



**HAL**  
open science

# Rotation-invariant features based on directional coding for texture classification

Farida Ouslimani, Achour Ouslimani, Zohra Ameer

► **To cite this version:**

Farida Ouslimani, Achour Ouslimani, Zohra Ameer. Rotation-invariant features based on directional coding for texture classification. *Neural Computing and Applications*, In press, 10.1007/s00521-018-3462-9 . hal-01800747

**HAL Id: hal-01800747**

**<https://hal.science/hal-01800747v1>**

Submitted on 27 May 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Rotation-invariant features based on directional coding for texture classification

Farida Ouslimani · Achour Ouslimani ·  
Zohra Ameur

Received: date / Accepted: date

**Abstract** A directional coding (DC) method is proposed to extract rotation invariant features for texture classification. DC uses four orientations in  $3 \times 3$  neighborhood pixel. For each orientation, the rank order of the central gray level pixel is calculated. The four ranks are used to get 15 codes. The codes are combined with the information of the central pixel to extract 30 rotation invariant features. For a multi-resolution study, DC is calculated by altering the window size around a central pixel. The number of samples is restricted to eight neighbors by local averaging. Therefore, in each single scale DC histogram, the number of bins is kept small and constant. Outex, CURET and KTH\_TIPS2 databases are used to evaluate and compare the proposed method against some state of the art local binary techniques and other texture analysis methods. The results obtained suggest that the proposed DC method outperforms other methods making it attractive for use in computer vision problems.

---

F.Ouslimani (✉)  
Laboratoire d'Analyse et de Modélisation des Phénomènes Aléatoires(LAMPA)  
Université Mouloud Mammeri  
Tizi ouzou , Algeria  
E-mail: ouslimanif@gmail.com

A.Ouslimani  
Laboratoire Quartz,EA -7393  
ENSEA  
6,avenue du ponceau  
95014 Cergy-Pontoise Cedex , France  
E-mail: achour.ouslimani@ensea.fr

Z.Ameur  
Laboratoire d'Analyse et de Modélisation des Phénomènes Aléatoires(LAMPA)  
Université Mouloud Mammeri  
Tizi ouzou , Algeria  
E-mail: z.ameur@ummto.dz

**Keywords** texture classification · rotation invariance · texture features · directional rank

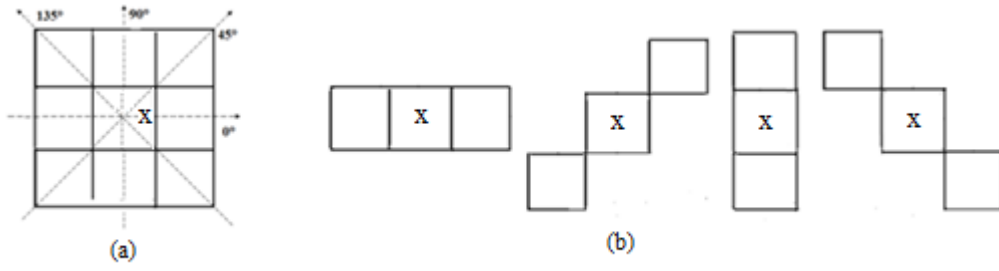
## 1 Introduction

Texture is one of the most important features in image processing and machine vision applications. Although texture analysis and classification have been extensively investigated over the past few decades, it is still an open problem, particularly for complex textured surfaces under varying luminance and viewing conditions. Many methods were proposed [1] where early ones were focussed on the statistical analysis of texture images such as the co-occurrence matrix method [2]. Later on, many model-based methods were proposed including circular autoregressive model [3] and Markov Random Fields [4–7]. Rotation and scale invariant features obtained through filter bank responses are also proposed in [8–10].

Local binary patterns (LBP) proposed by Ojala et al. [11] are one of the most important statistical features that have attracted much attention in recent years. In this method a neighborhood around each pixel is first defined, then the relationship between neighborhood pixel is exploited. Many extensions of this approach have recently been proposed [12–14]. For example, Liao et al. [15] extracted the most frequent patterns of LBP histograms to detect the dominant patterns which generate the new descriptors; the authors also used Gabor filter as a second feature to enforce the classification task. Heikkila et al. [16] proposed center-symmetric LBP (CS-LBP) by comparing center-symmetric pairs of pixels instead of comparing neighbors with the central pixels. Guo et al. [17] proposed to include both the signs and the magnitude and the center pixel intensity in order to improve the discriminative power of the original LBP operator. Khellah [18] proposed a Dominant Neighborhood Structure (DNS) method which fused the local LBP with global rotation-invariant features extracted from the generated image dominant neighborhood structure.

Although LBP and its variants have achieved impressive classification results on representative texture databases, there still remain some potential flaws mostly the limiting of LBP variants to three scales, failing to capture long range texture information [19]. Also, LBP variants often classify many different patterns into a same class [20].

In this paper, we propose a novel method to extract compact, discriminative and rotation invariant features. The Directional Coding method is based on directional information representing the rank order of the central gray level pixel calculated in four orientations in  $3 \times 3$  neighborhood pixel. The four ranks are exploited to get 15 codes which are combined with the information of the central pixel resulting in 30 rotation invariant features. For multi-resolution study, DC is calculated by altering the window size around a central pixel.



**Fig. 1** (a) a  $3 \times 3$  neighborhood pixel ,(b) the fourth elements used for coding .

The number of samples is restricted to eight neighbors by local averaging. Therefore, in each single scale DC histogram, the number of bins is kept small and constant.

The rest of this paper is organized as follows: section 2 introduces the rotation invariant texture descriptor, in section 3 results and discussions are presented. Finally the conclusions are given in section 4.

## 2 The directional coding method

We propose a new feature descriptor which is evaluated by taking into consideration the magnitude of local difference between the central pixel and its neighbors. Therefore, this scheme differs from the existing LBP in such a way that it extracts the information based on the distribution of local extrema in the four direction of a  $3 \times 3$  pixel neighborhood.

For this encoding, we consider a series of neighborhoods composed of four elements. Each element is defined by a central pixel  $x$  and two aligned adjacent pixels surrounding it in a particular direction as shown in Fig.1. This allows for the scanning of the four main directions  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$ .

This coding is processed in three steps:

First, for each direction, the rank of the gray level of the central pixel is defined. The detailed explanation of  $\theta$  direction calculation is given as follows: Let  $x$  be the central pixel,  $p_i$  ( $i = 1, 2$ ) are its two surrounding pixels according to  $\theta$  direction.  $I(x), I(p_i)$  are their corresponding gray levels .

$$rank_{\theta}(x) = \sum_{i=1}^2 F(I(x), I(p_i)) \quad \theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ \quad (1)$$

$$F(a, b) = \begin{cases} 1, & (a - b) \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The possible values of  $rank_{\theta}(x)$  are 0, 1 and 2. It can be noted that if  $rank_{\theta}(x)=0$ ;  $I(x)$  represents the minimum gray level according to the direction  $\theta$ ; in contrast it represents the maximum gray level if  $rank_{\theta}(x)=2$ .

The four directional ranks are associated in one vector noted  $Rank$ ,  $Rank = [rank_{135} \ rank_{90} \ rank_{45} \ rank_0]$ . So, the vector  $Rank$  brings directional extrema which is a significant and important information for texture characterization.

Second, to remove the effect of rotation and extract rotation invariant features, the three values  $n_0$ ,  $n_1$  and  $n_2$  corresponding respectively to the number of 0, 1 and 2 in the vector ranks are exploited.  $n_0$  corresponds to the number of times the central gray level pixel is a minimum ( $rank_{\theta}(x)=0$ ) while  $n_2$  represents the number of times the central gray level is a maximum ( $rank_{\theta}(x)=2$ ).  $n_1$  represents the number of times the central pixel has an intermediate gray level ( $rank_{\theta}(x)=1$ ).  $n_0$ ,  $n_1$  and  $n_2$  can be obtained using the following equations:

$$n_l = \sum_{i=1}^4 f(Rank(i), l) \quad l \in [0, 2] \quad (3)$$

$$f(a, b) = \begin{cases} 1 & a = b \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$n_l$  is included in the closed interval  $[0,4]$ , such that  $\sum_{l=0}^2 n_l = 4$ . There are 15 possible combinations of the three values  $n_0, n_1$  and  $n_2$  which are 400, 040, 004, 211, 112, 121, 013, 031, 022, 202, 220, 310, 103, 310 and 130. For example, the combination 301 means that the central gray level pixel is a minimum on three directions and a maximum on one direction.

It has been proven that the central pixel can be used to express the local gray level in image [16, 17]. Therefore, to find the DC operator,  $n_l$  are jointly combined with the information of the central pixel using  $C$  which thresholds the central pixel against the global mean gray value of the whole image according to equation (6). As  $n_l$  is included in the closed interval  $[0,4]$ , it can be presented in base 5. So to find the DC operator, the code  $(C \ n_2 \ n_1 \ n_0)$  is converted to decimal code using equation (5). The DC operator produces 30 rotation invariant features.

$$DC(x) = n_0 \times 5^0 + n_1 \times 5^1 + n_2 \times 5^2 + C \times 5^3 \quad (5)$$

$$C = F(I(x) - \frac{1}{N \times M} \sum_{i=1}^N \sum_{j=1}^M I(i, j)) \quad (6)$$

$N \times M$  represent the size of texture image,  $F$  is defined in (2).

Fig.2 shows two examples of calculation of the DC code. The gray level of the center pixel is first classified according to the four directions:  $0^\circ, 45^\circ, 90^\circ$  and  $135^\circ$ . In the first example, at direction  $0^\circ$ , the central pixel is surrounded by two neighbors having a corresponding gray level of 64 and 32. As such, the rank of the central pixel equal to 1 ( $Rank_0=1$ ). Also, at each direction  $45^\circ$  and  $90^\circ$ , the central pixel has the maximum gray level and its rank is equal to 2 ( $Rank_{45}=Rank_{90}=2$ ). Also, the rank of the central pixel at direction  $135^\circ$  is

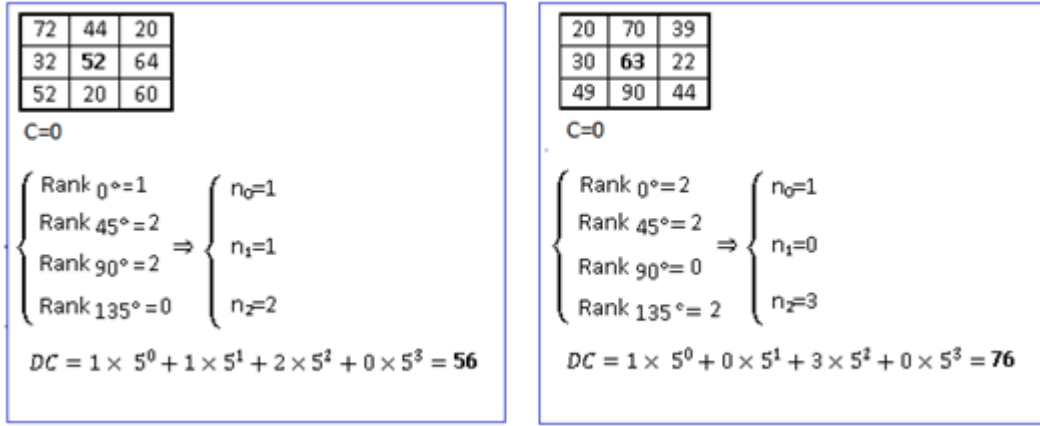


Fig. 2 Two examples of computing DC

0 ( $Rank_{135}=0$ ), since the gray level of the two neighbors are greater than that of the central pixel. DC is finally calculated by combining the three values  $n_0$ ,  $n_1$  and  $n_2$  with  $C$  according to equation 5 (we assume in this example that  $C = 0$ ). The final value of DC is 56. In the second example, the central gray level pixel is a minimum in three directions ( $n_0 = 3$ ) and a maximum on one direction ( $n_2 = 1$ ). The final value of DC is 76.

After computing the DC for each pixel ( $i,j$ ), the whole image  $I$  is represented by building an histogram using equation (7)

$$H_{DC(I)} = \sum_{i=1}^N \sum_{j=1}^M f(DC(i, j), l) \quad l \in [0, L] \quad (7)$$

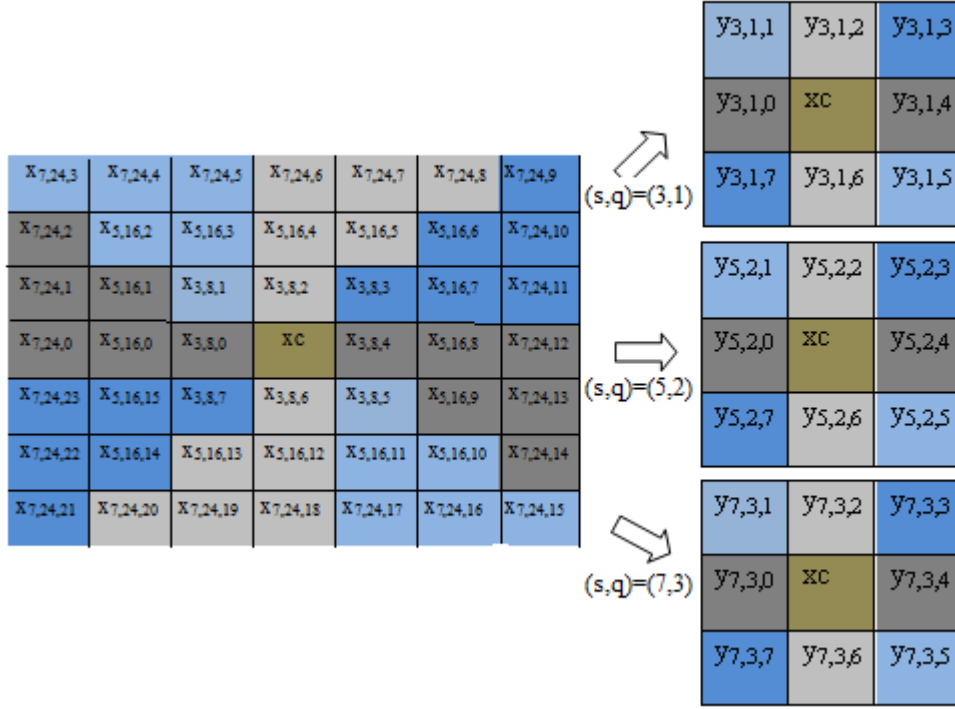
$L$  is the maximal DC code value,  $f$  is defined in (4).

## 2.1 Multi scale DC

For a multi-resolution study, DC feature Histogram is concatenated over the multiple scales. To limit the growth in histogram bins with scale, we deploy the strategy proposed in [19] with the neighbors of the central pixel  $x$  are sampled on square neighborhoods having a window of size  $s \times s$ . The number  $P$  of neighbors is restricted to be a multiple of eight, thus  $P=8 \times q$  for positive integer  $q$ .

$$x_{s,8q} = [x_{s,8q,0}, \dots, x_{s,8q,8q-1}] \quad (8)$$

The neighbors vector  $x_{s,8q}$  is transformed by local averaging to  $y_{s,q}$ , such that  $y_{s,q}=[y_{s,q,0}, \dots, y_{s,q,7}]$ , the number of neighbors in  $y_{s,q}$  is always eight according to equation (9)



**Fig. 3** Illustration of transforming original neighborhood into new one with  $3 \times 3$  window size. The different colors correspond to the 4 directional neighborhoods used in DC operator

$$y_{s,q,i} = \frac{1}{q} \sum_{k=0}^{q-1} x_{s,8q,(qi+k)} \quad i = 0, \dots, 7 \quad (9)$$

Fig. 3 illustrates an example for transforming the original neighborhood  $x_{s,8q}$  into  $y_{s,q}$  according to equation 8, for example  $y_{5,2,0} = \frac{x_{5,16,0} + x_{5,16,1}}{2}$  and  $y_{7,3,1} = \frac{x_{7,24,3} + x_{7,24,4} + x_{7,24,5}}{3}$

The DC code is calculated for different spatial resolutions by altering the window size  $s \times s$ . Given  $y_{s,q} = [y_{s,q,0}, \dots, y_{s,q,7}]$ , DC is computed with respect to the central pixel for different scales according to equations (1) and (2). Finally, the DC feature Histogram is concatenated over the multiple scale resulting in ,it has  $n \times 30$  dimensional features, where  $n$  is the number of scales.

### 3 Results and discussions

To demonstrate the performances of the proposed technique, experiments were performed on three comprehensive databases: Outex [21], CURET [22] and KTH.TIPS2b [23], summarized in table I. These databases have different properties such as different number of classes, rotation, illumination and poses.

**Table 1** Summary of texture datasets used for classification

Dataset name	No of classes	Images Per class	Total images	Image size(pixels)
OUTEX	24	200	4800	128 × 128
CURET	61	92	5612	200 × 200
KTH	11	432	4752	200 × 200

Each texture image is converted into gray scale and normalized to zero mean and unit standard deviation. DC feature Histogram is concatenated over five scales by altering the window size from  $3 \times 3$  to  $11 \times 11$ . The features are computed at each resolution as described in Sect. 2.1. The concatenating features histogram is noted MSn, n is the number of scales.

The k nearest neighbors (k-NN) classifier was trained with a subset of textures from each class, and its ability to recognize textures of the same class at different alterations was then evaluated. The k-NN is used with  $\chi^2$  distance defined as:

$$\chi^2(a, b) = \frac{1}{2} \sum_i \frac{(a_i - b_i)^2}{(a_i + b_i)} \quad (10)$$

Where  $a$  and  $b$  are the features vectors. The number of k-NN was fixed to one for all experiments.

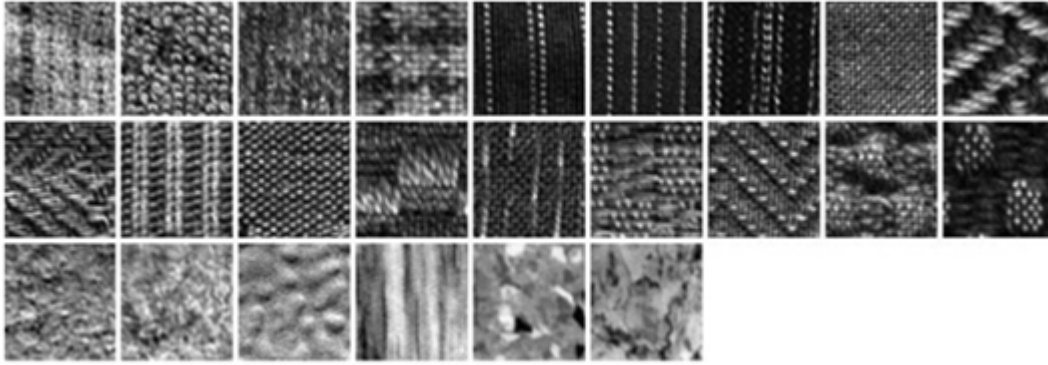
Our proposed method was compared with the conventional LBP approach and some of its modified approaches such as DLBP[15], CLBP[17], and DNS[18]. VZ-MR8 [8] and VZ- Joint[24] are also used for comparison.

### 3.1 Experiments on Outex Database

The Outex database includes two test suites: Outex \_TC\_0010 and Outex \_TC\_0012 which were created for the classification of rotation invariant textures, and also for classification of rotation and illumination invariant textures, respectively. These two test suites contain the same 24 classes of texture as shown in Fig.4. Each texture was collected under three different illumination conditions ("inca", "tl84" and "horizon") and nine different rotation angles ( $0^\circ, 5^\circ, 10^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ$ ). There are 20 nonoverlapping  $128 \times 128$  texture samples for each class under each setting. Therefore, there are three datasets in total each contains 4320( $24 \times 20 \times 9$ ) different texture samples. The experimental setups for both sets are as follows:

1. For TC10, samples of illuminant "inca" and angle  $0^\circ$  in each class were used for classifier training and the other eight rotation angles with the same illuminant were used for testing. Hence, there are 480( $24 \times 20$ ) models and 3840( $24 \times 8 \times 20$ ) validation samples.
2. For TC12, the classifier was trained with the same training samples as TC10, it was tested with all samples captured under tl84 and horizon light illumination referred respectively by "TC12\_000" and "TC12\_001". Hence, there are





**Fig. 4** Samples of 24 classes in Outex \_TC.0010 and Outex \_TC.0012.

**Table 2** DC performance as a function of number of scales, the highest classification accuracy highlighted in bold

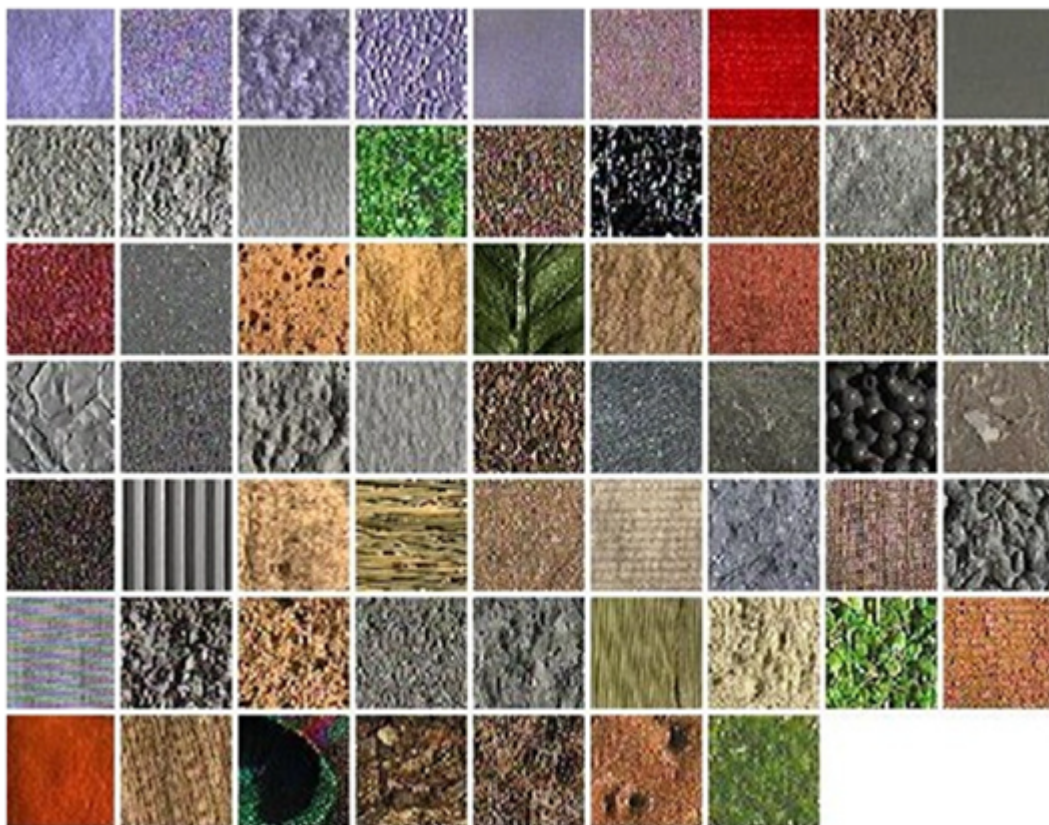
Outex Databases	MS1	MS2	MS3	MS4	MS5
TC10	95.39	96.48	<b>97.24</b>	96.80	96.22
TC12_000	87.11	91.46	92.85	<b>92.94</b>	92.87
TC12_001	86.71	90.97	92.57	<b>92.94</b>	92.73

**Table 3** Comparison of the proposed method with other texture analysis methods on TC10 and TC12 datasets. The two highest classification accuracies are highlighted in bold

Methods	TC10	TC12_000	TC12_001	Features number
$LBP_{P,R}^{riu2}/Var$ [11]	97.7	87.3	86.4	242
VZ-MR8[8]	93.59	92.55	92.82	610
VZ-Joint[24]	92.00	91.41	92.06	610
DLBP+NGF[15]	99.1	93.2	90.4	$K_{80\%}$
$CLBP_{P,R}^{riu2}$ [17]	<b>99.14</b>	<b>95.18</b>	<b>95.55</b>	2200
$DNS + LBP_{24,3}$ [18]	<b>99.27</b>	<b>94.40</b>	92.85	242
DC(MS3)	97.24	92.85	92.57	90
DC(MS4)	96.80	92.94	<b>92.94</b>	120

480( $24 \times 20$ ) models and 4320( $24 \times 20 \times 9$ ) validation samples for each illuminant.

We evaluated the classification performance of the proposed method as a function of number of scales. Table 2 presents the results obtained from one to five scales. The best performance is obtained using three scales for TC10 and four scales for TC12. Compared to several methods in literature (Table 3), the proposed method features is competitive with the most successful techniques. Especially on TC12, the classification rate of DC appears among the best using the smallest number of features, for example the classification rate of CLBP is more successful however CLBP feature size (2200) is about eighteen times the feature size of DC (120).



**Fig. 5** Sixty-one homogeneous texture samples from the CURET data set .

### 3.2 Experiments on CURET database

The CURET database consists of 61 different material surfaces where each one is observed with 205 combinations of viewing and illumination directions (see Fig.5). However, only 92 samples per class are selected for the classification task. In this experiment, different numbers of images (46, 23, 12 and 6) per class are randomly chosen for training. For each case, the average classification accuracies of 50 samples of training and testing sets are used to assess the performance.

As for the Outex data base, the results have shown an increase of the classification rates when the number of scales is increased (Table 4). The maximum is achieved using five scales (MS5) when the number of samples  $N$  equals 46 or 23. However, the maximum is obtained for three scales (MS3) when  $N$  equals 12 or 6. Compared against other well-known methods from the literature, the proposed method achieves attractive results. As indicated in Table 5, the proposed DC significantly improves the others methods . For exam-

**Table 4** DC performance as a function of number of scales, the highest classification accuracy highlighted in bold

Methods	Number of training sample (N)			
	46	23	12	6
MS1	58.01	56.71	53.28	52.17
MS2	95.72	94.40	91.92	87.45
MS3	98.41	96.84	<b>94.50</b>	<b>89.23</b>
MS4	98.50	96.88	94.01	89.21
MS5	<b>98.67</b>	<b>97.01</b>	94.14	88.48

**Table 5** Comparison of the proposed method with other texture analysis methods on CURET dataset. The two highest classification accuracies are highlighted in bold

Methods	features size	Number of training sample (N)			
		46	23	12	6
VZ-MR8[8]	610	97.79	95.03	90.48	82.60
VZ-Joint[24]	610	97.66	94.58	89.40	81.06
DLBP+NGF[15]	$K_{80\%}$	84.1	-	-	-
$DNS + LBP_{16,2}$ [18]	234	95.00	-	-	-
$CLBP_{P,R}^{riu2}$ [17]	2200	97.39	94.19	88.72	79.88
DC(MS3)	90	<b>98.41</b>	<b>96.84</b>	<b>94.50</b>	<b>89.23</b>
DC(MS5)	150	<b>98.67</b>	<b>97.01</b>	<b>94.14</b>	<b>88.48</b>

ple, the best classification rates obtained by VZ-MR8 with different number of training images (N=46, 23, 12 or 6) are respectively 97.79%, 95.03%, 90.48% and 82.60%. With the same configuration setting the DC(MS3) reaches classification rates of 98.41%, 96.84%, 94.50% and 89.23%. The best classification accuracy achieves (98.67%) which is obtained with DC(MS5) improves the state of the art performance on this date set. It is to be noted that when the number of training sample is small, larger improvement is achieved. For example DC(MS3) achieves 0.6 % higher than VZ-MR8 when N=46, while difference is more than 7% when N=6. Also VZ-MR8 feature size (610) is about seven times the feature size of DC (90).

### 3.3 Experiments on KTH-TIPS2 database

The major drawback of the CURET database is that materials are imaged at a constant scale. In contrast, the KTH-TIP2-b database [19] uses different distances from the camera. It includes four physical samples of 11 different materials similar to those used in CURET database (see Fig.6). These samples were imaged with variation in scale as well as variations in pose and illumination. Images were taken in combination of three poses, four illuminations and nine scale yielding 108 images per each physical sample. The images are then cropped to a size of (200 × 200) pixels and the four physical samples are categorized into one texture class. In result the database contains 4752 images



**Fig. 6** The KTH-TIPS2 data set includes 11 different texture classes.

**Table 6** Comparison of the proposed method with other texture analysis methods on KTH-TIPS2b dataset. The two highest classification accuracies are highlighted in bold

Methods	features size	Number of training set		
		1	2	3
VZ-MR8[8]	610	46.1	52.0	55.3
VZ-Joint[24]	610	53.5	60.0	61.0
DLBP+NGF[15]	$K_{80\%}$	49.3	55.1	58.0
$CLBP_{P,R}^{riu2}$ [17]	2200	<b>55.0</b>	<b>61.1</b>	<b>67.7</b>
DC(MS1)	30	54.17	61.08	64.88.0
DC(MS2) $s=3,17$	60	<b>59.99</b>	<b>64.47</b>	<b>68.56</b>

$(11 \times 108 \times 4)$ . For this purpose we used three configurations, with one, two and three training sets per class. The sets 1,2,3 and 4 were combined as (training, testing) couples with all possible combinations; for one training set we have the couples  $(1, 2 \cup 3 \cup 4)$ ,  $(2, 1 \cup 3 \cup 4)$ ,  $(3, 1 \cup 2 \cup 4)$ ,  $(4, 1 \cup 2 \cup 3)$ . For two training sets  $(1 \cup 2, 3 \cup 4)$ ,  $(3 \cup 4, 1 \cup 2)$ ,  $(1 \cup 3, 2 \cup 4)$ ,  $(2 \cup 4, 1 \cup 3)$ ,  $(1 \cup 4, 2 \cup 3)$  and  $(2 \cup 3, 1 \cup 4)$ . Three training sets yield  $(2 \cup 3 \cup 4, 1)$ ,  $(1 \cup 3 \cup 4, 2)$ ,  $(1 \cup 2 \cup 4, 3)$  and  $(1 \cup 2 \cup 3, 4)$ . The classification score is the average of all combinations scores for each configuration.

As indicated in Table 6, the proposed DC method achieves the best classification accuracy when compared against the other techniques including VZ-MR8, VZ-joint and CLBP. The best result is obtained using two scales(MS2), The two windows size( $s=3, s=17$ ) are chosen after many tests. The DC classification accuracy achieves 59.99%, 64.47% and 68.56% for one, two and three training sets, respectively. Note that with even one scale (MS1) using only 30 features, the proposed method achieves a significant classification rate compared to other methods which prove the relevance of the DC features.

### 3.4 Computational Complexity

The DC algorithm is very fast. Its computation is linear with respect to the number  $N$  of pixels in the image and to the cardinality  $P$  of the neighborhood. The procedure has a complexity in time of  $O(N \times P)$ .

All experiments in this paper have been implemented on a PC with Intel i5 core, 4G RAM Windows 7 and Matlab version 7.10(2010a). The time elapsed to extract the 30 features from a texture image of size  $200 \times 200$  is about 0.08 seconds.

## 4 Conclusion

We have proposed a novel method to extract compact, discriminative and rotation invariant features. We have evaluated the proposed descriptor over a wide range of textures using, Outex, CURET and KTH\_TIPS2 databases and studied different problems related to texture recognition such as the effect of illumination, rotation and scale change, and also the influence of the number of the training samples. Compared with the other state of the art methods, the proposed method acquired the highest classification rate on CURET and KTH-TIPS2 databases; it also had promising results on Outex database. According to its simplicity and its small features size, the proposed method opens opportunities for other researcher to work on different parts such as employing other multi resolution analysis.

## Conflict of interest

No conflict of interest.

## References

1. Zhang J. and Tan T., Brief review of invariant texture analysis methods, *Pattern Recogn.*, 35, 735-747 (2002)
2. Haralik R.M., Shanmugam K. and Dinstein I., Texture features for image classification, *IEEE Trans Syst Man Cybern* , 3(6),610-621(1973)
3. Kashyap RL. and Khotanzed A., A model-based method for rotation invariant texture classification, *IEEE Trans Pattern Anal Mach Intell*, 8(4), 610-621 (1973)
4. Cohen FS., Fan Z. and Patel M.A. , Classification of rotated and scaled textured image using Gaussian markov random field models, *IEEE Trans. Pattern Anal. Mach. Intell*, 13, 192-202 (1991)
5. Chen JL. and Kundu A. ,Rotation and gray scale transform invariant texture identification using wavelet decomposition and hidden Markov model, *IEEE Trans Pattern Anal Mach Intell*, 16(2) , 208-214( 1994)
6. Deng H. and Clausi D.A. , Gaussian MRF rotation invariant features for image classification, *IEEE Tran. Image Process.*,26(7),951-955(2004)
7. Dharmagunawardhana C. , Mahmoodi S. , Bennett M. , Niranjana M. ,Rotation invariant texture descriptors based on Gaussian Markov random fields for classification, *Pattern Recogn. Lett.* ,69 ,15-21 (2016)
8. Varma M. and Zisserman A. , A statistical approach to texture classification from single images. *Int. J. Comput. Vision.*, 62(1-2), 61-81(2005)
9. Ahmadvand A. , Daliri M.R. , Invariant texture classification using a spatial filter bank in multi-resolution analysis, *Image and Vision Computing*, 45, 1-10 (2016)
10. Ahmadvand A. and Reza Daliri M., Rotation invariant classification using extended wavelet channel combining and LL channel filter bank, *knowledge-Based Systems*, 97, 75-88(2016)
11. Ojala T., Pietikainen M. and Maenpaa M., Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Tran. Pattern Anal.*,24(7), 971-987 (2002)
12. Hafiane A., Palaniappan K. , Seetharaman G., Median Binary Patterns for texture classification. *Pattern Recogn.*,48, 2609-2620 (2015)
13. Hegenbart S. , Andreas Uhl., A scale-and orientation-adaptive extension of Local Binary Patterns for texture classification, *Pattern Recogn.*,48,2633-2644 (2015)
14. Zhao Y , Wang R.G. , Wang W.M. and Gao W. , Local quantization code histogram for texture classification , *Neurocomputing* , 207, 354-364(2016)

15. Liao S., Law M.W.K and Chung A.C.S. , Dominant Local Binary Patterns for Texture Classification, *IEEE Tran. Image Process.*, 18(5), 1107-1117( 2009)
16. Heikkila M.,Pietikainen M.,and Schmid C., Description of interest regions with center-symmetric local binary patterns, *Proceedings of Computer Vision, Graphics and Image Processing*4338,58-69(2006)
17. Guo Z., Zhang L. and Zhang D., A completed modeling of local binary pattern operator for texture classification, *IEEE Tran. Image Process.*,9(16),1657-1663( 2010)
18. Khellah F.M., Texture Classification Using Dominant Neighborhood Structure, *IEEE Tran. Image Process.*, 20(11), 3270-3279 (2011)
19. Liu L., Long Y. , Fieguth P., Lao S., and Zhao G., BRINT: Binary Rotation Invariant and Noise Tolerant Texture Classification, *IEEE Tran. Image Process.*, 23( 7), 3071-3084( 2014)
20. Zhao Y., Jia W. , Hu R.X., Min H., Completed robust local binary pattern for texture classification *Neurocomputing*, 106 , 68-76(2013)
21. Ojala T., Maenpaa T., Pietikainen P., Viertola J., Kyllonen J., and Huovinen S., OutexNew framework for empirical evaluation of texture analysis algorithms, in: *Int. Conf. Pattern Recognit.*, 1, 701-706( 2002),<http://www.outex.oulu.fi>
22. Dana K., Van-Ginneken B., Nayar S. and Koenderink J., Reflectance and texture of real world surfaces, *ACM Trans.Graph.*,181-34 (1999), <http://www.cs.columbia.edu/CAVE/exclude/curet/>
23. Mallikarjuna P. ,Fritz M., Targhi A.T. , Hayman E., Caputo B. and Eklundh J. , The KTH-TIPS2 Databases, <http://www.nada.kth.se/cvap/databases/kth-tips/>
24. Varma M. and Zisserman A., A statistical approach to material classification using image patch exemplars, *IEEE Trans.Pattern Anal.Mach.Intell.*,31,2032-2047(2009)