Texture Representations Using Subspace Embeddings

Xiaodong Yang and YingLi Tian
Department of Electrical Engineering
The City College of New York, CUNY
{xyang02, ytian}@ccny.cuny.edu

Abstract — In this paper, we propose a texture representation framework to map local texture patches into a low-dimensional texture subspace. In natural texture images, textons are entangled with multiple factors, such as rotation, scaling, viewpoint variation, illumination change, and non-rigid surface deformation. Mapping local texture patches into a low-dimensional subspace can alleviate or eliminate these undesired variation factors resulting from both geometric and photometric transformations. We observe that texture representations based on subspace embeddings have strong resistance to image deformations, meanwhile, are more distinctive and more compact than traditional representations. We investigate both linear and non-linear embedding methods including Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Locality Preserving Projections (LPP) to compute the essential texture subspace. The experiments in the context of texture classification on benchmark datasets demonstrate that the proposed subspace embedding representations achieve the state-of-the-art results while with much fewer feature dimensions.

Keywords — texture representation, texture classification, subspace embedding

1. Introduction

The automated analysis of texture is widely applied in a number of real-world applications, e.g., image and video retrieval, object recognition and segmentation, and natural scene classification [5, 8, 19, 22, 26]. However, it is a challenging problem to represent texture images due to scaling changes, affine deformations, and lighting variations. A desired texture representation is thus supposed to handle both geometric and photometric variations. There has been extensive research in the literature on designs of robust texture representations. Early work for modeling texture includes filter banks [23] and co-occurrence features [11]. They mainly concentrate on global 2D transformations such as rotation and scaling. Most recent work further captures the effects of 3D transformations such as viewpoint change and non-rigid surface deformation. The representation methods based on fractal analysis [30, 33] have also been proposed to model spatial distribution properties of textons with impressive recognition performance. In addition, most recent state-of-the-art results in texture recognition are obtained by using histograms of local image features as distributions of textons [9, 16, 35]. Therefore, the effective computations of textons are crucial for robust texture representations.

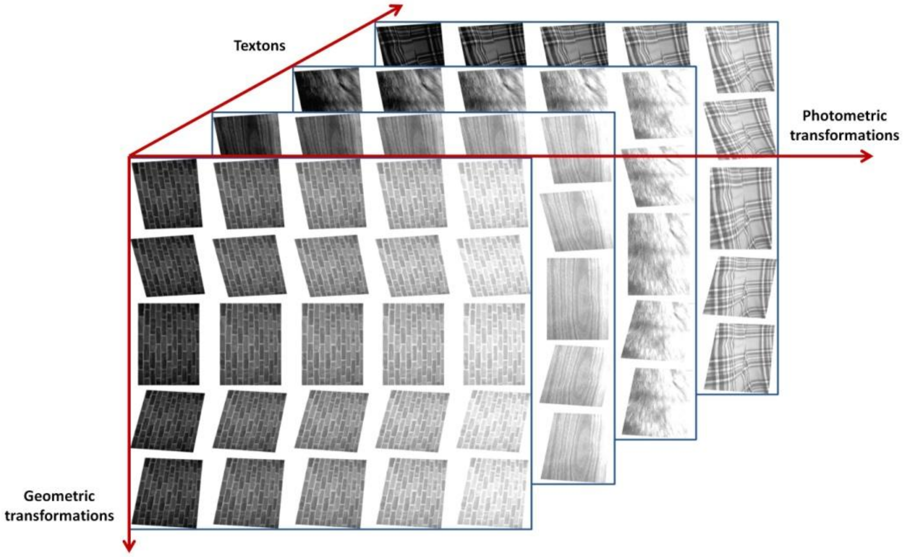

Figure 1: Texture images are generated from interactions of multiple factors including textons, geometric transformations, and photometric transformations, among which, textons are the essential factor for texture representation.

It is common to define texture as a visual pattern with the repetition of a set of basic primitives named textons. Accordingly, a histogram or distribution of textons can be used as an effective representation of texture images. For nature textures, textons can be approximated by the prototypes from clustering local texture patches. However, natural texture images are generated from interaction of multiple factors related to rotation, scaling, lighting, viewpoint, and non-rigid surface deformation, as illustrated in Fig. 1. The multiple factor variations result in severe difficulties for accurately capturing the essential factor, i.e., textons. In this paper, we propose to employ both linear and non-linear embedding approaches to map normalized local texture patches into a texture subspace for analysis. In our framework, the low-dimensional structures hidden in high-dimensional texture observations correspond to the essential factor for texture representation. In this way, the unwanted variation modes resulting from geometric and photometric transformations can be reduced or removed from the essential factor.

The approaches of subspace embedding have been demonstrated effectiveness in mining meaningful low-dimensional structures hidden in original high-dimensional feature space [24, 27]. They are based upon the biological observation that human brain extracts a manageably small amount of perceptually relevant features from high-dimensional sensory inputs (about 10^4 auditory nerve fibers or 10^6 optic nerve fibers) [27]. On the other hand, it also has been explored to transfer the design of local image descriptors to a dimensionality reduction problem in the context of image matching [13, 15]. Our proposed method of subspace embedding textons is mainly inspired by the appearance-based face recognition. As discussed in [3, 12, 20, 28], face images varying in rotation, pose, illumination, and expression reside in a manifold of original data space. Mapping face images into a face subspace is able to conserve the essential factors of person identity but suppresses other factor variations. *Eigenfaces*, *Fisherfaces*, and *Laplacianfaces* are the state-of-the-art embedding algorithms in face recognition literature [3, 12, 28]. *Eigenfaces* and *Fisherfaces* are linear methods which are used to effectively model the Euclidean structure of original feature space. *Laplacianfaces* is a non-linear approach that is able to preserve local data relationships and to discover the subspace of essential factor. Motivated by the success of subspace embedding methods in face recognition, in this paper we explore texture subspaces detected by PCA, LDA, and LPP, and then evaluate our approach in the context of texture classification. Following the conventions in face recognition, we name textons embedded by PCA, LDA, and LPP as *EigenTextons*, *FisherTextons*, and *LaplacianTextons*.

The remainder of this paper is organized as follows. Section 2 reviews existing approaches for texture representation and subspace embeddings. Section 3 describes PCA, LDA, and LPP methods. In Section 4, we provide the detailed procedures of representing texture images using the proposed methods. A variety of experimental results and discussions are presented in Section 5. Finally, Section 6 summarizes the remarks of this paper.

2. Related Work

A major challenge of texture representation is to achieve invariance under a wide range of geometric and photometric variations. Early research work [11, 23] in this domain mainly focused on the analysis of global 2D image transformations including in-plane rotation and scaling. Because of lacking invariance to general geometric transformations, these approaches however cannot effectively model texture images with large 3D transformations such as viewpoint change and non-rigid surface deformation. Multi-fractal analysis has recently been proposed and achieves good resilience to 3D deformations [30, 32, 33]. Texture representations based on this method benefit from the invariance of fractal dimension to geometric transformations. For example, MFS proposed by Xu *et al*. [32, 33] combined fractal dimensions of pixel sets grouped by density functions and orientation templates.

In order to make texture representations more robust to 3D image transformations (e.g., viewpoint change and non-rigid surface deformation) as well as illumination variations, most of recent methods on texture representation rely on extracting local features by local image detectors and descriptors [9, 16, 35]. A textons dictionary is then generated by clustering the extracted local features. For example, Lazebnik *et al*. [16] proposed a texture representation method based on affine-invariant detectors (Harris and Laplacian) and descriptors (RIFT and SPIN). Zhang *et al*. [35] represented textures by combining multiple local image features (SIFT, RIFT, and SPIN). Extensive experiments in texture classification and retrieval have demonstrated that histogram of local image feature is well adapted for texture representation. This is mainly because these textons inherit the resistance to geometric and photometric transformations of local image detectors and descriptors. However, computations of most local image descriptors [1, 2, 10, 18, 25, 31] are complicated and some choices behind their specific designs are not clear. Most descriptors are carefully crafted by hand with many parameters to be manually tuned, such as the number of orientation bins, the number of grids in each support region, and grids sampling schemes (e.g., Cartesian or log-polar). Another limitation is their high dimensions that result in expensive computations in the clustering process to generate textons dictionary. Instead of using manually designed local image descriptors, some research work attempted to employ the data-driven approach to compute texture representations. You and Cai [34] applied a family of PCA subspace decompositions to recognize the very specified wood texture. A null-space based LDA in [17] was used in the frequency domain to perform texture recognition.

Similar to texture images resulting from multiple factors of geometric and photometric transformations, facial images are also formed by interactions of multiple modes related to facial geometry (e.g., person identity and expression), pose, and illumination. In order to disentangle and extract the essential factor, i.e., person identity, for robust and fast face recognition, face images are usually mapped into a face manifold by subspace embedding techniques. Turk and Pentland [28] proposed to use PCA to represent face images.

Belhumeur *et al*. [3] used LDA with the class specific linear projection to compute a face subspace. Both PCA and LDA are linear embedding methods. A number of research efforts have developed to discover the non-linear structure hidden in original image space, e.g., Isomap [27], Local Linear Embedding (LLE) [24], and Lapacian Eigenmap [4]. However, these non-linear approaches suffer the out-of-sample problem, i.e., a subspace yielded by such techniques is only defined on training data but is not able to extend to new testing data. LPP proposed by He *et al*. [12] explicitly addressed this problem. LPP models a subspace by a nearest-neighbor graph where the local structure of original image space is preserved.

Motivated by the similarity of image formation between texture images and face images, we propose to use subspace embedding methods to map texture images into a texture subspace. This enables us to disentangle and extract essential factors of texture images. Compared to representations of local image descriptors, the data-driven textons based upon subspace embeddings are more distinctive, more compact, and with less parameters to tune.

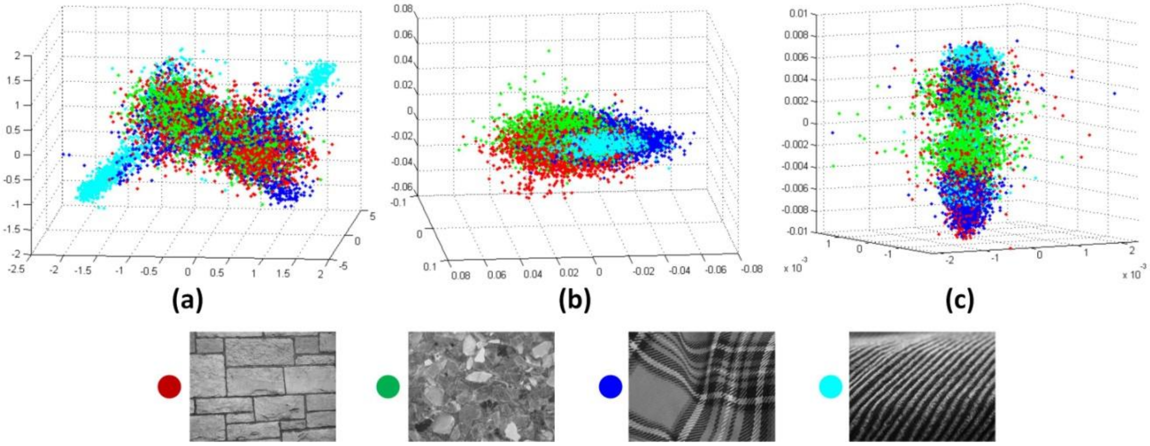

Figure 2: Visualization of normalized texture patches in texture subspaces with top three dimensions computed by PCA (a), LDA (b), and LPP (c). Each color is encoded according to one texture class. For figure clarity, four texture classes from UIUC Texture dataset are visualized.

3. Subspace Embedding Methods

We investigate both linear and non-linear embedding methods to compute a texture subspace. PCA effectively models the Euclidean structure and the variance of entire data. LDA incorporates class specific information and finds the projection that actively discriminates between different categories. LPP preserves intrinsic local structure and detects a non-linear subspace hidden in original data space.

As an illustration, Fig. 2 shows the distributions of normalized local texture patches mapped into a texture subspace with the top three dimensions. In this figure, (a-c) correspond to subspaces obtained by PCA, LDA, and LPP, respectively. The mapping of PCA tends to spread data to capture the factor of the maximum variance. The projection of LDA is based on the factor of texture identities, i.e., to cluster texture patches from the same class close while to separate the ones from different classes far from each other. The embedding of LPP also forms reasonably separated clusters. It maintains the similarities of local patches in the texture subspace and in the original data space.

Let us consider a set of n d-dimensional local texture patches $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$ belonging to l classes. $\boldsymbol{a} \in \Re^{d\times k}$ represents the embedding that maps original data to a new $k(\ll d)$-dimensional texture subspace, where new data $\boldsymbol{y}_i \in \Re^k$ are defined by $\boldsymbol{y}_i = \boldsymbol{a}^T\boldsymbol{x}_i, i = 1,2, \dots, n$.

3.1. EigenTextons of PCA

PCA is an eigenvector approach to model linear variations in the data with high dimensions. The goal of PCA is to construct a series of mutually orthogonal basis that are able to capture the maximum variance directions. It performs embedding by projecting original feature vectors with d dimensions to a k-dimensional linear subspace spanned by k leading eigenvectors of the covariance matrix. The objective function $J(\boldsymbol{a})$ is defined as following:

$$J(\boldsymbol{a}) = \sum_{i=1}^{n} (\boldsymbol{y}_i - \overline{\boldsymbol{y}})^2, \quad \overline{\boldsymbol{y}} = \sum_{i=1}^{n} \boldsymbol{y}_i \tag{1}$$

$$\boldsymbol{a}^* = \arg\max_{\boldsymbol{a}} J(\boldsymbol{a}) \tag{2}$$

The optimal embedding $\boldsymbol{a}^*$ in Eq. (2) is the EigenTextons, which correspond to the basis that maximizes the above objective function.

3.2. FisherTextons of LDA

LDA is a supervised linear subspace embedding algorithm. By encoding class specific information, LDA seeks a projection basis on which data points of different classes are separated far from each other while simultaneously clustering feature points of the same class close to each other. Therefore, the subspace yielded by LDA is efficient for discrimination. The objective function of LDA is:

$$J(\boldsymbol{a}) = \frac{\boldsymbol{a}^T \boldsymbol{S}_B \boldsymbol{a}}{\boldsymbol{a}^T \boldsymbol{S}_W \boldsymbol{a}}, \tag{3}$$

$$\boldsymbol{S}_B = \sum_{i=1}^{l} n_i (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T , \tag{4}$$

$$\boldsymbol{S}_W = \sum_{i=1}^{l} (\sum_{j=1}^{n_i} (\boldsymbol{x}_i^j - \boldsymbol{m}_i)(\boldsymbol{x}_i^j - \boldsymbol{m}_i)^T) \tag{5}$$

where $\boldsymbol{m}$ is the mean vector of local texture patches in training set, $\boldsymbol{m}_i$ is the average feature vector of the ith class, n_i is the number of local texture patches in the ith class, $\boldsymbol{x}_i^j$ is the jth local texture patch in the ith class, l is the number of classes. $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ are between-class scatter matrix and within-class scatter matrix, where the class specific information is incorporated. The optimal mapping basis $\boldsymbol{a}^*$ is the maximizer of the objective function in Eq. (3). It can be solved by a generalized eigenvalue problem defined in Eq. (6). Note the upper bound of reduced dimension $k = l - 1$ as there are at most $l - 1$ nonzero generalized eigenvalues.

$$\boldsymbol{S}_B \boldsymbol{a}^* = \lambda \boldsymbol{S}_W \boldsymbol{a}^* \tag{6}$$

In the case of face recognition, $\boldsymbol{S}_W$ usually becomes singular. This stems from the fact that the rank of $\boldsymbol{S}_W$ is less than or equal to $n - l$, but the number of training images n is much smaller than the number of pixels d in each image. In texture representation, this difficulty however can be avoided. In our framework,

n is the number of local patches in texture images of training set. This number is much larger (10^3) than the amount of images. In addition, the dimension d of each local texture patch is far smaller than the dimension of the entire image.

It was observed in [7] that the coefficients of $\boldsymbol{a}^*$ used to map spatially smooth features (e.g., local texture patches) tend to become spatially rough. We take the scheme of spatially smooth regularization in [7] to smooth and stabilize the mapping coefficients. Spatially smooth regularization takes advantage of the spatial relationships between pixels within each local texture patch and makes the embedding coefficients smoother and more stable. The regularized optimal embedding $\boldsymbol{a}^*$ is the FisherTextons.

3.3. LaplacianTextons of LPP

LPP is a non-linear subspace embedding approach that aims to preserve intrinsic geometry of original data space. It concentrates on discovering the manifold structure hidden in original space by modeling an adjacency graph. LPP addresses the out-of-sample problem of most non-linear embedding techniques. The embedding obtained by LPP is defined on both training and testing data. The objective function of LPP is defined as:

$$J(\boldsymbol{a}) = \sum_{i,j} (\boldsymbol{y}_i - \boldsymbol{y}_j)^2 \boldsymbol{S}_{ij} \,, \qquad (7)$$

$$\boldsymbol{S}_{ij} = \begin{cases} \exp\left(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2 / \gamma\right), & \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2 < \epsilon \\ 0, & otherwise \end{cases} \qquad (8)$$

where $\boldsymbol{S}$ is the adjacency matrix that measures the similarity between each pair of local texture patches $(\boldsymbol{x}_i, \boldsymbol{x}_j)$. ϵ defines the range of local neighborhood. γ is a constant scalar value. The intrinsic geometry of original data space is captured by $\boldsymbol{S}$. The objective function incurs heavy penalty if a pair of neighboring texture patches are mapped far apart. So it seeks to obtain such an embedding that, if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are close, they will be mapped to a subspace where $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are close as well. The optimal embedding $\boldsymbol{a}^*$ that minimizes Eq. (7) can be solved by the generalized eigenvalue problem:

$$\boldsymbol{XLX}^T\boldsymbol{a} = \lambda \boldsymbol{XDX}^T\boldsymbol{a} \qquad (9)$$

where $\boldsymbol{D}$ is a diagonal matrix with $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{S}_{ji}$. $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S}$ is the Laplacian matrix. The minimum eigenvalue solution $\boldsymbol{a}^*$ of Eq. (9) is the LaplacianTextons.

4. Texture Representation Framework

Our proposed framework to build effective texture representations is described in this section. A keypoint detector is first used to localize texture regions. We then normalize detected regions to make local texture patches invariant to scaling and rotation. The normalized texture patches are then mapped to a texture subspace using the embedding approaches described in Section 3. A textons dictionary generated from training set is employed to quantize embedded normalized texture patches. A texture image is in the end represented as a histogram of textons.

4.1. Region Detection and Normalization

We begin with a keypoint detector to search salient local image structures. The keypoint detector provides support regions of local texture patches. In this paper we adopt Harris-Laplace detector and Hessian-Laplace detector [21] as keypoint detectors. Both of them are rotation and scaling invariant. Harris-Laplace detector responses to corner-like structures, and Hessian-Laplace detects blob-like structures. They provide salient, complimentary, and sufficient local texture regions.

The support regions are three times larger than the detected regions in order to include more signal changes. All the support regions are first smoothed to reduce noise and aliasing and then normalized to a fixed patch size of 41×41 that provides sufficient resolution. A similar patch size was used in [21]. As most state-of-the-art local image descriptors [2, 10, 15, 18], a dominant orientation of a patch is computed based on gradient information. The dominant orientation corresponds to the largest bin of a histogram of gradient orientation weighted by gradient magnitudes and smoothed by a Gaussian window. A patch is then rotated to align its dominant orientation to a canonical direction. This normalization process simplifies the subspace modeling problem for embedding algorithms as variations of rotation and scaling are significantly suppressed.

4.2. Offline Computation of Embeddings

We compute embeddings $\boldsymbol{a}^*$ using the algorithms described in Section 3. For texture recognition, the embeddings can be pre-computed once and stored. It is important to note that the embeddings are computed based upon normalized local image patches rather than the entire images as used in face recognition. We explore two channels of normalized texture patches to compute embeddings: (1) image channel, i.e., local image patch with $41 \times 41 = 1681$ dimensions; (2) gradient channel, i.e., horizontal and vertical gradients with $2 \times 39 \times 39 = 3042$ dimensions. By using two channels of training set, we learn three embeddings: *EigenTextons*, *FisherTextons*, and *LaplacianTextons*. As discussed in Section 3, the upper bound of reduced dimension of LDA is $l - 1$. We make this number as the reduced dimension for LDA. To keep good performance and consistency with LDA, we also use the first $l - 1$ dimensions of PCA and LPP.

5. Experiments and Discussions

The proposed texture representation approaches are evaluated in the context of texture classification. As discussed in Sections 3 and 4, we have three embedding methods and two feature channels. So there are 6 different combinations of texture representations that are investigated in our experiments as shown in Table 1. We extensively compare the performances of our proposed methods with the existing state-of-the-arts. They are tested on two public available datasets: UIUC Texture [16] and UMD Texture [33]. In addition to in-plane rotation and scaling change presented in traditional datasets [6, 8, 29], the two datasets as shown in Fig. 3 capture more challenging variations including viewpoint, illumination, and non-rigid surface deformation.

Table 1: Texture representations based upon different combinations of embeddings and feature channels.

Embeddings	**Image Channel**	**Gradient Channel**
EigenTextons	PCA-Img	PCA-Grad
FisherTextons	LDA-Img	LDA-Grad
LaplacianTextons	LPP-Img	LPP-Grad

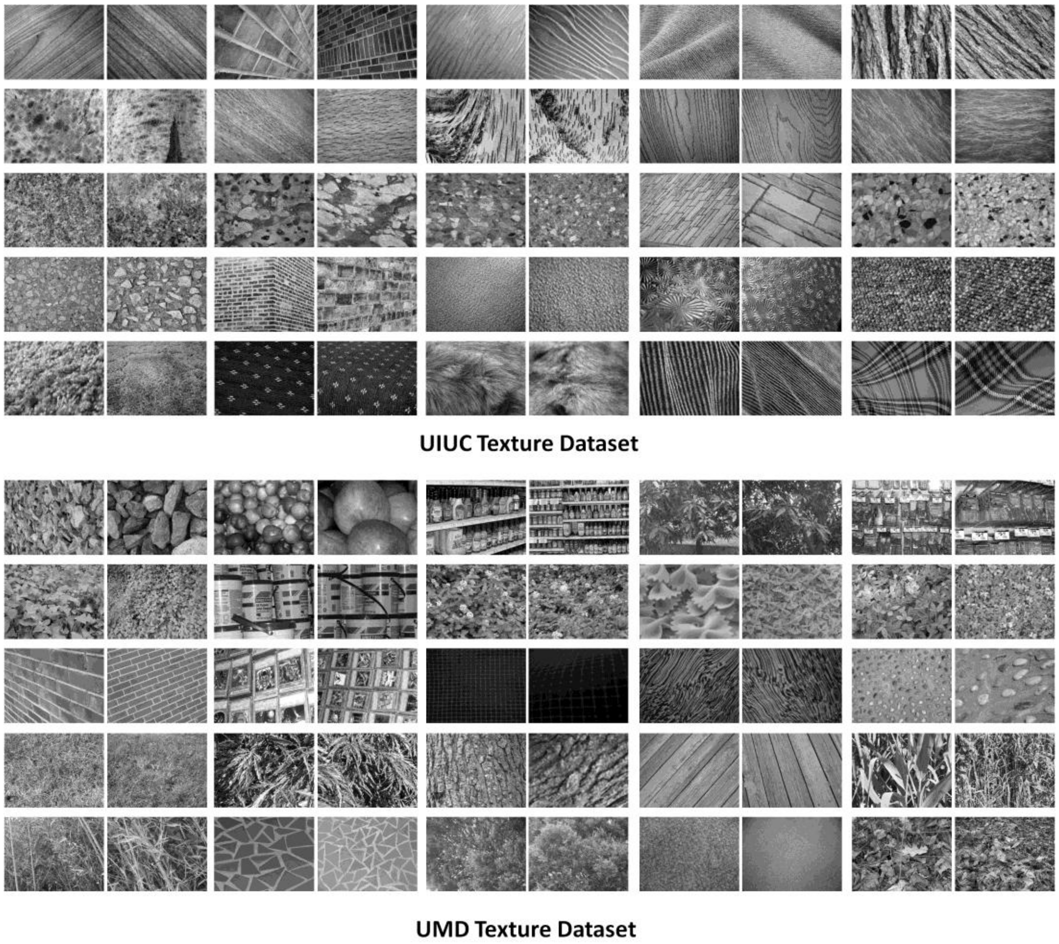

Figure 3: Two sample images of 25 texture categories in UIUC and UMD Texture Datasets.

5.1. Experimental Setup

The UIUC dataset includes 25 texture classes and 40 images with the resolution of 640×480 in each class. These images present strong rotation, scaling, viewpoint variation, non-rigid surface deformation, and lighting change. The UMD dataset consists of 1000 uncalibrated and unregistered images with the resolution of 1280×960 pixels. It contains 25 texture categories with 40 images for each class. These images are also taken under significant geometric and photometric transformations. We downsample original images of UMD dataset to the resolution of 640×480.

In order to facilitate a fair comparison, we follow the standard experimental setting to randomly select a portion of images from each class as the training set. The remaining images are used as the testing set. The training process is based on each corresponding randomly generated training set. The reported recognition accuracy rates in the following experiments are the average results over 50 runs by the random generated training and testing sets. K-means clustering ($K = 100$) is employed to build the textons dictionary. We employ Support Vector Machines (SVMs) with RBF kernels as the classifier. The optimal parameters of RBF kernels are obtained by 5-fold cross-validation. The SVMs classifier in essence finds the hyperplane that separates two-class data with maximal margin. In order to apply SVMs for multi-class problem, we take the *one-versus-one* strategy.

We compare the proposed methods against the state-of-the-art approaches including VG [30], MFS [33], Lazebnik [16], Zhang [35], SIFT [18], SURF [2], DAISY [31], ORB [25], CARD [1], and MROGH [8]. VG makes use of local density function properties of a set of image measurements. MFS combines the fractal dimensions of pixel sets grouped by three local density functions. Lazebnik extracts local image features by RIFT and SPIN from affine regions. Zhang combines local features by multiple local image detectors and descriptors. SIFT, SURF, DAISY, ORB, CARD, and MROGH are the most recent developed local image descriptors. They achieve the state-of-the-art performances in the context of object, texture, and scene classification.

5.2. Evaluations of Different Combinations of Embeddings and Feature Channels

The classification accuracies for different combinations of embeddings and feature channels on UIUC dataset and UMD dataset are shown in Fig. 4. The numbers of training images for each class are from 1 to 20 and the rest images are used for testing. Similar conclusions can be drawn from experiments on the two datasets.

As shown in the two figures, for each embedding method, the performances based on gradient channel outperform the ones based on image channel. This is probably because gradient is more resistant to lighting variations and preserves relative changes in intensity. Thus, gradient feature simplifies embedding methods to model the essential factor of texture images. The difference between two channels of LPP is more evident. As shown in Eq. (8), LPP measures pair-wise similarities of local texture patches. So the adjacency matrix $\boldsymbol{S}$ is more sensitive to illumination, which results in that LPP is relatively less robust to handle lighting change.

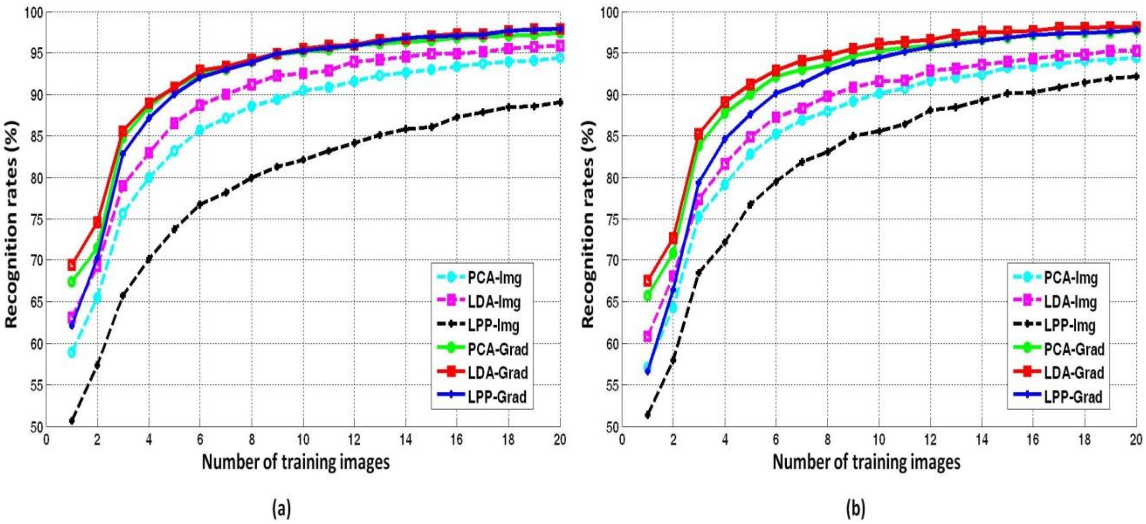

Figure 4: Recognition accuracy for different combinations of embedding methods and feature channels on (a) UIUC dataset and (b) UMD dataset.

LDA-Grad dominates the recognition rates when the training number is small. By explicitly encoding class labels in computing the texture subspace, LDA is forced to concentrate on the essential factor of texture identities, i.e., textons. PCA-Grad and LPP-Grad also demonstrate impressive performances. When the training images are sufficient (e.g., > 10), the performances of PCA-Grad and LPP-Grad are comparable to LDA-Grad. But LPP-Grad is more sensitive to the number of training images because non-linear methods require denser sampling of a manifold to reasonably recover the intrinsic structure.

5.3. Comparisons with the State-of-the-arts

Based on the evaluation results from different combinations of embedding methods and feature channels, we choose LDA-Grad to compare with the state-of-the-art methods for texture recognition on UIUC dataset and UMD dataset. The experimental results are shown in Table 2. N_t denotes the number of training images in each class. The best recognition rates of various training numbers are the numbers in bold. We can also obtain similar conclusions on both datasets.

Table 2: Recognition rates of proposed method compared with the state-of-the-art approaches on (top) UIUC dataset and (bottom) UMD dataset. $\boldsymbol{N_t}$ is the number of training images in each class.

$\boldsymbol{N_t}$	VG [30]	MFS [33]	Lazebnik [16]	Zhang [35]	SIFT [18]	SURF [2]	DAISY [31]	ORB [25]	CARD [1]	MROGH [10]	Our method
5	82.86	82.24	91.12	88.62	**91.96**	90.73	86.80	79.03	73.99	88.76	90.86
10	87.85	88.36	94.42	93.17	95.42	95.15	92.54	86.26	83.00	94.13	**95.55**
15	90.62	91.38	96.64	95.33	96.87	96.14	94.16	89.40	87.18	95.93	**97.07**
20	92.31	92.74	97.02	96.67	97.84	96.75	95.21	90.73	89.69	96.82	**97.91**

$\boldsymbol{N_t}$	VG [30]	MFS [33]	Lazebnik [16]	Zhang [35]	SIFT [18]	SURF [2]	DAISY [31]	ORB [25]	CARD [1]	MROGH [10]	Our method
5	90.92	85.63	90.71	91.56	**91.68**	90.41	90.81	81.85	84.38	90.39	91.23
10	94.09	90.82	94.54	96.00	96.01	94.49	94.92	87.87	90.41	94.54	**96.06**
15	96.22	92.67	96.29	96.79	97.21	96.13	96.47	90.87	93.05	96.01	**97.59**
20	96.36	93.93	96.95	97.62	97.64	96.98	97.58	92.84	94.23	97.03	**98.20**

The results in the two tables show that our proposed method outperforms the state-of-the-art approaches in most cases. For example, our method significantly and consistently outperforms ORB and CARD, both of which are the most recent state-of-the-art local image descriptors. The performance of our method is also much better than texture representations based on fractal analysis, i.e., VG and MFS. In most cases, our approach achieves better performances than the remaining methods that are based on the state-of-the-art local image descriptors. The impressive performances based on sophisticated descriptors originate from the resistance to photometric and geometric transformations of local image descriptors. Compared local image descriptors that are carefully crafted by hand, out methods are totally data-driven. It is based on the construction of a texture subspace where the essential factor (textons) is manifested but unwanted variation factors are reduced or removed. Our method is inferior to SIFT when $N_t = 5$. This is probably due to the fact that only 5 training images cannot provide sufficiently dense sampling of a texture subspace. The computation of embeddings is therefore biased by the rough sampling.

5.4. Computational Cost of Textons

The experimental results have demonstrated that textons-based methods are well-adapted for texture representation. In natural texture images, textons can be generated by clustering local texture features. However, the clustering process is always time consuming. If the clustering problem is exactly solved, the computational cost of K-means is $O(n^{dc+1}\log n)$ [14], where n is the number of local texture features to be clustered; c is the number of centers; and d is the dimension of feature. So when n and c are fixed, feature with fewer dimensions are able to reduce the computational cost and speed up clustering process.

The local image features computed by most descriptors are with high dimensions which result in expensive computations. As discussed in Section 3.2, the upper bound of reduced dimension of FisherTextons is $l-1$, where l is the number of classes. Both UIUC dataset and UMD dataset contains 25 classes. So we use 24 as the reduced dimension of textons. Fig. 5 compares the running time in each iteration of clustering on UIUC dataset on an Intel Core2 CPU 2.13GHz computer. We use 20 images of each class as the training set and extract approximately 3000 local patches from each image. We set the number of clustering centroids $c=100$. In the experiments, our method significantly reduces the running time compared with most state-of-the-art descriptors. Note the running time difference can become huge when a clustering needs a large number (e.g., 1000) of iterations.

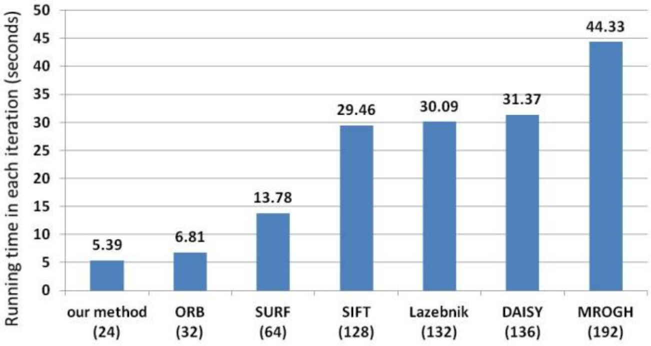

Figure 5: The running times of different methods in each iteration of K-means clustering. The numbers in parenthesis under each method denote corresponding feature dimensions.

5.5. Discussions

The experimental results in the context of texture recognition have validated the effectiveness of our proposed texture representation methods. They also reveal a number of interesting points:

First, in all embedding methods, gradient channel consistently performs better than image channel, especially for LPP. These experimental results demonstrate that gradient channel is more suitable for embedding approaches to model the texture subspace as gradients suppress lighting variation but preserve relative intensity change.

Second, both of the linear embedding methods with gradient channel achieve the state-of-the-art classification results. PCA provides the benefits of capturing the maximum variance of original data space but reducing noisy variations. This also confirms that the major factors, i.e., the leading eigenvectors, of texture images correspond to texture identities even though significant variations are presented. LDA encodes the class specific information in the texture subspace which enables the mapping actively discriminates between different texture classes.

Third, the non-linear embedding method, i.e., LPP-Grad, also achieves impressive performance on both datasets but is inferior to LDA-Grad when the training samples are insufficient. This is different from the observation in face recognition domain where non-linear methods consistently outperform linear ones. This might be explained by the difference of data sampling. Facial images are always densely sampled, which forms smooth variations in terms of pose and expression. The smooth changes make the Euclidean distance $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ in Eq. (8) more accurate as they are small enough to preserve the geodesic distance or the intrinsic geometry hidden in a facial manifold. In contrast, images in texture datasets always present drastic changes which are not smooth enough to capture local structures in original texture manifold.

6. Conclusion

In this paper, we have proposed several texture representations by subspace embeddings. To the best of our knowledge, this is the first work on texture representation that systematically and explicitly considers the texture subspace using both linear and non-linear embedding algorithms. The experimental results on benchmark texture datasets have demonstrated the texture subspace computed by embedding methods is effective to disentangle and extract the essential factor of texture images from the interactions of multiple factors resulting from geometric and photometric transformations. The experimental results also show that the state-of-the-art performances on existing texture classification datasets are now near ceiling (e.g., > 97%). But in addition to classification accuracy, our methods significantly improve the computational costs and are totally data-driven with much fewer parameters to tune. The experiments have validated that textons mapped into a texture subspace have strong resistance to image deformations, meanwhile, are more distinctive and more compact. The future work will focus on effective combinations (e.g., through Multiple Kernel Learning) of texture representations computed from different embedding methods.

7. Acknowledgement

This work was supported in part by NSF grant IIS-0957016, EFRI-1137172, NIH 1R21EY020990, ARO grant W911NF-09-1-0565, and FHWA DTFH61-12-H-00002.

8. References

[1] M. Ambai and Y. Yoshida. CARD: Compact and Real-time Descriptors. *International Conference on Computer Vision*, 2011.

[2] H. Bay, A. Ess, and L. Gool. SURF: Speed Up Robust Features. *Computer Vision and Image Understanding*, 110(3): 346-359, 2008.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7): 711-720, 1997.

[4] M. Belkin and P. Niyogi. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. *Neural Information Processing Systems*, 2001.

[5] N. Bouguila. Inifinite Liouville Mixture Models with Application to Text and Texture Categorization. *Pattern Recognition Letters*, vol.33, iss.2, pp.103-110, 2012.

[6] P. Brodatz. Texture: A Photographic Album for Artists and Designers. 1996.

[7] D. Cai, X. He, Y. Hu, J. Han, and T. Huang. Learning A Spatially Smooth Subspace for Face Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[8] B. Caputo, E. Hayman, M. Fritz, and J. Eklundh. Classifying Materials in the Real World. *Image and Vision Computing*, 28(1), 150-163, 2010.

[9] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual Categorization with Bags of Keypoints. *European Conference on Computer Vision*, 2004.

[10] B. Fan, F. Wu, and Z. Hu. Aggregating Gradient Distributions into Intensity Orders: A Novel Local Image Descriptor. *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[11] R. Haralick. Statistical and Structural Approaches to Texture. *Proc. IEEE*, 1979.

[12] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang. Face Recognition Using Laplacianfaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*. 27(3): 328-340, 2005.

[13] G. Hua, M. Brown, and S. Winder. Discriminant Embedding for Local Image Descriptors. *International Conference on Computer Vision*, 2007.

[14] M. Inaba, N. Katoh, and H. Imai. Application of Weighted Voronoi Diagrams and Randomization to Variance-based K-clustering. *ACM Symposium on Computational Geometry*, 1994.

[15] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[16] S. Lazebnik, C. Schmid, and J. Ponce. A Sparse Texture Representation Using Local Affine Regions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(8): 1265-1278, 2005.

[17] S. Liao and C. Chung. A New Subspace Learning Method in Fourier Domain for Texture Classification. *International Conference on Image Processing*, 2010.

[18] D. Lowe. Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*, 2004.

[19] A. Mailing and B. Cernuschi-Frias. A Method for Mixed States Texture Segmentation with Simultaneous Parameter Estimation. *Pattern Recognition Letters*, vol.21, iss.15, pp.1982-1989, 2011.

[20] A. Martinez and A. Kak. PCA vs. LDA. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(2): 228-233, 2001.

[21] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(10): 1615-1630, 2005.

[22] K. Nguyen, B. Sabata, and A. Jain. Prostate Cancer Grading: Gland Segmentation and Structural Features. *Pattern Recognition Letters*, vol.33, iss.7, pp.951-961, 2012.

[23] T. Randen and J. Husoy. Filtering for Texture Classification: A Comparative Study. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(4): 291-310, 1999.

[24] S. Roweis and L. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 2000.

[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An Efficient Alternative to SIFT and SURF. *International Conference on Computer Vision*, 2011.

[26] X. Yang, S. Yuan, and Y. Tian. Recognizing Clothes Patterns for Blind People by Confidence Margin based Feature Combination. *ACM Multimedia*, 2011.

[27] J. Tenenbaum, V. Silva, and J. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, vol. 290, 2000.

[28] M. Turk and A. Pentland. Face Recognition Using Eigenfaces. *IEEE Conference on Computer Vision and Pattern Recognition*, 1991.

[29] M. Varma and A. Zisserman. A Statistical Approach to Material Classification Using Image Patches. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(11), 2032-2047, 2009.

[30] M. Varma and R. Garg. Locally Invariant Fractal Features for Statistical Texture Classification. *International Conference on Computer Vision*, 2007.

[31] S. Winder, G. Hua, and M. Brown. Picking the Best DAISY. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[32] Y. Xu, H. Ji, and C. Fermuller. A Projective Invariant for Textures. *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[33] Y. Xu, H. Ji, and C. Fermuller. Viewpoint Invariant Texture Description Using Fractal Analysis. *International Journal of Computer Vision*, 83(1): 85-100, 2009.

[34] M. You and C. Cai. Wood Classification based on PCA, 2DPCA, $(2D)^2$PCA, and LDA. *International Symposium on Knowledge Acquisition and Modeling*, 2009.

[35] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *International Journal of Computer Vision*, 73(2): 213-238, 2007.