TEXTURE ANALYSIS VIA HIERARCHICAL SPATIAL-SPECTRAL CORRELATION (HSSC)

Kaitai Zhang[1], Hong-Shuo Chen[3], Ye Wang[1], Xiangyang Ji[2], and C.-C. Jay Kuo[1]

University of Southern California, Ming Heish Department of Electrical Engineering, Los Angeles, USA[1]
Tsinghua University, Department of Automation, Beijing, China[2]
National Chiao Tung University, College of Electrical and Computer Engineering, Hsinchu, Taiwan[3]

ABSTRACT

A hierarchical spatial-spectral correlation (HSSC) method is proposed for texture analysis in this work. The HSSC method first applies a multi-stage spatial-spectral transform to input texture patches, which is known as the Saak transform. Then, it conducts a correlation analysis on Saak transform coefficients to obtain texture features of high discriminant power. To demonstrate the effectiveness of the HSSC method, we conduct extensive experiments on texture classification and show that it offers very competitive results comparing with state-of-the-art methods.

Index Terms— Texture analysis, texture classification, spatial-spectral transform, spatial-spectral correlation, neural-network-inspired image transform

1. INTRODUCTION

Texture, as a combination of regularity and randomness, provides essential characteristics for surface and objection recognition. Texture analysis plays a critical role in the analysis of remote sensing photos, medical images, and many other images. A large amount of research has been conducted on texture analysis [1], classification [2, 3], segmentation [4, 5] and synthesis in the last five decades. Despite these efforts, texture analysis is still a challenging problem in image processing.

One major difficulty in texture research is lack of an effective mathematical tool for texture representation. Being inspired by recent advances in neural-network-inspired image transforms such as the Saak transform [6] and the Saab transform [7], we adopt the Saak transform as the texture representation tool since it offers several advantages. First, being different from Fourier and wavelet transforms, the Saak transform is a data-driven transform that learns transform kernels from training data samples. Second, all learnable parameters are determined by second order statistics of input images in an unsupervised manner. Neither data labels nor back-propagation is needed in transform kernel computation.

This work was supported by the Projects of International Cooperation and Exchanges NSFC with No. 61620106005.

The main contribution of this work is the proposal of a new texture feature extraction scheme from Saak transform coefficients. It is called the Hierarchical Spatial-spectral Correlation (HSSC) method. The HSSC method conducts a correlation analysis on Saak transform coefficients to obtain texture features of high discriminant power. To demonstrate the effectiveness of the HSSC method, we conduct extensive experiments on texture classification and show that it offers very competitive results in comparison with state-of-the-art methods. The proposed HSSC method is generic and flexible, and it can be integrated into various texture analysis tasks.

The rest of this work is organized as follows. Related previous work is reviewed in Sec. 2. The HSSC method is described in Sec. 3. Experimental results are given in Sec. 4. Finally, concluding remarks are drawn in Sec. 5.

2. REVIEW OF PREVIOUS WORK

Numerous methods have been proposed for texture analysis and feature representation. Many methods employ local features to represent the structural information of textures. One popular texture feature representation and extraction scheme is to convolve input textures with of a set of filter banks. Examples include: Laws filter [8], Gabor filters [9, 10], wavelet filters [1, 11] etc. The responses to these filter banks as well as their statistics can be used to obtain local texture features. Other textural features include local binary patterns (LBP) [12], co-occurrence matrices [13]. Although these feature extractors work to a certain degree, they share several common limitations.

All current filter-bank-based texture feature extractors are not efficient in adaption to different texture classes due to their pre-defined filter parameters. They are not data-driven. The effectiveness of texture features may vary significantly in multiple texture classes and even in different regions of a single texture image. Besides, some quasi-periodic textures own similar patterns in a relatively large region. To analyze textures in a local window may not be powerful enough. Local features could also be affected by randomness so that they are not as stable as one would expect. This could lead to inaccurate classification and segmentation results. In other words, it is desired to analyze textures with a flexible window size,

978-1-5386-6249-6/19/$31.00 ©2019 IEEE

and a hierarchical multi-layer texture analysis method seems to be a natural choice.

A new data-driven transform, called the Saak (Subspace approximation with augmented kernels) transform, was proposed in [6]. It has a set of orthonormal transform kernels so that its inverse transform can be performed in a straightforward manner. The Saak transform has two main ingredients: principal-component-analysis-based (PCA-based) subspace approximation and kernel augmentation. The latter is needed to resolve the sign confusion problem. Being different from the wavelet transform, the Saak transform can capture features of very fine resolution yet in a large region. The wavelet transform is a linear transform of a single layer while the Saak transform is a nonlinear transform of multiple layers.

3. PROPOSED HSSC METHOD

The HSSC method can be categorized into two types: 1) class-specific modeling, and 2) class-independent modeling. We focus on the class-specific case first, and describe the single-layer spatial-spectral correlation (SSC) and its hierarchical (i.e., multi-layer) extension in Secs. 3.1 and 3.2, respectively. Then, class-independent modeling is presented in Sec. 3.3.

3.1. Correlation of PCA Coefficients

We view texture as a two-dimensional random field that exhibits quasi-periodic patterns. We can first conduct correlation analysis on image pixels of different lags to build the correlation matrix in the spatial domain. Then, we obtain eigenvalues and eigenvectors from the correlation matrix. This process is called the principal component analysis (PCA) if we keep a subset of eigenvectors with the largest eigenvalues to span a signal subspace. The detailed procedure is elaborated below.

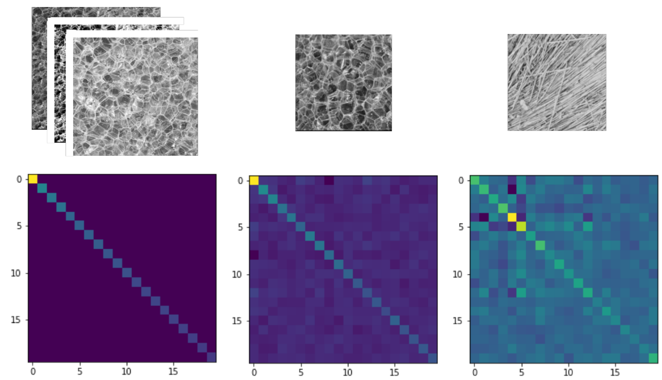
We first collect texture patches from a source texture, T, to form a collection of patches for texture T, denoted by

$$S = \{P_i | \, i \in I\}, \tag{1}$$

where i is the position index, I is the position index set, $P_i \in R^N$ is the patch located in position i, and N is the patch size. The patch size is a tunable parameter and could be adjusted according to specific application need and computational efficiency consideration. Note that we can obtain different patch samples by shifting only several pixels and, as a result, we can get a rich set of patches from only several source images.

We conduct PCA on members in S to obtain a set of K principal components (of dimension N) and use them as convolution kernels. For each patch, we get K PCA coefficients. Next, we compute the correlation matrix, R, of these K PCA coefficients. The correlation matrix of PCA coefficients is a diagonal matrix:

$$R(i,j) \equiv 0, \quad i,j = 1 \cdots K, \quad \forall i \neq j. \tag{2}$$



(a) target texture (b) matched one (c) mismatched one

Fig. 1. Illustration of correlation matrices of PCA coefficients of a target texture class, its matched and mismatched ones.

with decreasing diagonal elements:

$$R(i,i) \geq R(j,j), \forall i \geq j, \tag{3}$$

These properties are direct consequences of using PCA kernels in signal transform.

Then, we can use these diagonal elements as the texture feature. For unknown input texture X, we can determine whether it belongs to texture class T by the following two steps:

1. Collect patches from X and represent these patches using the PCA associated with texture T;

2. Compute the correlation of PCA coefficients obtained from the previous step.

If X belongs to class T, the correlation matrix should be a diagonal one and its diagonal elements should be similar to that obtained from the training textures of class T. Generally speaking, we can use the distance of two correlation matrices to measure the closeness of X and textures in class T. Examples of matched and unmatched test cases are given in Fig. 1, where correlation matrices are visualized by pixel brightness (the brighter the larger value). Clearly, if two textures are visually similar, their correlation matrices in the PCA domain of the target texture class are more similar. Since the above-mentioned analysis considers correlations of spectral components of local patches, it is called the spatial-spectral correlation analysis.

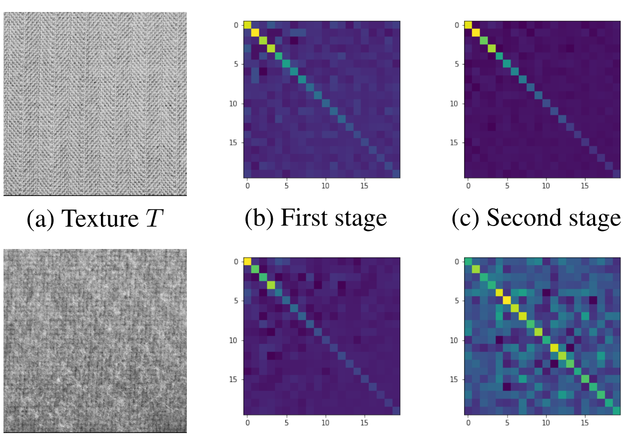
3.2. Correlation of Saak Coefficients

One limitation of the analysis using PCA coefficients is shown in Fig. 2. Textures T and T' are different while the correlation matrices of their PCA coefficients are quite similar. It is desired to conduct multi-stage PCA as an extension.

However, a straightforward cascade of multiple PCA transforms will lead to the sign confusion problem [14]. A kernel augmentation scheme was proposed in [6, 15] to address this issue.

For the example given in Fig. 2, we determine the two-stage Saak transform kernels based on target texture T, which is Herringbone Weave, and compute correlation matrices of the first-stage and the second-stage Saak transform coefficients, respectively. The visualizations of these matrices are given in Fig. 2 (b) and (c), respectively.

For input texture image X, we do a hypothesis test to check whether it belongs to texture class T or not. We apply the two-stage Saak transform to patches collected from X using kernels learned from texture class T and build correlation matrices of the first- and the second-stage Saak transform coefficients. If X happens to be Herringbone Weave as well, we will obtain correlation matrices that are similar to those in Figs. 2 (b) and (c). However, if it is a different texture class (say, Woolen cloth), we show visualizations of their first-stage and second-stage correlation matrices in Figs. 2 (e) and (f), respectively. Although its first-stage correlation matrix is similar to that of the target texture, its second-stage one is quite different. We will compare texture classification performance using multi-stage Saak transforms in Sec. 4.



(a) Texture T (b) First stage (c) Second stage

(d) Texture T' (e) First stage (f) Second stage

Fig. 2. Visualization of two-stage correlation matrices: (a) Herringbone weave (Texture T), and visualizations of its correlation matrices using (b) first-stage and (c) second-stage Saak coefficients; (d) Woolen cloth (Texture T'), and visualizations of its correlation matrices using (e) first-stage and (f) second-stage Saak coefficients based on T's Saak transform kernels.

To better understand and interpret our method, we further formalize our model into a statistical problem. PCA components or higher stage Saak kernels, which are computed using texture patches, could be understood as a set of representative pattern of texture. They are used to measure similarity between unknown texture X and target texture T. Each channel of filtered images contains a response of X against an associated pattern. Then, we compute the correlation between those responses. The diagonal terms are auto-correlations of individual channels while the non-diagonal terms are cross-correlations between different channels. Thus, our approach attempts to capture the energy distribution of representative texture patterns derived from PCA and the Saak transform.

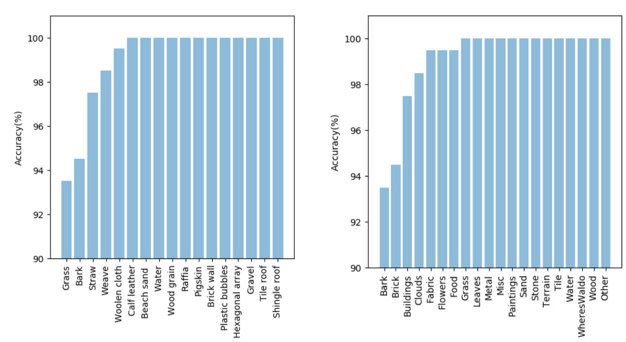
3.3. Classification with Shared Transform Kernels

The HSSC method described in Secs. 3.1 and 3.2 is suitable for hypothesis testing; namely, check whether unknown texture X belongs to target texture T. For a texture classification problem with M target textures, we need to conduct the test M times. The computational cost is higher. To reduce the computational cost, we can collect patches from all texture classes in Eq. (1) and conduct the PCA or the Saak transforms based on transform kernels learned from mixed classes.

Generally speaking, the class-independent transform reduces the computational cost at the expense of a lower classification performance. However, the classification performance can be compensated in another manner. That is, we can increase of the number of principal components at each Saak transform stage. Although the complexity will still increase a little bit, it is still lower than the solution using different transform kernels for different target textures.

4. EXPERIMENTAL RESULTS

In this section, we conduct experiments to demonstrate the effectiveness of the proposed HSSC method on several texture classification benchmark datasets.



(a)Boadatz Dataset b)VisTex Dataset

Fig. 3. Per-class accuracy results for the Brodatz and the VisTex datasets, respectively

Class-Specific Transforms. First, we report experimental results using class-specific transforms for the Brodatz and the VisTex datasets. The Brodatz texture dataset contains 155 monochrome images with standardized viewpoint and scale, which are collected from the Brodatz book [16]. The VisTex dataset[1] was initially built as an alternative to the Brodatz dataset. It does not conform to rigid frontal plane perspectives and studio lighting conditions. In this experiment, we select 17 and 20 classes from Brodatz and VisTex dataset, respectively. Images with non-texture background and significant visual variance are discarded.

The kernel sizes are set to 8×8 and 12×12 for the Brodatz and the Vistex datasets, respectively. The responses from the first quarter of kernels are selected for correlation computation. These parameters remain the same throughout the experiments.

In the training phase, we compute correlation matrices for Saak coefficients in various stages for each target texture class. In the testing phase, we pass each test texture to each class-specific transform and determine the corresponding correlation matrices. Then, we compute the Frobenius distance between the derived correlation matrices and those obtained in the training phase. The texture class that gives the smallest Frobenius distance is the predicted class. The per-class accuracy results for Brodatz and VisTex are given in Figs. 3 (a) and (b), respectively.

We also show the averaged classification accuracy in Table 1. As the stage number becomes larger, the classification becomes better. Texture features have stronger discriminant power as the stage number becomes higher. This is because it can capture features in a larger receptive field. Features from the first stage are not powerful enough. They tend to suffer from easy-to-confuse samples. As we proceed to the second and the third stages, accuracy increases gradually. This is especially obvious for the more challenging VisTex dataset.

Table 1. Averaged classification accuracies for the Brodatz and the VisTex datasets that improve as the number of stages increases. The best performance number is in bold.

Stage Number	I	II	III	IV
Brodatz	95.4	97.7	**98.7**	**98.7**
VisTex	78.8	89.1	96.2	**97.4**

Class-Independent Transforms. Next, we conduct experiments on the CUReT dataset [17] using shared (or class-independent) Saak transform kernels. The CUReT texture dataset contains 61 textures classes. Images in each class are taken from the same material but with different viewpoints and lighting conditions. Also, variations of background, shadowing and surface normals make the classification task challenging. We adopt a prepossessing step similar to that in [18]. A subset of images with a viewing angle approximately less than 30 degrees is selected in our experiment. This yields about 40 images per class. A central region of size 200×200 is cropped from each image to discard non-texture background. The dataset is randomly split into training and testing sets. We apply the 3-stage Saak transforms and compute correlation matrices based on on the 3rd-stage Saak coefficients.

[1]http://vismod.media.mit.edu/vismod/imagery/VisionTexture/

We set the number of Saak transform kernel to 8 at each stage and the kernel size is 5×5. The coefficients of the correlation matrices are selected as the feature vector, which is fed into a linear SVM classifier. The classification accuracy is shown in Table 2. As shown in the table, the proposed HSSC method offers competitive results comparing to other leading methods.

Table 2. Performance comparison of the proposed HSSC method and other state-of-the-art methods for the CUReT texture dataset.

Methods	Accuracy
Textons[2]	98.5
BIF[3]	98.6
VLAD	98.8
Histogram[19]	99.0
KCB	97.7
Ours	98.7

Many modern machine learning algorithms demand a large amount of labeled data. To examine this issue, we test the model capability with respect to the number of training patches for the CUReT dataset in Table 3. We see from this table that the classification accuracy of the proposed HSSC method becomes saturated when the number of training patches is still at a reasonable range.

Table 3. Classification accuracy as the number of training images increases with respect to the CUReT dataset.

No. of training patches	500	1000	2000	3000
Classification Accuracy	90.4	96.9	98.2	**98.7**

5. CONCLUSION

An effective hierarchical spatial-spectral correlation (HSSC) method was proposed for texture analysis and classification. It applies a multi-stage Saak transform to input texture patches and then conducts correlation analysis on Saak transform coefficients to obtain texture features of high discriminant power. Extensive experiments on texture classification with three benchmark datasets were conducted to demonstrate the effectiveness of the HSSC method. Both class-specific and class-independent transform kernels were examined.

6. REFERENCES

[1] Tianhorng Chang and C.-C. Jay Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Transactions on image processing*, vol. 2, no. 4, pp. 429–441, 1993.

[2] Eric Hayman, Barbara Caputo, Mario Fritz, and Jan-Olof Eklundh, "On the significance of real-world conditions for material classification," in *European conference on computer vision*. Springer, 2004, pp. 253–266.

[3] Michael Crosier and Lewis D Griffin, "Using basic image features for texture classification," *International journal of computer vision*, vol. 88, no. 3, pp. 447–460, 2010.

[4] Jiangye Yuan, Deliang Wang, and Anil M Cheriyadat, "Factorization-based texture segmentation," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3488–3497, 2015.

[5] Niklas Mevenkamp and Benjamin Berkels, "Variational multi-phase segmentation using high-dimensional local features," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016, pp. 1–9.

[6] C.-C. Jay Kuo and Yueru Chen, "On data-driven saak transform," *Journal of Visual Communication and Image Representation*, vol. 50, pp. 237–246, 2018.

[7] C.-C. Jay Kuo, Min Zhang, Siyang Li, Jiali Duan, and Yueru Chen, "Interpretable convolutional neural networks via feedforward design," *arXiv preprint arXiv:1810.02786*, 2018.

[8] Kenneth I Laws, "Rapid texture identification," in *Image processing for missile guidance*. International Society for Optics and Photonics, 1980, vol. 238, pp. 376–382.

[9] Anil K Jain and Farshid Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[10] Itzhak Fogel and Dov Sagi, "Gabor filters as texture discriminator," *Biological cybernetics*, vol. 61, no. 2, pp. 103–113, 1989.

[11] S Arivazhagan and L Ganesan, "Texture classification using wavelet transform," *Pattern recognition letters*, vol. 24, no. 9-10, pp. 1513–1521, 2003.

[12] Timo Ojala and Matti Pietikäinen, "Unsupervised texture segmentation using feature distributions," *Pattern recognition*, vol. 32, no. 3, pp. 477–486, 1999.

[13] Patrick C. Chen and Theodosios Pavlidis, "Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm," *Computer graphics and image processing*, vol. 10, no. 2, pp. 172–182, 1979.

[14] C.-C. Jay Kuo, "Understanding convolutional neural networks with a mathematical model," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 406–413, 2016.

[15] Yueru Chen, Zhuwei Xu, Shanshan Cai, Yujian Lang, and C.-C. Jay Kuo, "A saak transform approach to efficient, scalable and robust handwritten digits recognition," in *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 174–178.

[16] Phil Brodatz, *Textures: a photographic album for artists and designers*, Dover Pubns, 1966.

[17] Kristin J Dana, Bram Van Ginneken, Shree K Nayar, and Jan J Koenderink, "Reflectance and texture of real-world surfaces," *ACM Transactions On Graphics (TOG)*, vol. 18, no. 1, pp. 1–34, 1999.

[18] Manik Varma and Andrew Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 11, pp. 2032–2047, 2009.

[19] Robert E Broadhurst, "Statistical estimation of histogram variation for texture classification," in *Proc. Intl. Workshop on texture analysis and synthesis*, 2005, pp. 25–30.