Adaptive Matrices for Color Texture Classification

Kerstin Bunte[1], Ioannis Giotis[1], Nicolai Petkov[1], and Michael Biehl[1]

Johann Bernoulli Institute for Mathematics and Computer Science,
University of Groningen, The Netherlands
kbunte@rug.nl, i.e.giotis@rug.nl

Abstract. In this paper we introduce an integrative approach towards color texture classification learned by a supervised framework. Our approach is based on the Generalized Learning Vector Quantization (GLVQ), extended by an adaptive distance measure which is defined in the Fourier domain and 2D Gabor filters. We evaluate the proposed technique on a set of color texture images and compare results with those achieved by simple gray value transformation on the color images with a comparable dissimilarity measure and the same filter bank. The features learned by GLVQ improve classification accuracy and they generalize much better for evaluation data previously unknown to the system.

Keywords: adaptive metric, Gabor filter, color texture analysis, classification, Learning Vector Quantization

1 Introduction

Texture analysis and classification are topics of particular interest mainly due to their numerous possible applications, such as medical imaging, industrial quality control and remote sensing. A wide variety of methods for texture analysis has been already developed such as co-occurrence matrices [8], Markov random fields [24], autocorrelation methods [18, 16], Gabor filtering [22, 5, 11, 13, 15, 6] and wavelet decomposition [23]. However, these methods mostly concern intensity images and since color information is a vector quantity the transfer of traditional methods to the color domain is not always straightforward. With regards to color texture the possible approaches can be distinguished in three categories [17]. The most popular among them is called the integrative approach [10, 4, 17, 9] and is an attempt to describe texture by combining color information with the spatial relationships of image regions within each color channel and between different color channels.

In this contribution we introduce a novel integrative approach towards color texture classification and recognition based on 2D Gabor filters through supervised learning. Given a set of labeled color images (RGB) for training and a bank of 2D Gabor filters the goal here is to learn a transformation of a color image to a single channel (intensity) image, such that the Gabor responses of the transformed images will yield the best possible classification. Most signal processing techniques are based on insights or empirical observations from neurophysiology or optical physics. The proposed, novel approach incorporates data-driven adaptation of the system, e.g. example based learning. Furthermore, the filters used in our approach can be substituted, depending on the data domain and the task at hand. As an example we explore in this paper the use of rotation and scale invariant descriptors based on Gabor filter responses [7]. We demonstrate that our novel approach yields very good generalization ability with respect to previously unknown data.

2 Review of the Generalized LVQ

In this section we introduce a methodology to learn discriminative transformations for images. Our adaptation is based on the Generalized Learning Vector Quantization (GLVQ) [19]. GLVQ is an extension which introduces a cost function to the original Learning Vector Quantization (LVQ) [12] formulation. LVQ is a supervised prototype-based classification method, easy to implement and interpret, which makes it popular for many applications. The training is based on data points $\boldsymbol{x}^i \in \mathbb{R}^D$ and their corresponding label information $y^i \in 1, \ldots, C$, where D denotes the dimension of the feature vectors and C the number of classes. The protoytpes are characterized by their location in the feature space $\boldsymbol{w}^i \in \mathbb{R}^D$ and the respective class label $c(\boldsymbol{w}^i) \in 1, \ldots, C$. Given a dissimilarity measure $d(\boldsymbol{x}, \boldsymbol{w})$ (e.g. the Euclidean distance), any data point $\boldsymbol{x}$ is assigned to the class label $c(\boldsymbol{w}^i)$ of the closest prototype $\boldsymbol{w}^i$ with $d(\boldsymbol{x}, \boldsymbol{w}^i) \leqslant d(\boldsymbol{x}, \boldsymbol{w}^j)$ for all $j \neq i$. The training algorithm is guided by the minimization of a cost function

$$f_c(d, J, K) = \sum_i \frac{d(\boldsymbol{x}^i, \boldsymbol{w}^J) - d(\boldsymbol{x}^i, \boldsymbol{w}^K)}{d(\boldsymbol{x}^i, \boldsymbol{w}^J) + d(\boldsymbol{x}^i, \boldsymbol{w}^K)} \tag{1}$$

where the quantities $d(\boldsymbol{x}^i, \boldsymbol{w}^J)$ with $c(\boldsymbol{w}^J) = \boldsymbol{y}^i$ and $d(\boldsymbol{x}^i, \boldsymbol{w}^K)$ with $c(\boldsymbol{w}^K) \neq \boldsymbol{y}^i$ correspond to the distances of the feature vector $\boldsymbol{x}^i$ from the respective closest correct prototype $\boldsymbol{w}^J$ and the closest wrong prototype $\boldsymbol{w}^K$. The original algorithm follows a stochastic gradient descent for the optimization of the cost function Eq. (1). The gradients are evaluated with respect to the contribution of single instances $\boldsymbol{x}^i$, which are presented at random and sequentially during training. Further extensions like, for instance, the Generalized Matrix LVQ (GMLVQ) [20] employ an adaptive dissimilarity measure $d^{\Omega}(\boldsymbol{x}, \boldsymbol{w}) = (\boldsymbol{x} - \boldsymbol{w})^\top \Omega^\top \Omega (\boldsymbol{x} - \boldsymbol{w})$ which corresponds to a generalized Euclidean metric. GMLVQ and its modifications have proven beneficial in many applications, including classification, content based image retrieval and supervised dimension reduction [21, 2, 3]. In the following section we extend the original GLVQ formulation for color texture classification.

3 Adaptive Matrices for Texture Classification

Consider a data set consisting of color image patches of a priorly defined size $(p \times p)$ and a bank of Gabor kernels $\mathbf{G}$ with different scales and orientations. We use for both the image patches and the filter kernels their representation in the Fourier domain. After vectorizing we end up with complex data points $\boldsymbol{x}^i \in \mathbb{C}^D$ of dimension $D = p \cdot p{\cdot}3$ carrying a label $y^i \in \{1, \ldots, C\}$ that belong to one of C classes and a filter bank $\mathbf{G}$, where $\mathbf{G}^l \in \mathbb{C}^M$ with $M = p \cdot p$ is the vectorized kernel of the l-th filter of the bank. The general form of the descriptor for a vectorized image patch $\boldsymbol{v}$ given the filter bank $\mathbf{G}$ and parameterized by local transformations Ω_k can be written as $f_{\Omega_k}(\boldsymbol{v}, \mathbf{G}) : \mathbb{C} \to \mathbb{C}$. Here k corresponds to the index of the prototype $\boldsymbol{w}^k$ or the index of its class label $c(\boldsymbol{w}^k)$ for class-wise transformations. For the proposed optimization procedure it is necessary, that f_{Ω_k} is differentiable. In this contribution f_{Ω_k} corresponds to the sum of the responses of all filter kernels in $\mathbf{G}$ to the vectorized image patch, thus defining the descriptor:

$$f_{\Omega_k}(\boldsymbol{v},\mathbf{G}) : \boldsymbol{v} \rightarrow \boldsymbol{r}_k(\boldsymbol{v}) = \sum_l \boldsymbol{v}{\Omega_k}^{\top} \ast \mathbf{G}^l \ , \tag{2}$$

where $\ast$ denotes the convolution. The filter bank $\mathbf{G}$ may be chosen based on the user's preference, suitable to the data and the task at hand. The vector $\boldsymbol{v}$ is defined in the data domain $\mathbb{C}^D$ and $\Omega_k \in \mathbb{C}^{M \times D}$ is the local transformation, which maps the color values to scalar, "intensity" values used for filtering. The dissimilarity measure is defined by:

$$d_{\mathbf{G}}^{\Omega_k}(\boldsymbol{x}^i,\boldsymbol{w}^k) = \| \ |\boldsymbol{r}_k(\boldsymbol{x}^i)|^2 - |\boldsymbol{r}_k(\boldsymbol{w}^k)|^2 \ \|^2 \ , \tag{3}$$

and corresponds to the difference of descriptor magnitudes. This considers two patches containing the same texture pattern as similar, independent of the position where the pattern occurs within the patches.

We use the same cost function as in the original GLVQ algorithm Eq. (1). We follow a stochastic gradient descent procedure and present the samples $\boldsymbol{x}^i$ of the training set sequentially and update the parameters accordingly. We will refer to this algorithm as Color Image Analysis LVQ (CIA-LVQ) and to one sweep through the training set as one epoch E.

Explicit form of the learning rules: For the sake of completeness we present the explicit form of the learning rules of CIA-LVQ. The parameter updates read as follows:

$$\boldsymbol{w}^L = \boldsymbol{w}^L - \alpha \Delta \boldsymbol{w}^L, \ \Delta \boldsymbol{w}^L = \frac{\partial f_c(d_{\mathbf{G}}^{\Omega_J}, d_{\mathbf{G}}^{\Omega_K}, J, K)}{\partial \Re(\boldsymbol{w}^L)} + i \frac{\partial f_c(d_{\mathbf{G}}^{\Omega_J}, d_{\mathbf{G}}^{\Omega_K}, J, K)}{\partial \Im(\boldsymbol{w}^L)} \tag{4}$$

$$\Omega_L = \Omega_L - \epsilon \Delta \Omega_L, \ \Delta \Omega_L = \frac{\partial f_c(d_{\mathbf{G}}^{\Omega_J}, d_{\mathbf{G}}^{\Omega_K}, J, K)}{\partial \Re(\Omega_L)} + i \frac{\partial f_c(d_{\mathbf{G}}^{\Omega_J}, d_{\mathbf{G}}^{\Omega_K}, J, K)}{\partial \Im(\Omega_L)} \tag{5}$$

where $L \in \{J, K\}$ and α and ϵ are the learning rates for the prototypes and the matrix respectively. The derivatives with respect to the closest correct $\boldsymbol{w}^J$ and closest wrong prototype $\boldsymbol{w}^K$ together with their corresponding matrices Ω_J and Ω_K for the given training data point $\boldsymbol{x}^i$ read:

$$\Delta \boldsymbol{w}^L = -4 \cdot \gamma^L \left[\left(|\boldsymbol{r}_L(\boldsymbol{x}^i)|^2 - |\boldsymbol{r}_L(\boldsymbol{w}^L)|^2 \right) \cdot \boldsymbol{r}_L(\boldsymbol{w}^L)^* \ast \left(\sum_l \Omega_L \ast \mathbf{G}^l \right) \right]^* \tag{6}$$

$$\begin{aligned} \Delta \Omega_L = \gamma^L \Bigg(& 4 \left(|\boldsymbol{r}_L(\boldsymbol{x}^i)|^2 - |\boldsymbol{r}_L(\boldsymbol{w}^L)|^2 \right) \\ & \cdot \left[\boldsymbol{r}_L(\boldsymbol{x}^i)^* \ast \left(\sum_l \boldsymbol{x}^i \ast \mathbf{G}^l \right) - \boldsymbol{r}_L(\boldsymbol{w}^L)^* \ast \left(\sum_l \boldsymbol{w}^L \ast \mathbf{G}^l \right) \right] \Bigg)^*, L \in \{J, K\} \end{aligned} \tag{7}$$

with $\gamma^J = \frac{2\cdot d_{\mathbf{G}}^{\Omega_K}(\boldsymbol{x}^i,\boldsymbol{w}^K)}{\left(d_{\mathbf{G}}^{\Omega_J}(\boldsymbol{x}^i,\boldsymbol{w}^J)+d_{\mathbf{G}}^{\Omega_K}(\boldsymbol{x}^i,\boldsymbol{w}^K)\right)^2}$, $\gamma^K = \frac{-2\cdot d_{\mathbf{G}}^{\Omega_J}(\boldsymbol{x}^i,\boldsymbol{w}^J)}{\left(d_{\mathbf{G}}^{\Omega_J}(\boldsymbol{x}^i,\boldsymbol{w}^J)+d_{\mathbf{G}}^{\Omega_K}(\boldsymbol{x}^i,\boldsymbol{w}^K)\right)^2}$ and * denoting the complex conjugate. Note, that since we are working with complex values we have to take all derivatives with respect to the real and imaginary parts respectively.

In the next section we experiment with the algorithm and show its use in practice.

4 Experiments

In order to evaluate the usefulness of the proposed algorithm, we perform classification on patches of pictures taken from the VisTex database [1]. Our data consists of color images with size 128×128 pixels from the groups Bark, Brick, Tile, Fabric and Food. Although in texture classification literature each such image is often considered as a different class, here we distinguish into five different classes equivalent to the five aforementioned groups. Despite its increased difficulty, this classification task allows us to better demonstrate the ability of CIA-LVQ to describe general characteristics of real-world texture patterns.

At first we draw 15×15 patches randomly from each image shown in Fig. 1. The training set contains 150 patches per image, resulting in 3000 samples in total, while the test set holds 50 patches from each image. The test set may contain patches which partially overlap with those used for training. Therefore the images in Fig. 2 are used in order to create an evaluation set that was never seen in the training process. The evaluation set consists of 50 randomly drawn patches per image and is used to show the generalization ability of the approach.

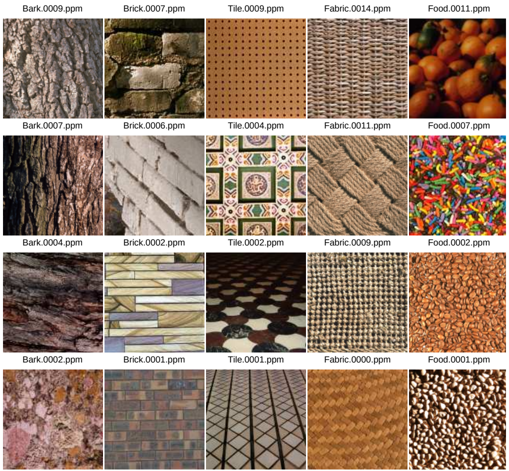

Fig. 1: Images, which are used to provide random patches for training and test.

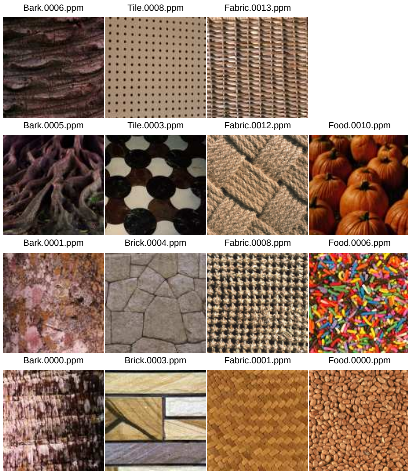

Fig. 2: Images, which are used to provide random patches for evaluation.

A note is due here to the nature of the filter used. A 2D Gabor filter is defined as a Gaussian kernel function modulated by a sinusoidal plane wave. All filter

Table 1: Confusion matrices (eval. set)

CIA-LVQ:

	1	2	3	4	5	$\sum$
1	176	10	12	7	2	207
2	1	57	11	9	3	81
3	18	25	43	31	10	127
4	1	5	23	127	4	160
5	4	3	11	26	131	175
$\sum$	200	100	100	200	150	750

class-wise accuracy of estimation in %

88.00 57.00 43.00 63.50 87.33

RGB2G:

	1	2	3	4	5	$\sum$
1	52	14	7	36	28	137
2	51	45	30	37	34	197
3	27	27	51	26	22	153
4	29	6	8	83	11	137
5	41	8	4	18	55	126
$\sum$	200	100	100	200	150	750

class-wise accuracy of estimation in %

26.00 45.00 51.00 41.50 36.67

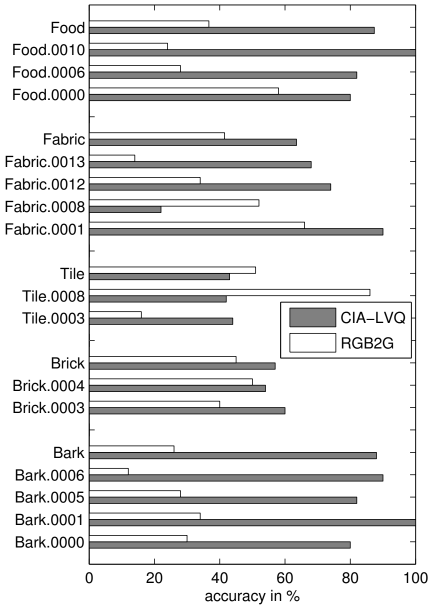

Fig. 3: Classwise and individual image accuracies

kernels can be generated from one basic wavelet by dilation and rotation. In this experiment our filter bank consists of 12 Gabor filters of bandwidth equal to 1 at six orientations $\theta = 0$, 30, 60, 90, 120 and 150 degrees and two scales (wavelegths) varying by one octave: $\lambda = 7$ and $7\sqrt{2}$. These scales ensure that the Gabor function yields an adequate number of visible parallel excitatory and inhibitory stripe zones. Dependent on the patch size different scales might be adequate. We set the phase offset $\phi = 0$ and the aspect ratio $\gamma = 1$ for all filters. In this way we create center-on symmetric filters with circular support. We run the localized version of CIA-LVQ with class-wise matrices Ω_c initialized with the identity matrix and 4 prototypes per class for E =300 epochs. The learning rates were chosen as $\alpha(t) = 0.002\,(0.005)^{t/E}$, $\epsilon(t) = 10^{-3}\left(10^{-2}\right)^{t/E}$ where t is the current epoch. Using more filters and more localized matrices Ω_j may cause overfitting effects. So it is advisable to increase the complexity of the system carefully. The training error is 10.6% and the error on the test set 28%.

We use the same data sets and the same filter bank to compare with the common approach of deriving textural information only from the luminance plane of images [4]. This approach is considered to often outperform combined color and texture features [14]. For this purpose an RGB to gray (RGB2G) transformation is used, which builds intensity values by a weighted sum of the color components of every pixel: 0.2989·R+0.587·G+0.114·B. We again vectorize all patches s and in this case the image patch descriptor is given by $\boldsymbol{r}_2(\boldsymbol{s}) = \sum_l \boldsymbol{s} \ast \mathbf{G}^l$. We use a one nearest neighbor (NN) classification scheme with a dissimilarity measure similar to Eq. (3): $d_{\mathbf{G}}(\boldsymbol{x}^i, \boldsymbol{x}^j) = \|\ |\boldsymbol{r}_2(\boldsymbol{x}^i)|^2 - |\boldsymbol{r}_2(\boldsymbol{x}^j)|^2\ \|^2$. The NN scheme based on the RGB2G transformation shows a test error of 37.5%, but most interesting is the comparison of the classification errors on the evaluation set. Here the NN scheme shows an error of 61.9%, while the CIA-LVQ

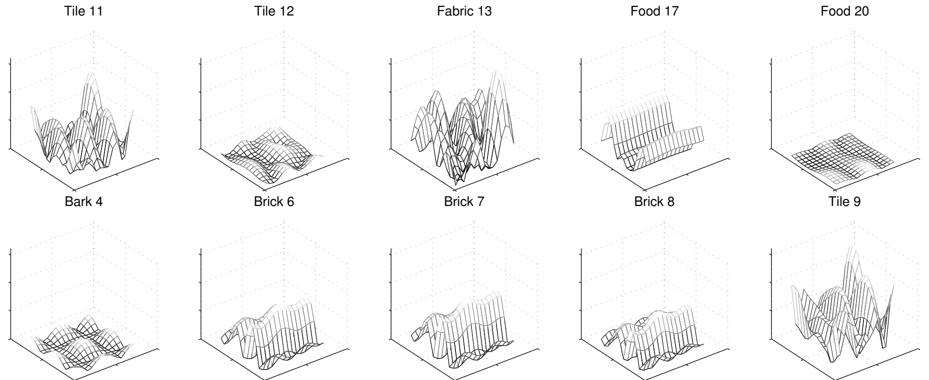

Fig. 4: Magnitude of the descriptors $|\boldsymbol{r}_L(\boldsymbol{w}^L)|$ of the prototypes which classify the evaluation set.

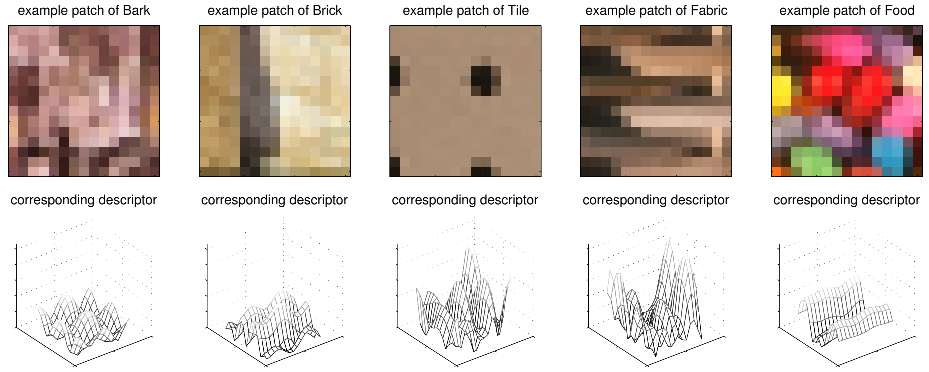

Fig. 5: Magnitude of the descriptors $|\boldsymbol{r}_L(\boldsymbol{w}^L)|$ of some correct classified example patches of the evaluation set.

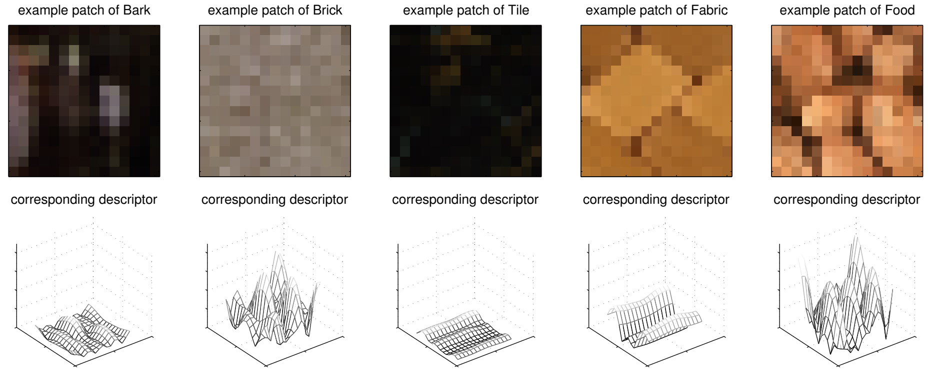

Fig. 6: Magnitude of the descriptors $|\boldsymbol{r}_L(\boldsymbol{w}^L)|$ of some wrongly classified example patches of the evaluation set.

still has an error of 28.8%. The LVQ scheme displays very good generalization, which is shown in Table 1 and Fig. 3. Note, that the accuracy rates among individual images of the same class can vary. Brick and Tile are the most difficult classes, because the texture is large, so it cannot be captured very well with such a small patch size, since a lot of patches might be drawn non-textured regions. On the other side classes like Food and Bark with less diversity regarding textural structures can be learned quite well.

The prototypes, which classify the evaluation set are shown in Fig. 4. Additionally we show some example patches from the evaluation set, which are classified correctly together with their descriptors in Fig. 5 and some examples of wrongly classified patches in Fig. 6. Some obvious problems occur due to the random sampling and the very small patchsize: a lot of samples of Brick and Tile, for example, show homogeneous regions coming from the area in-between the textural structure (see Fig. 6). We observe, that classes which vary a lot in the size of the actual structure (e.g. Brick and Tile) are more difficult to recognize than classes with small variations in the scale of texture (like Bark and Food). It is interesting to notice that random patches drawn from Food.0010.ppm are 100% correctly classified, even though no patch from this image was ever used to train the algorithm. The learned local transformation recognizes the channels leading to the orange color and increased their weights to distinguish this class from others.

5 Conclusion and Outlook

In this contribution we proposed a prototype based framework for color texture analysis. Contrary to standard approaches which are either based on a single channel representation of the images through a fixed transformation or empirical observations for combining color and textural information, we offer the alternative of data driven learning of suitable, parameterized image descriptors. The ability of weighting different color channels automatically according to their importance for the classification task is the most important factor which distinguishes our approach. We have formulated a novel general principle: based on a differentiable convolution and a predefined filter bank the CIA-LVQ algorithm optimizes the classification. It is also of conceptual value that this adaptation of LVQ is suitable for learning in the complex numbers domain. As an example we used Gabor filters to classify texture patterns in 15×15 patches randomly drawn from images of the VisTex database. The results show that the algorithm can learn typical texture patterns with very good generalization, even from relatively small patches and filter banks. Similarly to Gabor filters any other family of 2D filters commonly used to describe gray scale image information could be adapted and applied to color image analysis with this algorithm. A filter bank with differences of Gaussians for color edge detection is a possible example. Investigation of the performance of the system on other filters can be addressed in future. Furthermore, depending on the task it might be desirable that two patches in which the same texture occurs on different positions should not be interpreted as similar. In this case another similarity measure should be used: $\| \, |\boldsymbol{r}(\boldsymbol{x}^i) - \boldsymbol{r}(\boldsymbol{w}^L)| \, \|^2$, which is not based on the difference of magnitudes. This might be of advantage for example in the recognition of objects such as traffic signs, were a corner or an edge might have different meanings dependent on its position in the image.

References

1. Database VisTex of color textures from MIT. available at: http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html.

2. K. Bunte, M. Biehl, N. Petkov, and M. F. Jonkman. Learning effective color features for content based image retrieval in dermatology. Pattern Rec., 2011.
3. K. Bunte, B. Hammer, P. Schneider, and M. Biehl. Nonlinear discriminative data visualization. In M. Verleysen, editor, *Proc. of European Symposium on Artificial Neural Networks (ESANN)*, pages 65–70, Bruges, Belgium, April 2009.
4. A. Drimbarean and P. F. Whelan. Experiments in colour texture analysis. *Pattern Recognition Letters*, 22(10):1161 – 1167, 2001.
5. I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological Cybernetics*, 61(2):103–113, June 1989.
6. S. Grigorescu, N. Petkov, and P. Kruizinga. Comparison of texture features based on gabor filters. *IEEE Trans. on Image Processing*, 11(10):1160 – 1167, Oct. 2002.
7. J. Han and K.-K. Ma. Rotation-invariant and scale-invariant gabor features for texture image retrieval. *Image and Vision Computing*, 25(9):1474 – 1481, 2007.
8. R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Trans. on Systems, Man and Cybernetics*, 3(6):610 –621, 1973.
9. M. A. Hoang, J.-M. Geusebroek, and A. W. Smeulders. Color texture measurement and segmentation. *Signal Processing*, 85(2):265 – 275, 2005. SI on Content Based Image and Video Retrieval.
10. A. Jain and G. Healey. A multiscale representation including opponent color features for texture recognition. *IEEE Trans. on Image Processing*, 7(1):124 –128, Jan. 1998.
11. A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167 – 1186, 1991.
12. T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, Heidelberg, New York, 3rd edition, 2001.
13. P. Kruizinga and N. Petkov. A computational model of periodic-pattern-selective cells. In *From Natural to Artificial Neural Computation*, volume 930 of *Lecture Notes in Computer Science*, pages 90–99. Springer Berlin / Heidelberg, 1995.
14. T. Mäenpää and M. Pietikäinen. Classification with color and texture: jointly or separately? *Pattern Recognition*, 37(8):1629 – 1640, 2004.
15. B. Manjunath and W. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(8):837 –842, Aug. 1996.
16. T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:971–987, 2002.
17. C. Palm. Color texture classification by integrative co-occurrence matrices. *Pattern Recognition*, 37(5):965 – 976, 2004.
18. M. Pietikäinen, T. Ojala, and Z. Xu. Rotation-invariant texture classification using feature distributions. *Pattern Recognition*, 33(1):43 – 52, 2000.
19. A. S. Sato and K. Yamada. Generalized learning vector quantization. In M. C. M. D. S. Touretzky and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 423–429, Cambridge, MA, USA, 1996. MIT Press.
20. P. Schneider, M. Biehl, and B. Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21(12):3532–3561, 2009.
21. P. Schneider, K. Bunte, B. Hammer, and M. Biehl. Regularization in matrix relevance learning. *IEEE Transactions on Neural Networks*, 21(5):831–840, May 2010.
22. M. R. Turner. Texture discrimination by gabor functions. *Biological Cybernetics*, 55:71–82, 1986. 10.1007/BF00341922.
23. J.-W. Wang, C.-H. Chen, W.-M. Chien, and C.-M. Tsai. Texture classification using non-separable two-dimensional wavelets. *Pattern Recogn. Lett.*, 19:1225–1234, November 1998.
24. L. Wang and J. Liu. Texture classification using multiresolution markov random field models. *Pattern Recognition Letters*, 20(2):171 – 182, 1999.