Local Laplacian Coding From Theoretical Analysis of Local Coding Schemes for Locally Linear Classification

Junbiao Pang, Lei Qin, Chunjie Zhang, Weigang Zhang, Qingming Huang, *Senior Member, IEEE*, and Baocai Yin

***Abstract*—Local coordinate coding (LCC) is a framework to approximate a Lipschitz smooth function by combining linear functions into a nonlinear one. For locally linear classification, LCC requires a coding scheme that heavily determines the nonlinear approximation ability, posing two main challenges: 1) the locality making faraway anchors have smaller influences on current data and 2) the flexibility balancing well between the reconstruction of current data and the locality. In this paper, we address the problem from the theoretical analysis of the simplest local coding schemes, i.e., local Gaussian coding and local student coding, and propose local Laplacian coding (LPC) to achieve the locality and the flexibility. We apply LPC into locally linear classifiers to solve diverse classification tasks. The comparable or exceeded performances of state-of-the-art methods demonstrate the effectiveness of the proposed method.**

***Index Terms*—Image classification, local coordinate coding (LCC), local Gaussian coding (LGC), local Laplacian coding (LPC), local student coding (LSC), locally linear classification, nonlinear approximation.**

Manuscript received February 9, 2015; revised March 31, 2015; accepted May 13, 2015. Date of publication June 3, 2015; date of current version November 13, 2015. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2012CB316400, in part by the Natural Science Foundation of China under Grant 61332016, Grant 61202234, Grant 61202322, Grant 61133003, Grant 61227004, Grant 61303154, Grant 61390510, and Grant 61472387, in part by the Beijing Natural Science Foundation under Grant 4132010 and Grant KZ201310005006, in part by the Beijing Post-Doctoral Research Foundation, and in part by the Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality. This paper was recommended by Associate Editor D. Tao.

J. Pang and B. Yin are with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China (e-mail: junbiao_pang@bjut.edu.cn; ybc@bjut.edu.cn).

L. Qin is with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: lqin@jdl.ac.cn).

C. Zhang is with the School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: cjzhang@jdl.ac.cn).

W. Zhang is with the School of Computer Science and Technology, Harbin Institute of Technology at Weihai, Weihai 264209, China (e-mail: wgzhang@jdl.ac.cn).

Q. Huang is with the College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China, and the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: qmhuang@jdl.ac.cn).

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

Digital Object Identifier 10.1109/TCYB.2015.2433926

I. Introduction

LOCAL coordinate coding (LCC) is a general framework to approximate any nonlinear function with linear ones, requiring a set of anchor points (or called as anchors) to build local coordinates [44]. Compared with other nonlinear models, such as kernel machines [6], the computation cost of LCC is proportional to the number of anchors. Meanwhile, the accuracy of LCC is determined by coding schemes in local coordinates. Due to the balance between the effectiveness and the efficiency—achieving the nonlinear ability yet a lower computation cost, LCC has been successfully applied to visual feature learning [35], [43] or locally linear classifiers [15].

Given a large size of dataset and a sample to be encoded, LCC needs both a coding scheme and a set of anchors to locally reconstruct this sample. Rather than the overcomplete bases in spare coding [7], [14], the number of anchors is usually small [37] in locally linear classification in terms of balancing between accuracy and speed.

In recent research, some local coding schemes (LCSs) [25] adopt the "explicit" locality: the reconstruction coefficients (local codings) are defined by special decay functions (see Fig. 1). As theoretically discovered in [44], the nonlinear approximation ability of LCC is bounded by both reconstruction and locality errors. Therefore, explicit coding scheme faces two problems: which one of the definitions of locality is better than the others, and how to balance between reconstruction and locality errors. For instance, some of these coding schemes try to explain—to some degrees—the locality from the empirical results [35], [36], [43], [48].

The motivation of our solution is that the theoretical analysis of locality contributes to two aspects: compare different coding schemes, and further help understand the nature of locality. The upperbound is a natural choice in comparing locality errors of different LCSs. However, obtaining comparable upperbounds is a challenge problem.

In this paper, we first seek the comparable upperbounds, and use these theoretical results to discover an efficient local Laplacian coding (LPC). Rather than predefining decay functions in LCSs, LPC only requires that if two anchors are similar, the local codings from these anchors are also similar. This implicit coding scheme grants more freedom to minimize the reconstruction error. This paper is an extension of [25]. We extend this paper in the following two aspects: 1) first of all, we amend the theoretical results about LCSs, and extend them into a general scenario, i.e., the impact of negative local codings and

2168-2267 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

2) we propose implicit coding scheme, and present this paper more completely, including the optimization of local codings for LPC, etc. To the best of our knowledge, this paper is the first to investigate Laplacian matrix for LCC, presenting a comprehensive series of experiments to illustrate the benefits of this implicit scheme for locally linear classification.

The rest of this paper is organized as follows. Section II provides an in-depth review of LCC, and discuss its connections to sparse coding and locally linear classification. In Section III, we first discuss the theoretical aspects of LCSs, and then discover the drawback of the explicit coding scheme. Section IV presents the details of the proposed coding scheme, and discusses its pros and cons. In Section V, extensive experiments are carried out to compare the proposed method with the state-of-the-art ones.

II. Background and Related Work

A. Sparse Coding, LCC, and Locally Linear Classification

The seemingly most similar work to LCC may be sparse coding [7]: adding different constraints into a reconstruction loss. The goal of sparse coding is to represent an input signal approximately as a globally linear combination of an overcomplete dictionary [14], [37], which is often learned with ℓ_1 norm [17]. The locality in LCC brings sparsity into coding coefficients, since only the anchors closing to the input would be given more weights [44]. In contrast, dictionary of sparse coding selected from the whole dataset does not favor this choice.

LCC aims at achieving nonlinear ability by merging local structures into the globally nonlinear consistency [2], [44]. Instead of reconstructing an input signal with the number of bases as small as possible, LCC encodes a sample with anchors as locally as possible. There are two important threads of the applications of LCC. One is the feature learning, a work using the nonlinear representation ability [35], and the other is incorporating nonlinear ability into classification [15], [34].

In the former case, local codings are combined with other hand-crafted features [35], [43], [50]. For instance, a variation of LCC [35] is combined with spatial pyramid matching (SPM) [16] and max pooling [37]. Recent experiments have observed that the number of anchors should be large enough to characterize the features distribution in the whole feature space well [35]. "We do find that using a larger codebook helps the performance from what we have tried."[1] Therefore, feature learning requires a coding scheme that performs well when a large number of anchors are available.

The other is locally linear classification based on LCC framework. Let $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ be a training set, where $\mathbf{x}_i \in \mathbb{R}^D$ denote the ith sample, $y_i \in \{+1, -1\}$ denote the binary label for a given object category, and N is the number of samples. Local linear support vector machine (LLSVM) [15] combines a set of linear ones $f_m(\mathbf{x})$

$$F(\mathbf{x}) = \sum_{m=1}^{M} \gamma_m(\mathbf{x}) \mathbf{w}_m^T \mathbf{x} + \sum_{m=1}^{M} \gamma_m(\mathbf{x}) b_m = \gamma(\mathbf{x})^T \mathbf{W}\mathbf{x} + \gamma(\mathbf{x})^T \mathbf{b} \quad (1)$$

[1] http://www.ifp.illinois.edu/~jyang29/LLC.htm

where $\gamma_m(\mathbf{x})$ is a local coding for the linear classifier $f_m(\mathbf{x}) = \mathbf{w}_m^T \mathbf{x} + b_m$, the transformation $\mathbf{W} \in \mathbb{R}^{M \times D}$ can be considered as a finite kernel transformation which turns a D-dimension problem into a $M \times D$-dimension one [15]. From the viewpoint of computation complexity, LLSVM requires a coding scheme that performs well when the number of anchors is as small as possible. It is totally different from the applications of feature learning.

B. Related Work

Locality is the key notation to approximate a nonlinear function with linear ones in machine learning. The classical example is the k- or the ϵ-neighborhood graph in manifold learning [1], [20], [27]. Generally, locality at least contains the following two components: 1) the way to define a local area and 2) the coding scheme among these data points.

Once a local area is determined, the coding scheme would govern the quality of the nonlinear approximation ability. Explicit coding scheme, such as [15], first uses k-means to generate anchors, and then adopts the inverse Euclidean distance for coding. Locality-constrained linear coding (LLC) [35] uses Gaussian distribution as the decay function to constrain local codings. On the contrary, the other abandons the notation of locality, instead adopts "anchor plane"—assuming infinite anchors live on a plane [48].

In this paper, we propose implicit coding scheme which does not necessarily aim at defining a special decay function, but instead learning local codings from the manifolds of datasets. This paper follows the notation of locality, proposing to use the Laplacian matrix [20] to define locality.

Previously, Laplacian matrix has been studied as a general way to describe manifolds for numerous applications, e.g., semi-supervised learning [30], dimension reduction [1], [41], [47], image reranking [40], and classification [39], [42] as well as feature learning [9], [29], [50]. In contrast to these previous work, Laplacian matrix in this paper does not be built with every point in a dataset, but with the limited number of anchors, which allows our method to scale up well.

III. Theoretical Aspects of Local Coding Scheme

A. Revisit Local Coordinate Coding

Following the convention of pattern recognition, we give a summary of some notations used in this paper in Table I. Moreover, some definition and conclusion about LCC are firstly revisited.

Definition 1 (Lipschitz Smoothness [44]): A function $f(x) \in \mathbb{R}^D$ is (α, β, p)-Lipschitz smooth with respect to the $\|\cdot\|^2$ norm, if $|f(\mathbf{x}') - f(\mathbf{x})| \leq \alpha\|\mathbf{x} - \mathbf{x}'\|$ and $|f(\mathbf{x}') - f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{x}' - \mathbf{x})| \leq \beta\|\mathbf{x} - \mathbf{x}'\|^{1+p}$, where we assume $\alpha, \beta > 0$ and $p \in (0, 1]$.

Definition 2 (Coordinate Coding [44]): Let $(\gamma, \mathcal{C})$ be an arbitrary coordinate coding on $\mathbb{R}^D$. Let f be an (α, β, p)-Lipschitz smooth function. We have for all $\mathbf{x} \in \mathbb{R}^D : \gamma(\mathbf{x}) = \sum_{\mathbf{v} \in \mathcal{C}} \gamma_v(\mathbf{x})\mathbf{v}$.

Lemma 1 (Linearization [44]): Let $(\gamma, \mathcal{C})$ be an arbitrary coordinate coding on $\mathbb{R}^D$. Let f be an (α, β, p)-Lipschitz

TABLE I
SOME NOTATIONS IN THIS PAPER

Notation	Definition
$f(\mathbf{x})$	An (α, β, p)-Lipschitz smooth function
$\mathbf{v} \in \mathbb{R}^D$	A D-dimension anchor
$\mathcal{C} \subset \mathbb{R}^D$	A set of anchors
$\gamma_v(\mathbf{x}) \in \mathbb{R}$	The local coding of a data $\mathbf{x}$ on the anchor $\mathbf{v}(\mathbf{v} \in \mathcal{C})$
$\gamma_x \in \mathbb{R}^{\lvert\mathcal{C}\rvert}$	The local coding vector of a data $\mathbf{x}$ by all anchors
$\gamma_i \in \mathbb{R}^M$	The local coding for the data $\mathbf{x}_i$
$(\gamma, \mathcal{C})$	A coordinate coding with coding scheme γ
$L \in \mathbb{R}^{M \times M}$	The Laplacian matrix built from M anchors

smooth function. We have for all $\mathbf{x} \in \mathbb{R}^D$

$$\left| f(\mathbf{x}) - \sum_{\mathbf{v} \in \mathcal{C}} \gamma_v(\mathbf{x}) f(\mathbf{v}) \right| \leq \alpha \|\mathbf{x} - \gamma(\mathbf{x})\|^2 + \beta \sum_{\mathbf{v} \in \mathcal{C}} |\gamma_v(\mathbf{x})| \|\mathbf{v} - \gamma(\mathbf{x})\|^{1+p}. \quad (2)$$

Lemma 1 indicates that the nonlinear function $f(\mathbf{x})$ is bounded by the weighted sum of the reconstruction $\|\mathbf{x} - \gamma(\mathbf{x})\|^2$ error and the locality $|\gamma_v(\mathbf{x})| \|\mathbf{v} - \gamma(\mathbf{x})\|^{1+p}$ error. A common practice to define a coding scheme in (2) is to assign a special value to p [36]. But the optimal value p is difficult to be determined.

B. Local Gaussian Coding and Local Student Coding

If a sample $\mathbf{x}$ is closer to anchors $\mathbf{v}_m$, according to the principle of locality, the local codings $\gamma_m(\mathbf{x})$ should be larger and vice versa [27], [44]. Local Gaussian coding (LGC) and local student coding (LSC) represent the heavy-tail decay function and the short-tail one, respectively. More concretely, LGC presumes that the relation between samples and anchors is

$$\gamma_v^{lgc}(\mathbf{x}; \mathbf{v}, \sigma) \propto \exp\left(\frac{-\|\mathbf{v} - \mathbf{x}\|^2}{\sigma^2}\right) \quad (3)$$

where $\mathbf{v}$ ($\mathbf{v} \in \mathbb{R}^D$) is an anchor, and the hyper-parameter σ controls the weight decay speed to achieve locality. While LSC uses Student t-distribution with one degree of freedom which is the same as Cauchy distribution

$$\gamma_v^{lsc}(\mathbf{x}; \mathbf{v}, \sigma) \propto \left(\sigma^2 + \|\mathbf{v} - \mathbf{x}\|^2\right)^{-1} \quad (4)$$

where the hyper-parameter σ also controls the weight decay speed. Student t-distribution has the nice property that $(\sigma^2 + \|\mathbf{v} - \mathbf{x}\|^2)^{-1}$ approaches an inverse square law for a large pairwise distance $\|\mathbf{v} - \mathbf{x}\|^2$.

Both LGC (3) and LSC (4) belong to explicit coding scheme, as both e^{-d^2/σ^2} and $(\sigma^2 + d^2)^{-1}$ explicitly predefine how local codings change with respect to the distances $d = \|\mathbf{v}_m - \mathbf{x}\|$ (see Fig. 1).

1) Upperbound of Locality Error: Inspired by the idea in [48], we theoretically compare the locality errors of different coding schemes to answer the problem: given several different coding schemes, which one has a lower locality error than the others?

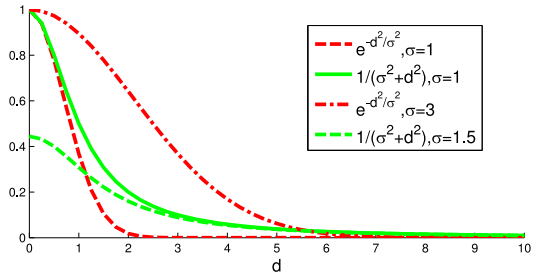


Fig. 1. Decay ability of the two LCSs: e^{-d^2/σ^2} for (3) and $1/(\sigma^2 + d^2)$ for (4).

Theorem 1 (Locality Error of LGC): Let $(\gamma, \mathcal{C})$ be an LGC in (3) on $\mathbb{R}^D$, and let f be an (α, β, p)-Lipschitz smooth function. For all $\mathbf{v} \in \mathcal{C}$, we denote $1 \leq \|\mathbf{v}\| \leq h$, and

$$d_u = \max_{\mathbf{v} \in \mathcal{C}} \max_{\mathbf{x}} \|\mathbf{x} - \mathbf{v}\|.$$

Then the locality error in Lemma 1 has the following upperbound:

$$\sum_{\mathbf{v} \in \mathcal{C}} |\gamma_v(\mathbf{x})| \, \| \, \mathbf{v} - \gamma(\mathbf{x}) \, \|^{1+p} \leq \begin{cases} \left[5h^2 + \frac{2}{|\mathcal{C}|}\left(\frac{d_u^2}{\sigma^2} - 1\right)\right]^{\frac{1+p}{2}} & \text{if } d_u \geq \sigma, \\ \left[5h^2 + \frac{2}{|\mathcal{C}|}\left(1 - \frac{d_u^2}{\sigma^2}\right)\right]^{\frac{1+p}{2}} & \text{otherwise.} \end{cases} \quad (5)$$

Theorem 1 discovers that the locality error of LGC is controlled by the hyper-parameter σ: if the choice of the parameter σ makes $(d_u^2/\sigma^2) < 1$, we would obtain more lower error than other ones; besides, $d_u^2 < \sigma^2$ means that there exists the optimal hyper-parameter σ for LGC.

Theorem 2 (Locality Error of LSC): Let $(\gamma, \mathcal{C})$ be an LSC (4) on $\mathbb{R}^D$, and f be an (α, β, p)-Lipschitz smooth function. For all $\mathbf{v} \in \mathcal{C}$, we denote $1 \leq \|\mathbf{v}\| \leq h$, and

$$d_l = \min_{\mathbf{v} \in \mathcal{C}} \max_{\mathbf{x}} \|\mathbf{x} - \mathbf{v}\|, \qquad d_u = \max_{\mathbf{v} \in \mathcal{C}} \max_{\mathbf{x}} \|\mathbf{x} - \mathbf{v}\|$$
$$c_l = \min_{\mathbf{v} \in \mathcal{C}} \left(\gamma_v^{lsc}(\mathbf{x})\left(\sigma^2 + \|\mathbf{v} - \mathbf{x}\|^2\right)\right),$$
$$c_u = \max_{\mathbf{v} \in \mathcal{C}} \left(\gamma_v^{lsc}(\mathbf{x})\left(\sigma^2 + \|\mathbf{v} - \mathbf{x}\|^2\right)\right).$$

Then the locality error in Lemma 1 has the following upperbound:

$$\sum_{\mathbf{v} \in \mathcal{C}} |\gamma_v(\mathbf{x})| \, \| \, \mathbf{v} - \gamma(\mathbf{x}) \, \|^{1+p} \leq \left[5h^2 - \frac{2c_l d_l^2}{c_u |\mathcal{C}| \left(\sigma^2 + d_u^2\right)}\right]^{\frac{1+p}{2}}. \quad (6)$$

Theorem 2 indicates that if we reduce the hyper-parameter σ, the locality error of LSC also decreases, because the term $-(2c_l d_l^2 / c_u |\mathcal{C}| (\sigma^2 + d_u^2))$ is always negative.

Example 1 (Drawback of Explicit Coding Scheme): A simple example, illustrated in Fig. 2, reconstructs a

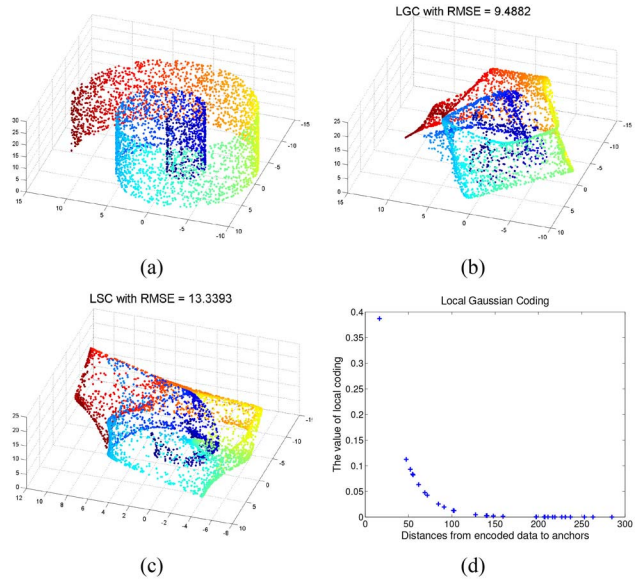


Fig. 2. Reconstruction of swiss-roll by both LGC and LSC. (a) Sampled swiss-roll. (b) Reconstructed manifold with 32 anchors by LGC. (c) Reconstructed manifold with 32 anchors by LSC. (d) Distribution of local codings generated by LGC.

swiss-roll manifold. The reconstructed manifolds from different LCSs are shown in Fig. 2(b) and (c), where 4096 points are randomly sampled from the ideal manifold. Thirty-two points are randomly sampled as anchors in the reconstruction stage, with performances evaluated by root mean square error. When a limited number of anchors are supplied, the swiss-roll is badly reconstructed by both LGC and LSC. Moreover, the distribution of local codings [Fig. 2(d)] almost follows the shapes of Gaussian function (Fig. 1). Therefore, the poorly reconstructed manifolds are caused by the difference between the local codings computed by minimizing reconstruction error and the limited freedom predefined in explicit coding scheme.

Although the locality errors of both LGC and LSC are bounded, explicit coding scheme does not balance well between reconstruction and locality in LCC. Nevertheless what the other useful observations can we draw from these theoretical results?

IV. Local Laplacian Coding

Revisiting Theorems 1 and 2, we find that both LGC and LSC share the same point—the built-in locality guarantees the existence of bounded locality error. However, the predefined locality does not balance well between reconstruction and locality errors. To overcome the deficiency of explicit coding scheme while still obtaining the bounded locality error, implicit coding scheme is proposed to grant more freedom to minimize reconstruction error. Rather than predefining how local codings decay with respect to the distances from encoded data to anchors, the distribution of local codings is implicitly described by the manifold of datasets, and local codings are optimized by jointly minimizing both reconstruction and locality errors.

Given Laplacian matrix $L = D - W$ [20], the locality of LPC satisfies the following optimization problem:

$$\arg\min_{\gamma_i} \gamma_i^T L \gamma_i = \arg\min_{\gamma_{i_m}, \gamma_{i_n}} \sum_{m,n} (\gamma_{i_m} - \gamma_{i_n})^2 w_{mn} \tag{7}$$

where L is a $M \times M$ matrix built with anchors $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_M]$, γ_{i_m} are the codings generated by anchors $\mathbf{v}_m$ for the ith sample, w_{mn} are the weights between anchors $\mathbf{v}_m$ and $\mathbf{v}_n$. The locality of LPC (7) has the built-in locality. Because if anchors $\mathbf{v}_m$ and $\mathbf{v}_n$ are close to each other, then local codings γ_{i_m} and γ_{i_n} will be similar to each other and vice versa. Besides, local codings γ_i are optimized from the manifold of datasets, instead of following the special decay function. Theorem 3 discovers that the negative local codings tend to increase the locality error of LPC.

Theorem 3 (Impact of Negative Local Codings in LPC): Let $(\gamma, \mathcal{C})$ be an LPC on $\mathbb{R}^D$, and let f be an (α, β, p)-Lipschitz smooth function. For all $\mathbf{v} \in \mathcal{C}$, we denote $1 \le \|\mathbf{v}\| \le h$, and $K = \max_{\mathbf{x}} |\{\mathbf{v} \in \mathcal{C} : \gamma_v^{lpc}(\mathbf{x}) < 0\}|$. We also denote

$$\gamma_l = \min_{\mathbf{v}\in\mathcal{C}} \left\{ \left| \min_{\mathbf{x}} \gamma_v^{lpc}(\mathbf{x}) \right|, \left| \max_{\mathbf{x}} \gamma_v^{lpc}(\mathbf{x}) \right| \right\}$$
$$\gamma_u = \max_{\mathbf{v}\in\mathcal{C}} \left\{ \left| \min_{\mathbf{x}} \gamma_v^{lpc}(\mathbf{x}) \right|, \left| \max_{\mathbf{x}} \gamma_v^{lpc}(\mathbf{x}) \right| \right\}.$$

Then the locality error of (7) has the following upperbound:

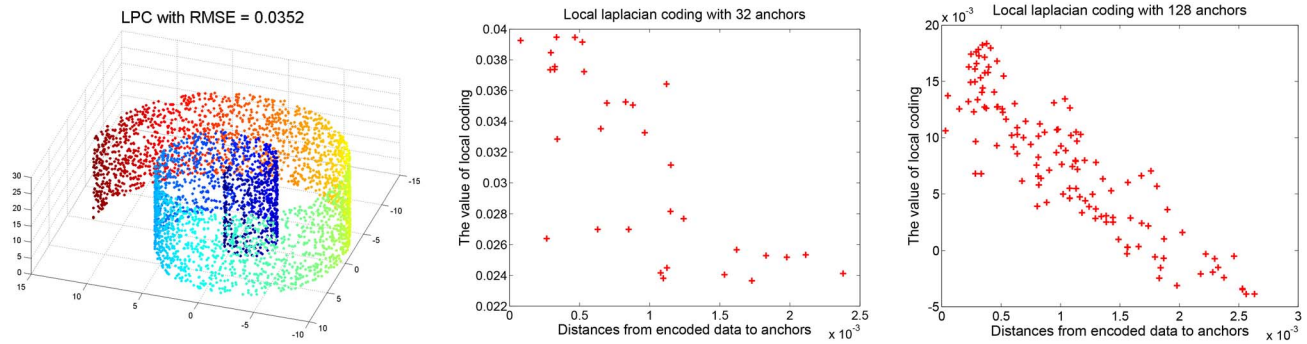
$$\sum_{\mathbf{v}\in\mathcal{C}} |\gamma_v(\mathbf{x})| \parallel \mathbf{v} - \gamma(\mathbf{x}) \parallel^{1+p} \le \frac{\gamma_u}{\gamma_l} \left[h^2 + 2\gamma_u h^2 \frac{K + |\mathcal{C}|}{\gamma_l |\mathcal{C}|} + \frac{\gamma_u^2 h^2 + 2|\mathcal{C}|\gamma_u^2 h^2}{|\mathcal{C}|\gamma_l^2} \right]^{\frac{1+p}{2}}. \tag{8}$$

Example 2 (Implicit Coding Scheme and Reconstruction Error): Fig. 3 plots the reconstructed manifold and the distribution of local codings. As illustrated in Fig. 3(b), LPC has the implicit locality by enforcing the anchors faraway from the encoded data have small values. When the number of anchors increases, some of the local codings faraway from the input data are negative [see Fig. 3(c)]. Because LPC does not enforce non-negativity in local codings.

To remedy this drawback and speedup coding process, we can simply use the neighborhood anchors of $\mathbf{x}_i$ by sampling k-NN graph, and solve a much smaller linear system. Concretely, let I_x ($I_x \in \mathbb{R}^M$) be the index of s nearest anchors of $\mathbf{x}$, the subsampled graph sW is a subgraph of W

$$sW = W(I_x, I_x) \tag{9}$$

where $W \in \mathbb{R}^{M\times M}$ is the original k-NN graph. Rather than using s nearest anchors to directly rebuild a new k-NN graph, the subsampled graph naturally preserves the structure of the original manifold. In the following section, LPC-S is used to denote LPC with the subsampled k-NN graph.



(a) (b) (c)

Fig. 3. Reconstruction of swiss-roll from LPC. (a) Reconstructed swiss-roll with 32 anchors and 5-NN. (b) Distribution of local codings with 32 anchors. (c) Distribution of local codings with 128 anchors.

A. Approximated Anchors for LPC

Since Theorem 3 proves that the locality error of LPC is also bounded, we only minimize the regularized reconstruction error as

$$\begin{aligned} \min_{\gamma_i, \mathbf{V}} \quad & J(\gamma_i, \mathbf{V}) = \frac{1}{2}\|\mathbf{x}_i - \gamma_i \mathbf{V}\|^2 + \lambda \gamma_i^T L \gamma_i \\ \text{s.t.}: \quad & \gamma_i^T \mathbf{1} = 1 \end{aligned} \tag{10}$$

where λ is the tradeoff parameter between reconstruction and locality, and $\mathbf{1}$ denotes the identity vector $[1, \ldots, 1]^T$.

For real applications, Laplacian matrix in (10) may be non-differentiable, for instance, the histogram intersection is used to build Laplacian matrix. Therefore, a simple way to generate anchors is clustering-based methods, i.e., k-means. According to our experimental results in Section V, the anchors generated by k-means produce satisfactory accuracy. In this paper, Laplacian matrix is computed by heat kernel [1].

Solving local codings is a constrained linear least-square problem when anchors are learned. To determine the optimal local codings γ_i, the constrained problem can be solved with the Lagrangian function $L(\gamma_i, v)$

$$\begin{aligned} L(\gamma_i, v) &= \frac{1}{2}||\mathbf{x}_i - \mathbf{V}\gamma_i||^2 + \lambda \gamma_i^T L \gamma_i + v\big(1 - \gamma_i^T \mathbf{1}\big) \\ &= \frac{1}{2}\gamma_i^T \Phi \gamma_i + \lambda \gamma_i^T L \gamma_i + v\big(1 - \gamma_i^T \mathbf{1}\big) \end{aligned} \tag{11}$$

where matrix Φ is $(\mathbf{x}_i \mathbf{1}^T - \mathbf{V})^T (\mathbf{x}_i \mathbf{1}^T - \mathbf{V})$, and v is Lagrangian multiplier. Let $\partial L(\gamma_i, v)/\partial \gamma_i = 0$, the optimal local codings γ_i satisfy that

$$\begin{aligned} \tilde{\gamma}_i &= (\Phi + 2\lambda L)^{-1} \mathbf{1} \\ \gamma_i &= \tilde{\gamma}_i / \big(\tilde{\gamma}_i^T \mathbf{1}\big). \end{aligned} \tag{12}$$

It should be noted that the matrix Φ is symmetric and semi-positive. If the matrix Φ is singular or nearly singular, the matrix $\Phi + 2\lambda L$ is still conditioned. Because $2\lambda L$ penalizes large distance that exploits correlation beyond some level of precision between data points.

V. Experiments

A. Methods in Comparison Study

Our specific experimental goal is to compare the proposed approach with state-of-the-art methods.

1) LGC and LSC [25]: For a sample $\mathbf{x}_i$, both LGC and LSC are directly computed as

$$\gamma_{i_m} = \frac{\exp\left(\dfrac{-\|\mathbf{v}_m - \mathbf{x}_i\|^2}{\sigma^2}\right)}{\sum_{\mathbf{v}_m \in \mathcal{C}} \exp\left(\dfrac{-\|\mathbf{v}_m - \mathbf{x}_i\|^2}{\sigma^2}\right)} \tag{13}$$

$$\gamma_{i_m} = \frac{\big(\sigma^2 + \|\mathbf{v}_m - \mathbf{x}_i\|^2\big)^{-1}}{\sum_{\mathbf{v}_m \in \mathcal{C}} \big(\sigma^2 + \|\mathbf{v}_m - \mathbf{x}_i\|^2\big)^{-1}}. \tag{14}$$

It should be noted that both LGC and LSC ignore the reconstruction problem in LCC.

2) LLC [35]: For a sample $\mathbf{x}_i$, LLC [35] is originally proposed to learn feature as follows:

$$\begin{aligned} \min_{\gamma_i} \quad & \|\mathbf{x}_i - \mathbf{V}\gamma_i\|^2 + \lambda \|p_i \odot \gamma_i\|^2 \\ \text{s.t.} \quad & \gamma_i^T \mathbf{1} = 1 \end{aligned} \tag{15}$$

where the symbol $\odot$ denotes the element-wise multiplication, and p is the locality adaptor as $\exp(\|\mathbf{v} - \mathbf{x}\|^2/\sigma^2)$. Compared with LGC, LLC is jointly optimized with both reconstruction and locality errors.

3) LCC-ℓ_2 [36]: This approach sets $p = 2$ to locality in (2), and formulates this special constraint as least angle regression shrinkage (LARS)-Lasso [31] problem

$$\min_{\gamma_i, \mathbf{v}_m} \|\mathbf{x}_i - \mathbf{V}\gamma_i\|^2 + \lambda \sum_{m=1}^{M} |\gamma_{i_m}| \|\mathbf{v}_m - \mathbf{x}_i\|^2. \tag{16}$$

4) Orthogonal Coordinate Coding [48]: For a sample $\mathbf{x}_i$, orthogonal coordinate coding (OCC) assumes that coding scheme satisfies that

$$\gamma_{i_m} \propto \frac{\mathbf{x}_i^T \mathbf{v}_m}{\sigma_m}. \tag{17}$$

In training stage, OCC applies singular value decomposition to find M largest singular values σ_m and the corresponding anchors $\mathbf{v}_m$.

5) Graph Regularized Sparse Coding [50]: For a sample $\mathbf{x}_i$, graph regularized sparse coding (GRSC) optimizes the following problem:

$$\min_{\gamma_i, \mathbf{v}_m} \|\mathbf{x}_i - \mathbf{V}\gamma_i\|^2 + \alpha \gamma_i^T L \gamma_i + \beta \sum_{m=1}^{M} |\gamma_{i_m}|. \tag{18}$$

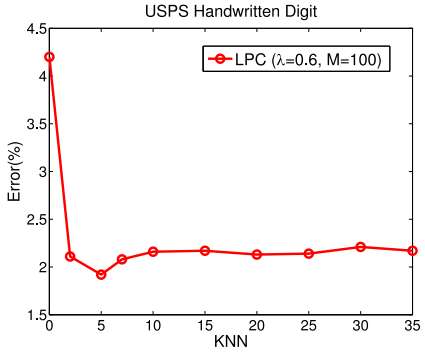


Fig. 4. Classification error as a function of the number of k in k-NN graph with $M = 100$ and $\lambda = 0.6$.

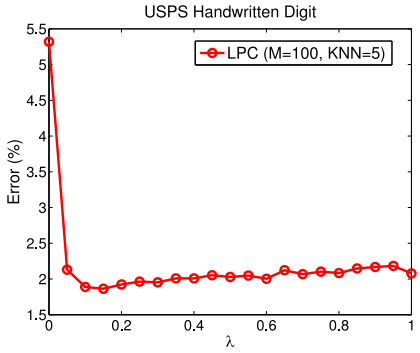


Fig. 5. Classification error as a function of λ with $M = 100$ and 5-NN.

GRSC and its extension [18] originally are not proposed for locally linear classification problem, but for image representation. The ℓ_1-based sparsity can be considered as an alternative to our subsampled k-NN graph in (9). As discussed in Section II-A, ℓ_1 globally selects anchors from all these anchors. Compared with GRSC, LPC-S is a more local approach by explicitly subsampling nearest anchors to reconstruct a sample. This simple change in LPC-S largely alleviates the well-known "crowding problem" [21] in manifold learning: high-dimensional points with moderate distances tend to crowd together in the low-dimensional map. The subsampled nearest anchors in LPC-S naturally make these points with moderate distances disappear in the low-dimensional map. Moreover, GRSC has to face the negative local coding problem stated in Theorem 3.

B. Experimental Tests

1) LPC Parameter Choices: In LPC formulation, the important parameters include: the number of k in k-NN graph, the weight λ about Laplacian constraint, and the sparsity s to sample a subgraph. We experimentally test the choice of parameters on United States Postal Service (USPS) data set [13]. USPS consists of 7291 training examples and 2007 gray-scale 16×16 ones for test. Each label corresponds to 0–9 digits. During the experiments, the means of raw images are first removed, and then we normalize images with ℓ_2 norm. LLSVM is trained by primal estimated sub-GrAdient solver for Svm method [28].

The effect of k in building k-NN graphs is shown in Fig. 4. We can see from the results that error rates decrease rapidly when the locality constraint is used. Although k changes from 7 to 35, the performances are very robust. This result verifies our claim that LPC has the implicit locality and balances well between reconstruction and locality errors in LCC.

Fig. 5 plots classification errors over a series of the trade-off parameter λ. LPC is an efficient locality constraint, as the error rates decrease from 5.43% (with $\lambda = 0$) to 2.14% (with $\lambda = 0.05$). Moreover, the implicit coding scheme makes LPC robust to the change of parameter λ. This phenomenon is consistent with the observations in Fig. 4.

Finally, the choice of s in LPC-S, the ratio of subsampled neighborhood anchors, is studied. Fig. 6 illustrates the effectiveness of sparsity in LPC-S, showing that the error rates first decrease when the number of neighborhood anchors is

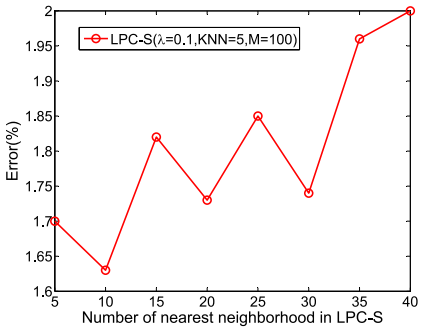


Fig. 6. Classification error as a function of the sparsity in LPC-S.

TABLE II
CLASSIFICATION ERROR RATE (%) ON USPS WITH DIFFERENT NUMBER OF ANCHORS

Coding scheme	The number of anchors									
	20	40	60	80	100	120	140	160	180	200
LGC	2.93	2.21	2.42	2.66	2.77	3.29	3.88	3.88	4.18	4.23
LSC	2.64	3.38	4.40	4.52	4.59	4.73	4.99	5.10	5.35	5.78
LPC	**2.37**	**2.01**	**1.76**	**1.74**	**1.74**	**1.91**	**1.89**	**1.83**	**1.96**	**1.95**

increased, but increase rapidly if the number of anchors is larger than a certain threshold. The minimal error rate achieves at $s = 10/100$. In our comprehensive tests, reported in the following sections, we will present results with the setting: $s = 0.1$, $\lambda = 0.1$, and $k = 5$.

2) LPC Versus LGC and LSC: As both LGC and LSC are the inspiration of LPC, even if not the state-of-the-art methods, we perform an initial comparison in Table II. We perform 5-fold cross-validation to find the best parameters in LGC and LSC. The test values for the σ are $\{0, 0.2, 0.5, 1, 1.5, 2, 3, 5, 7, 9, 10\}$. When a very small number of anchors are used (≤ 80 for all classes), LPC achieves the excellent performance among these coding schemes. Besides, if the number of anchors continuously increases (≥ 100 for all classes), the error rates of LPC increase slightly. This comparison discovers that LPC does not explicitly predefine how local codings decay with respect to the distances from encoded data to anchors. This grants more freedom to minimize reconstruction error in LCC (2). On the other hand, Theorem 3 already points out that the negative local codings of LPC increase the locality error. Consequently the error rate increases when the number of anchors is larger than a threshold (≥ 100 for all classes). This phenomenon will be further discussed in Section V-B3.

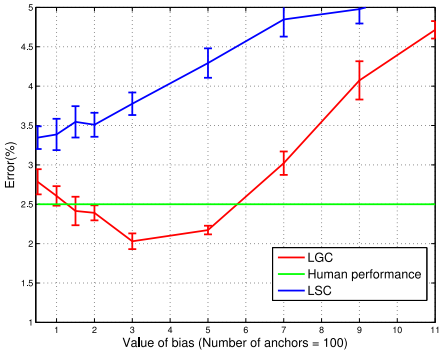


Fig. 7. Comparison between LGC and LSC with the different value of σ.

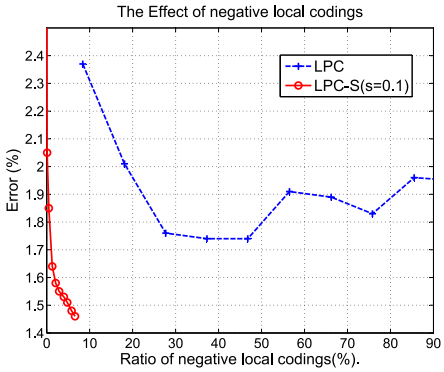


Fig. 8. Relation between the ratio of negative codings and classification accuracy on USPS dataset.

3) Justification of the Theoretical Results: Fig. 7 illustrates that the error rates of LGC first decrease until a certain number of anchors are used, and then the error increases gradually. This observation well verifies the conclusion in Theorem 1: there is an optimal σ in LGC to reduce the locality error. For Theorem 2, Fig. 7 illustrates that the error rates of LSC rise gradually when the σ increases. Obviously, Theorem 2 explains that the σ should be as small as possible to reduce the locality error.

To give a view of how negative local codings impact the locality error of LPC, the ratio of negative local codings is measured as, $1/N\sum_{i=1}^{N}(\sum_{m=1}^{M}[\gamma_{i_m}]/M)$, where the operation $[A]$ means that, $[A]=1$, if $A<0$, otherwise $[A]=0$. Fig. 8 shows the ratio of negative local codings for both LPC and LPC-S, where the number of anchors ranges from 20 to 200 with the step size 20. As discussed in Theorem 3, the number of negative local codings increase the locality error, and as a result the error rates of LPC increase at about 50% negative local codings. Also notice that the error rates of LPC-S consistently decrease with the increase of the number of anchors. Because the ratio of negative local codings is restricted to be smaller than that of LPC.

C. Comparative Evaluation

In this section, we compare the proposed approach specifically to the state-of-the-art methods on USPS, Mixed National Institute of Standards and Technology database (MNIST), and Chars74K datasets. Because these datasets are widely used to evaluate classification performance. MNIST contains 40 000 training and 10 000 test 28×28 gray-scale images, which are reshaped directly into the 784 dimension vectors. The label of each image is one of the ten digits from 0–9. Chars74K comprises 62 classes ("0"–"9," "A"–"Z," "a"–"z"), 7705 characters obtained from natural images, 3410 hand drawn characters using a tablet PC, and 62 992 synthesized characters from computer fonts, with over 74K images in total. To give a fair comparison, we follow [49] to resize each image into a 8×8 gray image, then randomly split it into two independent sets, 7400 images as test data and the rest as training data, and finally vectorize each image into a 64 dimensional vector. During the experiments, the mean of each raw image is also first removed, and then we normalize these images with ℓ_2 norm. To make the comparison as meaningful and fair as possible, we directly use some experimental results from OCC [48], [49].

Only local codings at testing stage are computed for locally linear classification, if anchors are already learned at training stage. Therefore, we begin with an analysis of the computation cost at testing stage in Table III. It is clear that the relative complexities are determined by the number of anchors M and the dimension of features D, respectively. The computational complexities of OCC, LSC, and LGC are lowest among these methods. Both LCC-ℓ_2 and GRSC have to solve a sparse problem which is computationally intensive process for a high-dimension coding. Compared with both LPC and LLC, LPC-S greatly accelerates the coding speed by sampling k-NN graphs.

We further compare the empirical computation time at both training and testing stages. To give a fair comparison, the training time includes both the anchors learning and local codings optimization without training LLSVM, and the testing time only contains the computation of local codings. Table IV lists the results based on unoptimized MATLAB code on a single thread of a 2.67 GHz CPU with 4G memory. For training both GRSC and LPC-S achieve faster speeds than the others. Because there is no complex optimization, just k-means. At testing stage, sparse coding-based method is slower than the others since both LARS-Lasso [31] in LCC-ℓ_2 and feature-sign search [17] in GRSC are applied to every sample. Generic OCC (G-OCC) has the lowest computation cost than the others at testing stage. Because G-OCC just multiples between anchors and samples and normalizes local codings. Compared with LLC, LPC-S only solves a much smaller linear system (12) by sampling k-NN graph.

The best published USPS, MNIST, and Chars74K classification error rates are 1.36% [4], 4.38% [32], and 16.48% [38], respectively, achieved by support vector machine (SVM)-based methods. Our goal is to achieve exceeded or comparable performances in LLSVM with the number of anchors as small as possible. Both Tables V and VI present a comparison of Class-specific OCC (C-OCC), G-OCC, LCC-ℓ_2, LLC, CRSC, and LPC-S, using different numbers of anchors. The results of C-OCC and G-OCC are taken from [48], and all the other error rates are computed by us.

In contrast to LLC, LCC-ℓ_2, and OCC, LPC-S not only tends to offer a higher classification accuracy, but also uses a much smaller number of anchors. For instance, LPC-S achieves 1.81% error rate with 100 anchors on MNIST, while

TABLE III
COMPARISON OF TIME COMPLEXITY AT TESTING STAGE AMONG LSC, LGC, LLC, OCC, GRSC, AND LPC

Method	Time Complexity	Comments
LSC (LGC)	$\mathcal{O}(M)$	LSC or LGC directly computes local codings without any optimization.
LLC	$\mathcal{O}(M)+\mathcal{O}(M^2)$	LLC firstly computes the locality adaptor with $\mathcal{O}(M)$, and operates matrix inverse on the $M \times M$ matrix with $\mathcal{O}(M^2)$ [12].
LCC-ℓ_2	$\mathcal{O}(MDs+Ms^2)+\mathcal{O}(M^2)$	LCC-ℓ_2 firstly costs $\mathcal{O}(MDs+Ms^2)$ to solve a Cholesky-based implementation of LARS-Lasso problem [22], and then computes the matrix inverse with $\mathcal{O}(M^2)$.
OCC	$\mathcal{O}(M)$	OCC directly computes dot product between anchors and samples.
GRSC	$\mathcal{O}(t \cdot M^2)$	In each iteration, GRSC solves a quadratic programming (QP) problem $\mathcal{O}(M^2)$ to optimize local codings by feature-sign search algorithm [17]
LPC	$\mathcal{O}(M^2)$	LPC calculates matrix inverse with $\mathcal{O}(M^2)$.
LPC-S	$\mathcal{O}(M)+\mathcal{O}(s^2)$	LPC-S first searches the s neighborhood anchors, and solves the matrix inverse problem with $\mathcal{O}(s^2)$, $s \ll M$.

Here, s counts the number of nonzero coefficients in LASSO problem, or the number of nearest anchors in LPC-S. To produce better results, typically, s is around the 10% of M. t is the number of iteration in sparse coding by feature-sign search algorithm.

TABLE IV
COMPUTATIONAL TIME COMPARISON BETWEEN DIFFERENT CODING SCHEMES ON MNIST AND USPS

Coding schemes	Training time (sec.)		Testing time (sec.)	
	MNIST	USPS	MNIST	USPS
G-OCC	1.85×10^3	50.00	**8.02**	**0.08**
LCC-ℓ_2	308.89	178.51	971.55	90.96
LLC	210.38	70.68	292.87	100.03
GRSC	**172.11**	**7.71**	1.56×10^3	172.96
LPC-S	172.11	7.71	28.94	2.03

The number of anchors is 200.

TABLE V
CLASSIFICATION ERROR RATE (%) ON USPS WITH DIFFERENT NUMBER OF ANCHORS

Coding schemes	The number of anchors									
	20	40	60	80	100	120	140	160	180	200
C-OCC	4.75	4.61	4.40	4.10	4.31	N/A	N/A	N/A	N/A	N/A
G-OCC	4.50	4.60	4.50	4.60	5.22	N/A	N/A	N/A	N/A	N/A
LCC-ℓ_2	10.75	8.62	7.14	7.06	6.85	6.62	6.01	5.87	5.52	5.33
LLC	**2.53**	2.16	1.95	1.86	1.83	1.77	1.61	1.59	1.60	1.58
GRSC	2.63	2.13	2.08	1.85	1.84	1.79	1.61	1.57	1.50	1.47
LPC-S	2.73	**2.05**	**1.85**	**1.64**	**1.58**	**1.55**	**1.53**	**1.51**	**1.48**	**1.46**

TABLE VI
CLASSIFICATION ERROR RATE (%) ON MNIST WITH DIFFERENT NUMBER OF ANCHORS

Coding schemes	The number of anchors									
	20	40	60	80	100	120	140	160	180	200
C-OCC	3.45	2.25	**1.85**	**1.81**	**1.75**	N/A	N/A	N/A	N/A	N/A
G-OCC	**2.25**	**1.85**	2.00	2.10	1.90	N/A	N/A	N/A	N/A	N/A
LCC-ℓ_2	9.63	7.67	6.26	5.35	4.76	4.52	4.21	3.85	3.47	3.32
LLC	3.78	2.85	2.30	2.12	1.87	1.78	1.75	1.71	1.68	1.67
GRSC	3.92	2.87	2.59	2.05	1.97	1.81	1.82	1.71	1.68	1.68
LPC-S	3.69	2.68	2.18	2.09	1.81	**1.73**	**1.67**	**1.65**	**1.64**	**1.64**

120 anchors is required for LLC to achieve a comparable performance. Because both LLC and LCC-ℓ_2 belong to explicit coding scheme which predefines the distribution of local codings by special decay functions. LPC-S achieves slightly better results than GRSC, as GRSC has to face the "crowding problem" and the negative local coding problem. In contrast, subsampling in LPC-S not only largely avoids above problems, but also accelerates local codings by solving a small linear system.

Table VII further presents the overall best classification performance achieved by different SVM-based classifications with any parameter setting. The proposed LPC-S obtains the lowest error rate on USPS, and achieves slightly higher errors than MCSVM with radial basis function (RBF) kernel on MNIST and adaptive local hyperplane on Chars74K, respectively, despite the fact that the computational costs of both $\text{SVM}_{\text{struct}}$ and MCSVM with RBF kernel are far larger than that of LLSVM with LPC-S.

TABLE VII
CLASSIFICATION ERROR RATE (%) ON MNIST, USPS, AND CHARS74K

Algorithms (# anchors for all classes)	MNIST	USPS	Chars74K
PEG-LLSVM+LPC-S	1.64(200)	**1.46**(200)	16.48(1280)
Linear SVM+LCC(4096) [44]	1.90	N/A	N/A
Linear SVM+ improved LCC(4096) [43]	1.64	N/A	N/A
Linear SVM + LLC (4096) [35]	2.28	4.38	20.88
LA-SVM(2 passes) [3]	**1.36**	N/A	N/A
SVM_{struct} [32]	1.40	4.38	N/A
LL-SVM(10 passes, 100) [15]	1.85	5.78	N/A
ALH [38]	2.15	4.19	**16.26**
LIBLINEAR [5]	8.18	8.32	54.61

D. Results on Other Datasets

We also wish to test LLSVM-LPC-S on other benchmark datasets. As we note that sometimes we could not reproduce their results, largely due to subtle engineerings, e.g., the way of dealing with preprocessing, the details in extracting features. Therefore, some results are directly taken from their published results.

1) Caltech 101 Dataset: The Caltech 101 dataset [8] contains 101 categories (including animals, vehicles, flowers, etc.) with high-shape variability. The number of images per category varies from 31 to 800. Most images are medium resolution, i.e., about 300×300 pixels. The SIFT [19] is extracted on the image with a step of ten pixels and for the four different radii 4, 8, 12, 16. The SPM [16] feature is further extracted from 1×1, 2×2, and 4×4 blocks on three levels with ℓ_2 norm. The codebook is learned by *k*-means with 1024 clusters, and as a result the final SPM features are very high-dimension. We follow the common experiment setup for Caltech 101, training on 15 and 30 images per category and testing on the rest.

Fig. 9 shows the performance among different coding schemes with different number of anchors. By varying the number of anchors per class we aim to test the coding schemes on concurrence of the high-dimension feature and the multi-class problem. Our method shows much higher accuracy than

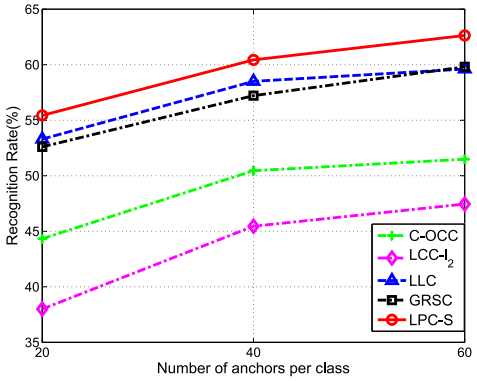


Fig. 9. Recognition performance of different coding schemes with different number of anchors per class for Caltech 101 dataset.

TABLE VIII
CLASSIFICATION ERROR RATE (%) ON CALTECH 101

Algorithms (#anchors)	15 training	30 training
SVM-KNN [45]	59.10±0.60	66.20±0.50
KSPM [16]	56.40	64.4±0.80
SVM+LLC [35]	**65.43**	**73.44**
KC [10]	N/A	64.14±1.18
PEG-LLSVM+LPC-S(60 per class)	62.63±0.43	71.60±0.54

other coding schemes. We also notice that LCC-ℓ_2 achieves the poorest recognition accuracy, and this may result from the fact that the local codings in (16) can be considered as a variant of LSC which generally achieves poor performance [25].

As shown in Table VIII, we can observe that the proposed method achieves similar performance to SVM-K-neighbor hood, but obtains a lower error than the kernel methods, i.e., kernel SPM. As listed in both Tables V and VI, the performance of LPC-S is comparable to other state-of-the-art methods. Note that SVM + LLC [35] applies LCC for feature learning, in which both max pooling [37] and local coding on codebook are adopted; while our current implementation only uses the classical SPM feature.

2) Face Recognition: We evaluate our method on Extend Yale B dataset [11] and A. Martinez and R. Benavente (AR) dataset [23]. Extended Yale B contains 38 categories, and 2414 frontal-face images. The cropped image size is 192 × 168. Following [14], we randomly select a half of images in each category as training images, and use the rest as test ones. AR dataset contains over 4000 frontal face images corresponding to 126 persons, and the images include more facial expressions, illumination and occlusions. Thus AR is more challenge than Extended Yale B. We choose 47 male persons and 43 female ones for our evaluation. For each person, 26 images are taken in two separate sessions, where 14 images only contain variance in illumination and facial expressions. Following [14], we use 7 out of 14 images as training images and use the rest as test images. We use eigenface [24], [33] and downsaple feature [14], and each feature is normalized by ℓ_2 norm.

Fig. 10 compares the average recognition rate. The dimension of the eigenface is 504, and we perform ten random trials. It can be observed that LPC-S consistently outperforms other coding schemes. Interestingly, LPC-S performs worse than LLC when the number of anchors is relatively small, and

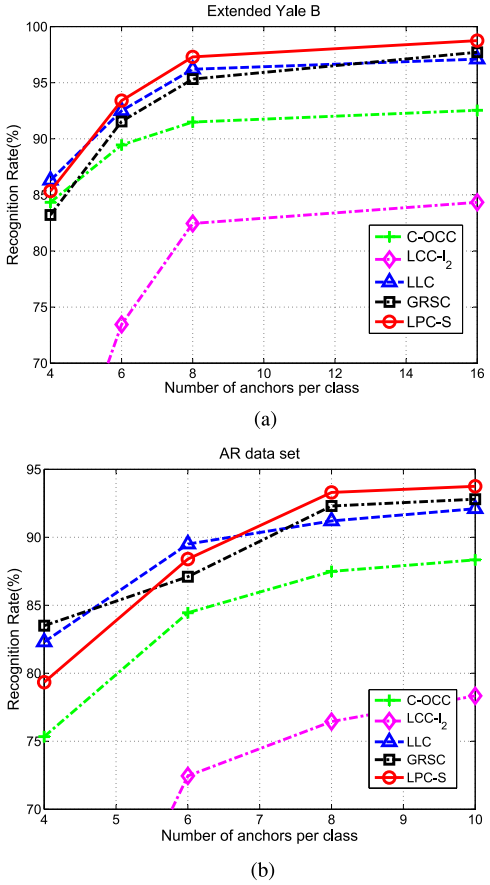


Fig. 10. Recognition performance of different coding schemes with different number of anchors per class for (a) extended Yale B dataset and (b) AR dataset.

TABLE IX
PERFORMANCE OF DIFFERENT METHODS FOR FACE RECOGNITION ON EXTENDED YALE B AND AR DATASET (%)

		Extended Yale B			AR		
	Dimension	56	120	504	54	130	540
Eigen	CRC [46]	85.62	94.41	98.55	76.67	87.30	87.62
	SC [14]	91.63	93.95	96.77	80.32	83.81	89.50
	Linear SVM	84.32	93.14	96.85	84.37	89.04	92.06
	LLSVM-LPC-S	**92.56**	**95.64**	**98.75**	**85.18**	**90.64**	**93.78**
Downsample	CRC [46]	82.84	92.93	97.23	69.21	83.17	71.90
	SC [14]	86.16	92.13	97.10	75.56	86.51	88.73
	Linear SVM	69.55	79.07	91.61	73.03	83.45	90.32
	LLSVM-LPC-S	**90.02**	**94.56**	**97.58**	**80.45**	**86.45**	**92.34**

this may be caused by Laplacian matrix that can not grasp manifold structure when a small number of anchors are used.

Table IX further compares the performances of LPC-S on Extended Yale B and AR face datasets with different features and feature dimension. All the results are based on ten independent trails. Experimental results show that LPC-S with LLSVM outperforms other methods. Especially for low-dimensional features, the improvement is significant, for example, LLSVM-LPC-S outperforms linear SVM about 8.24% on Extended Yale B. Moreover, we also note that LLSVM-LPC-S performs better than linear SVM when downsample feature is adopted.

VI. CONCLUSION

In this paper, we propose LPC from the theoretical analysis of LCSs, leading to results matching or surpassing the state-of-the-art methods for locally linear classification.

The theoretical analysis discovers that negative local codings in LPC increase the upperbound of the locality error. The LPC-S is therefore proposed to handle this problem by subsampling k-NN graphs. There are significant distinctions between the proposed coding schemes and previous studies in LCSs.

1) Granting more freedom to minimize the reconstruction error, the proposed implicit coding scheme balances well between the reconstruction and locality errors in LCC. When the same number of anchors is used, implicit coding scheme tends to achieve better performance than explicit one.
2) We theoretically analyze the impact of negative local codings in LPC and compare it with other LCSs.

The promising results of this paper motivate the following directions.

1) *Anchor Learning [26]:* Currently, we use k-means to approximately solve the anchors of LPC. How to optimize the anchors with Laplacian constraint is an interesting problem.
2) *Applications of LPC:* For example, LPC enforces the stable coefficients to help coding-based method more robust. Recently, the reconstruction-based classification has achieved promising results on object recognition [14], and LPC may also be used to solve this recognition problem.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the anonymous reviewers whose comments improved this paper greatly.

REFERENCES

[1] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2001, pp. 585–591.

[2] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Anchorage, AK, USA, 2008, pp. 1–8.

[3] A. Bordes, S. Ertekin, and L. Bottou, "Fast kernel classifiers with online and active learning," *J. Mach. Learn. Res.*, vol. 6, pp. 1579–1619, Nov. 2005.

[4] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *J. Mach. Learn. Res.*, vol. 6, pp. 1579–1619, Oct. 2005.

[5] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[6] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge, MA, USA: Cambridge Univ. Press, 2012.

[7] D. L. Donoho, "For most large underdetermined systems of linear equations, the minimal l1-norm solution is also the sparsest solution," Dept. Stat., Stanford Univ., Stanford, CA, USA, Tech. Rep., 2004. [Online]. Available: http://statweb.stanford.edu/~donoho/reports.html

[8] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Comput. Vis. Image Understand.*, vol. 106, no. 1, pp. 59–70, 2007.

[9] S. Gao, I. Tsang, and L.-T. Chia, "Laplacian sparse coding, hypergraph Laplacian sparse coding, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 92–104, Jan. 2013.

[10] J. Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders, "Kernel codebooks for scene categorization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 3. Marseille, France, 2008, pp. 696–709.

[11] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone modes for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643-660, Jun. 2001.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.

[13] J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

[14] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[15] L. Ladický and P. Torr, "Locally linear support vector machines," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Bellevue, WA, USA, 2011, pp. 985–992.

[16] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, 2006, pp. 2169–2178.

[17] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2006, pp. 801–808.

[18] W. Liu, D. Tao, J. Cheng, and Y. Tang, "Multiview Hessian discriminative sparse coding for image annotation," *Comput. Vis. Image Understand.*, vol. 118, pp. 50–60, Jan. 2014.

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

[20] U. Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 14, no. 7, pp. 395–416, 2007.

[21] L. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 1, pp. 2579–2605, Nov. 2008.

[22] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Jan. 2010.

[23] A. Martinez and R. Benavente, "The AR face database," Comput. Vis. Center, Univ. Autòn. Barcelona, Barcelona, Spain, Tech. Rep. 24, Jun. 1998.

[24] A. Mohammed, R. Minhas, Q. J. Wu, and M. Sid-Ahmed, "Human face recognition based on multidimensional PCA and extreme learning machine," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2588–2597, 2011.

[25] J. Pang, Q. Huang, B. Yin, L. Qin, and D. Wang, "Theoretical analysis of learning local anchors for classification," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, Tsukuba, Japan, 2012, pp. 1803–1806.

[26] J. Pang *et al.*, "Online dictionary learning for local coordinate coding with locality coding adaptors," *Neurocomputing*, vol. 157, pp. 61–69, Jun. 2015.

[27] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[28] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-GrAdient SOLver for SVM," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Corvalis, OR, USA, 2007, pp. 807–814.

[29] J. Shen, Y. Zhao, S. Yan, and X. Li, "Exposure fusion using boosting Laplacian pyramid," *IEEE Trans. Cybern.*, vol. 44, no. 9, pp. 1579–1590, Sep. 2014.

[30] V. Sindhwani, P. Niyogi, and M. Belkin, "Beyond the point cloud: From transductive to semi-supervised learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Bonn, Germany, 2005, pp. 824–831.

[31] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy. Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.

[32] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent out variables," *J. Mach. Learn. Res.*, vol. 6, pp. 1453–1484, Sep. 2005.

[33] M. Turk and A. Pentland, "Eigenfaces for recogntion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 3. Maui, HI, USA, 1991, pp. 71–86.

[34] J. Wang, A. Kalousis, and A. Woznica, "Parametric local metric learning for nearest neighbor classification," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Lake Tahoe, NV, USA, 2012, pp. 1610–1618.

[35] J. Wang *et al.*, "Locality-constrained linear coding for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, 2010, pp. 3360–3367.

[36] B. Xie, M. Song, and D. Tao, "Large-scale dictionary learning for local coordinate coding," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Aberystwyth, U.K., 2010, pp. 361–369.

[37] J. Yang, K. Yu, Y. Gong, and T. S. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, 2009, pp. 1794–1801.

[38] T. Yang and V. Kecman, "Adaptive local hyperplane classification," *Neurocomputing*, vol. 71, nos. 13–15, pp. 3001–3004, 2008.

[39] J. Yu, R. Hong, M. Wang, and J. You, "Image clustering based on sparse patch alignment framework," *Pattern Recognit.*, vol. 47, no. 11, pp. 3512–3519, 2014.
[40] J. Yu, Y. Rui, and D. Tao, "Click prediction for Web image reranking using multimodal sparse coding," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2009–2032, May 2014.
[41] J. Yu and D. Tao, *Modern Machine Learning Techniques and Their Applications in Cartoon Animation Research*. Piscataway, NJ, USA: Wiley/IEEE Press, 2013.
[42] J. Yu, D. Tao, and M. Wang, "Adaptive hypergraph learning and its application in image classification," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3262–3272, Jul. 2012.
[43] K. Yu and T. Zhang, "Improved local coordinate coding using local tangents," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, 2010, pp. 1215–1222.
[44] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2009, pp. 2223–2231.
[45] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. New York, NY, USA, 2006, pp. 2126–2136.
[46] L. Zhang and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Barcelona, Spain, 2011, pp. 471–478.
[47] Y. Zhang, K. Huang, X. Hou, and C. Liu, "Learning locality preserving graph from data," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2088–2098, Nov. 2014.
[48] Z. Zhang, L. Ladický, P. H. Torr, and A. Saffari, "Learning anchor planes for classification," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Granada, Spain, 2011, pp. 1611–1619.
[49] Z. Zhang, P. Sturgess, S. Sengupta, N. Crook, and P. Torr, "Efficient discriminative learning of parametric nearest neighbor classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2012, pp. 210–227.
[50] M. Zheng *et al.*, "Graph regularized sparse coding for image representation," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1327–1336, May 2011.

Junbiao Pang received the B.S. and M.S. degrees in computational fluid dynamics and computer science from the Harbin Institute of Technology, Harbin, China, in 2002 and 2004, respectively, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2011.

He was an Assistant Professor of Computer Science and Technology with the Beijing University of Technology (BJUT), Beijing, China, from 2011 to 2013. He is currently a Faculty Member with the College of Metropolitan Transportation, BJUT. His current research interests include multimedia and machine learning. He has authored or co-authored over 20 academic papers in publications such as the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, European Conference on Computer Vision (ECCV), International Conference on Computer Vision (ICCV), and ACM Multimedia.

Lei Qin received the B.S. and M.S. degrees in mathematics from the Dalian University of Technology, Dalian, China, in 1999 and 2002, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2008.

He is currently an Associate Professor with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include image/video processing, computer vision, and pattern recognition. He has authored or co-authored over 40 technical papers in the area of computer vision.

Dr. Qin is a Reviewer of the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and the IEEE TRANSACTIONS ON CYBERNETICS. He has served as a Technical Panel Community (TPC) member for various conferences, including ECCV, International Conference on Pattern Recognition, International Conference on Multimedia & Expro (ICME), Pacific Rim Symposium on Image and Video Technology (PSIVT), International Conference on Internet Multimedia Computing and Service, and PCM.

Chunjie Zhang received the B.E. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2006, and the Ph.D. degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2012.

He is currently a Faculty Member with the University of Chinese Academy of Sciences, Beijing. His current research interests include machine learning, image content analysis, and object categorization.

Weigang Zhang received the B.S. and M.S. degrees in computer science from the Harbin Institute of Technology (HIT), Harbin, China, in 2003 and 2005, respectively, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Technology.

He is an Associate Professor with the School of Computer Science and Technology, HIT Weihai, Weihai, China. His current research interests include multimedia computing and computer vision.

Qingming Huang (S'04–M'04–SM'08) received the B.S. degree in computer science and the Ph.D. degree in computer engineering from the Harbin Institute of Technology, Harbin, China, in 1988 and 1994, respectively.

He is currently a Professor with the University of the Chinese Academy of Sciences (CAS), Beijing, China, and an Adjunct Professor with the Institute of Computing Technology, CAS, Beijing, and the Beijing University of Technology, Beijing. His current research interests include multimedia computing, image processing, computer vision, pattern recognition, and machine learning. He has authored or co-authored over 200 academic papers in prestigious international journals, including the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and the IEEE TRANSACTIONS ON IMAGE PROCESSING, and top-level conferences, such as ACM Multimedia, ICCV, CVPR, ECCV, and VLDB.

Dr. Huang is an Associate Editor of *Acta Automatica Sinica* and a reviewer of various international journals, including the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and the IEEE TRANSACTIONS ON IMAGE PROCESSING. He has served as the Program Chair, the Track Chair, the Area Chair, and a TPC member for various conferences, including ACM Multimedia, CVPR, ICCV, ICME, and PSIVT.

Baocai Yin received the M.S. and Ph.D. degrees in computational mathematics from the Dalian University of Technology, Dalian, China, in 1988 and 1993, respectively.

He is currently a Professor with the Beijing University of Technology (BJUT), Beijing, China. He is also the Director of the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing, and the Dean of the College of Metropolitan Transportation, BJUT. His current research interests include multimedia, image processing, computer vision, and pattern recognition. He has authored or co-authored over 200 academic papers in prestigious international journals, including the IEEE TRANSACTIONS ON MULTIMEDIA and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and top-level conferences, such as InterNational conFerence on Computer coMmunications and ACM Special Interest Group on Computer GRAPHics.

Dr. Yin is currently an Editorial Member of the *Journal of Information and Computational Science* (USA).