TEXTURE CLASSIFICATION VIA PATCH-BASED SPARSE TEXTON LEARNING

Jin Xie, Lei Zhang, Jane You and David Zhang

Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong, China
Email: {csjxie, cslzhang, csyjia, csdzhang}@comp.polyu.edu.hk

ABSTRACT

Texture classification is a classical yet still active research topic in the past years. Recently, several new texture classification approaches based on modeling texture images as distributions over a set of textons have been proposed. These textons are learned as the cluster centers in the image patch feature space using the *K*-means clustering algorithm. However, the Euclidian distance based the *K*-means clustering process may not be able to well characterize the intrinsic feature space of texture textons, which is often embedded into a low dimensional manifold. Inspired by the great success of l_1-norm minimization based sparse representation (SR), in this paper we propose a novel texture classification method via sparse patch texton learning. Specifically, the dictionary of textons is learned by applying SR to image patches in the training dataset. The SR coefficients of the test images over the dictionary are used to construct the histograms for classification. Experimental results showed that the proposed method yields good performance.

***Index Terms*—**Texture classification, texton, sparse representation, *K*-means

1. INTRODUCTION

Texture classification is an important research topic in computer vision and pattern recognition applications, such as image understanding and object recognition. With the increasing demand of such applications, texture classification has been receiving considerable attention over the past decades. The use of co-occurrence matrices [1] is still a popular texture classification approach, relying on thenon-parametric statistics at the pixel level. Rotation invariance texture features can be extracted by using polar coordinate systems [2]. Ojala et al [3] proposed the Local Binary Pattern histogram as the rotation invariant texture feature.

However, the early traditional methods are sensitive to changes in viewpoint. Some recent approaches are proposed to solve this problem. Lazebnik et al [4] employed some invariant descriptors on affine invariant regions in texture images to extract texture features. Xu [5] et al proposed the multi-fractal spectrum vectors to describe textures while achieving global invariance. Leung and Malik [6] proposed to classify texture images by using three dimensional (3D) textons, which are cluster centers of filter responses over a stack of images with representative viewpoints and illuminations. Varma and Zisserman [7, 8] modeled texture images as distributions over a set of textons, which are learned from the responses of MR8 filter banks [7]. Furthermore, in [9, 10], good performance can also be obtained by textons learned from patches in the original image instead of MR8 filter responses.

In these texton based methods, the textons are usually learned by the *K*-means clustering algorithm. However, the *K*-means clustering algorithm is based on the l_2-norm Euclidean distance so that the elements of a cluster will have a ball-like distribution. The learned *K* ball-like clusters, nonetheless, may not be able to characterize reasonably well the intrinsic feature space of the texture images, which is often embedded into a lower dimensional manifold.

Recently, the theory and algorithms of sparse coding or sparse representation (SR) [11, 12] have been successfully used in image processing and pattern recognition [13, 14, 18]. The principle of SR reveals that a given natural signal can be often sparsely represented as the linear combination of an over-complete dictionary via l_1-norm minimization [12, 15]. Inspired by the great success of SR and patch textons used in [9, 10], in this paper we propose a sparse patch texton learning method for texture classification. A texton training dataset is first constructed from patches in the training images, and then an over-complete dictionary of patch textons is computed under the SR framework. A histogram feature of SR coefficients can be extracted for texture classification by coding the texture image with the patch texton dictionary. It will be seen that the proposed method can achieve better texture classification performance than the state-of-the-art texton based texture classification method using textons learned by *K*-means clustering.

The rest of the paper is organized as follows. Section 2 briefly reviews the concepts of SR. Section 3 describes in detail the sparse patch texton learning process and the SR coefficient histogram construction for classification. In Section 4, the proposed method is validated on the CUReT database. Section 5 concludes the paper and indicates our future work.

2. SPARSE REPRESENTATION OF SIGNALS

In recent years there has been a growing interest in the study of SR of signals. The success of SR largely owes to the fact that natural signals are intrinsically sparse in some domain. Therefore, if the dictionary that is used to define the sparse domain can be well trained, a good analysis of the input signal can be expected by representing over the dictionary.

For a given signal $x \in R^m$, we say that x has a sparse approximation over a dictionary $D=[\boldsymbol{d}_1,\boldsymbol{d}_2,\ldots,\boldsymbol{d}_l] \in R^{m\times l}$, if we can find a linear combination of only "a few" atoms from D that is "close" to the signal x. Under this assumption, the sparsest representation of x over D is the solution of

$$\arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_p \text{ s.t. } \|x - D\boldsymbol{\alpha}\|_2 \le \varepsilon \qquad (1)$$

where $\|\boldsymbol{\alpha}\|_p$ is a sparsity-inducing regularization term.

In some applications, the dictionary D is unknown and we need to learn it from a training dataset $X=[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n]\in R^{m\times n}$. It is expected that each training sample (i.e. each column of X) can be sparsely represented over the dictionary D, i.e. $\boldsymbol{x}_i=D\boldsymbol{\alpha}_i$ and only a few elements in $\boldsymbol{\alpha}_i$ are significant. The dictionary D, as well as the SR coefficient vectors $\boldsymbol{\alpha}_i$, can be solved by optimizing the following objective function

$$\arg\min_{D,\{\boldsymbol{\alpha}_i\}} \sum_{i=1}^{n} \|\boldsymbol{\alpha}_i\|_p \text{ s.t. } \|X - D\Lambda\|_F^2 \le \varepsilon \qquad (2)$$

where $\Lambda=[\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2,\ldots, \boldsymbol{\alpha}_n]$ and $\|\bullet\|_F$ is the Frobenius matrix norm.

3. TEXTURE CLASSIFICATION VIA SPARSE PATCH TEXTON LEARNING

In this section, we propose to learn the dictionary of textons under the SR framework, and then use the features based on sparse patch textons to classify texture images. In our work, the textons are learned from the original image patches. Therefore, in the following sub-sections, we first briefly introduce the pre-processing for training dataset construction, and then present the details of sparse patch texton learning and texture classification.

3.1. Patch texton learning by SR

Before learning patch textons, all texture images are converted to grey level images and are normalized to have zero mean and unit standard deviation. The normalization offers certain amount of invariance to the illumination changes. A square neighborhood around each pixel in the image is taken and a vector is formed along the row. In addition, patch vectors are contrast normalized using Weber's law. Hence, for each class of texture images, we can construct a training dataset $X=[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n]$, where $\boldsymbol{x}_i$, i=1,2,…, n, is the patch vector at a position in a training sample image of this class.

The dictionary of textons, denoted by $D=[\boldsymbol{d}_1,\boldsymbol{d}_2,\ldots,\boldsymbol{d}_l]$, can be learned from the constructed training dataset X, where $\boldsymbol{d}_j$, j=1,2,…, l, is one of the l textons. In [9, 10], the classical K-means clustering method was employed to determine the l patch textons by solving the following problem:

$$\arg\min_{D} \sum_{j=1}^{l} \sum_{\boldsymbol{x}_k\in\Omega_j} \|\boldsymbol{x}_k - \boldsymbol{d}_j\|_2^2 \qquad (3)$$

Obviously, the K-means clustering will partition the dataset X into l groups $\Omega_1,\ldots,\Omega_j,\ldots,\Omega_l$, and the texton $\boldsymbol{d}_j$ is defined as the mean vector of the vectors within Ω_j. However, as we explained in the Introduction, by using K-means clustering, the elements belong to the same cluster will distribute within a ball because the l_2-norm Euclidean distance is used in the clustering process. These ball-like clusters will cover the whole feature space. Nonetheless, such a dense coverage may not be able to effectively characterize the intrinsic feature space of texture images, which is often embedded into a lower dimensional manifold.

Let $\Lambda=[\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2,\ldots, \boldsymbol{\alpha}_n]$, the SR objective function in Eq. (2) is adopted to optimize D and $\{\boldsymbol{\alpha}_i\}$, and here we re-write it as follows by setting p=1:

$$\arg\min_{D,\Lambda} |\Lambda|_1 \ s.t. \|X - D\Lambda\|_F^2 \le \varepsilon$$

(4)

In practice, it is more convenient to convert Eq. (4) into an unconstrained optimization problem by using a form of l_1-penalized least-squares:

$$\arg\min_{D,\Lambda} \|X - D\Lambda\|_F^2 + \lambda\|\Lambda\|_1 \qquad (5)$$

Eq. (4) and (5) are equivalent with an appropriate parameter λ, which is used to balance the l_1-norm and l_2-norm terms in Eq. (5). In this paper, we adopted the recently proposed feature-sign search method [16] to solve Eq. (5) because this algorithm is more efficient.

In some sense, the K-means clustering method can be viewed as a special case of the SR based clustering in Eq. (5). If we let $\boldsymbol{\alpha}_i$ has only one non-zero element and let this non-zero element be 1, then Eq. (5) will be basically the same as Eq. (3). In this case, we use only one texton to represent the feature vector $\boldsymbol{x}_i$ and assign the label of $\boldsymbol{x}_i$ to that texton. In contrast, by using SR, $\boldsymbol{x}_i$ or any input vector y will be coded as a linear combination of more than one texton. Therefore, SR can achieve a much lower reconstruction error due to the less restrictive constraint. In addition, for an input vector y which may lie in the boundary of two or more clusters, the K-means clustering will randomly assign it to one of the classes. Such a representation may not be efficient enough in practice. In the experiments in Section 4, we will see that by using Eq. (5) to learn the textons and using the associated feature description method in Section 3.2, the texture classification accuracy can be improved.

3.2. Feature description and texture classification

Denote by D_k the texton dictionary for the k^{th} texture class, the dictionary for all the c classes of texture images can be formed by amalgamating the K dictionaries

$$D=[D_1, D_2,\ldots, D_c] \tag{6}$$

With this dictionary D, each training texture image can generate a model by mapping it to the texton dictionary. In Varma and Zisserman's method [9, 10], for each position of a training image, it is labeled with the elements in the texton dictionary D that is closest to the image patch vector at this position. Therefore, a histogram can be formed by normalizing the frequencies of texton labels of this image.

Different from the method in [9, 10], we can construct a histogram of the SR coefficients of a training image as the texture model. Denote by $\boldsymbol{x}_i$ the image patch vector at position i of a training image, we can represent $\boldsymbol{x}_i$ over D by SR to get the representation coefficient vector. However, this can be computationally very expensive because D can be very big. Let $D=[\boldsymbol{d}_1,\boldsymbol{d}_2,\ldots,\boldsymbol{d}_z]$, where z is the total number of textons learned from the c classes. In practice, we can use only a subset of D to represent $\boldsymbol{x}_i$. Specifically, we use the closest t textons ($t<<z$) to $\boldsymbol{x}_i$ in D to form the sub-dictionary for $\boldsymbol{x}_i$. Denote by $\boldsymbol{d}_1^i,\ldots,\boldsymbol{d}_t^i$ the t closest textons to $\boldsymbol{x}_i$, the sub-dictionary for $\boldsymbol{x}_i$ is then $D_i=[\boldsymbol{d}_1^i,\ldots,\boldsymbol{d}_t^i]$. The representation vector of $\boldsymbol{x}_i$ over D_i, denote $\boldsymbol{\alpha}_i=[\boldsymbol{\alpha}_1^i,\ldots,\boldsymbol{\alpha}_t^i]$, can then be computed by solving he following l_1-norm minimization problem:

$$\arg\min_{\boldsymbol{\alpha}_i} \|\boldsymbol{x}_i - D_i\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1 \tag{7}$$

The l_1-least square method in [10] can be used to solve Eq. (7).

Since the textons $\boldsymbol{d}_1^i,\ldots,\boldsymbol{d}_t^i$ in D_i have a one-to-one correspondence to the textons in D, by using $\boldsymbol{\alpha}_i$ we can easily construct another representation vector $\boldsymbol{h}_i$ of $\boldsymbol{x}_i$ over D such that

$$D_i\boldsymbol{\alpha}_i = D\boldsymbol{h}_i \tag{8}$$

Obviously, most of the entries in $\boldsymbol{h}_i$ will be 0, and only the entries corresponding to the same textons as those in D_i will have non-zero values, and these values are the same as those in $\boldsymbol{\alpha}_i$.

Finally, at each position i of a training texture image, we have a representation vector $\boldsymbol{h}_i$. Because the coefficients in $\boldsymbol{h}_i$ are real numbers instead of integers, we can form a fractional histogram, denoted by H_f, for this texture image by summing all the vectors of $\boldsymbol{h}_i$:

$$H_f = \sum \boldsymbol{h}_i \tag{9}$$

The fractional histogram H_f can serve as the texture model.

Denote by H_i, i =1,2,…, n, the model histograms in the database. Similarly, for an input test image Y, we can construct the sparse texton histogram for it, denoted by H_y. The similarity between H_i and H_y is computed:

$$\chi^2(H_{A_i}, H_{B_i}) = \frac{1}{2}\sum \frac{(H_{A_i} - H_{B_i})^2}{H_{A_i} + H_{B_i}} \tag{10}$$

The texture image Y is classified to the corresponding class by a nearest neighbor classifier.

4. EXPERIMENTAL RESULTS

In this section, we evaluate the proposed texture classification method on the CUReT database [17]. The CUReT texture database contains 61 classes, each consisting of 205 images. Here we choose 92 images per class for which a large region of texture is visible across all textures. There are a number of factors that make the CUReT texture database challenging. It has both large inter-class confusion and intra-class variation. The images of a class are obtained under unknown viewpoint and illumination, and some different classes look similar in appearance.

The evaluation methodology on the CUReT database is as follows: M images are chosen per class for training and the remaining 92-M images per class are used to form the test set. In the experiment, 46, 23, 12, and 6 texture images per class are chosen every some images as the training set. Table 1 compares the performance of our proposed method with the state-of-the-art texton based method proposed by Varma and Zisserman [9, 10] (we denote this method by VZ_Patch). For the VZ_Patch and proposed method, a 9×9 neighborhood around each pixel is taken and an 81 dimensional feature vector is formed. And 40, 30 and 20 textons per texture class were learned, which resulted in 2440, 1830 and 1220 dimensional histograms for texture representation respectively.

When M=46, the proposed method achieves the accuracies of 97.68%, 97.54% and 97.22% using 40, 30 and 20 textons per class respectively, while the VZ_Patch method achieves the accuracies of 97.11%, 97.08% and 96.61%. With the decrease of the number of training samples, the method can achieve higher accuracies than the VZ_Patch method. Note that in this experiment we did not compare our method with another two state-of-the-art methods, the Lazebnik's method [4] and Xu's method [5]. This is because on the CUReT texture database, the affine invariant detector cannot produce enough regions for a robust statistical characterization of the texture, while because of low resolution images in this database, the multi-fractal spectrum features used in [5] cannot be well extracted for classification. The codes of the proposed method can be downloaded at http://www4.comp.polyu.edu.hk/~cslzhang/code.htm.

5. CONCLUSIONS

In this paper, we proposed to use the sparse representation (SR) technique to learn the texton dictionary for texture image representation, and then use the SR coefficients to form the histogram for feature description. Consequently, a histogram of the SR coefficients can be constructed for texture classification. Our experimental results on the CUReT texture database validated that the proposed method can achieve the higher classification accuracy than the scheme using the *K*-means clustering method. In future work, we will investigate how to further improve the classification accuracy and reduce the size of texton dictionary.

Algorithms	**46 training images per class**		
	40 textons	**30 textons**	**20 textons**
VZ_Patch[3]	**97.11%**	**97.08%**	**96.61%**
proposed	**97.68%**	**97.54%**	**97.22%**

(a)

Algorithms	**23 training images per class**		
	40 textons	**30 textons**	**20 textons**
VZ_Patch[3]	**95.34%**	**95.06%**	**94.96%**
proposed	**95.75%**	**95.41%**	**95.24%**

(b)

Algorithms	**12 training images per class**		
	40 textons	**30 textons**	**20 textons**
VZ_Patch[3]	**91.23%**	**90.88%**	**90.47%**
proposed	**91.78%**	**91.56%**	**91.27%**

(c)

Algorithms	**6 training images per class**		
	40 textons	**30 textons**	**20 textons**
VZ_Patch[3]	**84.56%**	**84.06%**	**83.38%**
proposed	**85.36%**	**85.24%**	**84.56%**

(d)

Table1: Classification accuracies on the CUReT texture database using (a) 46; (b) 23; (c) 12 and (d) 6 training samples.

REFERENCES

[1] R.M.Haralick, K.Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Systems, Man and Cybernetics,* vol. 3, pp. 610–621, 1973.

[2] Chi-Man Pun, and Moon-chuen Lee, "Log-polar wavelet energy signatures for rotation and scale invariant texture classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1228–1333, 2004.

[3] T.Ojala, M.Pietikainen, and T.Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–987, 2004.

[4] S.Lazebnik, C.Schmid, and J.Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1265–1278, 2005.

[5] Y.Xu, H.Ji, and C.Fermuller, "A projective invariant for textures," in *Proceedings of Computer Vision and Pattern Recognition*, IEEE, 2006, pp. 1932-1939.

[6] T.Leung, and J.Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, pp. 29–44, 2001.

[7] M.Varma, and A.Zisserman, "Classifying images of materials: achieving viewpoint and illumination independence," *European Conference on Computer vision*, IEEE, 2002, vol. 3, pp. 255–271.

[8] M.Varma, and A.Zisserman, "A statistical approach to texture classification from single images," *International Journal of Computer Vision*: Special Issue on Texture Analysis and Synthesis, vol. 62, pp. 61–81, 2005.

[9] M.Varma, and A.Zisserman, "Texture classification: are filter banks are necessary?" in *Proceedings of Computer Vision and Pattern Recognition*, IEEE, 2003, vol. 2, pp. 691–698.

[10] M.Varma, and A.Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, pp. 2032–2047, 2009.

[11] D.Donoho, "For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution," *Communications on Pure & Applied Mathematics*, vol. 59, pp. 797-829, 2006.

[12] R.Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of Royal Statistical Society*, vol. 58, pp. 267–288, 1996.

[13] J.Mairal, M. Elad, and G.Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Processing*, Vol. 17, pp.53–69, 2008.

[14] John Wright, Allen Yang, Arvind Ganesh, Shankar Sastry, and Yi Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, 2009.

[15] S.S.Chen, and D.L.Donoho, and M.A.Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, pp.129–159, 2001.

[16] H.Lee, A.Battle, R.Raina, and A.Y.Ng, "Efficient sparse coding algorithms," *Advances in Neural Information Processing Systems*, 2006.

[17] K.J.Dana, B.vanGinneken, S.K.Nayar, and J.J.Koenderink, "Refelctance and texture of real world surfaces," *ACM Trans. Graphics*, vol. 18, pp. 1-34, 1999.

[18] M. Aharon, M. Elad, and A.M. Bruckstein, "The K-SVD: An Algorithm for Designing of Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. Signal Processing*, Vol.54, pp.4311–4322, 2006.