Chapter 4

Optimal representation filters

As we have seen in Section 2.2 and Chapter 3, many approaches to texture feature extraction with filters apply banks of filters that are more or less determined irrespective of the textures to be discriminated. As we will see in Chapter 6, in some cases even filter banks with a relatively large number of sub-bands are incapable of yielding adequate segmentation of some of the test images. Hence, there is no guarantee that the bank of filters will work properly for a specific set of textures. Another unpleasant consequence of filter bank approaches is that they are to varying degree computationally demanding. For these reasons, the design of *optimal* filters for texture discrimination is gaining increased interest, see Section 2.3. Optimization offers the potential of reduced feature dimensionality and hence reduced computational complexity and/or better feature separation.

In this chapter, an optimization approach based on least mean squared linear prediction error filtering is presented. The linear prediction error filters are optimized with respect to the prediction error. This filter bank design approach is applicable to problems with arbitrary numbers of textures, and the number of filters is equal to the number of textures. Prediction error filtering is reviewed in Section 4.1, least squares auto regressive (AR) parameter estimation is presented in Section 4.2, and some experimental results are given in Section 4.4.

4.1 Prediction error filtering

Linear prediction is an important topic in digital signal processing having many practical applications [94]. A linear predictor in image processing is an equation on the form

$$\hat{x}(m,n) = -\sum_{(k,l)\in\mathcal{N}} \theta(k,l)x(m-k,n-l), \tag{4.1}$$

predicting a pixel value on the basis of neighboring pixels in the neighborhood $\mathcal{N}$ using the coefficients $\theta(k,l)$. The neighborhood $\mathcal{N}$ is not including the pixel itself, $(k,l)=(0,0)$. Assuming wide sense stationarity, a least squared error (LSE) predictor is a predictor minimizing

$$\varepsilon = \sum_{m,n} \left(\hat{x}(m,n) - x(m,n)\right)^2. \tag{4.2}$$

Consequently, the least squares error predictor for a texture will be the predictor yielding maximum similarity between the texture and the predictor output, in the squared error sense. Hence, the energy of the image

$$\begin{aligned} e(m,n) &= x(m,n) - \hat{x}(m,n) \\ &= x(m,n) + \sum_{(k,l)\in\mathcal{N}} \theta(k,l)x(m-k,n-l) \end{aligned} \tag{4.3}$$

is minimized. The image $e(m,n)$ is denoted the *prediction error image*. It is noted that Equation (4.3) expresses essentially a linear filtering operation, thus the term *prediction error filtering*.

Generally the LSE predictors for different textures are different. Hence, for any texture, the predictor yielding minimum mean error energy will be the predictor designed with respect to that texture. This may be used for classification, since the predictor yielding minimum local prediction error energy most likely corresponds to the underlying texture.

4.2 Least squares AR parameter estimation

If we arrange the $x(m-k,n-l)$ and $\theta(k,l)$ terms for all (k,l) in $\mathcal{N}$ in the vectors $\mathbf{x}(m,n)$ and $\boldsymbol{\theta}$, respectively, then Equation (4.3) may be written in vector form as

$$e(m,n) = x(m,n) - \mathbf{x}(m,n)^T\boldsymbol{\theta}. \tag{4.4}$$

If, for example, the region of support, $\mathcal{N}$, is a 3×3 rectangle, we have

$$\boldsymbol{\theta} = \begin{bmatrix} \theta(-1,-1) \\ \theta(-1,0) \\ \theta(-1,1) \\ \theta(0,-1) \\ \theta(0,1) \\ \theta(1,-1) \\ \theta(1,0) \\ \theta(1,1) \end{bmatrix} \tag{4.5}$$

and

$$\mathbf{x}(m,n) = \begin{bmatrix} x(m-1,n-1) \\ x(m-1,n) \\ x(m-1,n+1) \\ x(m,n-1) \\ x(m,n+1) \\ x(m+1,n-1) \\ x(m+1,n) \\ x(m+1,n+1) \end{bmatrix}. \tag{4.6}$$

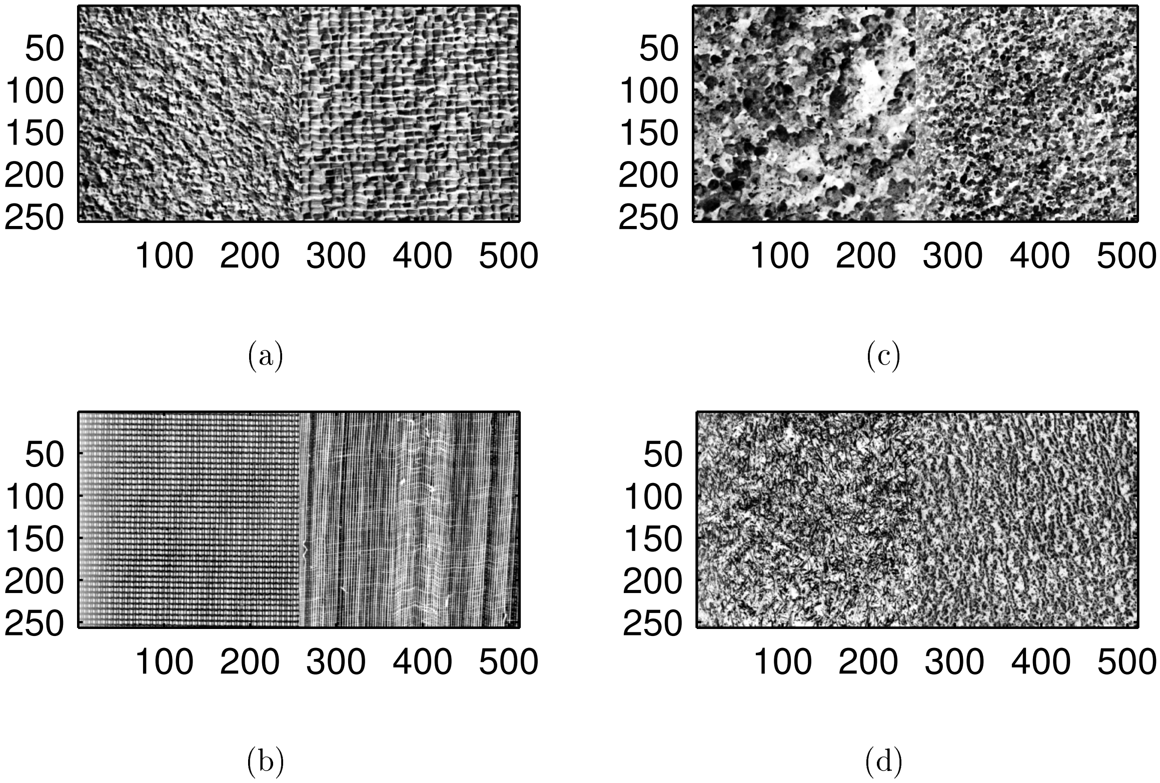

Figure 4.1: Textures pairs (a) D4-D84, (b) D21-D105, (c) D28-D29, and (d) D9-D24 [14] used in the experiments. These test textures have zero mean and have been histogram-equalized.

If we define $\mathbf{e}$ to be the vector of all $e(m,n)$, $\tilde{\mathbf{x}}$ to be the vector of all $x(m,n)$, and $\mathbf{X}$ to be the matrix with the $\mathbf{x}(m,n)^T$'s as the row vectors we may write the linear equation set determined by Equation (4.4) as

$$\mathbf{e} = \tilde{\mathbf{x}} - \mathbf{X}\boldsymbol{\theta}. \tag{4.7}$$

The $\boldsymbol{\theta}$ minimizing ε can then be found by minimizing $\mathbf{e}^T\mathbf{e}$. Finding the $\boldsymbol{\theta}$ yielding the minimum ε entails the solution of

$$\frac{\partial \mathbf{e}^T\mathbf{e}}{\partial \boldsymbol{\theta}} = 0, \tag{4.8}$$

which is found by solving the normal equations [44]

$$\mathbf{X^T X}\boldsymbol{\theta} = \mathbf{X^T}\tilde{\mathbf{x}}. \tag{4.9}$$

The normal equations can be solved effectively by Cholesky decomposition or other methods [38].

4.3 Texture feature extraction and classification

Now, we have established a framework for obtaining prediction error images, $e(m,n)$, one prediction error image corresponding to each texture. Denote the prediction error image for texture number i by $e_i(m,n)$. The first step of the feature extraction is to compute the $e_i(m,n)$-images by filtering. For the parts of an image containing texture number i, we will have

$$\sum e_i^2(m,n) \leq \sum e_j^2(m,n)\ \forall j \neq i. \tag{4.10}$$

In order to have a local estimate of this average, the $e_i^2(m,n)$-images are smoothed. It is noticed that this squaring and the subsequent smoothing correspond to the steps in the local energy function, Section 2.1.1. In the experiments reported here, the Gaussian low pass filter, Equation (2.1) with bandwidth empirically selected to be given by $\sigma_s = 8$ was applied.

Segmentation is obtained by assigning each pixel to the class corresponding to the minimum prediction error. In the experiments herein, the region of support of the predictor is circular with radius four pixels.

4.4 Results

The results given here will only be illustratory, i.e., chosen to illustrate the performance of the method. In order to make the illustrations simple, only two-texture images will be given here. Multiple texture examples are given in Chapter 6. The same texture pairs will be used in this and the next chapter for comparison purposes. Most texture pairs are well discriminated by this method and the methods of the next chapter, but a few texture pairs appear to be more problematic. The texture pairs in Figure 4.1 are used in the experiments. Figure 4.1(a) is easily discriminated by all tested approaches. The rest of the textures are chosen since they are problematic for some of the approaches, and thereby illustrate weaknesses well.

Figure 4.2 illustrates the performances of the prediction error energy features. The upper images are the features corresponding to minimum prediction error filtering for the left texture. The images in the second row correspond to the predictor for the right texture. Since the gray-levels are scaled to utilize the maximum range of the display, it is impossible to judge the distance between the features from, e.g., Figures 4.2(a) and 4.2(b). To illustrate these distances better, the graphs in the third row of Figure 4.2 are provided. These graphs show the average features per column for the left-texture predictor (solid) and the right-texture predictor (dotted). The final row of Figure 4.2 show the corresponding classification results.

The texture pairs of Figures 4.1(a) and 4.1(b) are the images with visually most distinct textures. As we see, these two textures are very well discriminated by the approach. The other two texture pairs are visually more similar and we see from Figures 4.1(k) and 4.1(o) that the predictor outputs are hardly different for the two textures. Hence, no good classification can be expected and the class maps in Figures 4.1(l) and 4.1(p) have little correspondence with the desired class maps.

A likely explanation of the breakdown for the last two images is that they are so similar that the predictor, which is optimal with respect to texture representation, do not yield discriminable features. A couple of other filtering approaches optimized with respect to texture representation have also been presented. These are the eigenfilter approach

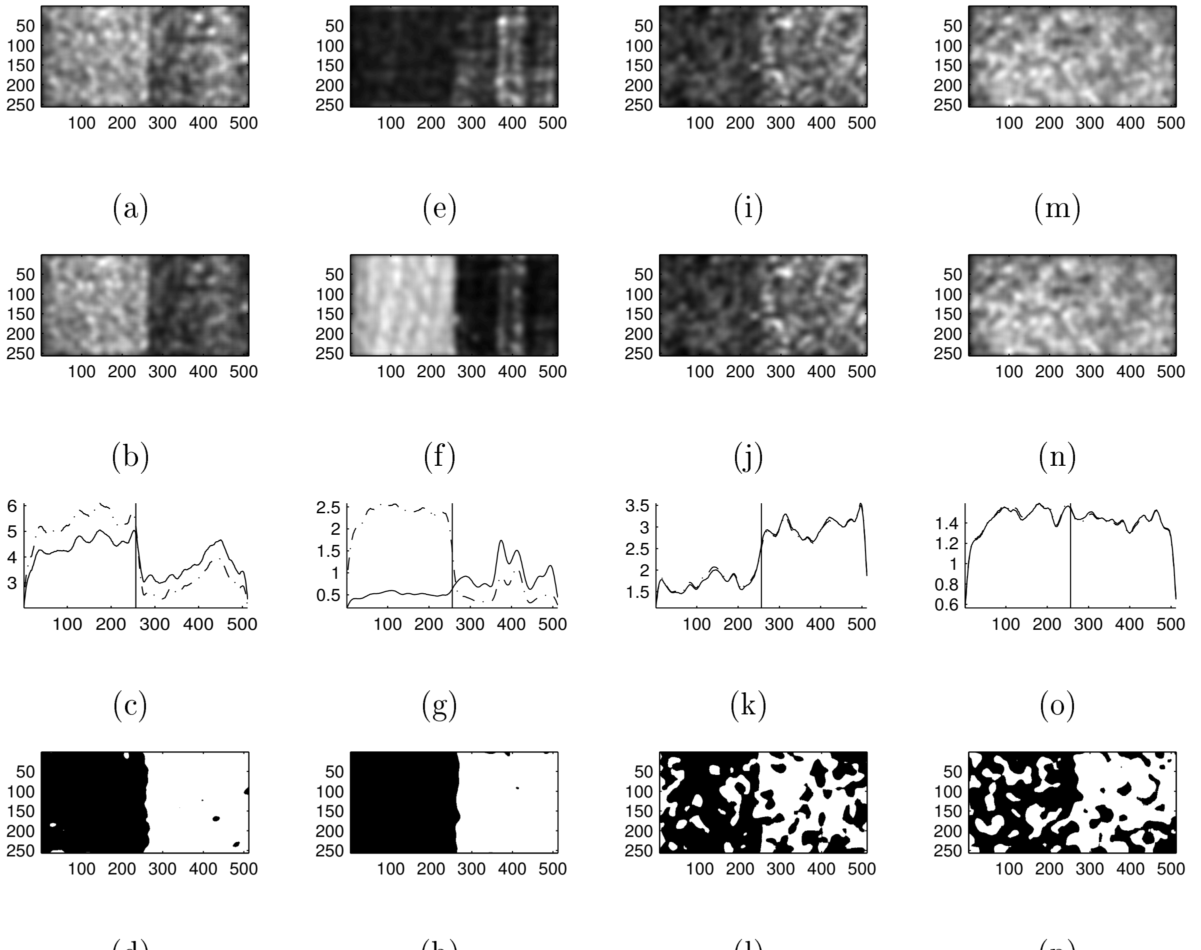

Figure 4.2: (a) Local prediction error energy for D4 in the pair D4-D84 and (b) for D84. (c) The features averaged per column (solid line corresponds to left texture, dotted to right), (d) and finally the corresponding class map. (e-h) Similar figures for D21-D105, (i-l) D28-D29, and (m-p) D9-D24. The vertical lines indicates the true texture boundaries.

of Section 2.3.1 and the optimal representation Gabor filter bank of Section 2.3.2. A discussion on possible implications of optimal representation versus optimal discrimination can be found in Section 2.3.1.

4.5 Summary

In this chapter, a new approach to texture feature extraction using optimal linear predictors has been proposed. The texture features are the local prediction error energies. Experimental results indicate that these features are suitable for segmentation for some textures, but inadequate for others. The method is optimal with respect to texture representation, thus it does *not guarantee* any *discriminable* difference in the prediction errors.