# Texture Classification from Random Features

Li Liu and Paul W. Fieguth, Member, IEEE

**Abstract**—Inspired by theories of sparse representation and compressed sensing, this paper presents a simple, novel, yet very powerful approach for texture classification based on random projection, 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 a compressed 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 each of the CUReT, the Brodatz, and the MSRC databases, comparing the proposed approach to four state-of-the-art texture classification methods: Patch, Patch-MRF, MR8, and LBP. We show that our approach leads to significant improvements in classification accuracy and reductions in feature dimensionality.

Index Terms—Texture classification, random projections, sparse representation, compressed sensing, textons, image patches, bag of words.

# **1** INTRODUCTION

**T**EXTURE 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].

The design of a texture classification system essentially involves two major steps: 1) feature extraction and 2) classification. Most research in texture classification focuses on the feature extraction part [3], with extensive surveys [2], [3], [4]. Early methods include Gray Level Cooccurrence Histograms [5], Gray Level Difference Histograms [6], Gray Level Run Length Histograms [6], Markov Random Fields [7], [8], Simultaneous AutoRegressive models [9], Fractal Models [10], among many others. A quantum jump occurred in the 1980s when Gabor filters [11], [12], pyramid filters [13], and Wavelets [14] were introduced, followed by further advances when simple

- L. Liu is with the Remote Sensing Information Processing Laboratory, School of Electronic Science and Engineering, National University of Defense Technology, Room 436, 47 Yanwachi, Changsha 410073, Hunan, China. E-mail: dreamliu2010@gmail.com.
- P.W. Fieguth is with the Vision and Image Processing (VIP) Laboratory, Department of Systems Design Engineering, University of Waterloo, 200 University Ave. W., Waterloo, ON N2L-3G1, Canada. E-mail: pfieguth@uwaterloo.ca.

Manuscript received 29 Oct. 2010; revised 24 May 2011; accepted 6 June 2011; published online 13 July 2011.

Recommended for acceptance by D. Cremers.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number

TPAMI-2010-10-0824.

statistics were replaced by multidimensional histograms (either marginal or joint), such as Gray Level Aura Histograms [15], Local Binary Patterns [17], and many others [8]. All of these choose a limited subset of texture features from local image patches. However, as Randen and Husøy [3] 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* or nonlocal appearance, representing the repetition of and the relationship among local patterns. Recently, a "Bag-of-Words" (BoW) model, borrowed from the text literature, opened up new prospects for texture classification [18], [19], [20], [21], [22], [23]. 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 by detecting points of interest, and then using local descriptors to extract features locally to each such point [22], [23], or
- 2. densely extracting local features pixel by pixel over the input image.

The success of the sparse approach largely depends on the type of texture, some of which might not produce enough regions for a robust representation. 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

Digital Object Identifier no. 10.1109/TPAMI.2011.145.

orientations [18], [20]. However, more recently, in [21] 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 the size of the patch. Small patch sizes cannot capture large-scale structures that may be the dominant texture feature, are not very robust against local changes in texture, and are highly sensitive to noise and illumination variations. However, a large patch size leads to a quadratic increase in the dimension of the patch space, with the high dimensionality posing two challenges to the clustering algorithms used to learn textons. First, the presence of irrelevant and noisy features can mislead the clustering algorithm; second, in high dimensions, data may be very sparse (the so-called 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 improved 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.

In this paper, we seek *nonadaptive, information-preserving, universal-dimensionality reduction* of texture patches, based on two motivations. First, although an enormous volume of literature has been devoted to data-dependent feature extraction and dimensionality reduction, there is little consensus about which features are better or worse, and practitioners lack guidelines on which to use. Second, we want to avoid the disadvantages of popular dimensionality reduction techniques such as principle component analysis (PCA): Its data-dependent nature, the computational burden of eigendecompositions, and the absence of any guarantee that distances in the original and projected spaces are well preserved. In summary, we desire a computationally simple method of dimensionality reduction that does not introduce significant distortions.

Random projection (RP) [34], [35], [39] refers to the technique of projecting a set of points from a high-dimensional space to a randomly chosen low-dimensional subspace. The technique has been used in combinatorial optimization, information retrieval, face recognition [33], and machine learning [38], [43]. Fig. 1 shows a simple example, contrasting the distribution of raw pixels, filter responses, and random features.

The information-preserving and dimensionality-reduction power of RP is firmly evidenced in the emerging theory of compressed sensing (CS) [24], [25], [26], which states that, for sparse and compressible signals, a small number of nonadaptive linear measurements in the form of random projections can capture most of the salient information in the signal and allow for perfect reconstruction of the signal. Moreover, RPs have also played a central role in providing feasible solutions to the well-known Johnson-Lindenstrauss (JL) lemma [35], which states that a point set in a highdimensional euclidean space can be mapped down onto a space of dimension logarithmic in the number of points,



Fig. 1. Random projections of local patches form good shape clusters and can distinguish texture classes. For the three Brodatz textures, left, the panels compare the distribution and separability of (a) (b) raw pixel values, (c) two linear filter responses, and pairs of random projections extracted from patches of size (d)  $9 \times 9$ , (e)  $15 \times 15$ , and (f)  $25 \times 25$ .

such that the distances between the points are approximately preserved.

When applying RP to texture classification, the key question is therefore how much information about texture patches can be preserved by random projections, the number of requisite dimensions for near-lossless mapping, and whether such a mapping leads to any advantages in classification. Fortunately, one important reason that the theory of sparse representation and compressed sensing has attracted significant attention in the signal processing community is due in part to the fact that an important variety of signals, such as audio and natural images (including texture images), can be well approximated by a linear combination of a few atoms of some redundant dictionary [25], [59]. CS theory implies that the precise choice of the number of features should not be 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.

To the best of our knowledge, this paper is the first to investigate RP for texture classification, presenting a comprehensive series of experiments to illustrate the benefits of this novel technique for texture classification. The proposed method is computationally simple, yet exceptionally powerful. Simply by selecting a set of random features in a bag-of-words classification context, with no further parameter tuning, we find a texture classifier that meets or exceeds the current state of the art!

The rest of this paper is organized as follows: Section 2 provides an in-depth review of RP, with connections to the JL lemma and CS theory. 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 provide an analysis of the benefits and advantages of the proposed approach. In Section 4, we test the proposed method with extensive experiments to compare with the current state of the art.

122.1

## **2** BACKGROUND AND RELATED WORK

## 2.1 Sparse Representation, Random Projection, and Compressed Sensing Background

Random features are essentially equivalent to the compressive measurements in the CS encoding stage. CS has been brought to the forefront by the work of Candès and Tao [25] and Donoho [26], who have shown that a small number of nonadaptive linear measurements (random projections) of a high-dimensional but sparse or compressible signal contains enough information for near-perfect reconstruction and processing. Sparsity and compressive sensing have had a growing impact on a much broader range of fields, including signal and image processing, pattern recognition, computer vision, and machine learning [42], [45], [48], [49].

In CS, however, the encoder requires no a priori knowledge of the signal structure. Only the decoder uses the sparse model to reconstruct the signal. So, although the decoding (reconstruction) process is of great importance in applications of CS, in this paper we focus on the encoder, the random projection. Viewed from a geometric perspective, stable CS reconstruction requires a "stable" embedding of all the *g*-sparse signals in the basis  $\Psi$  in  $\mathbb{R}^{m \times 1}$ , such that distinct g-sparse signals in  $\mathbb{R}^{n \times 1}$  remain well separated in  $\mathbb{R}^{m \times 1}$ , such that  $\Phi$  is *information preserving*. The restricted isometry property (RIP) [24], [25] is used to quantify this stability, and the cost for stability is a modest logarithmic "excess" dimensionality factor with respect to the sparsity level g. Baraniuk et al. [42] have identified a fundamental connection between CS and the Johnson-Lindenstrauss lemma [35], [37], [39], which is concerned with the stable embedding of a finite point set under a random dimensionality reducing projection, showing that the RIP can be thought of as a straightforward consequence of the JL lemma.

#### 2.2 Related Work

Classification using random measurements has received only minimal treatment to date. In [52], the utility of CS projection observations for signal classification is investigated, while in [40], random measurements are exploited to perform manifold-based image classification. One of the most successful applications of the sparse representation and CS in computer vision and pattern recognition has been the SRC algorithm for face recognition [49], which 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. The use of the face image data itself as dictionary without learning is novel. 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. Second, SRC is reconstruction based, *explicitly* reconstructing the sparse  $\underline{\theta}$ , a computationally intensive step which we avoid.

Lazebnik and Raginsky recently introduced an elegant dictionary learning algorithm based on Information Loss Minimization [31]. It learns a codebook with the objective of obtaining a quantization that does not cause high distortion and at the same time keeps nearly all of the information about the class of the original signal. Sparsity also has been recently incorporated into a very interesting robust face recognition framework by Ma et al. [49], motivated by the work on compressed sensing and random projections. This work does not explicitly enforce reconstruction and/or discrimination nor does it learn adapted dictionaries.

In the field of texture classification, there are two important threads of research. One is the sparse coding with overcomplete dictionaries, a work based on *generative modeling* [16], [44], and the other is the *k*-mean clustering for textons based on *discriminative modeling* [18].

In the former case, the analysis of a given texture involves the sparse coding of all of the patches of a texture into a dictionary of atoms. Research along this line, such as the work in [44], [45], focuses on learning nonparametric redundant dictionaries that facilitate a sparse representation of the data [30], [46]. Since originally trained to contain sufficient information for reconstruction, sparse representations are, from the point of view of signal classification, a reconstructive approach. Texture classification based on this framework is formulated in the following way: Assume that we have sets  $\mathcal{Y}_c \subset \mathbb{R}^{n\times 1}$  of training patches,  $c = 1, \ldots, C$ , extracted from *C* different texture classes. First, a texture dictionary  $\Psi_c \in \mathbb{R}^{n \times K}$  of *K* atoms is learned for each texture class with  $\mathcal{Y}_c$  via

$$\min_{\boldsymbol{\Psi}_{c}, \boldsymbol{\underline{\theta}}_{i}} \sum_{i=1}^{|\mathcal{Y}_{c}|} \| \underline{\boldsymbol{y}}_{i} - \boldsymbol{\Psi}_{c} \underline{\boldsymbol{\theta}}_{i} \|_{2}^{2} \quad \text{subject to} \quad \| \underline{\boldsymbol{\theta}}_{i} \|_{l_{0}} \leq g.$$
(1)

Texture classification is done by approximating each patch  $\underline{y}^{\text{new}}$  in the testing image using a constant sparsity g and the  $\overline{C}$  different dictionaries. This provides C different residual errors, which can then be used as classification features [44], [46]. Thus, assigning the class membership  $c_{\text{new}}$  for some patch  $\underline{y}^{\text{new}}$  is to write

$$\hat{c}_{\underline{\boldsymbol{y}}^{\text{new}}} = \arg\min_{c \in \{1, \dots, C\}} \mathcal{R}^{\star}(\underline{\boldsymbol{y}}^{\text{new}}, \boldsymbol{\Psi}_{c}), \qquad (2)$$

where

$$\begin{split} \mathcal{R}^{\star}(\underline{\boldsymbol{y}}, \boldsymbol{\Psi}) &= \|\underline{\boldsymbol{y}} - \boldsymbol{\Psi}\underline{\boldsymbol{\theta}}^{\star}(\underline{\boldsymbol{y}}, \boldsymbol{\Psi})\|_{2}^{2} \\ \underline{\boldsymbol{\theta}}^{\star}(\underline{\boldsymbol{y}}, \boldsymbol{\Psi}) &= \operatorname*{arg\ min}_{\boldsymbol{\theta} \in \ \mathbb{R}^{K \times 1}} \|\underline{\boldsymbol{y}} - \boldsymbol{\Psi}\underline{\boldsymbol{\theta}}\|_{2}^{2} \quad s.t. \quad \|\underline{\boldsymbol{\theta}}\|_{l_{0}} \leq g. \end{split}$$

Both the optimization problem (1) for learning the overcomplete dictionary and the reconstruction based classification problem (2) are very time consuming, especially when the number of texture classes C is large. In this paper, we seek to avoid these two computationally intensive problems.

In discriminative modeling, the texton dictionary learning step almost universally employs the *k*-means algorithm. There is an intriguing relation between sparse representation and clustering (i.e., vector quantization) [29], [30]. In clustering, a set of descriptive vectors is learned, and each sample is represented by one of those vectors, which can be thought of as an extreme sparse representation, where only one atom is allowed in the signal decomposition.

In this paper, we focus on image-level texture classification, which does not necessarily aim at classifying the individual image patch center at each pixel correctly, but rather combining all of the patches from an image into a highlevel representation, such as a global BoW model. Our work follows the discriminative modeling framework, proposing to learn the overcomplete nonparametric dictionary and to perform the texture classification problem in the compressed domain. In contrast with the work by Varma and Zisserman [21], we consider linear projections of the local texture patch as the available measurements, not the image patch itself.

Previously, random projection has been studied as a general dimensionality reduction method for numerous clustering problems [37], [38], [39], [43] as well as for learning nonlinear manifolds [42], [48]. In contrast to this previous work, the number of measurements in our work does not depend on the number of training examples, only on the sparsity g and a logarithmic dependence on the dimension of the data domain n, which allows our dimensionality reduction technique to scale well.

# 3 TEXTURE CLASSIFICATION USING RANDOM PROJECTION

### 3.1 Sparse Modeling of Textures

The premise underlying CS is one of signal sparsity or compressibility. The compressibility of textures is certainly well established: Most natural images are compressible, as extensive experience with the wavelet transform has demonstrated [59], and textures, being roughly stationary/periodic, are all the more sparse [44].

The key to the universality of CS is that the sparsifying dictionary  $\Psi$  does not need to be known, and that an explicit a priori knowledge of the signal is not necessary.

Dictionary  $\Psi$  maps the underlying coefficient vector  $\underline{\theta}$  to the sparse domain  $\underline{p} \approx \Psi \underline{\theta}$ ; in contrast,  $\Phi$  maps from the sparse to the measurement domain,  $\underline{x} = \Phi \underline{p}$ . Although it is generally not possible to reconstruct a signal without knowing its sparsity basis  $\Psi$ , if we are only interested in classification and not reconstruction there is no need to know  $\Psi$ .

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 known to be few in number. In [53], the author first uses a filter bank to reduce the patch space and then further reduces the dimensionality by projecting filter marginals onto lowdimensional manifolds by a Locally Linear Embedding algorithm [54], showing that classification accuracy can be increased by projecting onto a manifold of some suitable dimension. Cula and Dana [19] 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  $\underline{x} \in \mathbb{R}^{m \times 1}$ .

In practice, dimensionality reduction is important in handling high-dimensional data since it mitigates the curse of dimensionality and other undesired properties of highdimensional spaces. The most widely used methods are factorial methods, such as PCA and variations; unfortunately, these are computationally expensive, with no guarantee that the distances between the original and projected observations are well preserved. In this section, we argue that random projections are particularly well suited for our purposes.

We propose a dimensionality reduction

$$\underline{\boldsymbol{x}} = \boldsymbol{\Phi} \boldsymbol{p}, \tag{3}$$

ideally where  $m \ll n$ . Clearly,  $\Phi \in \mathbb{R}^{m \times n}$ , m < n, loses information in general since  $\Phi$  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 feature extractor  $\Phi$  is to have the null space of  $\Phi$  orthogonal to the low-dimensional subspace of the sparse signal p.

Ideally, we wish to ensure that  $\Phi$  is information preserving, by which we mean that  $\Phi$  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:

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

for small  $\epsilon > 0$ . One of the key results in [24] from CS theory is the Restricted Isometry Property, which states that (4) is indeed satisfied with overwhelming probability by certain random matrices. Moreover, (4) is also the direct result of the JL lemma. It is precisely on this very strong theoretical support that we propose to use random projections to rethink texture classification.

The Johnson-Lindenstrauss lemma [35] is concerned with the following problem: We are given a set  $\mathcal{D}$  of d points in  $\mathbb{R}^{n \times 1}$  with n typically large. We would like to embed these points into a lower dimensional euclidean space  $\mathbb{R}^m$  while approximately preserving the relative distances between any two of these points.

**Theorem 1 (Johnson-Lindenstrauss lemma [35]).** For any  $0 < \epsilon < 1$  and any positive integer d, let m be a positive integer such that

$$m \ge 4(\epsilon^2/2 - \epsilon^3/3)^{-1} \ln d.$$
 (5)

Then, for any set  $\mathcal{D}$  of d points in  $\mathbb{R}^{n \times 1}$ , there exists a Lipschitz mapping  $f : \mathbb{R}^{n \times 1} \to \mathbb{R}^{m \times 1}$  such that for every pair  $u, v \in \mathcal{D}$ ,

$$(1-\epsilon)\|u-v\|^{2} \le \|f(u) - f(v)\|^{2} \le (1+\epsilon)\|u-v\|^{2}.$$
 (6)

There are proofs of the lemma that show that (6) can be satisfied with very high probability with f taken as a linear mapping represented by an  $m \times n$  matrix  $\Phi$ , whose entries are randomly drawn from certain probability distributions [37], specifically including the Gaussian distribution [27], [37], [42]. Furthermore, Baraniuk et al. [42] give a simple technique for verifying the Restricted Isometry Property for random matrices that underlies CS, and clearly illustrates that the RIP can be thought of as a consequence of the JL lemma, and that any distribution that yields a satisfactory JL-embedding will also generate matrices satisfying the RIP. As a consequence, random Gaussian projections approximately preserve pairwise distances in the data set.

## 3.2.1 Example 1: RP and Texture

A simple example, illustrated in Fig. 2, reconstructs a texture patch based on random measurements. The reconstruction,



Fig. 2. Reconstruction of ideal sparse texture signals from random measurements.

results from different numbers of measurements, are shown in Figs. 2b, 2c, and 2d. The reconstruction algorithm CoSaMP [51] is used here. With a sufficiently large number of random measurements, the original sparse texture is perfectly reconstructed.

## 3.2.2 Example 2: RP and Classification

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

$$\underline{\boldsymbol{x}} = \boldsymbol{\Phi} \boldsymbol{p} + \underline{\boldsymbol{v}},\tag{7}$$

where  $\underline{\boldsymbol{v}} \in \mathbb{R}^{m \times 1}$  is a noise term, independent of  $\underline{\boldsymbol{p}}$ . Suppose we wish to classify  $\underline{\boldsymbol{p}}$  based on the noise-corrupted compressed measurements  $\underline{\boldsymbol{x}}$ , using a single nearest neighbor classifier with a euclidean distance measure. Our underlying patterns are a set of 100 sinusoids, as plotted in Fig. 3a:

$$\{\underline{\boldsymbol{p}}_{k}(t)\}_{k=1}^{100} = \{\cos(\omega_{k}t)\}_{k=1}^{100}.$$
(8)

Fig. 3b plots the probability of classification accuracy as a function of the number of measurements m, averaging over 100,000 trials, where for each trial, independent realizations of compressed sensing measurements and noise are generated. Note that classification was performed directly in the compressed domain and without any explicit sparse reconstruction.

## 3.3 Random Measurements and Clustering

A patch approach models textures by the joint distribution of pixel intensities in a local patch. Let  $\underline{p} \in \mathbb{R}^{n \times 1}$  be the pixels in a local patch of size  $\sqrt{n} \times \sqrt{n}$ . Similar to [8], we assume that there exists a "true" joint probability distribution density  $f(\underline{p})$  over the image patch space. We consider homogeneous textures; thus,  $f(\mathbf{p})$  is stationary.

We wish to preserve both local texture information, contained in a local image patch, and global texture appearance, representing the repetition of and the relationship among local textures. It has been shown that a texton-based approach is an effective local-global representation [18], [21].

The textons are trained by adaptively partitioning the feature space into clusters using *K*-means. For an input data set  $\mathcal{X} = \{\underline{x}_1, \dots, \underline{x}_{|\mathcal{X}|}\}, \underline{x}_i \in \mathbb{R}^{m \times 1}$ , and an output texton set  $\mathcal{W} = \{\underline{w}_1, \dots, \underline{w}_K\}, \underline{w}_i \in \mathbb{R}^{m \times 1}$ , the quality of a clustering solution is measured by the *average quantization error* [50], denoted as  $Q(\mathcal{X}, \mathcal{W})$ :

$$Q(\mathcal{X}, \mathcal{W}) = \frac{1}{|\mathcal{X}|} \sum_{j=1}^{|\mathcal{X}|} \min_{1 \le k \le K} \|\underline{\boldsymbol{x}}_j - \underline{\boldsymbol{w}}_k\|_2^2,$$
(9)

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



Fig. 3. Signal classification based on random features: (a) A set of 100 synthetic similar periodic signals, each of length n = 400. (b) Classification accuracy as a function of the number of random measurements for both noisy and noise-free cases.

 $Q(\mathcal{X}, \mathcal{W})$  goes as  $K^{-2/m}$  for large K [61], a problem when m 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 [21] 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.

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

Consequently, in this paper, we propose to cluster in the *compressed domain*:

$$\mathcal{X} = \{ \underline{\boldsymbol{x}} = \boldsymbol{\Phi} \boldsymbol{p} \mid \boldsymbol{p} \in \mathcal{P} \},$$
(10)

where  $\mathbf{\Phi} = [\phi_{i,j}] \in \mathbb{R}^{m \times n}$  is the Gaussian measurement matrix whose elements  $\phi_{i,j}$  are independent, zero-mean, unit-variance Gaussian random variables.

### 3.4 Proposed Approach

#### 3.4.1 Patch Extraction

Since we focus on the local geometry of textures through the extraction of local patches, we first illustrate the patch extraction strategy used in this paper. An image **I** of *N*-pixels is processed by extracting patches  $\{\underline{p}_{i,j}\}_{ij}$  of size  $\sqrt{n} \times \sqrt{n}$  around each pixel position (i, j) except those pixels on the image boundary. Formally, such a linear operator that extracts all the patches (with pixels on the boundary excluded) from an image **I** is denoted as follows:



Fig. 4. Overview of the texture classification system proposed in this paper: (a) Texton dictionary learning in the compressed patch domain. (b) The architecture of training and classification.

$$\Xi: \mathbf{I} \longmapsto \{\underline{\boldsymbol{p}}_{i,j}\}, \text{ for all } i, j \in (\sqrt{n}/2, \sqrt{N} - \sqrt{n}/2].$$
(11)

Each patch  $\underline{p}_i$  is handled as a vector of size *n*. For ease of notation, we will use one index *i* instead of two *i*, *j* for a patch.

Suppose 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 data set. A set of  $\sqrt{n} \times \sqrt{n}$  image patches  $\mathcal{P} = \{\underline{p}_{c,s}i\}_{i}$  is extracted from image  $\mathbf{I}_{c,s}$  via (11).

Our proposed classifier is identical to the Patch method [21] except that, instead of using  $\underline{p}$ , the random measurements  $\underline{x} = \Phi \underline{p}$  derived from  $\underline{p}$  are used as features, where the entries of  $\Phi$  are sampled from independent zero-mean, unit-variance normal distribution. The *compressed domain* 

$$\mathcal{X} = \{ \underline{\boldsymbol{x}} = \boldsymbol{\Phi} \boldsymbol{p} \mid \boldsymbol{p} \in \mathcal{P} \}$$
(12)

is a compressed representation of the patch domain

$$\mathcal{P} = \{ \underline{\boldsymbol{p}} \mid \underline{\boldsymbol{p}} \in \mathbb{R}^{n \times 1} \}.$$
(13)

Our texture classification system is illustrated in Fig. 4, consisting of the following stages:

- Compressed texton dictionary learning stage, illustrated in Fig. 4a, in which a universal compressed texton dictionary W is learned directly in the compressed domain X, not in P. For each texture class, we learn K textons with k-means for each class. Then, the universal compressed texton dictionary W is formed by concatenating the K textons of each texture class, resulting a dictionary of size CK (i.e., W = CK).
- Histogram of textons learning stage, illustrated in Fig. 4b (left): A histogram <u>h</u><sub>c,s</sub> of compressed textons is learned for each training sample I<sub>c,s</sub> by labeling each of its extracted patches (via (11)) with the closest texton in W. Each texture class is represented by a set of models H<sub>c</sub> = {<u>h</u><sub>c,s</sub>}<sub>s</sub>.
- 3. The classification stage, shown in Fig. 4b (right): A histogram  $\underline{h}_{new}$  for a given image is computing as in step 2, and  $\underline{h}_{new}$  is classified using a nearest

neighbor classifier, where the distance between two histograms is measured using the  $\chi^2$  statistic:  $\chi^2(\underline{h}_1, \underline{h}_2) = \frac{1}{2} \sum_{k=1}^{CK} \frac{[\underline{h}_1(k) - \underline{h}_2(k)]^2}{\underline{h}_1(k) + \underline{h}_2(k)}.$ 

## 4 EXPERIMENTAL EVALUATION

## 4.1 Methods in Comparison Study

Our specific experimental goal is to compare the proposed approach with the state of the art

**Patch [21]**. Based on local patches of size  $\sqrt{n} \times \sqrt{n}$ ; both training and testing are performed in the patch domain.

**Patch-MRF [21]**. A texture image is represented using a two-dimensional histogram: one dimension for the quantized bins of the patch center pixel, the other dimension for the learned textons from the patch with the center pixel excluded. The number of bins for the center pixel in [21] is as large as 200, and the size of the texton dictionary is  $61 \times 40 = 2,440$ , resulting in an extremely high dimensionality of  $2,440 \times 200 = 488,000$ .

MR8 [20], [21]. Eight filter responses derived from the responses of 38 filters (see Fig. 5). A complicated anisotropic Gaussian filtering method was used to calculate the MR8 responses.

**LBP** [17], [55]. The rotationally invariant, uniform LBP texton dictionary at different scales,  $LBP_{8,1}^{riu2}$ ,  $LBP_{8,1+16,2'}^{riu2}$ ,  $LBP_{8,1+16,2+24,3'}^{riu2}$ ,  $LBP_{8,1+16,2+24,3+24,4'}^{riu2}$ , and  $LBP_{8,1+16,2+24,3+24,4+24,5}^{riu2}$  advocated in [56], [57]. For simplicity, in the remainder of this paper, these LBP textons are denoted as 1-scale,..., 5-scale, respectively.



Fig. 5. The original filter bank for obtaining the MR8 filter responses: edge and bar filters at three scales and six orientations, plus a Gaussian and Laplacian of Gaussian.

TABLE 1 Summary of Texture Data Sets Used in the Experiments

Texture	Rotation	Rotation	Controlled	Illumination	Texture	Sample	Samples	$S_1$	$S_2$
Dataset	In plane	Off plane	Illumination	Variation	Classes	Size	per Class	(Train)	(Test)
$D^b$				$\checkmark$	24	$128\times128$	25	13	12
$D^B$				$\checkmark$	90	$128\times128$	25	13	12
$\mathcal{D}^{BFull}$				$\checkmark$	111	$215\times215$	9	3	6
$\mathcal{D}^{c}$			$\checkmark$		61	$106\times106$	9	5	4
$\mathcal{D}^{C}$	$\overline{\mathbf{v}}$	$\checkmark$	$\checkmark$	$\sim$	61	$200 \times 200$	92	46	46
$\mathcal{D}^M$	$\sim$		$\checkmark$	$\checkmark$	16	$1024 \times 768$	20	15	5

 $S_1$  and  $S_2$  denote the number of training and testing samples per texture class, respectively.



Fig. 6. Brodatz small: 24 textures used in [55] from the Brodatz database.

	A BAR	
	A AN	

Fig. 7. Five nonhomogeneous textures (D43, D44, D45, D91, and D97) from the Brodatz database.

#### 4.2 Texture Data Sets and Experimental Setup

For our experimental evaluation, we have used three commonly used texture data sets, summarized in Table 1: the Brodatz album [58], the CUReT database [20], and the MSRC database.

The **Brodatz small data set**  $D^b$  **(24 classes)** was chosen to allow a direct comparison with the state-of-the-art results from [55]. There are 24 homogeneous texture images (shown in Fig. 6), each of which was partitioned into 25 nonoverlapping subimages of size of  $128 \times 128$  pixels, of which 13 samples for training and the remaining 12 for testing.

The **Brodatz large data set**  $\mathcal{D}^B$  (90 classes) is a very challenging platform for classification 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 (e.g., D43, D44, D45, D91, and D97, as illustrated in Fig. 7). 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  $\mathcal{D}^b$ . For the **Brodatz Full data set**  $\mathcal{D}^{\text{BFull}}$ , we keep all 111 classes,

For the **Brodatz Full data set**  $\mathcal{D}^{\text{Brun}}$ , we keep all 111 classes, challenging due to the relatively large number of texture classes and the small number of samples per class. To obtain results comparable with Lazebnik et al. [22] and Zhang et al. [23], we used a consistent approach, dividing each texture image into nine nonoverlapping subimages, of which three samples were for training and the remaining six for testing.

The Brodatz database has been criticized because of its lack of intraclass variation, which motivated the development of



Fig. 8. Image examples from three different texture classes of CUReT textures under different illuminations and viewpoints.



Fig. 9. MSRC: 16 materials in the MSRC textile database.

the CUReT database [20] which has now become a benchmark and is widely used to assess classification performance.

For the **CUReT large data set**  $D^C$  (**61 classes**), we use the same subset of images as Varma and Zisserman [20], [21], with 92 images for each class. These images are captured under different illumination and viewing directions, a few of which are plotted in Fig. 8. Half of the samples are chosen for training and the remaining half for testing.

The **CUReT small data set**  $D^c$  (**61 classes**) preserves all texture classes of  $D^C$ ; however, each texture is represented by only a single image, as in [55], where all of the textures have the same illumination and imaging conditions. Each image is partitioned into nine  $106 \times 106$  nonoverlapping subimages, with five samples for training and the other four for testing.

The **MSRC data set**  $\mathcal{D}^M$  (16 classes), used by Varma and Zisserman [21], has 16 folded textile materials (shown in Fig. 9). Similarly to the CUReT database, the impact of non-Lambertian effects is very obvious. Furthermore, it is an interesting database to analyze due to the variations in pose and the nonrigid deformations of the textured surface. As in Varma and Zisserman [21], 15 images were randomly selected from each texture classes for training set and the remaining five for testing. Textons were learned from only three images per class randomly selected from the training set.

In terms of the extracted RP vector, we consider three kinds of normalization:

1. Weber's law [21]:

$$\underline{\boldsymbol{x}} \leftarrow \underline{\boldsymbol{x}} \left[ \frac{\log(1 + \|\underline{\boldsymbol{x}}\|_2 / 0.03)}{\|\underline{\boldsymbol{x}}\|_2} \right].$$
(14)

2. Unit norm:

$$\underline{\boldsymbol{x}} \leftarrow \frac{\underline{\boldsymbol{x}}}{\|\underline{\boldsymbol{x}}\|_2}.$$
 (15)

#### 3. No normalization.

TABLE 2 Classifier Variability: Standard Deviations Are Reported from 20 Runs on  $\mathcal{D}^C$  Using 10 Textons per Class, a Patch Size of  $11 \times 11$ , Weber's Law Normalization

Varied	Varied	Varied	RP Dimensionality	
Training/Tesing	RP Matrix	K-means	40	6
			0.09%	0.16%
		, V	0.15%	0.33%
$\checkmark$	$\overline{}$	, V	0.20%	0.69%

#### 4.3 Experimental Tests

## 4.3.1 Variability Analysis

Because RP 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,
- 2. variation in the random projection 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 RP matrix, it is a modest fraction of the total variability, and therefore in no way compromises the RP method as a classifier.

## 4.3.2 RP Parameter Choices

There are three key parameters in the RP classifier:

- 1. the number of textons *K* per class,
- 2. the patch size *n*,
- 3. the RP dimensionality  $m \ (m \le n)$ .

The effect of m on classification performance is shown in Fig. 10. 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. 12 plots classification accuracy over n and m. The results are consistent with the preliminary test in Fig. 10:



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



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

For each value of *n*, the performance improves rapidly for small *m*, then leveling off for  $m \approx n/3$ . For data set  $\mathcal{D}^c$  (Fig. 12a), 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. 12b) 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 RP, it is computationally feasible to consider a greater number 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. 11 demonstrates the impact of K on the classification accuracy, showing performance increasing with K. In our comprehensive tests,



Fig. 12. Classification results as a function of patch size ( $\sqrt{n} \times \sqrt{n}$ ) and RP dimensionality m on data set (a)  $\mathcal{D}^c$  and (b)  $\mathcal{D}^C$ .



Fig. 13. Classification accuracy as a function of feature dimension m, comparing the performance of RP, PCA, and the original Patch method. All three methods use the same classifier, only the method of feature extraction varies. The number of textons used per class is K = 10. Results are reported as averages over 50 runs. (a) Results for data set  $D^C$  using a patch of size  $11 \times 11$  with Weber's law normalization. (b) Results for data set  $D^C$  using a patch of size  $15 \times 15$  with unit-norm normalization. (c) Results for data set  $D^B$  using a patch of size  $5 \times 5$  with unit-norm normalization.

TABLE 3

Classification Accuracy in Percent for the CUReT Database with a Patch Size of  $15 \times 15$ , Comparing the Proposed RP with PCA

Dim	9	20	30	40	50	60	70	80	90
RP (10)	$96.60 \pm 0.68$	$96.98 \pm 0.54$	$97.20 \pm 0.54$	$97.37 \pm 0.43$	$97.49 \pm 0.34$	$97.52 \pm 0.41$	$97.53 \pm 0.35$	$97.60 \pm 0.41$	$97.61 \pm 0.45$
PCA (10)	$96.74 \pm 0.51$	$96.81 \pm 0.69$	$96.83 \pm 0.71$	$97.01 \pm 0.36$	$97.00 \pm 0.38$	$97.02 \pm 0.40$	$96.93 \pm 0.49$	$97.03 \pm 0.33$	$97.03 \pm 0.41$
Dim	100	110	120	130	150	170	190	210	225
RP (10)	$97.65 \pm 0.45$	$97.62 \pm 0.27$	$98.59 \pm 0.25$	$97.66 \pm 0.35$	$97.65 \pm 0.27$	$97.63 \pm 0.26$	$97.65 \pm 0.26$	$97.61 \pm 0.23$	$97.61 \pm 0.21$
PCA (10)	$97.11 \pm 0.34$	$97.04 \pm 0.42$	$97.16 \pm 0.35$	$97.04 \pm 0.40$	$97.13 \pm 0.40$	$97.13 \pm 0.36$	$97.01 \pm 0.44$	$97.15 \pm 0.38$	$97.05 \pm 0.39$

The number of training images per class is 46; the number of textons K per class is 10; unit-norm normalization is used.

TABLE 4 A Comparison of Computational Complexity between RP, Patch, and Patch-MRF

Method	Feature extraction	k-means clustering	Histogramming	Classification (1NN)
RP (Proposed)	$C \cdot S \cdot nmN_I$	$C \cdot T \cdot O(mKN_0)$	$C \cdot S \cdot mCKN_I$	$2(CS_1) \cdot (CS_2) \cdot CK$
Patch	-	$C \cdot T \cdot O(nKN_0)$	$C \cdot S \cdot nCKN_I$	$2(CS_1) \cdot (CS_2) \cdot CK$
Patch-MRF	-	$C \cdot T \cdot O((n-1)KN_0)$	$C \cdot S \cdot (n-1)CKN_I$	$2(CS_1) \cdot (CS_2) \cdot GCK$

Here, T counts the number of k-means iterations (typically  $T \ge 50$ ), G denotes the number of quantized bins for the central pixel,  $S_1$  and  $S_2$  are the number of training and testing samples per class, respectively,  $N_I$  denotes the number of pixels per sample, and  $N_0$  is the number of pixels per class for learning the textons. For  $D^C$ , C = 61, S = 92,  $S_1 = S_2 = 46$ ,  $N_I = 200 \times 200$ ,  $N_0 = S_1N_I$ , G = 200.



Fig. 14. Classification results on data set  $D^C$  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 [21]. Classification rates obtained based on the same patch size are shown in the same color.

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

## 4.3.3 RP versus PCA

As PCA is one of the principal approaches to dimensionality reduction, even if not state of the art, we wish to perform an initial comparison on data sets  $D^C$  and  $D^B$ , with results shown in Fig. 13 and Table 3. At very low dimensions, the targeted approach of PCA leads to comparable or slightly improved performance; however, at peak performance, the RP approach outperforms PCA, almost certainly because the second-order statistics used by PCA fail to fully characterize the patch space.

#### 4.4 Comparative Evaluation

In this section, we compare the proposed approach specifically to the current state of the art [20], [21] on the CUReT database. To make the comparison as meaningful as possible, we use the same experimental settings as Varma and Zisserman [21].

In their comprehensive study, Varma and Zisserman [20] presented six filter banks for texton-based texture classification on  $\mathcal{D}^{C}$ . They concluded that the rotationally

#### TABLE 5 Mean (Standard Deviation) Results on $\mathcal{D}^C$ : (a) Weber's Law Normalization; (b) No Normalization; (c) Unit-Norm Normalization

(a) Weber's Law Normalization

		(	/						
Patch Size	$3 \times 3$	$5 \times 5$	7 × 7	$9 \times 9$	$11 \times 11$	$13 \times 13$	$15 \times 15$	$17 \times 17$	$19 \times 19$
Dim	9	15	25	30	40	50	60	80	90
RP (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
RP (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)
PatchMRF-VZ (Best)	96.55	97.59	98.03	97.87	97.73	97.86	97.97	97.79	97.88
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

			(b)	No Nor	malizat	ion			
Patch Size	$3 \times 3$	$5 \times 5$	$7 \times 7$	$9 \times 9$	$11 \times 11$	$13 \times 13$	$15 \times 15$	$17 \times 17$	$19 \times 19$
Dim	5	13	15	20	40	50	60	80	90
RP (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
RP (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)	96.82(0.30)	97.04(0.43)	97.00(0.41)	97.00(0.37)
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

(c) Unit Norm Normalization

Patch Size	$3 \times 3$	$5 \times 5$	$7 \times 7$	$9 \times 9$	$11 \times 11$	$13 \times 13$	$15 \times 15$	$17 \times 17$	$19 \times 19$
Dim	6	10	15	30	40	50	60	70	80
RP (10)	95.26(0.45)	96.63(0.29)	96.87(0.22)	97.01(0.31)	97.09(0.36)	97.19(0.42)	97.52(0.41)	97.59(0.41)	97.69(0.31)
Dim	6	10	20	30	40	50	60	70	80
RP (40)	95.90(0.40)	97.86(0.35)	98.41(0.30)	98.46(0.21)	98.27(0.35)	98.26(0.20)	98.44(0.25)	98.52 (0.19)	98.49(0.24)
Dim	9	25	49	81	121	169	225	289	361
Patch (10)	95.23(0.35)	96.52(0.29)	96.82(0.25)	96.90(0.22)	96.88(0.21)	97.01(0.33)	96.99(0.43)	97.00(0.30)	97.23(0.20)
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

The mean and standard deviation of the classification accuracy as a function of patch size. The bracketed values denote *K*, the number of textons used per class. The "VZ" results are quoted directly from the paper of Varma and Zisserman [21]. The "PatchMRF-VZ (Best)" shows results obtained for the best combination of texton dictionary and number of bins for the center pixel for a particular patch size. For patch sizes up to  $11 \times 11$ , texons up to 50 per class and bins up to 200 are tried. For  $13 \times 13$  and larger neighborhoods, the maximum number of textons per class is restricted to 20 because of the huge computational expense of the Patch-MRF method.

invariant, multiscale, Maximum Response MR8 filter bank yielded better results than any other filter bank. However, in their more recent study [21], 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.

We begin with an analysis of the computational costs, summarized in Table 4. Between the Patch and RP methods, it is clear that the relative complexities are determined by the relative dimensionalities m and n, respectively. In terms of the Patch-MRF model, the computational complexity of classification is greatly increased by a factor of G, the number of bins to represent the center pixel.

The best published CUReT classification performance is 98.03 percent, as reported by Varma and Zisserman [21],

 
 TABLE 6

 Comparison of Highest Classification Performance on  $\mathcal{D}^C$  with a Common Experimental Setup

Method	Affine	LBP	MR8	Patch	Patch-MRF	RP
Accuracy (%)	95.30	95.72	97.43	97.17	98.03	98.52

achieved by Patch-MRF with G = 90, K = 40, resulting in a histogram model dimensionality as high as  $GCK = 90 \times 61 \times 40$ , even higher than the  $200 \times 200$  dimensionality of the image, violating the dimensionality reduction premise of this paper and introducing substantial computational and storage complexity. Our goal is to exceed this classification performance in a much simpler, reduced-dimensionality setting.

Fig. 14 and Table 5a present a comparison of the RP classifier, the Patch classifier, the MR8 filter bank, and LBP. The Patch-VZ and MR8-VZ results are taken from Varma and Zisserman [21]; all other results are computed by us, with the results averaged over tens of random partitions of training and testing sets.

The proposed RP method outperforms all other methods; a clear indication that the RP 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 RP offer higher classification accuracy, but also at a much lower dimensional feature space, reducing storage requirements and computation time.

Fig. 14 compares the three normalizations, together with Tables 5a, 5b, and 5c. The results show that the proposed approach outperforms the Patch method in all three normalizations, and that the classification accuracy differences caused by normalization are modest.

By way of comparison, from [23] the affine adaptation method of Lazebnik et al. [22] using nearest neighbour classification achieves an accuracy of 72.5 percent. Even when multiple high-dimensional descriptors are combined with multiple detectors and an SVM classifier, the affine adaptation results improve to only 95.30 percent [23].

To summarize the preceding figures and tables, Table 6 presents the overall best classification performance achieved by each method for any parameter setting. The proposed RP method gives the highest classification accuracy of 98.52 percent, even higher than the best of



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

TABLE 7Comparisons of Three Types of Feature VectorNormalization of the Patch and Proposed RP Methods onBrodatz Large  $\mathcal{D}^B$  with K = 10 Textons per Class

$\begin{array}{ c c c c c c c c c c c c c c c c c c c$			
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	2.5(0.09) 93.4(0.14) 93.3(0.1		
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	9 25 49		
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$	1.8(1.45) 93.0(0.44) 93.7(0.2		
$3 \times 3$ $5 \times 5$ $7 \times 7$ $3 \times 3$ $5 \times 5$ $7 \times 7$ $3$	6 15 25		
	$3 \times 3$ $5 \times 5$ $7 \times 7$		
Unit Norm Normalization Weber's Law Normalization	No Normalization		

Means (standard deviations) have been computed over 50 runs.

Patch-MRF in [21], despite the fact that the model dimensionality of the Patch-MRF method is far larger than that of the proposed RP method.

### 4.4.1 Results on Other Data Sets

We wish to compare to other benchmark data sets (described in Section 4.2); however, since  $D^C$  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  $\mathcal{D}^c$ . We can observe that the proposed method performs similarly to the Patch method but at a much lower dimensionality. As was seen in Fig. 12, it is clear that the classification performance goes down as the patch size is increased, quite different from the CURET large data set  $\mathcal{D}^C$ . Nevertheless, this test shows that the proposed RP approach can be well applied in this situation without loss of performance. By comparison, from a recent LBP paper [55], the best performance for this data set is 86.84 percent for LBP, and 92.77 percent for the combination of LBP and NGF with a SVM classifier, in contrast to our RP classification accuracy of 95.85 percent.

**Brodatz large**  $\mathcal{D}^{B}$ . Tables 7, 8, and 9 show the classification accuracy for the three Brodatz data sets  $\mathcal{D}^{B}$ ,  $\mathcal{D}^{b}$ , and  $\mathcal{D}^{BFull}$ , respectively. Our results outperform published accuracies, with the exception of  $\mathcal{D}^{BFull}$ , where our result of 94.2 percent is slightly worse than the rate of 94.9 percent reported in [23] using a SIFT descriptor and Laplacian blob detector based on an SVM classifier. The classification accuracy of 99.95 percent which is achieved by RP on  $\mathcal{D}^{b}$  is so good as to leave hardly any room for improvement. Recent published performance on  $\mathcal{D}^{b}$  in [55] was 98.49 percent for LBP alone and 99.54 percent for the combination of LBP and NGF with an SVM classifier.

**MSRC**  $\mathcal{D}^{M}$ . Table 10 shows the classification performance of the RP and the Patch methods on data set  $\mathcal{D}^{M}$ . Excellent results (as high as 99.57 percent) are obtained using the proposed RP approach, exceeding the published Patch results, reinforcing the broad applicability of RP.

Finally, a word about feature vector normalization. Based on all the results presented in this paper, we can

TABLE 9As in Table 7, with Mean (Standard Deviation) ResultsAveraged over 50 Runs on the Full Brodatz Data Set  $\mathcal{D}^{BFull}$ ,for K = 10, 40

	Unit No	orm Norm	alization	Weber's Law Normalization			No Normalization		
	$5 \times 5$	7 × 7	$9 \times 9$	$5 \times 5$	$7 \times 7$	$9 \times 9$	$5 \times 5$	$7 \times 7$	$9 \times 9$
Dim	10	20	30	10	20	30	10	20	30
RP (10)	92.5(0.71	93.0(0.98)	92.6(0.63)	89.1(1.41)	90.1(0.88)	90.8(1.26)	86.2(0.71)	87.4(0.64)	87.8(0.66)
RP (40)	93.6(0.55)	94.1(0.56)	94.2(0.49)	90.3(0.54)	91.5(0.70)	91.9(0.70)	87.6(0.55)	88.9(0.61)	89.4(0.68)
Dim	25	49	81	25	49	81	25	49	81
Patch (10)	91.8(0.84)	91.8(0.98)	90.9(0.89)	87.7(0.81)	88.5(0.98)	88.2(0.83)	84.9(1.02)	86.0(1.05)	86.6(0.99)

TABLE 10Results on the MSRC Database  $\mathcal{D}^M$ ,with Mean (Standard Deviation) Results Averaged over1,000 Random Partitionings of the Training and Testing Sets

	Web	er's Law Normaliza	tion	No	Normaliza	tion
1 1	$3 \times 3$	5×5	7 × 7	$3 \times 3$	$5 \times 5$	$7 \times 7$
Dim	6	10	20	6	10	20
RP (10)	99.14(0.75)	99.27(0.58)	98.86(0.78)	98.93(0.83)	98.44(0.88)	98.08(0.99)
RP (20)	99.30(0.59)	99.43(0.59)	98.92(0.78)	98.85(0.85)	98.87(0.78)	98.73(0.84)
RP (30)	99.37 (0.56)	99.30(0.65)	99.06(0.77)	98.93(0.82)	99.16(0.71)	98.90(0.83)
RP (40)	99.33(0.49)	99.53(0.57)	99.15(0.69)	99.05(0.71)	99.10(0.73)	99.04(0.77)
Dim	9	25	49	9	25	49
Patch (10)	98.60(0.85) 96.82(\$)	98.48(0.93) 99.21(◊)	96.46(1.23) 96.03(\$)	97.93(0.96)	95.70(1.45)	91.30(2.15)
Patch (20)	98.76(0.81) 96.82(◊)	$98.84(0.81)$ $99.21(\diamond)$	97.77(1.01) 97.62(o)	98.51(0.87)	97.88(0.90)	94.70(1.83)
Patch (30)	98.90(0.88) 96.82(\$)	98.92(0.80) 99.21(o)	98.15(0.95) 96.82(\$)	98.50(0.84)	98.17(0.85)	96.31(1.31)
Patch (40)	98.81(0.86) 96.82(\$)	99.01(0.82) 99.11(\$)	98.68(0.82) 97.62(\$)	98.65(0.84)	98.47(0.81)	97.26(1.06)
culto m	arkad by (a)	ara takan	from the rea	ont cti	dy of	Varma

Results marked by  $(\diamond)$  are taken from the recent study of Varma and Zisserman [21].

observe that the unit norm and Weber's Law normalizations perform equally well, with the former slightly better than the latter. However, they both outperform the no normalization case except for  $\mathcal{D}^c$ . This may be partially because all images in  $\mathcal{D}^c$  has the same controlled illumination condition, while other data sets have illumination variations (see Table 1). Nonuniform illumination can give rise to local texture appearance change, and feature vector normalization can enhance the intensity invariance and leads to better classification results.

## 5 CONCLUSIONS

In this paper, we have described a classification method based on representing textures as a small set of compressed, random measurements of local texture patches, leading to results matching or surpassing 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 RP 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 RP approach enjoys the advantage of the Patch method in achieving high classification

TABLE 8											
As in Table 7, with Results	Averaged over 100 Runs	s on the Small Brodatz Data Set ${\cal D}$									

	Unit Norm Normalization		Weber's Law Normalization			No Normalization			
	$3 \times 3 (\%)$	$5 \times 5$	$7 \times 7$	$3 \times 3$	$5 \times 5$	$7 \times 7$	$3 \times 3$	$5 \times 5$	$7 \times 7$
Dim	7	10	15	5	7	7	7	10	15
RP (10)	$99.93 \pm 0.13$	$99.93 \pm 0.13$	$99.93 \pm 0.10$	$99.95 \pm 0.05$	$99.90\pm0.06$	$99.85 \pm 0.20$	$99.88 \pm 0.16$	$99.85 \pm 0.20$	$99.85 \pm 0.18$
Dim	9	25	49	9	25	49	9	25	49
Patch (10)	$99.93 \pm 0.15$	$99.91 \pm 0.15$	$99.92 \pm 0.12$	$99.90 \pm 0.09$	$99.84 \pm 0.13$	$99.80 \pm 0.03$	$99.67 \pm 0.09$	$99.76 \pm 0.11$	$99.72 \pm 0.08$

performance and that of the preselected filter banks in its low-dimensional feature space.

The random features assume no prior information about the texture image, except the assumption of sparsity in some (overcomplete) basis, 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 RP-based texture classification. First, the use of a more sophisticated classifier, like SVM, may provide enhanced classification performance over the nearest neighbor classifier used here. Furthermore, the proposed approach can 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  $histograms/\chi^2$  distance framework [22], [23].

## **ACKNOWLEDGMENTS**

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. Information Theory, vol. 8, pp. 84-92, 1962.
- M. Tuceryan and A.K. Jain, "Texture Analysis," Handbook [2] Pattern Recognition and Computer Vision, C.H. Chen, L.F. Pau, and P.S.P. Wang, eds., ch. 2, pp. 235-276, World Scientific, 1993.
- 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, Apr. 1999.
- J. Zhang and T. Tan, "Brief Review of Invariant Texture Analysis [4]
- Methods," Pattern Recognition, vol. 35, no. 3, pp. 735-747, 2002. R.M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. Systems, Man, and Cyber-netics*, vol. 3, no. 6, pp. 610-621, Nov. 1973. [5]
- J.S. Weszka, C.R. Dyer, and A. Rosenfeld, "A Comparative Study [6] of Texture Measures for Terrain Classification," IEEE Trans. Systems, Man, and Cybernetics, vol. 6 no. 4, pp. 269-285, Apr. 1976.
- G.R. Cross and A.K. Jain, "Markov Random Field Texture [7] Models," IEEE Trans. Pattern Analysis and Machine Intelligence,
- vol. 5, no. 1, pp. 25-39, Jan. 1983. S.C. Zhu, Y. Wu, and D. Mumfors, "Filters, Random Fields and [8] Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling," Int'l J. Computer Vision, vol. 27, no. 2, pp. 107-126, 1998.
- J. Mao and A.K. Jain, "Texture Classification and Segmentation [9] Using Multiresolution Simultaneous Autoregressive Models," Pattern Recognition, vol. 25, no. 2, pp. 173-188, 1992.
- [10] L.M. Kaplan, "Extend Fractal Analysis for Texture Classification and Segmentation," IEEE Trans. Image Processing, vol. 8, no. 11, pp. 1572-1585, Nov. 1999.
- [11] 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, Jan. 1990.
- [12] 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, Aug. 1996.
- [13] A.J. Heeger and J.R. Bergen, "Pyramid-Based Texture Analysis/
- Synthesis," Proc. ACM Siggraph, pp. 229-238, 1995.
  [14] T. Chang and C.-C. Kuo, "Texture Analysis and Classification with Tree-Structured Wavelet Transform," IEEE Trans. Image
- Processing, vol. 2, no. 4, pp. 429-441, Oct. 1993.
  [15] X. Qin and Y.H. Yang, "Basic Gray Level Aura Matrices: Theory and Its Application to Texture Synthesis," Proc. IEEE Int'l Conf. Computer Vision, vol. 1, pp. 128-135, 2005.

- [16] B. Olshausen and D. Field, "Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1?" Vision Research, vol. 37, pp. 3311-3325, 1997.
- [17] 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.
- [18] T. Leung and J. Malik, "Representing and Recognizing the Visual Appearance of Materials Using Three-Dimensional Textons," Int'l J. Computer Vision, vol. 43, no. 1, pp. 29-44, 2001. [19] O.G. Cula and K.J. Dana, "3D Texture Recognition Using
- Bidirectional Feature Histograms," Int'l J. Computer Vision, vol. 59, no. 1, pp. 33-60, 2004.
- [20] M. Varma and A. Zisserman, "A Statistical Approach to Texture Classification from Single Images," Int'l J. Computer Vision, vol. 62, nos. 1/2, pp. 61-81, 2005.
- [21] 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, Nov. 2009.
- S. Lazebnik, C. Schmid, and J. Ponce, "A Sparse Texture Representation Using Local Affine Regions," *IEEE Trans. Pattern* [22] Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1265-1278, Aug. 2005.
- [23] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study," Int'l J. Computer Vision, vol. 73, no. 2, pp. 213-238, 2007.
- [24] E.J. Candès and T. Tao, "Decoding by Linear Programming" IEEE Trans. Information Theory, vol. 51, no. 12, pp. 4203-4215, Dec. 2005.
- [25] E.J. Candès and T. Tao, "Near-Optimal Signal Recovery from Random Projections: Universal Encoding Stratigies?" IEEE Trans. Information Theory, vol. 52, no. 12, pp. 5406-5425, Dec. 2006.
- [26] D.L. Donoho, "Compressed Sensing," IEEE Trans. Information Theory, vol. 52, no. 4, pp. 1289-1306, Apr. 2006.
- [27] G. Biau, L. Devroye, and G. Lugosi, "On the Performance of Clustering in Hilbert Spaces," *IEEE Trans. Information Theory*, vol. 54, no. 2, pp. 781-790, Feb. 2008.
- T. Linder, "Learning Theoretic Methods in Vector Quantization," [28] Principles of Nonparametric Learning, Springer, July 2001.
- [29] J.A. Tropp, "Topics in Sparse Approximation," PhD dissertation, Univ. of Texas at Austin, 2004.
- [30] M. Aharon, M. Elad, and A.M. Bruckstein, "The K-SVD: An Algorithm for Designing of Overcomplete Dictionaries for Sparse Representations," IEEE Trans. Signal Processing, vol. 54, no. 11, pp. 4311-4322, Nov. 2006.
- [31] S. Lazebnik and M. Raginsky, "Supervised Learning of Quanitizier Codebooks by Information Loss Minimization," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31, no. 7, pp. 1294-1309, July 2009.
- [32] D. Pollard, "Quanitization and the Method of k-Means," IEEE Trans. Information Theory, vol. 28, no. 2, pp. 199-205, Mar. 1982.
- [33] N. Goel, G. Bebis, and A. Nefian, "Face Recognition Experiments with Random Projection," Proc. SPIE, vol. 5779, pp. 426-437, 2005.
- [34] W.B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz Mappings into a Hilbert Space," Proc. Conf. Modern Analysis and Probability, pp. 189-206, 1984.
- S. Dasgupta and A. Gupta, "An Elementary Proof of a Theorem of Johnson and Lindenstrauss," *Random Structures and Algorithms*, [35] vol. 22, no. 1, pp. 60-65, 2003.
- [36] M. Davenport, P. Boufounos, M. Wakin, and R. Baraniuk, "Signal Processing with Compressive Measurements," IEEE J. Selected Topics in Signal Processing, vol. 4, no. 2, pp. 445-460, Apr. 2010.
- [37] D. Achlioptas, "Database-Friendly Random Projections," Proc. 20th ACM Symp. Principles of Database Systems, pp. 274-281, 2001.
- E. Bingham and H. Mannila, "Random Projection in Dimention-[38] ality Reduction: Applications to Image and Text Data," Proc. Seventh ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 245-250, 2001.
- [39] S. Dasgupta, "Experiments with Random Projections," Proc. 16th Conf. Uncertainty in Artificial Intelligence, pp. 143-151, 2000.
- [40] M.F. Duarte, M.A. Davenport, M.B. Wakin, J.N. Laskar, "Multiscale Random Projection for Compressive Classification," Proc. Int'l Conf. Image Processing, 2007.
- [41] H. Rauhut, K. Schnass, and P. Vandergheynst, "Compressed Sensing and Redundant Dictionaries," *IEEE Trans. Information Theory*, vol. 54, no. 5, pp. 2210-2219, May 2008.

- [42] R.G. Baraniuk, M. Davenport, R.A. DeVore, and M. Wakin, "A Simple Proof of the Restricted Isometry Property for Random Matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253-263, 2008.
- [43] X.Z. Fern and C.E. Brodley, "Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach," Proc. 20th Int'l Conf. Machine Learning, 2003.
- [44] G. Peyré, "Sparse Modeling of Textures," J. Math. Imaging and Vision, vol. 34, no. 1, pp. 17-31, 2009.
  [45] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman,
- [45] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative Learned Dictionaries for Local Image Analysis," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.
- [46] K. Skretting and J.H. Husøy, "Texture Classification Using Sparse Frame-Based Representations," EURASIP J. Applied Signal Processing, vol. 1, pp. 102-102, 2006.
- [47] 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, pp. 1395-1408, July 2009.
- [48] S. Dasgupta and Y. Freund, "Random Projection Trees for Vector Quantization," *IEEE Trans. Information Theory*, vol. 55, no. 7, pp. 3229-3242, July 2009.
- [49] 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, Feb. 2009.
- [50] R.M. Gray and D.L. Neuhoff, "Quantization," IEEE Trans. Information Theory, vol. 44, no. 6, pp. 2325-2383, Oct. 1998.
- [51] 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.
- [52] J. Haupt, R. Castro, R. Nowak, G. Fudge, and A. Yeh, "Compressive Sampling for Signal Classification," Proc. Asilomar Conf. Signals, Systems and Computers, pp. 1430-1434, 2006.
- [53] E. Levina, "Statistical Issues in Texture Analysis," PhD thesis, Univ. of California, Berkeley, 2002.
- [54] S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, vol. 290, pp. 2323-2326, 2000.
- [55] S. Liao, M.W.K. Law, and A.C.S. Chung, "Dominant Local Binary Patterns for Texture Classification," *IEEE Trans. Image Processing*, vol. 18, no. 5, pp. 1107-1118, May 2009.
- [56] 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.
- [57] T. Mäenpää and M. Pietikäinen, "Multi-Scale Binary Patterns for Texture Analysis," Proc. Scandinavian Conf. Image Analysis, 2003.
- [58] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. Dover Publications, 1966.
- [59] S. Mallat, A Wavelet Tour of Signal Processing: The Sparse Way, third ed. Academic Press, Dec. 2008.
- [60] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, "On the Significance of Real-World Conditions for Material Classification," *Proc. European Conf. Computer Vision*, pp. 253-266, 2004.
- [61] S. Graf and H. Luschgy, *Foundations of Quantization for Probability Distributions*. Springer-Verlag, 2000.



Li Liu received the BS degree in communication engineering and the MS 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 working toward the PhD degree. During her PhD study, she spent two years and three months as a visiting student at the University of Waterloo, Canada. Her current research

interests include computer vision, texture analysis, pattern recognition, and image processing.



**Paul W. Fieguth** received the BASc degree from the University of Waterloo, Ontario, Canada, in 1991, and the PhD degree from MIT, Cambridge, in 1995, both degrees in electrical engineering. He joined the faculty at the University of Waterloo in 1996, where he is currently a professor and department chair in systems design engineering. He has held visiting appointments at the University of Heidelberg in Germany, at INRIA/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. He is a member of the IEEE.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.