

Texture Classification using Compressed Sensing

Li Liu, Paul Fieguth, *Member, IEEE*

Abstract—This paper presents a simple, novel, yet very powerful approach for texture classification based on compressed sensing, suitable for large texture database applications. At the feature extraction stage, a small set of random features is extracted from local image patches. The random features are embedded into a bag of words model to perform texture classification, thus learning and classification are carried out in the compressed sensing domain. The proposed unconventional random feature extraction is simple, yet by leveraging the sparse nature of texture images, our approach outperforms traditional feature extraction methods which involve careful design and complex steps. We have conducted extensive experiments on both the CURET and the Brodatz databases, comparing the proposed approach to three state-of-the-art texture classification methods: Patch, MR8, and LBP. We show that our approach leads to significant improvements in classification accuracy and reductions in feature dimensionality.

Index Terms—Texture classification, compressed sensing, textons, image patches, bag of words

1 INTRODUCTION

Texture is ubiquitous in natural images and constitutes an important visual cue for a variety of image analysis applications like image segmentation, image retrieval and shape from texture. Texture classification is a fundamental issue in computer vision and image processing, playing a significant role in a wide range of applications that includes medical image analysis, remote sensing, object recognition, content-based image retrieval and many more. Due to its importance, texture classification has been an active research topic over several decades, dating back at least to Julesz's initial research in 1962 [1] [2].

The design of a texture classification system essentially involves two major steps: 1) feature extraction, and 2) classification. The literature on texture feature extraction is substantial, with extensive surveys [3]–[7]. Well known representative methods include Grey Level Cooccurrence Histograms [8], Markov Random Fields [9] [10], Gray Level Aura Histograms [11] [12], Local Binary Patterns [13] [14], Gabor filter banks [15] [16] [17], Wavelets [18], and Fractal Models [19]. All of these choose a limited subset of texture features from local image patches, where the number of features is usually less than the dimensionality of the source image patch. However, as Randen and Husøy [6] concluded in their recent excellent comparative study involving dozens of different filtering methods: “No single approach did perform best or very close to the

best for all images, thus no single approach may be selected as the clear winner of this study.”

By extracting features from a local patch, most feature extraction methods focus on local texture information, characterized by the gray level patterns surrounding a given pixel; however texture is also characterized by its global appearance, representing the repetition of and the relationship among local patterns. Recently, a “Bag of Words” (BoW) model, borrowed from the text literature, opens up new prospects for texture classification [20]–[25]. The BoW model encodes both the local texture information, by using features from local patches to form textons, and the global texture appearance, by statistically computing an orderless histogram representing the frequency of the repetition of the textons.

There are two main ways to construct the textons: 1) detecting a sparse set of points in a given image using Local Interest Point (LIP) detectors and then using local descriptors to extract features from a local patch centered at the LIPs [24] [25], 2) densely extracting local features pixel by pixel over the input image. The sparse approach largely depends on the texture images, some of which might not produce enough regions for a robust representation of the texture. As a result, the dense approach is more common and widely studied. Among the most popular dense descriptors are the use of large support filter banks to extract texture features at multiple scales and orientations [20] [21] [22]. However, more recently, in [23] the authors challenge the dominant role that filter banks have been playing in texture classification, claiming that classification based on textons directly learned from the raw image patches outperforms textons based on filter bank responses.

The key parameter in patch-based classification is

- L. Liu is with the School of Electronic Science and Engineering, National University of Defense Technology, Changsha, Hunan, China E-mail: feiyunlyi@hotmail.com
- P. Fieguth is with the Department of System Design Engineering, University of Waterloo, Waterloo, Ontario, Canada. E-mail:pfieguth@uwaterloo.ca.

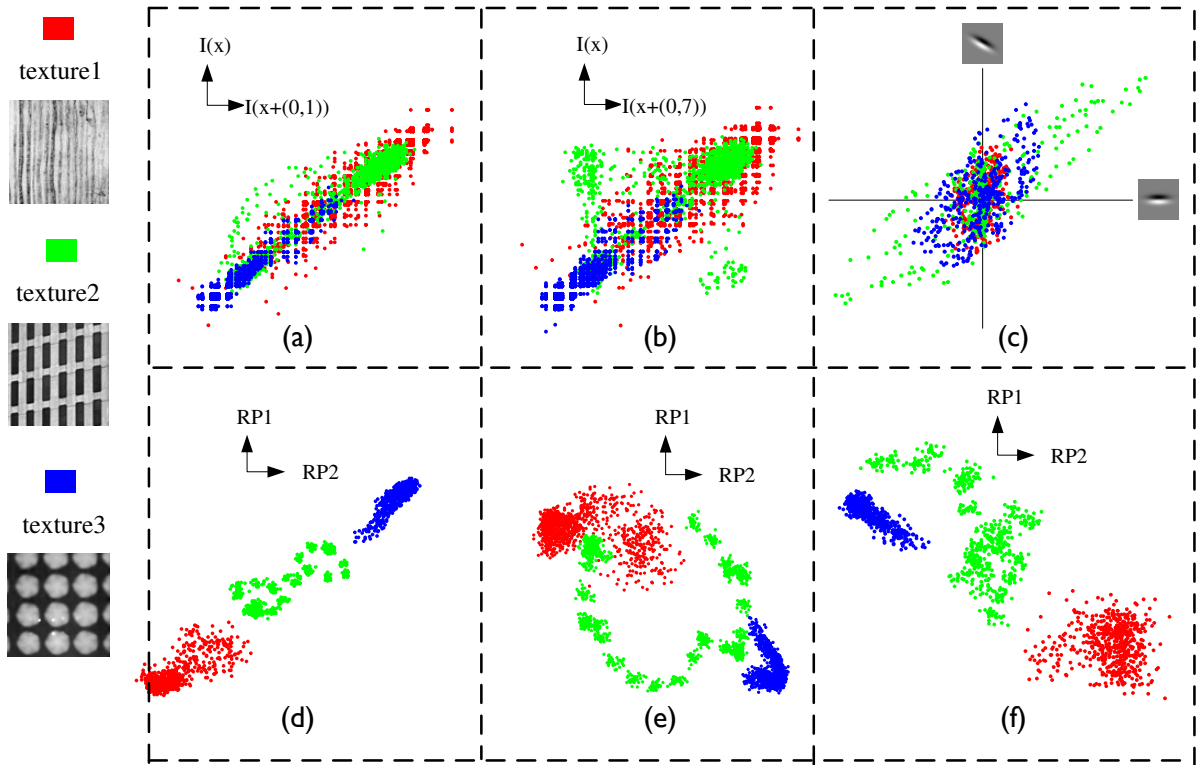


Fig. 1. **Compressed sensing measurements of local patches form good shape clusters and can distinguish texture classes.** Three textures are shown at left from the Brodatz database. Compare the spatial distribution and separability of (a) (b) raw pixel values, (c) two linear filter responses (computed from a 49×49 support region), and CS random projections (RP1, RP2) extracted from patches of size (d) 9×9 , (e) 15×15 , (f) 25×25 .

the size of the patch. Small patch sizes cannot capture large-scale structures that may be the dominant features of some textures, are not very robust against local changes in textures, and are highly sensitive to noise and missing pixel values caused by illumination variations. However, the disadvantage of the patch representation is the quadratic increase in the dimension of the patch space with the size of patch, with the high dimensionality posing two challenges to the clustering algorithms used to learn textures. First, the presence of irrelevant and noisy features can mislead the clustering algorithm. Second, in high dimensions data may be very sparse (the curse of dimensionality), making it difficult to represent the structure in the data.

Therefore, it is natural to ask whether high dimensional patch vectors can be projected into a lower dimensional subspace without suffering great information loss. There are many potential benefits of a low dimensional feature space: reduced storage requirements, reduced computational complexity, and defying the curse of dimensionality to improve classification performance. A small salient feature set would simplify both the pattern representation and the subsequent classifiers, however frequently-used

dimensionality reduction techniques result in a loss of information when projecting. This brings us into the realm of compressive sensing.

The compressed sensing (CS) approach [26] [27] [28] [29], which has been the motivation for this research, is appealing because of its surprising result that high-dimensional sparse data can be accurately reconstructed from just a few nonadaptive linear random projections. When applying CS to texture classification, the key question is therefore how much information about high-dimensional sparse texture signals in local image patches can be preserved by random projections, and whether this leads to any advantages in classification.

The abilities of CS for perfect signal reconstruction have been proved [27] [28], however the application of CS to texture classification has received only minimal treatment to date. Limited work has been reported [30] [31] [32] [33], exploiting the specific structure of sparse coding for texture patches, depending on the recovery process and careful design of the sparsifying redundant dictionary. In contrast, our work performs classification in the *compressed domain*, not relying on any reconstruction. In this paper we present a comprehensive series of experiments to illustrate the

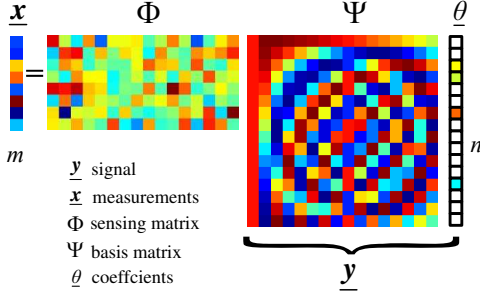


Fig. 2. Compressed Sensing measurement process

benefits of this novel theory for texture classification.

The proposed method is computationally simple, yet very powerful. Instead of performing texture classification in the original high-dimensional patch space or making efforts to figure out a suitable feature extraction method, by using random projections of local patches, we perform texture classification in a much lower-dimensional compressed patch space. The theory of CS implies that the precise choice of the number of features is no longer critical: a small number of random features, more than some threshold, contains enough information to preserve the underlying local texture structure and hence to correctly classify any test image. Fig. 1 is a preliminary exploration of this claim, contrasting the distribution of raw pixels, filter responses and random CS features.

The rest of this paper is organized as follows. Section 2 reviews the CS background and some related work. In Section 3, we first discuss the theoretical reasons for the proposed approach, then present the details of the proposed features and the texture classification framework, and at last provide analysis of the benefits and advantages of the proposed approach. In Section 4, we verify the proposed method with extensive experiments on popular texture datasets and present comparisons with three state-of-the-art methods: the Patch method, the MR8 filter bank, and the LBP method.

2 BACKGROUND AND RELATED WORK

The theory of compressed sensing has recently been brought to the forefront by the work of Candès and Tao [27] and Donoho [28], who have shown the advantages of random projections for capturing information about sparse or compressible signals. The premise of CS is that a small number of nonadaptive linear measurements of a compressible signal or image contain enough information for near perfect reconstruction and processing. This emerging theory has generated enormous amounts of research with applications to high-dimensional geometry [34] [35], image reconstruction [31] [37], and machine learning [36] [38] etc.

CS exploits the fact that many signal classes have a low-dimensional structure compared to the

high-dimensional ambient space. Compressed sensing refers to the idea that, for certain types of signals, a small number of nonadaptive measurements in the form of randomized projections can capture most of the salient information in a signal and can approximate the signal well. The beauty of the CS theory is that if a signal may be sparsely represented in some basis, it may be perfectly recovered based on a relatively small set of random projections:

- ◆ The key assumption in CS is that of sparsity or compressibility. Let $\underline{y} \in \mathbb{R}^{n \times 1}$ be an unknown signal of length n and $\Psi = [\psi_1 \dots \psi_n]$ an orthonormal basis, where $\psi_i \in \mathbb{R}^{n \times 1}$, such that

$$\underline{y} = \sum_{i=1}^n \theta_i \psi_i = \Psi \underline{\theta} \quad (1)$$

where $\underline{\theta} = [\theta_1 \dots \theta_n]^T$ denotes the vector of coefficients that represents \underline{y} in the basis Ψ , as illustrated in Fig. 2. Signal \underline{y} is said to be *sparse* if most of the coefficients in $\underline{\theta}$ are zero or can be discarded without much loss of information. Sparse signals are an idealization that we do not encounter in practice, but real signals are usually *compressible*, which means that the entries in $\underline{\theta}$ decay rapidly when sorted by magnitude in decreasing order.

- ◆ Let Φ be an $m \times n$ sampling matrix, with $m \ll n$, such that

$$\underline{x} = \Phi \underline{y} = \Phi \Psi \underline{\theta} \quad (2)$$

where \underline{x} is an $m \times 1$ vector of linear measurements. The sampling matrix Φ must allow the reconstruction of length- n signal \underline{y} from length- m measurement vector \underline{x} . Since the transformation from \underline{y} to \underline{x} is a *dimensionality reduction*, in general there is an information loss, however the measurement matrix Φ can be shown to preserve the information in sparse and compressible signals if it satisfies the so-called *restricted isometry property* (RIP) [26]. Intriguingly, a large class of random matrices have the RIP with high probability [26] [27] [28]. The measurement process (illustrated in Fig. 2) is non-adaptive in that Φ does not depend in any way on the signal \underline{y} .

- ◆ The signal reconstruction algorithm must take the m measurements in \underline{x} , the random measurement matrix Φ , and the basis Ψ to reconstruct $\underline{\theta}$. A large number of approaches have been proposed in the literature to solve the reconstruction problem [39], however the reconstruction algorithms tend to be computationally burdensome.

Research in CS has focused primarily on reducing the number of measurements m , increasing robustness, and reducing the computational complexity of the recovery algorithm [39]. The success of CS for signal reconstruction motivates the study of its potential for

signal classification [30] [38] [40]. One of the most successful applications of CS theory in computer vision and pattern recognition has been the SRC algorithm for face recognition [38]. The SRC algorithm uses the whole set of training samples as the basis dictionary and assumes that all of the samples from a class lie on a linear subspace, such that the recognition problem is cast as one of discriminatively finding a sparse representation of the test image as a linear combination of training images. It is important to note, however, that the SRC algorithm is based on global features, whereas texture classification almost certainly depends on the relationship between a pixel and its neighborhood. Secondly, SRC is reconstruction based, *explicitly* reconstructing the sparse θ , a computationally intensive step which we avoid.

3 TEXTURE CLASSIFICATION USING COMPRESSED SENSING

3.1 Texture images are sparse

The premise underlying CS is one of signal sparsity or compressibility, and the compressibility of textures is certainly well established. Certainly most natural images are compressible, as extensive experience with the wavelet transform has demonstrated [50]. Textures, being roughly stationary/periodic, are all the more sparse. Furthermore from the large literature on texture classification on the basis of feature extraction from small image patches, the degrees of freedom underlying a texture are quite few in number.

For example in [41], the author first uses a filter bank to reduce the patch space and then further reduces the dimension of texture filter responses by projecting filter marginals onto low-dimensional manifolds by a Locally Linear Embedding algorithm [42], showing that classification accuracy can be increased by projecting onto a manifold of some suitable dimension. Cula and Dana [21] first learn the histogram of textons for a texture and then project all of the models into a low-dimensional space using principle components analysis. A manifold was fitted to these projected points, and then reduced by systematically discarding those points which least affected the shape of the manifold.

3.2 Dimensionality Reduction and Information Preservation

In this paper we intend to use linear projections to embed a local patch $\underline{p} \in \mathbb{R}^{n \times 1}$ into a lower-dimensional space $\underline{x} \in \mathbb{R}^{m \times 1}$:

$$\underline{x} = \Phi \underline{p} \quad (3)$$

ideally where $m \ll n$. Clearly $\Phi \in \mathbb{R}^{m \times n}$, $m < n$ loses information in general, since Φ has a null space, implying the indistinguishability between \underline{p} and $\underline{p} + \underline{z}$, for $\underline{z} \in \mathcal{N}(\Phi)$. The challenge in identifying an effective

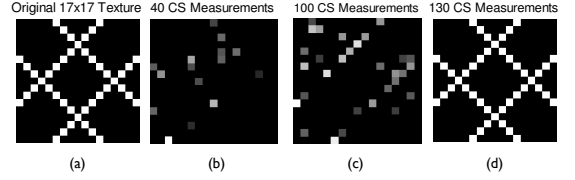


Fig. 3. Reconstruction of ideal sparse texture signals from CS measurements.

feature extractor Φ is to have the null-space of Φ orthogonal to the low-dimensional subspace of sparse signal \underline{p} .

Ideally, we wish to ensure that Φ is information-preserving, by which we mean that Φ provides a stable embedding that approximately preserves distances between all pairs of signals. That is, for any two patches, \underline{p}_1 and \underline{p}_2 , the distance between them is approximately preserved:

$$1 - \epsilon \leq \frac{\|\Phi(\underline{p}_1 - \underline{p}_2)\|_2}{\|\underline{p}_1 - \underline{p}_2\|_2} \leq 1 + \epsilon \quad (4)$$

for small $\epsilon > 0$. One of the key results in [26] from CS theory is the Restricted Isometry Property, which states that (4) is indeed satisfied by certain random matrices, including a Gaussian random matrix Φ . It is on this basis that we propose to use the emerging theory of compressed sensing to rethink texture classification.

Example 1: CS and Texture

A simple example, illustrated in Fig. 3, reconstructs a texture patch based on CS measurements. The reconstruction results, from different numbers of measurements, are shown in panels (b), (c) and (d). The reconstruction algorithm CoSaMP [39] is used here. For this setting, where the texture signal is highly sparse, with sufficiently large number of random measurements the original texture is perfectly reconstructed.

Example 2: CS and Classification

Real data are noisy, so (3) should be modified to explicitly account for noise:

$$\underline{x} = \Phi \underline{p} + \underline{v} \quad (5)$$

where $\underline{v} \in \mathbb{R}^{m \times 1}$ is a noise term, independent of \underline{p} . We wish to classify \underline{p} based on the noise-corrupted compressed measurements \underline{x} , using a single nearest neighbor classifier with a Euclidean distance measure.

Suppose we have a set of 100 sinusoids, as plotted in Fig. 4 (a):

$$\{\underline{p}_k(t)\}_{k=1}^{100} = \{\cos(\omega_k t)\}_{k=1}^{100} \quad (6)$$

We collect m random measurements, and we evaluate the probability of classification accuracy by averaging over 100,000 trials where, for each trial, independent realizations of compressed sensing measurements and

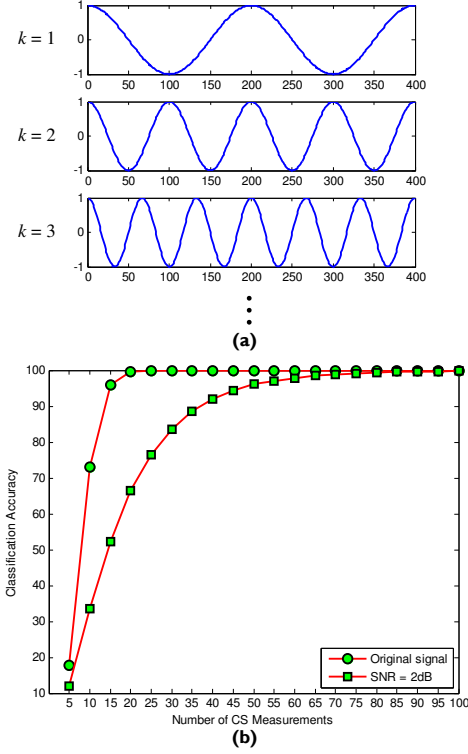


Fig. 4. Signal classification based on CS: (a) A set of 100 synthetic similar periodic signals, each of length $n = 400$. (b) Classification accuracy as a function of the number of CS measurements for both noisy and noise-free cases.

noise were generated. The results are shown in Fig. 4 (b). We can see that CS measurements can effectively be used in classification, both in noisy and noise-free cases. Note that classification was performed directly in the compressed domain, and without any explicit sparse reconstruction.

3.3 CS measurements vs. Original Patches

As mentioned before, we wish to preserve both local texture information, contained in a local image patch, and global texture appearance, representing the repetition of the local textures and the relationship among them. It has been shown that a texton-based approach is an effective local-global representation [20] [23].

The textons are trained by adaptively partitioning the feature space into clusters using K-means. For an input data set $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_{|\mathcal{X}|}\}$, $\mathbf{x}_i \in \mathbb{R}^{d \times 1}$, and an output texton set $\mathcal{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_K\}$, $\mathbf{w}_i \in \mathbb{R}^{d \times 1}$, the quality of a clustering solution is measured by the *average quantization error* [53] [54], denoted as $Q(\mathcal{X}, \mathcal{W})$:

$$Q(\mathcal{X}, \mathcal{W}) = \frac{1}{|\mathcal{X}|} \sum_{j=1}^{|\mathcal{X}|} \min_{1 \leq k \leq K} \|\mathbf{x}_j - \mathbf{w}_k\|_2^2 \quad (7)$$

measuring the average squared distance from each point to the centroid of the cluster where it belongs.

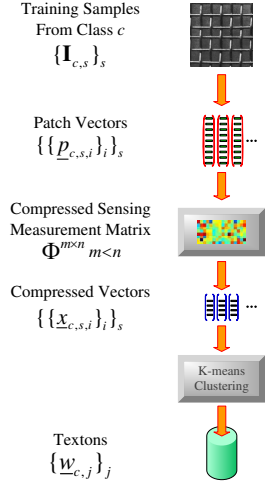


Fig. 5. Texton dictionary learning in the compressed patch domain.

However, $Q(\mathcal{X}, \mathcal{W})$ goes as $K^{-2/d}$ for large K [53], a problem when d is large, since K is then required to be extremely large to obtain satisfactory cluster centers, with computational and storage complexity consequences.

On the other hand, Varma and Zisserman [23] have shown that image patches contain sufficient information for texture classification, arguing that the inherent loss of information in the dimensionality reduction of feature extraction leads to inferior classification performance.

CS addresses the dilemma between these two perspectives very neatly. The high-dimensional texture patch space has an intrinsic dimensionality that is much lower, therefore CS is able to perform texture feature extraction without information loss (Example 1), and classification is possible in the CS compressed domain (Example 2). On the basis of the above analysis, we claim that the CS and BoW approaches are complementary, and will together lead to superior performance for texture classification.

3.4 Proposed Approach

We have C distinct texture classes, with each class having S samples. Let the samples of class c be represented by an ensemble $\{\mathbf{I}_{c,s}\}_{s=1}^S$ and let $\mathcal{D} = \{\{\mathbf{I}_{c,s}\}_{s=1}^S\}_{c=1}^C$ denote the whole texture dataset. A set of $\sqrt{n} \times \sqrt{n}$ image patches $P = \{\mathbf{p}_{c,s,i}\}_i$ is extracted from image $\mathbf{I}_{c,s}$. We shall assume hereafter that such an extraction does not include pixels on the image boundary.

Our proposed classifier is identical to the Patch method [23] except that instead of using \mathbf{p} , the compressed sensing measurements $\mathbf{x} = \Phi \mathbf{p}$ derived from \mathbf{p} are used as features. In this paper, we choose Φ to be a Gaussian random matrix, i.e., the entries of Φ are independent zero-mean, unit-variance normal.

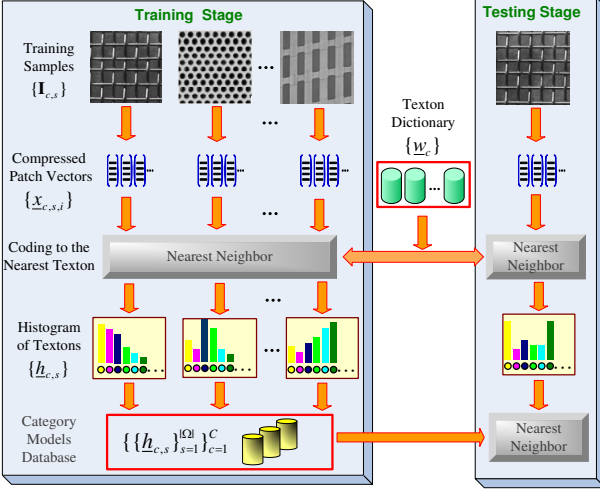


Fig. 6. The architecture of training and testing stages.

The *compressed domain*

$$\mathcal{X} = \{\underline{x} = \Phi \underline{p} \mid \underline{p} \in \mathcal{P}\} \quad (8)$$

is thus a compressed representation of *patch domain*

$$\mathcal{P} = \{\underline{p} \mid \underline{p} \in \mathbb{R}^{n \times 1}\} \quad (9)$$

Our texture classification system is illustrated in Fig. 5 and Fig. 6, consisting of the following stages:

- 1) Compressed texton dictionary learning stage, illustrated in 5, in which a universal compressed texton dictionary \mathcal{W} is learned directly in the compressed domain \mathcal{X} , as opposed to from the patch domain \mathcal{P} .
- 2) Histogram of textons learning stage, illustrated in Fig. 6 (left), a histogram $\underline{h}_{c,s}$ of compressed textons is learnt for each particular training sample $I_{c,s}$ by labeling each of its pixels with the closest texton in \mathcal{W} . Each texture class then is represented by a set of models $\mathcal{H}_c = \{\underline{h}_{c,s}\}_s$ corresponding to the training samples of that class.
- 3) The classification stage, shown in Fig. 6 (right), where the process to compute the normalized histogram of compressed textons \underline{h}_{new} for a novel image is the same as computing the model for each training sample. The calculated model \underline{h}_{new} is classified into one of the known classes using nearest neighbor classifier, where the distance between two histograms is measured using the χ^2 statistic:

$$\chi^2(\underline{h}_1, \underline{h}_2) = \frac{1}{2} \sum_{k=1}^{CK} \frac{[\underline{h}_1(k) - \underline{h}_2(k)]^2}{\underline{h}_1(k) + \underline{h}_2(k)} \quad (10)$$

4 EXPERIMENTAL EVALUATION

4.1 Methods in Comparison Study

Our specific experimental goal is to compare the proposed CS approach with the following four state-of-the-art methods:

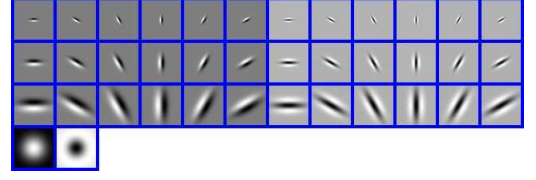


Fig. 7. The original filter bank for obtaining the MR8 filter responses: edge and bar filters at 3 scales and 6 orientations, plus a Gaussian and Laplacian of Gaussian.

Patch [23]: Each local patch of size $\sqrt{n} \times \sqrt{n}$ is reordered into an n -dimensional patch vector. Both training and testing are performed in the patch domain.

Patch-MRF [23] [43]: A texture image is represented using a two-dimensional histogram: one dimension the quantized bins of the patch's center pixel, the other dimension the learned textons from the patch vector with the center pixel excluded. The number of bins for the center pixel used in [23] is as large as 200 and the size of texton dictionary is $61 \times 40 = 2440$, resulting in an extremely high dimensionality of $2440 \times 200 = 488,000$.

MR8 [22] [23]: The MR8 consists of 8 filter responses derived from the original responses of 38 filters (see Fig. 7). A complicated anisotropic Gaussian filtering method [44] was used to calculate the MR8 responses.

LBP [14] [45]: The rotationally invariant, uniform LBP texton dictionary at different scales, $\text{LBP}_{8,1}^{\text{riu2}}$, $\text{LBP}_{8,1+16,2}^{\text{riu2}}$, $\text{LBP}_{8,1+16,2+24,3}^{\text{riu2}}$, $\text{LBP}_{8,1+16,2+24,3+24,4}^{\text{riu2}}$, $\text{LBP}_{8,1+16,2+24,3+24,4+24,5}^{\text{riu2}}$ advocated in [46] [47], will be used for comparison with the proposed approach. For simplicity, in the remainder of this paper these LBP textons are denoted as 1-scale, ..., 5-scale respectively.

4.2 Texture Datasets and Experimental Setup

For our experimental evaluation we have used four texture datasets, summarized in Table 1, derived from the two most commonly used texture sources: the Brodatz album [48] and the CURET database [49]. The Brodatz database is perhaps the best known benchmark for evaluating texture classification algorithms. Performing classification on the entire database is challenging due to the relatively large number of texture classes, the small number of examples for each class, and the lack of intra-class variation.

The **Brodatz small dataset** \mathcal{D}^b (**24 classes**) was chosen to allow a direct comparison with the state-of-the-art results from [45]. There are 24 homogeneous texture images selected from the Brodatz album (see Fig. 8), each of which was partitioned into 25 nonoverlapping sub-images of size of 128×128 pixels, of which 13 samples for training and the remaining 12 for testing.

TABLE 1
Summary of texture datasets used in the experiments

Texture Dataset	Rotation In plane	Rotation Off plane	Controlled Illumination	Texture Classes	Sample Size	Samples per Class	Training Samples per Class	Testing Samples per Class
\mathcal{D}^b				24	128×128	25	13	12
\mathcal{D}^B				90	128×128	25	13	12
\mathcal{D}^c			✓	61	106×106	9	5	4
\mathcal{D}^C	✓	✓	✓	61	200×200	92	46	46

The **Brodatz large dataset \mathcal{D}^B (90 classes)** is a very challenging platform for classification performance analysis due to the impressive diversity and perceptual similarity of some textures, some of which essentially belong to the same class but at different scales (D1 and D6, D25 and D26), while others are so inhomogeneous that a human observer would arguably be unable to group their samples correctly (D43, D44 and D97). Based on these considerations, we selected 90 texture classes from the Brodatz album by visual inspection, excluding textures D13, D14, D16, D21, D22, D25, D30, D32, D35, D36, D38, D43-45, D55, D58, D59, D61, D79, D91, D96 and D97. The partitioning of images in \mathcal{D}^B is the same as in \mathcal{D}^b .

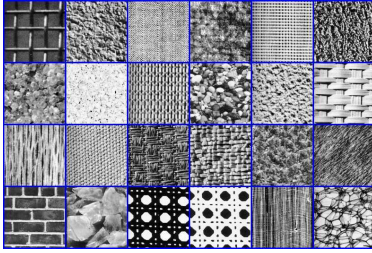


Fig. 8. Brodatz Small: The 24 texture used in [45] from the Brodatz database

The Brodatz database has been criticized because of the lack of intra-class variation, leading to the development of the CURET database [49] which has now become a benchmark and is widely used to assess classification performance.

For the **CURET large dataset \mathcal{D}^C (61 classes)**, we use the same subset of images as Varma and Zisserman [22] [23] [43], containing 61 texture classes shown in Fig. 9 with 92 images for each class, resulting in a total of $61 \times 92 = 5612$ images. These images are captured under different illuminations with seven different viewing directions, a few of which are plotted in Fig. 10. In the experiments on this dataset, half of the samples are chosen for training and the remaining half for testing.

The **CURET small dataset \mathcal{D}^c (61 classes)** preserves all texture classes of \mathcal{D}^C , however each kind of textures is represented by only a single texture image taken from the original CURET database [49], where all of the textures have the same illumination and imaging conditions. We partitioned each 320×320 texture image into nine 106×106 nonoverlapping sub-

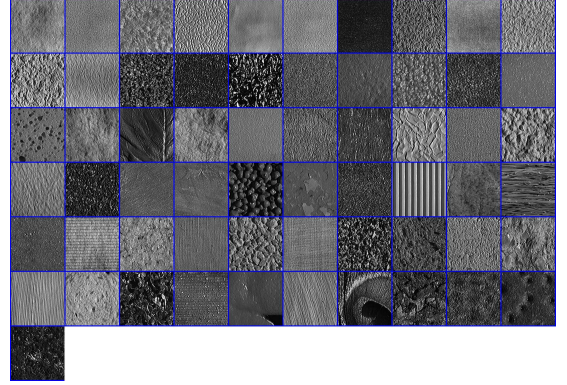


Fig. 9. CURET: The 61 textures in the CURET database [49].

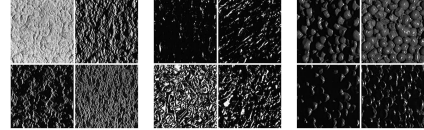


Fig. 10. Image examples from three different texture classes of CURET textures under different illuminations and viewpoints.

images, consistent with [45]. Out of the nine samples in each class, five samples are used for training, the other four for testing.

Finally, in terms of the extracted CS vector, we consider three kinds of normalizations:

- 1) Weber's law [23]:

$$\underline{x} \leftarrow \underline{x} \left[\frac{\log(1 + \|\underline{x}\|_2/0.03)}{\|\underline{x}\|_2} \right] \quad (11)$$

- 2) Unit norm:

$$\underline{x} \leftarrow \frac{\underline{x}}{\|\underline{x}\|_2} \quad (12)$$

- 3) No normalization.

4.3 Experimental Tests

Variability Analysis

Because CS performs random feature extraction, clearly one of the first questions is the extent to which this randomness is manifested in classifier variability. There are three sources of variability present:

- 1) Variation in learned textons from K -means;

TABLE 2

Classifier variability: Standard deviations are reported from 20 runs on \mathcal{D}^C using 10 textons per class, a patch size of 11×11 , Weber’s law normalization.

Varied Training/Tesing	Varied CS Matrix	Varied K-means	CS Dimensionality	
			40	6
		✓	0.09%	0.16%
	✓	✓	0.15%	0.33%
✓	✓	✓	0.20%	0.69%

- 2) Variation in the random CS matrix;
- 3) Variation in training/testing data.

The contribution of all three variations is presented in Table 2. Although there is clearly variability present due to the randomness of the CS matrix, it is a modest fraction of the total variability, and therefore in no way compromises the CS method as a classifier.

CS Parameter Choices

There are three key parameters in the CS classifier:

- 1) The number of textons K per class;
- 2) The patch size n ;
- 3) The CS dimensionality m ($m \leq n$).

We first examine the effect of the choice of m on the classification performance, shown in Fig. 12. We can see from the results that the classification accuracy increases rapidly, is level for a wide range of m , and ultimately decreasing for sufficiently large m . The decreased accuracy at large m is almost certainly the increased difficulty of clustering in high dimensions, consistent with our claim arguing against the high dimensionality of the Patch method.

Fig. 11 plots the classification accuracy for a variety of patch sizes n where, motivated by the results of Fig. 12 and by our underlying premise of dimensionality reduction, only modest values of m are tested. The results are consistent with the preliminary test in Fig. 12: for each value of n , the performance improves rapidly for small m , then leveling off for $m \approx n/3$. For dataset \mathcal{D}^c (Fig. 11 (a)), the performance decreases with patch size n , due to the small size (limited training samples) of \mathcal{D}^c , insufficient to train the classifier on large patches. In contrast, for \mathcal{D}^C (Fig. 11 (b)) the larger training set allows for sufficient classifier learning.

Finally consider the choice of K , the number of textons per class. Because of the dimensionality reduction of the CS method, it is computationally feasible to consider greater numbers of textons. Since a set of textons can be thought of as adaptively partitioning the compressed patch space into bins, K should be sufficiently large to allow the partitioning to meaningfully represent the space. Fig. 14 demonstrates the impact of K on the classification accuracy for a patch size of 11×11 and a CS dimension of 40. The figure shows performance increasing with K , though for $K > 30$ the benefits are limited. In our comprehensive

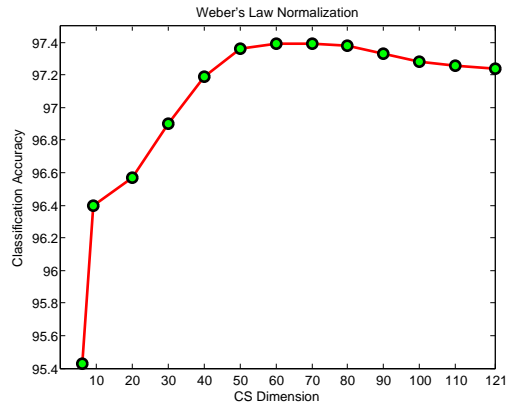


Fig. 12. Classification accuracy as a function of CS dimensionality on an 11×11 patch for dataset \mathcal{D}^C with $K = 10$.

tests, reported in the next section, we will present results for both $K = 10$ and $K = 40$.

4.4 Comparative Evaluation

In this section, we compare the proposed approach specifically to the current state-of-the-art Patch method [22] [23] on the CURET database. To make the comparison as meaningful as possible, we use the same experimental settings as Varma and Zisserman [23]. Moreover, we further show results comparing our method with the LBP method and the MR8 filter bank, described in Section 4.1.

In their comprehensive study, Varma and Zisserman [22] presented six filter banks for texture classification on \mathcal{D}^C . They concluded that the rotationally invariant, multi-scale, Maximum Response MR8 filter bank yielded better results than any other filter bank. However, in their more recent study [23], they challenged the dominant role that filter banks have come to play in the texture classification field and claim that their Patch method outperforms even the MR8 filter bank.

Fig. 13 and Table 3 present a comparison of the CS classifier, the Patch classifier, the MR8 filter bank and LBP. Patch-VZ and MR8-VZ are results reported by Varma and Zisserman [23], all other results are computed by us, with the results averaged over tens of random partitions of training and testing sets.

The CS method significantly outperforms all other methods, a clear indication that the CS matrix preserves the salient information contained in the local patch (as predicted by CS theory) and that performing classification in the compressed patch space is not a disadvantage. In contrast to the Patch method, not only does CS offer higher classification accuracy, but also at a much lower-dimensional feature space, reducing storage requirements and computation time.

In terms of the feature vector normalizations, Fig. 13 presents results of all three normalizations. Similarly,

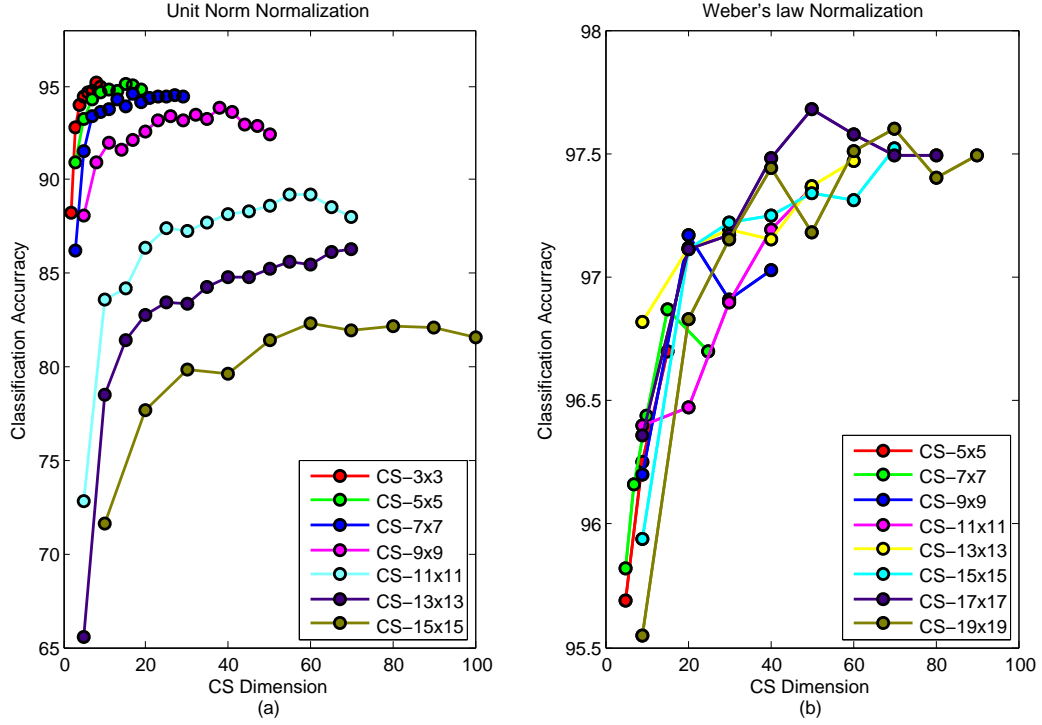


Fig. 11. Classification results as a function of patch size and CS dimensionality on dataset (a) \mathcal{D}^c and (b) \mathcal{D}^C

TABLE 3

Weber's law normalization on \mathcal{D}^C : The mean and standard deviation of the classification accuracy as a function of patch size. Results are reported over tens of runs. The bracketed values denote the number of textons used per class. Clearly, the performance of the CS method is better than the Patch method, but at much lower feature dimensionality.

	3 × 3 (%)	5 × 5 (%)	7 × 7 (%)	9 × 9 (%)	11 × 11 (%)	13 × 13 (%)	15 × 15 (%)	17 × 17 (%)	19 × 19 (%)
Dim	9	15	25	30	40	50	60	80	90
CS (10)	95.07 ± 0.24	96.70 ± 0.18	96.80 ± 0.14	96.91 ± 0.67	97.19 ± 0.20	97.37 ± 0.28	97.41 ± 0.21	97.49 ± 0.21	97.40 ± 0.34
Dim	5	10	15	30	40	50	60	80	90
CS (40)	96.37 ± 0.54	97.00 ± 0.48	97.49 ± 0.49	97.95 ± 0.23	98.19 ± 0.23	98.07 ± 0.28	98.24 ± 0.21	98.37 ± 0.27	98.43 ± 0.22
Dim	9	25	49	81	121	169	225	289	361
Patch (10)	95.16 ± 0.34	96.33 ± 0.36	96.53 ± 0.23	96.74 ± 0.24	96.84 ± 0.32	96.87 ± 0.30	97.18 ± 0.33	97.04 ± 0.22	97.30 ± 0.33
Patch-VZ (10)	95.33	95.62	96.19	96.38 ± 0.35	96.58 ± 0.35	96.63 ± 0.35	96.89 ± 0.33	97.11 ± 0.32	97.17 ± 0.32
MR8-VZ (10)	N/A	N/A	N/A	95.06 ± 0.41	95.57 ± 0.38	95.92 ± 0.37	96.16 ± 0.37	96.30 ± 0.37	96.37 ± 0.36
Scale	1-scale	2-scale	3-scale	4-scale	5-scale	N/A	N/A	N/A	N/A
LBP	81.46 ± 0.97	91.65 ± 0.70	94.06 ± 1.10	94.61 ± 0.66	95.72 ± 0.68	N/A	N/A	N/A	N/A

TABLE 4

As in Table 3, but with no normalization.

	3 × 3 (%)	5 × 5 (%)	7 × 7 (%)	9 × 9 (%)	11 × 11 (%)	13 × 13 (%)	15 × 15 (%)	17 × 17 (%)	19 × 19 (%)
Dim	5	13	15	20	40	50	60	80	90
CS (10)	95.00 ± 0.45	96.72 ± 0.65	96.90 ± 0.26	97.08 ± 0.20	97.17 ± 0.22	97.23 ± 0.20	97.41 ± 0.28	97.47 ± 0.27	97.66 ± 0.20
Dim	9	10	17	30	40	50	60	70	80
CS (40)	95.57 ± 0.83	96.91 ± 0.78	97.18 ± 0.40	97.88 ± 0.27	98.17 ± 0.32	98.00 ± 0.33	98.20 ± 0.31	98.25 ± 0.31	98.29 ± 0.45
Dim	9	25	49	81	121	169	225	289	361
Patch (10)	95.35 ± 0.35	96.63 ± 0.29	96.87 ± 0.35	96.96 ± 0.51	96.78 ± 0.37	N/A	N/A	N/A	N/A
Scale	1-scale	2-scale	3-scale	4-scale	5-scale	N/A	N/A	N/A	N/A
LBP	81.46 ± 0.97	91.65 ± 0.70	94.06 ± 1.10	94.61 ± 0.66	95.72 ± 0.68	N/A	N/A	N/A	N/A

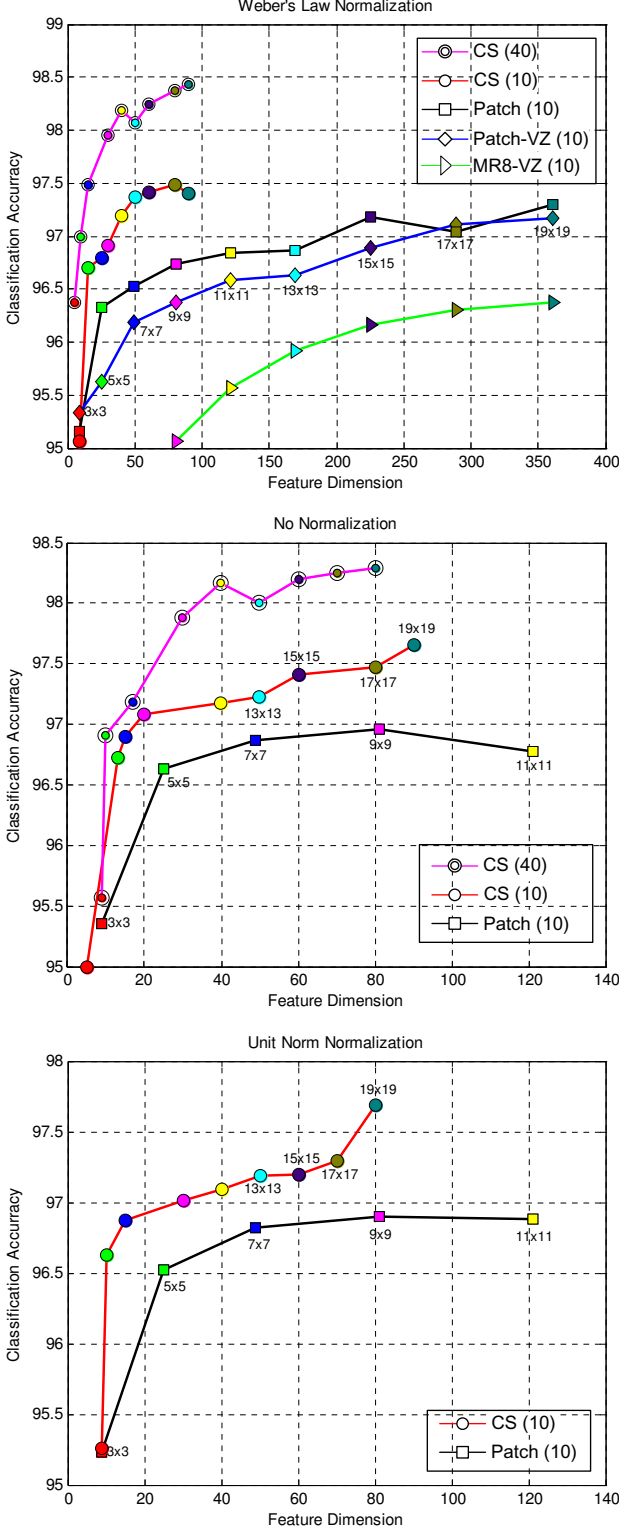


Fig. 13. Classification results on dataset \mathcal{D}^C as a function of feature dimensionality. The bracketed values denote the number of textons K per class. “Patch-VZ” and “MR8-VZ” results are quoted directly from the paper of Varma and Zisserman [23]. Classification rates obtained based on the same patch size are shown in the same color.

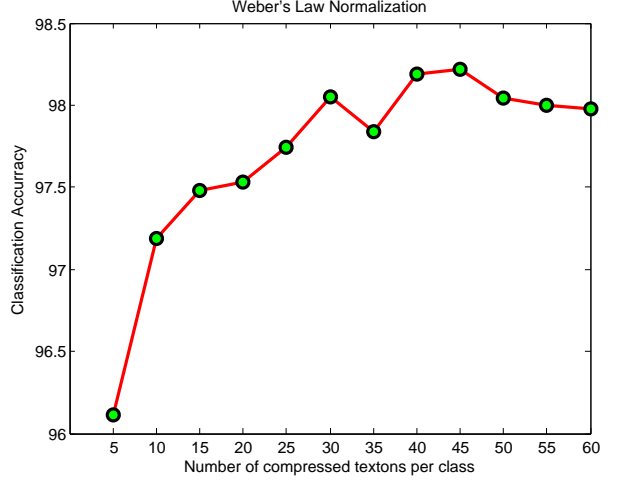


Fig. 14. Classification results on \mathcal{D}^C as a function of the number of compressed textons K per class, for a patch size $n = 11 \times 11$ and CS dimensionality $m = 40$.

TABLE 5

Comparison of highest classification performance on \mathcal{D}^C with a common experimental setup.

Method	LBP	MR8	Patch	Patch-MRF	CS
Accuracy (%)	95.72	97.43	97.17	98.03	98.43

Tables 3 and 4 compare the classification accuracies for Weber’s law and No-normalization. The results indicate that the proposed approach outperforms the Patch method in all three normalizations, and that the classification accuracy differences caused by normalization are only marginally significant.

To summarize the preceding figures and tables, Table 5 presents the overall best classification performance achieved by each method for any parameter setting. The proposed CS method gives the highest classification accuracy of 98.43%, even higher than the best of Patch-MRF in [23], despite the fact that the model dimensionality of the Patch-MRF method is far larger than that of the proposed CS method.

Results on other datasets

The proposed method also performs well on other benchmark datasets (described in Section 4.2). The same detailed sets of experiments for these datasets were performed as for \mathcal{D}^C , however since \mathcal{D}^C is the definitive test for texture classification the following discussion is kept brief.

CUReT small \mathcal{D}^C : Fig. 15 shows the classification accuracy of the CS and the Patch methods on dataset \mathcal{D}^C . We can observe that the proposed method performs similarly to the Patch method but at a much lower dimensionality. As was seen in Fig. 11, it is clear that the classification performance goes down as the patch size is increased, quite different from the CUReT

large dataset \mathcal{D}^C . Nevertheless, this test shows that the proposed CS approach can be well applied in this situation without loss of performance. By comparison, from a recent LBP paper [45], the best performance for this dataset is 86.84% for LBP, and 92.77% for the combination of LBP and NGF with a SVM classifier, in contrast to our CS classification accuracy of 95.85%.

Brodatz large \mathcal{D}^B : Table 6 shows the classification accuracy of the three different normalizations on the \mathcal{D}^B dataset. Similar to the results of the previous section, the proposed method performs better than the patch method, for all normalizations, although by a relatively small margin. We achieve a peak classification accuracy of 96.8% in the Unit-Norm normalization case. The example demonstrates that the proposed CS method can successfully classify 90 texture classes from the Brodatz dataset, despite the large number of classes contained in \mathcal{D}^B which can cause a high risk of mis-classification.

Brodatz small \mathcal{D}^b : Table 7 shows the results on dataset \mathcal{D}^b , where both methods achieve near perfect classification performance, clearly because of the smaller number of texture classes. The highest classification accuracy of 99.95% is achieved by the proposed CS method from a 3×3 patch, results so good that there is hardly room for improvement. By comparison, from a recent LBP paper [45], the best performance for this dataset is 98.49% for LBP alone, and 99.54% for the combination of LBP and NGF with a SVM classifier.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we have described a classification method based on representing textures as a small set of compressed sensing measurements of local texture patches. We have shown that CS measurements of local patches can be effectively used in texture classification. The proposed method has been shown to match or surpass the state-of-the-art in texture classification, but with significant reductions in time and storage complexity. Approximately one third the dimensionality of the original patch space is needed to preserve the salient information contained in the original local patch; any further increase in the number of features yields only marginal improvements in classification performance.

There are significant distinctions between the proposed CS approach and previous studies in texture classification:

- ◆ We demonstrated the effectiveness of random features for texture classification, and the effectiveness of texture classification in the compressed patch domain;
- ◆ The proposed CS approach enjoys the advantage of the Patch method in achieving high classification performance, and that of the pre-selected filter banks in its low dimensional feature space.

Moreover, the proposed CS method facilitates a very straightforward and efficient implementation;

- ◆ We collected the features for texture classification without assuming any prior information about the texture image, except the assumption of sparsity in some (unknown) basis. This is in contrast to conventional texture feature extraction methods which make strong assumptions about the texture being studied.

The promising results of this paper motivate a further examining of CS-based texture classification. First, the use of a more sophisticated classifier, like SVM, may, in some cases, provide enhanced classification performance over the nearest neighbor classifier used in the current study, as in [51] [52]. Furthermore, the proposed approach can also be embedded into the *signature/EMD* framework as is currently being investigated in the texture analysis community, which is considered to offer some advantages over the *histograms/ χ^2* distance framework [24] [25]. Another issue requiring further study is to extend the proposed framework to the pixel-level classification or texture segmentation problem, which is somewhat different from the image-level classification problem considered in this paper. We believe that significant performance gains are still to be realized.

ACKNOWLEDGMENT

The authors would like to thank NSERC Canada, the Department of Systems Design Engineering, and the China Scholarship Council.

REFERENCES

- [1] B. Julesz, "Visual Pattern Discrimination," *IRE trans. on Information Theory*, vol. 8, pp. 84-92, 1962.
- [2] B. Julesz, "Textons, the Elements of Texture Perception and Their Interactions," *Nature*, vol. 290, pp. 91-97, 1981.
- [3] R. M. Haralick, "Statistical and Structural Approaches to Texture," *Proc. of IEEE*, vol. 67, pp. 786-804, 1979.
- [4] T. R. Reed and J. M. H. du Buf, "A Review of Recent Texture Segmentation and Feature Extraction Techniques, *CVGIP: Image Understanding*," vol. 57, no. 3, pp. 359-372, 1993.
- [5] M. Tuceryan and A. K. Jain, "Texture Analysis," *Handbook Pattern Recognition and Computer Vision*, C.H. Chen, L.F. Pau, and P.S.P. Wang, eds., ch. 2, pp. 235-276. Singapore: World Scientific, 1993.
- [6] T. Randen and J. Husøy, "Filtering for Texture Classification: A Comparative Study," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291-310, April 1999.
- [7] J. Zhang and T. Tan, "Brief Review of Invariant Texture Analysis Methods," *Pattern Recognition*, vol. 35, no. 3, pp. 735-747, 2002.
- [8] R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610-621, 1973.
- [9] G. R. Cross and A. K. Jain, "Markov Random Field Texture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 5, no. 1, pp. 25-39, January 1983.
- [10] S. C. Zhu, Y. Wu and D. Mumford, "Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling," *International Journal of Computer Vision*, vol. 27, no. 2, pp. 107-126, 1998.

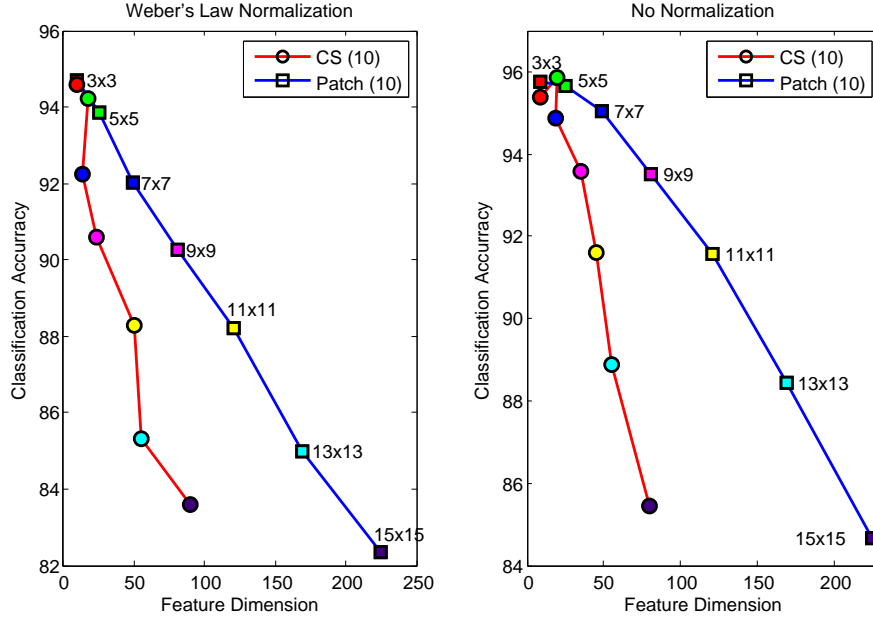


Fig. 15. Comparison classification results on \mathcal{D}^c for $K = 10$ as a function of feature dimensionality for the proposed CS and Patch methods, where color indicates patch size.

TABLE 6

Comparisons of three types of feature vector normalization of the Patch and proposed CS methods on Brodatz large \mathcal{D}^B with $K = 10$ textons per class. Means and standard deviations have been computed over 50 runs.

	Unit Norm Normalization			Weber's Law Normalization			No Normalization		
	3 × 3 (%)	5 × 5 (%)	7 × 7 (%)	3 × 3 (%)	5 × 5 (%)	7 × 7 (%)	3 × 3 (%)	5 × 5 (%)	7 × 7 (%)
Dim	6	15	21	5	15	25	6	15	25
CS (10)	96.2 ± 0.18	96.8 ± 0.14	96.2 ± 0.16	94.3 ± 0.30	95.4 ± 0.39	95.0 ± 0.34	91.8 ± 1.45	93.0 ± 0.44	93.7 ± 0.28
Dim	9	25	49	9	25	49	9	25	49
Patch (10)	96.6 ± 0.09	96.5 ± 0.09	96.1 ± 0.23	94.0 ± 0.11	94.7 ± 0.21	94.8 ± 0.30	92.5 ± 0.09	93.4 ± 0.14	93.3 ± 0.10

TABLE 7

As in Table 6, with results averaged over 100 runs on the small Brodatz dataset \mathcal{D}^b .

	Unit Norm Normalization			Weber's Law Normalization			No Normalization		
	3 × 3 (%)	5 × 5 (%)	7 × 7 (%)	3 × 3 (%)	5 × 5 (%)	7 × 7 (%)	3 × 3 (%)	5 × 5 (%)	7 × 7 (%)
Dim	7	10	15	5	7	7	7	10	15
CS (10)	99.93 ± 0.13	99.93 ± 0.13	99.93 ± 0.10	99.95 ± 0.05	99.90 ± 0.06	99.85 ± 0.20	99.88 ± 0.16	99.85 ± 0.20	99.85 ± 0.18
Dim	9	25	49	9	25	49	9	25	49
Patch (10)	99.93 ± 0.15	99.91 ± 0.15	99.92 ± 0.12	99.90 ± 0.09	99.84 ± 0.13	99.80 ± 0.03	99.67 ± 0.09	99.76 ± 0.11	99.72 ± 0.08

- [11] M. Elfadel and R. W. Picard, "Gibbs Random Fields, Cooccurrences, and Texture Modeling," *IEEE Trans. Pattern Analysis and Machine Intelligence* vol.16, no. 1, pp. 24-37, January 1994.
- [12] X. Qin and Y. H. Yang, "Basic Gray Level Aura Matrices: Theory and its Application to Texture Synthesis," In *Proc. International Conference on Computer Vision*, vol. 1 pp. 128-135, 2005.
- [13] T. Ojala, M. Pietikäinen, and D. Harwood, "A Comparative Study of Texture Measures with Classification Based on Feature Distributions", *Pattern Recognition*, vol. 29, no. 1, pp. 51-59, 1996.
- [14] 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, no. 7, pp. 971-987, July, 2002.
- [15] A. C. Bovik, M. Clark and W. S. Geisler, "Multichannel Texture Analysis Using Localized Spatial Filters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 55-73, Januray 1990.
- [16] B.S. Manjunath and W.Y. Ma, "Texture Features for Browsing and Retrieval of Image Data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837-842, August 1996.
- [17] D. A. Clausi and H. Deng, "Design-Based Texture Feature Fusion Using Gabor Filters and Co-Occurrence Probabilities," *IEEE Trans. Image Processing*, vol. 14, no. 7, pp. 925-936, July 2005.
- [18] T. Chang and C.-C. Kuo, "Texture Analysis and Classification with Tree-Structured Wavelet Transform," *IEEE Transactions on Image Processing*, vol. 2, no. 4, pp. 429-441, October 1993.
- [19] L. M. Kaplan, "Extend Fractal Analysis For Texture Classification and Segmentation," *IEEE Trans. Image Processing*, vol. 8, no. 11, pp. 1572-1585, November 1999.
- [20] T. Leung and J. Malik, "Representing and Recognizing the Visual Appearance of Materials Using Three-Dimensional Textons," *International Journal of Computer Vision*, vol. 43, no.

- 1, pp. 29-44, 2001.
- [21] O. G. Cula and K. J. Dana, "3D Texture Recognition Using Bidirectional Feature Histograms," *International Journal of Computer Vision*, vol. 59, no. 1, pp. 33-60, 2004.
- [22] M. Varma and A. Zisserman, "A Statistical Approach to Texture Classification from Single Images," *International Journal of Computer Vision*, vol. 62, no. 1-2, pp. 61-81, 2005.
- [23] M. Varma and A. Zisserman, "A Statistical Approach to Material Classification Using Image Patches," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2032-2047, November 2009.
- [24] S. Lazebnik, C. Schmid, and J. Ponce, "A Sparse Texture Representation Using Local Affine Regions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265-1278, August, 2005.
- [25] 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*, vol. 73, no. 2, pp. 213-238, 2007.
- [26] E. J. Candès and T. Tao, "Decoding by Linear Programming" *IEEE Trans. Information Theory*, vol. 51, no. 12, pp. 4203-4215, December, 2005.
- [27] E. J. Candès and T. Tao, "Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies?" *IEEE Trans. Information Theory*, vol. 52, no. 12, pp. 5406-5425, December, 2006.
- [28] D. L. Donoho, "Compressed sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289-1306, April, 2006.
- [29] E. J. Candès and M. B. Wakin, "An Introduction to Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21-30, March 2008.
- [30] G. Peyré, "Sparse Modeling of Textures," *Journal of Mathematical Imaging and Vision*, vol. 34, no. 1, pp. 17-31, 2009.
- [31] J. Mairal, F. Bach, J. Ponce, Guillermo Sapiro, and A. Zisserman, "Discriminative Learned Dictionaries for Local Image Analysis," In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [32] K. Skretting and J. H. Husøy, "Texture Classification Using Sparse Frame-Based Representations," *EURASIP Journal on Applied Signal Processing*, vol.1, pp. 102-102, 2006.
- [33] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to Sense Sparse Signals: Simultaneous Sensing Matrix and Sparsifying Dictionary Optimization," *IEEE Trans. Image Processing*, vol. 18, no. 7, 1935-1408, July 2009.
- [34] D. Donoho and J. Tanner, "Observed Universality of Phase Transitions in High-Dimensional Geometry, With Implications for Modern Data Analysis and Signal Processing," *Phil. Trans. R. Soc. A*, vol. 367, no. 1906, pp. 4273-4293, November, 2009.
- [35] R. Baraniuk and M. Wakin, "Random Projections of Smooth Manifolds," *Foundations of Computational Mathematics*, vol. 9, no. 1, pp. 51-77, January 2009.
- [36] S. Dasgupta and Y. Freund, "Random Projection Trees for Vector Quantization," *IEEE Trans. on Information Theory*, vol. 55, no. 7, pp. 3229-3242, July 2009.
- [37] J. Romberg, "Imaging via Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 14-20, 2008.
- [38] J. Wright, A. Yang, A. Ganesh, S. S. Sastry and Y. Ma "Robust Face Recognition via Sparse Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210-217, February, 2009.
- [39] D. Needell and J. A. Tropp, "CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301-321, May, 2009.
- [40] J. Haupt, R. Castro, R. Nowak, G. Fudge, A. Yeh, "Compressive Sampling for Signal Classification," In *Proc. Asilomar Conference on Signals, Systems and Computers*, pp. 1430-1434, 2006.
- [41] E. Levina, *Statistical Issues in Texture Analysis*. PhD thesis, University of California at Berkeley, 2002.
- [42] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323-2326, 2000.
- [43] M. Varma and A. Zisserman. "Texture Classification: Are Filter Banks Necessary?" In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 691-698, 2003.
- [44] J. M. Geusebroek, A. W. M. Smeulders, and J. V. Weijer, "Fast Anisotropic Gauss Filtering," *IEEE trans. on Image Processing*, vol. 12, no. 8, pp. 938-943, August, 2003.
- [45] S. Liao, Max W. K. Law, and Albert C. S. Chung, "Dominant Local Binary Patterns for Texture Classification," *IEEE trans. on Image Processing*, vol. 18, no.5, pp. 1107-1118, May 2009.
- [46] M. Pietikäinen, T. Nurmela, T. Mäenpää, and M. Turtinen, "View-based Recognition of Real-World Textures," *Pattern Recognition*, vol. 37, no. 2, pp. 313-323, 2004.
- [47] T. Mäenpää and M. Pietikäinen, "Multi-Scale Binary Patterns for Texture Analysis," *SCIA*, 2003.
- [48] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover Publications, New York, 1966.
- [49] K.J. Dana, B. van Ginneken, S.K. Nayar, and J.J. Koenderink. "Reflectance and Texture of Real-World Surfaces", *ACM Transactions on Graphics*, vol. 18, no. 1, pp. 1-34, January 1999.
- [50] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd Ed, Academic Press, Dec, 2008.
- [51] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, "On the Significance of Real-World Conditions for Material Classification," In *Proc. European Conference on Computer Vision*, pp. 253-266, 2004.
- [52] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim, "Support Vector Machines for Texture Classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, 1542-1550, November 2002.
- [53] S. Graf and H. Luschgy, *Foundations of Quantization for Probability Distributions*, New York: Springer-Verlag, 2000.
- [54] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification* (2nd Edition), Academic Press , 2000.



Li Liu received the B.S. degree in communication engineering and the M.S. degree in remote sensing and geographic information system from the National University of Defense Technology, Changsha, China, in 2003 and 2005, respectively, where she is currently pursuing the Ph.D. degree.

She is currently a Visiting Student at the University of Waterloo, Ontario, Canada. Her current research interests include computer vision, texture analysis, pattern recognition

and image processing.



Paul W. Fieguth (S'87 - M'96) received the B.A.Sc. degree from the University of Waterloo, Ontario, Canada, in 1991 and the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1995, both degrees in electrical engineering.

He joined the faculty at the University of Waterloo in 1996, where he is currently Professor in Systems Design Engineering. He has held visiting appointments at the University of Heidelberg in Germany, at IN-

RIA/Sophia in France, at the Cambridge Research Laboratory in Boston, at Oxford University and the Rutherford Appleton Laboratory in England, and with postdoctoral positions in Computer Science at the University of Toronto and in Information and Decision Systems at MIT. His research interests include statistical signal and image processing, hierarchical algorithms, data fusion, and the interdisciplinary applications of such methods, particularly to remote sensing.