Kernel Feature Analysis for Computer-Aided Detection of Polyps in CT Colonography

Yuichi Motai†, Janne Näppi‡, and Hiroyuki Yoshida§

Abstract—A fast kernel feature analysis is presented for 3-dimensional computer-aided detection of colonic polyps on CT colonographic images. The proposed algorithm, called Accelerated Kernel Feature Analysis (AKFA), extracts salient features from a sample of unclassified patterns by use of a kernel method. Unlike other kernel-based feature selection algorithms, AKFA iteratively constructs a linear subspace of a high-dimensional feature space by maximizing a variance condition for the nonlinearly transformed samples. The resulting linear subspace can then be used for defining efficient data representations and pattern classifiers. Numerical experiments based on a feature space, generated from 292 CT colonographic volume scans including 131 polyps on CT colonographic images, showed that AKFA generates concise feature representations, and it yields similar classification performance to that of Kernel Principal Component Analysis (KPCA) whereas AKFA is computationally much faster than KPCA.

Index Terms—computed tomographic colonography, virtual colonoscopy, polyp detection, kernel feature analysis

I. INTRODUCTION

Computed tomographic (CT) colonography, or virtual colonoscopy, is a promising technique for screening colorectal cancers by use of CT scans of the colon [1]. Current CT technology allows a single image set of the colon to be acquired in 10-20 seconds, which translates into an easier, more comfortable examination than is available with other screening tests. For CT colonography to be a clinically practical means of screening for colon cancers, the technique must be feasible for interpreting a large number of images in a time-effective fashion, and it must facilitate the detection of polyps—a precursor of colorectal cancers—with high accuracy. Currently, however, interpretation of an entire CT colonography examination is time-consuming, and the reader performance for polyp detection varies substantially [2, 3].

To overcome these difficulties while providing a high detection performance of polyps, researchers are developing computer-aided detection (CAD) schemes that automatically detect suspicious lesions in CT colonography images [4]. CAD for CT colonography provides the locations of the suspicious polyps to radiologists, which offers a second opinion that has the potential to improve radiologists’ detection performance.

† Y. Motai is with the School of Engineering, Virginia Commonwealth University, Richmond, VA 23284 USA.
‡ J. Näppi is with the Department of Radiology, Massachusetts General Hospital and Harvard Medical School, Boston, MA 02114 USA.
§ H. Yoshida is with the Department of Radiology, Massachusetts General Hospital and Harvard Medical School, Boston, MA 02114 USA.
Polyps appear as bulbous, caplike structures that adhere to the colonic wall and protrude to the lumen, whereas folds appear as elongated, ridgelike structures, and the colonic wall appears as a large, nearly flat, cuplike structure. Therefore, most CAD schemes employ a model-based approach for the detection of polyp candidates, in which shape analysis methods that differentiate among these distinct types of shapes plays an essential role [5-9]. Nevertheless, there are many naturally occurring normal colonic structures that occasionally imitate such shapes, and therefore the resulting polyp candidates typically include many false positives. The reduction of such false positives is often performed by first extracting a set of image features from segmented polyp regions, followed by application of a statistical classifier to the feature space for discrimination of false positives from actual polyps [8, 10-13]. Such CAD schemes tend to show a high sensitivity in the detection of polyps; however, they tend to suffer from a much large number of false positives than that of human readers [4].

The overall goal of this study is to achieve a high performance in the detection of polyps on CT colonographic images by effectively incorporating an appearance-based object recognition approaches into a model-based CAD scheme. The specific contribution of our studies is to develop a fast kernel feature analysis that, in combination with a shape-based polyp detection method, can efficiently differentiate polyps from false positives and thus improve the detection performance of polyps. The key idea behind the proposed algorithm is to reconstruct a feature space by use of a feature mapping that maps the original, raw feature space into a higher dimensional feature space. Such a high-dimensional feature space is expected to have a greater classification power than that of the original feature space, as suggested by the Vapnik-Chervonenkis theory [14]. We evaluated our fast kernel feature analysis on texture-based features that were extracted from the polyp candidates generated by our shape-based CAD scheme. The main contribution of this paper lies in the appearance-based approach that improves sparse kernel feature analysis for the classification of texture-based features. The method is then tested using real CT colonography data to show that the improved algorithm is faster than the sparse kernel feature analysis while achieving comparable accuracy.

The remainder of this paper is organized as follows. Section II provides a brief overview of our proposed appearance-based recognition scheme. Here, we describe how kernel feature analysis of the texture-based features of polyps complements the model-based polyp detection schemes. Section III provides a brief review of the existing kernel-based feature extraction methods. In Section IV, we present the proposed kernel feature analysis for the detection of polyp candidates. In
Section V, we evaluate the reconstruction and classification performance of the proposed kernel feature analysis algorithm based on the texture-based features of polyps on CT colonographic images. Section VI presents the conclusion.

II. APPEARANCE-BASED RECOGNITION FOR CAD IN CT COLONOGRAPHY

The development of conventional CAD schemes is usually based on model-based recognition, where the design of the detection methods is based on the use of explicit models of the target lesions. For example, the fully automated CAD scheme that was developed in Ref. [5] represents model-based recognition. The CAD scheme extracts the colonic wall based on an anatomical model of the colon, and compares the local shape characteristics of the extracted region with that of a polypoid shape model. The polypoid shape model was based upon two volumetric rotation-invariant shape features, the shape index (SI) and the curvedness (CV). The SI feature characterizes the topological 3-D shape of a local iso-intensity surface in the vicinity of a voxel, and the CV feature characterizes the flatness of that shape. They can be calculated from the principal curvatures $\kappa_1$ and $\kappa_2$ of the local 3-D iso-surface at voxel $p$ as follows:

$$SI(p) \equiv -\frac{2}{\pi} \arctan \left( \frac{\kappa_1(p) + \kappa_2(p)}{\kappa_1(p) - \kappa_2(p)} \right),$$

$$CV(p) \equiv \sqrt{\frac{1}{2} (\kappa_1(p)^2 + \kappa_2(p)^2)}.$$

Because of the high detection sensitivity of the CAD scheme, it also includes a Bayesian neural network model to reduce false-positive (FP) detections, where the input features have been modeled
to identify typical false positives. For example, a 3-D gradient concentration feature is used to identify folds based on the concentration of the gradient vectors in the vicinity of an operating point.

Although model-based CAD schemes can provide excellent detection performance, such schemes have several fundamental limitations. The model-based paradigm requires explicit mathematical modeling of the problem; however, it is often not obvious how to develop such a model with sufficient detail for practical applications in a clinical setting. This makes the development of model-based systems a demanding and time-consuming task that requires a considerable amount of human labor. For example, Fig. 1 shows images of 48 polyps in our database of CT colonographic images (see Section V.A). These images demonstrate that the shape of polyps varies substantially across different polyps; thus the characterization and detection of polyps based only on the shape information is limited. Furthermore, a specific model that is developed for a specific application tends to generalize poorly to other problems. Finally, despite the progress made in CAD methodology during the past years, the CAD schemes that are currently used in a clinical setting tend to generate a large number of FP detections that irritate experienced readers.

Appearance-based recognition is another widely used computer vision methodology, although, currently, it is rarely applied for medical imaging applications. We are developing an appearance-based method that complements the performance of a model-based CAD scheme in the detection of polyps. In this paper, we propose using an appearance-based method for texture-based feature analysis of the polyp candidates, obtained by a shape-based method, for discrimination of false positives from true polyps. Unlike the model-based approaches, the texture-based feature analysis does not specify the solid target model; instead, the texture features are simply handled as instances in the analysis process.
For efficient feature analysis, extraction of the salient features of polyps is essential because of the size and the 3-D nature of the polyp datasets. Moreover, the distribution of the image features of polyps is expected to be non-linear. Therefore, we employ Kernel Principal Component Analysis (Kernel PCA) [15, 16, 19, 20, 21], which is well known as a superior data compression method, and its extension to a non-linear feature space, kernel feature space, in this paper.

Fig. 2 illustrates the concept of feature dissimilarity and of classifier selection by use of the kernel feature space. Here, Fig. 2(a) shows a set of linearly inseparable two-class features in the input space, in which features belonging to a class is labeled by green and those belonging to the other class is labeled by red, and Fig. 2(b) shows a set of linearly separable features using a hyper plane. The features in Fig. 2(a) are transferable to the feature set in Fig. 2(b) by converting the input space into an appropriate higher-dimensional feature space using an operator, so that the features in the two classes can be linearly separable.

The problem is how to select such an ideal operator. A nonlinear, positive-definite kernel $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ of an integral operator, e.g., $k(x, y) = \exp\left(-\|x - y\|^2\right)$, computes the inner product of the transformed vectors $\langle \Phi(x), \Phi(y) \rangle$, where $\Phi : \mathbb{R}^d \rightarrow H$ denotes a nonlinear embedding (induced by $k$) into a possibly infinite dimensional Hilbert space $H$. Given $n$ sample points in the domain $X_n = \{x_i \in \mathbb{R}^d \mid i = 1, \ldots, n\}$, the image $Y_n = \{\Phi(x_i) \mid i = 1, \ldots, n\}$ of $X_n$ spans a linear subspace of at most $(n - 1)$ dimensions. Heuristically, the dominant linear correlations in the distribution of the image $Y_n$ may elucidate important nonlinear dependencies in the original data sample $X_n$. This is advantageous because it permits making PCA non-linear without complicating
the original PCA algorithm. The kernel function \( k \) is traditionally chosen in the form of a Gaussian function such as Radial Basis Functions (RBF): 
\[
k(x, y) = \exp\left\{-\frac{\|x - y\|^2}{2\sigma^2}\right\}. 
\]
As visually demonstrated in Fig. 3, when feature points in the input space are mapped, via the RBF kernel function, to higher-dimensional space corresponding to an inner product in an expanded feature space, features belonging to different classes can be well clustered, and thus the kernel space can be efficient in discriminating one class, e.g., false positives, from the other class, e.g., true polyps.

### III. KERNEL FEATURE EXTRACTION

Our new kernel feature analysis for CT colonographic images is based on the following two methods: Kernel Principal Feature Analysis and Sparse Kernel Feature Analysis. This section provides a brief review of these existing methods. In the next section, we will extend these existing methods and propose a new method called Accelerated Kernel Feature Analysis.

#### A. Kernel Principal Feature Analysis

Kernel Principal Component Analysis (KPCA) uses a Mercer kernel [15] to perform a linear principal component analysis of this transformed image. Without loss of generality, we assume that
the image of the data has been centered so that its scatter matrix in $H$ is given by

$$S = \sum_{i=1}^{n} \Phi(x_i)\Phi(x_i)^T.$$  

Eigenvalues $\lambda_j$ and eigenvectors $e_j$ are obtained by solving

$$\lambda_j e_j = Se_j = \sum_{i=1}^{n} \Phi(x_i)\Phi(x_i)^T e_j = \sum_{i=1}^{n} \left< e_j, \Phi(x_i) \right> \Phi(x_i).$$

for $j=1,\ldots,n$. Since $\Phi$ is not known, (1) must be solved indirectly. Letting $a_{ji} = \frac{1}{\lambda_j} \left< e_j, \Phi(x_i) \right>$ gives

$$e_j = \sum_{i=1}^{n} a_{ji} \Phi(x_i).$$

Multiplying by $\Phi(x_q)^T$ on the left, for $q=1,\ldots,n$, yields

$$\lambda_j \left< \Phi(x_q), e_j \right> = \sum_{i=1}^{n} \left< e_j, \Phi(x_i) \right> \left< \Phi(x_q), \Phi(x_i) \right>.$$

Substitution of (2) into (3) produces

$$\lambda_j \left< \Phi(x_q), \sum_{i=1}^{n} a_{ji} \Phi(x_i) \right> = \sum_{i=1}^{n} \left( \sum_{k=1}^{n} \left< a_{jk} \Phi(x_k), \Phi(x_i) \right> \left< \Phi(x_q), \Phi(x_i) \right> \right).$$

which can be rewritten as $\lambda_j K a_j = K^2 a_j$, where $K$ is an $n\times n$ Gram matrix, with the element $k_{ij} = \left< \Phi(x_i), \Phi(x_j) \right>$ and $a_j = [a_{ji} a_{j2} \ldots a_{jn}]^T$. The latter is a dual eigenvalue problem equivalent to the problem

$$\lambda_j a_j = K a_j.$$  

Since $\|e_j\|^2 = \left< \sum_{i=1}^{n} a_{ji} \Phi(x_i), \sum_{i=1}^{n} a_{ji} \Phi(x_i) \right> = \left< a_j, K a_j \right> = \lambda_j \|a_j\|^2$, the normalization of each eigenvector ($\|e_j\|=1$) requires $\|a_j\|^2 = 1/\lambda_j$.

In the following, we choose a Gaussian kernel, i.e.,

$$k_{ij} = \left< \Phi(x_i), \Phi(x_j) \right> = \exp \left( -\frac{1}{2\sigma^2} \|x_i - x_j\|^2 \right).$$

If the image of $X_n$ is not centered in the Hilbert space, we need to use the centered Gram Matrix deduced by Smola, Mangasarian, and Schölkopf [16]:

$$\hat{K} = K - KT - TK + TKT.$$  

where $K$ is the Gram Matrix of uncentered data, and

$$T = \begin{bmatrix} \frac{1}{n} & \ldots & \frac{1}{n} \\ \frac{1}{n} & \frac{1}{n} & \ldots & \frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \ldots & \frac{1}{n} & \frac{1}{n} \end{bmatrix}_{n\times n}$$
Keeping the \( \ell \) eigenvectors associated with the \( \ell \) largest eigenvalues, we can reconstruct data in the mapped space:

\[
\Phi_i' = \sum_{j=1}^{\ell} \langle \Phi_i, e_j \rangle e_j = \sum_{j=1}^{\ell} \beta_j e_j,
\]

where \( \beta_j = \sum_{i=1}^{n} a_{ji} k_{ii} \). The reconstruction square error of each data point \( \Phi_i, \) \( i = 1, \ldots, n \), is

\[
\text{Err}_i = \| \Phi_i - \Phi_i' \|^2 = k_{ii} - \sum_{j=1}^{\ell} \beta_j^2.
\]

The mean square error is

\[
\text{MErr} = \frac{1}{n} \sum_{i=1}^{n} \text{Err}_i.
\]

Using (5), \( \beta_j = \lambda_j a_{ji} \). Therefore, the mean square reconstruction error is

\[
\text{MErr} = \frac{1}{n} \sum_{i=1}^{n} \left( k_{ii} - \sum_{j=1}^{\ell} \lambda_j^2 a_{ji}^2 \right).
\]

And

\[
\sum_{i=1}^{n} a_{ji}^2 = \| a_j \|^2 = 1 / \lambda_j, \quad \text{MErr} = \frac{1}{n} \sum_{i=1}^{n} \lambda_i.
\]

Kernel PCA can now be summarized as follows:

**Step 1:** Calculate the Gram matrix using (6), which contains the inner products between pairs of image vectors.

**Step 2:** Use (5) to get the coefficient vectors \( a_j, j = 1, \ldots, n \).

**Step 3:** The projection of a test point \( x \in \mathbb{R}^d \) along the \( j \)-th eigenvector is

\[
\langle e_j, \Phi(x) \rangle = \sum_{i=1}^{n} a_{ji} \langle \Phi(x), \Phi(x) \rangle = \sum_{i=1}^{n} a_{ji} k(x, x_i).
\]

The above implicitly contains an eigenvalue problem of rank \( n \), so the computational complexity of Kernel PCA is \( O(n^3) \). In addition, each resulting eigenvector is represented as a linear combination of \( n \) terms. Thus, all data contained in \( X_n \) must be retained, which is computationally cumbersome and unacceptable for incremental or online learning.

**B. Sparse Kernel Feature Analysis**

Sparse Kernel Feature Analysis (SKFA) \[16\] extracts the features one by one in order of decreasing projection variance. SKFA improved the computational costs of KPCA, associated with both time complexity and data retention requirements. In that sense, SKFA is more compact and time efficient method than KPCA. The particular advantage of SKFA is that the \( \ell \) features only depend on \( \ell \) elements of \( X_n \), which is extremely useful for online learning. We let \( v_i \in H, i = 1, \ldots, \ell \) denote the features selected by SKFA. We intentionally avoid using \( e_i \) in order to distinguish these features from the eigenvectors obtained using KPCA. Again, we analyze the scatter matrix of the image data. Following (1), with \( e_j \) replaced by \( v_j \), we obtain

\[
v_j' \beta_j v_j = v_j' \sum_{i=1}^{n} \langle v_j, \Phi(x_i) \rangle \Phi(x_i),
\]

where \( v_j \) is the \( j \)-th feature with unit length.
\[ \lambda_j = \sum_{i=1}^{n} \left( v_j, \Phi(x_i) \right)^2. \]

Therefore, the first feature, corresponding to the maximum eigenvalue, is chosen as the direction with the maximum projected variance:

\[ v_1 = \arg \max_{\|v\|^2} \frac{1}{n} \sum_{i=1}^{n} |\langle v, \Phi(x_i) \rangle|^2. \] (8)

The global solution of (8), \( v_1 \), needs to satisfy an \( l_2 \) normalization constraint, i.e., unit Euclidian length. Changing the \( l_2 \) constraint to an \( l_1 \) constraint leads to a vertex solution. The \( l_1 \) constraint assumed by SKFA is

\[ V_j = \left\{ \frac{1}{n} \sum_{i=1}^{n} a_j \Phi(x_i) \middle| \sum_{j=1}^{n} |a_j| \leq 1 \right\}. \] (9)

The first feature selected by SKFA satisfies

\[ v_1 = \arg \max_{v \in V_j} \frac{1}{n} \sum_{i=1}^{n} |\langle v, \Phi(x_i) \rangle|^2. \]

Smola et al. showed that this feature corresponds to an element of the image \( Y_n \), hence

\[ v_1 = \arg \max_{\Phi(x_j) \in V_j} \frac{1}{n} \sum_{i=1}^{n} \left| \langle \Phi(x_j), \Phi(x_i) \rangle \right|^2. \] (10)

Subsequent features are obtained iteratively. Suppose \( i-1 \) features \( \{v_t | t = 1, \ldots, i-1\} \) have been found; then each image \( \Phi_j = \Phi(x_j) \) is projected into the orthogonal subspace to obtain

\[ \Phi'_j = \Phi_j - \sum_{t=1}^{i-1} v_t \frac{\left( \Phi_j, v_t \right)}{\|v_t\|^2} = \Phi_j - \sum_{t=1}^{i-1} a_{jt} v_t. \] (11)

where \( a_{jt} = \left( \Phi_j, v_t \right) \). Then \( \Phi'_j \) is normalized by the \( l_1 \) constraint in (9). The projection variance of the normalized \( \Phi'_j \) with all \( \Phi_k, k = 1, \ldots, n \) is then calculated. Finally, one identifies the maximum projection variance and selects the corresponding \( \Phi'_j \) as the \( i \)-th feature, \( v_i \).

Based on the \( \ell \) features extracted by SKFA, each training data in the mapped space can be reconstructed by

\[ \Phi'_j = \sum_{j=1}^{\ell} \frac{\left( \Phi'_j, v_j \right)}{\|v_j\|^2} v_j = \sum_{j=1}^{\ell} a_{ji} v_j. \] (12)

where \( a_{ji} \), the projection of \( i \)-th training data on \( j \)-th feature, is stored after extracting the \( j \)-th feature. According to (11), the feature set \( \{v_1, \ldots, v_{\ell}\} \) only depends upon the set of \( \ell \) image vectors, \( \{\Phi_{idx(i)}, i = 1, \ldots, \ell\} \), where \( idx(i) \) denotes the subscript of the projected image \( \Phi'_j \) that was selected
when constructing \( v_i \). Therefore, after training, we only need to retain the \( \ell \) input vectors \( \{ x_{vid(i)} \in \mathbb{R}^d \mid i = 1, \ldots, \ell \} \), where \( \ell \) is the number of features extracted.

In this way, SKFA extracts \( \ell \) features, where one assumes \( \ell << n \). As \( O(in^2) \) operations are required to extract the \( i \)-th feature, the total computational cost for \( \ell \) features is \( O(\ell^2n^2) \), which is an improvement over the \( O(n^3) \) operations required by kernel PCA.

IV. ACCELERATED KERNEL FEATURE ANALYSIS

To further improve the efficiency and accuracy of SKFA, we propose an Accelerated Kernel Feature Analysis (AKFA) that (i) saves computation time by iteratively updating the Gram Matrix, (ii) normalizes the images with the \( l_2 \) constraint before the \( l_1 \) constraint is applied, and (iii) optionally discards data that falls below a threshold magnitude \( \delta \) during updates.

First (i), instead of extracting features directly from the original mapped space, AKFA extracts the \( i \)-th feature based on the \( i \)-th updated Gram matrix \( K^i \), where each element has \( k^i_{jk} = \langle \Phi^i_j, \Phi^i_k \rangle \). Since \( \Phi^i_j = \Phi^{i-1}_j - v^i_{i-1} \langle \Phi^{i-1}_j, v^i_{i-1} \rangle \),

\[
k^i_{jk} = \langle \Phi^{i-1}_j, \Phi^{i-1}_k \rangle - \langle \Phi^{i-1}_j, v^i_{i-1} \rangle \langle \Phi^{i-1}_k, v^i_{i-1} \rangle = k^{i-1}_{jk} - \frac{\langle \Phi^{i-1}_j, \Phi^{i-1}_{vid(i-1)} \rangle \langle \Phi^{i-1}_k, \Phi^{i-1}_{vid(i-1)} \rangle}{\| \Phi^{i-1}_{vid(i-1)} \|^2} = k^{i-1}_{jk} - \frac{k^{i-1}_{jvid(i-1)}k^{i-1}_{vid(i-1)}}{k^{i-1}_{vid(i-1),vid(i-1)}},
\]

(13)

By updating the Gram Matrix, it is unnecessary to save the projection of each individual data on all previous features. The computational cost for extracting \( i \)-th feature becomes \( O(n^2) \), instead of \( O(in^2) \) as in SKFA.

The second (ii) improvement is to revise the \( l_1 \) constraint. As shown in (10), SKFA treats each data point...

![Diagram](image-url)  
**Fig. 4.** The organization of the methodology is illustrated. \( l \) is the number of features extracted for eigen dimension, and \( n \) is the data size.
individual sample data as a possible direction, and computes the projection variances with all data. Since SKFA includes its length in its projection variance calculation, it is biased to select vectors with larger magnitude. From the objective function (8), we know that we are actually looking for a direction with unit length. When we choose an image vector as a possible direction, we ignore the length and only consider its direction, which improves the accuracy of the features. Therefore, in our AKFA algorithm, we replace the $l_1$ constraint (9) by

$$V_{hn}^i = \left\{ \sum_{j=1}^n a_j \frac{\Phi_j^i}{\|\Phi_j^i\|} \mid \sum_{j=1}^n |a_j| \leq 1 \right\}.$$ 

The $i$-th feature is extracted by

$$v_i = \arg \max_{\Phi_j^i \in V_{hn}^i} \frac{1}{n} \sum_{j=1}^n \|v_i \Phi_j^i\|^2.$$ 

Since $v_i$ is extracted from $\Phi^i$ space, the solution is located on one of $\Phi_j^i = \Phi_j^i / \|\Phi_j^i\|$ for $j = 1, \ldots, n$. Equation (10) reduces to

$$v_i = \arg \max_{\Phi_j^i} \frac{1}{n} \sum_{j=1}^n \left( \Phi_j^i, \Phi_j^i \right)^2 = \arg \max_{\Phi_j^i} \frac{1}{n} \sum_{j=1}^n k_{jj}^i.$$ 

(14)

Each $\Phi_j^i$ satisfies the $l_1$ constraint, so the normalization step that appears in SKFA is not required.

Let $\Phi_{idx(i)}^i$ denote the image vector corresponding to the $i$-th feature. Suppose we have selected $(i-1)$ features with $V_{(i-1)} = [v_1, v_2, \ldots, v_{(i-1)}]$, $\Phi_{(i-1)} = [\Phi_{idx(1)}^i, \Phi_{idx(2)}^i, \ldots, \Phi_{idx(i-1)}^i]$, and $C_{(i-1)}$ is the coefficient matrix, which is upper-triangular.

Then $\Phi_{idx(i)}^i = \Phi_{idx(i)} - \sum_{j=1}^{i-1} \left( \Phi_{idx(i)}, v_j \right) v_j$. Let’s study the second term:

$$\sum_{j=1}^{i-1} \left( \Phi_{idx(i)}, v_j \right) v_j = \sum_{j=1}^{i-1} v_j v_j^i \Phi_{idx(i)} = \Phi_{(i-1)} C_{(i-1)} C_{(i-1)}^t K_{idx(i)}.$$ 

(15)

where $K_{idx(i)} = [k_{idx(i), idx(i)}^{11}, k_{idx(i), idx(i)}^{12}, \ldots, k_{idx(i), idx(i)}^{(i-1)(i-1)}]^\top$. Therefore,

$$v_i = \left( \Phi_{idx(i)} - \Phi_{(i-1)} C_{(i-1)} C_{(i-1)}^t K_{idx(i)} \right) / \sqrt{k_{idx(i), idx(i)}^i},$$

Let

$$C_{i,j} = 1 / \sqrt{k_{idx(i), idx(i)}^i},$$

(16)

and

$$C_{(i-1),j} = -C_{i,j} C_{(i-1), (i-1)} C_{(i-1), (i-1)}^t K_{idx(i)}.$$ 

(17)
then $\mathbf{V}_i = \Phi_i \mathbf{C}_i$, where $\mathbf{V}_i = [\mathbf{v}_{(i-1)}, \mathbf{v}_i]$ and $\Phi_i = [\Phi_{(i-1)}, \Phi_{id(i)}]$, $\mathbf{C}_i = \begin{pmatrix} \mathbf{C}_{(i-1)} & \mathbf{C}_{(i-1),i} \\ 0 & \mathbf{C}_{i,i} \end{pmatrix}$.

The third (iii) improvement is to discard negligible data and thereby eliminate unnecessary computations. In the $i$-th updated Gram matrix $K_i^{(i)}$, defined in (13), the diagonal elements $k_{jj}^{(i)}$ represents the reconstruction error of the $j$-th data point with respect to the previous $(i-1)$ features. If $k_{jj}^{(i)}$ is relatively small, then the previous $(i-1)$ features contain most of the information that would be acquired from this image vector. One can therefore optionally discard image points that satisfy $k_{jj}^{(i)} < \delta$, for some predetermined $\delta > 0$. In the following, we use a cut-off threshold that is useful when the data size is very large. It can also be used as a criterion to stop extracting features if one is unsure of the number of features that should be selected.

The entire algorithm of AKFA is summarized below:

Step 1: Compute the $n \times n$ Gram matrix $k_i = k(x_i, x_j)$, where $n$ is the number of input vectors. This part requires $O(n^2)$ operations.

Step 2: Let $\ell$ denote the number of features to be extracted. Initialize the $\ell \times \ell$ coefficient matrix $\mathbf{C}$ to 0, and $\mathbf{idx}(\cdot)$ as an empty list which will ultimately store the indices of the selected image vectors. Initialize the threshold value $\delta = 0$ for the reconstruction error. The overall cost is $O(\ell^2)$.

Step 3: For $i = 1$ to $\ell$ repeat:

1. Using the $i$-th updated $K_i^{(i)}$ matrix, extract the $i$-th feature using (14). If $k_{jj}^{(i)} < \delta$, then discard $j$-th column and $j$-th row vector without calculating the projection variance. Use $\mathbf{idx}(i)$ to store the index. This step requires $O(n^2)$ operations.

2. Update the coefficient matrix by using (16) and (17), which requires $O(i^2)$ operations.

3. Use (13) to obtain $K_i^{(i+1)}$, an updated Gram matrix. Neglect all rows and columns that contain a diagonal element less than $\delta$. This step requires $O(n^2)$ operations.

The total computational complexity is increased to $O(\ell n^2)$ when no data is cut during updating in the proposed AKFA. If we increase $\delta$, more data will be cut, and the total computational time will be decreased. If we chose to maintain good reconstruction accuracy, $\delta$ should be small. However, if we choose a shorter computation time, a large enough $\delta$ must be chosen. Since the
number of data being cut at each step depends on $\delta$ and the data set, we consider an average situation. Suppose after extracting a feature, we only keep a fraction $p$ of all possible directions, and let there be one point left after extracting $\ell$ features. That means $n p^\ell = 1$, where $n$ is the data size. It is equal to $p = n^{-2/\ell}$. The total computational time $T$ is

$$T_{AKFA} = \sum_{k=0}^{\ell-1} (np^k)^2 = \frac{n^2 - 1}{1 - n^{-2/\ell}} = \frac{n^2}{1 - n^{-2/\ell}}. \quad (18)$$

when $n$ is large. The computational time of SKFA is $T_{SKFA} = O(\ell^2 n^2)$. Therefore,

$$\frac{T_{SKFA}}{T_{AKFA}} = \ell^2 (1 - n^{-2/\ell}). \quad (19)$$

For example, when $\ell = 50$, data size $n = 3000$, $T_{SKFA}/T_{AKFA}$ is about 684, so the speed of AKFA is 684 times faster than SKFA. The experimental results confirm that our features have better performance than those obtained by SKFA. Fig. 4 summarizes the comparison for computational complexity.

V. EXPERIMENTAL ANALYSIS

A. CT Colonography Image Database

The proposed algorithm of AKFA was evaluated based on a CT image database of colonic polyps developed as follows. We retrospectively collected 292 CT colonography datasets of 146 patients who had undergone a colon-cleansing regimen in preparation for same-day optical colonoscopy. Helical single-slice and multi-slice CT scanners (GE HiSpeed CTi, LightSpeed QX/I, and LightSpeed Ultra; GE Medical Systems, Milwaukee, WI) were used in supine and prone positions with collimations of 1.25 - 5.0 mm, reconstruction intervals of 1.0 – 5.0 mm, X-ray tube currents of 50 – 260 mA with 120 – 140 kVp, in-plane voxel sizes of 0.51– 0.94 mm, and a CT image matrix size of 512 x 512. There were 108 normal cases and 38 abnormal cases with a total of 61 colonoscopy-confirmed polyps $\geq 6$ mm. Twenty-eight polyps were 6-9 mm and 33 polyps were $\geq 10$ mm (including 7 lesions $\geq 30$ mm) in size. The CAD scheme processed independently the supine and prone volumetric data sets generated from a single patient to yield polyp candidates. Application of the CAD scheme detected polyp candidates in the 292 CT colonography datasets at a 98% by-polyp detection sensitivity. There was a total of 131 true-positive polyp candidates (some of the larger lesions had multiple detections).
The set of volumes of interest (VOIs) representing the resulting polyp candidates was calculated. The center of a VOI was placed at the center of a polyp candidate. The center of the polyp candidate was determined by the calculation of the mass center of the region of the polyp candidate segmented automatically by the CAD scheme. All VOIs had dimensions of $16 \times 16 \times 16$ voxels, and the voxel intensities were used directly as the feature vector. The VOI was resampled to cover the complete region of the automatically segmented polyp candidate. To resample the region, we first determined the largest projected size $s$ of the polyp candidate along the $x$, $y$, and $z$ axes, placed a VOI with dimensions $s \times s \times s$ voxels at the center of the polyp candidate, and then resampled the VOI to $16 \times 16 \times 16$ voxels.

**B. Computational Time with Data Sizes**

The results on the computation time, in seconds, are shown in Table I. All the results were obtained using the Statistical Pattern Recognition Toolbox [17] on Matlab 7.0.1 (R14) for the Gram matrix calculation and KPCA algorithm, running on the Partners Research Computing cluster [18]. The cluster had 26 working nodes using an HP server. Each node had 72 GB storage (head node 380 GB storage), and two 3 GHz AMD Opteron 32/64 CPUs and 4 GB RAM. The nodes communicated via a GigE switch and used an NFS mount.

<table>
<thead>
<tr>
<th>Data Size</th>
<th>KPCA</th>
<th>SKFA</th>
<th>AKFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>5.94</td>
<td>200.49</td>
<td>1.55</td>
</tr>
<tr>
<td>1000</td>
<td>45.06</td>
<td>694.86</td>
<td>5.81</td>
</tr>
<tr>
<td>1500</td>
<td>155.44</td>
<td>1597.41</td>
<td>14.90</td>
</tr>
<tr>
<td>2000</td>
<td>377.51</td>
<td>2875.62</td>
<td>33.54</td>
</tr>
<tr>
<td>2500</td>
<td>785.89</td>
<td>4548.11</td>
<td>67.69</td>
</tr>
<tr>
<td>3000</td>
<td>1413.4</td>
<td>7653.02</td>
<td>121.53</td>
</tr>
<tr>
<td>3500</td>
<td>3166.6</td>
<td>10219.06</td>
<td>335.14</td>
</tr>
<tr>
<td>4000</td>
<td>4812.2</td>
<td>13344.42</td>
<td>402.06</td>
</tr>
<tr>
<td>4500</td>
<td>7008.1</td>
<td>17163.31</td>
<td>552.76</td>
</tr>
<tr>
<td>5000</td>
<td>9557.4</td>
<td>20960.26</td>
<td>725.28</td>
</tr>
</tbody>
</table>
For each of the algorithms, KPCA, SKFA, and AKFA, Table I indicates that the computational time of KPCA increased rapidly with the increase of the data size $n$. We set the eigen-dimension to 70 for measuring computation time. At $n = 3500$, the computation time of KPCA and SKFA were 9.4 and 30.5 times longer than that of AKFA. If the computational time versus data size $n$ is plotted with common-logarithm scales, the results fit the expected curves, and thus validate the complexity analyses in the methodology sections. This clearly shows that our proposed AKFA was much faster than the existing methods of KPCA and SKFA, especially when the data size is large.

### C. Reconstruction Accuracy with respect to Data Sizes

Reconstruction accuracy was evaluated when the data size was increased from 500 to 5000 for each of the algorithms, KPCA, SKFA, and AKFA. Here, reconstruction was calculated as $1 - \text{Err}_r = 1 - \| \Phi_r - \Phi \|^2$ (see the notations in Sections III and IV). We set the eigen-dimension to 70 for evaluating the reconstruction error. Table II shows that KPCA, which computes most accurate estimates of the eigenvectors, incurred a smaller reconstruction error than that of the other approximate methods. In this sense, the features generated by KPCA were more accurate in the reconstructed quality of features, but the difference among the others was not large, at or below 6%.

### Table II

<table>
<thead>
<tr>
<th>Data Size</th>
<th>KPCA</th>
<th>SKFA</th>
<th>AKFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.8804</td>
<td>0.8367</td>
<td>0.8271</td>
</tr>
<tr>
<td>1000</td>
<td>0.9411</td>
<td>0.9001</td>
<td>0.8895</td>
</tr>
<tr>
<td>1500</td>
<td>0.9655</td>
<td>0.9206</td>
<td>0.9115</td>
</tr>
<tr>
<td>2000</td>
<td>0.9720</td>
<td>0.9180</td>
<td>0.9180</td>
</tr>
<tr>
<td>2500</td>
<td>0.9792</td>
<td>0.9229</td>
<td>0.9229</td>
</tr>
<tr>
<td>3000</td>
<td>0.9838</td>
<td>0.9317</td>
<td>0.9267</td>
</tr>
<tr>
<td>3500</td>
<td>0.9876</td>
<td>0.9348</td>
<td>0.9301</td>
</tr>
<tr>
<td>4000</td>
<td>0.9895</td>
<td>0.9355</td>
<td>0.9319</td>
</tr>
<tr>
<td>4500</td>
<td>0.9920</td>
<td>0.9371</td>
<td>0.9332</td>
</tr>
<tr>
<td>5000</td>
<td>0.9975</td>
<td>0.9418</td>
<td>0.9372</td>
</tr>
</tbody>
</table>

### Table III

<table>
<thead>
<tr>
<th>Data Size</th>
<th>KPCA</th>
<th>SKFA</th>
<th>AKFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.8620</td>
<td>0.8700</td>
<td>0.7220</td>
</tr>
<tr>
<td>1000</td>
<td>0.9350</td>
<td>0.9227</td>
<td>0.8957</td>
</tr>
<tr>
<td>1500</td>
<td>0.9567</td>
<td>0.9321</td>
<td>0.8387</td>
</tr>
<tr>
<td>2000</td>
<td>0.9675</td>
<td>0.9659</td>
<td>0.8548</td>
</tr>
<tr>
<td>2500</td>
<td>0.9740</td>
<td>0.9746</td>
<td>0.9102</td>
</tr>
<tr>
<td>3000</td>
<td>0.9783</td>
<td>0.9823</td>
<td>0.9115</td>
</tr>
<tr>
<td>3500</td>
<td>0.9814</td>
<td>0.9832</td>
<td>0.9838</td>
</tr>
<tr>
<td>4000</td>
<td>0.9838</td>
<td>0.9894</td>
<td>0.9769</td>
</tr>
<tr>
<td>4500</td>
<td>0.9821</td>
<td>0.9849</td>
<td>0.9548</td>
</tr>
<tr>
<td>5000</td>
<td>0.9793</td>
<td>0.9688</td>
<td>0.9602</td>
</tr>
</tbody>
</table>
The reconstruction errors appeared to be stable for all methods as $n$ increased. The experimental results in Tables I and II show that there appeared to be a trade off between the reconstruction error and the computational time.

**D. Evaluation of Classification Performance for Polyp Candidates**

Table III shows how the three methods (KPCA, SKFA, and AKFA) cluster the feature distributions with respect to the data size by using a $k$-nearest neighbor classifier. The value of $k$ for the nearest neighbor classifier was set to 10. We defined the classification accuracy as $(\text{TP}+\text{TN})/(\text{TP}+\text{TN}+\text{FP}+\text{FN})$. Here, the following notations were used: TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative). As shown in Table III, the classification accuracy increased when the data size increased; however, there was not much difference in the overall classification performance among the three methods.

Fig. 5 shows the receiver-operating characteristic (ROC) curves that show the classification performance of the three methods by using the $k$-nearest neighbor classifier. In the $k$-nearest neighbor classifier, each member of the testing data in the kernel space was assigned to the majority
class among the k nearest training data. The number of k was called the majority voting number. Distances from the testing data to all training data were computed and the majority voting number for the nearest neighborhood data were selected. The testing data were classified to the closest class in kernel feature space of our CT colonography, for TP/FP/TN/FN. The best choice for the majority voting number depends upon the data; generally, larger values of k reduce the effect of noise on the classification but make boundaries between classes less distinct. The ROC curve shows the algorithm’s ability to differentiate polyps from false positives with respect to the selected majority voting number; thus the ROC curve can be used as a basis for setting the desired performance level.

To estimate the unbiased performance, we trained and tested the classifiers under the same condition as that of Table III, i.e., a 15 % cross-validation method with averaging of 6 sets at each data size ranging from 500 to 3000 at the eigen-dimension of 70. In this method, each candidate was removed, in turn, from the set of all polyp candidates and the classifier was trained using the remaining candidates. A simple $k$-nearest neighbor classifier in the kernel space was generated and evaluated on the removed candidate. Generally, a larger value of the area under the ROC curve indicates a higher performance in the classification task. As shown in Fig. 5, the ROC curves of the three methods were very close, showing that the feature selection using the proposed methods provided well-clustered feature distribution for the CT colonographic images, despite the fact that the proposed AKFA was computationally much faster than the other methods.

VI. CONCLUSION

This paper proposed Accelerated Kernel Feature Analysis (AKFA), a faster and more efficient feature extraction algorithm derived from the Sparse Kernel Feature Analysis (SKFA) as a means for the detection of polyps on CT colonographic images. The time complexity of AKFA is $O(\ell n^2)$, which was more efficient than the $O(\ell^2 n^2)$ time complexity of SKFA and the complexity $O(n^3)$ of a more systematic principal component analysis (KPCA). The polyp classification experiment showed that AKFA yields the same level classification performance to that of KPCA using a $k$-nearest neighbor classifier, thus demonstrating that the features extracted by AKFA are practically useful in discrimination of polyps from false positives detections. Therefore, AKFA has the potential to lead a model-based CAD scheme yielding high detection performance of polyps. Such a CAD scheme has the potential of making CT a viable option for screening large patient populations, resulting in early detection of colon cancers and leading to reduced mortality due to colon cancer.
ACKNOWLEDGMENT

This work was supported in part by USPHS Grant CA095279 and American Cancer Society Research Scholar Grant RSG-05-088-01-CCE. The authors would like to acknowledge X. Jiang who programmed the first matlab implementation of AKFA for improving the computation time. The authors are also grateful to R. Snapp and X. Zhu at the University of Vermont for providing initial feedback on the appropriateness of various strategies for the course of the project.

REFERENCES


