To appear in proc. Visual Communications and Image Processing 1996
17-20 March 1996, Orlando, Florida, USA

Optimal filtering for unsupervised texture feature extraction

Trygve Randen, Vidar Alvestad, John Håkon Husøy

Høgskolen i Stavanger
P.O. Box 2557 Ullandhaug, 4004 Stavanger, Norway
E-mail: t.randen@ieee.org

ABSTRACT

In this paper a technique for unsupervised optimal feature extraction and segmentation for textured images is presented. The image is first divided into cells of equal size, and similarity measures on the autocorrelation functions for the cells are estimated. The similarity measures are used for clustering the image into clusters of cells with similar textures. Autocorrelation estimates for each cluster are then estimated, and two-dimensional texture feature extractors using filters, optimal with respect to the Fisher criterion, are constructed. Further, a model for the feature response at and near the texture borders is developed. This model is used to estimate whether the positions of the detected edges in the image are biased, and a scheme for correcting such bias using morphological dilation is devised. The article is concluded with experimental results for the proposed unsupervised texture segmentation scheme.

Keywords: image processing, pattern recognition, texture, feature extraction, optimal filtering

1 INTRODUCTION

The design of filters yielding optimal texture feature separation has gained increased interest over the past few years. Works by Dunn et al.,[1] Teuner et al.,[2] Cohen and You,[3] Jain and Karu,[4] Singh and Mahalanobis,[5,6] and Randen and Husøy[7] all show promising results. As non-optimal approaches typically use a large number of filters, the major motivation for using optimal filters is the issue of computational complexity.

Most optimal filter design approaches are restricted to supervised problems, but the recently published work by Teuner et al.[2] shows that optimal filter design may be useful even for unsupervised problems. In their approach, the image is decomposed by a pyramidal Gabor transform with dyadic decreasing cell sizes. The cells are examined and compared to find the most significant spectral components, and the corresponding frequencies form the basis for designing the Gabor filters.

However, the Gabor filter is in principle a band pass filter where the only free parameters are the center frequency and the passband size and shape, and we believe that increasing the number of free parameters of the filter also will yield better filters. In this paper we therefore develop an unsupervised texture feature extraction and segmentation technique using optimal two-dimensional linear filters, only restricted to be finite impulse response (FIR) filters. The textures are modelled by their autocorrelation functions. With the feature extraction system of Figure 1, we are then able to get expressions for the means and variances of the extracted features. The filters

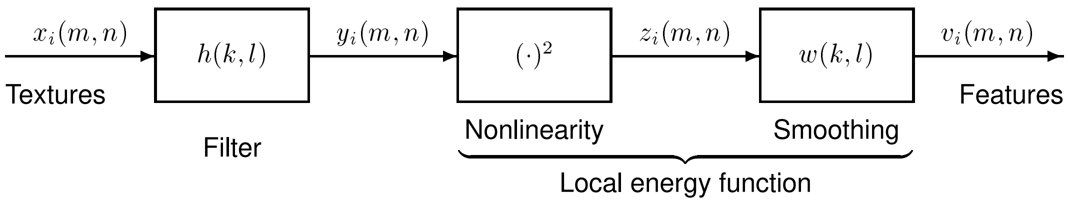


Figure 1: The generic filter-based feature extraction scheme.

are optimized with respect to the Fisher criterion[8] using the approach presented by Randen and Husøy.[7] One filter is designed for each texture pair.

Since these filters are not designed using a single significant frequency per filter, but rather using the autocorrelation functions of the textures, the pyramidal Gabor transform analysis of Teuner et al.[2] is not appropriate to get the texture properties for designing these optimal filters. We propose using a block-based clustering scheme where cells of the image are clustered by a measure on the autocorrelation functions of the cells. Once the cells are clustered, the autocorrelation functions are estimated for the major clusters, and the corresponding optimal filters are designed.

When the optimal features are extracted, we construct the region maps by thresholding the feature images.

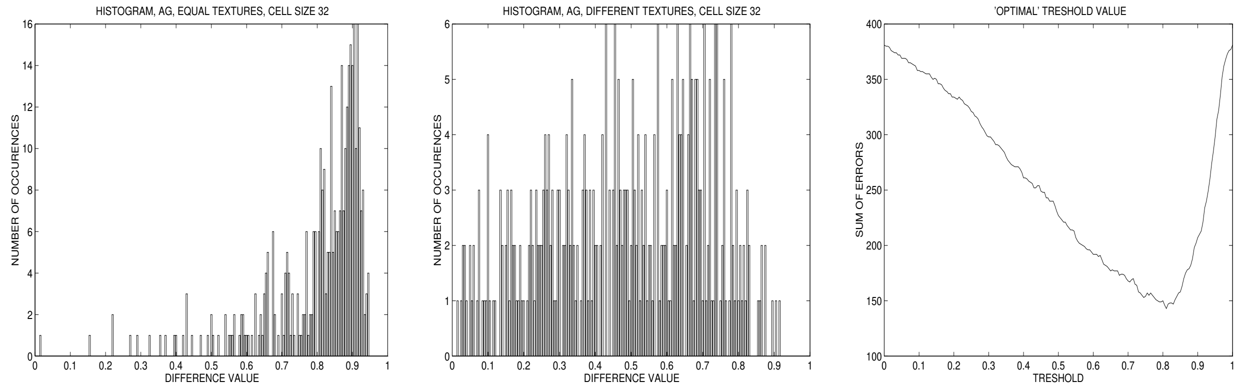
Our modelling and optimization of the feature extraction for maximal feature separation between the textures do not guarantee any specific feature response at the texture boundaries. Hence, depending on the edge response, the detected edges may be severely biased. Examples of this are shown in the experimental results of this paper. To detect if this problem may arise, we develop a modelling scheme for modelling the texture edge responses of the developed filters. If it is detected that the position of the detected edges may be biased, measures can be taken to avoid the biasing. In our experiments, we model horizontal and vertical edges between the textures, and morphological dilation[9] is used to correct edge bias.

In Section 2 we propose a block-based technique for clustering the image. In Section 3 we review the optimal filter design technique as proposed by Randen and Husøy[7] and in Section 4 we propose an extension for modelling the extracted features at the texture edges. In Section 5 we show some experimental results of our proposed technique. The article is concluded with comments and directions for future research, Section 6, and summary, Section 7.

2 BLOCK-BASED CLUSTERING

The first step of the unsupervised feature extraction and segmentation scheme proposed herein is block-based clustering of the input image. The image is divided into equally sized cells, and similar cells are assumed to belong to the same cluster.

It is desirable to compare cells globally to determine if they are similar and should be merged. However, due to the nature of the cell comparison technique developed in this section, the similarity is calculated pairwise between cells. Since the number of cell-pairs increases very rapidly with the number of cells, the similarity is only computed for neighboring cells. Finally, when the image is divided into a presumably small number of clusters of cells, all clusters are compared on a global basis, and similar clusters are merged.



(a) (b) (c)

Figure 2: Histograms of the cell similarity measure with cell size 32×32 pixels for (a) cells containing similar textures and (b) cells containing different textures, and (c) classification error vs. cell similarity.

2.1 Cell-similarity function

As we will see in Section 3, the optimal filter for a texture pair is found from one of the eigenvectors of the matrix $\mathbf{R}_{xx2}^{-1}\mathbf{R}_{xx1}$, where

$$\mathbf{R}_{xxi} = E\left\{\mathbf{x}_i(m,n)\mathbf{x}_i^T(m,n)\right\}, \tag{1}$$

and $\mathbf{x}_i(m,n) = [x_i(m,n)\cdots x_i(m,n-N+1)\cdots x_i(m-M+1,n-N+1)]^T$. $x_i(m,n)$ is the two-dimensional texture field for texture number i.

If $\mathbf{R}_{xx1} = \mathbf{R}_{xx2}$, all the eigenvalues of $\mathbf{R}_{xx2}^{-1}\mathbf{R}_{xx1}$ are equal to one, so if $\mathbf{R}_{xx1}$ and $\mathbf{R}_{xx2}$ are estimated from the same texture, all eigenvalues of $\mathbf{R}_{xx2}^{-1}\mathbf{R}_{xx1}$ are expected to be very similar. On the other hand, if $\mathbf{R}_{xx1}$ and $\mathbf{R}_{xx2}$ are estimated from two different textures, the eigenvalues of $\mathbf{R}_{xx2}^{-1}\mathbf{R}_{xx1}$ will be more spread. Hence, if we estimate $\mathbf{R}_{xx1}$ and $\mathbf{R}_{xx2}$ from two cells in an image, a large spread will indicate that two cells contain two different textures and small spread will indicate similarity.

A suitable function measuring the spread of the eigenvalues is the ratio of the geometric to the arithmetic mean of the eigenvalues,[10]

$$d_{i,j} = \frac{\frac{1}{N}\sum_{i=1}^{N}\lambda_i}{(\prod_{i=1}^{N}\lambda_i)^{\frac{1}{N}}}. \tag{2}$$

This function is equal to unity if all eigenvalues are equal (indicating cells containing equal textures), and approaches zero as the eigenvalue spread increases (indicating different texture cells). The threshold value ξ is introduced to uniquely decide whether or not two cells contains similar textures. If $d_{i,j} < \xi$ the two cells are assumed to contain different textures, and if $d_{i,j} > \xi$ they are assumed to contain the same.

The histogram for $d_{i,j}$ for cells of the same texture, Figure 2(a), and of different textures, Figure 2(b), was computed, using cell size 32×32 pixels. The histograms were computed using 20 different textures from Brodatz.[11] Cells from the same texture was obtained by extracting cells from different locations of a texture. The error of assuming similar when we have different textures and assuming different when we have similar textures is plotted in Figure 2(c) as a function of the threshold.

For cell sizes 128×128, 64×64, and 32×32 pixels, the optimal thresholds, ξ, are 0.935, 0.845, and 0.825 corresponding to error probabilities of 4.3%, 13.7%, and 17.6%. Hence the similarity functions have different optimal threshold values for different cell sizes. As expected, the error increases as the cell size decreases, since the variance of the parameter estimates decreases as the amount of data increases.

2.2 Label connected cells

A cell has four horizontal and vertical neighbors and four diagonal neighbors. Thus, a complete comparison of all the neighbors involves computing eight similarity values. The image is scanned cell by cell from left to right and from top to bottom. If all the neighbors of a cell which already have been labeled have similarity values below ξ, the cell is not similar to its surrounding cells, and is therefore assigned a new label. If two or more labeled neighbors are similar (i.e. $d_{i,j}$ greater than ξ), one of the labels is assigned to the current cell and a note of the equivalences is made. If only one neighbor is similar, its label is assigned to the current cell.

After a complete scan of the image, the similar label pairs are sorted into similarity classes and unique labels are assigned to each cluster. The net result of the previous step is a labeled map, representing the result of the first step of the clustering.

However, one problem still remains. If one of the textures has been splitted into several parts, or if for some reason the merging of the cells yields a splitted region in the clustering of the cells, the method identifies the texture as different textures. To overcome this problem, all regions are compared to each other using the average autocorrelation functions, and regions having a similarity measure above the threshold are merged.

If a region contains just one or two cells after the cluster merging, we assume that these cells are a result of a transition between neighboring texture regions, and the cells are omitted in the subsequent filter design. Hence, the number of classes is assumed to be equal to the number of clusters composed by more than two cells.

3 OPTIMAL FILTER DESIGN

In this section, the optimal filter design approach suggested in[7] is reviewed. The generic experimental setup for texture feature extraction with filters, Figure 1, is used as our model. We assume that we have an input image consisting of two textures, $x_1(m,n)$ and $x_2(m,n)$, and the output for texture $x_i(m,n)$ is the feature image $v_i(m,n)$. In the sequel, we denote the feature mean for texture i by $\mu_{v\,i}$ and the feature variance by $\sigma_{v}{}_i^2$. The filter $h(m,n)$ is optimized.

The filtered image, $y_i(m,n)$, is given by $y_i(m,n) = h(m,n) * x_i(m,n)$, where $*$ denotes two-dimensional convolution. We define $\mathbf{h}$ as the vector formed by lexicographically ordering the rows of $h(m,n)$, and if $h(m,n)$ is an $N \times N$ filter, we define $\mathbf{x}_i(m,n) = [x_i(m,n) \cdots x_i(m,n-N+1) \cdots x_i(m-M+1, n-N+1)]^T$. Then, the filter output at pixel (m,n) for texture i may be written

$$y_i(m,n) = \mathbf{h}^T \mathbf{x}_i(m,n). \tag{3}$$

Similarly, the smoothing operation may be defined as $v_i(m,n) = \mathbf{w}^T \mathbf{z}_i(m,n)$, where we assume that the smoothing filter $w(k,l)$ is an $N_w \times N_w$ rectangular filter.

The obvious aim of an optimal filter for two-texture discrimination is to make the features, i.e. the local energies of the filter outputs, maximally separable. If the feature separation is good, we will achieve good segmentation – and the segmentation may be performed by a simple linear two-class discriminant classifier,[8] typically thresholding. In order to achieve good feature separation, the filter is optimized with respect to the

Fisher criterion[8]

$$J(\mathbf{h}) = \frac{(\mu_{v1} - \mu_{v2})^2}{\sigma_{v1}^2 + \sigma_{v2}^2}. \tag{4}$$

To find the optimal filter we have to find the partial derivative of $J(\mathbf{h})$ with respect to $\mathbf{h}$, which is

$$\frac{\partial J(\mathbf{h})}{\partial \mathbf{h}} = \frac{2(\mu_{v1} - \mu_{v2})\left(\frac{\partial \mu_{v1}}{\partial \mathbf{h}} - \frac{\partial \mu_{v2}}{\partial \mathbf{h}}\right)(\sigma_{v1}^2 + \sigma_{v2}^2)}{(\sigma_{v1}^2 + \sigma_{v2}^2)^2} - \frac{(\mu_{v1} - \mu_{v2})^2\left(\frac{\partial \sigma_{v1}^2}{\partial \mathbf{h}} + \frac{\partial \sigma_{v2}^2}{\partial \mathbf{h}}\right)}{(\sigma_{v1}^2 + \sigma_{v2}^2)^2}. \tag{5}$$

Through modelling the feature extraction system of Figure 1 and using Equation (3), expressions for μ_{vi} and σ_{vi}^2 and their derivatives are found.[7] If it is assumed that the filter outputs, $y_i(m,n)$, may reasonably well be modelled as separable Gaussian AR(1) random fields, it is shown that the optimal filter design entails the solution of the eigenproblem

$$(\mathbf{R}_{xx2}^{-1}\mathbf{R}_{xx1})\mathbf{h} = \lambda \mathbf{h} \tag{6}$$

where $\mathbf{R}_{xxi}$ is the autocorrelation matrix of $\mathbf{x}_i(m,n)$, Equation (1) (note that the autocorrelation matrix of the vector $\mathbf{x}_i(m,n)$ is easily found from the autocorrelation function of the image $x_i(m,n)$). The optimal filter $h(k,l)$ is constructed from $\mathbf{h}$ by inverse lexicographical reordering, where $\mathbf{h}$ is the eigenvector yielding the maximum criterion value $J(\mathbf{h})$.

When optimal features are extracted according to Figure 1, the image is segmented by thresholding the feature values. If a pixel is above the threshold, it is recognized as belonging to one texture, otherwise it is recognized as belonging to the other. This scheme is readily extended to more than two textures by designing one filter per pair of textures. The threshold is determined using the feature histograms for the cell clusters, in a similar way as the threshold in Section 2.1. Special care must, however, be taken near the texture boundaries, as described in the next section.

4 IMPROVED EDGE DETECTION

The features we extract with the scheme of the previous section are optimized to have a large distance between the mean values and low feature variances *within* the texture regions. Hence, thresholding is a suitable classification algorithm. This scheme works well for recognizing the interiors of the texture regions. However, assume that the feature extraction system, Figure 1, responds very strongly to the *edge* between the two textures in the texture pair. Then the detected edge position will be biased, and the region corresponding to the largest expected feature value will become too large. In this section we therefore develop a technique for modelling the feature response near the texture edges for detecting edge localization bias, and a method for correcting the bias.

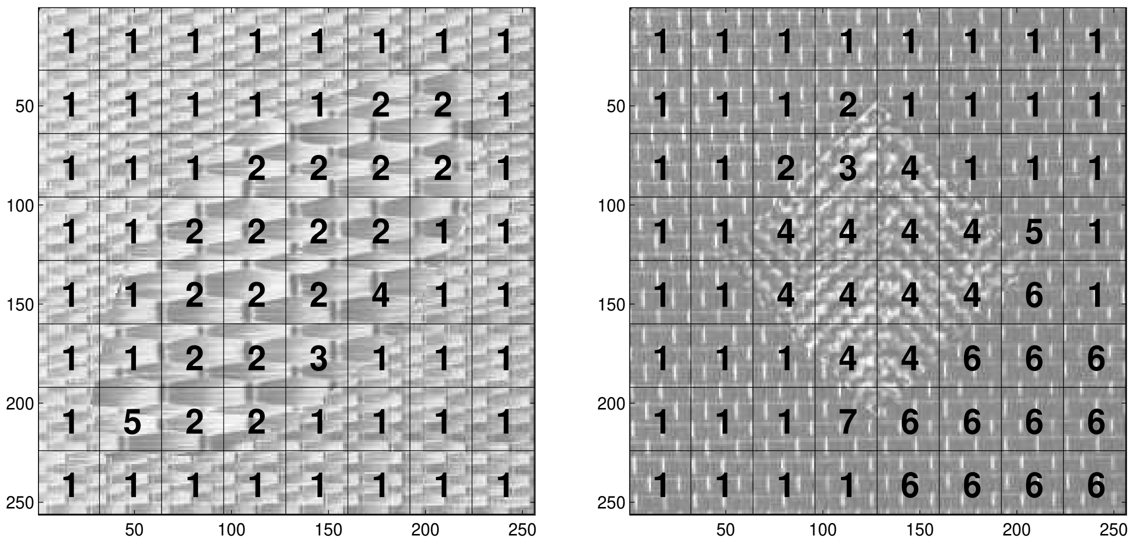
The mean feature value of texture i may be modelled as[7]

$$\mu_{vi} = E\left\{(\mathbf{h}^T\mathbf{x}_i(m,n))^2\right\} = E\left\{w(k,l) * z_i(m,n)\right\} = \mathbf{h}^T\mathbf{R}_{xxi}\mathbf{h},$$

assuming wide sense stationarity (WSS), where $\mathbf{R}_{xxi}$ is given by Equation (1). When all elements of $\mathbf{x}_i(m,n)$ are taken from one texture, $\mathbf{R}_{xxi}$ is constructed from the two-dimensional autocorrelation function of the texture $x_i(m,n)$. However, when computing the expected feature value somewhere on the edge between two textures, the WSS assumption is not any more appropriate. Since $w(k,l)$ is an averaging filter, we then have

$$\mu_{v\,edge}(m,n) = \frac{1}{N_w^2}\sum_{k=0}^{N_w-1}\sum_{l=0}^{N_w-1} E\left\{z(m-k,n-l)\right\} = \mathbf{h}^T\mathbf{R}_{xxi}(k,l)\mathbf{h},$$

where $\mathbf{R}_{xxi}(k,l) = E\left\{\mathbf{x}(m-k,n-l)\mathbf{x}^T(m-k,n-l)\right\}$. By writing the elements of $\mathbf{R}_{xxi}(k,l)$ explicitly for a given edge location relative to the region of support, it is readily seen that $\mathbf{R}_{xxi}(k,l)$ may be created from the



(a) (b)

Figure 3: Label map without merged clusters.

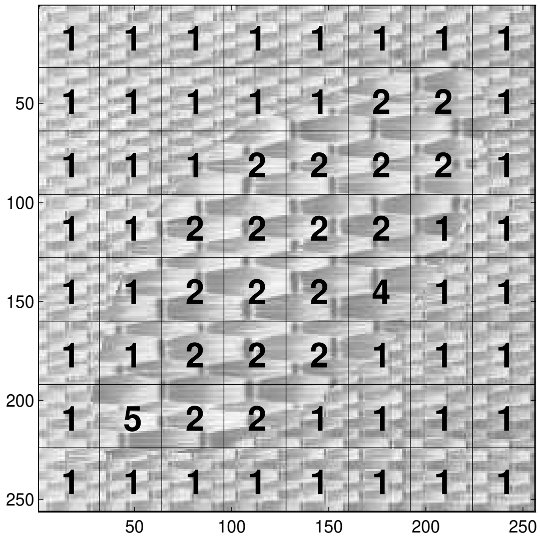
auto- and cross-correlation functions of the two textures making up the edge. Using this, we can model the feature response along edges of any desired orientation and shape.

At the end of the previous section, we proposed a technique for determining the threshold for separating the features for the two textures. If the modelling of the edge proposed here indicates that the feature values along, for instance, a vertical edge cross this threshold two pixels to the right, we know that the right region is two pixels too small. To overcome this limitation, we then extend the right region with morphological dilation[9] with a rectangular kernel extending the region two pixels to the right. Similar analyses may be performed for edges of any orientation. In this paper we perform this kind of modelling and edge correction for horizontal and vertical edges. Results of this modelling and correction are shown in the next section.

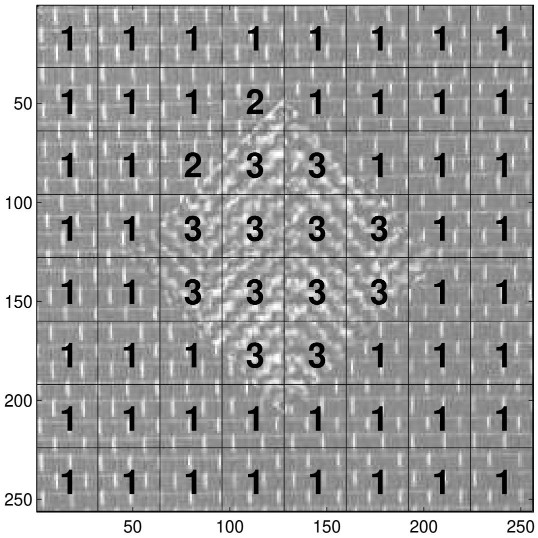
5 EXPERIMENTS

In all experiments we use optimal filters, $h(k,l)$, having $N = 8$, i.e. a region of support of 8×8. Further we use an $N_w \times N_w = 20 \times 20$ averaging filter as our smoothing operator. These values have empirically been found to be appropriate for the textures we analyze. No schemes to determine optimal N and N_w are yet developed. We further use cell size 32×32 in the cell clustering steps.

As described in the Section 2.2, the cell clustering is a two-step process. In the first step, each cell is only compared to its neighbors. However, the input image may contain the same texture in several disconnected regions of the image. Further, since the autocorrelation estimates for small cells have high variance, cells belonging to the same class may be identified as belonging to two different clusters. Consequently some homogeneous texture regions may be splitted into several clusters.



(a)



(b)

Figure 4: Label map with merged clusters.

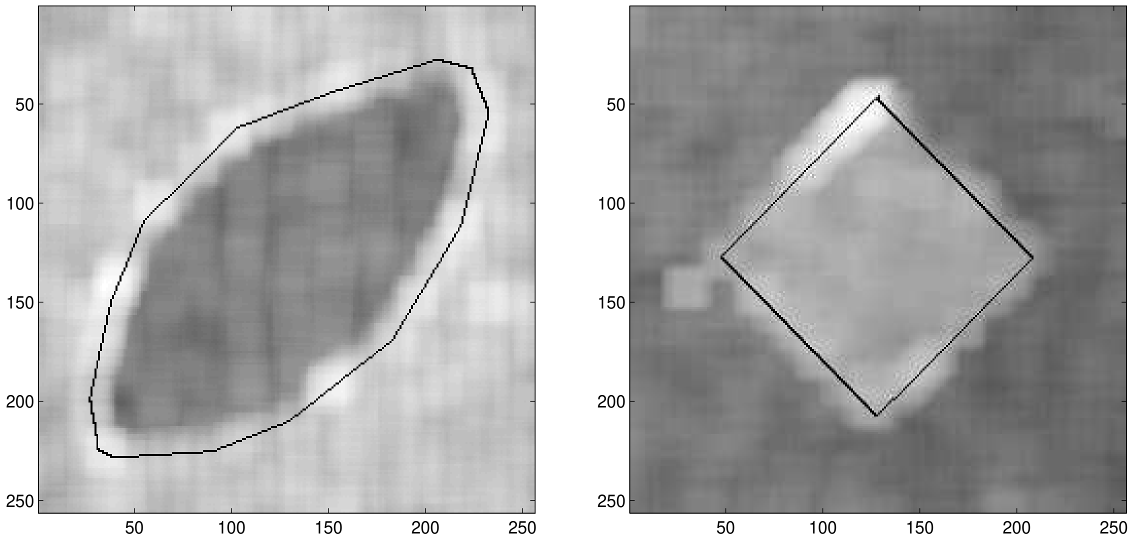
The results of applying the first cell clustering step are shown in Figure 3. The images are constructed by textures D55 and D56, Figure 3(a), and by D17 and D6, Figure 3(b), from Brodatz.[11] The effect of splitting one texture into more than one cluster is especially clear for the image in Figure 3(b) (cluster number 1 and 6). In all figures in this section, the left figure, (a), will be results corresponding to Figure 3(a) and the right figure, (b), will be results corresponding to Figure 3(b).

To overcome the problem of fragmented texture regions illustrated in Figure 3(b), a merging procedure was suggested in Section 2.2. In this merging, all clusters are compared to each other, and clusters found to be similar are merged. The results after this merging are shown in Figure 4. Note especially that the background texture, which was splitted into two large clusters in Figure 3(b), has been merged to one cluster in Figure 4(b).

The merged clusters are used as the basis for determining the autocorrelation functions of the textures in the images, and these autocorrelation functions are then used to determine the optimal filters, as described in Section 3. The extracted feature images are shown in Figure 5.

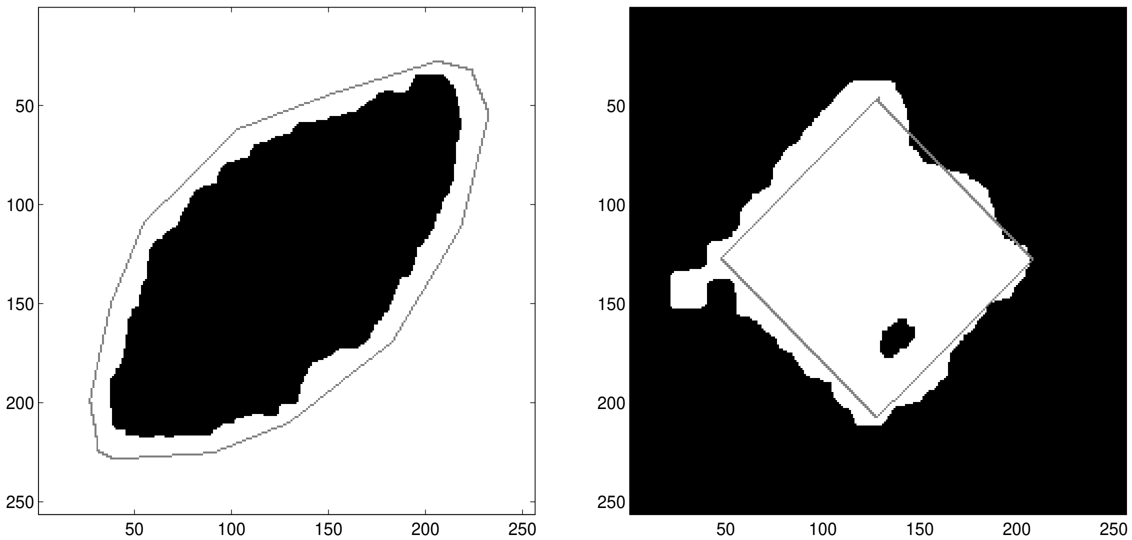
Using the thresholding scheme proposed in Section 3, the class maps constructed by this approach are as shown in Figure 6. As we can clearly see from Figures 5 and 6, the edges are severely biased due to the feature responses at the edges. Modelling the edge responses, we get the scan lines of the expected feature values for horizontal and vertical texture edges illustrated in Figure 7. From our modelling we find that the center texture region is expected to be 11 pixels too small at each vertical border and 8 at each horizontal border for the image of Figure 3(a). I.e. the curves of Figure 7(a) cross the threshold 11 and 8 pixels off the edges. Correspondingly, the center texture of Figure 3(b) is too large by 3 pixels at vertical borders and 5 pixels at horizontal borders. Incorporating these numbers into a morphological dilation kernel as described in Section 4, we get the class maps of Figure 8. We can clearly see that we now get texture boundaries that are very close to the true texture boundary, but at the expense of increasing the misclassified small region inside the center texture of Figure 3(b).

Remember that the reason why the thresholds correspond poorly with the cross-over points of the edge



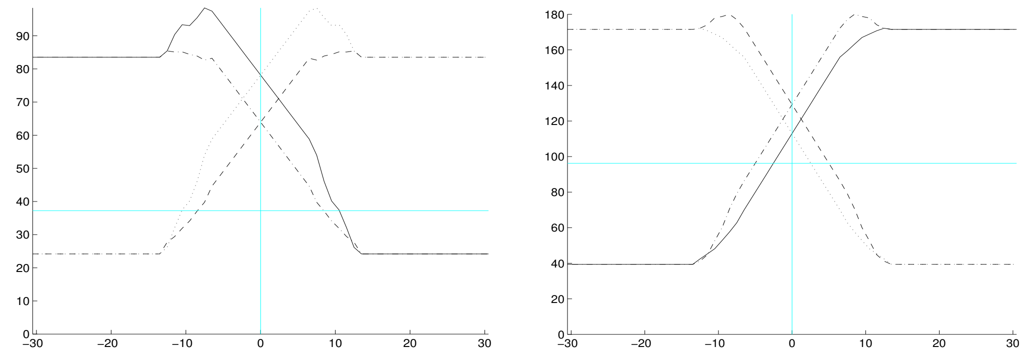
(a) (b)

Figure 5: Feature images. The lines in the images indicate the true texture boundaries. For display purposes, the logarithm of the feature images are shown. Light colors corresponds to high feature values.



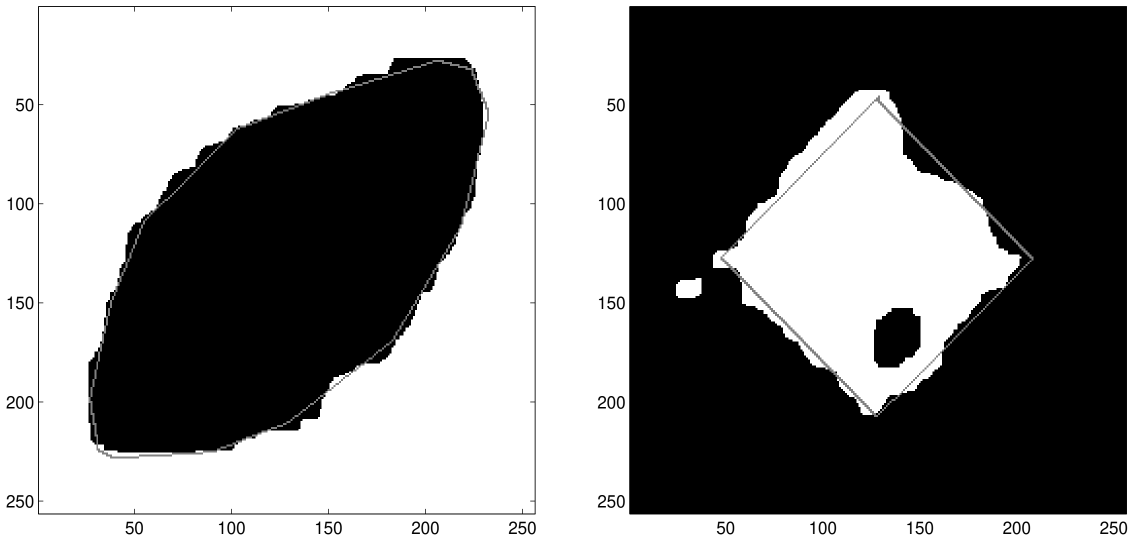
(a) (b)

Figure 6: Region map images. The lines in the images indicate the true texture boundaries.



(a) (b)

Figure 7: Scan line of the expected feature values across a texture edge. The horizontal axis represents pixel numbers relative to the edge, and the vertical axis represents the feature value. The situations illustrated are center texture right of background texture (—), center left of background (···), center below background (−·), and center above background (−−). The vertical lines indicate the true texture boundaries, and the horizontal lines indicate the selected feature threshold value.



(a) (b)

Figure 8: Region map image after morphological dilation. The lines in the image indicate the true texture boundaries.

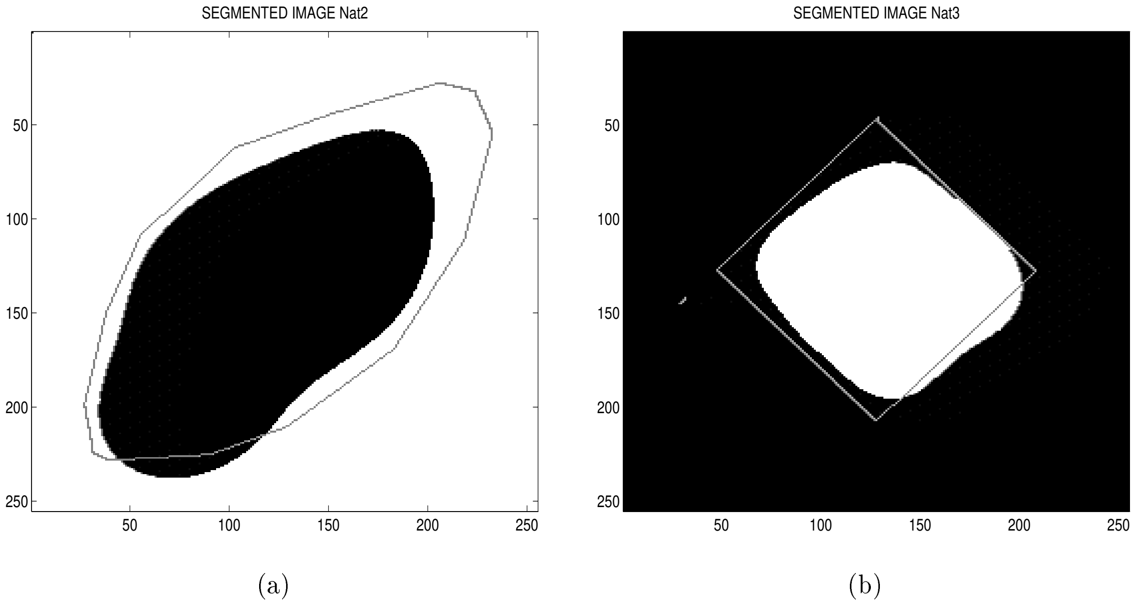


Figure 9: Segmentation result using optimal Gabor filters as proposed by Teuner et al.

responses, Figure 7, is that the optimal thresholds are determined partially by the variance of the features for the various classes. E.g. in Figures 5(a) and 7(a), the extracted features for the background texture has higher variance than the extracted features for the center texture. Hence the optimal threshold is closer to the feature mean of the center texture than the feature mean of the background, as we can see from the threshold-line in Figure 7(a).

The approach by Teuner et al.[2] has also been applied to the dataset used in this paper,[10] and the results were as illustrated in Figure 9. Two examples are by far not enough experimental data to judge which approach is best, but we clearly see that the approach we propose is better for these two images. In addition, notice that in the approach by Teuner et al. more than one filter was used, whereas in our approach only one filter was used.

6 COMMENTS AND FUTURE RESEARCH

As the experiments above indicates, the proposed cell clustering technique works well for some texture images. However, due to the fact that the first step in the cell-clustering is based on region growing where only local neighbors are compared, it is possible that cells which are very different may be combined in one cluster, through a series of neighbors.

A direction for future research is therefore to find a more robust way of clustering cells. One possible approach is hierarchical clustering. Start for instance with cells of size 128×128, split the cells into 4 children. If the similarity measure indicates that one of the children is significantly different from the parent, then keep the splitting, and so on. Since the number of cells of the starting size in the image is small, all cells may be compared to all others, and a more overall view of the clustering may be maintained.

Another possibility is to use ideas from the nearest mean reclassification clustering algorithm.[8] Assume that the number of classes is known, and start with initially assigning all cells to one of the classes randomly. Then compute the mean distance from each cell to all cells of the different classes, and assign the cell to the class with least distance. Continue recomputing mean distance and reassigning until convergence. This would, however, be a very computationally intensive approach.

Yet another direction for future research is the development and evaluation of other criteria than the geometric to arithmetic means, Equation (2), for determining if two cells are samples of the same texture.

7 SUMMARY

Recently, a technique for the design of optimal two-dimensional FIR filters for texture feature extraction was developed,[7] which was experimentally found to be robust. The design technique was, however, restricted to supervised problems. In this paper we have extended this technique to be applicable to unsupervised problems. In order to accomplish this, we proposed a block-based clustering of the textured image, combining cells with a high similarity. We found that the geometric to the arithmetic mean of the eigenvalues of $\mathbf{R}_{xx2}^{-1}\mathbf{R}_{xx1}$, a matrix which is essential in the optimal filter design, was an appropriate similarity measure. The clusters of the cells were then used to estimate the autocorrelation functions of the textures, which in turn were used for optimal filter design. We further developed a model for the feature extractors response to the edge between textures, and proposed a way of taking this into consideration in the constructing of a region map for the textured image. The article was concluded with experiments showing the practical utility of the proposed approach.

8 EXPERIMENTAL DATASET

The images which are used in this article are available for Internet users on the world wide web address *http://www.hsr.no/tele-signal-group* and via anonymous FTP to *ftp.hsr.no* in directory *tele-signal-group/textures/brodatz/*

9 REFERENCES

[1] D. F. Dunn, W. E. Higgins, and J. Wakeley, "Texture segmentation using 2-D Gabor elementary functions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 130–149, Feb. 1994.

[2] A. Teuner, O. Pichler, and B. J. Hosticka, "Unsupervised texture segmentation of images using tuned matched Gabor filters," *IEEE Trans. Image Processing*, vol. 4, pp. 863–870, June 1995.

[3] H. A. Cohen and J. You, "A multi-scale texture classifier based on multiresolution 'tuned' mask," *Pattern recognition letters*, vol. 13, pp. 599–604, Aug. 1992.

[4] A. K. Jain and K. Karu, "Automatic filter design for texture discrimination," in *Proc. Int. Conf. on Patt. Rec.*, pp. 454–458, Int. Assoc. for Patt. Rec., Oct. 1994.

[5] H. Singh and A. Mahalanobis, "Correlation filters for texture recognition and applications to terrain-delimination in wide-area surveillance," in *Proc. Int. Conf. Acoust. Speech, Signal Proc.*, pp. V153–V156, IEEE, 1994.

[6] A. Mahalanobis and H. Singh, "Application of correlation filters for texture recognition," *Applied Optics*, vol. 33, no. 11, pp. 2173–2179, 1994.

[7] T. Randen and J. H. Husøy, "Optimal texture filtering," in *Int. Conf. on Image Proc.*, (Washington, D.C.), pp. 374–377, Oct. 1995.

[8] K. Fukunaga, *Statistical Pattern Recognition.* San Diego: Academic Press, 2 ed., 1990.

[9] R. C. Gonzales and R. Woods, *Digital image processing.* Addison Wesley, 1992.

[10] V. Alvestad, "Unsupervised image texture segmentation with optimized filters," Master's thesis, Høgskolen i Stavanger, Norway, 1995.

[11] P. Brodatz, *Textures: A photographic album for artists and designers.* NY: Dover, 1966.