Image and Vision Computing 30 (2012) 501–512

Contents lists available at SciVerse ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Local phase quantization for blur-insensitive image analysis ☆

Esa Rahtu [a,*], Janne Heikkilä [a], Ville Ojansivu [b], Timo Ahonen [c]

[a] Center for Machine Vision Research, University of Oulu, Finland
[b] Institute for Molecular Medicine Finland, University of Helsinki, Finland
[c] Nokia Research Center, Palo Alto, CA, United States

ARTICLE INFO

Article history:
Received 16 September 2011
Received in revised form 18 January 2012
Accepted 20 April 2012

Keywords:
Feature extraction
Invariant features
Blur invariance
Texture recognition
Face recognition

ABSTRACT

One of the principal causes for image quality degradation is blur. This frequent phenomenon is usually a result of misfocused optics or camera motion, and it is very difficult to undo. Beyond the impaired visual quality, blurring causes problems to computer vision algorithms. In this paper, we present a simple yet powerful image descriptor, which is robust against the most common image blurs. The proposed method is based on quantizing the phase information of the local Fourier transform and it can be used to characterize the underlying image texture. We show how to construct several variants of our descriptor by varying the technique for local phase estimation and utilizing the proposed data decorrelation scheme. The descriptors are assessed in texture and face recognition experiments, and the results are compared with several state-of-the-art methods. The difference to the baseline is considerable in the case of blurred images, but also with sharp images our method gives a highly competitive performance.

1. Introduction

Textures are characteristic intensity variations of surfaces and in nature they can express an enormous number of different patterns. The analysis of textures is one of the fundamental aspects of computer vision, and the inventions in texture description have played an important role in many influential results [1]. Recognizing textures subject to arising perturbations and changes in image acquisition process, such as noise, illumination, or scale, is very challenging, which is often emphasized by the large intraclass variability.

Generally most distortion types are very difficult or expensive to undo, and furthermore, in computer vision, the restored image often has very little use as such. Therefore, many recent works particularly seek to generate descriptors that are invariant to selected aspects of image distortions. Another common property with the most successful descriptors is their local nature. Regardless of the numerous global methods introduced, the performance of the local descriptors is still incomparable. The difference is most likely due to the local nature of the texture, which is difficult to capture, with necessary details, using global attributes. This is usually the case despite the fact that invariance is often easier to formulate in global framework.

A classical and well studied group of local invariants consists of methods that are insensitive to certain view angle changes. These include image rotations [2–8], scalings [9–11], affine transformations [12], and even homography [13,14]. Quite often these approaches have been developed from previous successful non-invariant methods, like Gabor filtering [15], Markov random field [16], or Local Binary Patterns (LBP) [17]. The common ways to achieve the invariance, or more precisely insensitivity, are based on filtering the original image, applying multiresolution approach, seeking characteristic local scale and orientation, and integrating over the entire equivalence class.

Another set of methods consist of descriptors that are tolerant to image degradation, like noise [17] and nonuniform illumination [17–20]. Robustness to these attributes is achieved e.g. by soft assignments [21] and using floating quantization instead of fixed one [17]. A less frequently studied property in this class is the tolerance to image blurring, although the problem is very apparent in practical photography. Blurring is most often caused by camera motion, misfocused optics, movements in the scene, and atmospheric turbulence. Recently, the importance of this degradation type is further emphasized by the appearance of small scale imaging devices, with limited optical quality, like mobile phones.

This paper focuses on blur tolerant texture analysis. Quite surprisingly the number of works that consider blur invariant descriptors is very limited. A method based on color constancy was presented in [22]. Another three approaches based on moments and Fourier spectra were introduced in [23–25]. These techniques are however global in nature, and hence they are not well suited for texture characterization.

We propose a new texture description method, referred to as Local Phase Quantization (LPQ), which is insensitive to common blur types yet being surprisingly discriminative. To be more precise, we assume that the point spread function (PSF) of the blur can be locally approximated by centrally symmetric model, which is often sufficient for example in the case of camera motion, misfocused optics, and atmospheric turbulence. Furthermore, the experiments indicate that the method is

☆ This paper has been recommended for acceptance by Shishir Shah.

* Corresponding author. Tel.: +358 85532778.
E-mail address: erahtu@ee.oulu.fi (E. Rahtu).

0262-8856/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2012.04.001

tolerant of deviations from the centrally symmetric model. We emphasize that blur can be spatially varying, which is a very important property in real applications [26,27]. The experiments performed with face image database as well as in other works using LPQ [28–35], indicate that the applied blur model is sufficient in real world applications.

Our method is based on quantizing the local phase spectrum and characterizing the underlying texture by the distribution of different codewords in the image region. The proposed encoding scheme is similar to the LBP method [17]. In addition to the basic construction, we investigate the effects of using different phase estimation methods and present a data decorrelation scheme, which is able to boost the quantization performance with certain blur types. Local phase information has been widely applied in previous feature extraction works [36–40]. However, to our knowledge, LPQ is the first method to use local phase to construct blur invariant features.

In the experimental evaluations, we apply two commonly used datasets (Outex and Brodatz) and analyze the performance of LPQ variants in the case of sharp and blurred textures. We further assess the proposed descriptor in dynamic texture and face recognition applications using Dyntex++, CMU PIE and Face Recognition Grand Challenge datasets. In face recognition, the use of histogram of quantized Gabor phases has been explored before [38,41], though not in the context of blur invariant analysis.

The baseline for the results is formed using several state-of-the-art texture descriptors: LBP [17], Gabor filter bank [42], VZ-MR8 [2], VZ-joint [19], BIF-columns [3], and LTP [20]. Compared to these methods, a considerable improvement in performance was observed in the case of blurred images. Surprisingly, in the majority of cases, LPQ achieved equal or better performance compared to the best of the baseline methods.

The preliminary versions of the LPQ method were presented in [43–45]. However, the analysis and experiments performed here are more thorough and several variants of the techniques are introduced. Since its introduction, LPQ has been already adopted into several applications, particularly in face and activity recognition [28–32], and medical image analysis [33–35].

This paper is organized as follows. In Sections 2 and 3, we revisit the related important properties of the Fourier transform. Section 4 introduces the actual LPQ descriptors and in Section 5, we discuss the discretization and other implementational issues of the method. Finally in Sections 6, 7, 8, and 9, we present the experiments and concluding remarks.

2. Frequency domain analysis of spatially invariant blurring

Suppose $f(x):\mathbb{R}^2\mapsto\mathbb{R}$ is an image intensity function corresponding to a gray-scale image. The spatially invariant blurring of f can be expressed by a convolution [46] as

$$g(x) = (f * h)(x), \tag{1}$$

where * denotes 2-D convolution, g($\mathbf{x}$) is the blurred image, and $h(x):\mathbb{R}^2\mapsto\mathbb{R}$ is the point spread function (PSF) of the blur.

In the frequency domain, the expression (1) turns into

$$\hat{g}(u) = \hat{f}(u)\cdot\hat{h}(u), \tag{2}$$

where $\hat{a}(\mathbf{u})$ refers to the Fourier transforms of a function a. Eq. (2) can be further divided into magnitude and phase parts as

$$\begin{aligned} |\hat{g}(\mathbf{u})| &= \left|\hat{f}(\mathbf{u})\right|\cdot\left|\hat{h}(\mathbf{u})\right| \quad \text{and} \\ \angle\hat{g}(\mathbf{u}) &= \angle\hat{f}(\mathbf{u}) + \angle\hat{h}(\mathbf{u}), \end{aligned} \tag{3}$$

where $|\cdot|$ is the absolute value and $\angle$ denotes the phase angle.

Eq. (3) implies that blurring will have an effect to both magnitude and phase spectra of the original image. However, if we can assume that $h(\mathbf{x})$ is centrally symmetric, i.e. $h(\mathbf{x}) = h(-\mathbf{x})$, the Fourier transform $\hat{h}(\mathbf{u})$will always be real valued and as a consequence

$$\angle\hat{h}(u) = \begin{cases} 0 & \text{if } \hat{h}(u)\geq 0 \\ \pi & \text{if } \hat{h}(u)<0 \end{cases} \tag{4}$$

and

$$\angle\hat{g}(u) = \begin{cases} \angle\hat{f}(u) & \text{if } \hat{h}(u)\geq 0 \\ \angle\hat{f}(u)+\pi & \text{if } \hat{h}(u)<0. \end{cases} \tag{5}$$

The result (5) has been applied previously in [25] to construct blur invariants by doubling the phase $\angle\hat{g}(\mathbf{u})$ modulo 2π. However, doubling is unnecessary if we would know the frequencies where $\hat{h}(u)$ is positive, which is possible for certain blur types. For example, in the case of ideal motion and out of focus blur, the cross-section of $h(\mathbf{x})$ is rectangular [46], which causes the Fourier transform $\hat{h}(\mathbf{u})$ to be sinc and Bessel functions, respectively. Hence, at least the values before the first zero crossing, at frequency ≈ (sampling frequency)/(blur length), are positive. Furthermore, in the case of atmospheric blur the $\hat{h}(\mathbf{u})$ is approximately Gaussian and has solely positive values. In Fig. 1, we illustrate some examples of centrally symmetric PSFs and the corresponding Fourier transforms.

The ideal blur invariance is, however, not achieved in practice, since usually the assumptions above do not hold exactly. In addition, the linear convolution in Eq. (1) enlarges the size of the resulting image and the parts that spread over the viewing area are cut away from the observed image $g(\mathbf{x})$. This effect is further emphasized if the Fourier transform is computed from local patches instead of the entire image [47]. Nevertheless, it is possible to encode the phase information into a highly blur tolerant descriptor as illustrated by the experiments in Sections 6 and 8, [43,45,48].

3. Local frequency analysis

Since texture is rather a local than global property, the corresponding descriptors should also represent this quantity. Therefore, instead of computing the Fourier transform of the entire image, we perform the estimation locally. The further advantage resulting from local estimation is that instead of assuming spatially invariant blur in Eq. (1), the PSF is allowed to vary globally as long as it can be approximated to be invariant in a local neighborhood. In practice we use approximately 7×7 patches in the local frequency estimation, which means that the spatially invariant blur assumption needs to hold only in relatively small scale.

The local Fourier transform, needed for the proposed descriptors in Section 4, can be obtained using any desired method. In this work we apply a linear filtering technique which can be expressed by a convolution:

$$F(\mathbf{x},\mathbf{u}) = \int_{\mathbb{R}^2} f(\mathbf{x}-\mathbf{y})s(\mathbf{y},\mathbf{u})d\mathbf{y}, \tag{6}$$

where $\mathbf{x}$ denotes the 2-D position in the image, $\mathbf{u}$ denotes the 2-D frequency, and $s:\mathbb{R}^2\mapsto\mathbb{C}$ is the complex valued filter function.

The properties of the obtained local transform $F(\mathbf{x},\mathbf{u})$ are determined by the filter used. The local image neighborhood is defined by the support of s and the frequency properties can be seen by applying the Plancherel formula ($\int_{\mathbb{R}^2} f^* g = \int_{\mathbb{R}^2} \hat{f}^* \hat{g}$) resulting in

$$\int_{\mathbb{R}^2} f(\mathbf{x}-\mathbf{y})s(\mathbf{y},\mathbf{u})d\mathbf{y} = \int_{\mathbb{R}^2} \hat{f}_{\mathbf{x}}(\mathbf{v})\hat{s}(\mathbf{v},\mathbf{u})d\mathbf{v}, \tag{7}$$

where $\hat{f}_{\mathbf{x}}$ is the Fourier transform of $f(\mathbf{y}+\mathbf{x})$ (i.e. a translated image). Eq. (7) implies that the response $F(\mathbf{x},\mathbf{u})$ is a weighted sum of the

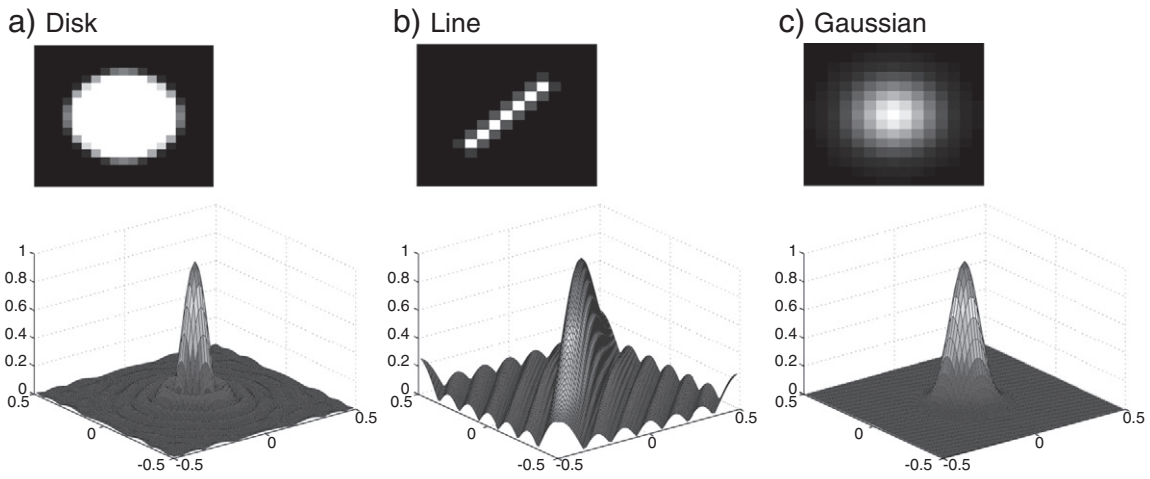


Fig. 1. Examples of blur PSFs and corresponding Fourier transforms. These PSFs roughly model out of focus, motion, and atmospheric turbulence, respectively.

frequency components of the image patch, where the weighting is determined by the frequency coefficients of the filter.

Since it is not possible to construct s with finite support and non-zero frequency only at position $\mathbf{u}$, the design of the filter is always a compromise between the locality in spatial and frequency domains. In this work we construct the filters using two well known approaches, namely a short-term Fourier transform (STFT) and a pair of band-pass quadrature filters. In the following we will shortly present how these methods are applied here, and for further particulars one can refer, for example, to [49].

3.1. Band-pass quadrature filters

The band-pass quadrature filters originate from the definition of 1-D analytical signal. The generalization to 2-D functions is not straightforward, although several possible approaches have been proposed in the literature [49]. The definition of monogenic function [37,50] is perhaps the most prominent method, but due to the isotropic nature it is not well suited for our framework.

Instead, we look for computationally efficient approximations by constructing a separable 2-D filter from two 1-D band-pass quadrature filters as

$$s_Q(\mathbf{x},\mathbf{u}) = \begin{cases} s_1(x_1,u_1)s_2(x_2,u_2), & \text{if } ||\mathbf{x}||_\infty \le m, \\ 0 & \text{otherwise}, \end{cases} \tag{8}$$

where $\|\cdot\|_\infty$ denotes infinity norm, m defines the size of the local $m\times m$ neighborhood, s_1 and s_2 are two 1-D filters, $\mathbf{x}=[x_1,x_2]$ is a 2-D coordinate vector, and $\mathbf{u}=[u_1,u_2]$ is a 2-D frequency variable.

A comprehensive study of commonly used 1-D filter options is presented in [49], and based on their observations we selected Gaussian derivative and Gabor filters to be applied in our study. The Gabor filters result in the equivalent estimator as the short-term Fourier transform with a Gaussian window and we will formulate this in the following section.

The 1-D Gaussian derivative quadrature filter is defined in frequency domain as

$$G_d(u,a) = \begin{cases} u^a e^{-\sigma_A^2 u^2}, & \text{if } u\ge 0, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where $a\ge 1$ determines the position of the peak frequency and $\sigma_A>0$ is a parameter to be selected (we use $\sigma_A=8/(m-1)$). We additionally define analogous filters for zero and negative values of a and as summary we have

$$s_1(x,a) = \begin{cases} \mathcal{F}^{-1}(G_d(u,a))(x), & \text{if } a\ge 1, \\ \mathcal{F}^{-1}(G_d(u,-a))(x)^*, & \text{if } a\le -1, \\ \mathcal{F}^{-1}\left(e^{-2\sigma_A^2 u^2}\right), & \text{if } a=0, \end{cases} \tag{10}$$

where $\mathcal{F}^{-1}$ and * denote the inverse Fourier transform and complex conjugate, respectively. The 1-D filter s_1 is then applied in Eq. (8) to obtain the actual 2-D filter s_Q. Fig. 4(a) illustrates s_Q and its Fourier transform for four different choices of $\mathbf{u}\in\{[1,0],[0,1],[1,1],[1,-1]\}$.

3.2. Short-term Fourier transform (STFT)

In the short-term Fourier transform, the filter function s is chosen to be the windowed complex exponential. This can be written as

$$s_S(\mathbf{x},\mathbf{u}) = w(\mathbf{x})e^{-j2\pi\mathbf{x}^T\mathbf{u}}, \tag{11}$$

where $j=\sqrt{-1}$ and $w(\mathbf{x}):\mathbb{R}^2\mapsto\mathbb{R}$ is a window function.

A canonical choice for w would be a Gaussian, but we consider also other possibilities. More precisely we apply the following three formulations for a window w:

$$w^U(\mathbf{x}) = \begin{cases} 1 & ||\mathbf{x}||_\infty \le m/2 \\ 0 & \text{otherwise}, \end{cases} \tag{12}$$

$$w^G(\mathbf{x}) = \begin{cases} e^{-\frac{1}{2\sigma_S^2}\mathbf{x}^T\mathbf{x}} & ||\mathbf{x}||_\infty \le m/2 \\ 0 & \text{otherwise}, \end{cases} \tag{13}$$

$$w^C(\mathbf{x}) = \begin{cases} e^{-\frac{1}{2\sigma_S^2}\mathbf{x}^T\mathbf{x}} & ||\mathbf{x}||_2 \le m/2 \\ 0 & \text{otherwise}, \end{cases} \tag{14}$$

where $\|\cdot\|_p$ denotes L^p norm, m defines the size of the local $m\times m$ neighborhood, and $\sigma_S\in\mathbb{R}$ is the standard deviation of the Gaussian envelope (we use $\sigma_S=(m-1)/4$). We denote the filter functions s_S corresponding to window functions (12), (13), and (14) as s_S^U, s_S^G, and s_S^C, respectively.

We note that window functions w are not required to satisfy $\int_{\mathbb{R}^2} w(\mathbf{x})d\mathbf{x}=1$, as in standard STFT, since it has no effect to the phase angle of $F(\mathbf{x},\mathbf{u})$. Moreover if s_S^G or s_S^C are applied in Eq. (6) the resulting estimates do not have invariance to uniform gray level

shifts. This requirement can be however enforced by normalizing the filters to have zero mean. Fig. 4(b–d) illustrate s_S^U, s_S^G, and s_S^C and their Fourier transforms using four different choices of $\mathbf{u}\in\{[1/m,0],[0,1/m],[1/m,1/m],[1/m,-1/m]\}$.

4. Local phase quantization

In principle, one could simply estimate the phase angles of F at frequencies $\mathbf{u}\in\{\mathbf{u}_1,\ldots,\mathbf{u}_L\}$, for which $\hat{h}(\mathbf{u})>0$, and use them as a blur-insensitive local descriptor. However, this would result in a real-valued feature vector of length L for every examined point $\mathbf{x}$, which as such is rather an impractical representation. Instead, some data reduction of the local descriptors is needed.

An important part our method is to implement the data reduction using the following quantizer:

$$Q(F(\mathbf{x},\mathbf{u}))=\operatorname{sgn}(\operatorname{Re}\{F(\mathbf{x},\mathbf{u})\})+2\operatorname{sgn}(\operatorname{Im}\{F(\mathbf{x},\mathbf{u})\}), \tag{15}$$

where $\operatorname{Re}\{\cdot\}$ and $\operatorname{Im}\{\cdot\}$ return the real and the imaginary parts of a complex number, respectively, and

$$\operatorname{sgn}(a)=\begin{cases}1 & a>0\\ 0 & \text{otherwise.}\end{cases} \tag{16}$$

The benefit of using this quantizer is that the phase angle does not need to be evaluated explicitly. A similar approach to quantization was also used in [57].

The quantizer results in a 2-bit integer representation ($Q\in\{0,1,2,3\}$) for a single frequency component at every point $\mathbf{x}$. For L coefficients the effective number of bits per point becomes $2L$. These representations can be concatenated into a codeword

$$C_{LPQ}(\mathbf{x})=\sum_{i=1}^{L}Q(F(\mathbf{x},\mathbf{u}_i))\cdot 2^{2(i-1)}, \tag{17}$$

which ranges from 0 to $2^{2L}-1$ and describes the local texture at location $\mathbf{x}$.

Now having the Local Phase Quantization (LPQ) codewords $C_{LPQ}(\mathbf{x})$, we can perform a simple histogramming operation to establish an estimate of their distribution. This is done by associating a bin for each of the 2^{2L} codewords and computing a vector $(m'_1,\ldots,m'_{2^{2L}})$, which contains the normalized mass m_j' in each bin. The normalization is performed to make the total mass of the histogram to equal 1.

To have some idea of the effects of quantizing phase angle to only quadrants we performed this operation to a test image, which is shown in Fig. 2(a). We computed a discrete Fourier transform of the original image, quantized the phase angle of frequency coefficients to quadrants, and computed reconstruction using an inverse discrete Fourier transform. The resulting image is illustrated in Fig. 2(b), which shows that despite the coarse quantization of phase angles, most of the information, from the image analysis point of view, is still recognizable.

We also note that it is possible to construct a rotation invariant version of the proposed descriptor. This is done by first running a filter that estimates the characteristic orientation at each location and then orienting the phase estimation filters accordingly. For more details about this technique, one can refer to [45].

5. Implementation

In this section, we explain the practical implementations of LPQ and also present a decorrelation scheme. Our implementations for all the presented LPQ variants are available online.[1]

5.1. Discretization

When applied to digital images, the integral representations must be discretized. Let $f(\mathbf{k}):\mathbb{Z}^2\mapsto\mathbb{R}$ be a discrete function representing a digital image. Define a discretized version of the convolution (6) as a sum

$$F(\mathbf{k},\mathbf{u})=\sum_{\mathbf{l}\in\mathbb{Z}^2}f(\mathbf{k}-\mathbf{l})s(\mathbf{l},\mathbf{u}), \tag{18}$$

where s is the filter function defined in Section 3. So in practice we just run a given $m\times m$ discrete filter over the input image.

Furthermore, since by design all the proposed filters, except for s_S^C, are separable, they can be efficiently evaluated by calculating 1-D convolutions for rows and columns successively. One must also keep in mind that if s_S^G or s_S^C are used, the filters must be normalized so that for given $\mathbf{u}$

$$\sum_{\mathbf{l}\in\mathbb{Z}^2}s(\mathbf{l},\mathbf{u})=0. \tag{19}$$

The resulting Local Phase Quantization (LPQ) codewords will be histogrammed to give the final texture descriptor vector $(m_{1'},\ldots,m_{'2^{2L}})$. In the case of discrete image function, we simply estimate the mass m_j' by counting the number of corresponding codewords in

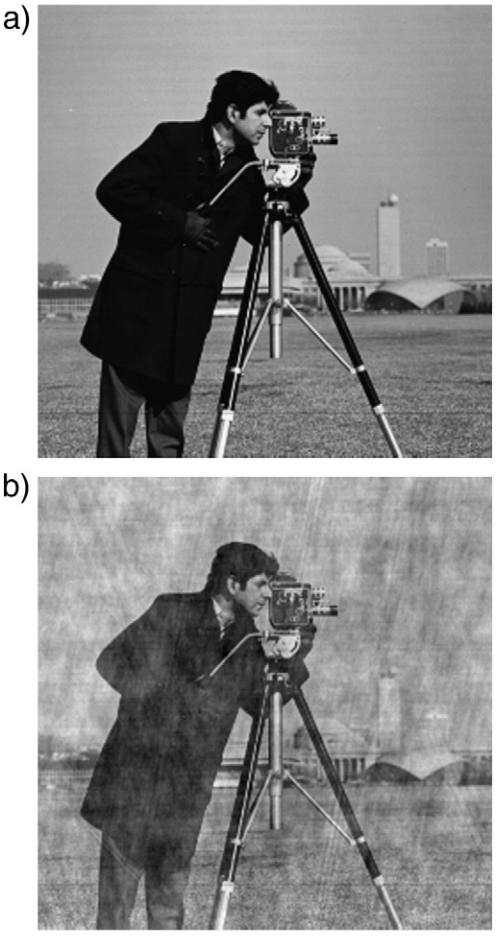


Fig. 2. The effect of coarse phase quantization: (a) original image, and (b) reconstructed image.

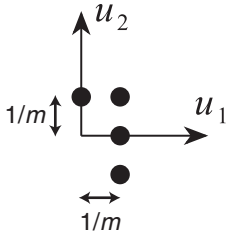


Fig. 3. The composition of the frequency samples used.

[1] http://www.cse.oulu.fi/Downloads/LPQMatlab.

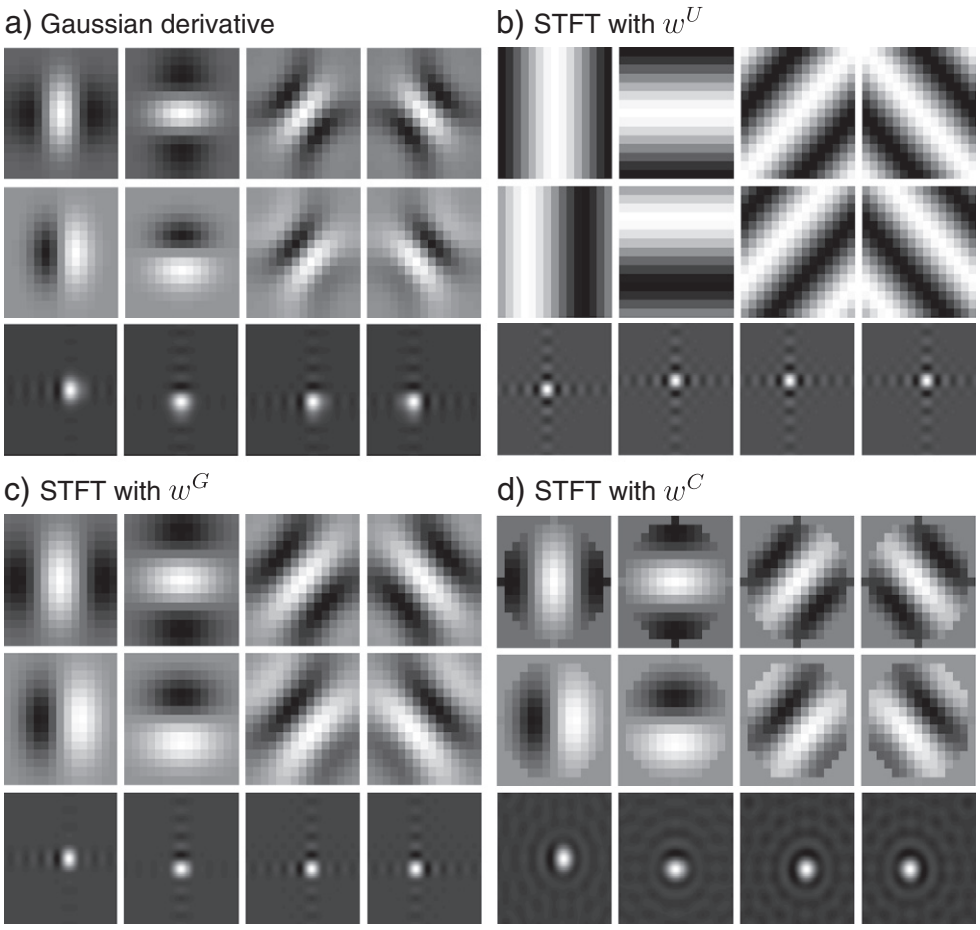


Fig. 4. The local estimation filters and their frequency responses for 15×15 neighborhood. In each subimage, the columns correspond to frequencies u_1, u_2, u_3, and u_4, respectively. First and second rows show the real and imaginary parts of the filters, and the third row shows the frequency responses (note that these are real valued due to symmetry).

the examined image area. The normalization is performed by dividing the mass is each bin with the total mass as

$$m_j' = \frac{m_j}{\sum_{j=1}^{2^{2L}} m_j}. \tag{20}$$

5.2. Parameter selection

In order to compute the LPQ descriptors, we have to select the frequency points $\{\mathbf{u}_1, \ldots, \mathbf{u}_L\}$. When selecting these we should keep in mind that the length of the feature vector becomes $2^{2L}-1$ and that the requirement $\hat{h}(\mathbf{u}_i) > 0$ should hold. The latter argument cannot be completely ensured for an unseen blur. However, most blur types have low pass characteristics, and hence the low frequency components are most likely to meet this requirement. Moreover, usually most of the signal energy is concentrated on the low frequencies. For this reason, we apply the four lowest frequency components corresponding to $\mathbf{u}\in\{[1,0],[0,1],[1,1],[1,-1]\}$ for Gaussian derivative filter and $\mathbf{u}\in\{[1/m,0],[0,1/m],[1/m,1/m],[1/m,-1/m]\}$ for STFT filters.

With the above selection we get 256 elements in the feature vector, which is comparable to other similar method and is still reasonable in many applications. Fig. 3 illustrates the composition of the selected points $\mathbf{u}_i$ in the frequency domain. We note that the other four low frequency components that are left out are complex conjugates of the selected ones. The resulting filters and their Fourier transforms are shown in Fig. 4.

Another parameter we have to select is the size of the local neighborhood m. When using small values in m, the low frequency components capture more details of the underlying image patch. Another benefit is that the blur has to be spatially invariant only in a small local scale. The drawback is the weaker insensitivity to blur, which is caused by the information lost at the patch borders. Hence the parameter m can be used to adjust the blur insensitivity versus discriminability of the method. The experimental results in Table 6 illustrate this behavior. The values we use for m range from 3 to 11 depending on the experiment.

5.3. Decorrelation

The scalar quantization (15) is efficient from the information theoretic point of view only if the coefficients F are statistically independent. In the case of correlated coefficients, vector quantization would be a more efficient technique. Another possibility however, used often in source coding, would be to decorrelate the samples before the quantization. Based on this idea, we will introduce an extension to the discretized basic LPQ method.

Let $\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_{m^2}$ denote the pixel positions in $m\times m$ neighborhood around origin. We can rewrite Eq. (18) for frequencies $\{\mathbf{u}_1, \ldots, \mathbf{u}_L\}$ as

$$\mathbf{F}_\mathbf{k} = \Psi \mathbf{f}_\mathbf{k}, \tag{21}$$

Table 1
Properties of the applied parts of Outex and Brodatz databases.

Database	Textures	Sample size	Samples
Outex01	24	64×64	2112
Outex02	24	32×32	8832
Brodatz	111	213×213	999

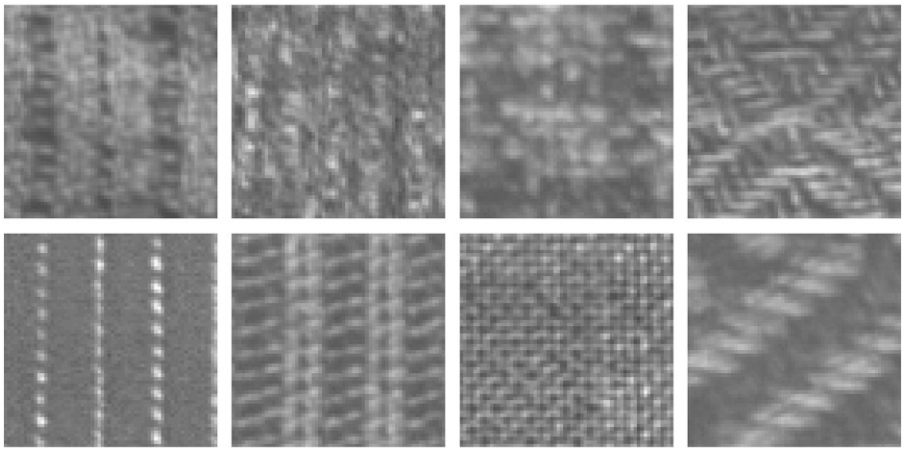


Fig. 5. Example images from Outex01 test set.

where

$$\mathbf{f}_{\mathbf{k}} = \left[f(\mathbf{l}_1 + \mathbf{k}), \cdots, f_{\mathbf{k}}(\mathbf{l}_{m^2} + \mathbf{k})\right]^T, \tag{22}$$

$$\mathbf{F}_{\mathbf{k}} = \left[\mathrm{Re}\left\{\tilde{\mathbf{F}}_{\mathbf{k}}\right\}, \mathrm{Im}\left\{\tilde{\mathbf{F}}_{\mathbf{k}}\right\}\right]^T \tag{23}$$

$$\tilde{\mathbf{F}}_{\mathbf{k}} = [F(\mathbf{k}, \mathbf{u}_1), \cdots, F(\mathbf{k}, \mathbf{u}_L)], \tag{24}$$

$$\Psi = \left[\mathrm{Re}\left\{\tilde{\Psi}\right\}, \mathrm{Im}\left\{\tilde{\Psi}\right\}\right], \quad \text{and} \tag{25}$$

$$\tilde{\Psi} = \begin{bmatrix} \{s(\mathbf{l}_1, \mathbf{u}_1)\} & \cdots & s(\mathbf{l}_{m^2}, \mathbf{u}_1)\} \\ \{s(\mathbf{l}_1, \mathbf{u}_2)\} & \cdots & s(\mathbf{l}_{m^2}, \mathbf{u}_2)\} \\ \vdots & \ddots & \vdots \\ \{s(\mathbf{l}_1, \mathbf{u}_L)\} & \cdots & s(\mathbf{l}_{m^2}, \mathbf{u}_L)\} \end{bmatrix}. \tag{26}$$

We will assume that the statistical dependence of the pixels $f(\mathbf{l})$ in $m \times m$ neighborhood can be adequately described by a covariance matrix $\mathbf{C}$ that is independent of the image position $\mathbf{k}$. Using Eq. (21) the corresponding covariance matrix of $\mathbf{F}_{\mathbf{k}}$ can be computed as $\mathbf{D} = \Psi \mathbf{C} \Psi^T$. In general, the $\mathbf{D}$ is not diagonal, which implies that the frequency coefficients are correlating. (Note that $\mathbf{D}$ is not diagonal even if $\mathbf{C}$ is.)

A standard approach for decorrelating sample vectors is to employ the whitening transform

$$\mathbf{G}_{\mathbf{k}} = \mathbf{V}^T \mathbf{F}_{\mathbf{k}}, \tag{27}$$

where $\mathbf{V}$ is an orthonormal matrix derived from the singular value decomposition (SVD): $\mathbf{D} = \mathbf{U} \Sigma \mathbf{V}^T$. Notice that $\mathbf{V}$ can be computed in advance if $\mathbf{C}$ is known. The scalar quantization for $\mathbf{G}_{\mathbf{k}}$ and the LPQ descriptor construction can be performed in the same way as in the original method.

There are several possibilities for obtaining an estimate of the matrix $\mathbf{C}$. One can, for example, use training data to learn a sample covariance. Another possibility is to use a model for the covariances and either learn or fix the parameters needed. We use a simple model based method, which assumes that the image patch $\mathbf{f}_{\mathbf{k}}$ is a realization of a random process, where the correlation coefficient between two pixel positions is exponentially related to their Euclidean distance.

Let ρ be the correlation between adjacent pixels (we use $\rho = 0.9$) and σ^2 be the variance of each sample. Without a loss of generality, we can assume that $\sigma^2 = 1$. Using this notation the ijth element of the matrix $\mathbf{C}$ (covariance between the positions $\mathbf{l}_i$ and $\mathbf{l}_j$) becomes

$$\mathbf{C}_{ij} = \rho^{||\mathbf{l}_i - \mathbf{l}_j||_2}. \tag{28}$$

If the estimated covariance $\mathbf{C}$ is adequately correct, the whitening transform may increase the information content of the LPQ descriptor. However, from the blur invariance point of view, the model (28) requires that the blur PSF is isotropic, which is particularly not true for motion blur. In practice, it might be useful to try out both approaches (with and without decorrelations) and to select the one that best fits to a given application. One can also apply different covariances $\mathbf{C}$, which might be estimated from the data. Moreover, if more specific assumptions about the PSF can be made (i.e. it is known to be motion blur), one can further optimize the decorrelation model. For more details and examples, one can refer to [56].

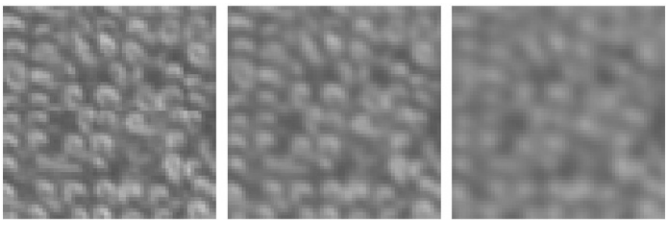


Fig. 6. Three versions of the same texture, blurred using circular PSF and radii 0, 1, and 2, respectively.

6. Experiments with benchmark datasets

To our knowledge, none of the benchmark texture recognition datasets contain blurred images. Therefore, in this section, the blur was generated by convolving the images with point spread functions[2]. The selected PSFs include the examples illustrated in Fig. 1 as well as the formulations that do not fulfill the assumption of central symmetry. The experiments involving real blur are introduced later in Section 8. The blurring was always applied only to the test images and the training was performed using the original sharp textures.

We use two well known benchmark datasets: Outex[3] [51] and Brodatz.[4] The main properties and some example images of both databases are illustrated in Table 1 and Figs. 5 and 6, respectively. In this paper, we refer to Outex test sets Outex_TC_00001-00002 as Outex01–02 for short.

The proposed method was compared against several baseline methods: Local Binary Patterns (LBP)[5] [17], Gabor filter banks[6] [42], VZ-MR8 filters [2] [7], VZ-joint [19], BIF-columns (BIFc) [3], and Local Ternary Patterns (LTP) [20]. For all LBP and LTP methods, we applied

[2] Matlab procedure for generating blurred test images is available at http://www.cse.oulu.fi/Downloads/LPQMatlab.
[3] http://www.outex.oulu.fi/.
[4] http://www.ux.uis.no/~tranden/brodatz.html.
[5] http://www.cse.oulu.fi/MVG/Downloads/LBPMatlab.
[6] http://vision.ece.ucsb.edu/texture/software/.
[7] http://www.robots.ox.ac.uk/~vgg/research/texclass/.

Table 2
Texture classification accuracies using the Outex02 test set without blur and k-NN classifier. The best result is shown with bold numbers.

	LPQ_u	LPQ_{ud}	LPQ_g	LPQ_{gd}	LPQ_q	LPQ_{qd}	LBP
Accuracy	87.3%	93.3%	84.8%	92.9%	88.1%	**93.7%**	90.2%
	LBP_{u2}	LBP_{riu2}	Gabor	BIFc	VZ-joint	VZ-MR8	LTP
Accuracy	89.6%	64.7%	90.2%	70.6%	85.5%	71.8%	**94.8%**

Table 3
Texture classification accuracies using the Outex02 test set without blur and SVM classifier. The best result is shown with bold numbers.

	LPQ_u	LPQ_{ud}	LPQ_g	LPQ_{gd}	LPQ_q	LPQ_{qd}	LBP
Accuracy	92.1%	96.1%	91.5%	96.1%	92.9%	96.1%	93.9%
	LBP_{u2}	LBP_{riu2}	Gabor	BIFc	VZ-joint	VZ-MR8	LTP
Accuracy	93.2%	66.7%	94.6%	84.1%	85.5%	72.5%	**97.2%**

8 samples and a radius $(m-1)/2$, which results in the similar spatial extent as LPQ. For VZ methods we learned 10 cluster centers per class and in VZ-joint we further apply 7×7 patches as in [3]. LBP, Gabor, and VZ-MR8 descriptors were computed using the programs available online. VZ-joint, BIF-columns, and LTP were implemented by us based on the descriptions in the original papers.

The classification was performed using k-nearest neighbor (k-NN) and SVM with RBF kernel. The value of k was dependent on the dataset and we used $k=1$ with Brodatz, $k=3$ with Outex01, and $k=15$ with Outex02. We applied χ^2 distance for all methods except Gabor features, for which we used the distance proposed in the original paper. The results reported with Outex dataset refer to the average accuracy over all predefined train–test splits. For the Brodaz database we apply the procedure described in [52].

The LPQ variants corresponding to local frequency estimation filters s_Q, s_S^U, s_S^G, are denoted as LPQ_q, LPQ_u, and LPQ_g, respectively. If the decorrelation scheme, presented in Section 5.3, is applied, we add a letter d to the subscript (e.g. LPQ_{qd}). We also use three variants of LBP, namely standard LBP, uniform pattern LBP, and rotation invariant LBP. These variants are denoted as LBP, LBP_{u2}, and LBP_{riu2}, respectively.

In the first experiment, we use Outex02 dataset, neighborhood size $m=3$ (for LBP, LTP, and LPQ), and no blur at all. The classification results are shown in Tables 2 and 3. The best result is achieved by LTP followed closely by decorrelated LPQ variants, LBP, and Gabor filters. For all methods SVM classifier results to equal or better performance compared to k-NN. The results illustrate that LPQ descriptors have high discriminability also with sharp images, although the main design criterion was tolerance to blur.

In the next experiment, we use Outex01 and Brodatz datasets, and introduce different blur types and strengths to the test images. The PSFs applied were circular of radii $\{0, 0.25, \ldots, 2\}$, line (motion) with lengths $\{0, 0.5, \ldots, 4\}$, and Gaussian with standard deviations $\{0.5, 0.75, \ldots, 1.5\}$. Fig. 6 shows three examples of the same image, blurred with circular PSF of radii 0, 1, and 2, respectively. In this test, we use $m=7$ for LBP, LTP, and all LPQ variants. The achieved classification results are illustrated in Figs. 7–10.

Without blur, the results of LPQ, LBP, LTP, Gabor, BIF columns, and VZ joint are very close to each other. When the level of the blur increases, the performance changes depending on the blur type. In the case of circular and Gaussian blurs the decorrelated LPQ_{ud} and LPQ_{qd} have the best performance, losing relatively little from their initial accuracy. In the case of line blur, the non-decorrelated LPQ variants outperform their decorrelated counterparts, which is an expected behavior since the PSF of line blur is not isotropic as assumed in the decorrelation. From the baseline methods VZ-joint and BIF-columns appear to be relatively tolerant to line blur.

A key aspect of our method was to allow PSF to vary spatially. To have some idea of this effect we added a test case where we roughly mimic a scene with different focus depths. Here the test images of Outex01 are divided into $3\times3=9$ blocks which are then blurred using circular PSF of radii $\{0,0,0,1,1,1,2,2,2\}$ in random order. The classification performances, using $m=3$, are shown in Tables 4 and 5. The results are very similar to those with spatially invariant blur, which indicates that the methods are very tolerant for this type of perturbations (Table 4).

The window size m has an impact to the blur tolerance of the methods. We tested this effect by sliding m from 3 to 11 in LPQ_{ud} operator, and observing the changes in the classification accuracy. The experiment was performed using Outex01 database and circular PSF of radii $\{0, 0.25, \ldots, 3\}$. The result shown in Table 6 clearly illustrates the fact that the stronger the blur is the larger the window we need to achieve the best accuracies (bold letters). However, with the light blur the performance is, on the other hand, compromised by enlarging the window size. This indicates the trade-off between discriminability and blur insensitivity of LPQ with different m values.

We conclude this set of experiments by illustrating the sensitivity to the requirement of centrally symmetric PSF. In the first test we use rectangular $P\times P$ PSFs with $P=\{1,2,3,4,5\}$ containing positive

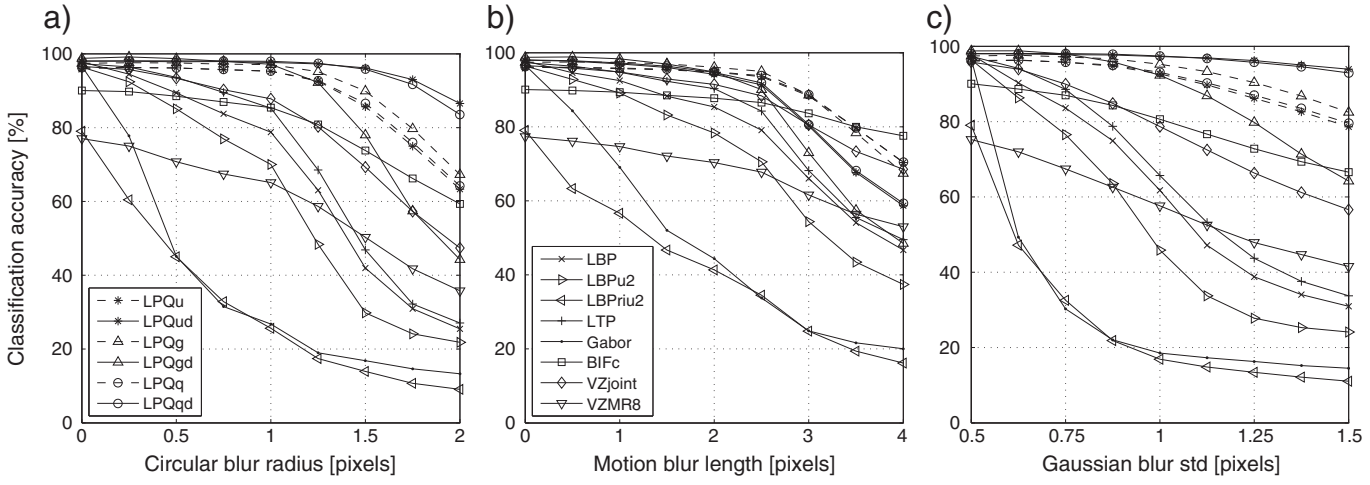


Fig. 7. Classification accuracies using Outex01 data, k-NN classifier, and blur with different types and strengths: (a) circular blur, (b) line blur, and (c) Gaussian blur. Due to space constraints, the legend is split into two parts.

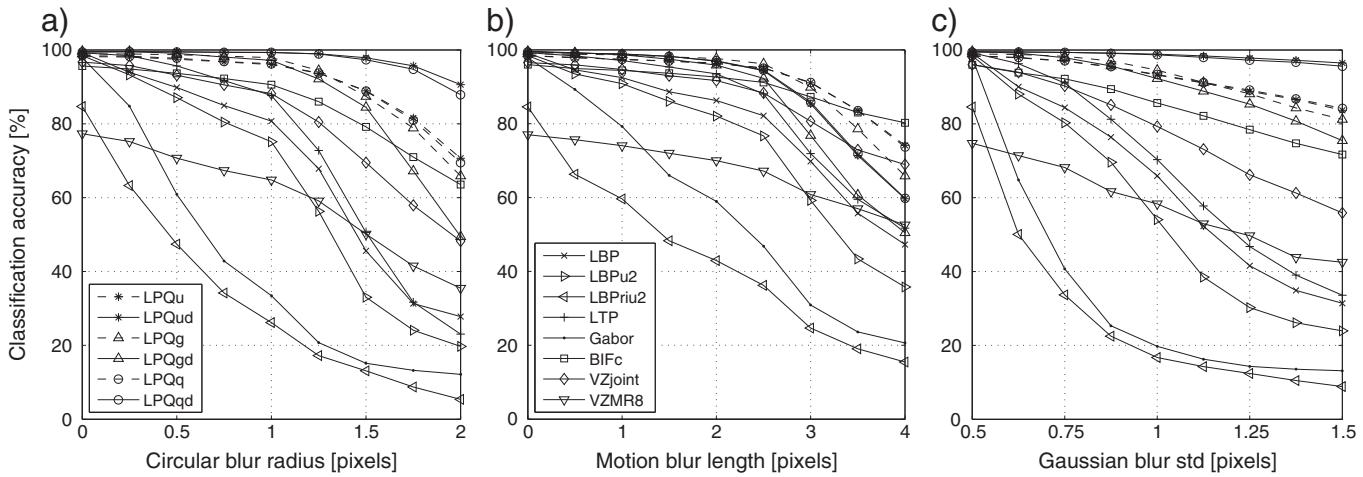


Fig. 8. Classification accuracies using Outex01 data, SVM classifier, and blur with different types and strengths: (a) circular blur, (b) line blur, and (c) Gaussian blur. Due to space constraints, the legend is split into two parts.

random numbers (the sum of the PSFs was normalized to one). The results are shown in Fig. 11. Comparing these to the results of symmetric circular PSF in Fig. 7(a) and 8(a), we can notice some drop in performance for all the methods. However, still especially LPQ_u and LPQ_{ud} operators perform very well despite of the violation in the assumption of centrally symmetric PSF (Fig. 6).

In the second test, we use Outex01 and the eight real blur kernels estimated in [62][8]. Since Outex images are smaller than those in [62], we need to adjust the scale of the blur kernels with respect to the images. Otherwise the blur strength would be greatly emphasized. The scaling is done by first upsampling the Outex images by a factor of four, then applying blurring using the original kernels, and finally downsampling the results back to the initial size. The classification results are displayed in Table 7. In all but two cases the best performance is given by LPQ variants. The best mean performance (84.8%) is achieved by LPQ_{qd}, followed closely by LPQ_{ud} (84.1%). In general we conclude that LPQ_{ud} achieves the best average performance over all tested databases and blur PSFs.

7. Application: dynamic texture recognition

Another type of texture recognition problem is encountered in moving scenes. Such textures are referred as dynamic textures in the literature, and they can be seen as image sequences that exhibit certain stationarity properties in time [59]. Some real world examples of dynamic textures are fire, moving clouds, a waving flag, and sea waves.

Traditional texture representation methods, have provided a good starting point for describing dynamic textures. One efficient approach was introduced in [58], which concatenates LBP descriptors that are computed in three orthogonal planes in small spatio-temporal cubes. To test LPQ in this application, we simply replace LBP with LPQ_{ud} in the framework of [58].

The performance was evaluated using dynamic texture database DynTex++ [60], which is a new database compiled from the original DynTex database [61][9]. DynTex++ consists of 3600 dynamic textures of size $50\times50\times50$. The textures are divided into 36 classes, each holding 100 videos.

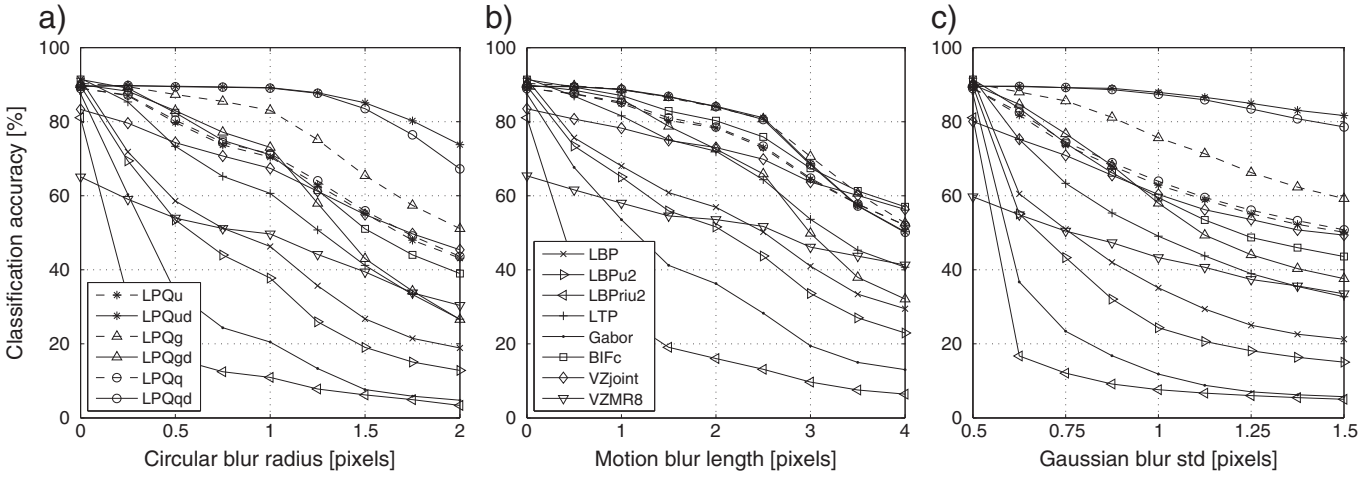


Fig. 9. Classification accuracies using Brodatz data, k-NN classifier, and blur with different types and strengths: (a) circular blur, (b) line blur, and (c) Gaussian blur. Due to space constraints, the legend is split into two parts.

[8] http://www.wisdom.weizmann.ac.il/~levina/papers/LevinEtalCVPR09Data.rar.

[9] http://www.cwi.nl/projects/dyntex/.

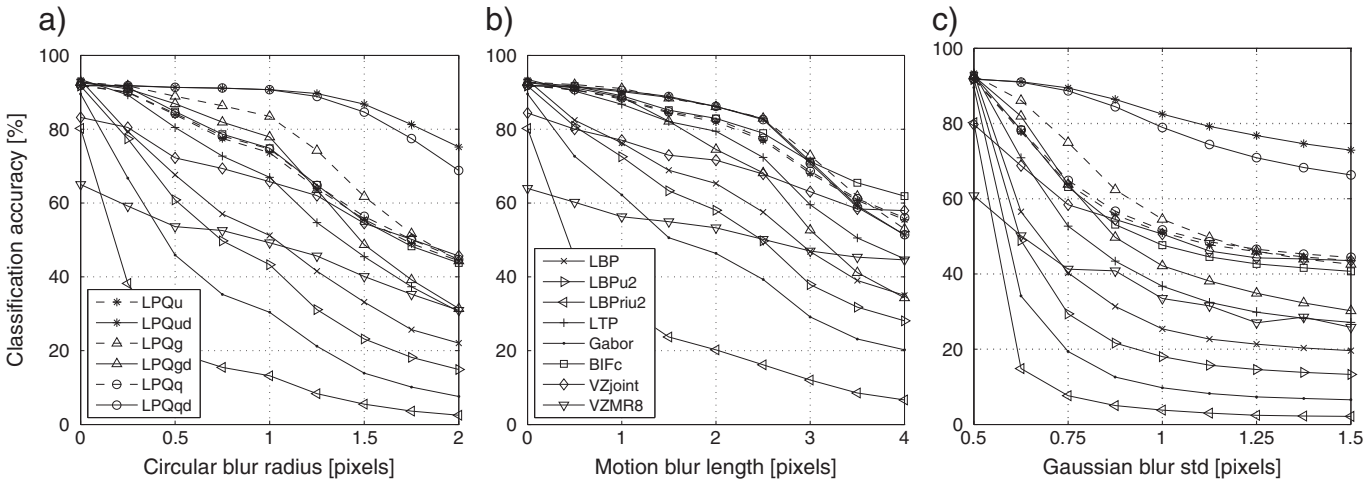


Fig. 10. Classification accuracies using Brodatz data, SVM classifier, and blur with different types and strengths: (a) circular blur, (b) line blur, and (c) Gaussian blur. Due to space constraints, the legend is split into two parts.

Table 4
Outex01 texture classification accuracies for k-NN classifier and randomly varying circular blur radius {0,1,2} in different image blocks ($3\times3=9$ blocks). The best result is shown with bold numbers.

	LPQ_u	LPQ_{ud}	LPQ_g	LPQ_{gd}	LPQ_q	LPQ_{qd}	LBP
Accuracy	91.0%	95.6%	92.6%	87.1%	91.3%	**95.9%**	69.2%
	LBP_{u2}	LBP_{riu2}	Gabor	BIFc	VZ-joint	VZ-MR8	LTP
Accuracy	56.8%	28.1%	55.8%	82.3%	87.0%	59.2%	70.8%

Table 5
Outex01 texture classification accuracies for SVM classifier and randomly varying circular blur radius {0,1,2} in different image blocks ($3\times3=9$ blocks). The best result is shown with bold numbers.

	LPQ_u	LPQ_{ud}	LPQ_g	LPQ_{gd}	LPQ_q	LPQ_{qd}	LBP
Accuracy	93.7%	**97.9%**	93.3%	90.1%	93.8%	97.7%	74.2%
	LBP_{u2}	LBP_{riu2}	Gabor	BIFc	VZ-joint	VZ-MR8	LTP
Accuracy	62.9%	29.3%	60.3%	88.6%	85.5%	59.2%	74.5%

In the evaluation each class is randomly divided into train and test sets of equal size. Every test is repeated 20 times and average recognition rates are reported. The classification is done using nearest neighbor classifier with χ^2 distance. We use both sharp and spatially blurred dynamic textures, where the blur was generated in a similar manner as in the previous experiment. The performance baseline was provided by LBP-TOP [58].

Table 8 shows the obtained recognition accuracies with different blur types and strengths. The results using LPQ instead of LBP are consistently better over the different experiments. Both methods result in a similar performance with sharp textures, but as soon as blurring appears, LPQ based descriptor starts to perform clearly better.

8. Application: face recognition

Blurring is one of the most commonly encountered degradations also in face image analysis, especially in face recognition. It is often present because of the motion of the subject or the camera, misfocused optics, or low quality of the imaging device such as a web camera. For example, in the Face Recognition Grand Challenge dataset [53], many of the "uncontrolled still" images appear blurred since the auto-focus of the digital pocket camera did not focus into the face area. For these reasons, the blur insensitive descriptors, including LPQ, offer a prominent tool for face analysis, which was also observed in [28,48].

To use local phase quantization for face description, we apply the following procedure from [54]. The face image is first labeled using the LPQ operator with $m=7$. Then, the label image is divided into 8×8 non-overlapping rectangular regions and a histogram of labels is computed independently within each region. Finally, the histograms from different regions are concatenated to build a global description of the face. The recognition is performed using the obtained histograms and a nearest neighbor classifier with χ^2-distance. As a baseline we use the well known LBP based technique proposed in [54].

In the face recognition experiments, we applied two databases, the CMU PIE (Pose, Illumination, and Expression) dataset [55] and the Face Recognition Grand Challenge (FRGC) experiment 1.0.4 [53]. In both cases, the images were first converted to gray-scale and then registered using given ground truth eye positions. In the CMU PIE experiment, the images were further cropped to size 128×128 pixels. The image size in the FRGC test was 130×150 pixels. The images in CMU PIE do not contain much blur, and this test serves as a performance baseline of LPQ descriptors in face recognition. The FRGC dataset on the other hand contain many types of distortions including blur. Some examples of the images in FRGC dataset are shown in Fig. 12.

Table 6
Texture classification accuracies using Outex01, LPQ_{ud} descriptor, and different window and blur sizes. The best result for each blur radius is shown with bold numbers.

m	Blur radius												
	0	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25	2.50	2.75	3.00
3	98.9%	94.0%	78.2%	73.5%	70.3%	61.9%	45.6%	28.1%	20.9%	17.3%	13.6%	12.8%	12.1%
5	**99.1%**	**99.2%**	**99.2%**	**98.9%**	**98.6%**	95.8%	83.8%	64.7%	49.4%	35.3%	22.4%	18.3%	17.3%
7	97.9%	97.9%	97.7%	97.8%	97.7%	**97.2%**	**96.0%**	**92.9%**	**86.2%**	69.5%	51.4%	37.7%	31.6%
9	93.1%	93.3%	93.3%	93.2%	93.0%	92.3%	90.9%	88.8%	86.0%	**78.8%**	**67.6%**	57.2%	48.7%
11	84.7%	84.5%	84.4%	84.2%	84.1%	83.7%	82.6%	81.0%	78.3%	73.6%	67.3%	**61.9%**	**57.3%**

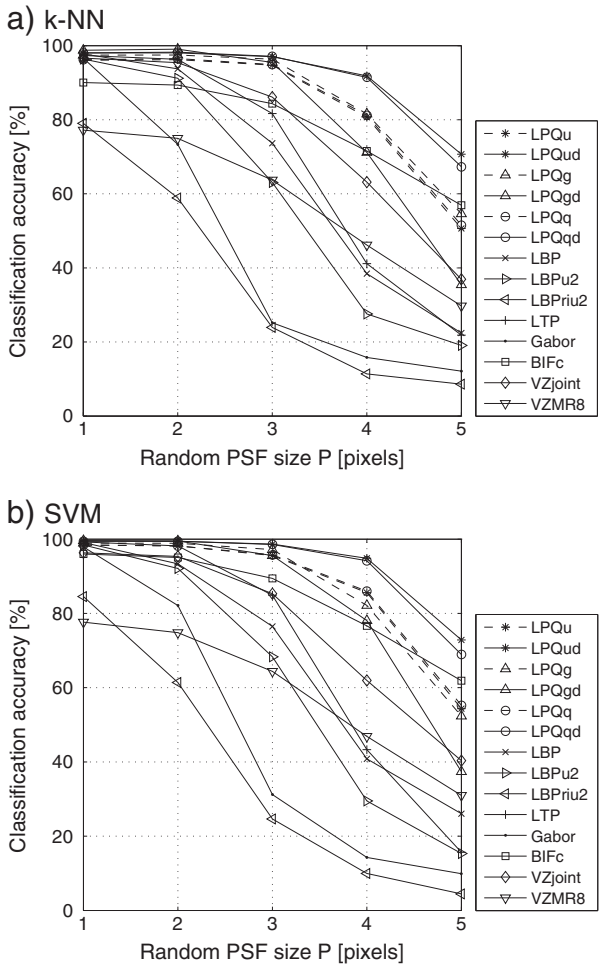
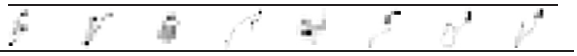


Fig. 11. Classification accuracies using Outex01, random PSF of size P × P, and different classifiers.

In the case of CMU PIE dataset we selected a set of 23 images of each of the 68 subjects in the database. Two of these are taken with the room lights on and the remaining 21 each with a flash at varying positions. One image per person was used as a gallery image, and the remaining 22 images were used as probes for the tested methods. The classification results are shown in Table 9. We can notice that the margin between the LPQ variants is almost negligible, but the difference to the LBP is rather clear. The LPQ variants provide to be well suited for face description and also prove tolerance to the illumination changes.

Table 7
Classification accuracies using Outex01, k-NN classifier, and blur kernels from [62].

Method	Blur kernel							
								
LPQ_u	95.6%	94.1%	96.4%	57.0%	95.9%	92.5%	65.9%	59.1%
LPQ_{ud}	95.3%	94.1%	97.8%	62.6%	97.8%	93.2%	70.3%	61.3%
LPQ_g	**97.1%**	**95.2%**	97.5%	60.2%	97.4%	93.1%	62.0%	54.2%
LPQ_{gd}	94.9%	91.3%	**99.0%**	**72.5%**	**99.0%**	93.0%	61.9%	53.5%
LPQ_q	95.7%	94.4%	96.4%	57.2%	96.1%	92.7%	65.7%	58.7%
LPQ_{qd}	95.5%	94.4%	97.9%	64.2%	97.9%	**93.5%**	71.9%	63.1%
LBP	84.2%	78.4%	92.5%	51.3%	92.1%	77.4%	47.9%	42.9%
LBP_{u2}	76.9%	69.9%	89.6%	46.0%	89.0%	67.9%	39.5%	36.6%
LBP_{riu2}	40.1%	33.2%	55.9%	25.3%	54.2%	32.0%	16.1%	14.4%
Gabor	37.6%	31.9%	64.9%	13.4%	62.8%	30.2%	16.6%	15.6%
BIFc	86.9%	85.7%	89.0%	71.8%	89.3%	86.8%	**76.2%**	**74.0%**
VZ-joint	90.0%	89.2%	94.8%	45.5%	95.3%	88.1%	70.8%	66.3%
VZ-MR8	69.3%	66.4%	74.2%	37.0%	73.7%	66.2%	52.2%	50.6%
LTP	89.4%	87.7%	95.9%	47.7%	95.8%	85.4%	56.8%	51.1%

The best result is shown with bold numbers.

Table 8
Recognition accuracies using DynTex++ and different blur types and strengths. LBP and LPQ refer to framework [58] with corresponding texture descriptor.

(a) Circular PSF			(b) Gaussian PSF			(c) Line PSF		
Radius	LPQ	LBP	Std	LPQ	LBP	Length	LPQ	LBP
0	95%	95%	0	95%	95%	0	95%	95%
0.5	92%	78%	0.5	94%	73%	1	94%	93%
1.0	83%	57%	1.0	72%	47%	2	92%	83%
1.5	59%	41%	1.5	44%	31%	3	88%	81%
2.0	48%	32%	2.0	31%	23%	4	64%	58%

In the second experiment, we compared the descriptors using the FRGC experiment 1.0.4. The dataset is divided into training, probe and gallery sets which have no overlap. The probe and gallery images represent 152 subjects, and there are 1 gallery image and 2–7 probes per subject, totaling 152 images in the gallery set and 608 images in the probe set. The gallery images are taken with good quality camera under controlled conditions whereas the probe images are acquired with a pocket digital camera in uncontrolled conditions. The uncontrolled probes contain variation in lighting, facial expression, and blur. The fact that the probes contain significant out-of-focus blur makes this dataset very interesting for experiments with LPQ descriptors. We also note that the blur was not controlled in any way and the PSF would probably vary over the image.

To reduce the effect of lighting variations in facial images, Tan and Triggs proposed an illumination normalization procedure consisting of gamma correction, difference of Gaussian filtering, and contrast equalization [20]. In their experiments, they showed that this method significantly increases the recognition results of LBP based face description on the FRGC dataset. Based on these observations, we applied the same approach in our experiments with FRGC 1.0.4 data.

The recognition rates in the FRGC experiment are reported in Table 10. The results without preprocessing show that the LPQ operators handle the blur and other variations in the probe set better than LBP. The differences between LPQ operators are larger than in the experiments with CMU PIE images. With illumination normalized images, the LPQ_{ud} operator gives the highest recognition rate of 74.5% followed by LPQ_u, 71.9%. The LPQ_g and LPQ_q operators obtain slightly higher performance than their decorrelated counterparts. In general the preprocessing was observed to significantly improve all tested methods.

9. Conclusions

Image descriptors need to be discriminative, but also tolerant to common degradations such as blurring. In this paper, we have shown a blur insensitive approach to encode the local phase

Fig. 12. A sample gallery image and two probe images from FRGC experiment 1.0.4. The blurring seen in the probe images is a real out-of-focus blur.

Table 9
Recognition rates for the CMU PIE dataset using LPQ and LBP.

	LPQ_u	LPQ_{ud}	LPQ_g	LPQ_{gd}	LPQ_q	LPQ_{qd}	LBP
Accuracy	99.3%	99.4%	99.0%	99.1%	99.3%	99.1%	92.7%

Table 10
Recognition rates for the FRGC 1.0.4 dataset using LPQ and LBP, with and without illumination normalization preprocessing.

	LPQ_u	LPQ_{ud}	LPQ_g	LPQ_{gd}	LPQ_q	LPQ_{qd}	LBP
Without preprocessing	48.8%	45.9%	47.5%	38.0%	48.2%	41.8%	32.6%
With preprocessing	71.9%	**74.5**%	70.4%	68.6%	69.2%	65.6%	63.8%

information into discriminative features. The proposed approach, called Local Phase Quantization (LPQ), can be formulated using any of the phase estimation approaches presented in the literature. In our evaluation, especially in the case of heavy blur, we found the short-term Fourier transform with uniform window function to result in the best performance (LPQ_u and LPQ_{ud}).

We also presented a data decorrelation scheme, which could be embedded into the introduced descriptor construction. The decorrelation was found to increase the performance in the case of sharp image and if the blur PSF was isotropic. In face recognition, the effect of decorrelation was also dependent on the selected phase estimation method. Hence, we suggest that if the blur level is low or it is known to be isotropic, it would be better to use decorrelation. In the other cases, we suggest the non-decorrelated version. We expect that, with some new decorrelation approaches, it would be possible to further enhance the descriptor performance.

The comparisons with the state-of-the-art texture descriptors showed that the proposed LPQ method performs clearly better in the case of blur. Surprisingly, LPQ produced highly competitive results with sharp textures and face images. We further noticed that the blur tolerance is dependent on the extent of LPQ operator. However, even with fixed filters one can deal with a relatively large range of blur lengths as well as sharp images.

Acknowledgments

This work was supported by the Academy of Finland (grant nos. 127702 and 128975) and the EC project IST-214324 MOBIO.

References

[1] M. Mirmehdi, X. Xie, J. Suri, Handbook of Texture Analysis, Imperial College Press, 2008.
[2] M. Varma, A. Zisserman, A statistical approach to texture classification from single images, Int. J. Comput. Vision 62 (1) (2005) 61–81.
[3] M. Crosier, L.D. Griffin, Using basic image features for texture classification, Int. J. Comput. Vision 88 (3) (2010) 447–460.
[4] J. Chen, A. Kundu, Rotation and gray scale transform invariant texture identification using wavelet decomposition and hidden Markov model, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994) 208–214.
[5] S. Fountain, T. Tan, Efficient rotation invariant texture features for content-based image retrieval, Pattern Recognit. 31 (1998) 1725–1732.
[6] J. Mao, A. Jain, Texture classification and segmentation using multiresolution simultaneous autoregressive models, Pattern Recognit. 25 (1992) 173–188.
[7] W.R. Wu, S.C. Wei, Rotation and gray-scale transform-invariant texture classification using spiral resampling, sub-band decomposition and hidden Markov model, IEEE Trans. Image Process. 5 (1996) 1423–1434.
[8] T. Ahonen, J. Matas, C. He, M. Pietikäinen, Rotation invariant image description with local binary pattern histogram Fourier features, Proc. Scandinavian Conference on Image Analysis, 2009, pp. 61–70.
[9] F. Cohen, M. Patel, Classification of rotated and scaled texture images using Gaussian random field models, IEEE Trans. Pattern Anal. Mach. Intell. 13 (1991) 192–202.
[10] M. Mellor, B.=.W. Hong, M. Brady, Locally rotation, contrast, and scale invariant descriptor for texture analysis, IEEE Trans. Pattern Anal. Mach. Intell. 30 (1) (2008) 52–61.
[11] V. Manian, R. Vasquez, Scaled and rotated texture classification using a class of basis functions, Pattern Recognit. 31 (1998) 1937–1942.
[12] S. Lazebnik, C. Schmid, J. Ponce, A sparse texture representation using local affine regions, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1265–1278.
[13] S. Chang, L. Davis, S. Dunn, J. Eklundh, A. Rosenfeld, Texture discrimination by projective invariants, Pattern Recognit. Lett. 5 (1987) 337–342.
[14] D. Chetverikov, Pattern regularity as a visual key, Image Vision Comput. 18 (2000) 975–985.
[15] J. Daugman, Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two dimensional visual cortical filters, J. Opt. Soc. Am. 2 (7) (1985) 1160–1169.
[16] M. Varma, A. Zisserman, Texture Classification: Are Filter Banks Necessary? Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2003.
[17] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[18] M. Varma, A. Zisserman, Classifying Images of Materials: Achieving Viewpoint and Illumination Independence, Proc. European Conference on Computer Vision, 2002.
[19] M. Varma, A. Zisserman, A statistical approach to material classification using image patch exemplars, IEEE Trans. Pattern Anal. Mach. Intell. 31 (11) (2009) 2032–2047.
[20] X. Tan, B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, Proc. Analysis and Modeling of Faces and Gestures, 2007.
[21] T. Ahonen, M. Pietikäinen, Soft histograms for local binary patterns, Finnish Signal Processing Symposium, 2007.
[22] J. van de Weijer, C. Schmid, Blur robust and color constant image description, Proc. IEEE International Conference on Image Processing, 2006, pp. 993–996.
[23] J. Flusser, T. Suk, Degraded image analysis: An invariant approach, IEEE Trans. Pattern Anal. Mach. Intell. 20 (6) (1998) 590–603.
[24] H. Zhang, H. Shu, G. Han, G. Coatrieux, L. Luo, J. Coatrieux, Blurred image recognition by Legendre moment invariants, IEEE Trans. Image Process. 19 (3) (2010) 596–611.
[25] V. Ojansivu, J. Heikkilä, A method for blur and similarity transform invariant object recognition, Proc. International Conference on Image Analysis and Processing, 2007, pp. 583–588.
[26] N. Joshi, S. Kang, L. Zitnick, R. Szeliski, Image deblurring using inertial measurement sensors, ACM Trans. Graphics 29 (4) (2010).
[27] W. Oliver, J. Sivic, A. Zisserman, J. Ponce, Non-uniform deblurring for shaken images, Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2010.
[28] C. Chan, J. Kittler, N. Poh, T. Ahonen, M. Pietikäinen, (Multiscale) local phase quantization histogram discriminant analysis with score normalisation for robust face recognition, Proc. IEEE Workshop on Video-Oriented Object and Event Classification, 2009, pp. 633–640.
[29] Masashi N., Hadid A., Hidenori T., Shotton J., Kozakaya T., and Yamaguchi O., Facial Deblur Inference using Subspace Analysis for Recognition of Blurred Faces, IEEE Trans. Pattern Anal. Mach. Intell., to appear 2011.
[30] Z. Lei, T. Ahonen, M. Pietikäinen, L. Stan, Local frequency descriptor for low-resolution face recognition, Proc. IEEE Conference on Automatic Face and Gesture Recognition, 2011, pp. 161–166.
[31] B. Jiang, M. Valstar, M. Pantic, Action unit detection using sparse appearance descriptors in space-time video volumes, Proc. 9th IEEE Conference on Automatic Face and Gesture Recognition, 2011, pp. 314–321.
[32] A. Dhall, A. Asthana, R. Goecke, T. Gedeon, Emotion Recognition Using PHOG and LPQ features, Proc. Workshop on Facial Expression Recognition and Analysis Challenge, 2011.
[33] L. Nanni, A. Lumini, S. Brahnam, Local Binary Patterns variants as texture descriptors for medical image analysis, Artif. Intell. Med. (2010) 117–125.
[34] L. Nanni, S. Brahnam, A. Lumini, A very high performing system to discriminate tissues in mammograms as benign and malignant, Expert Syst. Appl. 39 (2) (2012) 1968–1971.
[35] S. Brahnam, L. Nanni, J.=.Y. Shi, A. Lumini, Local phase quantization texture descriptor for protein classification, Proc. International Conference on Bioinformatics and Computational Biology, 2010, pp. 159–165.
[36] G. Granlund, H. Knutsson, Signal Processing for Computer Vision, Kluwer Academic, 1995.
[37] L. Zhang, L. Zhang, Z. Guo, D. Zhang, Monogenic-LBP: a new approach for rotation invariant texture classification, Proc. International Conference on Image Processing, 2010, pp. 2677–2680.
[38] B. Zhang, S. Shan, X. Chen, W. Gao, Histogram of Gabor Phase Patterns (HGPP): A Novel Object Representation Approach for Face Recognition, IEEE Trans. Image Process. 16 (1) (2007) 57–68.
[39] S. Fischer, J. Bigun, Texture boundary tracking with Gabor phase, in: G. Borgefors (Ed.), Theory and Applications of Image Analysis, II1995, pp. 101–112.
[40] F. Zhou, J. Feng, Q. Shi, Texture feature based on local Fourier transform, Proc. International Conference on Image Processing, 2001, pp. 610–613.
[41] W. Zhang, S. Shan, L. Qing, X. Chen, W. Gao, Are Gabor phases really useless for face recognition? Pattern Anal. Appl. 12 (3) (2009) 301–307.
[42] B. Manjunathi, W. Ma, Texture features for browsing and retrieval of image data, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 837–842.
[43] V. Ojansivu, J. Heikkilä, Blur insensitive texture classification using local phase quantization, Proc. Image Signal Process., 2008, pp. 236–243.
[44] J. Heikkilä, V. Ojansivu, Methods for local phase quantization in blur-insensitive image analysis, Proc. International Workshop on Local and Non-Local Approximation in Image Processing, 2009, pp. 104–111.
[45] V. Ojansivu, E. Rahtu, J. Heikkilä, Rotation invariant blur insensitive texture analysis using local phase quantization, Proc. 19th International Conference on Pattern Recognition, 2008.
[46] M. Banham, A. Katsaggelos, Digital image restoration, IEEE Signal Process. Mag. 14 (2) (1997) 24–41.
[47] Z. Wang, E.P. Simoncelli, Local phase coherence and the perception of blur, Proc. Adv. Neural Information Processing Systems, 2003, pp. 786–792.
[48] T. Ahonen, E. Rahtu, V. Ojansivu, J. Heikkilä, Recognition of blurred faces using local phase quantization, Proc. 19th International Conference on Pattern Recognition, 2008.

[49] D. Boukerroui, J. Noble, M. Brady, On the choice of band-pass quadrature filters, J. Math. Imaging Vision 21 (2004) 53–80.
[50] M. Felsberg, G. Sommer, The monogenic signal, IEEE Trans. Signal Process. 49 (12) (2001) 3136–3144.
[51] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, S. Huovinen, Outex — new framework for empirical evaluation of texture analysis algorithms, Proc. International Conference on Pattern Recognition, 2002, pp. 701–706.
[52] J. Zhang, M. Marszałek, S. Lazebnik, C. Schmid, Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study, Int. J. Comput. Vision 73 (2) (2006) 213–238.
[53] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 947–954.
[54] T. Ahonen, A. Hadid, M. Pietikäinen, Face description with local binary patterns: Application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 2037–2041.
[55] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression database, IEEE Trans. Pattern Anal. Mach. Intell. 25 (12) (2003) 1615–1618.
[56] J. Heikkilä, V. Ojansivu, E. Rahtu, Improved blur insensitivity for decorrelated local phase quantization, Proc. International Conference on Pattern Recognition, 2010, pp. 818–821.
[57] J. Daugman, High confidence visual recognition of persons by a test of statistical independence, IEEE Trans. Pattern Anal. Mach. Intell. 15 (11) (1993) 1148–1161.
[58] G. Zhao, M. Pietikäinen, Dynamic texture recognition using Local Binary Patterns with an application to facial expressions, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 915–928.
[59] G. Doretto, A. Chiuso, Y. Wu, S. Soatto, Dynamic textures, Int. J. Comput. Vision 51 (2) (2003) 91–109.
[60] B. Ghanem, N. Ahuja, Maximum margin distance learning for dynamic texture recognition, Proc. European Conference on Computer Vision, 2010.
[61] R. Péteri, S. Fazekas, M.J. Huiskes, DynTex: a comprehensive database of dynamic textures, Pattern Recognit. Lett. 31 (12) (2010) 1627–1632.
[62] A. Levin, Y. Weiss, F. Durand, W. Freeman, Understanding and evaluating blind deconvolution algorithms, Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2009.