Chapter 6

Texture segmentation: Comparative experiments

Although similar in concept, several quite different filtering schemes have been presented, see Sections 2.2 and 2.3 and Chapters 3, 4, and 5. One of the pioneering approaches was the approach by Laws [62], Section 2.2.1, where a bank of band-pass filters was applied. Subsequent works have focused on different filter bank bases, different sub-band decompositions, and on optimization of the filters for texture feature separation. The large number of different test images applied, along with different system setups, have made comparison of the filtering approaches based on published results very difficult.

A few comparisons between texture feature extraction schemes have been presented. Weszka et al. [140] compared the Fourier power spectrum, second order gray-level statistics, co-occurrence statistics, and gray-level run length statistics features. They concluded that the co-occurrence features were the best of these features. The co-occurrence features were also the best features in a study by Conners and Harlow [24], when compared with run length difference, gray-level difference density, and power spectrum.

du Buf et al. [28], on the other hand, reported that several features, were performing approximately similarly. They evaluated co-occurrence features, fractal dimension, transform and filter bank features, dominant local frequency and orientation features, number of gray-level extrema per unit area, and curvilinear integration features. Filtering features have been compared to the co-occurrence features in some studies, with different conclusions. Strand and Taxt [123] concluded that the co-occurrence features were performing best, while Laws [62], Pietikäinen et al. [92], and Clausi and Jernigan [19] had the opposite conclusion. Different setups, different test images, and different filtering methods may be the reasons for the contradicting results. Ojala et al. [81] compared gray-level difference, Laws filter, covariance, local binary patterns, and complementary feature pairs. They concluded that the gray-level difference features were performing best and noted a particularly poor performance of the Laws features.

It is noted that no extensive evaluation of filtering approaches has been performed. The aim of this chapter is to provide such a comparative study. The system setup will be as illustrated in Figure 2.1. In order to make the results comparable, the focus will be on the filtering part of this system, keeping the other system parameters to a large extent the same for all methods. However, some variation is allowed due to inherent properties

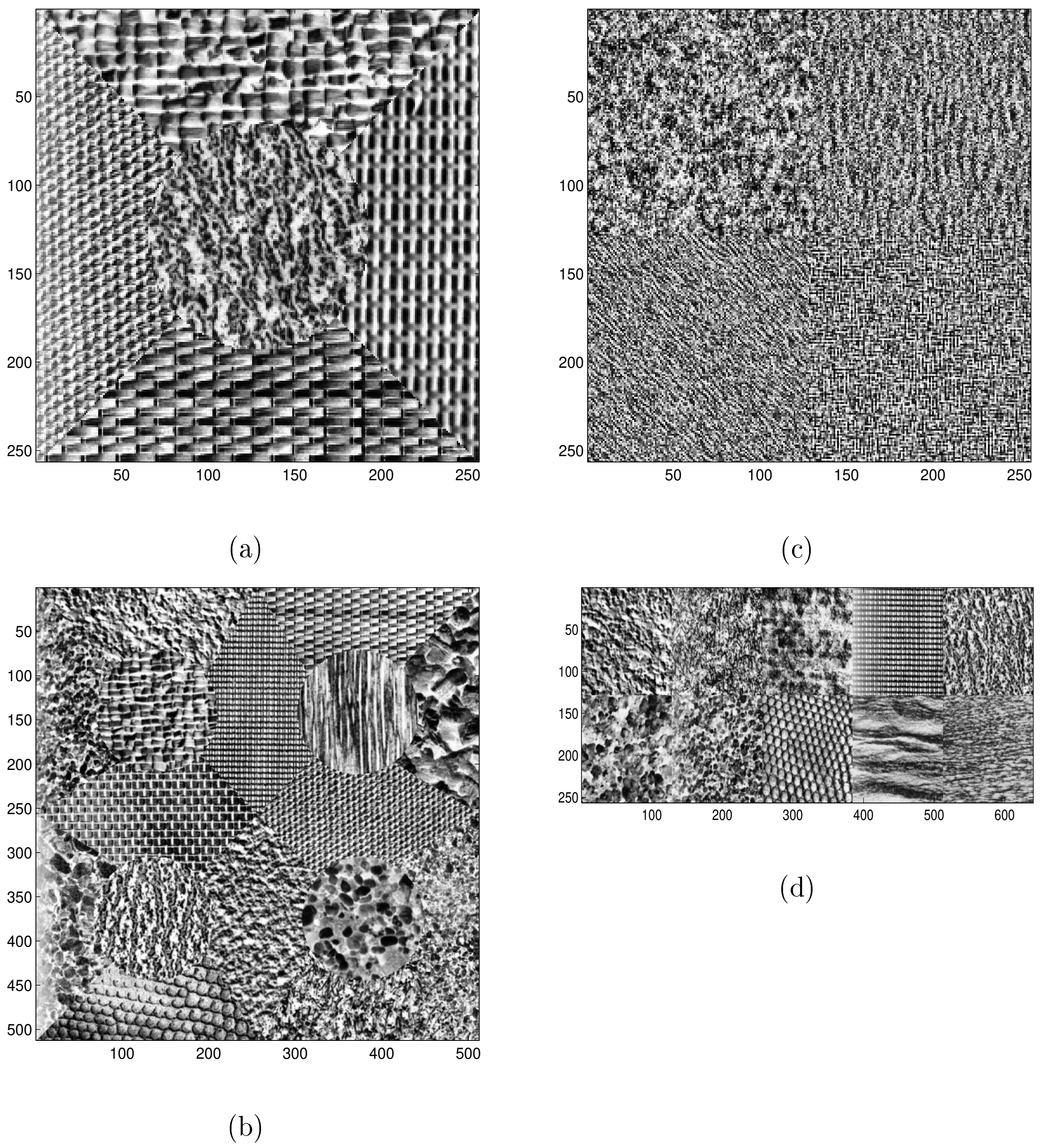
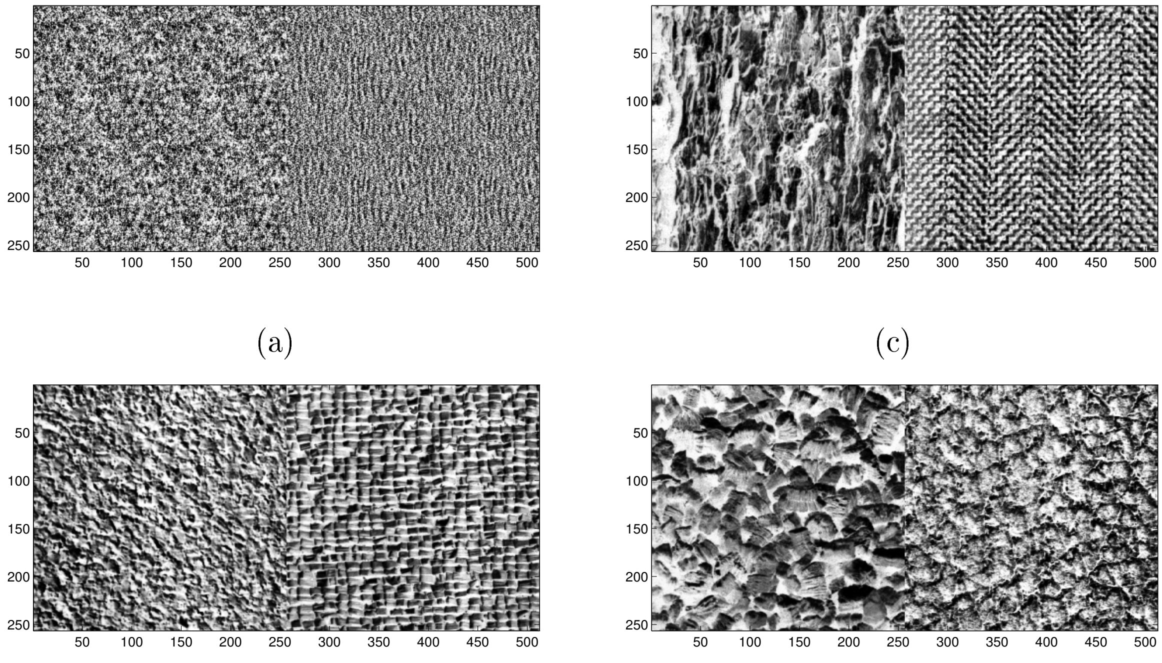


Figure 6.1: Composite texture images used in the experiments. (a) 256×256 pixels 5-texture image consisting of D77, D84, D55, D53, and D24 [14] (the same as shown in Figure 1.1(a)). (b) 512×512 pixels 16-texture image consisting of D3, D4, D5, D6, D9, D21, D24, D29, D32, D33, D54, D55, D57, D68, D77, and D84 [14]. (c) 256×256 pixels 4-texture image consisting of synthetic Gauss Markov random samples [50]. (d) 256×640 pixels 10-texture image consisting of D4, D9, D19, D21, D24, D28, D29, D36, D37, and D38 [14].



(a) (c)

(b) (d)

Figure 6.2: 256×512 pixels two-texture images used in the experiments, (a) the upper two textures of Figure 6.1(c) (replicated to be 256×256 pixels each) (b) D4 and D84 [14], (c) D12 and D17 [14], and (d) D5 and D92 [14].

of the approaches. For reference, the filtering approaches will also the compared with two classes of non-filtering features, model based and statistical features. In the experiments, supervised segmentation results for images with 2 to 16 textures with borders ranging from simple to difficult are compared. Special emphasis is put on making the results realistic, thus the design and test data sets are disjoint. Interestingly enough, this is far from always the case in experiments on supervised texture segmentation, see Section 2.1.2. This also makes the experiments different from experiments in earlier chapters.

This chapter is organized as follows. In Section 6.1, the experimental setup is described, in Section 6.2, the experimental results are presented and discussed, and in Section 6.3 the results are summarized.

6.1 Experimental setup

In the experiments, the different feature extraction approaches are evaluated by performing supervised segmentation on several test images of varying complexity, Figures 6.1 and 6.2. As feature quality criterion, the classification error, "the most common measure of performance for a recognition system" [80], is used. All filter optimization and classifier training are performed on 256×256 subimages of the texture images that are not part of the test images. For the classifier design, features were extracted from these 256×256 images individually and used for training. However, due to the availability of image data,

Identifier	Smoothing filter	Classifier
Laws filters	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
Ring/wedge filters	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$ and $\sigma_s = 16$	LVQ
Dyadic Gabor filter bank	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
Arbitrary Gabor filter bank	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
DCT	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
Co-occurrence	N/A	LVQ
AR	$\sigma_s = 3$	LVQ
Daubechies n	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
f16b	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
F_2_1_smpl	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
F_2_1_09	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
Eigenfilter	$\sigma_s = 8$	LVQ
Opt. repr. Gabor filter bank	$\sigma_s = \frac{1}{2\sqrt{2}f_0}$	LVQ
Prediction error filter	$\sigma_s = 8$	Minimum value selection
Optimal Gabor filter $\sigma =?$	$\sigma_s = 2\sigma$	Threshold
n-filter optimal Gabor filter bank	$\sigma_s = 2\sigma$	LVQ
J_{MS}	$\sigma_s = 8$	Threshold
J_U	$\sigma_s = 8$	Threshold
J_F	$\sigma_s = 8$	Threshold
BackProp. NN, mask size n	N/A	Feed-forward neural net

Table 6.1: Details on the smoothing filters and classifiers used in the experiments. For the smoothing filters, the σ_s-parameter of a Gaussian lowpass-filter is given. Any f_0 is the sub-band center frequency and σ is the parameter of a Gabor filter in the filter bank.

it was not possible to optimize and train on a separate dataset for the images in Figures 6.1(c) and 6.2(a). This makes the results with these images less reliable, but since interesting results are seen with these images, they are still kept.

The local energy function used is the one described in Section 2.1.1, i.e., nonlinearity and smoothing with a Gaussian smoothing filter. For sub-sampled filters, the modification of Section 3.2.2 is used. For filters not having a narrow pass-band, Gaussian low-pass filters with $\sigma_s = 8$ (empirically determined) are used. The nonlinearity used is the squaring with logarithmic normalization after filtering, since this combination was found to be optimal [132]. As described in Section 2.1.2, the classifier is LVQ [58] in most cases. However, with the optimal Gabor filters [30], Section 2.3.3 and the optimal FIR filters, Sections 5.3 and 5.4, thresholding is used. With the optimal prediction error filters, Chapter 4, the image is classified by assigning the class label associated with the minimum error predictor. The classifiers and energy functions applied are summarized in Table 6.1. The method identifiers used in this and subsequent tables are defined in Table 6.2.

A normalization by a global normalization factor is applied for each feature. The global normalization factors are determined as the factors yielding unity variance for the image in Figure 6.1(a).

Identifier	Max number of classes	Description	References
Laws filters	∞	Section 2.2.1	[62]
Ring/wedge filters	∞	Section 2.2.2	[21]
Dyadic Gabor filter bank	∞	Section 2.2.3	[50]
Arbitrary Gabor filter bank	∞	Section 3.1	
DCT	∞	Section 2.2.5	[79]
QMF	∞	Section 3.2	
Co-occurrence	∞	Section 2.4.1	[42]
AR	∞	Section 2.4.2	[128]
Daubechies n	∞	Daubechies n-tap wavelet, Section 2.2.4	[25, 16, 60] [115, 131]
f16b	∞	f16b QMF-filter, Section 3.2	
F_2_1_smpl	∞	F_2_1_smpl QMF-filter, Section 3.2	
F_2_1_09	∞	F_2_1_09 QMF-filter, Section 3.2	
Eigenfilter	∞	Section 2.3.1	[1]
Opt. repr. Gabor filter bank	∞	Section 2.3.2	[12, 11]
Prediction error filter	∞	Section 4.1	
Optimal Gabor filter $\sigma = ?$	2	Section 2.3.3	[30]
n-filter optimal Gabor filter bank	∞	Section 2.3.4	[138, 139]
J_{MS}	$2/\infty$	Sections 2.3.5, 5.3, and 5.4	[66]
J_U	$2/\infty$	Sections 5.3 and 5.4	
J_F	2	Section 5.3	
BackProp. NN, mask size n	∞	Section 2.3.7	[51]

Table 6.2: Summary of identifiers used to identify the methods in the result tables.

6.2 Results

The presentation of the result is divided in three parts. First, in Subsection 6.2.1, heuristicall designed, i.e., fixed, filter banks are considered. Next, in Subsection 6.2.2, critically sampled filter banks (including wavelet transform/packets) are discussed. In Subsection 6.2.3, results with the optimized filters and filter banks are discussed. Finally, in Subsection 6.2.4, a discussion of the test images is given.

6.2.1 Heuristically designed filter banks

The results for all the heuristically designed feature extractors are presented in Table 6.3. The best results from the wavelet frame, arbitrarily decomposed Gabor filter, and QMF filter experiments are shown in the rows labeled “Wavelet”, “Arbitrary Gabor” and “QMF”, respectively. The results from these filter classes are detailed in Table 6.4.

In order to get an increased insight from the experiments, there are multiple “dimensions” in the results. Some approaches allow variations of the filter basis and sub-band decomposition, some are fixed on basis, some on decomposition, and some on both. The

Method	Test image (figure number) 6.1(a)	6.1(b)	6.1(c)	6.1(d)	6.2(a)	6.2(b)	6.2(c)	6.2(d)
Laws filters	18.6%	48.3%	6.3%	41.9%	2.9%	24.8%	3.3%	8.6%
Ring/wedge filters	14.6%	43.8%	6.1%	44.5%	5.5%	3.0%	4.5%	18.4%
Dyadic Gabor filter bank	10.7%	60.1%	10.6%	32.3%	0.9%	3.6%	2.0%	11.3%
Arbitrary Gabor (best, Tab. 6.4)	8.2%	54.8%	14.3%	39.7%	1.0%	6.5%	1.2%	15.6%
DCT	13.2%	40.9%	7.0%	38.2%	6.9%	6.4%	2.2%	2.5%
Wavelet fr. (best, Tab. 6.4)	8.7%	38.2%	5.4%	40.9%	0.7%	4.4%	0.4%	4.7%
QMF (best, Tabs. 6.4)	8.7%	36.4%	3.5%	41.7%	0.7%	6.7%	0.8%	6.0%
Co-occurrence	9.9%	49.6%	5.3%	35.3%	2.0%	1.9%	4.8%	3.3%
AR	19.6%	58.0%	13.9%	56.7%	3.5%	11.3%	1.9%	3.0%

Table 6.3: Classification errors for different heuristically designed texture feature extractors. The numbers for the wavelet frame, arbitrarily decomposed Gabor, and QMF methods represent the best numbers for different setups.

number of feature images also vary much, from eight to 40 features, having considerable impact on the complexity. The complexity, sub-band decomposition, and filter basis issues will be discussed later, but first an overall discussion of the results is given.

Overall discussion

No clear hierarchy of classification performances is observed in Table 6.3, thus no approaches are clear winners. For some images, some approaches are good, for different images, others are good. No methods are consistently poor, but many methods have examples with poor relative performance.

Considering Table 6.3, evidently the "old" Laws and ring/wedge filters not very good. They never stand out as clear winners, and even fail completely in some cases: The Laws approach fails for the images in Figures 6.1(a), 6.1(b), 6.2(b), and 6.2(d). The ring/wedge filters fail with Figures 6.1(a), 6.1(d), 6.2(a), 6.2(c), and 6.2(d). Also the commonly referenced dyadic Gabor filter bank show rather poor performance for some of the test images, Figures 6.1(c), 6.1(d), and 6.2(d). Similar comments apply to the arbitrary Gabor filter bank. The DCT approach did also show poor relative performance for a few images, Figures 6.1(a), 6.2(a), and 6.2(b). However, no "breakdown" of the DCT features was observed.

With the QMF and wavelet approaches, multiple experiments were performed for each entry in Table 6.3. These results are summarized in Table 6.4. The QMF and wavelet frame approaches are among the best for most images. Examples of the classification results for the best QMF filter bank are shown in Figure 6.3. These results correspond to the textures in Figures 6.1(a) and 6.1(b). Most of the best results were obtained with the same sub-band decomposition, Figure 2.6(d), but the performances of the filter bases were less consistent. Further discussions on bases and decompositions follow later.

The co-occurrence method was quite good for most images, with poor relative results only for Figures 6.1(b) and 6.2(c). The AR features, on the other hand, were poor for most images. However, the best filtering approaches yield classifications that are as good as or better than the co-occurrence and AR methods in practically all cases.

It is a quite common practice in texture segmentation to pick training data from

Image in Figure 6.1(a)

Basis	Subband decomp. (fig. no) 2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	14.9%	13.0%	18.6%	8.2%
Daubechies 4	15.6%	18.0%	23.8%	8.7%
Daubechies 6	14.8%	18.6%	30.5%	10.7%
Daubechies 8	14.7%	43.5%	34.4%	12.1%
Daubechies 10	18.2%	24.3%	39.3%	15.9%
f16b	9.5%	13.2%	28.4%	8.7%
F_2_1_smpl	8.9%	15.0%	22.8%	12.3%
F_2_1_09	11.2%	21.2%	80.7%	15.7%

Image in Figure 6.1(b)

Basis	Subband decomp. (fig. no) 2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	60.4%	60.1%	62.7%	54.8%
Daubechies 4	48.8%	46.7%	60.1%	38.2%
Daubechies 6	46.4%	50.8%	63.3%	39.7%
Daubechies 8	46.5%	53.2%	68.9%	43.8%
Daubechies 10	47.3%	57.4%	72.8%	48.7%
f16b	42.4%	45.9%	55.1%	36.4%
F_2_1_smpl	40.7%	43.8%	60.8%	37.1%
F_2_1_09	44.3%	47.0%	58.8%	39.8%

Image in Figure 6.1(c)

Basis	Subband decomp. (fig. no) 2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	20.5%	18.2%	17.1%	14.3%
Daubechies 4	8.2%	8.9%	12.7%	5.4%
Daubechies 6	9.8%	12.1%	15.1%	9.8%
Daubechies 8	13.9%	15.5%	23.5%	13.3%
Daubechies 10	18.2%	20.1%	27.0%	18.1%
f16b	7.9%	9.1%	13.9%	3.5%
F_2_1_smpl	10.2%	8.9%	13.7%	4.9%
F_2_1_09	9.6%	10.0%	24.6%	4.6%

Image in Figure 6.1(d)

Basis	Subband decomp. (fig. no) 2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	45.9%	43.8%	53.7%	39.7%
Daubechies 4	41.2%	44.5%	47.0%	40.9%
Daubechies 6	43.2%	47.0%	50.1%	44.3%
Daubechies 8	44.8%	51.9%	53.5%	45.0%
Daubechies 10	46.4%	51.9%	56.9%	48.5%
f16b	47.0%	48.9%	46.6%	39.8%
F_2_1_smpl	47.2%	46.8%	42.9%	41.7%
F_2_1_09	47.1%	49.4%	49.6%	44.2%

Image in Figure 6.2(a)

Basis	Subband decomp. (fig. no) 2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	3.0%	1.0%	1.0%	1.1%
Daubechies 4	2.4%	0.7%	2.3%	1.4%
Daubechies 6	1.3%	1.9%	4.0%	2.6%
Daubechies 8	2.5%	3.5%	6.4%	3.4%
Daubechies 10	4.2%	4.8%	7.8%	4.4%
f16b	1.1%	0.5%	1.2%	0.7%
F_2_1_smpl	2.6%	2.5%	3.6%	2.1%
F_2_1_09	2.5%	2.3%	3.4%	2.0%

Image in Figure 6.2(b)

Basis	Subband decomp. (fig. no) 2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	9.2%	6.9%	7.9%	6.5%
Daubechies 4	15.5%	19.2%	13.1%	5.7%
Daubechies 6	16.1%	20.3%	11.9%	4.7%
Daubechies 8	14.8%	17.8%	13.0%	4.9%
Daubechies 10	15.4%	17.9%	14.5%	4.4%
f16b	16.0%	21.6%	16.7%	8.1%
F_2_1_smpl	15.3%	20.1%	13.8%	7.2%
F_2_1_09	14.8%	21.2%	18.6%	6.7%

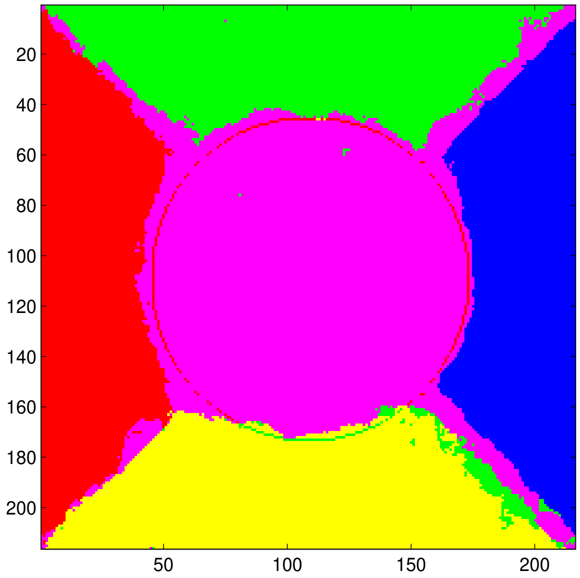
Image in Figure 6.2(c)

Basis	Subband decomp. (fig. no) 2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	2.0%	1.5%	9.5%	1.2%
Daubechies 4	1.3%	1.0%	4.0%	0.6%
Daubechies 6	0.9%	0.7%	4.6%	0.6%
Daubechies 8	1.0%	0.4%	3.3%	1.3%
Daubechies 10	0.9%	0.5%	2.4%	1.2%
f16b	0.3%	0.4%	6.2%	0.8%
F_2_1_smpl	1.5%	2.4%	4.3%	2.3%
F_2_1_09	1.2%	2.3%	6.2%	2.1%

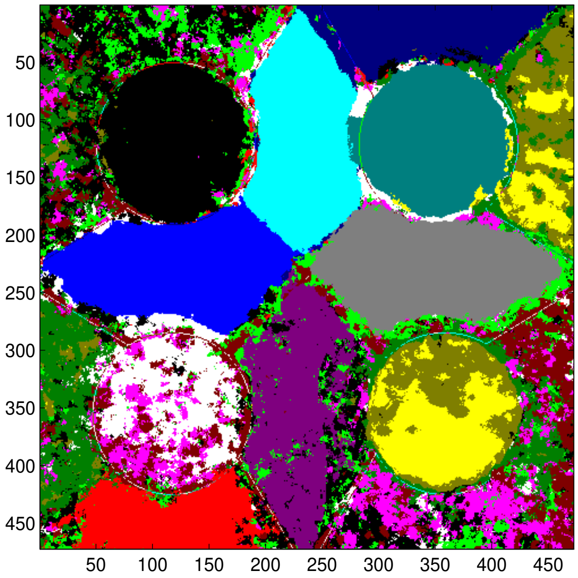
Image in Figure 6.2(d)

Basis	Subband decomp. (fig. no) 2.6(a)	2.6(b)	2.6(c)	2.6(d)
Gabor	20.9%	16.5%	27.5%	15.6%
Daubechies 4	13.4%	13.5%	6.5%	8.2%
Daubechies 6	12.3%	11.4%	6.4%	6.1%
Daubechies 8	9.3%	8.0%	6.9%	4.7%
Daubechies 10	9.0%	7.4%	7.2%	5.1%
f16b	11.2%	10.9%	4.1%	8.2%
F_2_1_smpl	10.4%	10.4%	6.9%	7.4%
F_2_1_09	10.6%	10.1%	7.6%	6.0%

Table 6.4: Classification errors for different wavelet, Gabor, and QMF full rate filter banks.



(a)



(b)

Figure 6.3: Classification results for the QMF "f16b" filter bank for the textures in Figures 6.1(a) and 6.1(b).

the test features, e.g., [32, 51, 104, 123]. This practice has primarily not been used in these experiments. In order to simulate a real application environment, all classifier training has been performed on features from separate subimages. Furthermore, no edge or texture border specific training data are picked. These issues must be kept in mind when comparing the results with previously published results. For a further analysis of the impact of this choice, see Section 2.1.2.

Filter basis

Table 6.4 gives the numbers for the performances of the wavelet frames, QMF, and arbitrary Gabor filter banks. For each column in any of the sub-tables of Table 6.4, the decomposition is constant. Hence, any variation in performance is due to differences in the filter basis. For some of the image/decomposition combinations, there is hardly any impact of the filter basis at all, e.g., Figure 6.2(a). However, for most multi-texture images and some two-texture images, there is a noticeable performance variation as a function of filter basis. For some images, some filter bases are consistently poor, e.g. the Gabor filter for Figure 6.2(d). In other cases, the performance of some of the bases is very dependent on the decomposition, see next subsection.

It can be noted that the larger Daubechies bases are rather poor. This may be caused by two effects. First, the stop-band attenuations of the Daubechies filters are not as good as for the QMF filters. Next, the response is not symmetric, yielding alignment problems with the filter responses and less accurate texture boundary localization.

Many papers propose filtering approaches based on the Gabor basis, e.g., [11, 12, 13, 30, 31, 50, 127, 138, 139]. However, from these experiments, we find no evidence that the Gabor basis is superior. If we may draw any conclusion on the appropriateness of the Gabor basis from these experiments, it is rather the opposite.

Sub-band decomposition

For some images, there is hardly any impact of the decomposition, while for others, the impact is considerable. For the image in Figure 6.1(a), the decomposition has large impact, whereas for Figure 6.2(a), there is hardly any. It appears that for the most complex or difficult images, the impact of the decomposition is large, while it is small for simpler images.

For most images and filter bases, the best results were obtained with the decomposition of Figure 2.6(d). The drawback with this decomposition is the high dimensionality of the feature vectors; 40. This dimensionality imposes high computational complexity. However, for many of the images, the relatively low-dimensional decomposition of Figure 2.6(a) yielded good results.

Another conclusion that may be drawn from the experiments is that increasing the depth of a dyadic decomposition do not improve the results. All the decompositions of Figures 2.6(a-c) are dyadic. Of these, the decomposition of Figure 2.6(a), the most shallow, is overall best

Complexity issues

When comparing the results, complexity issues should be kept in mind. Clearly, most of the filtering approaches are significantly less computationally complex than the co-occurrence and AR methods. For the filtering approaches, filtering and classification are the main contributors to the total complexity of the system. The DCT approach has low filtering complexity (short separable filter masks) and low feature dimensionality, yielding relatively low classifier complexity. However, the results are only moderately good. The Laws, ring and wedge, Gabor, and some of the wavelet and QMF filter banks have high filtering complexities, whereas the IIR QMF filters have low complexities.

The Laws approach yields 25 features, the ring/wedge 11, the dyadic Gabor 20, the DCT 8, and the wavelet, arbitrary Gabor, and QMF filter banks yield from 10 to 40 features. Low feature count yields low classification complexity.

Summary

From Tables 6.3 and 6.4, no single feature extraction method is consistently superior and the impression is rather confusing. The main insight we may learn from this, is that when designing a filter bank for a particular texture problem, it is advantageous to test multiple approaches.

An inherent assumption made when filter banks are used for texture feature extraction, is that the spectral signatures of different textures are different,[1] see Section 2.1. For problems with many textures or spectrally similar textures, it is reasonable to assume

[1] Analog discussions do also apply to other representation spaces than the frequency space.

that the spectral decomposition has a significant effect. This intuitive idea is supported by the experiments, and we see great importance of the bases and decompositions for difficult two-texture images and for most multi-texture images.

In many circumstances, the effort in testing numerous filter banks may be too large. For these applications, one filter bank should be recommended. This filter bank should have a reasonably good performance in most cases. One such choice is the bank with the QMF filter "F_2_1_smpl" using the sub-band decomposition in Figure 2.6(d). This filter bank has very low filtering complexity, see Section 3.2.3, but the feature dimensionality is quite high. An alternative, with lower feature dimensionality and overall second best results, is the decomposition of Figure 2.6(a).

6.2.2 Critically sampled filter banks

Considering the wavelet frame approaches, Table 6.4, relative to the results from the wavelet transform and wavelet packet approaches, Table 6.5, a few conclusions may be drawn. First of all, the filter responses for the wavelet transforms and packets are simply subsampled versions of the ones obtained by the overcomplete wavelet frames. We see that, as expected, using decimated filter outputs overall degrades the results slightly. The major cause is worse edge resolution due to the subsampling. However, the significantly decreased computational complexity (discussed in Section 3.2.3) should be kept in mind.

Next, we see that the trend in the results is the same with the critically sampled approaches as it was with the full rate approaches. That is, image, base, and decomposition combinations that yielded poor relative results in full rate do also yield poor results when critically sampled, and vice versa.

6.2.3 Optimal filters

As has been stressed earlier, a heuristically designed filter bank may yield a tremendous computational complexity. It may also give unsatisfactory results. It is, for instance, noted that the best filter banks in many cases had feature dimensionalities as high as 40. Some reduction in dimensionality may be obtained by feature selection [52], but for good results, a relatively high dimensionality is still needed [52].

Another approach is filter or filter bank optimization. The results from the optimal filtering experiments are presented in Table 6.6. Some of the optimization approaches are only applicable for two-texture cases, and the multi-texture table entries for these are marked by "N/A". The impression with the optimization approaches is much more clear than it was with the heuristic approaches. Some approaches are less powerful and some are rather good.

Despite the fact that the training times for the *back propagation designed masks* were significant, the results were not useful for *any* of the images in this test. Useful results were only obtained for some very simple texture pairs. One example is shown in Figure 6.4. Improvements using a modified scheme have been reported [51], but has for implementational and complexity reasons not been tested here.[2]

[2] The results produced here were actually produced using the same software that was applied for the

Image in Figure 6.1(a)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	22.8%	22.6%	22.7%	11.9%
Daubechies 6	17.6%	15.2%	29.7%	12.3%
Daubechies 8	21.4%	27.2%	32.6%	13.9%
Daubechies 10	21.0%	26.0%	36.5%	18.9%
f16b	17.2%	15.2%	25.4%	10.8%
F_2_1_smpl	15.5%	16.5%	26.1%	17.6%
F_2_1_09	17.1%	19.8%	26.4%	21.4%

Image in Figure 6.2(a)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	4.5%	1.3%	0.4%	0.6%
Daubechies 6	4.5%	0.2%	1.2%	1.2%
Daubechies 8	4.9%	2.6%	1.6%	2.2%
Daubechies 10	7.1%	3.6%	3.1%	2.1%
f16b	4.5%	0.8%	0.9%	0.6%
F_2_1_smpl	5.1%	3.4%	5.8%	3.3%
F_2_1_09	4.3%	4.1%	5.8%	2.5%

Image in Figure 6.1(b)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	58.1%	52.5%	66.5%	41.5%
Daubechies 6	56.2%	55.2%	65.8%	44.8%
Daubechies 8	56.7%	58.2%	62.6%	46.8%
Daubechies 10	56.4%	59.1%	65.7%	53.9%
f16b	50.2%	51.5%	54.4%	39.1%
F_2_1_smpl	51.1%	52.9%	58.6%	41.4%
F_2_1_09	52.9%	54.3%	61.3%	45.3%

Image in Figure 6.2(b)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	22.5%	25.7%	23.4%	11.0%
Daubechies 6	22.2%	23.3%	19.7%	11.2%
Daubechies 8	21.5%	25.9%	21.7%	8.5%
Daubechies 10	18.8%	19.8%	20.4%	8.2%
f16b	22.2%	24.2%	17.5%	10.5%
F_2_1_smpl	22.3%	22.4%	19.4%	11.7%
F_2_1_09	20.6%	22.7%	18.4%	10.6%

Image in Figure 6.1(c)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	15.9%	10.4%	11.9%	4.8%
Daubechies 6	15.9%	12.2%	12.3%	6.4%
Daubechies 8	18.1%	17.1%	20.0%	12.3%
Daubechies 10	20.4%	22.7%	28.8%	14.0%
f16b	13.2%	9.5%	12.0%	4.1%
F_2_1_smpl	16.2%	15.4%	24.5%	11.2%
F_2_1_09	16.8%	16.9%	20.6%	12.0%

Image in Figure 6.2(c)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	2.8%	1.1%	2.4%	1.3%
Daubechies 6	2.7%	1.8%	3.6%	1.4%
Daubechies 8	2.2%	1.3%	1.8%	1.3%
Daubechies 10	2.9%	0.8%	1.6%	1.5%
f16b	1.0%	0.8%	1.0%	0.5%
F_2_1_smpl	2.4%	3.1%	6.7%	3.9%
F_2_1_09	2.3%	3.2%	7.1%	3.4%

Image in Figure 6.1(d)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	49.9%	53.0%	52.6%	48.0%
Daubechies 6	47.1%	47.4%	56.4%	46.6%
Daubechies 8	51.0%	53.2%	54.9%	49.7%
Daubechies 10	52.5%	56.9%	58.1%	50.1%
f16b	48.2%	43.3%	45.6%	45.1%
F_2_1_smpl	48.3%	47.0%	51.8%	49.9%
F_2_1_09	50.9%	48.2%	54.6%	49.1%

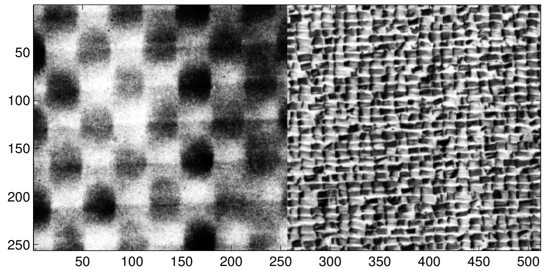
Image in Figure 6.2(d)

Basis	Subband decomp. (fig. no)			
	2.6(a)	2.6(b)	2.6(c)	2.6(d)
Daubechies 4	15.5%	15.6%	9.7%	10.0%
Daubechies 6	12.9%	13.2%	12.1%	7.9%
Daubechies 8	10.6%	8.1%	7.3%	6.8%
Daubechies 10	8.5%	8.4%	9.1%	7.3%
f16b	10.7%	11.0%	7.8%	7.8%
F_2_1_smpl	9.5%	9.7%	9.0%	6.9%
F_2_1_09	9.9%	11.4%	11.3%	7.4%

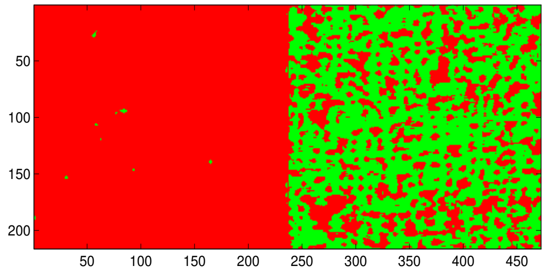
Table 6.5: Classification errors for different wavelet transforms/packets and critically decimated QMF filters.

Method	Test image (figure number)							
	6.1(a)	6.1(b)	6.1(c)	6.1(d)	6.2(a)	6.2(b)	6.2(c)	6.2(d)
Eigenfilter	12.1%	56.1%	6.9%	37.1%	4.2%	4.0%	3.7%	4.7%
Opt. repr. Gabor filter bank	7.2%	37.2%	18.6%	41.9%	4.8%	5.5%	5.8%	36.2%
Prediction error filter	14.7%	34.7%	2.0%	44.6%	0.6%	0.9%	4.6%	31.7%
Optimal Gabor filter $\sigma = 2$	N/A	N/A	N/A	N/A	5.0%	10.4%	3.6%	10.8%
Optimal Gabor filter $\sigma = 4$	N/A	N/A	N/A	N/A	4.7%	5.3%	3.3%	11.6%
Optimal Gabor filter $\sigma = 8$	N/A	N/A	N/A	N/A	8.2%	5.7%	5.2%	11.4%
Optimal Gabor filter $\sigma = 16$	N/A	N/A	N/A	N/A	13.3%	14.0%	13.7%	24.2%
4-filter optimal Gabor filter bank	24.7%	54.0%	24.3%	52.8%	7.0%	4.3%	3.8%	18.7%
6-filter optimal Gabor filter bank	16.3%	49.4%	11.2%	50.3%	6.7%	3.8%	3.8%	17.1%
10-filter optimal Gabor filter bank	16.1%	40.7%	10.0%	39.9%	6.7%	4.1%	3.9%	12.8%
J_{MS}	16.9%	47.3%	10.0%	59.7%	0.9%	0.7%	8.3%	28.6%
J_U	12.7%	45.6%	4.0%	35.9%	0.9%	0.7%	2.1%	5.1%
J_F	N/A	N/A	N/A	N/A	0.9%	0.7%	2.1%	5.1%
BackProp. NN, mask size 11	47.4%	61.1%	74.1%	69.3%	45.3%	32.1%	30.0%	43.6%
BackProp. NN, mask size 21	72.4%	84.9%	74.7%	76.5%	48.5%	30.8%	28.8%	48.6%

Table 6.6: Classification errors for different optimized texture feature extractors. See Table 6.2 for an overview of the method descriptions.



(a)



(b)

Figure 6.4: (a) Texture pair D8-D84 [14] and (b) supervised segmentation by the back propagation designed mask [51].

The *eigenfilter* approach is never emphasized with either remarkably good or remarkably bad results. However, it must be remembered that the number of filters is relatively high with this approach. For two-texture images, the number of filters is typically in the range 14-18, while for multi-texture images it is 40 or close to 40. Hence, the argument of low complexity versus the heuristic filters is not applicable.

The *optimal representation Gabor* filter bank was the only optimized filtering approach doing a really good job on the image of Figure 6.1(a). However, for most of the other test images, it is only giving similar, and even in some cases significantly worse results than the best optimal filtering approaches, despite higher feature dimensionality. The poor performances for the images of Figures 6.1(c) and 6.2(d) are particularly noted.

original article [51], provided by Kalle Karu. The results should therefore be representative. The software for the modified scheme was not available.

The best results for the images in Figure 6.1(b), 6.1(c), and 6.2(a) in this study were obtained with the *prediction error filters*. However, despite these nice results, a remarkably poor result was obtained with the image of Figure 6.2(d) and a relatively poor result with the image of Figure 6.1(a). Hence, this approach is not very robust. The cause for the problem is most likely that the approach is optimal with respect to representation, not separation, as discussed in Section 2.3.1.

The results from the *optimal Gabor filters* and *optimal Gabor filter banks* are generally good, but the results indicate problems with similar textures – especially when using only one or a few filters. In particular, the results for the images of Figures 6.1(a), 6.1(c), 6.2(a), and 6.2(d) are noted. On the other hand, these approaches yield one-dimensional feature spaces for two textures. Hence, class labels may be assigned to the feature values by thresholding. Thresholding has a very low computational complexity compared to classifiers like the LVQ. Furthermore, training the threshold classifier means determining *one* scalar parameter per feature image. This is clearly a simpler task than training the parameters of, say, the LVQ classifier.

FIR filters optimized with respect to J_{MS} show remarkably poor performance for some images, in particular the texture pair of Figure 6.2(d) is noted. Furthermore, the results are unacceptable for the images in Figures 6.1(a), 6.1(b), 6.1(d), and 6.2(c). From this, we may conclude that these filters are not very robust. The number of resulting filters for the images in Figure 6.1 are 4, 46, 3, and 12, respectively.

Considerably better results are obtained by filters optimized with respect to J_U and J_F. For two-texture images, these filters are clearly the winners of this study. Overall, they outperform all other approaches, irrespective of number of features, for texture-pair images. Since these approaches yield only one feature image that may be classified by thresholding, they are also the least computational demanding for two-texture images.

Even for multi-texture images, filters optimized with respect to J_U are among the best optimal filters, but the performance is not of the same extra-ordinary quality. However, performance issues should be kept in mind. For the multi-texture images of Figure 6.1, there are 4, 25, 5, and 15 filters in the filter bank, respectively. Classification is done by thresholding each feature image once, thus the classification complexity is low. No multi-texture approach has yet been developed for the J_F criterion.

6.2.4 Discussion of the test images

Clearly, some images yield more differences between the methods than others. Specifically, this applies for the images in Figures 6.1(a), 6.1(c), 6.2(b), and 6.2(d). Apparently, the textures in these images are difficult to discriminate between, and some of the least robust or powerful methods experience problems.

6.3 Summary

We have seen how various filtering approaches yield different results for different images. No single approach did perform best or very close to the best for *all* images, thus no single approach may be selected as the clear “winner” of this study. In addition to the classification error issue is the issue of computational complexity. If we take the classification error and computational complexity into consideration, the following conclusions may be drawn:

- Much of the focus in the filter *optimization* approaches is on a low feature count, thus many of the optimalization schemed yield nice computational characteristics. In most cases, some of the optimal techniques were best or very close to the best techniques. Especially the filters optimized with respect to J_U or J_F had low classification errors. Hence, these filters should be selected for many applications. However, some problems with multi-texture images can be observed.

- For multi-texture images, the optimal prediction error filter approach may be a good candidate. It yields low feature extraction complexity and very low classification complexity. However, there is a risk that some textures are not discriminable by the approach.

- If the performances of the optimal filters or filter banks are too poor for a specific application, several heuristic filter banks should be tested. The results are not very conclusive on which approach to select, but both QMF and wavelet frame filter banks should be considered. However, compared to the optimal approaches, this implies increased computational complexity.

- The classification error increases slightly when using critical sampling, as opposed to overcomplete representations for the wavelet and the QMF approaches. However, the computational complexity decreases formidably. A critically sampled representation may therefore be selected before an overcomplete, if the increased classification error can be tolerated.

- The dyadic decomposition (also known as wavelet transform decomposition or octave band decomposition) may yield features with good separating properties. However, increasing the depth (i.e., adding more sub-bands) do not improve the classification results. Hence, we may conclude that important texture information is located in other sub-bands than the low-pass band, supporting the findings of Chang and Kuo [16].

- The computational complexities of the filtering approaches are generally low compared to the two other approaches tested here. Since the results were equally good or better for the filtering approaches, the filtering approaches should therefore be preferred.