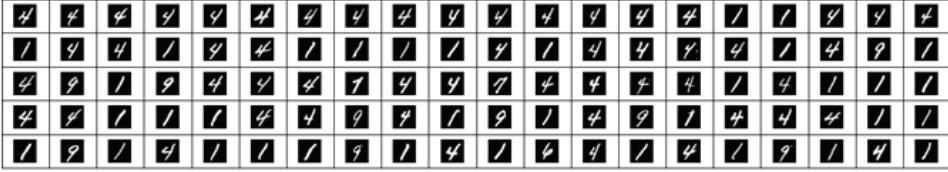


Fig. 7. The 100 images returned using the LPCA algorithm (precision = 54%).

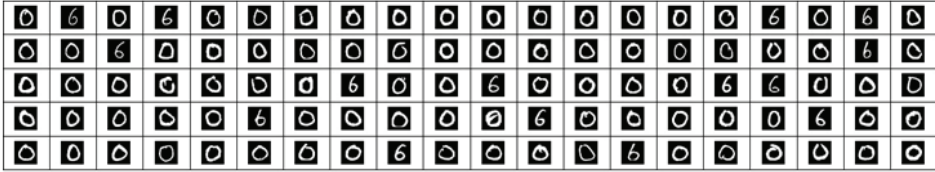


Fig. 8. The 100 images returned using the KPCA algorithm (precision = 100%). The query objects are the two images in the top-left corner.

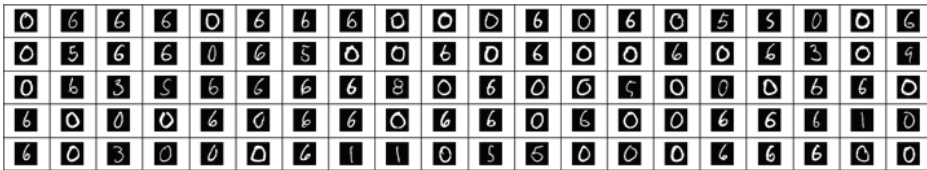


Fig. 9. The 100 objects returned using the LPCA algorithm (precision = 84%).

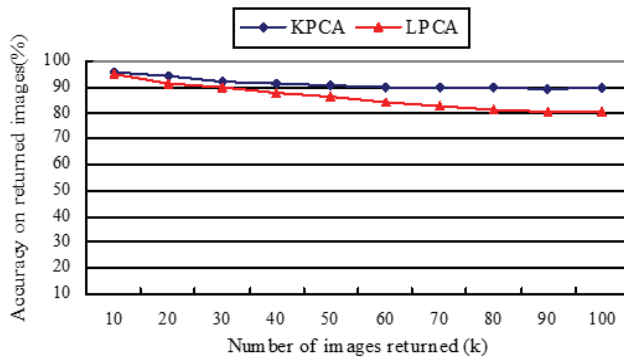


Fig. 10. Precision for single-object queries.

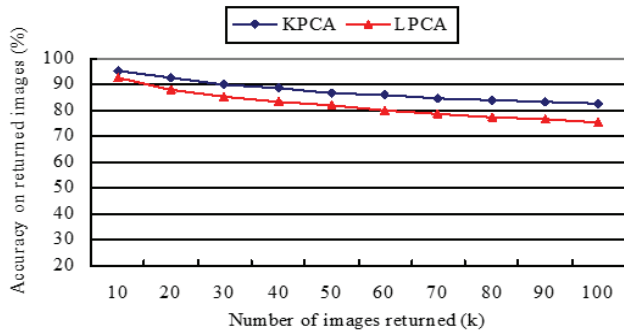


Fig. 11. Precision for multi-object queries.

Figs. 10 and 11 compare the performance of precision between KPCA and LPCA for single-point and multi-point queries, respectively. Fig. 10 shows that KPCA achieves more than the precision of 90% for  $k$ -NN queries, on the other hand, LPCA could not achieve this performance. As shown in Fig. 11 for precisions for multi-object  $k$ -NN queries, KPCA performs more than the precision of 80% in any case, on the other hand, LPCA could not achieve this efficiency.

In order to assess the statistical significance on the performance (i.e., the precision) superiority of KPCA to LPCA, we also conduct the paired t-test for the average precisions for KPCA and LPCA. Some symbols used for the paired t-test are defined as follows, where all  $k$ -NN queries are processed 100 times randomly and  $k = 10, 20, \dots, 100$ :

- $a_i$  and  $b_i$  for  $1 \leq i \leq 10$ : average precisions of KPCA and LPCA, respectively.
- $a'$  and  $b'$ : means of samples  $a_i$  and  $b_i$ , respectively.
- $m_1 (=10)$  and  $m_2 (=10)$ : number of samples come out of average  $k$ -NN query precisions of KPCA and LPCA, respectively.
- $\alpha'^2$ : the population variance estimator and is defined by

$$\alpha'^2 = \frac{1}{m_1 + m_2 - 2} \left( \sum_{i=1}^{m_1} (a_i - a')^2 + \sum_{i=1}^{m_2} (b_i - b')^2 \right)$$

We also make some assumptions as follows:

- The populations of two techniques (that is, average precisions calculated from KPCA and LPCA) follow the normal distribution.
- The variances of the two populations are the same.

We define the hypothesis to examine the average precision difference between KPCA and LPCA as follows:

null hypothesis  $T_0$ :  $\theta_1 = \theta_2$ ,

alternative hypothesis  $T_1$ :  $\theta_1 > \theta_2$

where  $\theta_1$  and  $\theta_2$  are the average precisions of KPCA and LPCA, respectively. If it is significant that the performance improvement made by KPCA, then the alternative hypothesis  $T_1$  is accepted and the null hypothesis  $T_0$  is rejected.

$t$  value is computed as follows:

$$t = \frac{a' - b'}{\alpha' \sqrt{1/m_1 + 1/m_2}}$$

$T_0$  is rejected with the significance level 5%, in other words,  $\theta_1 > \theta_2$  is justified if  $t \geq t_{m_1+m_2-2}(0.10)$  in the  $t$ -distribution table.

$t$  and  $t_{m_1+m_2-2}(0.10)$  are calculated in our experiments as follows:

$$t = 3.116 \text{ for Fig. 10, } t = 2.358 \text{ for Fig. 11, and } t_{m_1+m_2-2}(0.10) = 1.732.$$

Thus, in all cases,  $t \geq t_{m_1+m_2-2}(0.10)$  and it is concluded that the performance of KPCA with respect to precision is better than that of LPCA.

## 6. Conclusion

This paper presents the superiority of KPCA to LPCA for feature extraction and dimensionality reduction in CBMR and the potential of the kernel trick. With the use of Gaussian radial basis kernel, it is demonstrated that KPCA can be used to extract nonlinear embeddings and is successively utilized for dimensionality reduction in high-dimensional feature spaces of large media databases. Compared to LPCA, KPCA demonstrated its superior performance with regard to not only the retrieval precision but also the retrieval quality in CBMR. Thus, it can be concluded that KPCA can be adopted effectively to nonlinearly extend the functionality LPCA.

Compared with other algorithms for nonlinear feature extraction, KPCA has merits that it only requires the solution for the eigenvector problem. Different kernels such as gaussian, sigmoid, and polynomial can lead to fine classification performances [18].

LPCA is actually widely used in many domains such as CBMR (i.e., image indexing and retrieval systems), the natural image analysis, density estimation, and noise reduction. KPCA can be employed to all applications where traditional LPCA has been successfully employed and where a nonlinear extension of feature extraction is needed.

## Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2017R1D1A1B03036561).

## References

- [1] W. Ahmad and R. Ali, "Information retrieval from social networks: a survey," in *Proceedings of 2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, Dhanbad, India, 2016, pp. 631-635.
- [2] N. Beckmann, H. P. Kriegel, R. Schneider, and B. Seeger, "The R\*-tree: an efficient and robust access method for points and rectangles," in *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, Atlantic City, NJ, 1990, pp. 322-331.
- [3] A. W. C. Fu, P. M. S. Chan, Y. L. Cheung, and Y. S. Moon, "Dynamic VP-tree indexing for n-nearest neighbor search given pair-wise distances," *The VLDB Journal*, vol. 9, no. 2, pp. 154-173, 2000.
- [4] S. S. Lee, M. Shishibori, and C. Y. Han, "An improvement video search method for VP-tree by using a trigonometric inequality," *Journal of Information Processing Systems*, vol. 9, no. 2, pp. 315-332, 2013.
- [5] I. Kamel and C. Faloutsos, "Hilbert R-tree: an improved R-tree using fractals," in *Proceedings of 20th International Conference on Very Large Data Bases*, Santiago de Chile, Chile, 1994, pp. 500-509.
- [6] G. H. Cha and C. W. Chung, "A new indexing scheme for content-based image retrieval," *Multimedia Tools and Applications*, vol. 6, no. 3, pp. 263-288, 1998.
- [7] R. Weber, H. J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," in *Proceedings of 24th International Conference on Very Large Data Bases*, New York City, NY, 1998, pp. 194-205.
- [8] G. H. Cha, X. Zhu, P. Petkovic, and C. W. Chung, "An efficient indexing method for nearest neighbor searches in high-dimensional image databases," *IEEE Transactions on Multimedia*, vol. 4, no. 1, pp. 76-87, 2002.

- [9] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Communications of the ACM*, vol. 51, no. 1, pp. 117-122, 2008.
- [10] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proceedings of 2006 47th Annual IEEE Symposium on Foundations of Computer Science*, Berkeley, CA, 2006, pp. 459-468.
- [11] I. Jolliffe, "Principal component analysis," in *Encyclopedia of Statistics in Behavioral Science*. Chichester, UK: John Wiley & Sons, 2005.
- [12] J. Lever, M. Krzywinski, and N. Altman, "Points of significance: principal component analysis," *Nature Methods*, vol. 14, no. 7, pp. 641-643, 2017.
- [13] G. Strang, *Introduction to Linear Algebra*, 5th ed. Wellesley, MA: Wellesley-Cambridge Press, 2016.
- [14] G. Strang and K. Borre, *Linear Algebra, Geodesy, and GPS*. Wellesley, MA: Wellesley-Cambridge Press, 1997.
- [15] N. Pfister, P. Buhlmann, B. Scholkopf, and J. Peters, "Kernel-based tests for joint independence," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 80, no. 1, pp. 5-31, 2018.
- [16] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2018.
- [17] C. J. Simon-Gabriel and B. Scholkopf, "Kernel distribution embeddings: universal kernels, characteristic kernels and kernel metrics on distributions," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 1-29, 2018.
- [18] P. B. Scholkopf, C. Burgest, and V. Vapnik, "Extracting support data for a given task," in *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining (KDD)*, Montreal, Canada, 1995, pp. 252-257.
- [19] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, et al., "Query by image and video content: the QBIC system," *Computer*, vol. 28, no. 9, pp. 23-32, 1995.
- [20] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and effective querying by image content," *Journal of Intelligent Information Systems*, vol. 3, no. 3-4, pp. 231-262, 1994.
- [21] J. Payne, L. Hepplewhite, and T. J. Stonham, "Texture, human perception, and information retrieval measures," in *Proceedings of ACM SIGIR MF/IR Workshop*, Athens, Greece, 2000.
- [22] MPEG-7 [Online]. Available: <https://mpeg.chiariglione.org/standards/mpeg-7>.
- [23] L. Wu, C. Faloutsos, K. Sycara, and T. R. Payne, "FALCON: feedback adaptive loop for content-based retrieval," in *Proceedings of 26th International Conference on Very Large Data Bases*, Cairo, Egypt, 2000, pp. 297-306.



**Guang-Ho Cha** <https://orcid.org/0000-0003-2035-1142>

He received the Ph.D. degree in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1997. From 1999 to 2000, he was a visiting scientist at the IBM Almaden Research Center, San Jose, CA. He is currently a full professor in the Department of Computer Science and Engineering at the Seoul National University of Science and Technology, Seoul, South Korea. His research interests include content-based media indexing and retrieval, data mining, similarity search, and data management on new hardware.