A FRAMEWORK FOR ANALYZING TEXTURE DESCRIPTORS

Timo Ahonen and Matti Pietikäinen
Machine Vision Group, University of Oulu, PL 4500, FI-90014 Oulun yliopisto, Finland

Keywords: Local Binary Pattern, KTH-TIPS, MR8, Gabor, Texture descriptor.

Abstract: This paper presents a new unified framework for texture descriptors such as Local Binary Patterns (LBP) and Maximum Response 8 (MR8) that are based on histograms of local pixel neighborhood properties. This framework is enabled by a novel filter based approach to the LBP operator which shows that it can be seen as a special filter based texture operator. Using the proposed framework, the filters to implement LBP are shown to be both simpler and more descriptive than MR8 or Gabor filters in the texture categorization task. It is also shown that when the filter responses are quantized for histogram computation, codebook based vector quantization yields slightly better results than threshold based binning at the cost of higher computational complexity.

1 INTRODUCTION

Texture is a fundamental property of surfaces, and automated analysis of textures has applications ranging from remote sensing to document image analysis (Tuceryan and Jain, 1998). Recent findings in applying texture methods to face image analysis, for example, indicate that texture might have applications in new fields of computer vision that have not been considered texture analysis problems. Because of the importance of texture analysis, a wide variety of different texture descriptors have been presented in the literature. However, there is no formal definition of the phenomenon of texture itself that the researchers would agree upon. This is possibly one of the reasons that so far no unified theory or no unified framework of texture descriptors has been presented.

The Local Binary Pattern (LBP) (Ojala et al., 2002), Maximum Response 8 (Varma and Zisserman, 2005) and Gabor filter based texture descriptors are among the most studied and best known recent texture analysis techniques. Despite the large number of publications discussing these methods, the connections and differences between them are not well understood. This paper presents a new unified framework for these texture descriptors, which allows for a systematic comparison of these widely used descriptors and the parts that they are built of.

LBP is an operator for image description that is based on the signs of differences of neighboring pixels. It is fast to compute and invariant to monotonic gray-scale changes of the image. Despite being simple, it is very descriptive, which is attested by the wide variety of different tasks it has been successfully applied to. The LBP histogram has proven to be a widely applicable image feature for, e.g., texture classification, face analysis, video background subtraction, etc. (The Local Binary Pattern Bibliography, 2007).

Another frequently used approach in texture description is using distributions of quantized filter responses to characterize the texture (Leung and Malik, 2001), (Varma and Zisserman, 2005). In the field of texture analysis, filtering and pixel value based texture operators have been seen as somewhat contradictory. However, in this paper we show that the local binary pattern operator can be seen as a filter operator based on local derivative filters at different orientations and a special vector quantization function. Apart from clarifying the connections between LBP and filter based methods, this also helps analyzing the properties of the LBP operator.

2 THE LOCAL BINARY PATTERN OPERATOR

The local binary pattern operator (Ojala et al., 2002) is a powerful means of texture description. The original version of the operator labels the pixels of an image by thresholding the 3x3-neighborhood of each

Ahonen T. and Pietikäinen M. (2008).
A FRAMEWORK FOR ANALYZING TEXTURE DESCRIPTORS.
In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pages 507-512
DOI: 10.5220/0001077305070512

5	9	1
4	4	6
7	2	3

→ Threshold

1	1	0
1		1
1	0	0

Weights

1	2	4
128		8
64	32	16

LBP code: 1+2+8+64+128=203

Figure 1: The basic LBP operator.

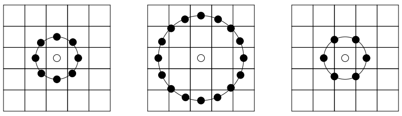

Figure 2: Three circular neighborhoods: (8,1), (16,2), (6,1). The pixel values are bilinearly interpolated whenever the sampling point is not in the center of a pixel.

pixel with the center value and summing the thresholded values weighted by powers of two. Then the histogram of the labels can be used as a texture descriptor. See Fig. 1 for an illustration of the basic LBP operator.

The operator can also be extended to use neighborhoods of different sizes (Ojala et al., 2002). Using circular neighborhoods and bilinearly interpolating the pixel values allow any radius and number of pixels in the neighborhood. For neighborhoods we will use the notation (P,R) which means P sampling points on a circle of radius of R. See Fig. 2 for an example of different circular neighborhoods.

3 FRAMEWORK FOR FILTER BANK AND VECTOR QUANTIZATION BASED TEXTURE DESCRIPTORS

Apart from LBP and other similar methods working directly on pixel values, another widely used approach to texture analysis is to convolve an image with N different filters whose responses at a certain position (x,y) form an N-dimensional vector. At learning stage, a set of such vectors is collected from training images and the set is clustered using, e.g., k-means to form a codebook. Then each pixel of a texture image is labeled with the label of the nearest cluster center and the histogram of these labels over a texture image is used to describe the texture. (Leung and Malik, 2001), (Varma and Zisserman, 2005).

More formally, let $I(x,y)$ be the image to be described by the texture operator. Now the vector valued image obtained by convolving the original image with filter kernels is

$$\mathbf{I}_f(x,y) = \begin{bmatrix} I_1(x,y) = I(x,y) \star F_1 \\ I_2(x,y) = I(x,y) \star F_2 \\ \vdots \\ I_N(x,y) = I(x,y) \star F_N \end{bmatrix} \quad (1)$$

The labeled image $I_{lab}(x,y)$ is obtained with a vector quantizer $f : \mathbb{R}^N \mapsto \{0,1,2,\cdots,M-1\}$, where M is the number of different labels produced by the quantizer. Thus, the labeled image is

$$I_{lab}(x,y) = f(\mathbf{I}_f(x,y)) \quad (2)$$

and the histogram of labels is

$$H_i = \sum_{x,y} \delta\{i, I_{lab}(x,y)\}, i = 0,\ldots,M-1, \quad (3)$$

in which δ is the Kronecker delta.

If the task is classification or categorization as in this work, several possibilities exist for classifier selection. The most typical strategy is to use nearest neighbor classifier using, e.g., χ^2 distance to measure the distance between histograms (Leung and Malik, 2001), (Varma and Zisserman, 2005). In (Varma and Zisserman, 2004), the nearest neighbor classifier was compared to Bayesian classification but no significant difference in the performance was found. In (Caputo et al., 2005) it was shown that the performance of a material categorization system can be boosted by using suitably trained support vector machine based classifier. In this work, the main interest is not in the classifier design but in the local descriptors and thus the nearest neighbor classifier with χ^2 distance was selected for the experimental part.

In the following two subsections we take a look at the two main parts of the proposed texture description framework, the filter bank and the quantization method.

3.1 Filter Bank

In this paper we compare three different types of filter kernels that are used in texture description. The first filter bank is a set of oriented derivative filters whose thresholded output is shown to be equivalent to the local binary pattern operator. The other two filter banks included in the comparison are Gabor filters and Maximum Response 8 filter set.

A novel way to look at the LBP operator proposed in this paper is to see it as a special filter-based texture operator. The filters for implementing LBP are approximations of image derivatives computed at different orientations. The filter coefficients are computed so that they are equal to the weights of bilinear interpolation of pixel values at sampling points of the LBP

0	1	0
0	−1	0
0	0	0

0	0.207	0.5
0	−0.914	0.207
0	0	0

0	0	0
0	−1	1
0	0	0

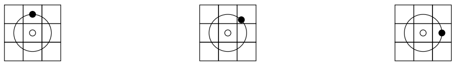

Figure 3: Filters $F_1 \cdots F_3$ of the total of 8 local derivative filters at (8,1) neighborhood. The remaining 5 filters are obtained by mirroring the filters shown here.

operator and the coefficient at filter center is obtained by subtracting 1 from the center value. For example, the kernels shown in Fig. 3 can be used for filter based implementation of local binary pattern operator in the circular (8,1) neighborhood. The response of such filter at location (x,y) gives the signed difference of the center pixel and the sampling point corresponding to the filter. These filters, which will be called local derivative filters in the following, can be constructed for any radius and any number of sampling points.

Another type of filter kernels that is widely used in texture description is Gabor filters, which are complex filters whose spatial representation is obtained by multiplying a Gaussian with a complex sinusoid. The typical way Gabor filters are applied in texture description is to convolve the input image with a bank of Gabor filters and compute a set of features from the output images. A lot of work has been devoted to designing the filter bank and feature computation methods, see, e.g., (Manjunath and Ma, 1996), (Clausi and Jernigan, 2000), (Grigorescu et al., 2002). In this work we apply the Gabor filters in the proposed texture description framework, i.e. the responses of the filter bank at a certain position are stacked into a vector which is used as an input for the vector quantizer.

The third considered filter bank is the Maximum Response 8 bank (Varma and Zisserman, 2005). That filter set consists of 38 filters: two isotropic filters (Gaussian and Laplacian of Gaussian) and an edge and a bar filter both at 3 scales and 6 orientations. As an intermediate step between filtering and vector quantization, the maximum of the 6 responses at different orientations is computed which results in a total of 8 responses. In the proposed unified framework this maximum selection falls more conveniently into the vector quantization operation and it is discussed in more detail in the next subsection.

3.2 Vector Quantization

The assumption onto which the proposed texture description framework is based on is that the joint distribution of filter responses can be used to describe the image texture. Depending on the size of the filter bank, the dimension of the vectors in the image $\mathbf{I}_f(x,y)$ can be high and quantization of the vectors is needed for reliable estimation of the histogram.

A simple, non-adaptive way to quantize the filter responses is to threshold them and to compute the sum of thresholded values multiplied by powers of two:

$$I_{lab}(x,y) = \sum_{n=1}^{N} s\{I_n(x,y)\}\, 2^{n-1}, \qquad (4)$$

where $s(z)$ is the thresholding function

$$s\{z\} = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \qquad (5)$$

Thresholding divides each dimension of the filter bank output into two bins. The total number of different labels produced by threshold quantization is 2^N where N is the number of filters.

Now, if the filter bank that was used to obtain the image $\mathbf{I}_f(x,y)$ is the set of local derivative filters (e.g. the filters presented in Fig. 3), the filter responses are equal to the signed differences of the pixel $I(x,y)$ and its neighbors. As the quantizer (4) is applied to $\mathbf{I}_f(x,y)$, the resulting labels are equal to those obtained with the local binary pattern operator using the same neighborhood. Therefore, the LBP operator can be represented in the proposed framework.

Another method for quantizing the filter responses is to construct a codebook of them at the learning stage and then use the nearest codeword to represent the filter bank output at each location:

$$I_{lab}(x,y) = \arg\min_{m} \left|\left|\mathbf{I}_f(x,y) - \mathbf{c}_m\right|\right|, \qquad (6)$$

in which $\mathbf{c}_m$ is the m-th vector (codeword) in the codebook. This approach is used in (Leung and Malik, 2001) and (Varma and Zisserman, 2005), which use k-means to construct the codebook whose elements are called textons. Codebook based quantization of signed differences of neighboring pixels (which correspond to local derivative filter outputs) was presented in (Ojala et al., 2001).

When comparing these two methods for quantizing the filter responses, one might expect that the if the number of labels produced by the quantizers is kept roughly the same, the codebook based quantizer handles the possible statistical dependencies between the filter responses better. On the other hand, since the codebook based quantization requires the search for the closest codeword at each pixel location, it is clearly slower than simple thresholding, even though a number of both exact and approximate techniques have been proposed for finding the nearest codeword without exhaustive search through the codebook (Gray and Neuhoff, 1998, p. 2362).

It is important to note that a clever co-design of the filter bank and the vector quantizer can also make the texture descriptor rotationally invariant. Again, two different strategies have been proposed. Rotationally invariant LBP codes are obtained by circularly shifting a LBP binary code to its minimum value (Ojala et al., 2002). In the joint framework this can be represented as further combining the labels of threshold quantization (4) so that all the different labels that can arise from rotations of the local gray pattern are joined to form a single label.

On the other hand, the approach chosen for the MR8 descriptor to achieve rotational invariance is to select only the maximum of the 6 different rotations of each bar and edge filters. Only these maximum values and the responses of the two isotropic filters are used in further quantization so the 8-dimensional response of the filter is invariant to rotations of the gray pattern.

4 EXPERIMENTS

To test the proposed framework and to systematically explore the relative descriptiveness of the different filter banks and vector quantization methods, the challenging task of material categorization using the KTH-TIPS2 database (Mallikarjuna et al., 2006) was utilized.

4.1 Experimental Setup

The KTH-TIPS2 database contains 4 samples of 11 different materials, each sample imaged at 9 different scales and 12 lighting and pose setups, totaling 4572 images.

Caputo *et al.* performed material categorization tests using the KTH-TIPS2 and considered especially the significance of classifier selection (Caputo et al., 2005). In that paper, the main conclusions were that the state-of-the-art descriptors such as LBP and MR8 have relatively small differences in the performance but significant gains in classification rate can be obtained by using support vector machine classifier instead of nearest neighbor. Moreover, the classification rates can be enhanced by increasing the number of samples used for training.

In this work, the main interest is to examine the relative descriptiveness of different setups of the filter bank based texture descriptors. To facilitate this task, we chose the most difficult test setup used in (Caputo et al., 2005), namely using the nearest neighbor classifier with only one sample (i.e. 9*12 images) per material class for training.

Table 1: Properties of the tested filter kernels.

Filter bank	Size	Num of filters
Local der. filters	3×3	8
Gabor(1,4)	7×7	8
Gabor(4,6)	49×49	48
MR8	49×49	38

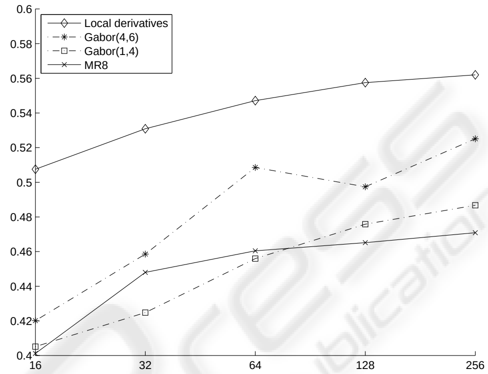

Figure 4: The categorization rates for different filter banks as a function of codebook size

The proposed framework allowed testing the performance of different filters and different quantization methods independently. The filter banks that were included in the test were local derivative filters, two different banks of Gabor filters and MR8 filters. The local derivative filter bank was chosen to match the $LBP_{8,1}$ operator which resulted in 8 filters (see Fig. 3). Two very different types of Gabor were tested, one with only 1 scale and 4 orientations and small spatial support (7×7) and another one with 4 scales and 6 orientations and larger spatial support. The properties of the tested filter kernels are listed in table 1.

4.2 Codebook based Vector Quantization

All the 4 filter banks were tested using two types of vector quantization: thresholding and codebook based quantization. For codebook based quantization, the selected approach was to aim for compact, universal texton codebooks, i.e. codebooks of rather small size that are not tailored for this specific set of textures. Therefore, the KTH-TIPS1 database (Mallikarjuna et al., 2006) was used to learn the codebooks. That database is similar to KTH-TIPS2 in terms of imaging conditions but it contains partly different set of materials (textures). The codebook sizes that were tested were $32 \ldots 256$ codewords.

The categorization rates as a function of the code-

Table 2: Recognition rates for different filter banks and quantization methods.

	Codebook	Thresholding
Local der. filters	0.562	0.532
Gabor(1,4) filters	0.487	0.383
Gabor(4,6)	0.525	-
MR8 filters	0.471	0.498

book size obtained with each filter bank and codebook based quantization is plotted in Figure 4. The figure shows that for most of the time, using a larger codebook enhances the categorization rate but the selection of the filter bank is a more dominant factor than the codebook size. For example, local derivative filters achieve a higher categorization rate with the smallest codebook size than the MR8 filters with any codebook size.

4.3 Thresholding based Vector Quantization

In the next experiment the material categorization tests were performed using the same filter banks but thresholding based vector quantization. The local derivative and Gabor filters have zero mean, thus the thresholding function (5) was applied directly. For the edge and bar filters in the MR8 filter set, only the maximum of responses over different orientations is measured and therefore in that case the mean of 8-dimensional response vectors over all the training images was computed and subtracted from the response before applying thresholding.

Table 2 lists the categorization rates using thresholding based quantization and the maximum categorization rate over different codebook sizes for the four tested filter banks. The Gabor(4,6) filter bank was omitted from this experiment due to the large number of filters in the filter bank (the resulting histograms would have been of length 2^{48}). Codebook based quantization yields slightly better categorization rate than thresholding when using local derivative filters. With the Gabor(1,4) filter bank thresholding performs much worse than codebook based quantization, but interestingly with MR8 filters, thresholding yields better rate.

To conclude the performed experiments, the local derivative filters give the best categorization rate over the tested filter sets with both quantization functions. The results obtained in these experiments also attest those presented in (Ojala et al., 2001) which showed that codebook based quantization of signed gray-level differences yields slightly better recognition than LBPs, however at the cost of higher computational complexity. Considering the computational cost of the presented methods, thresholding based quantization is much faster than codebook based quantization. As for the filter bank operations, the computational cost grows with the size and number of filters, but using FFT based convolution can make the operations faster. Still, at two extremes, the computations for local derivative filter and thresholding based labeling of an image of size of 256×256 take 0.04 seconds whereas the codebook based labeling of the same image using Gabor(4,6) filters (and performing convolutions using FFT) take 10.98 seconds. Both running times were measured using unoptimized Matlab implementations of the methods on a PC with AMD Athlon 2200 MHz processor.

4.4 Filter Subset Selection

The third experiment tested whether it is possible to select a representative subset of filters from a large filter bank for thresholding based quantization. The number of labels produced by the quantizer is 2^N in which N is the number of filters, which means that the length of the label histograms grows exponentially with respect to the number of filters. Thus a small filter bank is desirable for the thresholding quantization.

In this experiment, the Sequential Floating Forward Selection (SFFS) (Pudil et al., 1994) algorithm was used to select a maximum of 8 filters from a larger filter bank. The optimization criterion was the recognition rate over the training set (KTH-TIPS1). Two different initial filter banks were tested. First, 8 filters were selected from the 48 filters in the Gabor(4,6) filter bank. However, the resulting 8-filter bank did not perform well on the testing database, yielding a categorization rate of only 0.295.

In the face recognition literature, there are findings that LBP and Gabor filter based information are complementary. In (Yan et al., 2007), score level fusion of LBP and Gabor filter based similarity scores was done. Motivated by these findings, SFFS was used to select 8 filters from the union of local derivative and Gabor(1,4) filter banks. This resulted in a set of 6 local derivative and 2 Gabor filters and the resulting filter bank reached categorization rate of 0.544 which is significantly higher than the rate of Gabor(1,4) filter bank and slightly higher than the rate of the local derivative filter bank.

5 DISCUSSION AND CONCLUSIONS

In this paper we have presented a novel unified framework under which the histogram based texture description methods such as local binary pattern and MR8 descriptors can be explained and analyzed. This framework allows for systematic comparison of different texture descriptors and the parts that the descriptors are built of. Such novel approach can be useful in analyzing texture descriptors since they are usually presented as a sequence of steps whose relation to other texture description methods is unclear. The framework presented in this work allows for explicitly illustrating the connection between the parts of the LBP and MR8 descriptors and experimenting with the performance of each part.

The filter sets and vector quantization techniques for LBP, MR8 and Gabor filter based texture descriptors were compared in the this paper. In this comparison it was found out that the local derivative filter responses are both fastest to compute and most descriptive. This somewhat surprising result further attests the previous findings that texture descriptors relying on small-scale pixel relations yield comparable or even superior results to those based on filters of larger spatial support (Ojala et al., 2002), (Varma and Zisserman, 2003).

When comparing the different vector quantization methods, codebook based quantization was discovered to be slightly more descriptive than thresholding in most cases. Finally, the preliminary experiments on combining local derivative and Gabor filter responses showed that these filter sets may be complementary and may yield better performance than either of the sets alone.

Not only does the presented framework contribute to understanding and comparison of existing texture descriptors but it can be utilized for more systematic development of new, even better performing methods. The framework is simple to implement and together with the publicly available KTH-TIPS2 image database it can be easily used for comparing novel descriptors with the current state-of-the-art methods. We believe that further advances in both the filter bank and vector quantizer design are possible, especially as new invariance properties of the descriptors are aimed for.

REFERENCES

Caputo, B., Hayman, E., and Mallikarjuna, P. (2005). Class-specific material categorisation. In *Proc. ICCV 2005*, pages 1597–1604.

Clausi, D. A. and Jernigan, M. E. (2000). Designing Gabor filters for optimal texture separability. *Pattern Recognition*, 33(11):1835–1849.

Gray, R. M. and Neuhoff, D. L. (1998). Quantization. *IEEE Trans. Image Process*, 44(6):2325–2383.

Grigorescu, S. E., Petkov, N., and Kruizinga, P. (2002). Comparison of texture features based on Gabor filters. *IEEE Trans. Image Process.*, 11(10):1160–1167.

Leung, T. and Malik, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Comput. Vision*, 43(1):29–44.

Mallikarjuna, P., Fritz, M., Targhi, A. T., Hayman, E., Caputo, B., and Eklundh, J.-O. (2006). The KTH-TIPS and KTH-TIPS2 databases. http://www.nada.kth.se/cvap/databases/kth-tips/.

Manjunath, B. S. and Ma, W.-Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842.

Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987.

Ojala, T., Valkealahti, K., Oja, E., and Pietikäinen, M. (2001). Texture discrimination with multidimensional distributions of signed gray-level differences. *Pattern Recognition*, 34(3):727–739.

Pudil, P., Novovicová, J., and Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters*, 15(10):1119–1125.

The Local Binary Pattern Bibliography (2007). http://www.ee.oulu.fi/mvg/page/lbp_bibliography.

Tuceryan, M. and Jain, A. K. (1998). Texture analysis. In Chen, C. H., Pau, L. F., and Wang, P. S. P., editors, *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, pages 207–248. World Scientific Publishing Co.

Varma, M. and Zisserman, A. (2003). Texture classification: Are filter banks necessary? In *Proc. CVPR 2003*, volume 2, pages 691–698.

Varma, M. and Zisserman, A. (2004). Unifying statistical texture classification frameworks. *Image Vision Comput.*, 22(14):1175–1183.

Varma, M. and Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1–2):61–81.

Yan, S., Wang, H., Tang, X., and Huang, T. S. (2007). Exploring feature descriptors for face recognition. In *Proc. ICASSP 2007*.