Kernel Sparse Representation with Pixel-level and Region-level Local Feature Kernels For Face Recognition

Cuicui Kang*, Shengcai Liao, Shiming Xiang, Chunhong Pan

Institute of Automation, Chinese Academy of Sciences
{*cckang, scliao, smxiang, chpan*}*@nlpr.ia.ac.cn*

Abstract

Face recognition has been popular in pattern recognition field for decades, but it is still a difficult problem due to the various image distortions. Recently, Sparse Representation based Classification (SRC) was proposed as a novel image classification approach, which is very effective with sufficient training samples for each class. However, the performance drops when the number of training samples is limited. In this paper, we show that effective local image features and appropriate nonlinear kernels are needed in deriving a better classification method based on sparse representation. Thus, we propose a novel kernel SRC framework and utilize effective local image features in this framework for robust face recognition. First, we present a kernel coordinate descent (KCD) algorithm for the LASSO problem in the kernel space, and we successfully integrate it in the SRC framework (called KCD-SRC) for face recognition. Second, we employ local image features and develop both pixel-level and region-level kernels for KCD-SRC based face recognition, making it discriminative and robust against illumination variations and occlusions. Extensive experiments are conducted on three public face databases (Extended YaleB, CMU-PIE and AR) under illumination variations, noise corruptions, continuous occlusions, and registration errors, demonstrating excellent performances of the KCD-SRC algorithm combining with the proposed kernels.

Keywords: ℓ_1-norm Minimization, Sparse Representation Classification, Kernels, Coordinate Descent, Face Recognition, LBP

*Corresponding Author. Email: cckang@nlpr.ia.ac.cn. Phone: +86 01062612582.

1. Introduction

Face recognition is an important research area in computer vision. It has many useful applications in real life, such as face attendance, access control, security surveillance, etc. Face recognition is also a challenging problem, which suffers from aging, occlusion, and intra-personal variations with pose, illumination, and expression. Many researchers have been attracted to solve these problems, making a great development in face recognition techniques during the past two decades.

Recently, Wright et al. proposed the Sparse Representation based Classification (SR-C) framework for robust face recognition [1], which makes use of the well-known ℓ_1-norm constrained least-square reconstruction technique (LASSO) [2]. In their work image pixel values are used to represent faces. The training set needs to be carefully constructed, that is, each subject in the training set is constituted with many images representing various lighting conditions. Thus a probe image of certain illumination condition can be represented by a sparse linear combination of the training samples. However, in realistic applications it is hard for every enrolled user to have such varying lighting images. When the number of available images per subject is limited, the linear SRC method may have difficulty in learning the correct representation. Fig.1 shows an example where the linear SRC method (denoted as Pixel Linear) fails to identify a probe face image with severe illumination changes, when only five images per subject are available in the gallery.

The basic assumption in the original SRC algorithm is that the probe image can be linearly represented by gallery images of the same class. However, this assumption is hard to hold in difficult scenarios, for example, illumination variations and occlusions may be present in the probe face image but meanwhile the gallery may have small sample size. Therefore, nonlinear features and nonlinear kernels are needed to deal with difficult face recognition problems. Yuan et al. [3], Min et al. [4] and Chan et al. [5] proposed to utilize the Local Binary Pattern (LBP) descriptor [6] in the SRC framework, so that the system would be more robust against illumination variations, making it potentially possible to apply SRC for small-sample-size face recognition problem.

LBP descriptor was originally proposed by Ojala et al. for texture classification [6]. It is a binary string (often converted to the corresponding decimal number as a label) resulted from local neighboring pixel comparisons. Ahonen et al. have applied LBP to

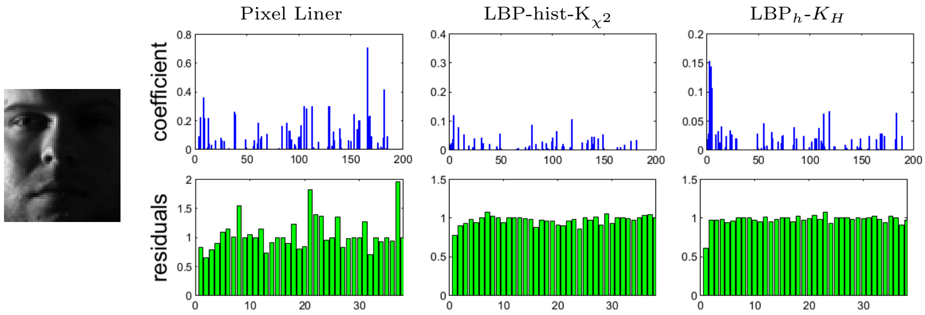

Figure 1: Sparse codings (blue bars) and residuals (green bars) by three kernel methods in representing a probe face image (left) in terms of a gallery of 38 subjects (5 images per subject) from the Extended YaleB database. Pixel Linear denotes the linear SRC method with raw image pixels as features. LBP-Hist-K_{χ^2} denotes kernel SRC with LBP histogram based χ^2 kernel. LBP$_h$-K_H denotes kernel SRC with the proposed LBP binary string and Hamming distance based kernel. The first five images in the gallery have the same identity with the probe image.

face recognition and proved that it is robust for illumination variations [7, 8]. In [3], the LBP encoded images were directly used in the linear SRC system. However, since LBP codes are converted from binary comparisons, i.e. they are labels but not regular numerical values, thus it is not very reasonable to linearly combine LBP values directly. Actually, LBP is mostly used in the form of histogram features counted in local regions, and the χ^2 distance is preferred to calculate distance between two LBP histogram features [6, 7]. Different from [3], Chan et al. [5] proposed to extract LBP histogram features instead of single label for linear SRC based face recognition. However, in all these studies, LBP histogram features have only been integrated in the linear SRC algorithm, which is not able to further discover nonlinear relationships between probe and gallery images via useful nonlinear kernels, such as the histogram intersection and χ^2 kernels. Therefore, in this paper, we propose a kernel SRC based algorithm to effectively integrate nonlinear features in nonlinear kernels for face recognition. In the examples of Fig.1, it shows that kernel SRC with LBP histogram based χ^2 kernel can deal with illumination variations better than the linear SRC method.

The original LBP, as shown in Fig.2, encodes each 3×3 local region by comparing the center pixel value with the 8 neighboring pixels and concatenate the results. Thus, the LBP feature is actually a binary string. Motivated by this, in this paper, we further develop an effective local image feature and Hamming distance based kernel (LBP$_h$-K_H), for better utilizing the advantages of LBP features. Note that while LBP histogram based kernels work in region-level, LBP$_h$-K_H is a pixel-level kernel which can represent local image details better. In Fig.1 it can be observed that, the proposed LBP$_h$-K_H is better than LBP in representing the probe face having severe illumination changes. The success is because with the proposed nonlinear kernel, the relationship between the probe image and the gallery images of the same class can be better discovered.

Therefore, in this work, we propose a novel kernel SRC framework, and apply it together with local image descriptors to face recognition. The contributions can be highlighted as follows.

1. We propose a novel kernel coordinate descent (KCD) algorithm with covariant update technique for the kernel LASSO problem.
2. The KCD algorithm is further applied to face recognition in the SRC framework (KCD-SRC), which enables many effective similarity kernels to be used in the SRC framework for face recognition.
3. We integrate the well-known LBP features into the proposed KCD-SRC framework, where both χ^2 distance based and histogram intersection (HI) based kernels can be successfully applied for effective face recognition.
4. Furthermore, we propose a new pixel-level kernel for the LBP feature, denoted as LBP$_h$-K_H, to the KCD-SRC based face recognition. We show that this simple kernel is fast and robust for face recognition under occlusion and illumination variations.

We have conducted extensive experiments on the Extended YaleB, CMU-PIE and the AR face databases to illustrate the effectiveness of the proposed approach in four scenarios, including illumination variations, noise corruptions, occlusions, and registration errors. The results demonstrate that the kernel enabled methods are more effective and robust than methods without kernel. For example, in many situations, KCD-SRC with LBP$_h$-K_H outperforms raw pixel based SRC method with more than 20% recognition rates under only five training samples per subject.

This work is based on our preliminary work in [9]. The main improvement is the development of the LBP$_h$-K_H kernel, which utilizes only one half of the LBP encodings.

We demonstrate that LBP$_h$-K_H is equivalent to the K_H kernel that we have proposed in [9], but reduces one half of the computation. It is also equivalent to the linear kernel by a modification to the LBP encoding, but LBP$_h$-K_H is more efficient. It is impressive that while the computation time of LBP$_h$-K_H based KCD-SRC method is comparable to that of raw pixels based, the former performs much better. Other main changes contained in this paper include,

1. We give a more detailed formulation of the proposed KCD-SRC algorithm.
2. The HI kernel is evaluated in addition to the χ^2 kernel for LBP histogram based KCD-SRC algorithm.
3. The AR database is included for evaluation in real occlusion scenarios (with sunglasses and scarf).
4. Extended experiments are performed on the CMU-PIE database with various scenarios.
5. A more challenging scenario, misalignment, is evaluated to test the proposed methods for face recognition, including both synthesized and real registration errors.

1.1. Related Work

The kernel LASSO problem (Eq.(2)) has also been suggested by Yuan et al. [3] and Gao et al. [10]. In the work of Yuan et al. [3], the formulation is addressed in a multi-task learning setting, and the resulted optimization problem is solved via the accelerated proximal gradient method [11]. Differently, our formulation does not need to combine multiple features in face recognition. The LBP descriptor is also utilized in their work. However, they combined the gray-level images and LBP encoded images directly to represent each face image, which resulted in a two-task classification problem using sparse representation. As mentioned before, directly operating on LBP labels in a numerical way lacks of physical meanings. In contrast, we suggest applying LBP histograms extracted from local regions for our kernel formulation. In addition, we propose a new LBP$_h$-K_H kernel, which works in pixel-level and it is effective for kernel SRC based face recognition.

Grounded on the feature-sign search [12] algorithm, Gao et al. [10] developed a gradient descent algorithm for finding the sparse codes and the dictionary, alternatively, in the kernel space. In their work, Gaussian kernel is employed for face recognition.

A drawback could be that, giving limited number of training samples, simply using Gaussian kernel may not achieve good performance. In contrast, we apply the coordinate descent based approach to solve the kernel LASSO problem. The computation in each iteration is simple and the convergence can be guaranteed [13]. Furthermore, other than Gaussian kernel, we propose to employ local image feature based kernels, resulting in high performance under only a few training samples per class.

There are also other existing papers working on deriving kernel LASSO algorithms in the machine learning research field, like Wang et al. [14] and Roth [15]. These approaches are targeting on the general regression problem, which is not the main focus of this paper.

LBP descriptor has been proved to be robust under illumination variations, and it is widely applied for face recognition [7]. Typically, Min et al. [4] and Chan et al. [5] have applied LBP features in the SRC framework for face recognition. In their methods, LBP was employed in a popular way where histograms are calculated to gain the performance. In the work of Chan et al. [5], multi-scale LBP histograms are applied to the original SRC framework. They also proved that LBP histogram feature is robust to illumination changes and misalignment. In contrast, Min et al. [4] proposed a pyramid architecture based SRC algorithm with LBP features. However, these two approaches are not able to employ effective kernels (e.g. χ^2) for LBP histogram features. Moreover, the region-level LBP histograms are sensitive to image noises (which will be demonstrated later).

Instead of using LBP histograms, a more straightforward way is to use the LBP binary string directly for image representation. Bai et al. [16] proposed a face recognition method by applying the Hamming distance to measure the difference between two LBP binary strings, and classification is done by a hierarchical model called Multi-expert Intelligent Decision System (MIDS). Yao et al. [17] also proposed to apply the Hamming distance for LBP binary string based representation. Further with a point-to-point matching similarity measure, they showed a better performance over the LBP histogram and χ^2 based [7] matching method. Previously, in [9] we also applied the Hamming distance on LBP features to measure the difference between two LBP images, and hence derived a kernel called K_H for kernel SRC based face recognition. In this paper, we further propose the LBP$_h$-K_H, which utilizes only half of the LBP encodings, and applies the Hamming distance to measure the difference between two reduced LBP encodings. The main contribution is that we give a novel view of this kind of kernel, and we reduce half of the computation compared to the previous kernel. We will show more details in the main text.

Many efforts have been made to deal with the occluded face recognition problem, including subspace based methods [18, 19, 20, 21], SVM based methods [22, 23], SRC based methods [1, 24, 3, 4, 9, 25], and so on. SRC is a recent approach that has been shown to be effective for face recognition under occlusion, and our effort is also along this direction.

1.2. Organization

The remainder of this paper is organized as follows. Section 2 illustrates how to solve the kernel LASSO problem using coordinate descent technique and how to combine it with the SRC framework in detail. Section 3 describes two kinds of local image kernels in both pixel-level and region-level for KCD-SRC based face recognition. In Section 4, a series of experiments are conducted on the Extended YaleB, the CMU-PIE, and the AR face databases to verify the effectiveness of the proposed approach. Finally, we give a discussion in Section 5, and conclude this paper in Section 6.

2. Kernel Sparse Representation based Classification

2.1. Sparse Representation based Classification

Wright et al. exploited the sparsity to classification in [1], showing that sparse representation is effective for face recognition under illumination variations and occlusions. Their approach, known as SRC, is developed on the prior knowledge that a probe image has a strong relationship with its buddies and a weaker relationship with the others. When it is integrated with the important characteristic of LASSO [2] that can shrink parts of the coefficients to zero, it displays a remarkable performance for classification.

The problem can be addressed as follows. Suppose we are given sufficient training samples for every individual, and $\mathbf{x}_{i,j}$ is the descriptor of j-th sample for the i-th individual. Accordingly, we can construct the bases matrix as $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_k] \in \mathbb{R}^{d \times n}$, where the sub-matrix $\mathbf{X}_i = [\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,n_i}] \in \mathbb{R}^{d \times n_i}$ is the bases matrix of the i-th individual, d is the feature dimensionality, and $n = n_1 + n_2 + \ldots + n_k$ is the number of training samples for all of the k individuals. The matrix $\mathbf{X}$ is also called as the dictionary. Given a probe sample $\mathbf{y} \in \mathbb{R}^d$, the linear representation of $\mathbf{y}$ in terms of $\mathbf{X}$ can be written as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}, \tag{1}$$

where $\boldsymbol{\beta}$ is the coefficient vector. If the individual referred by $\mathbf{y}$ has been included in the bases matrix $\mathbf{X}$, the resulting coefficient vector $\boldsymbol{\beta}$ is tend to be a sparse vector whose most none zero entries are associated with the corresponding individual in the gallery. Considering this, in order to ensure the sparsity of $\boldsymbol{\beta}$ and recover it as precise as possible, SRC casts ℓ_1-norm minimization as the following optimization problem

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{2}$$

where $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{n} |\beta_j|$ denotes the ℓ_1 norm of $\boldsymbol{\beta}$, and $\lambda \geq 0$ is a constant parameter for sparsity. Then we can identity $\mathbf{y}$ once the solution to Eq.(2) is obtained. The class identity of $\mathbf{y}$ is based on the minimization of residuals:

$$\min_{c} residual_c(\mathbf{y}) = \|\mathbf{X}\delta_c(\boldsymbol{\beta}) - \mathbf{y}\|_2^2, \tag{3}$$

where $\delta_c(\boldsymbol{\beta})$ is a function that selects the coefficients corresponding to the c-th class and makes the rest to zero. For k classes there are k δ_c functions, which result in k residuals. Thus SRC algorithm labels $\mathbf{y}$ to the class that has the minimum residual.

2.2. Kernel Sparse Representation based Classification

The well-known LASSO problem, i.e. the linear regression problem with the ℓ_1-norm penalty, was first proposed by Tibshirani in [2]. Since then, many researchers have been attracted to this problem. Given n observations with zero means and unit lengths in $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, and their respondences $\mathbf{y} \in \mathbb{R}^d$, the task of LASSO can be written as the following optimization problem

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\left\|\sum\nolimits_{i=1}^{n} \beta_i\mathbf{x}_i - \mathbf{y}\right\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{4}$$

where $\|\boldsymbol{\beta}\|_1$ is the Lasso penalty, and $\lambda \geq 0$ is a constant weight. In this paper, we focus on the LASSO problem of Eq.(4) in kernel space, *i.e.*

$$\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) \doteq \frac{1}{2}\left\|\sum\nolimits_{i=1}^{n} \beta_i\varphi(\mathbf{x}_i) - \varphi(\mathbf{y})\right\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{5}$$

where $\varphi(\cdot)$ is an implicit mapping which maps a feature vector to a kernel space. We assume that $\varphi(\cdot)$ satisfies $\varphi(\mathbf{x})^T\varphi(\mathbf{x}) = 1$. To solve the kernel LASSO of Eq.(5), we develop the KCD algorithm which employs the Coordinate Descent approach [13] due to its simplicity and efficiency. Given all β_i fixed except for $i = j$, minimizing Eq.(5) is equivalent to minimizing

$$\frac{1}{2} \|\beta_j \varphi(\mathbf{x}_j) - \hat{\mathbf{r}}_j\|_2^2 + \lambda|\beta_j|, \tag{6}$$

where $\hat{\mathbf{r}}_j \doteq \varphi(\mathbf{y}) - \sum_{i=1,i\neq j}^{n} \hat{\beta}_i \varphi(\mathbf{x}_i)$ is the *partial residual* [13] for fitting β_j. Considering that $\varphi(\mathbf{x}_j)^T\varphi(\mathbf{x}_j) = 1$, Eq.(6) is further equivalent to

$$J_j(\beta_j) \doteq \frac{1}{2}\beta_j^2 - \hat{\alpha}_j\beta_j + \lambda|\beta_j|, \tag{7}$$

where $\hat{\alpha}_j \doteq \varphi(\mathbf{x}_j)^T\hat{\mathbf{r}}_j$ is the residual correlation. If $\beta_j \neq 0$, by taking $\frac{\partial J_j(\beta_j)}{\partial \beta_j} = 0$ in Eq.(7), we get the minimizers as $\hat{\alpha}_j - \lambda$ for $\beta_j > 0$, and $\hat{\alpha}_j + \lambda$ for $\beta_j < 0$. The corresponding conditions are $\hat{\alpha}_j > \lambda$ and $\hat{\alpha}_j < -\lambda$, respectively. Otherwise, if $|\hat{\alpha}_j| \leq \lambda$, we have

$$J_j(\beta_j) \geq \frac{1}{2}\beta_j^2 - \hat{\alpha}_j\beta_j + |\hat{\alpha}_j\beta_j| \geq 0. \tag{8}$$

The equality holds if and only if $\beta_j = 0$. Consequently, the coordinate-wise update of β_j is

$$\beta_j \leftarrow \beta_j' \doteq \begin{cases} \hat{\alpha}_j - \lambda, & \text{if } \hat{\alpha}_j > \lambda, \\ \hat{\alpha}_j + \lambda, & \text{if } \hat{\alpha}_j < -\lambda, \\ 0, & \text{if } |\hat{\alpha}_j| \leq \lambda. \end{cases} \tag{9}$$

This update, denoted as $\beta_j \leftarrow \mathrm{sign}(\hat{\alpha}_j)(|\hat{\alpha}_j| - \lambda)_+$, is the well-known soft-thresholding shrinkage operation [26, 2].

Now that

$$\hat{\alpha}_j = \varphi(\mathbf{x}_j)^T\hat{\mathbf{r}}_j = \varphi(\mathbf{x}_j)^T \left[\varphi(\mathbf{y}) - \sum\nolimits_{i=1,i\neq j}^{n} \hat{\beta}_i \varphi(\mathbf{x}_i)\right], \tag{10}$$

it can be rewritten as

$$\hat{\alpha} = K(\mathbf{x}_j, \mathbf{y}) - \sum\nolimits_{i=1,i\neq j}^{n} \widehat{\beta}_i K(\mathbf{x}_j, \mathbf{x}_i), \tag{11}$$

where $K(\mathbf{x}, \mathbf{y}) \doteq \varphi(\mathbf{x})^T\varphi(\mathbf{y})$ is the kernel function and $K(\mathbf{x}, \mathbf{x}) \doteq \varphi(\mathbf{x})^T\varphi(\mathbf{x}) = 1$. This is a kernel extension of the *covariance update* idea suggested in [13]. Therefore, providing a kernel function K, the above proposed KCD algorithm is able to update $\boldsymbol{\beta}$ iteratively in kernel space by Eq.(9) and Eq.(11).

For classification, we developed the corresponding kernel SRC criterion as follows

$$\begin{aligned} \text{identity} &= arg \min_{c} \ \left\| \sum\nolimits_{l(i)=c} \beta_i \varphi(\mathbf{x}_i) - \varphi(\mathbf{y}) \right\|_2^2 \\ &= arg \min_{c} \ \delta_c(\boldsymbol{\beta})^T \mathbf{R} \delta_c(\boldsymbol{\beta}) - 2\mathbf{z}^T \delta_c(\boldsymbol{\beta}), \end{aligned} \tag{12}$$

where $\mathbf{R} \doteq (K(\mathbf{x}_i, \mathbf{x}_j))_{n \times n}$ is the training kernel matrix, $\mathbf{z} \doteq (K(\mathbf{x}_i, \mathbf{y}))_{n \times 1}$ contains correlation coefficients in the kernel space, and $l(i)$ is the class label of the i-th sample. The resulting algorithm, named KCD-SRC, is summarized in Algorithm 1.

```
Algorithm 1: KCD-SRC
Data: Kernel matrix $\mathbf{R}$, correlation vector $\mathbf{z}$, and Lasso parameter $\lambda \geq 0$.
Result: $\hat{\boldsymbol{\beta}}$ for Eq.(5).
Init: $\hat{\boldsymbol{\beta}} = \mathbf{0}$.
Sparse Coding Stage:
while not converged do
    for $j = 1$ to $n$ do
        $\hat{\alpha}_j \leftarrow z_j - \sum_{i=1, i \neq j}^{n} \hat{\beta}_i \mathbf{R}_{ji}$;
        $\hat{\beta}_j \leftarrow \text{sign}(\hat{\alpha}_j)(|\hat{\alpha}_j| - \lambda)_+$;
    end
end
Classification Stage: identity = $arg \min_c \delta_c(\boldsymbol{\beta})^T \mathbf{R} \delta_c(\boldsymbol{\beta}) - 2\mathbf{z}^T \delta_c(\boldsymbol{\beta})$.
```

3. LBP Based Kernels

3.1. Region-Level Kernels

Our region-level kernels are based on the well-known LBP features [6]. LBP is an effective local image descriptor, and has been successfully applied to face recognition [7]. The original LBP is an 8-bits binary string or a corresponding decimal label extracted from the local neighboring pixel comparisons, as shown in Fig.2. Uniform LBP [27] is an extension of the original LBP that only counts LBP patterns with no more than two 0-1 or

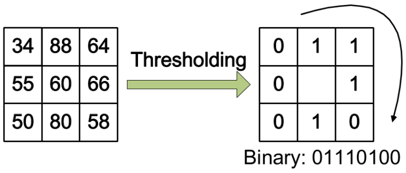

Figure 2: LBP operator.

1-0 transitions in the circular encoding string. This results in 58 uniform LBP patterns, and all other patterns are considered in an additional group. Various works [6, 7, 5] have shown that LBP features are discriminative and robust against illumination variations.

In this work, we develop several region-level image kernels based on the above LBP histogram representation. The process of the LBP histogram representation is demonstrated in Fig.3. First, the face image is divided into non-overlapping subregions of 8×8 pixels. Then the uniform LBP histogram (59 bins) is computed within each subregion and all resulting histograms are concatenated to form the final representation.

We utilize the χ^2 [6, 7] and the HI [7] kernels for the LBP histogram representation, which are defined as

$$K_{\chi^2}(\mathbf{a}, \mathbf{b}) = \sum\nolimits_{i=1}^{L} \frac{2a_i b_i}{a_i + b_i}, \tag{13}$$

$$K_{HI}(\mathbf{a}, \mathbf{b}) = \sum\nolimits_{i=1}^{L} \min(a_i, b_i), \tag{14}$$

where $\mathbf{a}$ and $\mathbf{b}$ ($\sum_{i=1}^{L} a_i = \sum_{i=1}^{L} b_i = 1$) are two normalized LBP histograms with L histogram bins ($L = 59$ in this paper). It can be easily verified that $0 \leq K_{\chi^2} \leq 1$, and $0 \leq K_{HI} \leq 1$. What's more, $\forall \mathbf{a}$, $K_{\chi^2}(\mathbf{a}, \mathbf{a}) = 1$, and $K_{HI}(\mathbf{a}, \mathbf{a}) = 1$, which are required by our KCD-SRC algorithm (see Eq.(7)). To compare with the LBP based linear SRC method proposed in [5], we also apply the linear kernel with the same LBP histogram representation in the KCD-SRC framework, which is defined as

$$K_{linear}(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T \mathbf{b}, \tag{15}$$

where all feature vectors should be normalized to unit length with the ℓ_2 norm, so that $\forall \mathbf{a}$, $K_{linear}(\mathbf{a}, \mathbf{a}) = 1$.

3.2. Pixel-Level Kernel

While LBP histogram representation based kernels are robust to illumination changes and misalignment to some extent [5], these region-level kernels lose spatial information

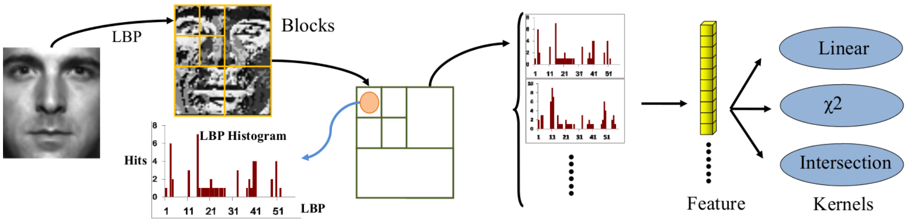

Figure 3: Process of the LBP histogram representation for the linear, χ^2, and HI kernels.

about local image details. Therefore, LBP histogram based kernels may be limited in fine discrimination of similar faces. In contrast, we propose another kernel working in pixel-level, named as half LBP based hamming kernel (LBP$_h$-K_H).

Before introducing the LBP$_h$-K_H, we give a brief review of the LBP-K_H proposed in our previous work [9]. As we have said, LBP is actually a binary string, and the corresponding decimal number is meaningless other than a label. Thus, if the image pixels are affected by noise or other image degradation, the corresponding decimal number may change extremely and irregularly, which makes the region-level histogram representation unstable. Therefore, the well-known Hamming distance [28] metric in information theory has been introduced to measure the dissimilarity between two LBP encoding strings directly. Applying the Hamming distance to LBP binary strings, we define the following Hamming Kernel (K_H) to measure the similarity between two LBP binary strings,

$$K_H(\mathbf{s}', \mathbf{s}'') = 1 - \frac{1}{N} \sum\nolimits_{i=1}^{N} s_i' \oplus s_i'' = \frac{1}{N} \sum\nolimits_{i=1}^{N} s_i' \odot s_i'', \tag{16}$$

where $\mathbf{s}'$ and $\mathbf{s}''$ are two LBP binary strings, $\oplus$ is the Exclusive Or (XOR) operator, $\odot$ denotes the Exclusive NOR (XNOR) operator, and N is the number of the defined LBP neighbors. For example, as in Fig.2, LBP$_{8,1}$ code is computed from 8 neighboring points. In this way, each pixel is represented as an 8-bits binary string. However, the pixelwise LBP-K_H computes redundant comparisons (see details below). Therefore, instead of applying the original LBP encodings of two images to the K_H kernel, we propose to use half of the LBP (LBP$_h$) encodings for the K_H kernel in this paper.

In the original LBP, each pixel is compared with the 8 neighboring pixels (see Fig.2). As a result, each pixelwise comparison is computed twice in LBP, thus half of the comparisons are redundant in the whole LBP encodings. Such as in Fig.4(a), where the

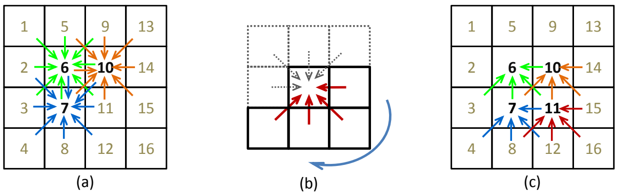

Figure 4: (a) Pixels and their relationships in LBP encodings. It shows that the LBP encoding of pixel 10 recomputes the relationship with pixel 6 and 7, respectively. In other words, the relationship between pixel 6 and pixel 10, pixel 6 and pixel 7, as well as pixel 7 and pixel 10 have been counted twice. (b)Half LBP encoding (LBP$_h$), which are shown in solid lines. (c) Pixels and their relationships in LBP$_h$ encodings. In this subfigure the redundant comparisons are got rid of by using LBP$_h$.

relationships between pixel 6 and pixel 7, pixel 6 and pixel 10, as well as pixel 7 and pixel 10 are computed twice. In contrast, LBP$_h$ only calculates pixelwise comparisons with the nearest neighbors in 4 orientations, as the solid lines in Fig.4(b). Therefore, there is no redundant computation in LBP$_h$-K_H but rich local image details are remained. This is clearly displayed in Fig.4(c), where all neighboring relationships are compared and no redundant comparison exists. Note that the LBP binary string based K_H kernel (both of the LBP-K_H and the LBP$_h$-K_H) ranges in [0,1], and $K_H(\mathbf{a}, \mathbf{a}) = 1$, where $\mathbf{a}$ is a LBP binary string.

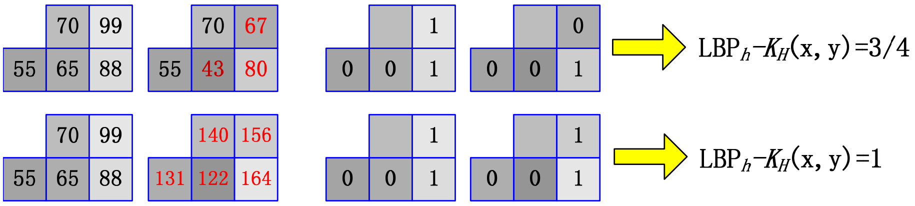

Figure 5: LBP$_h$-K_H with additive image noise (top) and illumination variations (bottom). Left: image patches. Right: LBP$_h$-K_H values at the center pixel.

Furthermore, LBP$_h$ is computationally more efficient than LBP because redundant computation in LBP is avoided in LBP$_h$ and saves half of the storage memory in the results. It is also efficient for the K_H calculation since the feature length of LBP$_h$ is also half of the LBP's. Fig.5 demonstrates the robustness of LBP$_h$-K_H, where the original image has been changed by noise and almost doubled illumination intensity, but the LBP$_h$-K_H values remain the same or not much affected. Besides, the proposed LBP$_h$-

K_H is equivalent to the linear kernel when the linear kernel is applied to a modified LBP encoding where all 0's in the original LBP encoding are replaced by -1's (details in Supplementary Part 2).

4. Experiments

In this section, a series of experiments are conducted on the Extended YaleB [29], the CMU-PIE [30] and the AR [31] face databases to verify the effectiveness of the proposed method. The experiments are designed in four scenarios: i) with illumination variations; ii) with different levels of simulated noises; iii) with synthesized occlusions and real occlusions; and iv) with registration errors. For all experiments we randomly separated gallery images and probe images from a database for face identification, and the rank-1 recognition rate is used as the performance measure. We repeated each evaluation for ten times to obtain the mean recognition rates and the corresponding standard deviations.

The KCD-SRC algorithm with local image feature kernels proposed in Section 3 are tested for face recognition in all experiments. For convenience, we denote the two proposed kernels based on LBP histogram features by "LBP-Hist-K_{χ^2}" (Eq.(13)) and "LBP-Hist-K_{HI}" (Eq.(14)), respectively. Note that the proposed KCD-SRC algorithm with linear kernel (Eq.(15)) is actually equivalent to the original SRC algorithm with the coordinate descent solver for the ℓ_1 minimization problem. Thus we also implement two other existing methods in this setting for comparison. One is the SRC algorithm with LBP histogram features developed in [5] (denoted by "LBP-Hist"), and the other is SRC with raw pixels proposed in [1] (denoted by "Pixel" for convenience). For all experiments, the parameter λ is consistently set to be 0.01 and the maximum iteration number is 100.

4.1. Database

The Extended YaleB [29] Database consists of 16,128 face images of 38 subjects under 9 poses and 64 illumination conditions. In the experiments, we selected 2,414 frontal face images of the 38 subjects under 64 illumination conditions. The original cropped face images provided are 192 × 168 pixels. We resize all the cropped images to

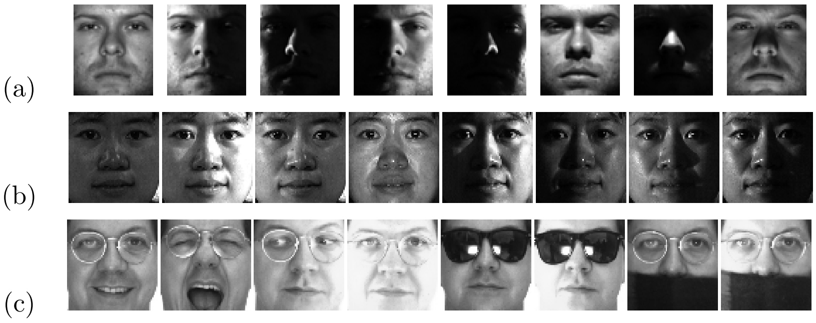

Figure 6: Cropped face examples in (a) the Extended YaleB database, (b) the CMU-PIE database, and (c) the AR database.

64 × 56 pixels for our usage. Fig.6 (a) shows some examples of cropped faces from the Extended YaleB database.

The CMU-PIE [30] Database includes 41,368 face images of 68 subjects under 13 different poses, 43 different illumination conditions, and 4 different expressions. We selected 2,989 images from the database where the two eye coordinates are available. These images are almost frontal images with different expressions and illumination variations. We cropped all the 2,989 images according to the two eye coordinates and resized them to 64 × 64 pixels in the experiments. Some cropped face examples from the CMU-PIE database are demonstrated in Fig.6 (b).

The AR [31] Database contains more than 4,000 color images of 126 subjects (70 men and 56 women). They are all frontal faces with different facial expressions, illumination conditions, and occlusions (sunglasses and scarf). Each person participated in two sessions, separated by two weeks (14 days). In each session, there are 4 images with different facial expressions under normal illumination, 3 images under different illumination conditions (left, right, and both-side lighted) with normal expression, 3 images with sunglasses under different illuminations, and 3 images with scarf under different illuminations. Images with the same conditions were taken in both sessions, resulting in 26 images per subject. All color images were first converted to gray scale, and then cropped and resized to 64 × 64 pixels according to the two eye coordinates. We got the eye coordinates of the AR face database from [32]. Fig.6 (c) shows some examples of cropped faces from the AR database.

4.2. With Illumination Variations

In this section, we perform experiments with illumination variations on the Extended YaleB and the CMU-PIE face databases, respectively. The Extended YaleB frontal face images are taken under 64 different directions of the light source for each person. As for the CMU-PIE database, there are 43 frontal facial images in different illuminations for each person. For each individual in the Extended YaleB database, we randomly chose 5, 10, 15, 20, 25, 30 images, respectively, as gallery, and the rest as probe (with various illumination variations). The gallery is used as the training set for SRC based methods. The random partitions were repeated for 10 times. As for the CMU-PIE database, we randomly chose 5, 10, 15, 20, 25 images, respectively, as gallery and the rest as probe. The procedure was also repeated for 10 times. The average recognition rates with respect to the number of training samples are shown in Fig.7 for the Extended YaleB database and in Fig.8 for the CMU-PIE database, where the 5 algorithms (Pixel, LBP-Hist, LBP-Hist-K_{χ^2}, LBP-Hist-K_{HI}, and LBP$_h$-K_H) under the KCD-SRC framework are compared. The standard deviations are also displayed along with the corresponding average recognition rates.

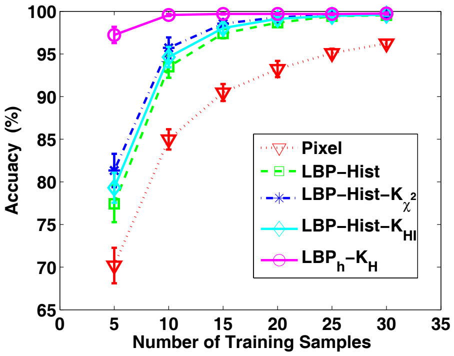

Figure 7: Average face recognition rates w.r.t. the number of training samples on the Extended YaleB database.

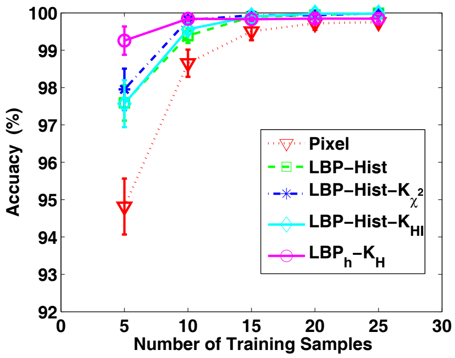

Figure 8: Average face recognition rates w.r.t. the number of training samples on the CMU-PIE database.

From the results shown in Fig.7 and Fig.8, we summarize the observations as follows. First, the proposed LBP$_h$-K_H algorithm performs the best with different number of training samples, followed by LBP-Hist-K_{χ^2}, LBP-Hist-K_{HI}, LBP-Hist, and Pixel.

Second, with the decreasing amount of training samples per subject, the performances of all algorithms generally drop. However, LBP$_h$-K_H is the most robust one, with small performance degradation. This is because the LBP binary string encodes intrinsic structure of individual face images regardless of the wide range of illumination variations, and the KCD-SRC framework successfully makes use of this advantage in the LBP$_h$-K_H kernel, despite the small training sample size. It is impressive that under only 5 training samples per subject, the recognition rate of LBP$_h$-K_H on Extended YaleB can still reach over 97%, while that of the Pixel is only 70%, with 27% improvement.

Furthermore, as also observed in [5], LBP based methods in the SRC framework perform better than the pixel based method under illumination variations, since the LBP feature is known to be robust to illumination changes. In experiments on both the Extended YaleB and the CMU-PIE database, the LBP-Hist-K_{χ^2} performs slightly better than LBP-Hist-K_{HI} and LBP-Hist. Interestingly, the recognition rates on the Extended YaleB database are not as good as that on the CMU-PIE database. The reason is probably to be relatively stronger illumination variations in the Extended YaleB database than the CMU-PIE database.

4.3. With Noise Corruption

Next, we test the robustness of the proposed methods with noise corrupted images on the Extended YaleB and CMU-PIE databases. In all the following experiments, we fixed the number of randomly chosen training samples per subject to be five, a relatively small training sample size for challenge. All other remaining images were used as probe, and the same procedure was repeated for 10 times. For each image from the probe set, we randomly corrupted a portion of the pixels with noise. The noise corruption levels (the proportion of the corrupted pixels) were from 10% to 80%, respectively, with a step of 10%. These chosen pixels' values were replaced by independently and identically distributed samples from a uniform distribution in [0, 255]. The corrupted pixels were randomly chosen for every image, and the corrupted locations were unknown to any algorithm. Some examples of different levels of noise corrupted images are shown in Fig.9 (a). Note that all images in the gallery were not corrupted.

We compared the five methods (Pixel, LBP-Hist, LBP-Hist-K_{HI}, LBP-Hist-K_{χ^2}, and LBP$_h$-K_H) under the KCD-SRC framework on both the Extended YaleB and the CMU-

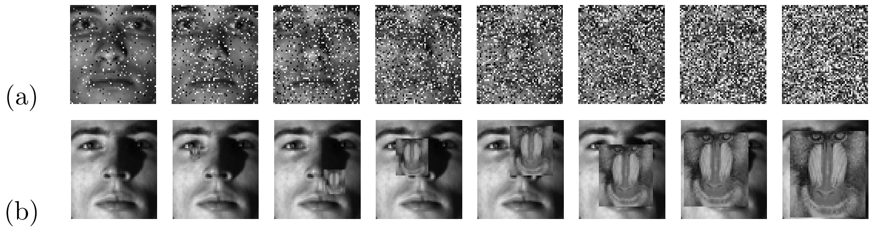

Figure 9: (a) Images corrupted with 8 levels (10% to 80%) of random noises. (b) Images corrupted with 7 levels of synthesized continuous occlusions, with the original image displayed in the left.

PIE face databases. The mean recognition rates as well as the corresponding standard deviations are shown in Fig.10 (Extended YaleB) and Fig.11 (CMU-PIE). As expected, with the increasing amount of noises, the performances of all algorithms drop. Yet again, LBP$_h$-K_H is the best performer over all levels of noise corruptions. In the Extended YaleB database, LBP$_h$-K_H consistently outperforms Pixel, the second best one, by over 20% of recognition rates under 10% to 70% of noise corruptions.

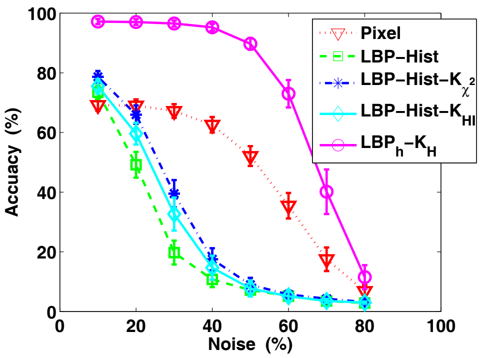

Figure 10: Average face recognition rates w.r.t. the level of noise corruptions on the Extended YaleB database.

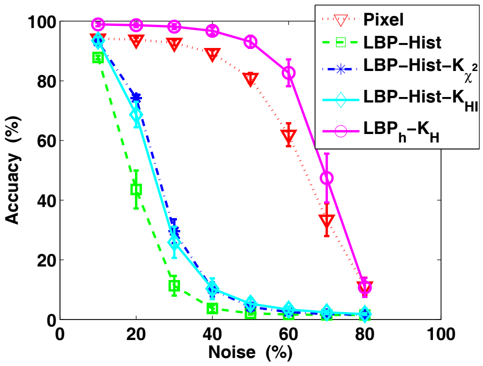

Figure 11: Average face recognition rates w.r.t. the level of noise corruptions on the CMU-PIE database.

We see that the performances of all LBP histogram based methods drop drastically with the increasing amount of noises. The results imply that while LBP histogram feature is robust against illumination variations, it is sensitive to image noise. This may be mainly due to the fact that local noise might change the neighboring comparison results (see Fig.2) in some orientations, and so the affected LBP codes could disturb the histogram distribution over LBP bins due to the discrete nature of LBP. To illustrate this, we draw an example in Fig.12 showing local LBP histograms of a face image with and without noise. Clearly, the distribution of LBP codes is severely affected by noise. In contrast, thanks to the local binary feature representation and the Hamming distance measure which is tolerable to some flipping bits (see Fig.5), the LBP$_h$-K_H is more robust to image noise. In the same example of Fig.12, similarities of two regions with LBP$_h$-K_H (Eq.(16)) and χ^2 (Eq.(13)) are 0.78 and 0.61, respectively, which indicates that LBP$_h$-K_H is more robust than the χ^2 kernel (also the other two LBP histogram based kernels) against noise.

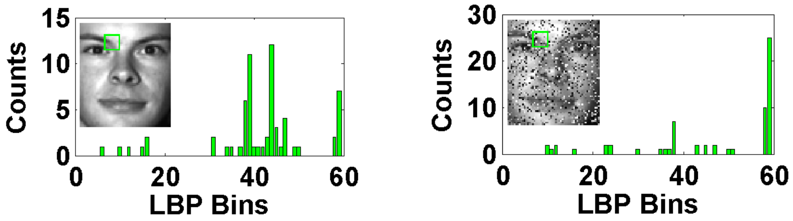

Figure 12: An example of LBP histograms in a 8×8 local region of the same face without noise (left) and with 30% of noise (right).

Despite the poor performances of all LBP histogram based methods under image noise, it looks that LBP-Hist-K_{χ^2} performs slightly better than LBP-Hist-K_{HI}, and both of them are better than LBP-Hist. For example, under 20% and 30% of noises, LBP-Hist-K_{χ^2} outperforms LBP-Hist by about 20% of recognition rates on both the Extended YaleB and CMU-PIE database. These improvements indicate that, when integrating LBP histogram into the SRC framework, using proper nonlinear kernels (like χ^2 and HI) is more robust than using the linear kernel in measuring similarity between possibly corrupted histograms, and so kernel SRC is required in this situation.

4.4. With Continuous Occlusions

Next, we will demonstrate experiments with continuous occlusions on facial images. We conducted experiments in two scenarios: different levels of synthesized occlusions in the Extended YaleB and CMU-PIE database, and real occlusions with sunglasses and scarf in the AR database.

4.4.1. Synthesized Occlusions

First, we conducted experiments with randomly synthesized continuous occlusions. Like in [1], we chose an irrelative picture, resized it to a fixed size and attached it on a random position of the test image. We tested 7 levels of occlusions, with the size of the attached picture corresponding to 8×7, 16×14, $\dots$, 56×49 for the Extended YaleB database (whole face size is 64×56) and 8×8, 16×16, $\dots$, 56×56 for the CMU-PIE database (whole face size is 64×64). Fig.9 (b) shows some examples of these 7 levels of occlusions from the Extended YaleB database. We also evaluated the five algorithms in this scenario, with randomly selected uncorrupted five images per subject for training.

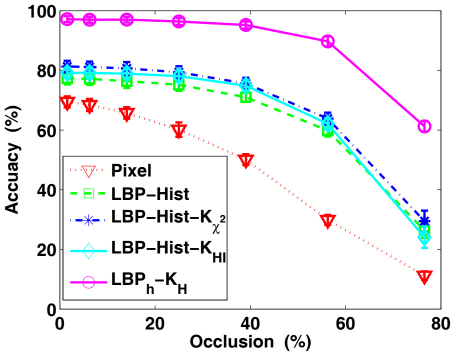

Figure 13: Face recognition performances with 7 levels of occlusions on the Extended YaleB database.

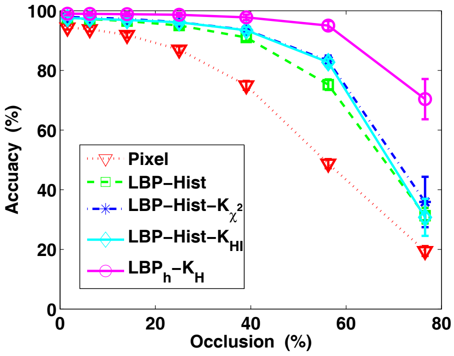

Figure 14: Face recognition performances with 7 levels of occlusions on the CMU-PIE database.

Fig.13 (Extended YaleB) and Fig.14 (CMU-PIE) demonstrate the mean recognition rates and their corresponding standard deviations under different levels of occlusions. Obviously, the performances of all algorithms drop with the increasing amount of occlusion. Similar as in subsection 4.2, LBP$_h$-K_H consistently performs the best under all levels of occlusions, followed by LBP-Hist-K_{χ^2}, LBP-Hist-K_{HI}, LBP-Hist, and Pixel. Notice that LBP$_h$-K_H maintains over 90% recognition rate with up to 56% of the face region occluded on both the Extended YaleB and CMU-PIE databases. LBP$_h$-K_H outperforms the second best algorithm, LBP-Hist-K_{χ^2}, by roughly 20% on the Extended YaleB database. Besides, it can also be observed that the use of local image features in kernel SRC is superior than raw pixel values. Furthermore, regarding the same LBP

Table 1: Face recognition performances with real occlusions on the AR face database (%).

	Raw Pixels	LBP-Hist	LBP-Hist-K_{χ^2}	LBP-Hist-K_{HI}	LBP$_h$-K_H
Sunglasses	13.53±1.34	53.90±1.91	51.48±1.93	49.15±1.25	29.43±1.43
Scarf	11.38±0.97	84.27±1.22	88.17±1.86	89.30±1.05	93.03±0.95

histogram features, the proposed KCD-SRC with χ^2 kernel slightly outperforms that with histogram intersection kernel and linear kernel.

4.4.2. Real Occlusions

We also conducted experiments with real occlusion on the AR face database [31]. We selected 100 subjects who participated both the two image capturing sessions, so that each selected subject has 26 images. In the selected images, each subject has 6 face images wearing sunglasses and 6 face images with scarf occlusion, which were used as the probe set. Moreover, we partitioned the occluded images into two sub sets, sunglasses set and scarf set, and tested them separately. There were 14 images left for each individual, including 8 with facial expressions and 6 with illumination variations. We randomly selected 5 from the remaining 14 images for training, and repeated 10 times to obtain the mean recognition rates.

The recognition performances are shown in Table 1. Clearly, our algorithms perform much better on the scarf subset than on the sunglasses subset. For the scarf subset, the LBP$_h$-K_H algorithm is still the best one (93.03%), followed by LBP-Hist-K_{HI} (89.30%), LBP-Hist-K_{χ^2} (88.17%), LBP-Hist (84.27%), and Pixel (11.38%). However, compared to the scarf subset, in the sunglasses subset the performances of all algorithms drop except Pixel. With sunglasses, the kernels based on LBP histogram features are the most robust, followed by the LBP$_h$-K_H method, and Pixel is still the worst one. Note that in our evaluation on the AR database, the SRC method with raw pixel values performs not as good as what has been reported in [1]. This is because i) we cropped faces more tightly in 64×64 pixels, where the forehead region and the face counter were excluded; ii) in real occlusion experiments, images with illumination variations were not considered in [1]; and iii) we used only 5 training samples randomly selected from the 14 non-occluded images, while [1] selected 8 non-occluded images per subject for training.

One reason why our algorithms perform worse with sunglasses than with scarf may be that the ocular region contains the most important features for distinguishing human beings. However, with sunglasses the eyes' coordinates might be labeled not very accurate compared to the scarf subset. Fig.15 shows some of the cropped AR face images. It can be observed that the cropped AR face images are not well aligned, especially for those with sunglasses. There are some translation, rotation, and scale variations between probe and gallery images. Therefore, the imprecise eye coordinates might generate disturbance on cropping face and finally affect the performances of all algorithms, especially for LBP$_h$-K_H, which belongs to pixel-level. This phenomena is also known as registration error, and will be further studied in the next subsection.

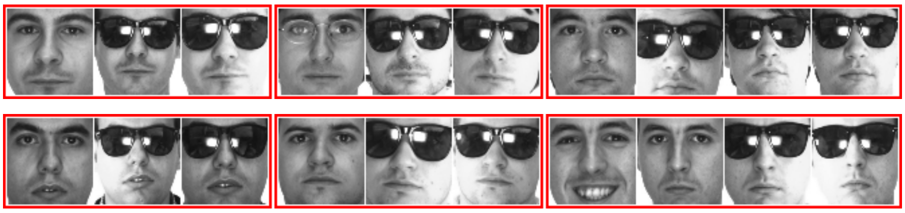

Figure 15: Cropped AR facial images.

4.5. Registration Error

In this subsection, we conducted experiments on the CMU-PIE database under registration errors. Registration error often happens in realistic applications, where some facial images to be recognized might not be aligned precisely. This has been studied in [5] by comparing SRC based methods. We give an independent study in this paper, where two scenarios of registration errors were evaluated. One is with the synthesized registration errors, where we added random disturbances to the two eye coordinates, and the other one is with all facial images automatically detected by Viola-Jones face detector implemented in OpenCV, without further alignment to simulate real registration errors.

4.5.1. Synthesized Registration Error

In this experiment, we also randomly selected five images per subject in the gallery set, with the remaining images being the probe set, and run 10 trails of this random partition. This procedure is the same as in subsection 4.2, and we also kept the 10 random partitions of the CMU-PIE database the same as in subsection 4.2 for comparison. For each trail, the gallery face images were normally cropped as before, while each probe face image was cropped according to disturbed eye coordinates to synthesize a registration error. For each of the two eyes of a probe face, each of the two (x,y) coordinates was independently disturbed by a Gaussian distribution of zero mean and standard deviation of 2.0. Some face image examples with these synthesized registration errors are displayed in Fig.16. The misaligned face images may cause problems for most face recognition algorithms.

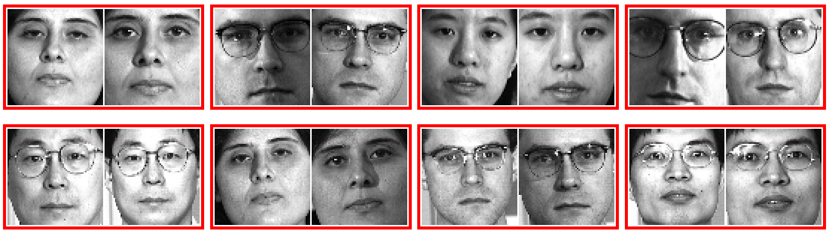

Figure 16: Face images from the CMU-PIE database with synthesized registration errors (top) and automatic detections (bottom).

4.5.2. Automatically Detected Faces

Instead of cropping face images according to manually labeled eye coordinates, in this experiment we used the Viola-Jones face detector implemented in OpenCV 2.2 to detect all face images automatically, and used the detected and cropped face images directly, without further alignment, for face recognition. Some examples of the OpenCV detected and cropped face images from the CMU-PIE database are shown in Fig.16. It can be seen that the automatically detected and cropped face images are not consistently aligned. For face recognition, all other settings are the same as in subsection 4.2.

Fig.17 shows the mean recognition rates, where the same face images in three scenarios are compared: i) original images; ii) synthesized registration errors; and iii) automatically detected faces. The first group of results in Fig.17 are with the original CMU-PIE images, which are identical to the corresponding experiments in subsection 4.2 on the CMU-PIE database with 5 images per subject for training. The second and the third groups are results of synthesized registration errors and automatically detected face images on the CMU-PIE database, respectively. As we can see from Fig.17, kernels based on LBP histogram features are more robust against registration errors. LBP-Hist-K_{χ^2} performs

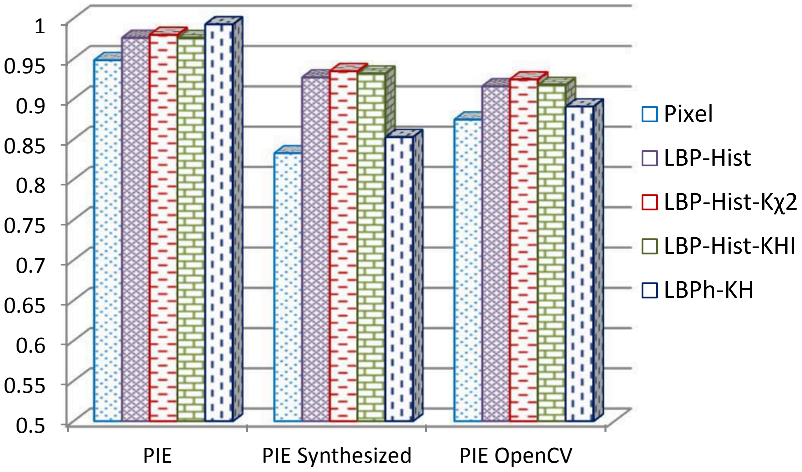

Figure 17: Average recognition rates on the CMU-PIE database with original images, synthesized registration errors and OpenCV detected images.

the best, with a small gap above LBP-Hist-K_{HI}, followed by LBP-Hist, LBP$_h$-K_H and Pixel. As also discussed in [5], the SRC based methods generally degrade with registration errors. In our experiments, the recognition rates of the five methods all drop with both kinds of registration errors. Since the LBP histogram representation extracts region-level characteristics over local subregions, it is more robust against misalignment. In contrast, LBP$_h$-K_H works in pixel-level; it performs less stable than the LBP histogram based kernels under registration errors (about 5% lower in average). Yet LBP$_h$-K_H still slightly outperforms Pixel, another pixel-level kernel.

5. Discussion

In this paper, two kinds of kernel methods for face recognition are proposed under the KCD-SRC framework. One is region-level kernel based on the LBP histogram representation, including LBP-Hist-K_{HI}, LBP-Hist-K_{χ^2}, and the linear kernel LBP-Hist. Another one is the LBP$_h$-K_H based on half of LBP features. Extensive experiments show that LBP$_h$-K_H has the best overall performance, which surpasses the other kernel based methods and the original SRC with raw pixel values under illumination variations, noise corruptions, and occlusions, particularly with only five training samples per subject. While LBP-Hist-K_{χ^2} performs slightly better than LBP-Hist-K_{HI}, both of them are superior to the linear kernel LBP-Hist, which demonstrate the effectiveness of the kernel SRC. Under the registration error scenario, LBP histogram based kernels are the most robust, followed by LBP$_h$-K_H, and they all outperform the raw pixel based one. However, the LBP$_h$-K_H is not as good as LBP histogram based kernels under registration errors (about 5% lower). But considering the impressive performance of today's commercial softwares for automatic face alignment (such as PittPatt [33]), the proposed LBP$_h$-K_H with KCD-SRC method would still be more preferred in handling many kinds of practical facial image variations.

We have implemented the proposed KCD-SRC algorithm as well as several kernels with local image features in MATLAB. The source codes are available in http://www.cbsr.ia.ac.cn/users/scliao/projects/KCD-SRC.html. The proposed KCD-SRC framework for face recognition generally involves three steps: feature extraction, kernel computation, and KCD-SRC learning and classification. Table 2 lists the average computation time of recognizing a probe face image by the five evaluated kernel methods in our MATLAB implementation. The evaluation time is separated in feature extraction, kernel computation (the vector $\mathbf{z}$ in Eq. 12), and KCD-SRC procedures, respectively, and the overall computation time is also given in the last row of Table 2. All algorithms were run on a ThinkPad T420 laptop with Intel Core i7-2640M 2.80 GHz CPU and 4GB memory. In this evaluation the AR database was used, with 500 face images of 100 subjects (5 images per subject) enrolled in the gallery and 500 images being used as the probe set. The computation time is averaged over the whole probe set. Note that feature vectors of gallery images, as well as the training kernel matrix $\mathbf{R}$ in Eq.(12) can be pre-computed before recognition, therefore they are excluded in the reporting time.

From the first row of Table 2, we can see that feature extraction of LBP$_h$-K_H is significantly faster than that of LBP histograms. This is because i) LBP$_h$-K_H avoids redundant computations; ii) LBP$_h$-K_H does not need to convert binary string to decimal numbers; and iii) LBP$_h$-K_H does not need to divide images into blocks and compute histograms for each block. It is impressive that LBP$_h$-K_H only needs 0.57 ms for feature extraction, which is extremely fast. For the kernel computation, it is obvious that the linear kernel is the fastest, followed by LBP$_h$-K_H, LBP-Hist-K_{HI}, and LBP-Hist-K_{χ^2}. As for the kernel sparse learning and classification step, there is no notable difference

Table 2: Average computation time (ms) of recognizing a probe image by different kernel methods.

	Raw Pixels	LBP-Hist	LBP-Hist-K_{χ^2}	LBP-Hist-K_{HI}	LBP$_h$-K_H
Feature Extraction	-	348.18			0.57
Kernel Computation	1.94	1.72	90.85	73.70	68.21
KCD-SRC	736.98	738.46	734.05	736.06	725.36
Overall	738.92	1088.26	1173.08	1157.94	794.14

among all compared methods. From the last row of Table 2, it can be seen that the overall computation time of LBP$_h$-K_H is comparable to that of Pixel, the fastest method. Compared to Pixel, LBP$_h$-K_H requires about 0.06 second more time to recognize one probe image in average. Also notice that both LBP$_h$-K_H and Pixel are faster than all LBP histogram based methods, which require more than one second for the same task.

6. Conclusion

We have proposed a novel kernel coordinate descent (KCD) algorithm based on the covariance update technique for the kernel LASSO problem. The proposed KCD algorithm is applied to the sparse representation based classification framework, resulting in the KCD-SRC algorithm for face recognition. We have also developed an effective local image kernel called half LBP encoding based Hamming kernel (LBP$_h$-K_H) for face recognition. LBP histogram features with two kernels based on the χ^2 distance and Histogram Intersection distance are also suggested in the proposed approach. The linear kernels with raw pixel values and LBP histogram features are also embedded for comparison. We have evaluated the proposed approaches on the Extended YaleB, the CMU-PIE, and the AR face databases. Extensive evaluations under illumination variations, random noises, continuous occlusions, and registration errors show that the KCD-SRC approach successfully explores the potential discriminant power of local image features. Experimental results also show that, the two region-level LBP kernels, the χ^2 distance and the Histogram Intersection distance based LBP methods, are the most robust to registration error, but they are very sensitive to random noise. Alternatively, the proposed LBP$_h$-K_H based pixel-level kernel method performs slightly worse than the region-level kernels under the registration error situation, but it has the best overall performance among the five evaluated methods.

References

[1] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, Y. Ma, Robust Face Recognition via Sparse Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.

[2] R. Tibshirani, Regression shrinkage and selection via the Lasso, Journal of the Royal Statistical Society. Series B (Methodological) 58 (1996) 267–288.

[3] X.-T. Yuan, S. Yan, Visual classification with multi-task joint sparse representation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3493 –3500.

[4] R. Min, J.-L. Dugelay, Improved combination of LBP and sparse representation based classification (SRC) for face recognition, in: 2011 IEEE International Conference on Multimedia and Expo (ICME), 2011, pp. 1–6.

[5] C. H. Chan, J. Kittler, Sparse representation of (multiscale) histograms for face recognition robust to registration and illumination problems, in: IEEE International Conference on Image Processing, 2010, pp. 2441–2444.

[6] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognition 29 (1) (1996) 51–59.

[7] T. Ahonen, A. Hadid, M.Pietikäinen, Face Recognition with Local Binary Patterns, in: European Conference on Computer Vision, 2004, pp. 469–481.

[8] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: Application to face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (12) (2006) 2037–2041.

[9] C. Kang, S. Liao, S. Xiang, C. Pan, Kernel sparse representation with local patterns for face recognition, in: IEEE International Conference on Image Processing, 2011, pp. 3009 – 3012.

[10] S. Gao, I. W.-H. Tsang, L.-T. Chia, Kernel sparse representation for image classification and face recognition, in: Proceedings of the 11th European Conference on Computer Vision: Part IV, 2010, pp. 1–14.

[11] P. Tseng, On accelerated proximal gradient methods for convex-concave optimization, submitted to SIAM Journal on Optimization.

[12] H. Lee, A. Battle, R. Raina, A. Y. Ng, Efficient sparse coding algorithms, in: Proceedings of Neural Information Processing System, 2007, pp. 801–808.

[13] J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, Journal of Statistical Software 33 (1) (2009) 1–22.

[14] G. Wang, D. Yeung, F. Lochovsky, The kernel path in kernelized LASSO, in: International conference on artificial intelligence and statistics, 2007, pp. 580–587.

[15] V. Roth, The generalized LASSO, IEEE Transactions on Neural Networks 15 (1) (2004) 16–28.

[16] G. Bai, Y. Zhu, Z. Ding, A hierarchical face recognition method based on local binary pattern, in: Congress on Image and Signal Processing, Vol. 2, 2008, pp. 610–614.

[17] B. Yao, H. Ai, S. Lao, Matching texture units for face recognition, in: Proceedings of the IEEE International Conference on Image Processing, 2008, pp. 1920–1923.

[18] S. Z. Li, X. Hou, H. Zhang, Q. Cheng, Learning spatially localized, parts-based representation., in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, pp. 207–212.

[19] A. M. Martinez, Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (6) (2002) 748–763.

[20] J. Kim, J. Choi, J. Yi, M. Turk, Effective representation using ICA for face recognition robust to local distortion and partial occlusion, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (12) (2005) 1977–1981.

[21] H. J. Oh, K. M. Lee, S. U. Lee, Occlusion invariant face recognition using selective local non-negative matrix factorization basis images, Image and Vision Computing 26 (11) (2008) 1515–1523.

[22] K. Hotta, Robust face recognition under partial occlusion based on support vector machine with local gaussian summation kernel, Image and Vision Computing 26 (11) (2008) 1490 – 1498.

[23] H. Jia, A. Martinez, Support vector machines in face recognition with occlusions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 136–141.

[24] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, Y. Ma, Toward a practical face recognition system: Robust alignment and illumination by sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (2) (2012) 372–386.

[25] S. Liao, A. Jain, S. Li, Partial face recognition: Alignment-free approach, IEEE Transactions on Pattern Analysis and Machine Intelligence PP (99) (2012) PrePrint.

[26] D. Donoho, J. Johnstone, Ideal spatial adaptation by wavelet shrinkage, Biometrika 81 (3) (1994) 425–455.

[27] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.

[28] R. Hamming, Error detecting and error correcting codes, Bell System Technical Journal 29 (2) (1950) 147–160.

[29] K. Lee, J. Ho, D. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intelligence 27 (5) (2005) 684–698.

[30] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression (PIE) database, in: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, 2002.

[31] Martínez, Aleix and Benavente, R., The AR face database, Tech. Rep. 24, Computer Vision Center (1998).

[32] http://blues.ius.cs.cmu.edu/ralph/facelabels/.

[33] http://www.pittpatt.com/.