Texture classification using local discriminative features and Fisher encoding

Rakesh Mehta
Dept. of Signal Processing
Tampere University of Technology
Tampere, Finland
Email: rakesh.mehta@tut.fi

Karen Egiazarian
Dept of Signal Processing
Tampere University of Technology
Tampere, Finland
Email: karen.egiazarian@tut.fi

Abstract—In this paper we introduce a new image representation for texture classification. Our work is motivated by recent developments in the field of local patch based features, compressive sensing and descriptor encoding methods. Novel features called Compressed Random Pixel Difference (CRPD) are proposed. These features are low in dimensionality, highly discriminative, and easy to compute. Combined with an efficient encoding method, an expressive and robust image descriptor is obtained. Experiments conducted on widely used texture datasets (KTH-TIPS-2a and Brodatz) demonstrate an efficiency of the proposed approach. On KTH-TIPS-2a dataset we have achieved the highest recognition accuracy (to the best of our knowledge) and on Brodatz dataset achieved performance is comparable to the state-of-the-art methods.

I. INTRODUCTION

Texture classification is an important topic in the field of computer vision and pattern recognition. Texture is one of the basic vision cues and it has been utilized in a number of high level computer vision applications such as face recognition [1], object detection [2], facial expression recognition [3], texture segmentation [4], etc. A number of methods have been proposed for texture recognition, however it still remains an open field of study because of the problems such as, the variations in scale, illuminations, rotation and the subtle difference in the different texture patterns.

Earlier methods for texture classification were based on filter banks [5] and co-occurrence statistics [6]. In last one decade a number of approaches have been proposed that use SIFT [7] and HOG [8] features, in principle they extract the image gradients in sparse or dense manner and then compute certain statistics from it to represent the image. Lazebnik et. al. [9] demonstrated that the texture images are sparse in nature and extracted feature from the affine invariant keypoints. Ojala et. al. [10] introduced Local Binary Patterns (LBP) and showed it is possible to capture the discriminative information from small patches of size $3 \times 3$. Varma et. al. proposed MR8 features [11] based on wavelet decomposition and multichannel filtering and further consolidated the fact that small patches can capture discriminative information. Recently, Sharma et. al. [12] used the higher order statistics of the LBP like features from the local neighbourhood and reported state-of-the-art recognition rate on certain texture datasets.

The pipeline for robust texture classification approach consists of the three main steps: (1) Extract the features from the texture images (e.g. SIFT, LBP type features), (2) Encode the features into an image descriptor (e.g. Histogram, Bag of Words (BoW) [13]) (3) Classify the image descriptor using a machine learning algorithm (e.g. Nearest Neighbour, Support Vector Machines (SVM) [14]). In this paper we focus on the first two step of this pipeline. For the first step of feature extraction, we introduce novel features CRPD which have advantage of being discriminative and low dimensional. These are also very fast to compute with only few operation per patch. In the encoding step, we apply efficient Fisher Vectors [15], as these capture the higher order information from the features. These too are fast to compute and result in a robust image descriptor. Finally, the image descriptor is combined with linear kernel SVM for the task of texture classification. Experiments performed on the standard texture datasets demonstrate that the proposed approach outperforms a number of state-of-the-art methods.

Rest of the paper is organized as follows: In Section 2, we discuss the desired property required for a robust texture descriptor and give a brief overview of our design. Section 3 provides the details of the local features CRPD and the encod-
ing technique used in the proposed pipeline. The experiments and the implementation details are provided in Section 4 and finally the paper is concluded in Section 5.

II. DESIRED PROPERTIES AND DESIGN OVERVIEW

In this section we study the desired properties for a texture descriptor followed by the overview of proposed approach:

*Patch based representation:* It has been demonstrated that highly discriminative information can be extracted from small patches of the texture images [11], [10]. State-of-the-art performance has been reported by features captured from the patch of size as small as 3x3 [11]. These approaches exhibit that texture is intrinsically a local visual cue rather than a global one. Hence it is more reasonable to have features that are captured from the image patches or local neighbourhood than the whole image. The size of the patch is an important criterion for these features, it should be large enough to encompass that inherent characteristic of the texture. However, as the size of the patch increases the dimensionality of the features also increases. Thus, it is also important to maintain the balance between the dimensionality of the patch based feature and the size of patch.

*Utilizing Compressibility or Sparseness:* The recent advancement in the image processing and computer vision field has proved, by the means of wavelets, that the images are sparse in nature. The sparse and compressible nature of the image has been utilized for image denoising [16], face recognition [17], object recognition [18], etc and it has shown very promising results. Texture images being repetitive are even more sparse and ‘stationary’. Texture descriptor with dimensionality as small as 9 has shown good performance which further demonstrates this fact [10]. The sparse nature of the images has already been utilized in some of the texture recognition approaches [19], [9]. We believe by taking advantage of the sparse nature of the texture image compressed yet efficient features can be designed.

*Local characteristic structure:* A number of studies [10], [20], [21] have suggested that the local structure of the texture exhibit a certain kind of repetitive order. This order is confirmed by the high occurrence of distinct patterns in specific texture images. A number of texture descriptors have been designed based on this principle. The basic idea behind these is to define a dictionary of the patterns, then for each image compute the occurrence of these patterns. Based on the distribution of the occurrences in a test image it is classified to a specific texture class. Various approaches have been used for defining the dictionary. For examples, LBP defines the dictionary based on the sign of the difference of pixels in circular neighbourhood, Basic Image Features (BIF) [21] explicitly designed 7 features, where each feature correspond to a specific structure.

*Efficient representation of the features:* Once the features are extracted (e.g. LBP, SIFT, HOG, etc) from the image, these have to be encoded to form a single image descriptor. The encoding involves computation of certain statistics of the feature, that would represents the generic structure of the image. The simplest and the most widely used approach is to compute the histogram of these local features [22]. Histograms compute the zero order statistics of these features. It has been demonstrated that incorporating higher order information results in a more expressive representation and it has been utilized for object recognition [15]. We also agree with this observation and believe that capturing higher order statistics from the local features would lead to a better representation of texture image.

Taking motivation from the above discussed ideas, we design a texture descriptor. The architecture of the proposed texture descriptor is shown in the Fig. 1 and it consists of the following steps:

1) Capture the local structure of the texture from patches using Random Pixel Difference (RPD) features.
2) Use the Random Projections to compress the RPD features captured in the first step and obtain CRPD features.
3) Model the distribution of the compressed local features (CRPD) using the Fisher Vectors and generate the image descriptor.
4) Normalize the image descriptor and classify the images using SVM with linear kernel.

Further in this paper these steps are described in details.

III. IMAGE REPRESENTATION FOR TEXTURE CLASSIFICATION

The main steps of the image representation are described in this section. First, we introduce the local features RPD, then, we present the scheme to compress these local descriptor by utilizing their sparse characteristics. Finally, the Fisher Vectors technique is discussed to encode the compressed local features.

A. Random Pixel Difference (RPD)

Earlier work [10], [23] on texture classification has demonstrated, that, the difference of the pixels intensity values from an image patch can provide discriminative information. However, these approaches binarize or threshold the difference based on its sign. Some researchers have argued that the magnitude of the difference, which is completely ignored in above mentioned approaches, also provide discriminative information about the patch. They showed [12], [24], [23] that by incorporating the information about the magnitude the performance of these features can be improved.

In this section we present novel, simple yet effective local features Random Pixel Difference (RPD) which are very fast.
to compute. It is designed on the idea that difference of pixel intensity provide expressive information. However, unlike the above mentioned approaches [10], [23] the pixel difference is not thresholded into binary values. The RPD features are computed by taking the difference of intensity between pixel pairs. The position or the coordinates of these pixels is fixed at the beginning of the feature extraction process. Therefore, for each patch a vector of numbers is obtained.

Formally, given a patch \( g \) from image and two sets of coordinates \( X \in \{x_1, x_2, ..., x_N\} \), \( Y \in \{y_1, y_2, ..., y_N\} \), which denote the position of the pixel pairs in the patch, the RPD feature vector is defined as:

\[
v(g) = [g(x_1) - g(y_1), g(x_2) - g(y_2), ..., g(x_N) - g(y_N)]^T \tag{1}
\]

where \( g(x_1) \) is the intensity of the pixel at the coordinate, \( x_1 = (a, b)^T \), in the patch, the size of the patch is denoted by \( L \times L \) and \( N \) is the number of pixel pairs considered in the patch. The RPD features are completely specified by the following parameters: \( X, Y, N \), and \( L \). The selection of the coordinates pair \( (X, Y) \) is an important factor in the design of RPD. Colander et. al. [25] studied the different spatial arrangement for selecting the points in the patch for key point matching. Following the results from [25], we follow the similar configuration and select the pixel coordinates from isotropic Gaussian distribution, i.e. \( (X, Y) \sim i. i. d. Gaussian(0, L^2/25) \). Fig. 2 shows an image patch and the coordinates used for computing the difference of the pixels. In this examples the patch is of size \( 25 \times 25 \) and the number of sampling points, \( N \), is 32. It can be observed that the coordinates are densely distributed around the center of the patch. Thus, more weight is given to the center of the patch then to its boundaries. The RPD features measure the difference in the pixel intensity from the center towards the patch boundaries. The pixel difference is computed at different scales, as the distance between the coordinates, \(|y_i - x_i|\) is not constant. The size of the patch and the number of sampling points \( N \) are determined experimentally and discussed in the implementation details section.

The RPD features have certain advantages over other features. These are very fast and easy to compute (for each patch only \( N \) subtraction operations are required). Since we are dealing with integers and no multiplication is required, it can be implemented very efficiently on various architecture. The features are invariant to monotonic grey level changes, as these are computed by taking the difference of the pixel values. These incorporate information from multiple scale because the distance between the points \(|y_i - x_i|\) is not fixed, unlike other binary features like [10], [23] and [12].

### B. Compressed RPD (CRPD)

The RPD features described above provide an \( N \)-dimensional representation for a patch of size \( L \times L \). For texture recognition, the patch size is usually in range of 3x3 to 15x15 and the number of sampling points, \( N \), required to capture the structure is 32 or more. It results in a dimensionality of 32 or more for a small patch. This dimensionality is significantly high, considering the fact that texture images are sparse and compressible in nature. The number of sampling points, \( N \), cannot be reduced, as these many points are necessary to capture the structural pattern from the patches. To reduce the dimensionality of the vector \( v(g) \) and to make it more compressed, we utilize the Random Projections (RP) [26]. The RP exhibit important properties of dimensionality reduction and information preservation. It is based on the idea that if the signal lies in a low dimensional manifold and is represented in a high dimensional ambient space, then, a small number of random projections of that signal preserve most of the information from it. Fig. 3 shows CRPD descriptor computation for a single patch from a texture images. First, a patch is selected from the image and the RPD feature are extracted using the set of coordinate pairs. Then, the RPD vector is projected using a random matrix to obtain CRPD feature vector. The projection is obtained by multiplying the RPD vector with the random matrix.

Given the RPD feature vector \( v(g) \), its random projection is defined as:

\[
d(g) = \Phi v(g) \tag{2}
\]

where \( \Phi \) is a \( C \times N \) matrix, with \( C \ll N \) and \( d(g) \) is the compressed representation of the patch \( g \). With \( C \ll N \) a loss in information is expected, however, if the signal
is sparse and the matrix $\Phi$ exhibit the Restrictive Isometric Property (RIP) then the information is shown to be preserved during this transformation [27]. A number of matrices have shown to exhibit RIP property with high probability [27]. We use Gaussian random matrix as the $\Phi$ and more details about it are provided in the implementation section.

C. Fisher Vector based representation

The next step in the classification pipeline is to encode the local patch based features (CRPD) into image descriptors. We use the Fisher Vector [15] as the encoding method. The Fisher Vector use the Gaussian Mixture Models (GMM) to derive a probabilistic representation of the CRPD. The encoding captures the first and the second order differences between the image descriptors and the GMM centres. The higher order statistics that are learnt, provide a robust representation compared to other encoding methods such as histograms and kernel codebook.

The encoding starts by learning a GMM model for the CRPD features vectors. For the CRPD vectors the model is represented as:

$$p(\mathbf{d}|\theta) = \sum_{k=1}^{K} p(\mathbf{d}|\mu_k, \Sigma_k) \pi_k$$

where, $p(\mathbf{d}|\mu_k, \Sigma_k)$ is the multivariate Gaussian distribution with mean, $\mu_k$, and covariance matrix, $\Sigma_k$, (assumed to be diagonal). $\pi_k$ is the mixing coefficient of the Gaussian component and $\theta = (\pi_1, \mu_1, \Sigma_1; \pi_2, \mu_2, \Sigma_2; \cdots; \pi_K, \mu_K, \Sigma_K)$ is the vector of the parameters for the model, $K$ is the total number of Gaussian component assumed to be present while modelling the feature distribution. The parameter of the GMM are learned using Expectation Maximization (EM) using the CRDF features from the training samples.

Given the model, Fisher Vector is characterized by the gradient with respect to the parameter of the models. Thus, the gradient is computed with respect to the mean $\mu_k$ and the covariance $\Sigma_k$ of the GMM. It is given as:

$$\frac{\partial \log p(\mathbf{d}|\theta)}{\partial \mu_k} = h_k \Sigma_k^{-1} (\mathbf{d} - \mu_k),$$

$$\frac{\partial \log p(\mathbf{d}|\theta)}{\partial \Sigma_k^{-1}} = \frac{h_k}{2} (\Sigma_k - (\mathbf{d} - \mu_k)^2),$$

where,

$$h_k = \frac{\pi_k p(\mathbf{d}|\mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} p(\mathbf{d}|\mu_{k'}, \Sigma_{k'})}.$$

The Fisher encoding is obtained by concatenating the parametric gradient for all the K components of the GMM. Thus, the length of the feature vector is $2KC$, where $C$ is the dimensionality of the CRPD. After concatenation, we apply $l_2$ and power normalization [15] on the feature vectors. The $l_2$ normalization helps in compensating for the fact that different images contain different amount of relevant information. The power normalization ($z \leftarrow \text{sign}(z)|z|^\rho$) helps to ‘unsparsify’ the feature vector that becomes sparse when the number of Gaussian components in GMM are increased.

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid Spacing</td>
<td></td>
<td>3 pixels</td>
</tr>
<tr>
<td>Patch Size</td>
<td>LxL</td>
<td>7x7</td>
</tr>
<tr>
<td>Number of points</td>
<td>N</td>
<td>32</td>
</tr>
<tr>
<td>Compressed dimension</td>
<td>C</td>
<td>10</td>
</tr>
<tr>
<td>Gaussian Components</td>
<td>K</td>
<td>32</td>
</tr>
<tr>
<td>Power Normalization Factor</td>
<td>$\rho$</td>
<td>0.5</td>
</tr>
</tbody>
</table>

IV. Experiments

The proposed method is tested on two standard publicly available texture datasets: KTH-TIPS-2a and Brodatz. These datasets include illumination, rotation, scale and viewpoints variations. The results are compared with the baseline approaches such as LBP [10], LTP [23] and the state-of-the-art algorithms such as Local Higher Order Static (LHS) [12], Weber Law Descriptor (WLD) [28] and Local Quantized Patterns (LQP) [29]. First we provide the implementation details of our algorithms and then results on both these datasets are presented.

Implementation Details: The RPD features are extracted from the grid with a spacing of 3 pixels. It is observed that the performance of the features is maintained as long as the size of the grid does not exceed 5 pixel. With a larger grid size, the local structure is not captured efficiently and for denser grid spacing the number of features becomes too large with no significant increase in performance. The patch size is set to $7 \times 7$ in all our experiments. The size is determined by varying the size from $3 \times 3$ to $15 \times 15$. With too small patches (e.g. $3 \times 3$) the spatial structure cannot be fully captured, as it restricts the scale variation to smaller values. As the patch size is increased the pixel difference can be computed at variable scales and thus more information can be captured from the patch. However, as the size of the patch becomes too large (e.g. $10 \times 10$) the number of scales becomes too large and the pixel difference per scale is not enough to capture the structural patterns. The number of the points for pixel difference per patch is fixed to 32. It results in a 32 dimensional RPD vector for a patch, it is still significantly lower compared to the SIFT features which for a patch has a dimensionality of 128. Compared to other features it is very fast to compute, as it only requires 32 subtraction operations per patch.

The matrix $\Phi$ is a Gaussian random matrix that is normalized to zero mean and unit variance. Its is of dimension $C \times N$, where $N$ is the dimension of the RPD vector while the $C$ is the dimension of the CRPD. The values of $C$ is set to 10 in our experiments. The GMM parameters are estimated using 500,000 CRPD vectors that are randomly sampled from the training images. The center for GMM are initialized with k-mean clustering. The number of components $K$ is set to 128 following the results from the [12].

The parameter $\rho$ is set to 0.5 for the power normalization. In all our experiments SVM classifier with linear kernels is used. The linear SVM requires less training time over types of kernels, during the testing it only requires a simple dot product. Another advantage of linear kernel is that they directly operate
Fig. 4. Sample image from datasets (a) KTH-TIPS-2a (b) Brodatz

on the feature, thus any improvement in the classification performance can be attributed to the features rather than the classifier. The summary of all the parameters involved in the algorithm is provided in Table I.

A. KTH-TIPS-2a dataset

The KTH-TIPS2-a texture dataset [32] contains 11 texture classes (e.g. cork, wool, linen, etc) with 4,395 images. The images are $256 \times 256$ pixels in size, and they are transformed into 256 gray levels. Each texture class consists of images from four different samples. The images for each sample are taken at nine scales, under four different illumination directions, and three different poses. The variations in scales, illumination and pose makes it a challenging dataset.

We use the standard testing protocol [28], [32] where at each run the three sample sets are used for training and fourth samples images for testing. The results are reported as the mean over the four test runs. The results are compared with the LBP, LTP, LQP, WLD, Caputo et. al [32] and LHS. The LBP and LTP are computed by the binary and ternary thresholding of the pixel difference in the local circular neighbourhood and use histogram as the encoding method. LQP is also a pattern based descriptor, however the number of patterns sampled are very large, which are quantized using k-mean clustering. LHS performs the Fisher encoding of the pixel difference with LBP like geometry. WLD captures the local the pattern from the image gradient images. The results of all these approaches on the KTH-TIPS-2a dataset are shown in Table II.

It can be observed from the results that the proposed CRPD-Fisher combination achieves the best results on this dataset. This is the highest accuracy reported on the KTH-TIPS-2a dataset to the best of our knowledge. It is interesting to note that the accuracy is still far from perfect even for the best results. The first reason being that the variation in this dataset are much stronger than other texture datasets such as Brodatz, etc, for which near perfect accuracy can be achieved. Another reason for lower accuracy on this dataset is the testing protocol for this dataset. Since the three samples are used for training and the fourth sample for testing, there is a considerable difference between the training and the testing images. The images from the four samples for a texture class are shown in Fig 4. It can be seen that there is a significant difference in the images. Thus, to perform on this dataset the algorithm should have a generalization property. The high recognition rate of the proposed algorithm shows that it also has a generalization property and can easily adapt the variation during training and testing.

The comparison of the accuracies show that the LBP and LTP are inferior to the state-of-the-art descriptor LHS. LTP achieves higher accuracy than LBP because it has three quantization levels compared to the two levels of LBP. Since LHS has even more quantization levels than LTP, there is a further increase in the performance from LTP to LHS. Therefore, we can infer that with more quantization levels the pixel difference is modelled in a better way, hence an improvement in performance is observed. Although CRPD and LHS both have same number of quantization levels, the gain of CRPD over LHS can be attributed to the fact that CRPD captures the information from the patches at the multiple scales rather than the single scale that is used in LHS. Also the compressed vectors of CRPD, by means of random projection, capture the inherent structure of the patch in an effective way. The proposed method outperforms the LBP, LTP by 5.8%, 4.3% respectively. Compared to state-of-the-art descriptors, WLD, Caputo et. al. [32] and LHS the proposed approach shows a significant improvement of 8.9%, and 4.6% and 2.6% respectively.

Since the main difference between the LBP/LTP and LHS is that of the feature encoding and quantization, we can conclude that the Fisher based encoding is more efficient that the histogram based encoding and that the discrete quantization results in the loss of the information.

B. Brodatz dataset

The original Brodatz dataset [30], [31] has 32 texture classes with 16 images per class. The images are of dimension 64x64. To make the test more challenging, three samples are generated from each image by (1) rotating (2) scaling and (3) both rotating and scaling the original images. The resulting

<table>
<thead>
<tr>
<th>Method</th>
<th>Recognition Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WLD</td>
<td>64.7</td>
</tr>
<tr>
<td>LQP</td>
<td>64.2</td>
</tr>
<tr>
<td>LBP</td>
<td>69.8</td>
</tr>
<tr>
<td>LTP</td>
<td>71.3</td>
</tr>
<tr>
<td>Caputo et. al [32]</td>
<td>71.0</td>
</tr>
<tr>
<td>LHS</td>
<td>73.0</td>
</tr>
<tr>
<td>CRPD-Fisher</td>
<td>75.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Recognition Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBP</td>
<td>87.2</td>
</tr>
<tr>
<td>LTP</td>
<td>95.0</td>
</tr>
<tr>
<td>LQP</td>
<td>96.9</td>
</tr>
<tr>
<td>WLD</td>
<td>96.5</td>
</tr>
<tr>
<td>Urbach et. al. [33]</td>
<td>97.5</td>
</tr>
<tr>
<td>LHS</td>
<td>99.3</td>
</tr>
<tr>
<td>CRPD-Fisher</td>
<td>99.4</td>
</tr>
</tbody>
</table>
images are resized to \(64 \times 64\) pixels, converted to grayscale and histogram normalized. Therefore, the final test set-up consists of 2048 images with 64 images in each class. Following the usual protocol in our experiment [28], we randomly select 32 images from each class for training and rest are used for testing. The accuracy is reported on 5 fold cross validation.

The methods used for comparison are LBP, LTP, LQP, WLD, Urbach et al. [33] and LHS. The results for all the approaches are shown in Table III. Again it can be observed that the simple pattern based descriptors LBP, LTP and LQP are inferior to LHS, owing to their naive encoding approach and coarse quantization. A near perfect recognition rate is shown by the LHS and CRPD descriptors.

It can be seen that for Brodatz dataset all the descriptors achieve better recognition rate compared to the KTH-TIPS-2a dataset. The variation between the training and the testing samples are not as high as the previous dataset as the samples for both are taken from similar image samples. It is easier to model the texture samples and moreover it does not require the generalization property.

V. CONCLUSION

We presented a novel approach for texture classification based on compressed patch based features (CRPD) and Fisher Vector encoding. The CRPD features are very fast to compute, easy to implement and discriminative in nature. When combined with efficient coding technique we obtain a robust texture descriptor. The tests performed on challenging datasets demonstrated the efficiency of the proposed approach. The results showed that the proposed features have the generalization property that models the inherent structure of the texture even if the training and the testing images are considerable different.

REFERENCES