

Research Article

An Efficient Kernel Optimization Method for Radar High-Resolution Range Profile Recognition

Bo Chen, Hongwei Liu, and Zheng Bao

National Key Laboratory for Radar Signal Processing, Xidian University, Xi'an 710071, Shaanxi, China

Received 15 September 2006; Accepted 5 April 2007

Recommended by Christoph Mecklenbräuer

A kernel optimization method based on fusion kernel for high-resolution range profile (HRRP) is proposed in this paper. Based on the fusion of l_1 -norm and l_2 -norm Gaussian kernels, our method combines the different characteristics of them so that not only is the kernel function optimized but also the speckle fluctuations of HRRP are restrained. Then the proposed method is employed to optimize the kernel of kernel principle component analysis (KPCA) and the classification performance of extracted features is evaluated via support vector machines (SVMs) classifier. Finally, experimental results on the benchmark and radar-measured data sets are compared and analyzed to demonstrate the efficiency of our method.

Copyright © 2007 Bo Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Radar automatic target recognition (RATR) is to identify the unknown target from its radar-echoed signatures. Target high-range-resolution profile contains more detail target structure information than that of low-range-resolution radar echoes, so it plays an important role in RATR community [1–4]. As is known, radar HRRP is a strong function of target aspect, and serious speckle fluctuation may exist when target-radar orientation changes, which makes HRRP RATR a challenge task. In addition, target may exist at any position in real system, thus the position of an observed HRRP in a time window will vary between measurements, and this time-shift variation should be compensated when performing classification [1–4].

Kernel methods have been applied successfully in solving various problems in machine learning community. A kernel-based algorithm is a nonlinear version of linear algorithm where, through a nonlinear function $\Phi(x)$, the input vector x has been previously transformed to a higher dimensional space F in which we only need to compute inner products (via a kernel function). The attractiveness of such algorithms stems from their elegant treatment of nonlinear problems and their efficiency in high-dimensional problems. For the HRRP recognition, there exists complex nonlinear relation between targets due to the noncooperation and maneuvering characteristic of targets. Therefore, kernel methods cannot be

directly applied to recognition unless the above three problems of influencing HRRP recognition can be solved, which will significantly improve the classification performance [1].

Given two input vectors x and y , their inner product in feature space, F , can be written in the form of kernel K as

$$K(x, y) = (\Phi(x) \cdot \Phi(y)). \quad (1)$$

The popular kernel functions are Gaussian kernel $K(x, y) = \exp(-\gamma\|x - y\|^2)$ with $\gamma > 0$ and polynomial kernel $K(x, y) = ((x \cdot y) + 1)^p$ with $p \in N$. The choice of the right embedding is of crucial importance, since each kernel will create a different structure in the embedding space. The ability to assess the quality of an embedding is hence a crucial task in the theory of kernel machines. Recently, Xiong et al. [5] propose an alternate method for optimizing the kernel function by maximizing a class separability criterion in the empirical feature space. In this paper, we give an extension of the method which can fuse multiple kernel functions. Then for the HRRP recognition, the proposed method is employed to combine the two Gaussian kernels based on l_1 -norm and l_2 -norm distance to eliminate the speckle fluctuation. Unlike other kernel mixture model, in our method every element of a kernel matrix has a different coefficient because of the use of data-dependent kernel [6], which is the reason that we call it fusion kernel. To show its performance, the method is applied to optimize the kernel of KPCA for HRRP RATR proposed by [1].

Finally, the classification performance of features extracted by optimized KPCA is evaluated via support vector machines (SVMs) [7] based on the benchmark and radar-measured HRRP datasets.

2. PROPERTIES OF RADAR HRRP

The radar works in the optics region, and the electromagnetism characteristics of targets can be described by the scattering center target model, which is widely used and also proved to be a suitable target model in SAR and ISAR applications. An HRRP is the coherent sum of time returns from target scatterers located within a range resolution cell, which represents the distribution of target scattering centers along the radar line of sight [3]. The m th complex returned echo in the n th range cell can be written as

$$x_n(m) = \sum_{i=1}^{I_n} \sigma_{n,i} \exp \left[-j \left(\frac{4\pi R_{n,i}(m)}{\lambda} + \theta_{n,i} \right) \right], \quad (2)$$

where I_n denotes the number of target scatterers in the n th range cell, $R_{n,i}(m)$ denotes the distance between radar and the i th scatterer in the m th sampled echo, $\sigma_{n,i}$ and $\theta_{n,i}$ denote the amplitude and initial phase of the i th scatterer echo, respectively.

If the target orientation changes, its HRRP will be changed subsequently. Two phenomena are responsible for it. The first is the scatterer's motion through range cell (MTRC). Given target rotation angle larger enough, the scatterers range variation will be larger than a range resolution cell, thus make the HRRP changed. Apparently the target rotation angle, which leads to MTRC, is subjective to the range resolution of radar and target-cross length. The second phenomenon is the HRRP's speckle effect. Since an HRRP is the coherent summation of multiple scatterers echoes in one range cell, even the target rotation angle meets the condition of target rotation angle limitation to avoid the occurring of MTRC, the phase of each scatterer echo will be changed, thus their coherent summation will be changed subsequently.

If MTRC occurs, it means that the target scattering center model changed. In this case, it is required more templates to represent the target HRRPs. As to the speckle effect, an effective method of HRRP similarity scalar is needed to eliminate its influence on recognition performance, such as the l_1 -norm distance [8].

3. FUSION KERNEL BASED ON l_1 -NORM AND l_2 -NORM GAUSSIAN KERNELS

3.1. l_1 -norm and l_2 -norm Gaussian Kernels

Due to complicated nonlinear relations between radar targets, empirically Gaussian kernel is chosen to perform HRRP recognition, which is proved by the empirical results in [1]. As the above, radar HRRP has the property of speckle effect especially for propeller-driven aircraft, the running propeller of which modulates the echoes and leads to the great fluctuation of echoes. Usually, l_2 -norm Gaussian kernel is used,

which includes a square operation and augments the influence of the elements of large value in a vector, which will also enhance the effect of speck fluctuation on recognition. Since [8] shows us that l_1 -norm distance criterion can decrease the fluctuation produced by propeller, l_1 -norm Gaussian kernel can eliminate the speckle effect of HRRP,

$$K(X_1(t), X_2(t)) = \exp(-\gamma \|X_1(t) - X_2(t)\|_{l_1}), \quad (3)$$

where $X_1(t)$ and $X_2(t)$ denote two individual HRRPs, and γ is a kernel parameter, which can be determined by a particular criterion.

However, the useful information of HRRP exists in only a part of all range cells and the rest are noise signal. Although l_1 -norm distance can eliminate the speckle effect, the side lobes also have been driven up, which means the increase of the interference of the noise to signal. Whereas l_2 -norm distance can work well on decreasing the noise effect, so we expect to combine the two Gaussian kernels based on different scales to learn a kernel function adaptive to HRRP data. In the next section, a kernel optimization method will be given.

3.2. Kernel optimization based on fusion Kernel in the empirical feature space

Although kernel-based methods such as KPCA [1] can represent complex nonlinear relations among targets, the choices of kernels and the kernel parameters still greatly influence the classification performance. Obviously, a poor choice will degrade the final results. Ideally, we select the kernel based on our prior knowledge of problem domain and restrict the learning to the task of selecting the particular pattern function in the feature space defined by the chosen kernel. Unfortunately, it is not always possible to make the right choice of kernel a priori. Furthermore, there is no general kernel suitable to all datasets. Therefore, it is necessary to find a data-dependent objective function to evaluate kernel functions. The method by Xiong et al. [5] employs a data-dependent kernel similar to that used in [6] as the objective kernel to be optimized. In this section, we firstly review the kernel optimization method.

3.2.1. Kernel optimization based on the single Kernel (SKO)

Given a two-class training data $(x_1, z_1), (x_2, z_2), \dots, (x_m, z_m) \in R^d \times \{\pm 1\}$, where $x_i \in R^d$ is the i th sample and $z_i \in \{\pm 1\}$ the label corresponding to x_i . Given two data samples x and y , a data-dependent kernel function is used,

$$k(x, y) = q(x)q(y)k_0(x, y), \quad (4)$$

where $x, y \in R^d$, $k_0(x, y)$, called the basic kernel, is an ordinary kernel such as a Gaussian or polynomial kernel, and $q(\cdot)$ is a factor function of the form

$$q(x) = \alpha_0 + \sum_{i=1}^n \alpha_i k_1(x, a_i), \quad (5)$$

where $k_1(x, a_i) = e^{-\gamma_1 \|x - a_i\|^2}$, $\{a_i \in R^d, i = 1, 2, \dots, n\}$, called the "empirical cores," can be chosen from the training data

or local centers of the training data, and α_i 's are the combination coefficients which need normalizing. According to [9, 10], evidently the data-dependent kernel satisfies the Mercer condition for a kernel function.

The kernel matrices corresponding to $k(x, y)$ and $k_0(x, y)$ are denoted by K and K_0 , so (4) can be rewritten as

$$K = QK_0Q, \quad (6)$$

where Q is a diagonal matrix, whose diagonal elements are $\{q(x_1), q(x_2), \dots, q(x_m)\}$. We denote the vectors $(q(x_1), q(x_2), \dots, q(x_m))^T$, and $(\alpha_0, \alpha_1, \dots, \alpha_n)^T$ by \mathbf{q} and $\boldsymbol{\alpha}$, respectively. Then, we have

$$\mathbf{q}(\boldsymbol{\alpha}) = \begin{pmatrix} 1 & k_1(x_1, a_1) & \cdots & k_1(x_1, a_n) \\ 1 & k_1(x_2, a_1) & \cdots & k_1(x_2, a_n) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & k_1(x_m, a_1) & \cdots & k_1(x_m, a_n) \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = K_1 \boldsymbol{\alpha}. \quad (7)$$

Here, the following quantity for measuring the class separability is used as the kernel quality function in the empirical feature space

$$J = \frac{\text{trace}(S_b)}{\text{trace}(S_w)}, \quad (8)$$

where $S_b = \sum_{i=1}^2 p_i (\mu_i - \mu)(\mu_i - \mu)^T$ is the ‘‘between-class scatter matrix’’ and $S_w = (1/n) \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T$ the ‘‘within-class scatter matrix,’’ μ is the global mean vector, μ_i is the mean vector of i th class, and $p_i = n_i/n$ is the prior of i th class. It is obvious that optimizing the kernel through J means increasing the linear separability of training data in feature space so that the performance of kernel machines is improved.

Now for the sake of convenience, we assume that the first m_1 data belong to class C_1 , that is, $z_i = 1, i \leq m_1$, and the remaining m_2 data belong to $C_2 (m_1 + m_2 = m)$. Then, the kernel matrix can be written as

$$K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}, \quad (9)$$

where K_{11}, K_{12}, K_{21} , and K_{22} represent the submatrices of K of order $m_1 \times m_1, m_1 \times m_2, m_2 \times m_1, m_2 \times m_2$, respectively. Now we can construct two kernel scatter matrices in the feature space as the following matrices:

$$B = \begin{pmatrix} \frac{1}{m_1} K_{11} & 0 \\ 0 & \frac{1}{m_2} K_{22} \end{pmatrix} - \begin{pmatrix} \frac{1}{m} K_{11} & \frac{1}{m} K_{12} \\ \frac{1}{m} K_{21} & \frac{1}{m} K_{22} \end{pmatrix}, \quad (10)$$

$$W = \begin{pmatrix} k_{11} & 0 & \cdots & 0 \\ 0 & k_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & k_{mm} \end{pmatrix} - \begin{pmatrix} \frac{1}{m_1} K_{11} & 0 \\ 0 & \frac{1}{m_2} K_{22} \end{pmatrix}.$$

Similarly, matrices B_0 and W_0 correspond to the basic kernel K_0 . According to [3, Theorem 1], we can use the kernel scatter matrices to represent J ,

$$J(\boldsymbol{\alpha}) = \frac{\mathbf{1}_m^T B \mathbf{1}_m}{\mathbf{1}_m^T W \mathbf{1}_m} = \frac{\mathbf{q}(\boldsymbol{\alpha})^T B_0 \mathbf{q}(\boldsymbol{\alpha})}{\mathbf{q}(\boldsymbol{\alpha})^T W_0 \mathbf{q}(\boldsymbol{\alpha})}, \quad (11)$$

where $\mathbf{1}_m$ is the vector of ones of the length m .

To maximize $J(\boldsymbol{\alpha})$, the standard gradient approach is employed and an updating equation for maximizing the class separability J is given by the following:

$$\boldsymbol{\alpha}_{(n+1)} = \boldsymbol{\alpha}_{(n)} + \eta \left(\frac{K_1^T B_0 K_1}{\mathbf{q}(\boldsymbol{\alpha}_{(n)})^T W_0 \mathbf{q}(\boldsymbol{\alpha}_{(n)})} - J(\boldsymbol{\alpha}_{(n)}) \frac{K_1^T W_0 K_1}{\mathbf{q}(\boldsymbol{\alpha}_{(n)})^T W_0 \mathbf{q}(\boldsymbol{\alpha}_{(n)})} \right) \boldsymbol{\alpha}_{(n)}, \quad (12)$$

η is the learning rate and to ensure the convergence of the algorithm, a gradually decreasing learning rate is adopted,

$$\eta(t) = \eta_0 \left(1 - \frac{t}{N} \right), \quad (13)$$

where η_0 is the initial learning rate, N denotes a prespecified number of iterations, and t the current iteration number.

We utilize artificially-generated data in two dimensions in order to illustrate graphically the influence of kernel functions on classification. Both class 1 (denoted by ‘‘*’’) and class 2 (‘‘o’’) were generated from mixtures of two Gaussians by Ripley [11] with the classes overlapping to the extent that the Bayes error is around 8.0% and the linear SVM error is 10.5%.

KPCA was used to extract feature with three initial ‘‘bad guesses’’ of kernel matrices (Gaussian kernels with $\gamma = 10$ and $\gamma = 0.25$, polynomial kernel with $p = 2$). For notation simplification: the three kernels were respectively noted as G10, G0.25, and P2) which were all normalized. Linear SVM was used as a classifier. Figure 1(a) shows the original distribution with 125-example training set (randomly chosen from original Ripley’s 250). Figure 1(b) shows the projection of training set in the Gaussian kernel ($\gamma = 10$) induced feature space. The test error (the associated 1000-example test set) for KPCA_{G10} (26.8%) is far inferior to the original without KPCA which means it is a mismatched kernel. Figure 1(c) shows the projection of the training set through $\text{KPCA}_{G0.25}$. The test error for $\text{KPCA}_{G0.25}$ (10.3%) is slightly inferior to the original, which means a matched kernel. Figure 1(d) shows the projection of the training set through KPCA_{P2} . The test error for KPCA_{P2} (10.7%) is slightly inferior to the original. Figures 1(e), 1(f), 1(g) show the projections of the training set after SKO-KPCA $_{G10}$, SKO-KPCA $_{G0.25}$, and SKO-KPCA $_{P2}$. A value of γ_1 of the function $k_1(\cdot, \cdot)$ in (3) for SKO-KPCA was selected using cross-validation (CV). 50 fifty centers were selected to form the empirical core set $\{a_i\}$. The initial learning rate η_0 was set to 0.01 and the total iteration number $N = 200$. The test errors for SKO-KPCA $_{G10}$ (25.6%), SKO-KPCA $_{G0.25}$ (9.4%),

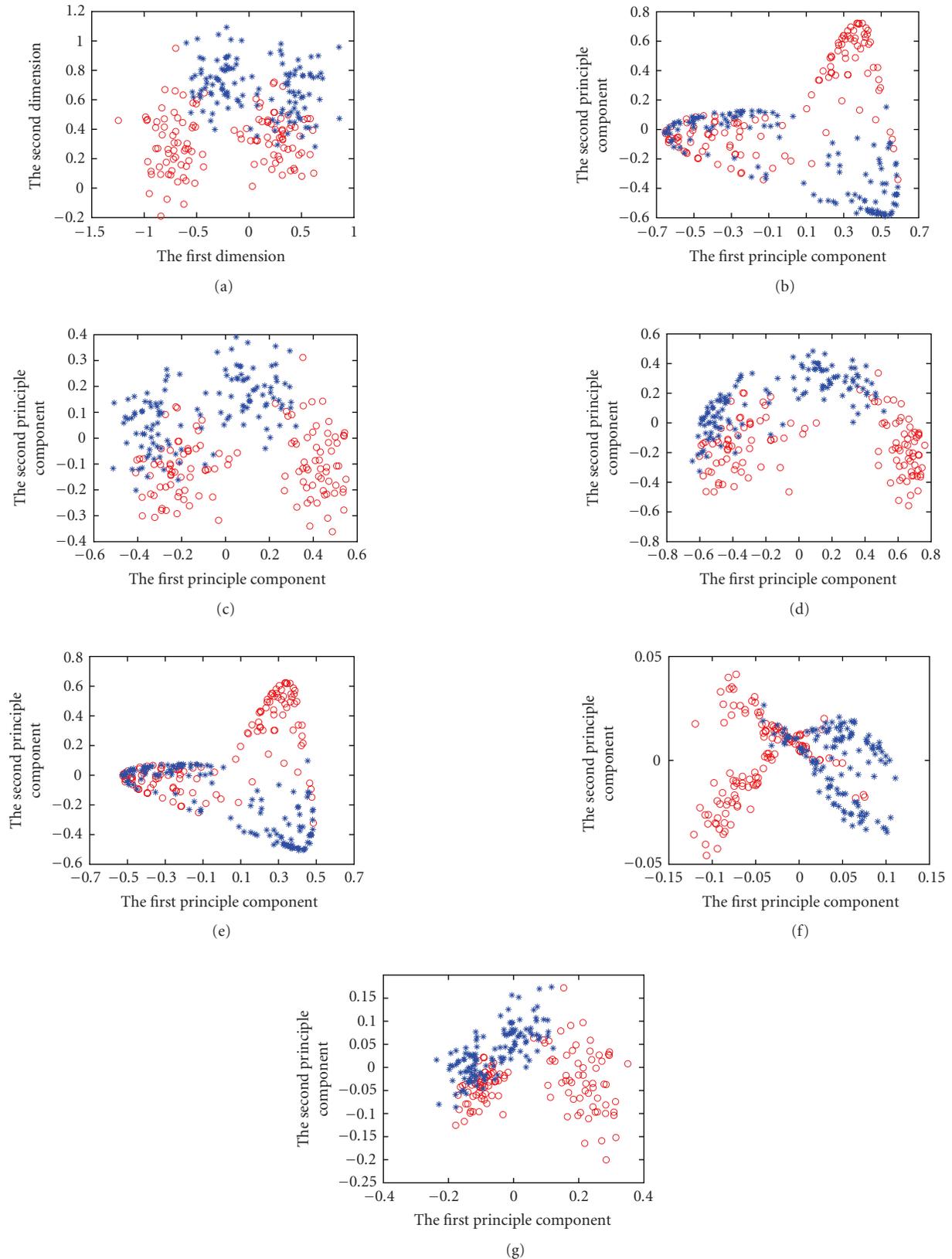


FIGURE 1: Ripley's Gaussian mixture data set and its projections in the empirical feature space onto the first two significant dimensions. (a) The original training data set. (b)–(d) two-dimensional projections of the original training data set, respectively, in G10, G0.25, and P2 kernel induced feature space. (e)–(g) two-dimensional projections of the original training data set, respectively, in G10, G0.25, and P2 kernel induced feature space after the single kernel optimization.

and SKO-KPCA_{p2} (10.1%) were superior to those before kernel optimization.

However, we can see that the performance of SKO method is very dependent on and limited by the initial selected kernel. Which kernel function should be selected to be optimized, Gaussian kernel or other ones? How can we learn a better kernel matrix from different kernels corresponding to different physical interests adaptive to the input data? These problems are difficult to handle for the SKO method. In the next section, we generalize the SKO method to a kernel optimization algorithm based on fusion kernel (FKO).

3.2.2. Kernel optimization based on fusion Kernel (FKO)

It is evident that the method by Xiong et al. [5] is effective to improve the performance of the kernel machines, since the targets are linearly separable in the feature space. Also the experimental results in [5] prove it valid. Nevertheless, the kernel optimization method is based on the single kernel, which means that if a basic kernel function K_0 is chosen beforehand, we have to optimize the kernel based on the single embedding space, the optimization capability will be limited consequentially. To generalize the method we extend it to propose a more general kernel optimization approach combining with the idea of fusion kernel mentioned in the above.

If we choose L kernel functions, (6) can be represented as

$$K = \sum_{i=1}^L Q_i K_0^{(i)} Q_i, \quad (14)$$

where $K_0^{(i)}$ is the i th basic kernel, Q_i is the factor matrix corresponding to $K_0^{(i)}$. B and W are modified as

$$\begin{aligned} B^{\text{fusion}} &= \sum_{i=1}^L B_i, \\ W^{\text{fusion}} &= \sum_{i=1}^L W_i, \end{aligned} \quad (15)$$

where

$$\begin{aligned} B_i &= \begin{pmatrix} \frac{1}{m_1} K_{11}^{(i)} & 0 \\ 0 & \frac{1}{m_2} K_{22}^{(i)} \end{pmatrix} - \begin{pmatrix} \frac{1}{m} K_{11}^{(i)} & \frac{1}{m} K_{12}^{(i)} \\ \frac{1}{m} K_{21}^{(i)} & \frac{1}{m} K_{22}^{(i)} \end{pmatrix}, \\ W_i &= \begin{pmatrix} k_{11}^{(i)} & 0 & \cdots & 0 \\ 0 & k_{22}^{(i)} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & k_{mm}^{(i)} \end{pmatrix} - \begin{pmatrix} \frac{1}{m_1} K_{11}^{(i)} & 0 \\ 0 & \frac{1}{m_2} K_{22}^{(i)} \end{pmatrix}. \end{aligned} \quad (16)$$

According to (11), the fusion kernel quality function J^{fusion} can be written as

$$J^{\text{fusion}} = \frac{\mathbf{1}_m^T B^{\text{fusion}} \mathbf{1}_m}{\mathbf{1}_m^T W^{\text{fusion}} \mathbf{1}_m} = \frac{\sum_{i=1}^L q_i^T B_0^{(i)} q_i}{\sum_{i=1}^L q_i^T W_0^{(i)} q_i}, \quad (17)$$

where

$$q_i = \begin{pmatrix} 1 & k_1(x_1, a_1) & \cdots & k_1(x_1, a_n) \\ 1 & k_1(x_2, a_1) & \cdots & k_1(x_2, a_n) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & k_1(x_m, a_1) & \cdots & k_1(x_m, a_n) \end{pmatrix} \begin{pmatrix} \alpha_0^{(i)} \\ \alpha_1^{(i)} \\ \vdots \\ \alpha_n^{(i)} \end{pmatrix} = K_1 \alpha^{(i)}. \quad (18)$$

The matrices $B_0^{(i)}$ and $W_0^{(i)}$ correspond to the basic kernel $K_0^{(i)}$. $\alpha^{(i)}$ is the combination coefficient vector corresponding to $K_0^{(i)}$.

Therefore, (17) can be derived as

$$J^{\text{fusion}} = \frac{q^T B_0^{\text{fusion}} q}{q^T W_0^{\text{fusion}} q}, \quad (19)$$

where

$$\begin{aligned} B_0^{\text{fusion}} &= \begin{bmatrix} B_0^{(1)} & 0 & \cdots & 0 \\ 0 & B_0^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & B_0^{(L)} \end{bmatrix}, \\ W_0^{\text{fusion}} &= \begin{bmatrix} W_0^{(1)} & 0 & \cdots & 0 \\ 0 & W_0^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & W_0^{(L)} \end{bmatrix}, \\ q &= [q_1 \ q_2 \ \cdots \ q_L] = \begin{bmatrix} K_1 & 0 & \cdots & 0 \\ 0 & K_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K_1 \end{bmatrix} \begin{bmatrix} \alpha^{(1)} \\ \alpha^{(2)} \\ \vdots \\ \alpha^{(L)} \end{bmatrix} \\ &= K_1^{\text{fusion}} \alpha^{\text{fusion}}, \end{aligned} \quad (20)$$

and K_1^{fusion} is an $Lm \times L(n+1)$ matrix, α^{fusion} is a vector of length $L(n+1)$.

Obviously, we can find that the form of (19) is the same as that of the right-hand side of (11), so through (12), our result can also be given by the following

$$\begin{aligned} \alpha_{(n+1)}^{\text{fusion}} &= \alpha_{(n)}^{\text{fusion}} + \eta \left(\frac{(K_1^{\text{fusion}})^T B_0^{\text{fusion}} K_1^{\text{fusion}}}{q(\alpha_{(n)}^{\text{fusion}})^T W_0^{\text{fusion}} q(\alpha_{(n)}^{\text{fusion}})} \right. \\ &\quad \left. - J^{\text{fusion}} \frac{(K_1^{\text{fusion}})^T W_0^{\text{fusion}} K_1^{\text{fusion}}}{q(\alpha_{(n)}^{\text{fusion}})^T W_0^{\text{fusion}} q(\alpha_{(n)}^{\text{fusion}})} \right) \alpha_{(n)}^{\text{fusion}}. \end{aligned} \quad (21)$$

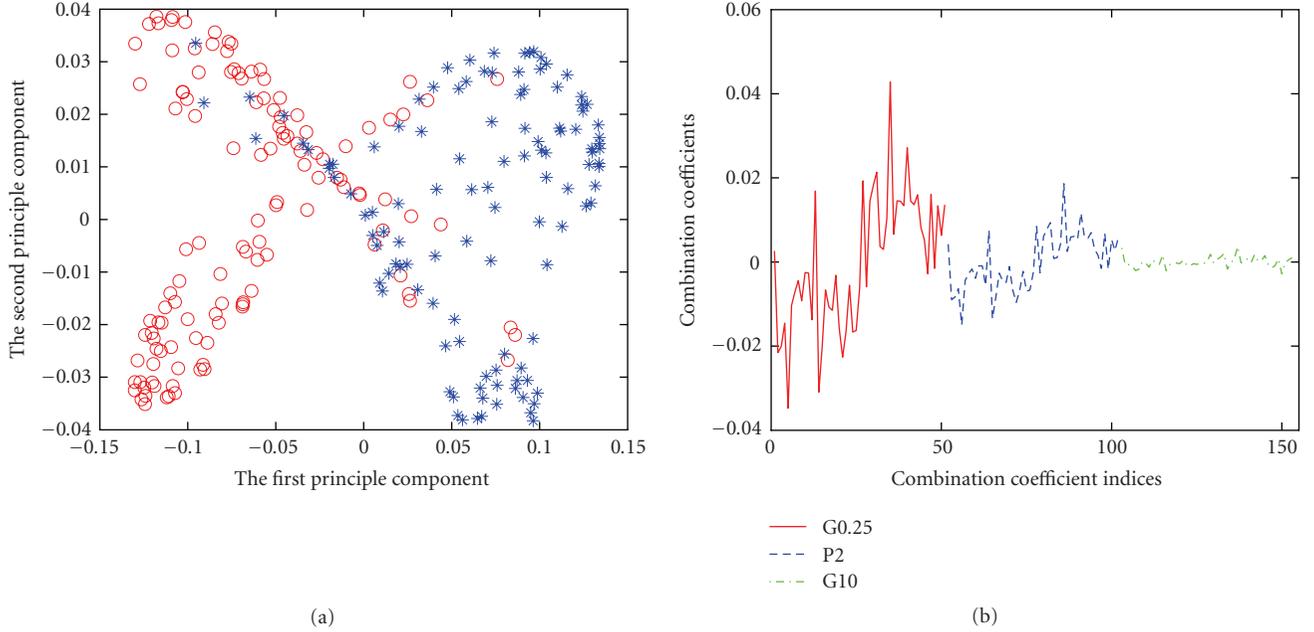


FIGURE 2: The results of Ripley's data after FKO. (a) Two-dimensional projection of the training data in the optimized feature space. (b) the combination coefficients α^{fusion} .

When $L = 1$, it is just the single kernel optimization.

Figure 2(a) shows the projection of the training set in the empirical feature space after the FKO. The parameters were the same as those of the single kernel optimization. The classifier was still linear SVM. The test error (9.0%) demonstrates the improvement of the performance of the classification. Meanwhile, shown in Figure 2(b) are the combination coefficients of three initial kernels, α^{fusion} . From Figure 2(b), we can clearly find that after kernel optimization, the combination coefficients of the mismatched kernel G0.1 have been far less than other ones. Equivalently, our method automatically selected G4 and P2 kernels to be optimized, which both match the Ripley's data for classification.

4. EXPERIMENTAL RESULTS

4.1. Benchmark datasets

In order to evaluate the performance of our method, we firstly test it on the four-benchmark datasets, namely, the ionosphere, Pima Indians diabetes, liver disorder, wisconsin breast cancer (WBC, where the 16 database samples with missing values have been removed) which are downloaded from the UCI benchmark repository [12]. Except Pima data with training and test sets, in order to evaluate the true performance, other data are randomly partitioned into two equal and disjoint parts which are respectively used as training and test sets.

As the above, kernel optimization methods were applied to KPCA. Linear SVM classifier was utilized to evaluate the classification performances. We used a Gaussian kernel

function, a polynomial kernel function $K_2(x, y) = ((x^T \cdot y) + 1)^p$, and a linear kernel function $K_3(x, y) = x^T \cdot y$ as initial basic kernel matrices. And all kernels were normalized. Firstly, the values of kernel parameters for the three kernel functions of KPCA without kernel optimization were respectively selected by 10-fold cross-validation. Then the chosen kernel functions were applied as the basis kernels in (4). σ_1 's in (5) were also selected using 10-fold cross-validation. 20 local centers were selected to form the empirical core set $\{a_i\}$. The initial learning rate η_0 was set to 0.08 and the total iteration number T was set to 400. Meanwhile, the procedure of determining the parameters of SKO was the same as FKO.

Experimental results on benchmark data are summarized in Table 1. It is evident that FKO can further improve the classification performance and at least as the same as the SKO method. The combination coefficients of three kernels in the four experiments has also been illustrated in Figure 3. We find that the combination coefficients of FKO are dependent on the classification performance of the corresponding kernel in SKO. As shown in Figure 3, the better the kernels can work after the optimization of SKO, the larger the combination coefficients of FKO are. Apparently, FKO can automatically combine three fixed parameter kernels.

4.2. Measured high-resolution range profile (HRRP) radar data set

The data used to further evaluate the classification performance are measured from a C band radar with bandwidth of 400 MHz. The radar high range resolution profile (HRRP) data of three airplanes, including An-26, Yark-42, and Cessna

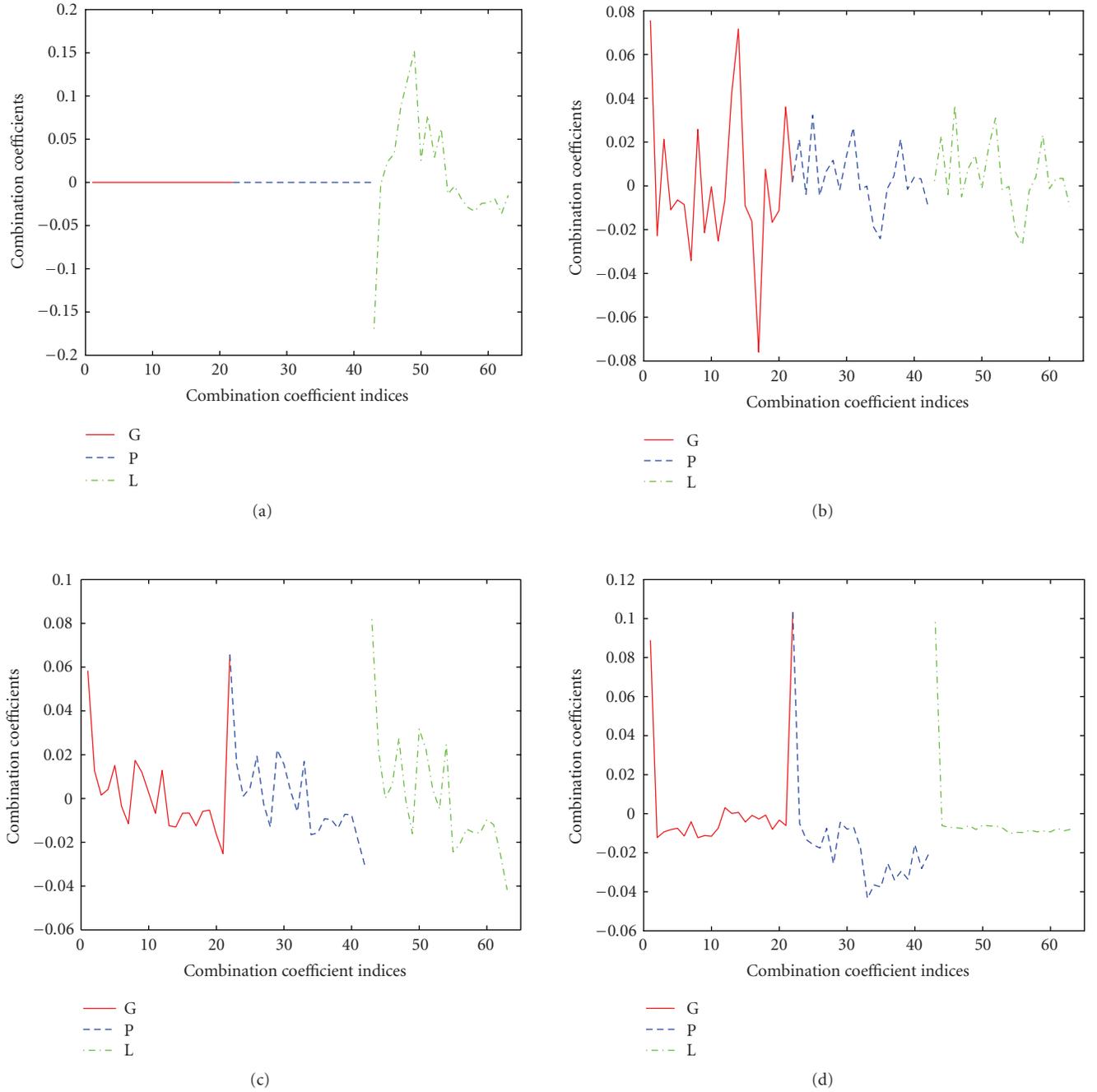


FIGURE 3: The combination coefficients corresponding to four datasets. (a) BCW; (b) pima; (c) liver; (d) ionosphere.

Citation S/II, are measured continuously when the targets are flying. The projections of target trajectories onto the ground plane are shown in Figure 4. The measured data of each target are divided into several segments, the training data and test data are chosen from different data segments, respectively, which means, the target orientations corresponding to the test data and training data are different, the maximum elevation difference between the test data and training data is about 5 degrees. The 2nd and the 5th segments of Yark-42, the 5th and the 6th segments of An-26, the 6th and the 7th

segments of Cessna Citation S/II are chosen as the training data the total number of which is 300, all the rest data segments are chosen as tests data, the total number of which is 2400. And in the kernel optimization, 50 local centers from the training data are used as empirical cores. Additionally, the original HRRPs are preprocessed by power transformation (PT) to improve the classification performance, which is defined as

$$Y(t) = X(t)^\nu, \quad 0 < \nu < 1, \quad (22)$$

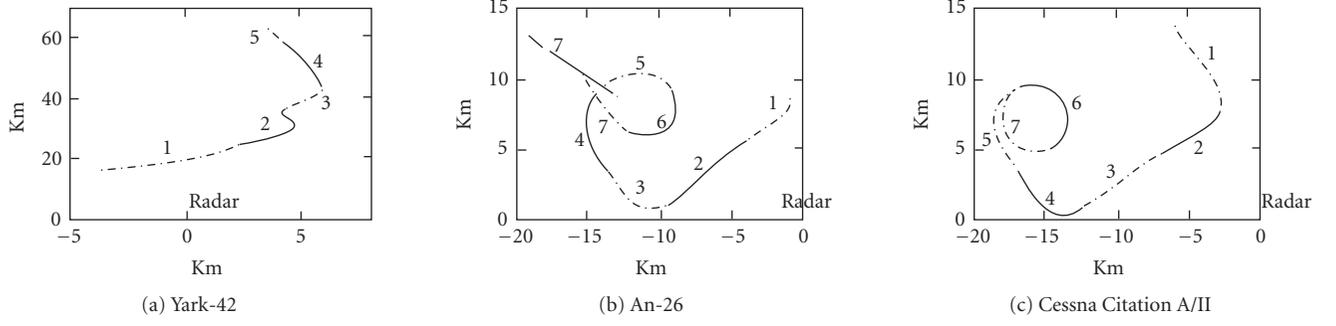


FIGURE 4: The projection of target trajectories onto the ground plane. (a) Yark-42, (b) An-26, (c) cessna citation S/II.

TABLE 1: The comparison of recognition rates of different methods in different experiments. K_1 , K_2 , and K_3 , respectively, correspond to Gaussian, polynomial, and linear kernels.

		K_1	K_2	K_3
BCW	KPCA	88.96%	90.45%	88.58%
	KPCA with SKO	88.96%	96.94%	97.1%
	KPCA with FKO	—	97.33%	
Pima	KPCA	73.72%	64.15%	63.48%
	KPCA with SKO	73.72%	66.21%	64.63%
	KPCA with FKO	—	74.10%	
Liver	KPCA	71.19%	69.47%	66.19%
	KPCA with SKO	74.67%	73.36%	73.47%
	KPCA with FKO	—	75.17%	
Ionosphere	KPCA	93.11%	93.11%	89.73%
	KPCA with SKO	93.11%	93.55%	89.38%
	KPCA with FKO	—	93.55%	

where $X(t)$ denotes an individual HRRP. The reason that using PT can improve the classification performance is explained as that the nonnormality distributed original HRRPs will become near normality distribution after PT, thus makes the performance of many classifiers optimal. From HRRP physical properties of view, PT amplifies the weaker echoes and compresses the stronger echoes so as to decrease the speckle effect in measuring the HRRPs similarity. The details about PT can be referred to [13].

One-against-all linear SVM classifiers are trained for the feature vectors extracted by SKO-KPCA, FKO-KPCA, and KPCA without the kernel optimization, respectively. The parameters in the experiments are listed in Table 2. The experimental results are shown in Figure 5, where the x -axis represents the number of principle components and the y -axis the recognition rate.

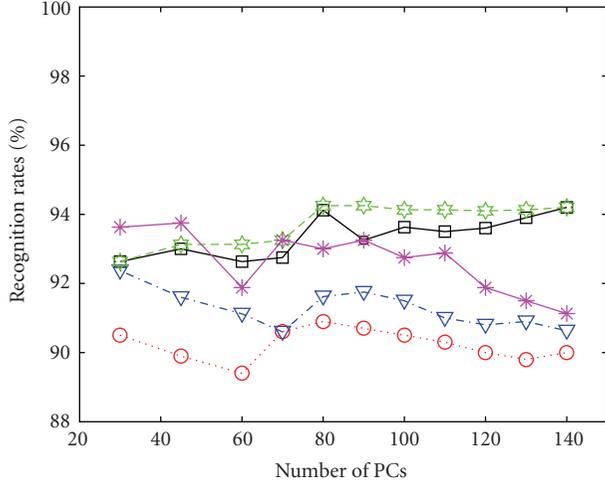
In Figures 5(a)–5(c), each target recognition rates in different KPCA methods are shown respectively. Figure 5(d) indicates the average recognition rates. From Figures 5(a) and 5(d), we can find that KPCA with l_1 -norm and fusion

TABLE 2: The parameters in the experiment.

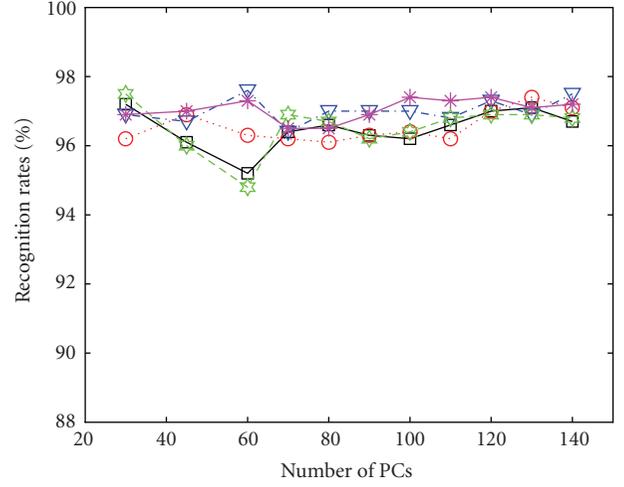
	γ	Empirical centers no.	γ_1	η_0	Iteration no.
KPCA1	0.001				
KPCA2	0.001				
SKO-KPCA1	0.001	50	1	0.02	200
SKO-KPCA2	0.001	50	1	0.001	200
FKO-KPCA	0.001	50	1	0.0003	200

Note: KPCA1 and KPCA2 correspond to KPCA with l_1 -norm and 2-norm Gaussian kernels; SKO-KPCA1 and SKO-KPCA2 correspond to KPCA with l_1 -norm and 2-norm Gaussian kernels after single kernel optimization; FKO-KPCA represents KPCA after fusion kernel optimization based on l_1 -norm and 2-norm Gaussian kernels.

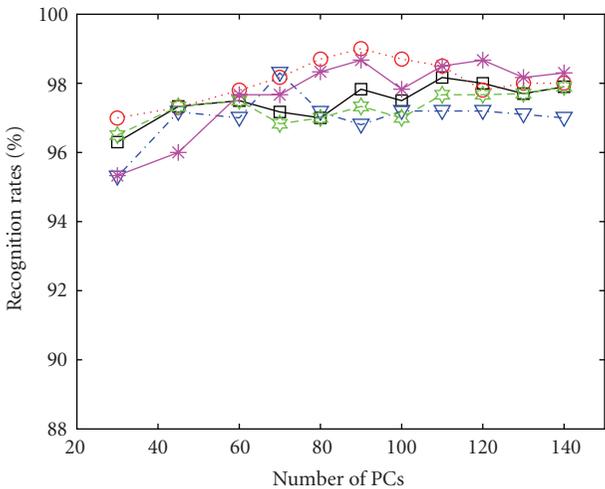
Gaussian kernels perform better than with l_2 -norm Gaussian kernel because of the different performances on An26. Due to the modulability of the propeller of An26 on the HRRPs, there still exists speckle effect even in a small angle sector. Therefore, l_1 -norm Gaussian kernel can be employed to eliminate the large fluctuation so as to improve the recognition performance. By the fusion of l_1 -norm Gaussian kernel, FKO-KPCA also works well. Meanwhile, Figure 5(d) shows that the recognition rate of l_1 -norm Gaussian kernel SKO-KPCA reaches 96.30% when the number of principle component equals 140, while FKO-KPCA method only needs 90 components to reach its best classification rate 96.27%. Since the fewer components mean lower computational complexity, FKO-KPCA can extract effective features to reduce the computational complexity in comparison KPCA with l_1 -norm Gaussian kernel. Why can FKO-KPCA outperform KPCA with l_1 -norm Gaussian kernel? In our opinion, the possible reason is that when restricting the speckle effect, l_1 -norm distance also augments the noise to interfere the signal, therefore, FKO-KPCA achieves better performance on Cessna and Yark42 than KPCA with l_1 -norm Gaussian kernel, as shown in Figures 5(b) and 5(c), which suggests that our optimization method can adaptively combine the characteristics of two kinds of kernels. From Figure 5(d), we can also observe that SKO-KPCA cannot effectively optimize the



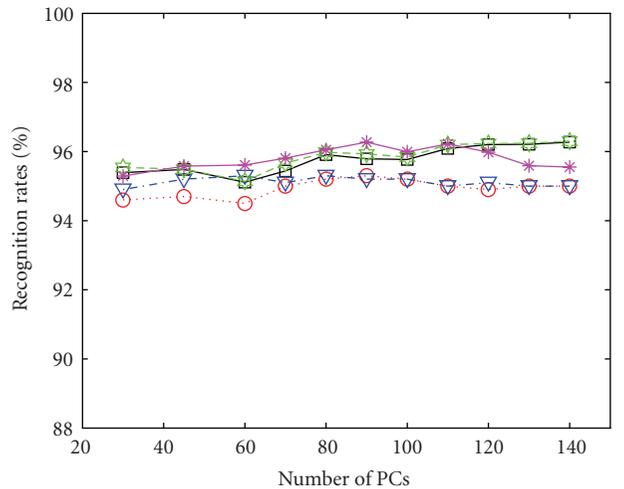
(a)



(b)



(c)



(d)

FIGURE 5: Recognition rates on the measured radar HRRP data versus number of principle components in three experiments. (a) An-26 (b) Cessna (c) Yark-42 (d) average recognition rates.

origin KPCA and l_1 -norm SKO-KPCA even decreased the recognition rates.

5. CONCLUSIONS

In this paper, a kernel optimization method with learning ability for radar HRRP recognition is proposed. The method can adaptively combine the different characteristics of l_1 -norm and l_2 -norm Gaussian kernels, so that not only is the

kernel function optimized but also the speckle fluctuations of HRRP are restrained. Because of the use of kernel function adaptive to data, each element in kernel matrix corresponds to independent coefficient, which is the reason why it is called fusion kernel optimization method. The classification performance of features extracted by optimized KPCA are analyzed and compared via support vector machines (SVMs) based on benchmark and measured HRRP datasets, which demonstrates the efficiency of our method.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation of China (no.60302009).

REFERENCES

- [1] B. Chen, H. Liu, and Z. Bao, "PCA and kernel PCA for radar high range resolution profiles recognition," in *Proceedings of IEEE International Radar Conference*, pp. 528–533, Arlington, Va, USA, May 2005.
- [2] B. Chen, H. Liu, and Z. Bao, "An efficient kernel optimization method for high range resolution profile recognition," in *Proceedings of IEEE International Radar Conference*, pp. 1440–1443, Shanghai, China, October 2006.
- [3] L. Du, H. Liu, Z. Bao, and M. Xing, "Radar HRRP target recognition based on higher order spectra," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2359–2368, 2005.
- [4] L. Du, H. Liu, Z. Bao, and J. Zhang, "A two-distribution compounded statistical model for radar HRRP target recognition," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2226–2238, 2006.
- [5] H. Xiong, M. N. S. Swamy, and M. O. Ahmad, "Optimizing the kernel in the empirical feature space," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 460–474, 2005.
- [6] S. Amari and S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Neural Networks*, vol. 12, no. 6, pp. 783–789, 1999.
- [7] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [8] Z. Bao, M. Xing, and T. Wang, *Radar Imaging Technique*, Publishing House of Electronics Industry, Beijing, China, 2005.
- [9] B. Schölkopf, S. Mika, C. J. C. Burges, et al., "Input space versus feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, 1999.
- [10] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, UK, 2004.
- [11] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, UK, 1996.
- [12] C. Blake, E. Keogh, and C. J. Merz, "UCI repository of machine learning databases," Tech. Rep., Department of Information and Computer Science, University of California, Irvine, Calif, USA, 1998. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- [13] H. Liu and Z. Bao, "Radar HRR profiles recognition based on SVM with power-transformed-correlation kernel," in *Proceedings of International Symposium on Neural Networks (ISNN '04)*, vol. 3173 of *Lecture Notes in Computer Science*, pp. 531–536, Dalian, China, August 2004.

Bo Chen received his B.Eng. and M.Eng. degrees in electronic engineering from Xidian University in 2003 and 2006, respectively. He is currently a Ph.D. student in the National Key Lab of Radar Signal Processing, Xidian University. His research interests include radar signal processing, radar automatic target recognition, kernel machine.



Hongwei Liu received his M.S. and Ph.D. degrees all in electronic engineering from Xidian University in 1995 and 1999, respectively. He joined the National Key Lab of Radar Signal Processing, Xidian University since 1999. From 2001 to 2002, he is a visiting scholar at the department of electrical and computer engineering, Duke University, USA. He is currently a Professor and Director of National Key Lab of Radar Signal Processing, Xidian University. His research interests are radar automatic target recognition, radar signal processing, and adaptive signal processing. He is with the Key Laboratory for Radar Signal Processing, Xidian University, Xi'an, China.



Zheng Bao graduated from the Communication Engineering Institution of China in 1953. Currently, he is a Professor at Xidian University and an Academician of the Chinese Academy of Science. He is the author or coauthor of six books and has published more than 300 papers. Now his research work focuses on the areas of space-time adaptive processing, radar imaging, and radar automatic target recognition. He is with the Key Laboratory for Radar Signal Processing, Xidian University, Xi'an, China.

