Gray level cooccurrence histograms via learning vector quantization

Timo Ojala, Matti Pietikäinen and Juha Kyllönen
Machine Vision and Media Processing Group,
Infotech Oulu and Department of Electrical Engineering,
University of Oulu,
P.O.Box 4500, FIN-90401 Oulu, Finland

Abstract

In this paper, we propose to use learning vector quantization for the efficient partitioning of a cooccurrence space. A simple codebook is trained to map the multidimensional cooccurrence space into a 1-dimensional cooccurrence histogram. In the classification phase a nonparametric log-likelihood statistic is employed for comparing sample and prototype histograms. The advantages of vector quantization are demostrated with a difficult texture classification problem involving 32 textures. We also point out two problems in the use of cooccurrence matrices that should be taken into account in order to achieve the best possible classification accuracy. Finally, we compare the performance of cooccurrence histograms to that of GMRF features and Gabor filtering, proving that gray level cooccurrences are a powerful approach if used properly.

1 Introduction

Second order gray level statistics is a widely used concept in texture analysis [2]. Conventionally, second order statistics are accumulated into a set of 2-dimensional matrices which are computed for displacements in different directions and displacements. Prior to the computation of a cooccurrence matrix, the number of gray levels is normally reduced for example with histogram equalization to 16 or 32, in order to keep the size of the cooccurrence matrix manageable. The number of gray levels is also related to the statistical reliability of the matrices, for the number of entries per matrix cell should be adequate.

When cooccurrences of several pixel pairs located in different directions are considered, they are often averaged into a single 2-dimensional matrix with the argument that this single matrix provides rotation-invariant texture information. Another motivation for using 2-dimensional matrices is that matrices of higher dimensionality are impractical.

Two underlying flaws can be pointed out in this conventional approach:

- Inefficient partitioning of the cooccurrence space. The quantization is straightforwadly derived from the distribution of gray levels, which corresponds to dividing both feature axes separately into G bins. This results in a suboptimal overall partition of the coocurrence space.
- Inefficient description of multipixel (>2) cooccurrences. If coocurrences of several pixel pairs are accumulated into a single two-dimensional matrix, we erroneously assume texture information to be the average of informations in several directions. Experiments in Section 3 will show how costly this assumption can be.

Both drawbacks have one thing in common: they are related to the quantization of the cooccurrence space. Assuming that we want to describe the cooccurrences of D pixels, we will use a D-dimensional cooccurrence space, i.e. the gray levels of the D pixels are presented as D-dimensional vectors. However, if these D-dimensional entries are straightforwardly stored into matrices, we obtain matrices of size G^D. These matrices can be very large, even with modest values of D and G, thus computationally expensive and suspect to statistical unreliability. Consequently, a more efficient quantization of the D-dimensional cooccurrence space is needed.

In this paper we demonstrate how an efficient quantization improves the performance of coocurrence matrices. Following the work of Valkealahti and Oja [6], we propose to use learning vector quantization for this purpose. Where Valkealahti and Oja used fairly complex codebook structure and learning algorithms, we employ a simple codebook with the basic optimized LVQ1 training algorithm by Kohonen *et al.* [3]. In addition, we also point out two shortcomings in the standard cooccurrence matrix methodology that is normally used in texture analysis.

This paper is organized as follows. Section 2 describes the basic idea of quantizing the cooccurrence space using vector quantization. In Section 3 the performance of the proposed method is experimentally compared to that of the conventional approach with a difficult texture classification problem. Section 4 provides discussion and concludes the paper.

2 From cooccurrence space to cooccurrence histograms via vector quantization

2.1 Cooccurrence space and 'conventional' cooccurrence matrices

In this paper we consider cooccurrences within 3x3-pixel subimages,

g_4	g_2	g_3
g_5	g_0	g_1
g_6	g_7	g_8

and estimate following distributions:

$$p_2(g_0, g_1) \tag{1}$$

$$p_3(g_0, g_1, g_2) \tag{2}$$

$$p_5(g_0, g_1, g_2, g_3, g_4) \tag{3}$$

$$p_9(g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8) \tag{4}$$

Let C_{ij} denote the 'conventional' 2-dimensional cooccurrence matrix corresponding to the cooccurrences of gray levels g_i and g_j. Then the cooccurrence matrix estimating distribution p_2 is simply $C_2 = C_{01}$. Usually, cooccurrence matrices are made symmetrical by replicating entries (g_i,g_j) as (g_j,g_i), i.e. $C_2^{sym} = C_{01} + C_{10}$. However, we argue against doing this, for replicating entries effectively means ignoring the effect of texture orientation in opposite directions which can be costly. We will compare the performance of C_2 and C_2^{sym} quantitatively in the experiments.

We have two alternatives for estimating p_3, p_5, and p_9 with 2-dimensional cooccurrence matrices. First, the 'conventional' approach of accumulating cooccurrences into a single matrix (C^{acc}). As an alternative approach we propose to concatenate the cooccurrence matrices of different pixel pairs into one large matrix (C^{con}). Consequently, we have the following cooccurrence matrices estimating distributions p_3, p_5, and p_9:

$$C_3^{acc} = C_{01} + C_{02} \tag{5}$$

$$C_5^{acc} = C_{01} + C_{02} + C_{03} + C_{04} \tag{6}$$

$$C_9^{acc} = C_{01} + C_{02} + C_{03} + C_{04} + C_{05} + C_{06} + C_{07} + C_{08} \tag{7}$$

$$C_3^{con} = [C_{01}C_{02}] \tag{8}$$

$$C_5^{con} = [C_{01}C_{02}C_{03}C_{04}] \tag{9}$$

$$C_9^{con} = [C_{01}C_{02}C_{03}C_{04}C_{05}C_{06}C_{07}C_{08}] \tag{10}$$

Fig. 1a illustrates the 256x256 cooccurrence space of p_2 for a particular training set extracted from the texture data used in the experiments (see Section 3.1 for details on how the image data is divided into training and testing sets). The training set contains 1024 64x64 samples which produce almost 4 million entries into the cooccurrence space. The intensity reflects the probability of a given cooccurrence; the darker a cooccurrence cell is the higher is its probability and *vice versa*. Most of the probability mass is located around the diagonal of the matrice, which reflects the correlation between adjacent pixels.

2.2 Cooccurrence histograms via vector quantization

To estimate cooccurrences of D pixels, we propose to partition the D-dimensional cooccurrence space using vector quantization, instead of using the original multidimensional space or accumulating the cooccurrences into a 2-dimensional matrix. For this purpose we employ a codebook of N D-dimensional codewords, which have indeces n=0,1,...,N-1. The codebook is trained with the optimized LVQ1 training algorithm (Kohonen *et al.* 1992), by selecting R random vectors from each of the 1024 samples in the training set. The R*1024 random vectors are presented to the codebook T times. The small white rectangles in Fig. 1a correspond to the locations of the codewords, when a codebook of 96 codewords was trained with 100 random vectors from each sample in the training set (T=1).

We describe the cooccurrence information of a texture sample with a cooccurrence histogram. The mapping from the cooccurrence space to the cooccurrence histogram is straightforward. Given a particular cooccurrence vector, the index of the nearest codeword corresponds to the bin index in the cooccurrence histogram. In other words, a codebook of N codewords produces a histogram of N bins. The cooccurrence histogram of a texture sample is obtained by searching the nearest codeword to each vector present in the sample, and incrementing the bin denoted by the index of this nearest codeword. The cooccurrence histogram of a 64x64 texture sample is illustrated in Fig. 1b.

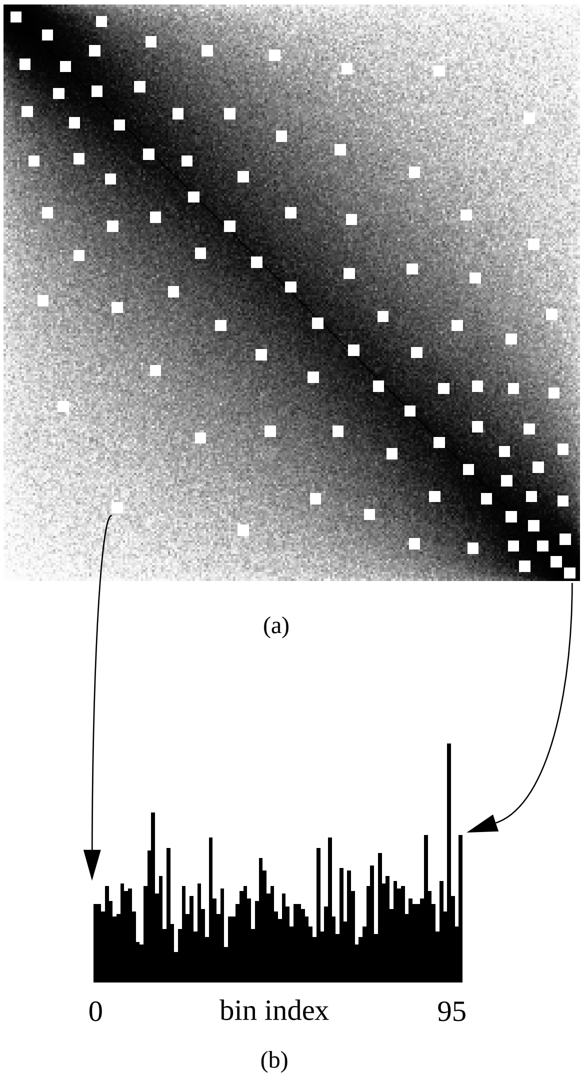

Figure 1: The cooccurrence space of p_2 and its quantization with a codebook of 96 codewords (a) and the cooccurrence histogram of a 64x64 texture sample (b). The indeces of the 96 codewords correspond to the 96 bins in the histogram.

In the following, we denote the cooccurrence histograms obtained using vector quantization as H_D, where D=2,3,5,9, corresponding to the distribution we are estimating. In the experiments we used codebooks of four different sizes, N=96,192,288,384. These codebooks correspond to cooccurrence histograms with roughly 40, 20, 13, and 10 entries per bin, respectively, which keeps them statistically reliable. We always picked R=100 random vectors from each of the 1024 training samples, i.e. 102400 vectors were used in training, and presented them T=1-4 times to the codebook.

3 Experiments

3.1 Texture data

The 32 Brodatz [1] textures used in the experiments are shown in Fig. 2. The images are 256x256 pixels in size and they have 256 gray levels. Each image has been divided into 16 disjoint 64x64 samples, which are independently histogram-equalized to remove luminance differences between textures. To make the classification problem more challenging and generic, three additional samples have been generated from each sample: a sample rotated by 90 degrees, a 64x64 scaled sample obtained from the 45x45 pixels in the middle of the 'original' sample, and a sample that is both rotated and scaled. Consequently, the classification problem involved a total of 2048 samples, 64 samples in each of the 32 texture categories [6].

The performance of a particular classifier was evaluated with ten different randomly chosen training and test sets. The texture classifier was trained by randomly choosing, in each texture class, eight 'original' samples, together with the corresponding 24 transformed samples, as models. The other half of the data, eight 'original' samples and the corresponding 24 transformed samples in each texture class, was used for testing the classifier. In the classification phase a test sample *S* was assigned to the class of the model *M* that maximized the log-likelihood measure:

$$L(S, M) = \sum_{n=1}^{N} S_n \ln M_n \tag{11}$$

where S_n and M_n correspond to the sample and model probabilities of bin *n*, respectively.

3.2 Experimental results

First, we estimated the distribution p_2 with cooccurrence matrices C_2 and C_2^{sym}, to analyze the effect of dismissing the information of texture orientation in opposite directions. Prior to the extraction of the cooccurrence matrices the number of gray levels was reduced from 256 to G with histogram equalization. Fig. 3 shows the average classification accuracies over ten experiments as a function of G. We see that asymmetric C_2 clearly outperforms its symmetrical counterpart, until its average number of entries per bin drops to 15 (G=16). Since C_2^{sym} produces twice as many entries, it manages better with larger matrix dimensions. C_2 peaks at 71.1% (G=7, std.dev. of the 10 experiments is 1.3%), while C_2^{sym} reaches 68.1% (G=11, std.dev. 0.9%).

The corresponding cooccurrence histogram H_2(N=96,T=1) provides a comparable result to C_2 with 70.8% (std.dev. 1.5%), which is sketched for reference in Fig. 3.

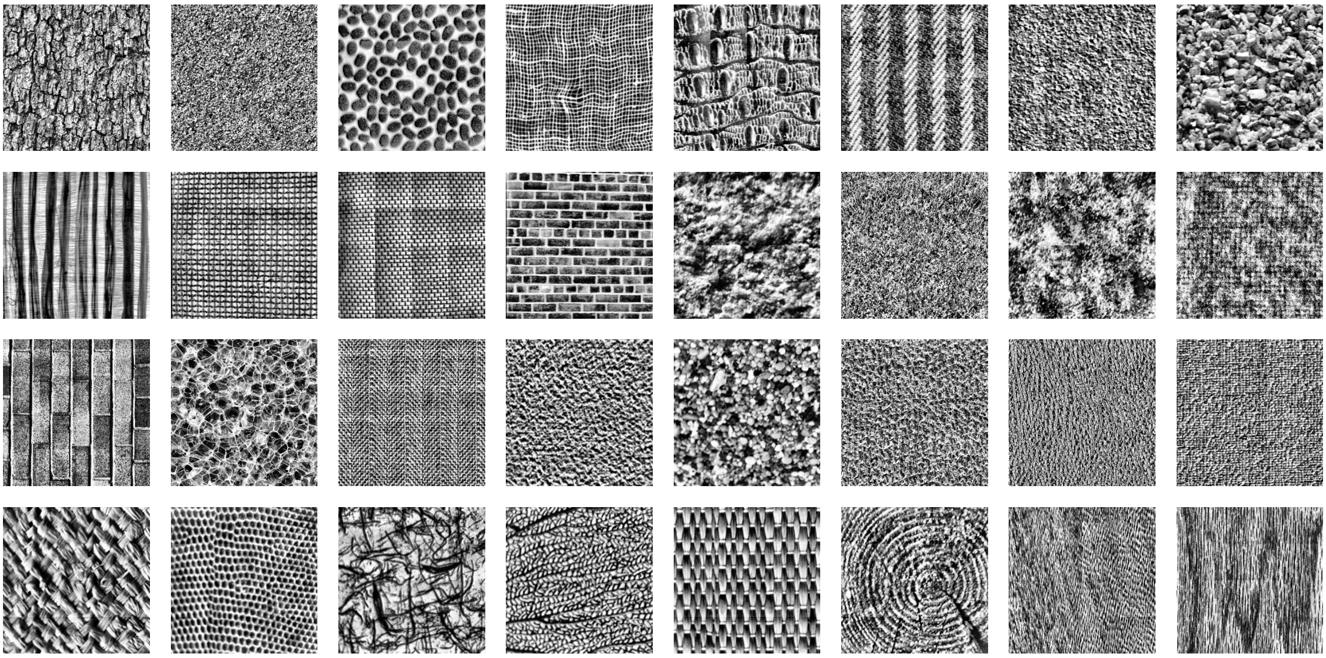

Figure 2: The 32 Brodatz textures used in the experiments.

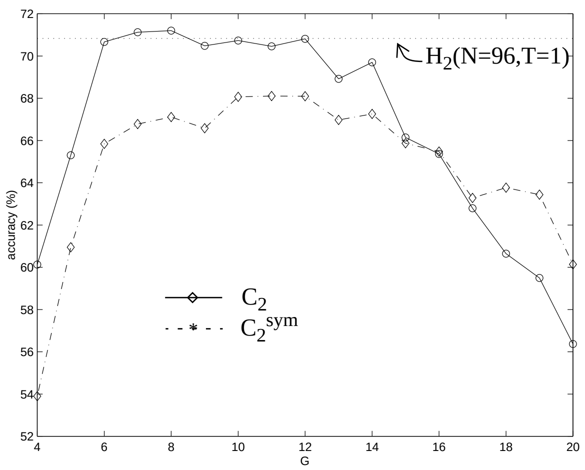

Figure 3: The asymmetric C_2 outperforms its symmetrical counterpart, while the cooccurrence histogram H_2(N=96,T=1) provides comparable performance.

An interesting observation is that increasing the number of codewords slightly decreases the performance, even if the training vectors are presented to the codebook several times. This has to do with the statistical reliability of the histograms, for the higher is the number of codewords, the smaller is the average number of entries per bin. The benefits of vector quantization will become more apparent when we consider the joint cooccurrences of more than two pixels.

Next, we estimated distributions p_3, p_5, and p_9 with the corresponding cooccurrence matrices and histograms. Note that given our log-likelihood statistic L C^{con} equals summing up the individual log-likelihood statistics over the submatrices, i.e. for example $L(C_3^{con}) = L(C_{01}) + L(C_{02})$. Effectively this means that we assume C_{01} and C_{02} to be independent, ignoring their possible correlation.

Fig. 4 shows the average accuracies of C^{con} and C^{acc} matrices as a function of the number of gray levels. We see that it is clearly more beneficial to concatenate individual 2-dimensional matrices instead of summing them up into a single matrix. C_3^{con} peaks at 87.5% (G=8, std.dev. 1.3%), C_5^{con} at 89.3% (G=7, std.dev. 1.2%) and C_9^{con} at 89.4% (G=7, std.dev. 1.1%). Respectively, C_3^{acc} reaches 81.2% (G=11, std.dev. 1.4%), C_5^{acc} 80.1% (G=11, std.dev. 1.3%) and C_9^{acc} 77.9% (G=11, std.dev. 1.5%). The worse results of C_5^{cc} and C_9^{acc} with respect to C_3^{acc} underline the observation that inserting additional cooccurrences into a single matrix only blurs the information. Therefore, it is advisable to use accumulated matrices only if the rotation invariance is a real concern. The reason for the C^{con} matrices peaking with smaller values of G is due to the fact that they have a smaller number of entries per bin than the C^{acc} matrices.

Again, we examined the effect of using symmetrical matrices instead of asymmetric ones, repeating the best classification experiment for each of the six cooccurrence matrices. The performance of C_3^{con} (G=8) decreased by 3.7% to 83.8% (std.dev. 1.0%), C_5^{con} (G=7) by 2.1% to 87.2% (std.dev. 0.8%), and C_9^{con} (G=7) by 1.7% to 87.7% (std.dev. 0.9%). Similarly, the accuracy of C_3^{acc} (G=11) deteriorated by 7.9% to 73.3% (std.dev. 1.4%), C_5^{acc} by 9.2% to 70.9% (std.dev. 0.5%), and C_9^{acc} by 1.9% to 76.0% (std.dev. 0.9%).

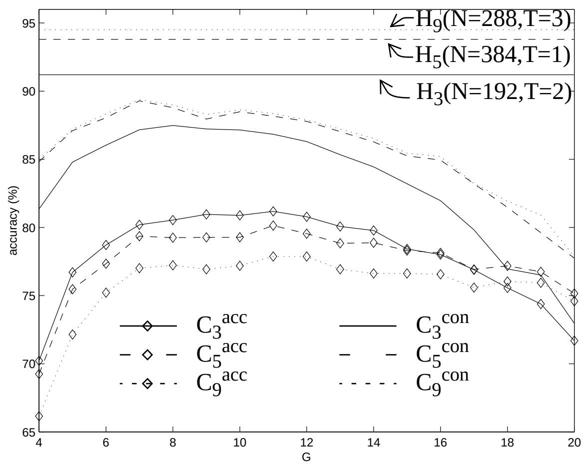

Figure 4: The advantages of vector quantization become apparent in the case of multipixel cooccurrences. Also, it seems to be more beneficial to concatenate matrices instead of summing them.

The average accuracies of corresponding cooccurrence histograms H_3, H_5 and H_9 are also marked in Fig. 4. H_3(N=192,T=2) reaches 91.2% (std.dev. 0.9%), H_5(N=384,T=1) 93.8% (std.dev. 0.6%) and H_9(N=288,T=3) 94.5% (std.dev. 0.7%). This clear improvement over C^{con} can be attributed to the more efficient approximation of the joint multidimensional cooccurrence space, i.e. cooccurrence histograms address the correlation between pixels (pixel pairs).

3.3 Results for GMRF and Gabor energy features

Gray level cooccurrences are a powerful method, if they are employed efficiently. For comparison purposes, the classification problem was also tackled with the Gaussian Markov Random Field (GMRF) and Gabor energy features, which are widely regarded as the state-of-the-art methods in texture analysis. The implementations of GMRF and Gabor energy features were obtained from the MeasTex site, which is a framework for measuring the performance of texture classification algorithms, providing large image databases and source codes of standard paradigms [5].

The GMRF features were computed using the standard symmetric masks, and all models from the 1st order to the 6th were attempted. Additionally, the features of all six models were combined into one large set of 48 GMRF features. The Gabor energy measures were extracted with a filter bank of three different wavelengths (2, 4, and 8 pixels) and four different orientations (0, 45, 90, and 135 degrees), resulting in a set of 12 features. The width of the Gaussian window was set to wavelength/2, and all odd mask sizes between 7x7 and 17x17 pixels were attempted. Again, the features obtained with different mask sizes were combined into one large set of 72 Gabor energy features.

Both the multivariate Gaussian discriminant and the 3-NN classifier were used for classification. When the 3-NN classifier was used, the features were normalized to have a unit variance. We report the results for the classifier which provided the better performance; the Gaussian discriminant in the case of GMRF features and the 3-NN classifier in the case of Gabor energy features. Because the GMRF and Gabor energy features extracted with a particular model or mask size are fairly correlated, the best classification accuracy is not necessarily obtained by using all features simultaneously, due to the curse of dimensionality. For this purpose a stepwise search for best feature combinations was performed during classification. The search included both forward and backward selection of features.

When features extracted with an individual GMRF mask were used, the best classification accuracy was 68.2% (6th order mask, std.dev. 1.5%). When feature selection was done from the set of 48 GMRF features, result of 87.7% (std.dev. 1.2%) was obtained. Similarly, when features computed with a single Gabor filter bank were used, the best result was 87.6% (mask size 15x15, std.dev. 1.3%). When all 72 Gabor energy features were utilized, an average accuracy of 90.2% (std.dev. 1.4%) was achieved.

4 Discussion and conclusions

We showed that an efficient approximation of a high dimensional cooccurrence space can be achieved with a simple codebook and an ‘off-the-shelf’ vector quantization algorithm. Obviously, the performance of the proposed approach could still be improved with a more thorough study on the vector quantization procedure. For example, we picked just 100 random vectors from each training sample, thus using less than 3% of the available training data.

We also pointed out two problems in the conventional use of cooccurrence matrices that should be addressed in order to achieve the best possible classification accuracy. First, asymmetric cooccurrence matrices are preferable over symmetrical ones in this type of a texture classification problem, for they contain information about texture orientation in opposite directions. Second, it is more beneficial to concatenate 2-dimensional matrices computed for pixel pairs in different directions than to sum them up into a single matrix, if the rotation invariance is not an issue.

Our experimental results indicate that cooccurrence matrices are a powerful texture description method, if they are used properly. The performance can be further enhanced by utilizing the high correlation between gray levels of adjacent pixels, and using distributions of signed gray level differences instead of gray level cooccurrences [4].

Acknowledgements

The financial support of the Academy of Finland is gratefully acknowledged.

References

[1] P. Brodatz, *Textures: A photographic album for artists and designers*, Dover, New York, 1966.

[2] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision*, Vol. 1, Addison-Wesley, Reading, 1992.

[3] T. Kohonen, J. Kangas, J. Laaksonen and K. Torkkola, LVQ_PAK: A program package for the correct application of Learning Vector Quantization algorithms, *Proc. International Joint Conference on Neural Networks*, Baltimore, 1992, pp. 1725-1730.

[4] T. Ojala, K. Valkealahti, E. Oja and M. Pietikäinen, Texture discrimination with multidimensional distributions of signed gray level differences, submitted.

[5] G. Smith and I. Burns, Measuring texture classification algorithms, *Pattern Recognition Letters*, Vol. 18, 1997, pp. 1495-1501, *http://www.cssip.elec.uq.edu.au/~guy/meastex/meastex.html.*

[6] K. Valkealahti and E. Oja, Reduced multidimensional cooccurrence histograms in texture classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, 1998, pp. 90-94.