Image Quality Assessment Using Kernel Sparse Coding

Zihan Zhou, Jing Li, Yuhui Quan* and Ruotao Xu

***Abstract*—One key in image quality assessment (IQA) is the design of image representations that can capture the changes of image structures caused by distortions. Recent studies show that sparse coding has emerged as a promising approach to analyzing image structures for IQA. However, existing sparse-coding-based IQA approaches use linear coding models, which ignore the nonlinearities of manifolds of image patches and thus cannot analyze complex image structures well. To overcome such a weakness, in this paper, we introduce nonlinear sparse coding to IQA. A kernel dictionary construction scheme is proposed, which combines analytic dictionaries and learnable dictionaries to guarantee both the stability and effectiveness of kernel sparse coding in the context of IQA. Built upon the kernel dictionary construction, an effective full-reference IQA metric is developed. Benefiting from the considerations on nonlinearities during sparse coding, the proposed IQA metric not only characterizes image distortions better, but also achieves improvement on the consistency with subjective perception, when compared to the metrics built upon linear sparse coding. Such benefits are demonstrated with the experimental results on eight benchmark datasets in terms of common criteria.**

***Index Terms*—Image quality assessment, Sparse representation, Kernel sparse coding, Dictionary learning**

I. Introduction

In the field of multimedia, Image Quality Assessment (IQA) refers to the task of automatically measuring the visual quality of an image by well-designed computational models, which plays an important role in visual data processing. In image compression, IQA can be used to derive constraints for balancing the image quality and the compression rate [1]. Similarly, it can be used to win the trade-off between the visual imperceptibility and embedding capacity of an image watermarking algorithm [2]. IQA can also guide the model design in image restoration and generation [3], [4], or the optimization of parameters in image enhancement [5]. In addition, IQA is helpful in image communication for optimizing the performance of encoders and decoders [6].

This work was supported in part by National Natural Science Foundation of China under Grants 61872151 and U1611461, in part by Natural Science Foundation of Guangdong Province under Grants 2017A030313376 and 2020A1515011128, and Fundamental Research Funds for Central Universities of China under Grant x2js-D2181690.

Zihan Zhou, Yuhui Quan and Ruotao Xu are with the School of Computer Science and Engineering at South China University of Technology, Guangzhou 510006, China, as well as with the Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, Guangzhou 510006, China (e-mail: cszzh@mail.scut.edu.cn, csyhquan@scut.edu.cn, xu.ruotao@mail.scut.edu.cn).

Jing Li is with the Moku Lab, Alibaba Group, Beijing 100016, China (e-mail: lj225205@alibaba-inc.com).

Asterisk indicates the corresponding author.

Based on how much information about the reference image (*i.e.* corresponding pristine image) is available during IQA, the existing IQA approaches (metrics) can be divided into three types: full-reference (FR), no-reference (NR) and reduced-reference (RR). In this paper, we focus on FR-IQA. Concretely, the FR-IQA methods (*e.g.* [7]–[12]) are designed for the scenarios where the original image that often has very high quality is given as the reference for estimating the quality of its distorted version. Such scenarios are often seen in image processing, such as image compression, image watermarking, as well as the training stages of learning-based image recovery methods; see *e.g.* [1], [4], [5].

The early FR-IQA metrics are based on pixel-wise difference, *e.g.* PSNR and MSE. However, these metrics cannot reveal the small image distortions that human eyes are insensitive to. There are mainly two kinds of strategies in existing methods for designing FR-IQA metrics with improvement. The first kind is the model-based strategy which uses well-established mathematical tools or computational models to derive a robust metric with certain mathematical properties. For instance, the VIF metric [11] is built upon the Gaussian scale mixture (GSM) model in wavelet domain together with the analysis tools from information theory. The second kind is simulating some mechanisms of conscious perception in the Human Visual System (HVS), which is referred to as the HVS-inspired strategy. One classic metric of this kind is the SSIM index [8], which extracts the structural information of images to simulate how human process visual scenes.

A. Motivations of Studying Sparse-Coding-Based IQA

In recent years, sparse coding has emerged as a promising approach for IQA; see *e.g.* [13]–[15]. One advantage of sparse coding is that it enjoys both the benefits of model-based approaches and HVS-inspired approaches. Not only with useful mathematical properties, sparse coding also has strong biological motivations.

From the computational perspectives, sparse coding has been proven theoretically and mathematically to be a natural and effective framework for characterizing the low-dimensional manifold of image data. An image patch, expressed as an array, can be viewed as a point in a linear space of very high dimensionality. It is widely accepted that the set of non-distorted natural image patches are concentrated on some low-dimensional subspaces in such a high-dimensional linear space [16], [17]. The geometries of such low-dimensional subspaces indeed encode possible image structures, and image distortions may be well characterized by the displacement of

the distorted image patches within or around such a subspace. Therefore, sparse coding is undoubtedly useful for IQA. In addition, using sparse coefficients as features can reduce the storage or bandwidth usage in some scenarios.

From the biological perspectives, sparse coding benefits IQA as follows. In the procedure of visual signal perception in HVS, images are first projected onto the retina, and then the generated visual signal is transmitted to the primary visual cortex (also called V1) through the Lateral Geniculate Nucleus (LGN) for visual abstraction. There are massive experimental evidences indicating that different neurons of retina and LGN can be activated in various situations, which can be accounted by the principles of parsimony and redundancy reduction [18]. Indeed, such two principles can be well exploited by sparse coding, in the sense that only a few vectors (*i.e.* atoms in the dictionary) are activated in sparse representation. Furthermore, learning an over-complete dictionary with the sparsity prior can imitate the properties, such as localization, orientation, band-pass and sparse activation, of the receptive field of simple cells in V1 [12], [19]. It can also provide good quantitative predictions (*i.e.* the non-zero values of sparse coefficients) that are often considered to be consistent with the measurements from V1 [20].

Inspired by the advantages and potentials of sparse coding for IQA, in this paper, we investigate the exploitation of sparse coding for FR-IQA and make a further step along the line of related research.

B. Basic Ideas of Using Kernel Sparse Coding for IQA

The existing sparse-coding-based IQA methods (*e.g.* [13]–[15]) apply conventional sparse coding to both the reference and distorted images, and then estimate the visual quality of the distorted image by comparing the resulting sparse codes. The sparse coding models used in these methods are linear, which assume image patches lie in some low-dimensional linear subspaces with Euclidean geometry. This assumption is not effective for IQA, as many studies have shown that real-life images usually exhibit high nonlinearities [21], [22] and the patches of such images tend to lie on some low-dimensional nonlinear manifolds instead of linear subspaces embedded in the high-dimensional linear space [23]. As a result, the metrics defined on the results of linear sparse coding cannot reveal the distance between the reference image (patches) and distorted image (patches) along the non-Euclidean geometric structures of the nonlinear manifold. In other words, the existing sparse-coding-based approaches cannot exploit the complex structures of images well.

To overcome the weakness of linear sparse coding models, in this work, we introduce nonlinear sparse coding to IQA. Motivated by the recent advances of kernel sparse coding beyond conventional sparse coding in analyzing non-linear data [22], [24], we develop some effective kernel sparse coding models on image patches, together with a kernel dictionary construction scheme designed for the IQA task. Based on the coding coefficients and reconstruction errors from our models, we propose an effective sparse-coding-based IQA method which can exploit the intrinsic geometric structures of image data effectively.

Due to the irreversibility of kernel mapping, existing kernel sparse coding methods are mainly for pattern recognition instead of image processing. As a result, these methods do not consider using analytic dictionaries. However, many studies (*e.g.* [25], [26]) have shown that analytic dictionaries, such as Gabor and wavelet, are very useful for IQA. To improve the effectiveness of kernel sparse coding for IQA, our scheme uses an analytic dictionary and a learned dictionary for kernel sparse coding respectively and combine their results, by which the advantages from both dictionaries can be enjoyed.

In addition, most existing kernel dictionary learning approaches define the kernel dictionary by the linear combination of the training samples in the kernel-associated implicit space. When the data of dictionary learning is insufficient to span a meaningful space for the test data, which is likely to occur in IQA, the stability and effectiveness of kernel sparse coding will decrease [27]. Our scheme addresses this issue by fixing some atoms in the learned dictionary to be the analytic ones during learning, so as to ensure the minimum span of the learned dictionary. Such analytic atoms can also be regarded as the clean ones that can reduce the sensitivity of the kernel mapping to the noises in training data.

C. Contributions

The contributions of this work are summarized as follows:

- We introduce kernel sparse representation with a nonlinear coding model for IQA. Compared to the linear coding models used in the existing methods [12]–[15], [28]–[30], the nonlinear one is more effective in revealing the nonlinear structures of images, leading to the improved representations of image patches and a better IQA metric.
- We propose to use both coding coefficients and reconstruction errors in sparse coding for constructing the IQA metric. Compared to the existing methods [13], [14] which only utilize coding coefficients, ours can exploit additional information from sparse coding.
- A kernel dictionary construction scheme is developed, which includes a learning-free dictionary with analytic form and a learnable dictionary with partially-fixed analytic atoms. Such a scheme differs from the ones used in existing kernel sparse coding approaches, and it can improve the effectiveness of kernel sparse coding in IQA.
- The proposed kernel dictionary construction scheme results in several non-trivial optimization problems, for which we develop effective and efficient numerical solvers.

D. Notations

Throughout the paper, unless specified, bold upper letters are used for matrices, bold lower letters for column vectors, light letters for scalars, and calligraphic letters for sets. The t_0-th element of a sequence $\{\mathbf{y}^{(t)}\}_{t\in\mathcal{N}}$ is denoted by $\mathbf{y}^{(t_0)}$. The i-th element of a column vector $\mathbf{x}$ is denoted by $\mathbf{x}(i)$. Given $\Omega \subset \{1, \cdots, M\}$ and $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_M]$, $\mathbf{X}_\Omega$ or $\mathbf{X}(\Omega)$ denotes the sub-matrix of $\mathbf{X}$ formed by deleting the columns of $\mathbf{X}$ whose indexes are not in Ω. The ℓ_0 pseudo-norm and ℓ_2 norm of a vector $\mathbf{x}$ are denoted by $\|\mathbf{x}\|_0$ and $\|\mathbf{x}\|$ (or $\|\mathbf{x}\|_2$) respectively. The Frobenius norm of a matrix $\mathbf{X}$ is denoted by

$\|\mathbf{X}\|_{\mathrm{F}}$. Besides, let $\mathbf{I}$ and $\mathbf{0}$ denote the identity matrix and the zero matrix with appropriate sizes respectively.

II. Related Work and Preliminaries

A. Full-Reference Image Quality Assessment

1) Pixel-difference-based approaches: Early FR-IQA metrics measure the image quality based on the pixel-by-pixel difference between the distorted and reference images. The representative approaches include PSNR, MSE, etc. These approaches are simple and widely used in image processing. Nevertheless, they totally ignore the correlations among adjacent pixels and how HVS processes images, with limited accuracy across various types of distortions.

2) Model-based approaches: There are many FR-IQA metrics built upon mathematical models. Lai *et al.* [31] proposed to encode the correlations among adjacent pixels by applying Haar wavelet transform to images. Sheikh *et al.* [11] proposed to model an image by the GSM model of wavelet coefficients and measure the difference between models based on information theory. Instead of in the wavelet domain, Zhang *et al.* [32] proposed to model images with the GSM model in the image domain directly. Their metric is also built upon the analysis tools in information theory. These early model-based approaches, although partly related to HVS's mechanisms, are mainly focused on the information itself and lack of neurobiological support. In recent years, there are several metrics proposed for FR-IQA based on sparse coding, *e.g.* [12]–[14], [29], [33]. These methods not only have well-established models from sparse representation theory, but also have strong support from neurobiology. Since our metric is a sparse-coding-based one, we give a detailed review on sparse-coding-based metrics in a separate subsection later.

It is worth mentioning that, taking the advance of deep learning, some approaches (*e.g.* [34], [35]) have been proposed which use deep neural network models to learn an IQA metric by the end-to-end supervised manner. Though with very impressive results, one main challenge to these methods is their heavy requirement on the large amount of training samples. In fact, images with subjective scores in existing IQA datasets may be inadequate for the effective training of deep models. Many recent studies (*e.g.* [35]) have made effort on addressing these challenges.

3) HVS-inspired approaches: The design of many FR-IQA metrics are inspired by certain mechanisms of HVS. Assuming that HVS is highly competent to extract the structural information, the SSIM [8] index measures image quality degradation by the change of structural information. To capture the structural changes in multiple scales, the MS-SSIM [7] index extended SSIM through a process of multiple stages of subsampling and multi-scale processing. For better capturing local distortions, the IW-SSIM [9] uses local information content as perceptual weights for measuring local image distortion.

Many methods extract structural information from image gradients/edges. Zhang *et al.* [10] proposed to quantify the structural similarity based on the features extracted from image gradients and phase congruency. Liu *et al.* [36] proposed to use gradient similarity to encode the changes of image structures and the image contrast. Xue *et al.* [37] exploited pixel-wise gradient magnitude similarity to measure the global distortion. Jin *et al.* [38] proposed to attain perceptual gradients for improvement, which is done by automatically selecting the pixel-wise gradient directions with maximum changing rates in the reference image. Considering the hypothesis that the visual masking effect has an important impact on the perception of HVS, Liu *et al.* [36] combined visibility thresholding with the gradient similarity. Shi *et al.* [39] proposed a visual metric based on edge-feature-based segmentation, in which the low-level features of segmented parts are pooled as the final score.

B. Sparse Coding for Image Quality Assessment

The sparse-coding-based approaches compute the sparse coefficients of images or image patches under some dictionaries and then use them as the features to estimate the visual quality scores. Chang *et al.* [12] proposed to acquire the sparse features by a feature detector trained on natural images with independent component analysis. Guha *et al.* [13] proposed to learn an individual dictionary from each reference image and use it for the sparse coding of the corresponding reference and distorted images. Such a scheme may be time-consuming since the dictionary learning process is run for every input reference image. To overcome this weakness, Li *et al.* [14] proposed to pre-learn a universal dictionary from a set of clear natural images instead of learning individual ones. This method is extended in [29] to better utilize the color information of image. Ahar *et al.* [33] proposed to conduct ranking on the amplitudes of the sparse coefficients under Fourier bases, and then use a complex correlation metric that assesses the correspondence between the ranked coefficient amplitude profiles of the reference and distorted images.

The aforementioned sparse-coding-based approaches are for FR-IQA. There are also some developed for RR-IQA and NR-IQA. Liu *et al.* [15], [40] assumed the prediction manner of the internal generative model in free-energy principle with sparse representation, based on which they proposed an RR-IQA metric that only extracts a single scalar (*i.e.* entropy of the prediction residuals) from the reference image. This approach can be roughly regarded as NR. See also [30], [41], [42] for the recent development of NR-IQA related to sparse coding.

Though the existing sparse-coding-based approaches have achieved impressive results, they are all based on linear coding models which cannot well handle the nonlinearities of data [22], [24], [27]. In this paper, we explore the exploitation of nonlinearities of image data for sparse-coding-based FR-IQA and introduce kernel sparse coding for better modeling.

C. Kernel Sparse Coding

Sparse coding is a popular tool for discovering the low-dimensional structures of high-dimensional data. Traditional sparse coding assumes a signal $\mathbf{y} \in \mathcal{R}^M$ can be expressed by $\mathbf{y} \approx \mathbf{D}\mathbf{c}$ with a dictionary $\mathbf{D} \in \mathcal{R}^{M\times K}$ and a sparse coefficient vector $\mathbf{c} \in \mathcal{R}^K$. The sparsity pattern of $\mathbf{c}$ combined with the dictionary can reveal the underlying structures of data and yield some compact representation of data. The dictionary for sparse coding is often learned from data to

improve the effectiveness of sparse representation, which is often formulated as the following minimization problem:

$$\min_{\mathbf{D},\{\mathbf{c}_i\}} \sum_{i=1}^{L} \|\mathbf{y}_i - \mathbf{D}\mathbf{c}_i\|_2^2, \tag{1}$$

subject to $\|\mathbf{c}_i\|_0 \leq T$ and $\|\mathbf{d}_j\|_2 = 1,\ 1 \leq j \leq K$, where $\{\mathbf{y}_i\}_{i=1}^L \subset \mathcal{R}^M$ is a set of input signals, and T is the sparsity degree. The sparse coding using (1) is called linear sparse coding as it uses the linear reconstruction model $\mathbf{y} \approx \mathbf{D}\mathbf{c}$.

Linear sparse coding can well analyze the data lying on low-dimensional linear subspaces. However, it does not work for the data lying on low-dimensional nonlinear manifolds; see *e.g.* [22], [24], [27]. Kernel sparse coding remedies this problem by mapping the data to a high-dimensional (or infinite-dimensional) implicit space with some kernel function and then conducting linear sparse coding in the implicit space. The basic idea of kernel sparse coding is that nonlinear data are likely to exhibit linear structures after being non-linearly mapped to some higher-dimensional spaces. Let $\phi(\cdot) : \mathcal{M} \to \mathcal{H}$ to be a nonlinear mapping from a Riemannian manifold $\mathcal{M} \subset \mathcal{R}^M$ into a high-dimensional or infinite-dimensional dot product space $\mathcal{H}$. Instead of using the linear model $\mathbf{y} \approx \mathbf{D}\mathbf{c}$, kernel sparse coding assumes $\phi(\mathbf{y}) \approx \bar{\mathbf{D}}\mathbf{c}$ which is a linear model in the space $\mathcal{H}$, where $\bar{\mathbf{D}}$ denotes a dictionary in $\mathcal{H}$. In general, kernel sparse coding solves the minimization model

$$\min_{\bar{\mathbf{D}},\{\mathbf{c}_i\}} \sum_{i=1}^{L} \|\phi(\mathbf{y}_i) - \bar{\mathbf{D}}\mathbf{c}_i\|_2^2. \tag{2}$$

One key in developing kernel sparse coding methods is how to define $\bar{\mathbf{D}}$ such that the kernel trick can be efficiently used to solve the model in (2) without involving $\phi(\cdot)$. There have been some works on the construction of the kernel dictionary $\bar{\mathbf{D}}$. In [22], [24], [43], the kernel dictionary is defined by the linear combination of $\{\phi(\mathbf{y}_i)\}_i$, while the coefficients of the combination are learned from data. To further improve the efficiency, sparsity constraints are imposed on the coefficients of linear combination in [44]. In [27], an equiangular kernel dictionary construction scheme is proposed to control the stability of sparse coding in infinite-dimensional spaces. To alleviate the computational cost in computing the gram matrix in kernel representation, a linearized kernel dictionary learning scheme is proposed in [45]. Since the kernel mapping ϕ is irreversible which is inapplicable to image processing that needs to solve $\phi^{-1}(\bar{\mathbf{D}}\mathbf{c})$, all these methods only study the construction of kernel dictionary for recognition. In this paper, we investigate the construction of kernel dictionary for IQA.

III. Proposed Method

The proposed IQA metric is referred to as *KSCM (Kernel Sparse Coding based Metric)*, whose flowchart is shown in Fig. 1. It mainly contains four steps. Firstly, a kernel dictionary is prepared. Secondly, the features of the reference image including sparse coefficients and reconstruction errors are extracted via kernel sparse coding on image patches. Then, the shared dictionary is obtained with the constraint of the non-zero locations. Thirdly, the features of the distorted image are

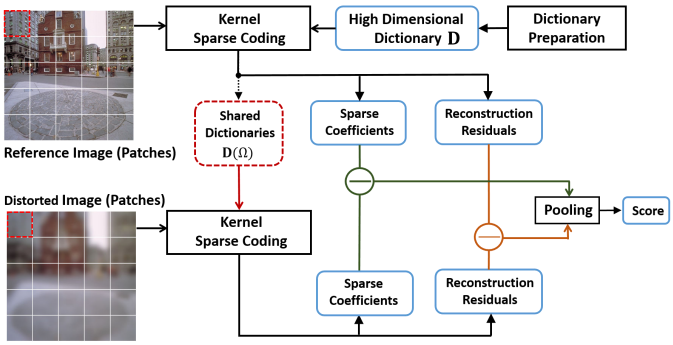

Fig. 1: Flowchart of proposed KSCM.

computed via kernel sparse coding on image patches. Note that the kernel sparse coding model in this stage involves additional constraints from the reference features in the second step, which distinguishes it from the sparse coding model on the reference image. Finally, the coefficients and residuals of the reference and distorted images are pooled to obtain the quality score. In the next, we will detail each step in KSCM. For notational convenience, we denote the patch size in kernel sparse coding by $\sqrt{B} \times \sqrt{B}$ where B is a perfect square.

A. Kernel Dictionary Construction Scheme

The dictionary in kernel sparse coding is crucial to both the effectiveness and efficiency of the coding process. In traditional linear sparse coding of images, there are two types of dictionaries mainly used:

- Analytic dictionaries such as wavelets, which are mathematically derived and mainly used for image processing;
- Data-driven dictionaries which are learned or constructed from data, with applications to both image processing and image recognition.

For kernel sparse coding, due to the irreversibility of kernel mapping, the sparse coefficients under the kernel dictionary cannot be transformed to original image space, implying kernel sparse coding cannot be applied to image processing. As a result, most existing approaches of kernel sparse coding focus on image recognition and employ data-driven dictionaries. However, for IQA the analytic dictionaries are very useful, their effectiveness have been demonstrated in many studies (*e.g.* [25], [26], [46], [47]). Thus, we propose to use two kernel dictionaries, an analytic one $\bar{\mathbf{D}}_1$ and a learnable one $\bar{\mathbf{D}}_2$, to conduct kernel sparse coding respectively, which can encode the image structures from different aspects. Suppose $\bar{\mathbf{D}}_1$ and $\bar{\mathbf{D}}_2$ are in the dot product space $\mathcal{H}$. We construct $\bar{\mathbf{D}}_1, \bar{\mathbf{D}}_2$ with the following scheme.

Let $k(\cdot,\cdot)$ denote the kernel which is associated with ϕ by $k(\mathbf{x},\mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}$ for $\mathbf{x},\mathbf{y} \in \mathcal{M}$. In the matrix form, we denote $\Phi(\mathbf{X}) = [\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_L)]$ for any $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_L]$, and $\mathbf{K}(\mathbf{X}_1, \mathbf{X}_2) = \Phi(\mathbf{X}_1)^\top \Phi(\mathbf{X}_2)$ for any $\mathbf{X}_1, \mathbf{X}_2 \in \mathcal{M}$. In practice, $k(\mathbf{x},\mathbf{y})$ is computed by some simple functions and the kernel trick without using $\phi(\cdot)$ explicitly. Since $\mathcal{H}$ is implicit in kernel mapping, the analytic form of $\bar{\mathbf{D}}_1$ is inaccessible. To address this issue, we define

$$\bar{\mathbf{D}}_1 = \Phi(\mathbf{W}), \tag{3}$$

where $\mathbf{W} \in \mathcal{R}^{B\times N}$ is an analytic dictionary with N atoms in the original space. In practice, we set $\mathbf{W}$ to be the Haar framelet dictionary [48]. As shown in the next subsections, such a form of $\bar{\mathbf{D}}_1$ allows the use of kernel trick in the sparse coding process. Regarding the construction of the data-driven dictionary $\bar{\mathbf{D}}_2$, we propose the following scheme. Let $\{\mathbf{y}_i\}_{i=1}^{Z} \subset \mathcal{R}^B$ be a set of vectorized image patches sampled from the training images (*i.e.* a set of predefined clear natural images), and denote $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_Z]$. In order to apply the kernel trick to both dictionary learning and sparse coding, the dictionary $\bar{\mathbf{D}}_2$ in $\mathcal{H}$ is defined as

$$\bar{\mathbf{D}}_2 = [\Phi(\mathbf{Y})\mathbf{A}, \Phi(\mathbf{G})], \tag{4}$$

where $\mathbf{A} \in \mathcal{R}^{Z\times K_1}$ is a coefficient matrix to be learned and $\mathbf{G} \in \mathcal{R}^{B\times K_2}$ is the concatenation of some well-known analytic dictionaries in the low-dimensional space $\mathcal{R}^B$.

There are two sub-dictionaries in $\bar{\mathbf{D}}_2$. In the first sub-dictionary $\Phi(\mathbf{Y})\mathbf{A}$, each dictionary atom in the space $\mathcal{H}$ is expressed by the linear combination of all mapped patches $\Phi(\mathbf{Y})$, and the dictionary learning problem is turned into optimizing the coefficient matrix $\mathbf{A}$. Note that $\Phi(\mathbf{Y})\mathbf{A}$ is often used as the kernel dictionary in many existing approaches to kernel sparse coding (*e.g.* [22], [24], [43]). However, in real applications the scale of the training data $\mathbf{Y}$ cannot be too large with the considerations on computational and storage burdens. As a result, the space spanned by $\Phi(\mathbf{Y})$ may be insufficient to effectively represent new input samples. Furthermore, the kernel mapping ϕ may be sensitive to the noises in the training samples, making the dictionary unstable.

Regarding the above issues, we introduce the other sub-dictionary $\Phi(\mathbf{G})$ which is an analytic dictionary to improve the quality of the dictionary $\bar{\mathbf{D}}_2$. The benefits of introducing $\Phi(\mathbf{G})$ are two-fold. First, the analytic atoms are noiseless, which decreases the sensitivity of $\bar{\mathbf{D}}_2$. Second, it is shown in [27] that the expressive power of a kernel dictionary as well as the stability of kernel sparse coding is related to the incoherence of the dictionary atoms in the original space. Since an analytic dictionary often forms an orthogonal basis or an equiangular tight frame whose atoms have very low incoherence, the use of $\Phi(\mathbf{G})$ can increase the effectiveness of $\bar{\mathbf{D}}_2$. In practice, we set $\mathbf{G}$ to be a Discrete Fourier Transform (DFT) dictionary.

To learn the dictionary $\bar{\mathbf{D}}_2$, we solve the following minimization problem:

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{C}} \|\Phi(\mathbf{Y}) - [\Phi(\mathbf{Y}), \Phi(\mathbf{G})] \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{C}\|_{\mathrm{F}}^2, \\ &\text{s.t.} \quad \|\mathbf{c}_i\|_0 \le T, \ \|\Phi(\mathbf{Y})\mathbf{a}_j\|_2 = 1, \ \forall i, j, \end{aligned} \tag{5}$$

where $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_{K_1}]$ is the dictionary coefficient matrix in $\bar{\mathbf{D}}_2$, $\mathbf{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_Z] \in \mathcal{R}^{(K_1+K_2)\times Z}$ is the sparse coding matrix, T is the sparsity degree, and the normalization constraint is for avoiding trivial solutions with positive scaling. The problem of (5) can be irrelevant to the explicit definition of $\Phi(\cdot)$ with the use of kernel trick. Let $\mathbf{S} = [\mathbf{Y}, \mathbf{G}]$ and $\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$. By definition we have $\Phi(\mathbf{S}) = [\Phi(\mathbf{Y}), \Phi(\mathbf{G})]$ and $\bar{\mathbf{D}}_2 = \Phi(\mathbf{S})\bar{\mathbf{A}}$. By using $\|\mathbf{X}\|_{\mathrm{F}}^2 = tr(\mathbf{X}^\top\mathbf{X})$ for any $\mathbf{X}$, we can rewrite the objective function and normalization constraints in (5) into the kernel-based forms as follows:

$$\begin{aligned} ||\Phi(\mathbf{Y}) - \Phi(\mathbf{S})\bar{\mathbf{A}}\mathbf{C}||_{\mathrm{F}}^2 &= tr(\mathbf{K}(\mathbf{Y},\mathbf{Y})) - tr(2\mathbf{K}(\mathbf{Y},\mathbf{S})\bar{\mathbf{A}}\mathbf{C}) \\ &\quad + tr(\mathbf{C}^\top\bar{\mathbf{A}}^\top\mathbf{K}(\mathbf{S},\mathbf{S})\bar{\mathbf{A}}\mathbf{C}), \end{aligned} \tag{6}$$

$$||\Phi(\mathbf{Y})\mathbf{a}_j||_2^2 = \mathbf{a}_j^\top\mathbf{K}(\mathbf{Y},\mathbf{Y})\mathbf{a}_j, \forall j, \tag{7}$$

where $\mathbf{K}(\mathbf{S},\mathbf{S}) = \Phi(\mathbf{S})^\top\Phi(\mathbf{S})$ is a kernel matrix that can be computed by some kernel function without defining $\phi(\cdot)$.

B. Kernel Dictionary Learning Algorithm

The problem of (5) is a challenging non-smooth and non-convex optimization problem. We solve the problem with an alternating iterative scheme, which alternatively updates the unknown variables $\mathbf{C}$ and $\mathbf{A}$ in the model one at a time, breaking the original problem into two simpler ones. We first initialize the dictionary coefficient matrix $\mathbf{A} = \mathbf{A}^{(0)}$ and start with $t = 1$. Then the update scheme is as follows:

1) Update of sparse codes: At the beginning of the t-th iteration, we fix $\mathbf{A} = \mathbf{A}^{(t-1)}$ and calculate $\mathbf{C}^{(t)}$ by

$$\begin{aligned} &\mathbf{C}^{(t)} \in \arg\min_{\mathbf{C}} \|\Phi(\mathbf{Y}) - [\Phi(\mathbf{Y}), \Phi(\mathbf{G})] \begin{bmatrix} \mathbf{A}^{(t-1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{C}\|_{\mathrm{F}}^2, \\ &\text{s.t.} \quad \|\mathbf{c}_i\|_0 \le T, \ \forall i. \end{aligned} \tag{8}$$

By using $\Phi(\mathbf{S}) = [\Phi(\mathbf{Y}), \Phi(\mathbf{G})]$ and $\bar{\mathbf{A}}^{(t-1)} = \begin{bmatrix} \mathbf{A}^{(t-1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$, we can rewrite (8) as

$$\begin{aligned} &\mathbf{C}^{(t)} \in \arg\min_{\mathbf{C}} \|\Phi(\mathbf{Y}) - \Phi(\mathbf{S})\bar{\mathbf{A}}^{(t-1)}\mathbf{C}\|_{\mathrm{F}}^2, \\ &\text{s.t.} \quad \|\mathbf{c}_i\|_0 \le T, \ \forall i, \end{aligned} \tag{9}$$

which is a kernel sparse approximation problem that can be solved by KOMP (Kernelized Orthogonal Pursuit Matching) [24]. The KOMP algorithm involves the calculation of the inverse of kernel-related matrices, which may be time-consuming. For acceleration, we propose another solver based on projected gradient [27], called kernelized projected gradient, which updates $\mathbf{C}$ by

$$\mathbf{C}^{(t)} \in Proj^{\mathcal{P}}(\mathbf{C}^{(t-1)} - \tau_t\nabla_{\mathbf{C}}h(\mathbf{C}^{(t-1)}, \mathbf{A}^{(t-1)})), \tag{10}$$

where $Proj^{\mathcal{P}}(\cdot)$ denotes the projection onto the constraint set $\mathcal{P} = \{\mathbf{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_Z] : \|\mathbf{c}_i\|_0 \le T\}$, $h(\mathbf{C}, \mathbf{A}) = \|\Phi(\mathbf{Y}) - \Phi(\mathbf{S})\bar{\mathbf{A}}\mathbf{C}\|_{\mathrm{F}}^2$, and τ_t is the step size. By applying the kernel trick, we have

$$\nabla_{\mathbf{C}}h(\mathbf{C}, \mathbf{A}) = \bar{\mathbf{A}}^\top\mathbf{K}(\mathbf{S},\mathbf{S})\bar{\mathbf{A}}\mathbf{C} - \bar{\mathbf{A}}^\top\mathbf{K}(\mathbf{S},\mathbf{Y}).$$

Then the solution of problem (8) is given by

$$\begin{aligned} \mathbf{C}^{(t)} = \mathcal{S}_T(&\mathbf{C}^{(t-1)} - \tau_t(\bar{\mathbf{A}}^{(t-1)})^\top\mathbf{K}(\mathbf{S},\mathbf{S})\bar{\mathbf{A}}^{(t-1)}\mathbf{C}^{(t-1)} \\ &+ \tau_t(\bar{\mathbf{A}}^{(t-1)})^\top\mathbf{K}(\mathbf{S},\mathbf{Y})), \end{aligned} \tag{11}$$

where $\mathcal{S}_T(\mathbf{X})$ keeps the T largest elements in each column of $\mathbf{X}$ in terms of magnitude.

2) Update of dictionary: At the t-th iteration, after $\mathbf{C}^{(t)}$ is updated, we fix $\mathbf{C} = \mathbf{C}^{(t)}$ and calculate $\mathbf{A}^{(t)}$ by

$$\begin{aligned}\mathbf{A}^{(t)} \in \arg\min_{\mathbf{A}} \ & \|\Phi(\mathbf{Y}) - [\Phi(\mathbf{Y}), \Phi(\mathbf{G})] \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{C}^{(t)}\|_{\mathrm{F}}^2, \\ \text{s.t.} \quad & \mathbf{a}_j^\top \mathbf{K}(\mathbf{S}, \mathbf{S}) \mathbf{a}_j = 1, \ \forall j, \end{aligned} \tag{12}$$

which is equivalent to

$$\begin{aligned}\mathbf{A}^{(t)} \in \arg\min_{\mathbf{A}} \ & \|\Phi(\mathbf{Y}) - \Phi(\mathbf{G})\mathbf{C}_2^{(t)} - \Phi(\mathbf{Y})\mathbf{A}\mathbf{C}_1^{(t)}\|_{\mathrm{F}}^2, \\ \text{s.t.} \quad & \mathbf{a}_j^\top \mathbf{K}(\mathbf{S}, \mathbf{S}) \mathbf{a}_j = 1, \ \forall j, \end{aligned} \tag{13}$$

where $\mathbf{C}_1^{(t)} \in \mathcal{R}^{K_1 \times Z}$, $\mathbf{C}_2^{(t)} \in \mathcal{R}^{K_2 \times Z}$, and $\mathbf{C}^{(t)} = [(\mathbf{C}_1^{(t)})^\top, (\mathbf{C}_2^{(t)})^\top]^\top$. The dictionary coefficient matrix $\mathbf{A}$ in (13) is updated column by column. Let $\mathbf{r}_i^\top$ indicate the i-th row of $\mathbf{C}_1^{(t)}$. Following the idea of K-SVD [49] and kernel K-SVD [24], on the calculation of $\mathbf{a}_j$, we fix all $\mathbf{a}_i$ for $i \neq j$ to be the previous estimate and rewrite the objective function in (13) as

$$\begin{aligned} & \|\Phi(\mathbf{Y}) - \Phi(\mathbf{G})\mathbf{C}_2^{(t)} - \Phi(\mathbf{Y})\mathbf{A}\mathbf{C}_1^{(t)}\|_{\mathrm{F}}^2 \\ =& \|\Phi(\mathbf{Y}) - \Phi(\mathbf{G})\mathbf{C}_2^{(t)} - \Phi(\mathbf{Y})(\mathbf{a}_j\mathbf{r}_j^\top + \sum_{i=1, i\neq j}^{K_1} \mathbf{a}_i\mathbf{r}_i^\top)\|_{\mathrm{F}}^2 \\ =& \|\Phi(\mathbf{Y})\mathbf{E}_j - \Phi(\mathbf{G})\mathbf{C}_2^{(t)} - \Phi(\mathbf{Y})\mathbf{a}_j\mathbf{r}_j^\top\|_{\mathrm{F}}^2, \end{aligned} \tag{14}$$

where $\mathbf{E}_j = \mathbf{I} - \sum_{i=1, i\neq j}^{K_1} \mathbf{a}_i\mathbf{r}_i^\top$.

Let $\mathbf{M}_j = \Phi(\mathbf{Y})\mathbf{E}_j - \Phi(\mathbf{G})\mathbf{C}_2^{(t)}$. The minimization of (14) implies $\Phi(\mathbf{Y})\mathbf{a}_j\mathbf{r}_j^\top$ is the rank-1 approximation of $\mathbf{M}_j$ with the Euclidean norm. As a result, $\Phi(\mathbf{Y})\mathbf{a}_j$ is set to the first left singular vector of $\mathbf{M}_j$, and $\mathbf{r}_j$ is set to the product of the largest singular value and the first right singular vector of $\mathbf{M}_j$. Note that the largest singular value and the first right singular vector of $\mathbf{M}_j$ can be calculated respectively from the largest eigenvalue and the first eigenvector of

$$\begin{aligned}\mathbf{M}_j^\top\mathbf{M}_j = & \mathbf{E}_j^\top\mathbf{K}(\mathbf{Y}, \mathbf{Y})\mathbf{E}_j + (\mathbf{C}_2^{(t)})^\top\mathbf{K}(\mathbf{G}, \mathbf{G})\mathbf{C}_2^{(t)} \\ & -\mathbf{E}_j^\top\mathbf{K}(\mathbf{Y}, \mathbf{G})\mathbf{C}_2^{(t)} - (\mathbf{C}_2^{(t)})^\top\mathbf{K}(\mathbf{G}, \mathbf{Y})\mathbf{E}_j. \end{aligned} \tag{15}$$

Let $\sigma_1(\mathbf{M}_j)$ denote the largest singular value of $\mathbf{M}_j$. By the definition of SVD (Singular Value Decomposition) we have

$$\mathbf{M}_j\mathbf{r}_j = \sigma_1^2(\mathbf{M}_j)\Phi(\mathbf{Y})\mathbf{a}_j. \tag{16}$$

By multiplying $\Phi(\mathbf{Y})$ on both sides on (16), we have

$$\Phi(\mathbf{Y})^\top\mathbf{M}_j\mathbf{r}_j = \sigma_1^2(\mathbf{M}_j)\Phi(\mathbf{Y})^\top\Phi(\mathbf{Y})\mathbf{a}_j, \tag{17}$$

which can be kernelized as

$$(\mathbf{K}(\mathbf{Y}, \mathbf{Y})\mathbf{E}_j - \mathbf{K}(\mathbf{Y}, \mathbf{G})\mathbf{C}_2^{(t)})\mathbf{r}_j = \sigma_1^2(\mathbf{M}_j)\mathbf{K}(\mathbf{Y}, \mathbf{Y})\mathbf{a}_j. \tag{18}$$

This is a linear system which can be solved by iterative methods when the scale of system is large.

C. Kernel Sparse Representation

Given a reference image $\mathbf{I}^{\mathrm{r}} \in \mathcal{R}^{M_1 \times M_2}$, we first sample L image patches denoted by $\{\mathbf{P}_i^{\mathrm{r}} \in \mathcal{R}^{\sqrt{B} \times \sqrt{B}}\}_{i=1}^L$ from $\mathbf{I}^{\mathrm{r}}$ using a sliding window with step size S, where $L = \lfloor \frac{M_1 - \sqrt{B} + 1}{S} \rfloor \times \lfloor \frac{M_2 - \sqrt{B} + 1}{S} \rfloor$. On the distorted image $\mathbf{I}^{\mathrm{d}} \in \mathcal{R}^{M_1 \times M_2}$, the same sampling process is done and we can collect L image patches from $\mathbf{I}^{\mathrm{d}}$, which are denoted by $\{\mathbf{P}_i^{\mathrm{d}} \in \mathcal{R}^{\sqrt{B} \times \sqrt{B}}\}_{i=1}^L$. The patches $\{\mathbf{P}_i^{\mathrm{r}}\}_{i=1}^L$ and $\{\mathbf{P}_i^{\mathrm{d}}\}_{i=1}^L$ are ordered respectively such that $\mathbf{P}_i^{\mathrm{r}}$ and $\mathbf{P}_i^{\mathrm{d}}$ correspond to the same spatial location in images for all i. The extracted patches contain rich local structures of the reference and distorted images. In order to measure the visual quality of image $\mathbf{I}^{\mathrm{d}}$, we resort to analyzing the difference between each patch pair $(\mathbf{P}_i^{\mathrm{r}}, \mathbf{P}_i^{\mathrm{d}})$ in some specific domain which is correlated to visual perception.

It has been shown in many studies (*e.g.* [21], [23], [50]) that local image patches tend to align on some nonlinear manifolds. To effectively analyze the patch pairs $\{(\mathbf{P}_i^{\mathrm{r}}, \mathbf{P}_i^{\mathrm{d}})\}_{i=1}^L$ in the presence of nonlinearities, we employ kernel sparse coding to represent each pair of patches. The outline of kernel sparse coding in the proposed method is shown in Fig. 2. Let $\mathbf{y}_i^{\mathrm{r}} \in \mathcal{R}^B$ and $\mathbf{y}_i^{\mathrm{d}} \in \mathcal{R}^B$ denote the vectorized version of the patches $\mathbf{P}_i^{\mathrm{r}}$ and $\mathbf{P}_i^{\mathrm{d}}$ respectively. For the reference image, the kernel sparse representations $\mathbf{c}_i^{\mathrm{r}_1}$ and $\mathbf{c}_i^{\mathrm{r}_2}$ of each patch $\mathbf{y}_i^{\mathrm{r}}$ are computed using the dictionaries $\bar{\mathbf{D}}_1$ defined by (3) and $\bar{\mathbf{D}}_2$ defined by (4) respectively. The details are as follows:

- Kernel sparse representation using $\bar{\mathbf{D}}_1$:

$$\mathbf{c}_i^{\mathrm{r}_1} \in \arg\min_{\mathbf{c}} \ \|\phi(\mathbf{y}_i^{\mathrm{r}}) - \Phi(\mathbf{W})\mathbf{c}\|_{\mathrm{F}}^2, \ \text{s.t.} \ \|\mathbf{c}\|_0 \leq T_1, \tag{19}$$

- Kernel sparse representation using $\bar{\mathbf{D}}_2$:

$$\mathbf{c}_i^{\mathrm{r}_2} \in \arg\min_{\mathbf{c}} \ \|\phi(\mathbf{y}_i^{\mathrm{r}}) - \Phi(\mathbf{S})\bar{\mathbf{A}}\mathbf{c}\|_{\mathrm{F}}^2, \ \text{s.t.} \ \|\mathbf{c}\|_0 \leq T_2, \tag{20}$$

where T_1 and T_2 are the predefined sparsity degrees. The problem (19) and (20) share similar forms with (9), which can be solved by KOMP [24] or the kernelized projected gradient method proposed in Section III-B.

Let $\Omega_i^1 = \{j : \mathbf{c}_i^{\mathrm{r}_1}(j) \neq 0\}$ and $\Omega_i^2 = \{j : \mathbf{c}_i^{\mathrm{r}_2}(j) \neq 0\}$ denote the support (*i.e.* positions of nonzeros) of $\mathbf{c}_i^{\mathrm{r}_1}$ and $\mathbf{c}_i^{\mathrm{r}_2}$ respectively. The kernel sparse representations $\mathbf{c}_i^{\mathrm{r}_1}$ and $\mathbf{c}_i^{\mathrm{r}_2}$ of the reference patch $\mathbf{y}_i^{\mathrm{r}}$ satisfy

$$\phi(\mathbf{y}_i^{\mathrm{r}}) = \Phi(\mathbf{W}_{\Omega_i^1})\mathbf{c}_i^{\mathrm{r}_1}(\Omega_i^1) + \mathbf{n}_i^{\mathrm{r}_1}, \tag{21}$$

$$\phi(\mathbf{y}_i^{\mathrm{r}}) = \Phi(\mathbf{S})\bar{\mathbf{A}}_{\Omega_i^2}\mathbf{c}_i^{\mathrm{r}_2}(\Omega_i^2) + \mathbf{n}_i^{\mathrm{r}_2}, \tag{22}$$

where $\mathbf{n}_i^{\mathrm{r}_1}$ and $\mathbf{n}_i^{\mathrm{r}_2}$ are the representation errors of $\mathbf{y}_i^{\mathrm{r}}$, $\Phi(\mathbf{W}_{\Omega_i^1}) = \bar{\mathbf{D}}_1(\Omega_i^1)$ and $\Phi(\mathbf{S})\bar{\mathbf{A}}_{\Omega_i^2} = \bar{\mathbf{D}}_2(\Omega_i^2)$ are the sub-dictionaries of $\bar{\mathbf{D}}_1$ and $\bar{\mathbf{D}}_2$ that contain the very dictionary atoms used by $\mathbf{y}_i^{\mathrm{r}}$ in sparse representation, and $\Phi(\mathbf{S})$ and $\bar{\mathbf{A}}$ are defined in the previous section. Since $\bar{\mathbf{D}}_1(\Omega_i^1)$ and $\bar{\mathbf{D}}_2(\Omega_i^2)$ can well represent $\mathbf{y}_i^{\mathrm{r}}$, they are also used to represent $\mathbf{y}_i^{\mathrm{d}}$, *i.e.*

$$\phi(\mathbf{y}_i^{\mathrm{d}}) = \Phi(\mathbf{W}_{\Omega_i^1})\mathbf{c}_i^{\mathrm{d}_1} + \mathbf{n}_i^{\mathrm{d}_1}, \tag{23}$$

$$\phi(\mathbf{y}_i^{\mathrm{d}}) = \Phi(\mathbf{S})\bar{\mathbf{A}}_{\Omega_i^2}\mathbf{c}_i^{\mathrm{d}_2} + \mathbf{n}_i^{\mathrm{d}_2}, \tag{24}$$

where $\mathbf{c}_i^{\mathrm{d}_1}, \mathbf{c}_i^{\mathrm{d}_2}$ are the representation coefficients, and $\mathbf{n}_i^{\mathrm{d}_1}, \mathbf{n}_i^{\mathrm{d}_2}$ are the representation errors of $\mathbf{y}_i^{\mathrm{d}}$.

The visual distortion causes the displacements of image patches in the space defined by the sparse coding, which are reflected in the variations of the representation coefficients as well as the representation errors between the reference image patches and distorted image patches. In [13], the coefficients from sparse coding are employed as features, while in [14] the features are shortened by using the norm of sparse coefficients. In our scheme, considering the compactness of features, we use the ℓ_2 norm of sparse coefficients of each

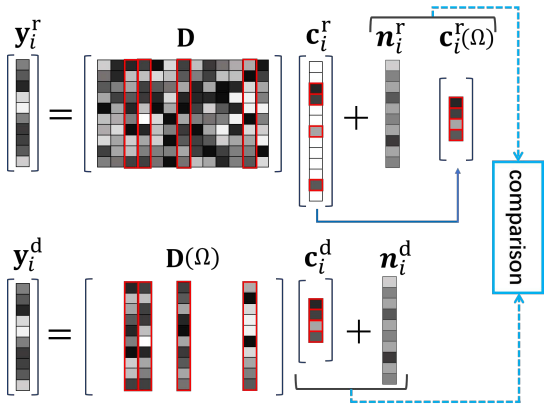

Fig. 2: Outline of kernel sparse coding in proposed KSCM.

patch to construct the IQA features. Furthermore, we also utilize the representation errors for IQA. In details, the visual quality of $\mathbf{y}_i^{\mathrm{d}}$ is measured by comparing the reference features $\{\|\mathbf{c}_i^{\mathrm{r}_1}(\Omega_i^1)\|, \|\mathbf{c}_i^{\mathrm{r}_2}(\Omega_i^2)\|, \|\mathbf{n}_i^{\mathrm{r}_1}\|, \|\mathbf{n}_i^{\mathrm{r}_2}\|\}_i$ with the distortion features $\{\|\mathbf{c}_i^{\mathrm{d}_1}\|, \|\mathbf{c}_i^{\mathrm{d}_2}\|, \|\mathbf{n}_i^{\mathrm{d}_1}\|, \|\mathbf{n}_i^{\mathrm{d}_2}\|\}_i$.

To compute $\mathbf{c}_i^{\mathrm{d}_1}$ and $\mathbf{c}_i^{\mathrm{d}_2}$, we solve the following minimization problems:

- Kernel sparse representation using $\bar{\mathbf{D}}_1(\Omega_i^1)$:

$$\mathbf{c}_i^{\mathrm{d}_1} \in \arg\min_{\mathbf{c}} \|\phi(\mathbf{y}_i^{\mathrm{d}}) - \Phi(\mathbf{W}_{\Omega_i^1})\mathbf{c}\|_{\mathrm{F}}^2, \tag{25}$$

- Kernel sparse representation using $\bar{\mathbf{D}}_2(\Omega_i^2)$:

$$\mathbf{c}_i^{\mathrm{d}_2} \in \arg\min_{\mathbf{c}} \|\phi(\mathbf{y}_i^{\mathrm{d}}) - \Phi(\mathbf{S})\bar{\mathbf{A}}_{\Omega_i^2}\mathbf{c}\|_{\mathrm{F}}^2, \tag{26}$$

which are the kernelized least squares problems whose solutions are given by the solutions of the linear systems

$$\mathbf{K}(\mathbf{W}_{\Omega_i^1}, \mathbf{W}_{\Omega_i^1})\mathbf{c}_i^{\mathrm{d}_1} = \mathbf{K}(\mathbf{W}_{\Omega_i^1}, \mathbf{y}_i^{\mathrm{d}}), \tag{27}$$

$$(\bar{\mathbf{A}}_{\Omega_i^2}^{\top}\mathbf{K}(\mathbf{S}, \mathbf{S})\bar{\mathbf{A}}_{\Omega_i^2})\mathbf{c}_i^{\mathrm{d}_2} = \bar{\mathbf{A}}_{\Omega_i^2}^{\top}\mathbf{K}(\mathbf{S}, \mathbf{y}_i^{\mathrm{d}}). \tag{28}$$

Though $\{\mathbf{n}_i^{\mathrm{r}_1}, \mathbf{n}_i^{\mathrm{r}_2}, \mathbf{n}_i^{\mathrm{d}_1}, \mathbf{n}_i^{\mathrm{d}_2}\}$ cannot be obtained due to the existence of Φ, we can still calculate $\{\|\mathbf{n}_i^{\mathrm{r}_1}\|, \|\mathbf{n}_i^{\mathrm{r}_2}\|, \|\mathbf{n}_i^{\mathrm{d}_1}\|, \|\mathbf{n}_i^{\mathrm{d}_2}\|\}$ by the kernel trick. Take $\|\mathbf{n}_i^{\mathrm{r}_1}\|_2^2$ for example, which can be calculated by

$$\begin{aligned}
\|\ \mathbf{n}_i^{\mathrm{r}_1}\|_2^2 &= \|\phi(\mathbf{y}_i^{\mathrm{r}}) - \Phi(\mathbf{W}_{\Omega_i^1})\mathbf{c}_i^{\mathrm{r}_1}(\Omega_i^1)\|_2^2 \\
&= \big(\phi(\mathbf{y}_i^{\mathrm{r}}) - \Phi(\mathbf{W}_{\Omega_i^1})\mathbf{c}_i^{\mathrm{r}_1}(\Omega_i^1)\big)^{\top}\big(\phi(\mathbf{y}_i^{\mathrm{r}}) - \Phi(\mathbf{W}_{\Omega_i^1})\mathbf{c}_i^{\mathrm{r}_1}(\Omega_i^1)\big) \\
&= \mathbf{K}(\mathbf{y}_i^{\mathrm{r}}, \mathbf{y}_i^{\mathrm{r}}) - \mathbf{K}(\mathbf{y}_i^{\mathrm{r}}, \mathbf{W}_{\Omega_i^1})\mathbf{c}_i^{\mathrm{r}_1}(\Omega_i^1) - (\mathbf{K}(\mathbf{y}_i^{\mathrm{r}}, \mathbf{W}_{\Omega_i^1})\mathbf{c}_i^{\mathrm{r}_1}(\Omega_i^1))^{\top} \\
&\quad + \mathbf{c}_i^{\mathrm{r}_1}(\Omega_i^1)^{\top}\mathbf{K}(\mathbf{W}_{\Omega_i^1}, \mathbf{W}_{\Omega_i^1})\mathbf{c}_i^{\mathrm{r}_1}(\Omega_i^1).
\end{aligned} \tag{29}$$

The calculation of $\|\mathbf{n}_i^{\mathrm{r}_2}\|$,$\|\mathbf{n}_i^{\mathrm{d}_1}\|$,$\|\mathbf{n}_i^{\mathrm{d}_2}\|$ can be done by analogy.

D. Calculation of Visual Quality Score

With the previous steps, we have extracted the features from the reference image $\mathbf{I}^{\mathrm{r}}$ and the distorted image $\mathbf{I}^{\mathrm{d}}$, which are denoted by $\{\mathbf{e}_{\mathrm{c}_1}^{\mathrm{r}}, \mathbf{e}_{\mathrm{n}_1}^{\mathrm{r}}, \mathbf{e}_{\mathrm{c}_2}^{\mathrm{r}}, \mathbf{e}_{\mathrm{n}_2}^{\mathrm{r}}\}$ and $\{\mathbf{e}_{\mathrm{c}_1}^{\mathrm{d}}, \mathbf{e}_{\mathrm{n}_1}^{\mathrm{d}}, \mathbf{e}_{\mathrm{c}_2}^{\mathrm{d}}, \mathbf{e}_{\mathrm{n}_2}^{\mathrm{d}}\}$ respectively, where $\mathbf{e}_{\mathrm{c}_1}^{\mathrm{r}} = (\|\mathbf{c}_1^{\mathrm{r}}(\Omega_1^1)\|, \cdots, \|\mathbf{c}_L^{\mathrm{r}}(\Omega_L^1)\|)$ and $\mathbf{e}_{\mathrm{n}_1}^{\mathrm{r}}, \mathbf{e}_{\mathrm{c}_2}^{\mathrm{r}}, \mathbf{e}_{\mathrm{n}_2}^{\mathrm{r}}, \mathbf{e}_{\mathrm{c}_1}^{\mathrm{d}}, \mathbf{e}_{\mathrm{n}_1}^{\mathrm{d}}, \mathbf{e}_{\mathrm{c}_2}^{\mathrm{d}}, \mathbf{e}_{\mathrm{n}_2}^{\mathrm{d}}$ are constructed in analogy with $\mathbf{e}_{\mathrm{c}_1}^{\mathrm{r}}$.

When viewing an image patch as a point on an implicit subspace defined by the kernel, the extracted features indeed encode the energy of the point as well as its displacement in or around the subspace. As a result, the extracted features from the reference image and distorted image can depict the discriminative information about the distortions with different types and different strengths. For instance, noises increase the variation of image patches and thus probably increase the coding residuals; Blur and JPEG compression that reduce the variations of image patches may decrease the coding residuals and the energy of coding vectors. We illustrate such properties in Fig. 3. It can be seen that our extracted features based on the ℓ_2 norm (*i.e.* energy) of the coding vectors and representation error vectors can well distinguish the noise corruption and image blur with different distortion strengths.

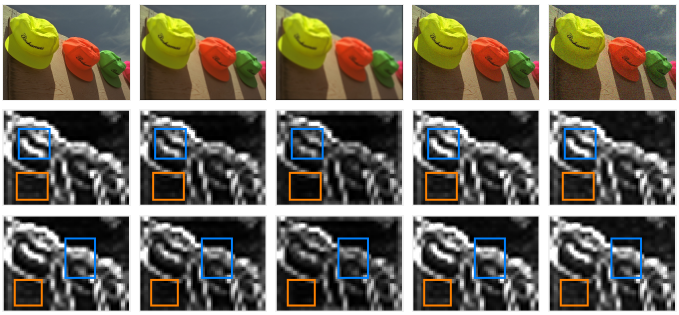

Fig. 3: Illustration of effectiveness of features generated by proposed method. Top row (from left to right): a non-distorted image, two blurry versions generated with a smaller/larger Gaussian blur kernel respectively, and two noisy versions generated with lighter/heavier additive Gaussian white noise respectively. Middle row: corresponding coding feature maps $\{\|\mathbf{c}_i\|\}_i$. Bottom row: corresponding residual feature maps $\{\|\mathbf{n}_i\|\}_i$. In all maps, brighter pixels denote larger values. It can be seen that blurring leads to magnitude decay of coding vectors. Heavier blur leads to lower values (*e.g.* blue and orange rectangles) in $\{\|\mathbf{c}_i\|\}_i$. In contrast, heavier noise leads to larger values in $\{\|\mathbf{n}_i\|\}_i$. Difference can also be found in the feature maps between blurry/noisy versions.

Two metrics are used to pool the features as scores:

- Cross similarity [14] defined by

$$M_{\mathrm{crs}}(\mathbf{x}, \mathbf{y}) = \frac{1}{L}\sum_{i=1}^{L}\frac{2\mathbf{x}(i)\mathbf{y}(i) + c}{(\mathbf{x}(i))^2 + (\mathbf{y}(i))^2 + c}, \tag{30}$$

  where $c > 0$ is a stabilizer set to a small constant. This metric is often used in IQA for estimating the structural similarity; see also [8], [12].
- Pearson correlation coefficient denoted by M_{pcc}, which is a classic metric for measuring the statistical linear correlation between two variables.

Accordingly, we define two scores as follows:

$$\begin{aligned}
S_{\mathrm{crs}}(\mathbf{I}^{\mathrm{d}}; \mathbf{I}^{\mathrm{r}}) &= \beta_1 M_{\mathrm{crs}}(\mathbf{e}_{\mathrm{c}_1}^{\mathrm{r}}, \mathbf{e}_{\mathrm{c}_1}^{\mathrm{d}}) + \beta_2 M_{\mathrm{crs}}(\mathbf{e}_{\mathrm{c}_2}^{\mathrm{r}}, \mathbf{e}_{\mathrm{c}_2}^{\mathrm{d}}) \\
&\quad + \beta_3 M_{\mathrm{crs}}(\mathbf{e}_{\mathrm{n}_1}^{\mathrm{r}}, \mathbf{e}_{\mathrm{n}_1}^{\mathrm{d}}) + \beta_4 M_{\mathrm{crs}}(\mathbf{e}_{\mathrm{n}_2}^{\mathrm{r}}, \mathbf{e}_{\mathrm{n}_2}^{\mathrm{d}}),
\end{aligned} \tag{31}$$

$$\begin{aligned}
S_{\mathrm{pcc}}(\mathbf{I}^{\mathrm{d}}; \mathbf{I}^{\mathrm{r}}) &= \gamma_1 M_{\mathrm{pcc}}(\mathbf{e}_{\mathrm{c}_1}^{\mathrm{r}}, \mathbf{e}_{\mathrm{c}_1}^{\mathrm{d}}) + \gamma_2 M_{\mathrm{pcc}}(\mathbf{e}_{\mathrm{c}_2}^{\mathrm{r}}, \mathbf{e}_{\mathrm{c}_2}^{\mathrm{d}}) \\
&\quad + \gamma_3 M_{\mathrm{pcc}}(\mathbf{e}_{\mathrm{n}_1}^{\mathrm{r}}, \mathbf{e}_{\mathrm{n}_1}^{\mathrm{d}}) + \gamma_4 M_{\mathrm{pcc}}(\mathbf{e}_{\mathrm{n}_2}^{\mathrm{r}}, \mathbf{e}_{\mathrm{n}_2}^{\mathrm{d}}),
\end{aligned} \tag{32}$$

where $\beta_i, \gamma_i > 0$ for $i = 1, \cdots, 4$. Then the final visual quality score is computed by

$$S(\mathbf{I}^{\mathrm{d}}; \mathbf{I}^{\mathrm{r}}) = \lambda_1 S_{\mathrm{crs}}(\mathbf{I}^{\mathrm{d}}; \mathbf{I}^{\mathrm{r}}) + \lambda_2 S_{\mathrm{pcc}}(\mathbf{I}^{\mathrm{d}}; \mathbf{I}^{\mathrm{r}}), \quad (33)$$

where $\lambda_1, \lambda_2 > 0$.

IV. Experiments

A. Experimental Settings and Implementation Details

Eight benchmark datasets are used for experimental evaluation, which include IVC [51], CSIQ [52], TID2008 [53], TID2013 [54], LIVE [55], LIVEMD [56], MDID2013 [57] and CCID2014 [5]. Each of these datasets contains a number of color images with various types of distortions. The characteristics of these datasets are summarized in Table I.

TABLE I: Characteristics of eight benchmark datasets.

Dataset	# Reference Images	# Distorted Images	# Distortion Types
IVC	14	185	4
CSIQ	30	866	6
LIVE	29	779	5
TID2008	25	1700	17
LIVEMD	15	450	2
TID2013	25	3000	24
MDID2013	20	1600	5
CCID2014	15	655	2

To measure the performance of the proposed KSCM from different aspects, five widely-used criteria are employed, including Spearman rank order correlation coefficient (SROCC), Kendall rank order correlation coefficient (KROCC), Pearson linear correlation coefficient (PLCC), Root mean square error (RMSE), and Mean absolute error (MAE). The SROCC and KROCC measure the prediction monotonicity, PLCC measures the linear correlation, while RMSE and MAE measure the prediction accuracy. An effective IQA metric is expected to yield high values of PLCC, SROCC and KROCC, while low values of RMSE and MAE. All the criteria are calculated after mapping the objective score x to the subjective one $f(x)$ by the nonlinear regression

$$f(x) = \omega_1 \left(\frac{1}{2} - \frac{1}{1 + exp(\omega_2(x - \omega_3))}\right) + \omega_4 x + \omega_5, \quad (34)$$

where the parameters ω_i for $i = 1, 2, .., 5$ are determined by least squares fitting. Such a mapping scheme is widely used in existing literature for bridging the gap between the objective and subjective domains; see *e.g.* [26], [55].

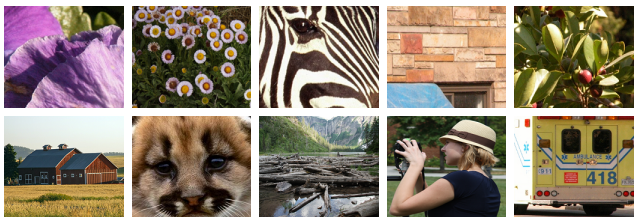

Fig. 4: Natural images for kernel dictionary learning.

The implementation details of KSCM are as follows. The dataset with 43 natural images used by many existing dictionary learning methods (*e.g.* [58], [59]) is used for our kernel dictionary learning; see Fig. 4 for some samples. These 43 images are content-independent to the test images of IQA in the experiments. As the number of image patches is very large, we only sample 10000 patches from the images for learning. The sampling is different from that in image denoising (*e.g.* KSVD [49]) which only considers high-variance patches. We also include low-variance patches as they are useful for IQA. For instance, image compression such as JPEG often produces blurring effects on low-variance patches. Since the high-variance patches in real images are much more than the low-variance ones, we discard parts of low-variance patches during sampling. All the sampled patches are transformed to the YUV color space, and the dictionary learning as well as sparse coding are done on the Y channel. Before processing, we remove the mean of each patch and use it as the coefficient of the uniform atom $\frac{1}{\sqrt{B}}[1, \cdots, 1]^T$. the patch size $\sqrt{B} \times \sqrt{B}$ is set to 2×2 on $\bar{\mathbf{D}}_1$ and 8×8 on $\bar{\mathbf{D}}_2$. The number of atoms in kernel dictionary is set to 256, and the polynomial kernel is used with parameters set by empirical experience.

For the kernel sparse coding in Section III-C, the step size S is set to 1 for $\bar{\mathbf{D}}_1$ and 8 for $\bar{\mathbf{D}}_2$. Accordingly, the sparsity degrees T_1 and T_2 are set to 2 and 5 respectively. In the calculation of the final score, we set $(\lambda_1, \lambda_2) = (1, 1)$ and $\beta_1, \beta_2, \beta_3, \beta_4, \gamma_1, \gamma_2, \gamma_3, \gamma_4$ equal to $1/8$ throughout all the datasets. In such a way, the predicted score is in the range of $[0, 1]$, and higher score represents better quality. For acceleration, all the input images are resized with ratio 0.8.

B. Kernel Sparse Coding versus Linear Sparse Coding

To demonstrate the advantage of nonlinear sparse coding over the linear one for IQA, we compare the proposed KSCM with the sparse-coding-based IQA metric proposed in [14], which is constructed from linear sparse representation and denoted by *LSCM (Linear Sparse Coding based Metric)*. The LSCM method [14] is a good baseline as it employs a very similar framework to our KSCM for computing the quality scores from sparse codes. The main difference between these two metrics is that KSCM employs the nonlinear sparse coding model in (2) while LSCM uses the linear model in (1).

In [14], the experimental results are reported by combining LSCM with other types of metrics, and there are no available results or codes of using LSCM individually. For fair comparison as well as for focusing on the performance of using sparse coding features, we implemented the pure version of LSCM (denoted by LSCM*) with our best effort on parameter tuning for optimal performance, following the reproducible implementation details given in [14] without combining other types of features. Furthermore, since LSCM* only uses one dictionary, we remove the kernel dictionary $\bar{\mathbf{D}}_1$ and its related module in our KSCM for fairness, resulting in a simplified KSCM, denoted by KSCM*. In the comparison, the sparsity degrees in both the methods are set to 3. The results are listed in Table II, from which noticeable improvement of KSCM* over LSCM* can be observed. The improvement comes from the effectiveness of KSCM* in characterizing complex image structures with nonlinearities, and it has demonstrated that nonlinear sparse coding is more effective in capturing the non-Euclidean geometric structures than the linear one for IQA.

TABLE II: Performance comparison of KSCM* and LSCM*. Better ones are boldfaced.

Method	Criterion	IVC	CSIQ	LIVE	TID2008	TID2013
LSCM*	PLCC	0.8181	0.8283	0.9042	0.7435	0.7435
	SROCC	0.8160	0.8460	0.9092	0.7099	0.6649
	KROCC	0.6145	0.6471	0.7295	0.5232	0.4907
	RMSE	0.7006	0.1471	11.668	0.8974	0.8291
	MAE	0.5456	0.1159	9.2455	0.6929	0.6546
KSCM*	PLCC	**0.9154**	**0.9411**	**0.9205**	**0.8538**	**0.8349**
	SROCC	**0.9044**	**0.9370**	**0.9273**	**0.8601**	**0.7758**
	KROCC	**0.7218**	**0.7734**	**0.7613**	**0.6740**	**0.5974**
	RMSE	**0.4904**	**0.0888**	**10.676**	**0.6987**	**0.6822**
	MAE	**0.3868**	**0.0705**	**8.3418**	**0.5231**	**0.5246**

C. Comparison with General IQA Metrics

The proposed KSCM is compared with several FR-IQA metrics for comparison, including PSNR, SSIM [8], VIF [11], MAD [52], IW-SSIM [9], FSIM [10], GMSD [37], PGSD [38], SSRM [33] and EFS [39]. Among them, IW-SSIM and FSIM are regarded as the top-performers among the sixteen FR-IQA methods in [60], while PGSD, SSRM and EFS are recently-proposed methods. The results of these methods in comparison are cited from the existing literature whenever available. If not, we run their published codes to produce the results. If both results and codes are unavailable, we leave the results blank. See Table III for the results and comparison. To evaluate the overall performance on all the datasets, we define two additional performance metrics for each method: (i) the average of the scores over all the datasets; and (ii) the average weighted by the normalized number of distorted images in each dataset. As shown in Table III, our KSCM consistently performs the best across all the datasets under all criteria on IVC. In addition, it outperforms several FR-IQA metrics on all datasets with all criteria. The top performer varies on different datasets: VIF on MDID2013 and CCID2014, EFS on TID2008 and TID2013, and PGSD on CSIQ. As a whole, KSCM performs on a par with FSIM and a bit worse than GMSD and EFS.

The performance of our KSCM as well as other compared metrics has a noticeable drop on MDID2013 and CCID2014. One main reason for MDID2013 is, compared to other datasets, a single image in MDID2013 is likely to contain more distortion types. As a result, the distortions occurring on the image patches in MDID2013 can differ from each other much, which is very challenging. In comparison to traditional dictionary learning, ours can be better at analyzing multiple types of distortions as the analysis is done in a higher-dimensional feature space. However, the single kernel we use may be still insufficient for fully characterizing all combinations of distortion types, limiting the performance. This may be remedied by introducing multiple kernel learning, and we leave it to our future work. Regarding CCID2014, it is a large dataset where the image distortions are mainly caused by contrast changes. Such contrast distortions are not very related to the structural information changes of images and thus the advantage of sparse coding in analyzing local image structures does not benefit much in this case. Thus, it is not surprising to see the performance drop of our KSCM as well as other IQA metrics that mainly focus on structural information.

A significance test is also conducted to identify the difference in performance between different metrics. The approach in [61], [62] is followed whereby an F-test at 95% significance level is performed on the residual between the subjective score and the one predicted by the tested IQA metrics. The null hypothesis states that variances of the error residuals from the two different IQA metrics are equal, and thus the test indeed tells whether one IQA metric is statistically superior over another. See Fig. 5 for the F-test results of every pair of compared metrics. In addition, the sum of F-test results of each metric over other metrics is calculated, and the corresponding ranking of each metric is also provided in Fig. 5. It can be seen that our metric is significantly better than most compared methods on CSIQ and TID2013. On LIVE, our KSCM is slightly worse than FSIM, GMSD and MAD. On LIVEMD, there is no method significantly better than ours.

See Fig. 6 for the performance visualization, which shows the scatter plots of the predicted quality scores (before nonlinear regression) against subjective scores, regarding six degradation types on CSIQ. The black curve is obtained by fitting the data points with the nonlinear regression of (34). From the visual results, it can be seen that the data points in each scatter plot are very close to the fitting curve and the curve is almost monotonous, implying that our KSCM is highly consistent with HVS. It is also observed that the fitting curves of other all compared methods are more divergent than ours, which demonstrates the superior performance of our method.

D. Comparison on Individual Distortion Types

It is important to study how an IQA metric performs on different types of distortions. We conduct the performance evaluation on each type of distortion individually with the same experimental protocol used on the whole dataset. The results on LIVE, CSIQ and TID2013 are listed in Table IV. We count how many times of a method being rank-1 or top-3, and show them in the bottom row of Table IV. It can be seen that our KSCM consistently performs well on different types of distortions, with superior performance to other compared methods on some distortion types. Totally, our KSCM is rank-1 for 9 times, followed by EFS (6 times). The worst case of KSCM happens on the distortion of local block damage. One main reason is that the distortion occurring in just a few of image blocks only changes the coefficients and representation errors of several patches. Such changes are averaged out over all the patches of the whole image, and thus the corresponding metric value is insufficient to reflect the degree of the distortion.

E. Performance of Using Individual Dictionary

To demonstrate the necessity of using the analytic dictionary $\bar{\mathbf{D}}_1$ and the data-driven dictionary $\bar{\mathbf{D}}_2$ in the proposed kernel dictionary construction scheme, we test the performance of our KSCM by only using $\bar{\mathbf{D}}_1$ and $\bar{\mathbf{D}}_2$ respectively. Furthermore, we verify the benefits of combining the analytic sub-dictionary $\Phi(\mathbf{G})$ with the adaptive sub-dictionary $\Phi(\mathbf{Y})\mathbf{A}$ in the construction of $\bar{\mathbf{D}}_2$ by testing the performance of KSCM

TABLE III: Performance comparison on eight benchmark datasets. The top three in each row are boldfaced and the best one in each row is underlined. The notation '-' denotes unavailable entry.

Database	Criteria	OURS	FSIM	IW-SSIM	SSIM	PSNR	GMSD	MAD	VIF	EFS	PGSD	SSRM
IVC	PLCC	**0.9387**	**0.9376**	0.9231	0.9119	0.7196	**0.9235**	0.9210	0.9028	0.9048	–	0.9132
	SROCC	**0.9301**	**0.9262**	0.9125	0.9018	0.6884	**0.9145**	0.9146	0.8964	0.8920	–	0.9050
	RMSE	**0.4199**	**0.4236**	0.4686	0.4999	0.8460	**0.4674**	0.4746	0.5239	0.5188	–	0.4966
CSIQ	PLCC	**0.9531**	0.9120	0.9144	0.8613	0.8000	**0.9541**	0.9502	0.9258	0.9287	**0.9564**	0.9287
	SROCC	**0.9519**	0.9242	0.9213	0.8756	0.8058	**0.9570**	0.9466	0.9194	0.9371	**0.9572**	0.9367
	RMSE	**0.0794**	0.1077	0.1063	0.1334	0.1575	**0.0786**	0.0818	0.0992	0.0973	**0.0767**	0.0974
LIVE	PLCC	0.9524	**0.9597**	0.9522	0.9449	0.8723	**0.9603**	**0.9675**	0.9411	0.9506	0.9564	0.9570
	SROCC	0.9585	**0.9634**	0.9567	0.9479	0.8756	**0.9603**	**0.9669**	0.9636	0.9550	0.9572	0.9604
	RMSE	8.3316	7.6780	8.3473	8.9455	13.3597	**7.6214**	**6.9073**	9.2402	8.4794	**6.6872**	7.9283
TID2008	PLCC	**0.8760**	0.8738	0.8579	0.7732	0.5734	**0.8788**	0.8306	0.8084	**0.8810**	–	0.8379
	SROCC	0.8790	**0.8805**	0.8559	0.7749	0.5531	**0.8907**	0.8340	0.7491	**0.8925**	–	0.8331
	RMSE	**0.6473**	0.6525	0.6895	0.8511	1.0994	**0.6404**	0.7473	0.7898	**0.6349**	–	0.7324
TID2013	PLCC	**0.8812**	0.8589	0.8319	0.7895	0.7017	0.8590	0.8267	0.7720	**0.9067**	**0.8611**	0.8078
	SROCC	**0.8603**	0.8022	0.7779	0.7417	0.7028	0.8044	0.8086	0.6679	**0.8948**	**0.8565**	0.7506
	RMSE	**0.5859**	0.6349	0.6880	0.7608	0.8832	0.6346	0.6975	0.7879	**0.5230**	**0.6303**	0.7308
LIVEMD	PLCC	**0.9051**	0.8926	**0.9090**	0.8908	0.7386	0.8794	0.8938	**0.8976**	0.8863	–	0.8724
	SROCC	**0.8914**	0.8673	**0.8866**	0.8636	0.6781	0.8502	0.8646	**0.8745**	0.8711	–	0.8545
	RMSE	**8.0432**	8.5259	**7.8815**	8.5939	12.7490	9.0029	8.4812	**8.3361**	8.7579	–	9.2434
MDID2013	PLCC	0.8193	**0.8970**	0.8983	0.8457	0.6164	0.8776	0.7552	**0.9367**	0.8707	–	**0.8785**
	SROCC	0.7988	**0.8873**	**0.8911**	0.8328	0.5784	0.8613	0.7249	**0.9306**	0.8572	–	0.8689
	RMSE	1.2664	**0.9738**	0.9682	1.1757	1.7350	1.0565	1.4442	**0.7717**	1.1435	–	**1.0528**
CCID2014	PLCC	0.8048	0.8202	0.8342	**0.8308**	0.5705	**0.8521**	0.7928	**0.8588**	0.8122	–	0.8166
	SROCC	0.7514	0.7655	0.7811	**0.8174**	0.6399	**0.8077**	0.7430	**0.8349**	0.7672	–	0.7676
	RMSE	0.3881	0.3741	0.3606	**0.3640**	0.5370	**0.3422**	0.3985	**0.3350**	0.3814	–	0.3775
Average	PLCC	0.8913	**0.8940**	0.8901	0.8560	0.6991	**0.8981**	0.8672	0.8804	**0.8973**	–	0.8765
	SROCC	**0.8777**	0.8771	0.8729	0.8445	0.6903	**0.8808**	0.8504	0.8546	**0.8898**	–	0.8596
	RMSE	2.4702	2.4213	**2.4388**	2.6655	3.9209	2.4805	**2.4041**	2.6105	**1.7725**	–	2.5824
Weighted Average	PLCC	0.8793	**0.8824**	0.8720	0.8267	0.6802	**0.8853**	0.8415	0.8510	**0.8939**	–	0.8556
	SROCC	**0.8654**	0.8597	0.8489	0.8084	0.6724	**0.8625**	0.8256	0.8037	**0.8868**	–	0.8311
	RMSE	1.6814	**1.6164**	**1.6657**	1.8446	2.6311	**1.6426**	1.6671	1.7804	1.6882	–	1.7331

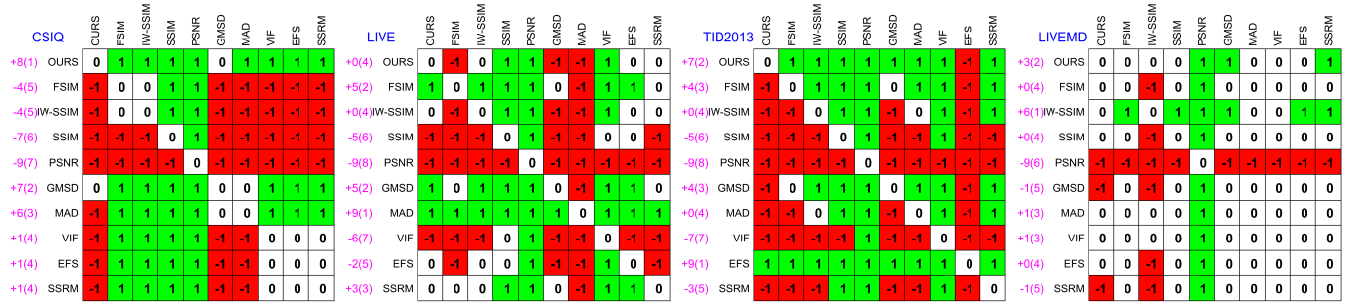

Fig. 5: F-test results regarding prediction errors on four datasets between each pair of metrics. The values of '1' (green) / '-1' (red) imply that the metric associated with the row is significantly better/worse than the metric associated with the column. The value '0' implies there is no significant difference. The overall scores (ranks) of each method in the row are summarized vertically in the left positions. The higher scores or the smaller numbers in rank imply higher performance.

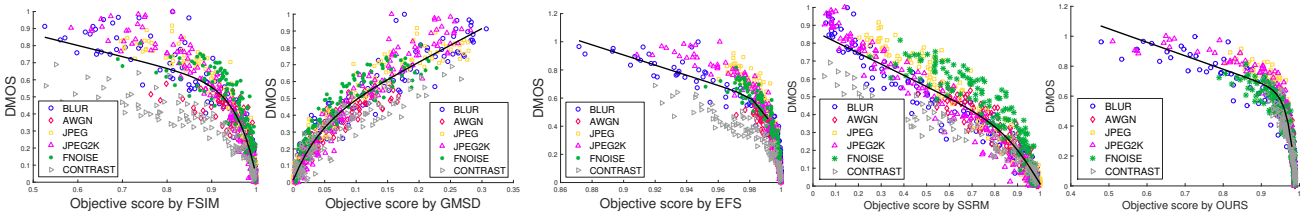

Fig. 6: Scatter plots of predicted scores (before nonlinear regression) against subjective scores (DMOS) by different IQA metrics on the CSIQ dataset. Different distortion types are associated with different shapes and colors.

TABLE IV: SROCC comparison on individual distortion type of three datasets. The top three in each row are boldfaced, and the best one in each row is underlined.

Database	Distortion	OURS	FSIM	IW-SSIM	GMSD	MAD	VIF	EFS	PGSD	SSRM
LIVE	Gaussian blur	**0.9883**	0.9652	0.9667	**0.9711**	0.9676	0.9683	0.9679	**0.9752**	0.9634
	Additive Gaussian noise	0.9614	0.9706	0.9719	0.9782	0.9764	**0.9846**	0.9762	**0.9894**	**0.9823**
	JPEG compression	0.9695	0.9714	0.9649	0.9737	**0.9844**	**0.9858**	**0.9838**	0.9619	0.9807
	JPEG2000 compression	0.9802	**0.9834**	**0.9808**	0.9567	0.9465	0.9728	0.9663	**0.9820**	0.9693
	JPEG2000 trans. error	0.9488	0.9499	0.9442	0.9416	0.9569	**0.9650**	0.9488	**0.9681**	**0.9694**
CSIQ	Gaussian blur	**0.9750**	0.9721	**0.9781**	0.9676	0.9542	0.9575	0.9664	**0.9772**	0.9543
	Additive Gaussian noise	0.9572	0.9258	0.9380	0.9651	0.9614	**0.9703**	**0.9662**	**0.9673**	0.9656
	JPEG compression	**0.9742**	0.9656	0.9660	0.9717	**0.9752**	0.9671	**0.9771**	0.9695	0.9737
	JPEG2000 compression	**0.9808**	**0.9683**	0.9682	0.9502	0.9568	0.9509	0.9591	**0.9790**	0.9445
	Additive pink noise	0.9455	0.9230	0.9057	0.9712	0.9681	**0.9744**	**0.9763**	0.9513	**0.9779**
	Contrast change	**0.9542**	0.9422	0.9540	0.9037	0.9210	0.9345	**0.9557**	0.9488	**0.9530**
TID2013	Additive Gaussian noise	0.9148	0.8984	0.8449	**0.9464**	0.8856	0.8999	0.9456	**0.9521**	0.8558
	Noise in color comp.	**0.8868**	0.8177	0.7515	0.8684	0.8014	0.8433	**0.8830**	0.8688	0.7711
	Spatially corr. noise	0.9084	0.8751	0.8167	**0.9350**	0.8913	0.8888	**0.9354**	**0.9463**	0.8388
	Masked noise	**0.9487**	0.7937	0.8020	0.7170	0.7376	**0.8447**	0.7961	0.7561	**0.8177**
	High frequency noise	0.9041	0.8986	0.8589	0.9160	0.8950	0.8972	**0.9227**	**0.9184**	0.8771
	Impulse noise	**0.8154**	0.8076	0.7281	0.7637	0.3261	**0.8537**	**0.8713**	0.8124	0.7862
	Quantization noise	0.8619	0.8713	0.8468	**0.9049**	0.8514	0.8160	0.8666	**0.8958**	0.8488
	Gaussian blur	0.9603	0.9550	**0.9701**	0.9113	0.9319	**0.9650**	0.9641	0.9054	**0.9674**
	Image denoising	**0.9671**	0.9301	0.9152	0.9525	0.9252	0.9063	0.9555	**0.9566**	0.9284
	JPEG compression	**0.9782**	0.9382	0.9198	0.9507	0.9264	0.9192	**0.9672**	0.9468	0.9287
	JPEG2000 compression	**0.9784**	0.9577	0.9506	0.9657	0.9514	0.9516	**0.9755**	0.9650	0.9561
	JPEG trans. errors	**0.9327**	0.8466	0.8388	0.8497	0.8487	0.8447	**0.9239**	0.8694	**0.8767**
	JPEG2000 trans. errors	0.8994	0.8912	0.8656	**0.9136**	0.8788	0.8761	**0.9130**	**0.9092**	0.8763
	Non ecc. patt. noise	**0.8274**	0.7917	0.8011	0.8140	**0.8313**	0.7720	0.8087	**0.8306**	0.7929
	Local block-wise dist.	0.2179	0.5533	0.3722	**0.6625**	0.2366	0.5306	**0.6398**	**0.6164**	0.3180
	Mean shift	0.6094	**0.7524**	**0.7833**	0.7351	0.6450	0.6272	**0.7542**	0.6442	0.6913
	Contrast change	0.2425	0.4675	0.4593	**0.6212**	0.3420	**0.8523**	0.6028	**0.6320**	0.5184
	Color saturation change	0.6983	0.3790	0.4196	0.3801	0.2414	0.3205	**0.7980**	**0.7772**	0.3729
	Multiplicative Gaussian noise	0.8639	0.8468	0.7728	**0.8886**	0.8405	0.8476	**0.9051**	**0.8968**	0.8058
	Comfort noise	**0.9427**	0.9118	0.8762	**0.9298**	0.9140	0.8946	0.9192	**0.9385**	0.8922
	Lossy com. of noisy images	0.9501	0.9470	0.9037	**0.9629**	0.9443	0.9228	**0.9550**	**0.9721**	0.9161
	Color quantization with dither	0.8830	0.8757	0.8401	**0.9102**	0.8745	0.8453	**0.9007**	**0.9144**	0.8534
	Chromatic aberrations	**0.8838**	0.8713	0.8682	0.8530	0.8310	**0.8848**	**0.8954**	0.8589	0.8835
	Sparse samp. and recons.	0.9614	0.9563	0.9474	**0.9683**	0.9581	0.9377	**0.9630**	**0.9676**	0.9540
	#Top-3 / #Top-1	15 / 9	3 / 1	4 / 3	12 / 4	3 / 1	10 / 3	21 / 7	22 / 5	7 / 2

using $\Phi(\mathbf{Y})\mathbf{A}$ separately. The results of using $\bar{\mathbf{D}}_1, \bar{\mathbf{D}}_2, \Phi(\mathbf{Y})\mathbf{A}$ are summarized in Table V, where we also list the results of using all the dictionaries for comparison.

It can be seen that $\bar{\mathbf{D}}_1$ and $\bar{\mathbf{D}}_2$ are both effective for IQA, and their combination has noticeable improvement over the individual ones. Such improvement has demonstrated the necessity of using both the analytic dictionary and the learnable one in KSCM. We can also see that using $\bar{\mathbf{D}}_2 = [\Phi(\mathbf{Y})\mathbf{A}, \Phi(\mathbf{G})]$ is superior to that of only using $\Phi(\mathbf{Y})\mathbf{A}$ on most datasets, which has demonstrated the benefit of introducing the fixed sub-dictionary $\Phi(\mathbf{G})$ in the learnable dictionary $\bar{\mathbf{D}}_2$.

F. Impact of Sparsity Degree

The sparsity degree is one important parameter in kernel sparse coding. Regarding the sparsity degrees T_1 and T_2 in the kernel sparse coding model of KSCM, each of them is about the dimension of the manifold that the data (*i.e.* image patches) lie on. To examine the influence of these two parameters, we set them to 1 to 10 respectively and then report the resulting performance. The results regarding T_2 in terms of SROCC and RMSE are shown in Fig. 7. From the results, our KSCM is insensitive to the two parameters within a reasonable range. This phenomenon is reasonable, as very small T_1 (or T_2) may cause noticeable reconstructive error in kernel sparse coding which decreases the representative power of the model, while a large T_1 (or T_2) may go beyond the true dimension of data manifold which results in over-fitting.

G. Computational Cost

The computational cost of KSCM is evaluated by comparing its running time with other methods. The test is conducted with MATLAB R2018b run on an Intel Core I7-7700K CPU (4.20 GHz) and 32 GB RAM. All the Matlab source codes of compared methods are from their websites. Ten 512×512 images are randomly chosen from the CSIQ dataset for test and the average time over them is recorded. Then the average time over 100 runs is reported. See Table VI for results. Our KSCM runs faster than MAD and VIF, while slower than other methods. Nevertheless, we note that the computational cost of KSCM is still in a reasonable range for real-world applications. One main cause of computational cost of KSCM is its sparse coding procedure, which may be accelerated by parallel computing.

V. Conclusion and Future Work

We investigated the sparse-coding-based approach for IQA. Instead of following existing approaches which employ linear

TABLE V: Performance of KSCM using different dictionaries.

Dictionary	Criterion	IVC	CSIQ	LIVE	TID2008	TID2013
$\bar{\mathbf{D}}_1$	PLCC	0.8750	0.9367	0.9156	0.8265	0.8193
	SROCC	0.8645	0.9334	0.9210	0.8274	0.7603
	KROCC	0.6823	0.7665	0.7499	0.6354	0.5774
	RMSE	0.5898	0.0919	10.9859	0.7553	0.7108
	MAE	0.4328	0.0727	8.6288	0.5749	0.5552
$\Phi(\mathbf{Y})\mathbf{A}$	PLCC	0.8841	0.8680	0.8870	0.7603	0.7698
	SROCC	0.8817	0.8646	0.8952	0.7445	0.7066
	KROCC	0.6908	0.6807	0.7252	0.5556	0.5252
	RMSE	0.5693	0.1304	12.6152	0.8716	0.7913
	MAE	0.4422	0.1041	9.9091	0.6745	0.6206
$\bar{\mathbf{D}}_2$	PLCC	0.8809	0.9417	0.9030	0.8070	0.8121
	SROCC	0.8748	0.9483	0.9152	0.8334	0.7687
	KROCC	0.6788	0.7963	0.7458	0.6384	0.5894
	RMSE	0.5767	0.0883	11.7414	0.7925	0.7234
	MAE	0.4717	0.0705	9.1645	0.5843	0.5435
$\bar{\mathbf{D}}_1$, $\bar{\mathbf{D}}_2$	PLCC	0.9387	0.9531	0.9524	0.8760	0.8812
	SROCC	0.9301	0.9519	0.9585	0.8790	0.8603
	KROCC	0.7690	0.8035	0.8195	0.6984	0.6760
	RMSE	0.4199	0.0794	8.3316	0.6473	0.5859
	MAE	0.3128	0.0614	6.6158	0.4848	0.4419

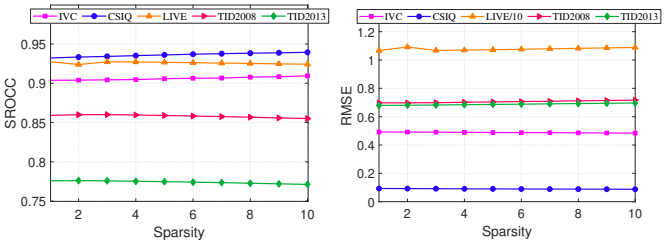

Fig. 7: Impact of sparsity degree.

TABLE VI: Comparisons of running time (seconds).

OURS	FSIM	IW-SSIM	SSIM	PSNR	GMSD	MAD	VIF	SSRM	EFS
0.239	0.173	0.341	0.035	0.002	0.005	0.791	0.589	0.088	0.149

coding models, we proposed to use the kernel sparse coding model, which is nonlinear, to construct the IQA metric. To increase the effectiveness and stability of kernel sparse coding in IQA tasks, we proposed a kernel dictionary construction scheme which combines learnable and analytic dictionaries. In the experimental evaluation, the proposed approach not only showed improvement over the ones built upon linear sparse coding, but also competed against the state-of-the-art ones. Such results have demonstrated the benefits of using nonlinear sparse representation in IQA, suggesting sparse representation is a promising technology for IQA. In future, we would like to investigate the acceleration of our kernel sparse coding for large-scale IQA, as well as the extension of sparse coding with multiple kernel learning for better handling mixed distortions.

REFERENCES

[1] S. Wang, A. Rehman, Z. Wang, S. Ma, and W. Gao, "Ssim-motivated rate-distortion optimization for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 516–529, 2012.

[2] M. K. Samee and J. Götze, "Reduced reference image quality assessment for transmitted images using digital watermarking," in *Proc. IEEE Int. Symp. Image and Signal Process. and Anal.*, 2011, pp. 425–430.

[3] X. Liu, D. Zhai, J. Zhou, S. Wang, D. Zhao, and H. Gao, "Sparsity-based image error concealment via adaptive dual dictionary learning and regularization," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 782–796, 2017.

[4] K. Gu, D. Tao, J.-F. Qiao, and W. Lin, "Learning a no-reference quality assessment model of enhanced images with big data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1301–1313, 2018.

[5] K. Gu, G. Zhai, W. Lin, and M. Liu, "The analysis of image contrast: From quality assessment to automatic enhancement." *IEEE Trans. Cybernetics*, vol. 46, no. 1, pp. 284–297, 2016.

[6] P. Le Callet, C. Viard-Gaudin, and D. Barba, "Continuous quality assessment of mpeg2 video with reduced reference," in *Int. Workshop Video Process. and Quality Metrics for Consumer Electronics, Phoenix*, 2005.

[7] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. IEEE Conf. Signals, Syst. and Comput.*, vol. 2, 2003, pp. 1398–1402.

[8] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[9] Z. Wang and Q. Li, "Information content weighting for perceptual image quality assessment." *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1185–1198, 2010.

[10] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "Fsim: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, 2011.

[11] H. R. Sheikh and A. C. Bovik, "Image information and visual quality." *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, 2006.

[12] H. W. Chang, H. Yang, Y. Gan, and M. H. Wang, "Sparse feature fidelity for perceptual image quality assessment," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 4007–18, 2013.

[13] T. Guha, E. Nezhadarya, and R. K. Ward, "Sparse representation-based image quality assessment," *Signal Process. Image Commun.*, vol. 29, no. 10, pp. 1138–1148, 2014.

[14] L. Li, H. Cai, Y. Zhang, W. Lin, A. C. Kot, and X. Sun, "Sparse representation-based image quality index with adaptive sub-dictionaries," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3775–3786, 2016.

[15] Y. Liu, G. Zhai, K. Gu, X. Liu, D. Zhao, and W. Gao, "Reduced-reference image quality assessment in free-energy principle and sparse representation," *IEEE Trans. Multimedia*, vol. 20, no. 2, pp. 379–391, 2018.

[16] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2011, pp. 457–464.

[17] G. Peyré, "Manifold models for signals and images," *Comput. Vision and Image Understanding*, vol. 113, no. 2, pp. 249–260, 2009.

[18] H. Barlow, "Redundancy reduction revisited," *Network: computation in neural systems*, vol. 12, no. 3, pp. 241–253, 2001.

[19] B. A. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 07 1996.

[20] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by v1?" *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[21] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, and D. Cai, "Graph regularized sparse coding for image representation," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1327–1336, 2011.

[22] M. Harandi and M. Salzmann, "Riemannian coding and dictionary learning: Kernels to the rescue," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2015, pp. 3926–3935.

[23] B.-D. Liu, Y.-X. Wang, Y.-J. Zhang, and B. Shen, "Learning dictionary on manifolds for image classification," *Pattern Recognition*, vol. 46, no. 7, pp. 1879–1890, 2013.

[24] H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Design of non-linear kernel dictionaries for object recognition," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5123–5135, 2013.

[25] L. Ma, S. Li, F. Zhang, and K. N. Ngan, "Reduced-reference image quality assessment using reorganized dct-based image representation," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 824–829, 2011.

[26] Y. Xu, D. Liu, Y. Quan, and P. Le Callet, "Fractal analysis for reduced reference image quality assessment," *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2098–2109, 2015.

[27] Y. Quan, C. Bao, and H. Ji, "Equiangular kernel dictionary learning with applications to dynamic texture analysis," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2016, pp. 308–316.

[28] L. He, D. Tao, X. Li, and X. Gao, "Sparse representation for blind image quality assessment," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2012, pp. 1146–1153.
[29] L. Li, W. Xia, Y. Fang, K. Gu, J. Wu, W. Lin, and J. Qian, "Color image quality assessment based on sparse representation and reconstruction residual," *J. Visual Commun. and Image Representation*, vol. 38, pp. 550–560, 2016.
[30] L. Li, D. Wu, J. Wu, H. Li, W. Lin, and A. C. Kot, "Image sharpness assessment by sparse representation," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 1085–1097, 2016.
[31] Y.-K. Lai and C.-C. J. Kuo, "A haar wavelet approach to compressed image quality measurement," *J. Visual Commun. and Image Representation*, vol. 11, pp. 17–40, 03 2000.
[32] Z. Di and E. Jernigan, "An information theoretic criterion for image quality assessment based on natural scene statistics," in *Proc. IEEE Int. Conf. Image Process.*, 2007, pp. 2953–2956.
[33] A. Ahar, A. Barri, and P. Schelkens, "From sparse coding significance to perceptual quality: A new approach for image quality assessment," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 879–893, 2018.
[34] G. Fei, W. Yi, P. Li, T. Min, J. Yu, and Y. Zhu, "Deepsim: Deep similarity for image quality assessment," *Neurocomputing*, vol. 257, pp. 104–114, 2017.
[35] S. Bosse, D. Maniry, K. R. Muller, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 206–219, 2018.
[36] A. Liu, W. Lin, and M. Narwaria, "Image quality assessment based on gradient similarity," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1500–1512, 2012.
[37] W. Xue, Z. Lei, X. Mou, and A. C. Bovik, "Gradient magnitude similarity deviation: A highly efficient perceptual image quality index," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 684–695, 2014.
[38] M. Jin, T. Wang, Z. Ji, and X. Shen, "Perceptual gradient similarity deviation for full reference image quality assessment," *Comput., Mater. Continua*, vol. 56, no. 3, pp. 501–515, 2018.
[39] Z. Shi, J. Zhang, Q. Cao, K. Pang, and T. Luo, "Full-reference image quality assessment based on image segmentation with edge feature," *Signal Process.*, vol. 145, pp. 99–105, 2018.
[40] Y. Liu, G. Zhai, X. Liu, and D. Zhao, "Perceptual image quality assessment combining free-energy principle and sparse representation," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2016, pp. 1586–1589.
[41] J. Shi, L. Xu, and J. Jia, "Just noticeable defocus blur detection and estimation," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2015, pp. 657–665.
[42] Y. Liu, K. Gu, Y. Zhang, X. Li, G. Zhai, D. Zhao, and W. Gao, "Unsupervised blind image quality evaluation via statistical measurements of structure, naturalness and perception," *IEEE Trans. Circuits Syst. Video Technol.*, 2019.
[43] Z. Wang, Y. Wang, H. Liu, and H. Zhang, "Structured kernel dictionary learning with correlation constraint for object recognition," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4578–4590, 2017.
[44] C. O'Brien and M. D. Plumbley, "Sparse kernel dictionary learning," *Proc. IMA Intl. Conf. Math. Signal Process.*, 2016.
[45] A. Golts and M. Elad, "Linearized kernel dictionary learning," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 726–739, 2016.
[46] Q. Li and Z. Wang, "Reduced-reference image quality assessment using divisive normalization-based image representation," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 2, pp. 202–211, 2009.
[47] R. Soundararajan and A. C. Bovik, "Rred indices: Reduced reference entropic differencing for image quality assessment," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 517–526, 2012.
[48] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic press, 1998.
[49] M. Aharon, M. Elad, and A. Bruckstein, "k-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.
[50] D. Gong, F. Sha, and G. Medioni, "Locally linear denoising on image manifolds," *J. Machine Learning Research*, vol. 2010, no. 9, pp. 265–272, 2010.
[51] P. Le Callet and F. Autrusseau, "Subjective quality assessment irccyn/ivc database," 2005, http://www.irccyn.ec-nantes.fr/ivcdb/.
[52] E. C. Larson and D. M. Chandler, "Most apparent distortion: full-reference image quality assessment and the role of strategy," *J. Electronic Imaging*, vol. 19, no. 1, p. 011006, 2010.
[53] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti, "Tid2008 - a database for evaluation of full-reference visual quality assessment metrics," *Adv. Modern Radioelectron*, vol. 10, pp. 30–45, 2004.
[54] N. Ponomarenko, O. Ieremeiev, V. Lukin, K. Egiazarian, L. Jin, J. Astola, B. Vozel, K. Chehdi, M. Carli, and F. Battisti, "Color image database tid2013: Peculiarities and preliminary results," in *Proc. European Workshop Visual Inform. Process.*, 2013, pp. 106–111.
[55] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms." *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3440–3451, 2006.
[56] D. Jayaraman, A. Mittal, A. K. Moorthy, and A. C. Bovik, "Objective quality assessment of multiply distorted images," in *Proc. IEEE Conf. on Signals, Syst. and Comput.*, 2012, pp. 1693–1697.
[57] W. Sun, F. Zhou, and Q. Liao, "Mdid: a multiply distorted image database for image quality assessment," *Pattern Recognition*, vol. 61, pp. 153–168, 2017.
[58] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, 2010.
[59] L. He, H. Qi, and R. Zaretzki, "Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2013, pp. 345–352.
[60] K. Ma, Z. Duanmu, Z. Wang, Q. Wu, W. Liu, H. Yong, H. Li, and L. Zhang, "Group maximum differentiation competition: Model comparison with few samples," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 851–864, 2020.
[61] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1427–1441, 2010.
[62] R. Zhu, F. Zhou, W. Yang, and J. Xue, "On hypothesis testing for comparing image quality assessment metrics," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 133–136, 2018.