Texture Segmentation Using Gabor Filters

Khaled Hammouda
Prof. Ed Jernigan
University of Waterloo, Ontario, Canada

Abstract **–Texture segmentation is the process of partitioning an image into regions based on their texture. Inspired by the multi-channel operation of the Human Visual System for interpreting texture, research has been focused on using a multi-channel approach based on Gabor filtering to mimic the operation of HVS for identifying different texture regions. In this paper we employ this multi-channel approach to the problem in order to gain insight into the ability of this methodology in solving the texture segmentation problem.**

Index Terms—Texture segmentation, multi-channel filtering, Gabor filters.

I. INTRODUCTION

TEXTURE is an important property of surfaces which characterizes the nature of the surface. An important task in image processing and machine vision is the task of segmenting regions of different texture in an image. However, it is not precisely defined what constitutes a "proper" region. Ideally, we would want each region to represent different "object" in the image, for the purpose of object recognition for example, or scene analysis. Still we cannot define what constitutes an "object" exactly. For example, if a bookshelf is filled with books, do we want to consider each book to be a separate object, or do we want the bookshelf and everything on it to be considered as one object? It is clear, then, that there is no one segmentation of an image that can be considered to be "right." The "right" segmentation exists only in the mind of the observer, which can change not only between observers, but also within the same observer at different times.

To be able to solve the problem of texture segmentation we have to define what is texture. Unfortunately, there exists on one unified definition of texture. However, a typical definition of texture in the literature is "a spatial arrangement of local (gray-level) intensity attributes which are correlated (in some way) within areas of the visual scene corresponding to surface regions. [4]" Another definition is "one or more basic local patterns that are repeated in a periodic manner." So it could be agreed on that texture exhibits some sort of periodicity of basic patterns. This fact leads us to the idea that people use texture properties to identify different textures. Rao and Lohse [1] indicate that people are sensitive to three texture properties: repetition, directionality and complexity [2]. This fact is very important if we would like to design a methodology for texture segmentation. Further investigation of how the Human Visual System (HVS) works on interpreting texture resulted in a robust approach for the texture segmentation problem.

According to the above investigation, a model for the HVS interpretation of texture has been based on multi-channel filtering of narrow bands. Simple cells in the visual cortex were found to be sensitive to different channels of combinations of various spatial frequencies and orientations. Since texture repetition can be characterized by its spatial frequency, and directionality by its orientation, then we can fit the HVS model into a methodology that uses multi-channel filtering at different spatial-frequencies and orientation for texture analysis.

The multi-channel filtering approach is actually a multi-resolution decomposition process, which is similar to wavelet analysis. In fact, one well known class of functions that are known to achieve both spatial and spatial-frequency local-

K. M. Hammouda,
Department of Systems Design Engineering,
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

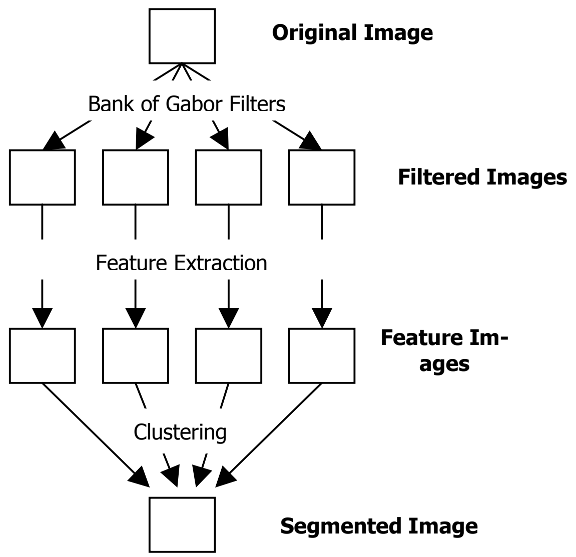

Figure 1. Texture segmentation process

ization is the Gabor function. This wavelet has been used extensively in texture segmentation due to the ability to tune a Gabor filter to specific spatial frequency and orientation, and achieve both localization in the spatial and the spatial-frequency domains.

The process of texture segmentation using Gabor filters involves proper design of a filter bank tuned to different spatial-frequencies and orientations to cover the spatial-frequency space; decomposing the image into a number of filtered images; extraction of features from the filtered images; and the clustering of pixels in the feature space to produce the segmented image. Figure 1 shows a schematic of this process. Both supervised and unsupervised approaches were used in texture segmentation. Supervised approaches rely on training methods and reference segmentations for performance assessment, while unsupervised approaches mainly rely on subjective assessment.

In this paper we design and implement a texture segmentation system based on the work done by Jain and Farrokhnia [5], and some recommendations given by Clausi and Jernigan [2]. Segmentation is done in unsupervised mode. The rest of this paper is organized as follows. Section 2 introduces Gabor filter characterization and design. Section 3 discusses the segmentation process in detail. We discuss the results in section 4. Finally, section 5 provides a summary and some concluding remarks.

II. Gabor Filter Characterization

As mentioned above, Gabor filters have the ability to perform multi-resolution decomposition due to its localization both in spatial and spatial-frequency domain. Texture segmentation requires simultaneous measurements in both the spatial and the spatial-frequency domains. Filters with smaller bandwidths in the spatial-frequency domain are more desirable because they allow us to make finer distinctions among different textures. On the other hand, accurate localization of texture boundaries requires filters that are localized in the spatial domain. However, normally the effective width of a filter in the spatial domain and its bandwidth in the spatial-frequency domain are inversely related according the uncertainty principle. That is why Gabor filters are well suited for this kind of problem.

A Gabor function in the spatial domain is a sinusoidal modulated Gaussian. For a 2-D Gaussian curve with a spread of σ_x and σ_y in the x and y directions, respectively, and a modulating frequency of u_0, the real impulse response of the filter is given by

$$h(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right]\right\}\cdot\cos(2\pi u_0 x) \tag{1}$$

The impulse response function is shown graphically in Figure 3.

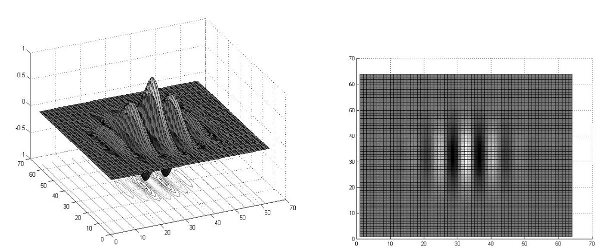

Figure 3. Gabor function in the spatial domain

In the spatial-frequency domain, the Gabor filter becomes two shifted Gaussians at the location of the modulating frequency. The equation of the 2-D frequency response of the filter is given by

$$\begin{aligned} H(u,v) = {} & \exp\left\{-2\pi^2[\sigma_x^2(u-u_0)+\sigma_y^2v^2\right\} \\ & +\exp\left\{-2\pi^2[\sigma_x^2(u+u_0)+\sigma_y^2v^2\right\} \end{aligned} \tag{2}$$

Figure 4 shows the frequency response graphically.

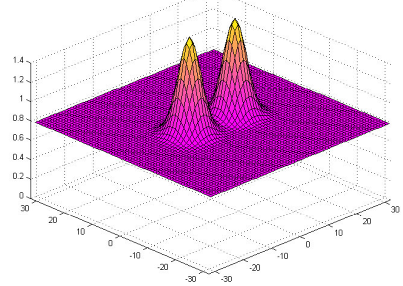

Figure 4. Gabor function in the spatial-frequency domain

The above equations show only an orientation of zero degrees with respect to the *x*-axis. An arbitrary rotation of the filter can be achieved spatially by rotating the spatial function in the spatial domain in the *x*-*y* plane, and in the spatial-frequency domain by rotating the frequency response function in the *u*-*v* plane.

The frequency u_0 and the rotation angle θ define the center location of the filter. By tuning u_0 and θ to different center locations we can create multiple filters that cover the spatial-frequency domain. However, another important aspect of the filter is the frequency bandwidth B_F and the orientation bandwidth B_θ. These can be set to constant values that match psychovisual data [2]. In particular a frequency bandwidth of 1 octave is found to perform well. (The frequency bandwidth, in octaves, from frequency f_1 to frequency f_2, is given by $\log_2(f_2/f_1)$.) It has to be noted that the frequency bandwidth increases with frequency in a logarithmic fashion. An orientation bandwidth of 30° is recommended by Clausi and Jernigan [2]. More details on filter parameter specification for designing the filter bank set is discussed in section 3.A. In the next section the steps for performing the texture segmentation process are discussed.

III. TEXTURE SEGMENTATION

As outlined in section 1, the process of texture segmentation using multi-channel filtering involves the following steps:

- Filter bank design,
- Decomposition of the input image using the filter bank,
- Feature extraction, and
- Clustering of pixels in the feature space.

We now turn to each one of these steps in detail.

A. *Filter Bank Design*

Specification of the filter set properties involves choosing a set of frequencies and orientations that will cover the spatial-frequency space, and capture texture information as much as possible. In this paper we test two orientation separation angles, one at 30° as recommended in [2], and the other is 45° as proposed in [5]. It is expected that the former cases should produce better results since we are doing finer quantization of orientation, which should capture more texture features than the latter case; however, it is more computationally expensive. In both cases, the frequencies used for the filters are

$$1\sqrt{2},\ 2\sqrt{2},\ 4\sqrt{2},\ \ldots,\ \text{and } (N_c/4)\sqrt{2} \ \text{ cycles/image width} \tag{3}$$

For the case of orientation separation of 30°, the number of filters required is $6\log_2(N_c/2)$ (for an image of width 256 pixels, 42 filters are required), and in the case of orientation separation of 45°, the number of filters required is $4\log_2(N_c/2)$ (for an image of width 256 pixels, 28 filters are required). For efficiency considerations, we do not use the filters at frequencies $1\sqrt{2}$ and $2\sqrt{2}$, since they are not very useful, because these filters capture spatial variations that are too large to explain textural variations in an image [5]. Also, the DC gain of the filters is set to zero in order to prevent any response to constant areas in the image.

The above proposed set of filter parameters were particularly selected so that it can properly capture texture information. Center frequencies of channel filters must lie close to the characteristic texture frequencies or else the filter responses will fall off rapidly [2]. Also care has to be taken so that the filters do not overlap in the frequency domain to avoid aliasing. Figure 5 shows the filter bank arrangement in the spatial-frequency domain. (Only half peak support regions are shown.)

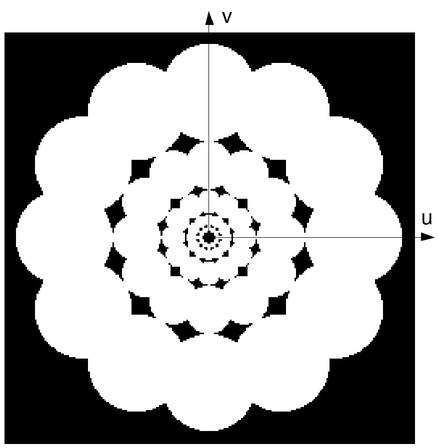

Figure 5. Gabor filter set in the spatial-frequency domain
Frequencies are one octave apart; Frequency bandwidths are one octave each; Orientation separation is 30°.

B. Feature Extraction of Filter Outputs

Filter outputs by default are not appropriate for identifying key texture features. A number of feature extraction methods were proposed to extract useful information from the filter outputs. Clausi and Jernigan reviewed some feature extraction methods. Some of the feature extraction methods include:

- using the magnitude response,
- applying spatial smoothing,
- using only the real component,
- using a non-linear sigmoidal function,
- using pixel adjacency information.
- applying full wave rectification,
- creating moments based on the spatial-frequency plane, and

We adopt the first five methods for the purpose of this paper. Spatial smoothing actually can be applied to any of the above mentioned methods, and is known to enhance the performance of the segmentation process because it suppresses large variations in the feature map in areas which belong to the same texture. However, too much smoothing can have a negative effect on the localization of texture region edges. Each filter output is smoothed using a Gaussian smoothing function that matches the corresponding filter spatial Gaussian curve. The smoothing Gaussian is selected to be wider than the matched Gabor modulated Gaussian [5].

Jain and Farrokhnia suggested using a non-linear sigmoidal function, which saturates the output of the filters. Their explanation was that it acts as a blob detector, and most textures can be characterized by blobs of different sizes and orientations. They also suggested using pixel adjacency information as extra features due to the fact that pixels belonging to the same texture are close to each other, so they should be clustered together. However, this method will not perform well if there are some similar texture regions that are not adjacent in the image.

Figure 6 shows a five-texture image taken from the Brodatz texture album [6]. Figure 7 shows a comparison between the different feature extraction methods applied to this image.

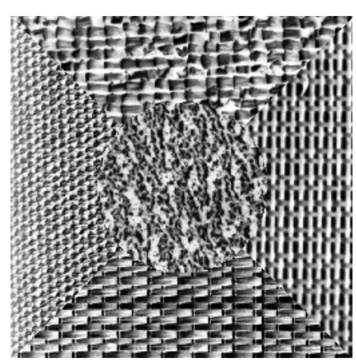

Figure 6. Five texture Brodatz image

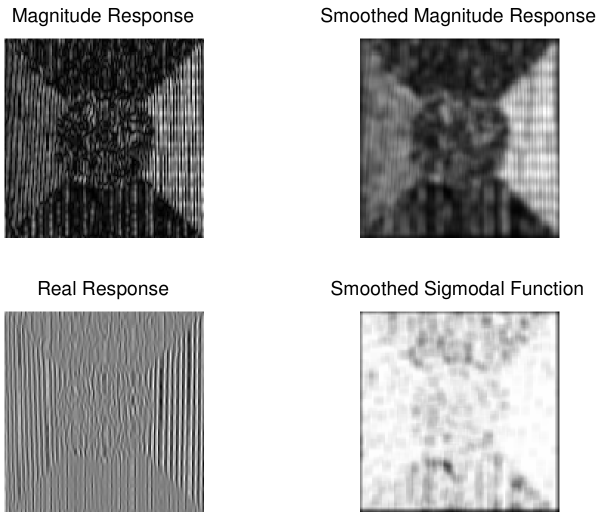

Figure 7. Different feature extraction methods
All the images are extracted from a filter output at frequency $6\sqrt{2}$ cycles/image width, and orientation of 0°

From the figure it could be subjectively argued that the smoothed magnitude response should perform better than the other methods because it captures most of the right side texture information as opposed to the others that exhibit no real distinctness in that part of the image.

C. *Clustering in the Feature Space*

At the end of the feature extraction step we are left with a set of feature images extracted from the filter outputs. Pixels that belong to the same texture region have the same texture characteristics, and should be close to each other in the feature space. The final step in unsupervised texture segmentation is to cluster the pixels into a number of clusters representing the original texture regions. Labeling each cluster yields the segmented image.

Different approaches were taken for the clustering process. Jain and Farrokhnia use a clustering algorithm known as CLUSTER, which is based on *K*-means clustering. Clausi uses another algorithm known as *K*-means Iterative Fisher (KIF). In both cases the number of clusters is determined automatically. In this paper we use the basic *K*-means clustering algorithm for simplicity. However, this means we have to provide the algorithm with the number of clusters beforehand, which means the number of different textures in the image is known, or could be estimated.

K-means starts by assigning the cluster centers to random points in the input set. Then it starts calculating the distance to from each point to the cluster centers and assigns each point to its nearest cluster center (based on the Euclidean distance.) The next step is to recalculate the cluster centers as the mean of each cluster. The algorithm works iteratively by assigning the points to their nearest cluster center and updating the cluster centers until it converges and no more changes could be made.

When clustering is done, each pixel is labeled with its respective cluster, finally producing the segmented image.

IV. EXPERIMENTAL RESULTS

The multi-channel approach mentioned above was implemented and tested against a number of textured images from the Brodatz album due to their popularity in this field.

Figure 8 shows the segmentation result of the Nat-5 five-texture image from the Brodatz album. The figure shows segmentation based on orientation separation of filters at 45° and 30°, respectively. It is clear from the result that the filter set at 30° separation is performing much better than the other set as was mentioned earlier. This was expected since at 30° orientations we can cover more of the spatial-frequency space, which results in capturing more texture information. Obviously the set at 45° orientations was confused between the upper texture region and the lower one that they got classified as belonging to the same texture. The filter set at 30° was able to detect this slight change and classified each region separately.

We can also see that the texture boundaries are well localized to some extent. However, we do not get sharp localization due to the smoothing that is done as a post-processing step to the magnitude of filter outputs. To see if smoothing is really producing a better result, we tested the segmentation of the same image without smoothing. The result is shown in Figure 9.

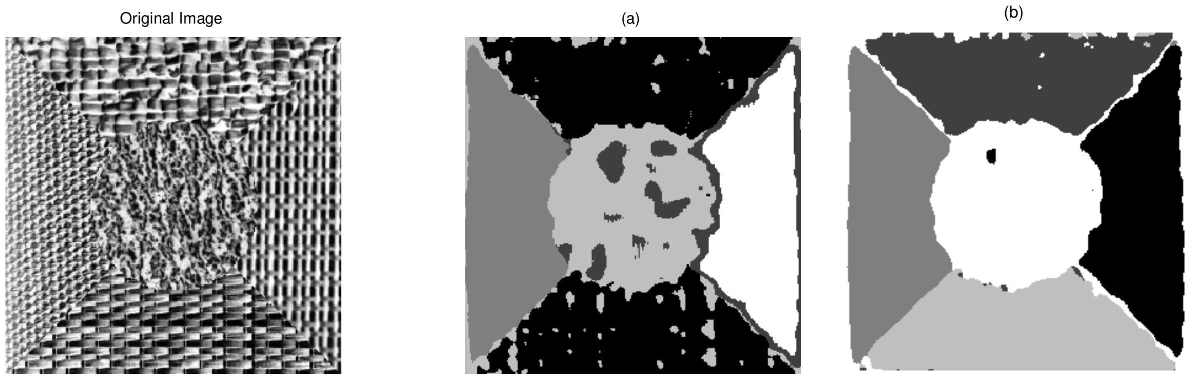

Figure 8. Segmentation of the five-texture Nat-5 image
Using the magnitude response for feature extraction
(a) using 45° orientation separation; and (b) using 30° orientation separation

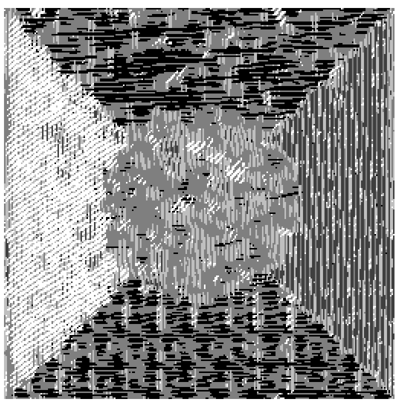

Figure 9. Effect of non-smoothed segmentation

The effect is clear and shows that smoothing suppresses the variations in the texture features within the same texture. Non-smoothed segmentation is severely affected by this variation and the result suffers from non-contiguous labeled regions as in the smoothed case.

To better understand the localization of texture edges, a pure sinusoidal texture image was synthesized containing four sinusoids with different frequencies and orientations carefully selected to match the filter set frequency locations and orientations. The result is shown in Figure 10. The texture regions were recovered almost perfectly, thus justifying the system design, and leading us to the fact that Gabor filters indeed have joint spatial/spatial-frequency localization ability.

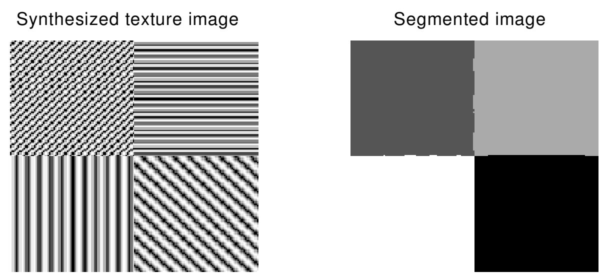

Figure 10. Segmentation of pure sinusoid texture

Segmentation based on different feature extraction methods was tested. The methods tested are:

- Magnitude response,
- Real response,
- Sigmoidal function, and
- Pixel adjacency information.

The results of this test is shown in Figure 11. As was recommended by Clausi and Jernigan, the magnitude response always produced better results. When combined with pixel adjacency information as proposed by Jain and Farrokhnia, we got almost perfect segmentation of the four-texture image shown in the figure. Real response seems to produce noisy segmentation due to the inability of the real component of the response to represent the full response of the filter. The sigmodial function produced an alias segmentation for the first and fourth quadrant of the image, due to blob-like similarity between these quadrants.

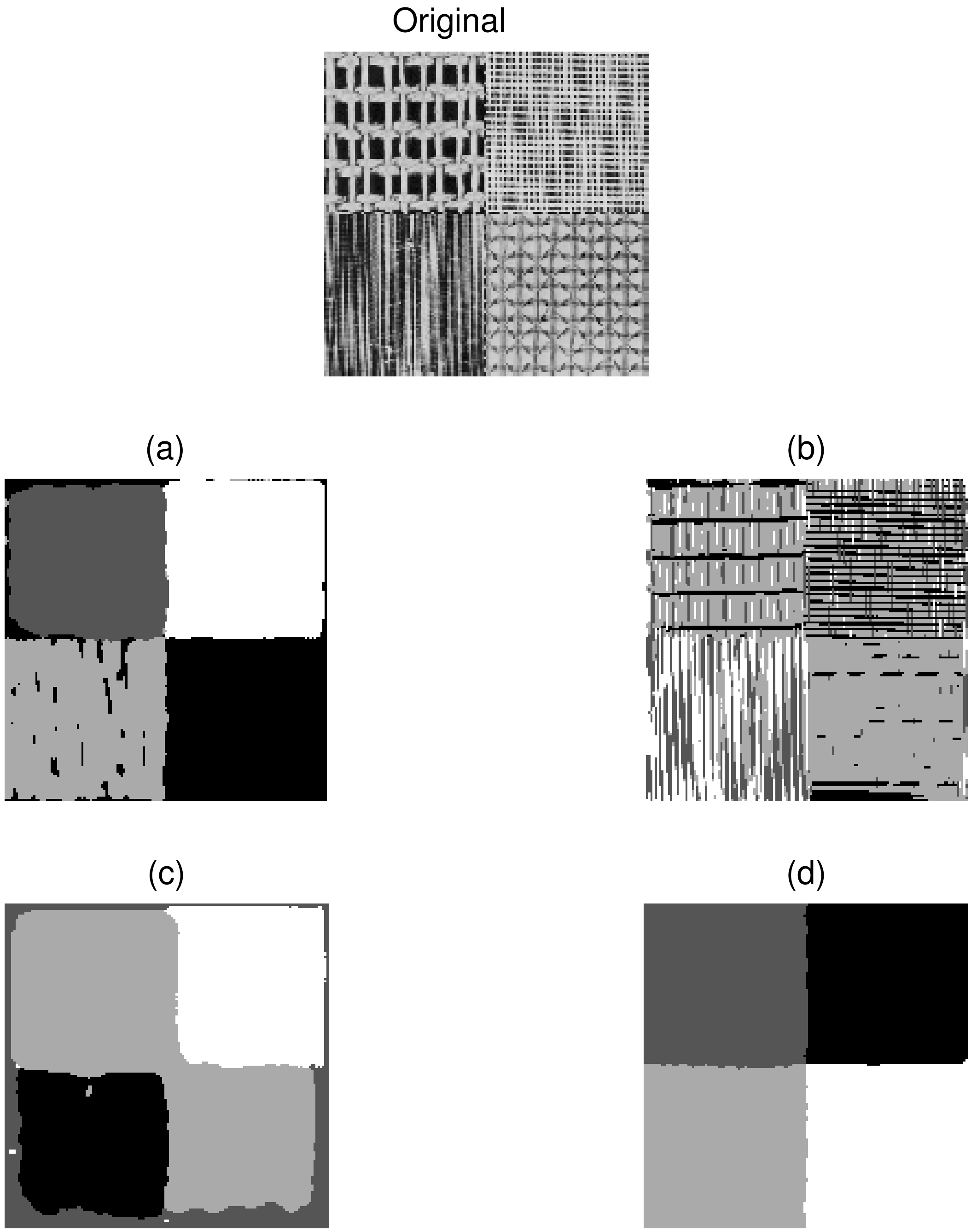

Figure 11. Segmentation based on different feature extraction methods
(a) using magnitude response; (b) using real response;
(c) using sigmoidal function; (d) using magnitude response and pixel adjacency information

V. SUMMARY AND CONCLUSION

In this paper we demonstrated the multi-channel approach to the texture segmentation problem. Based on HVS operation, the multi-channel filtering approach attempts to tune multiple filters at different spatial-frequencies and orientations to capture key texture information through separate channels. A well known class of filter that have join spatial and spatial-frequency localization are Gabor filters. By carefully designing a Gabor filter bank covering the spatial-frequency domain we can decompose an image into multi-resolutions that correspond to different texture characteristics. Two different orientation separation of filters were tested, 30 and 45 degrees. At 45 degree separation, the set of filters were able to capture some texture information, but was not subjectively good enough. At 30 degree separation on the other hand, the set of filters seemed to capture most of the texture characteristics in the image and produced more accurate results. Moreover, at 30 degree orientation separation, the set of filters are suited for general texture segmentation problems that exhibit wide varying texture characteristics.

Extraction of texture information from filter responses can be done in several ways. In this paper some of the feature extraction methods were applied and tested. Among those, the magnitude response of the filters seemed to produce the most correct partitioning of the image. Combined with pixel adjacency information as additional features enhanced the segmentation process significantly. Another important post-filtering operation is smoothing of the filter outputs. A Gaussian smoother that matches the Gaussian of the filter is used to smooth the output image. Smoothing resulted in suppressing the variations within the same texture region in the output image, thus enhancing the segmentation process significantly. However, care has to be taken not to over-smooth the outputs so that correct localization of the texture edges can be achieved.

Unsupervised texture segmentation based on multi-channel filtering seems to be a natural approach to the problem. Today, the need for a multi-resolutional approach to texture analysis is well recognized, and a lot of work is being done using wavelet analysis to tackle this problem. While other approaches to texture analysis have had to be extended to accommodate this paradigm, the multi-channel filtering approach, is inherently multi-resolutional, and lends itself to the similarity with the HVS operation for texture interpretation.

VI. REFERENCES

[1] A. R. Rao, G. L. Lohse, “Identifying high level features of texture perception,” *CVGIP: Graphical Models and Image Proc.* 55(3), pp. 5218-5233, 1993.

[2] D. Clausi, M. Ed Jernigan, “Designing Gabor filters for optimal texture separability,” Pattern Recognition, vol. 33, pp. 1835-1849, 2000.

[3] K. R. Castleman, “Digital Image Processing,” Prentice Hall, 1996.

[4] M. Clark, A. Bovik, W. Geisler, “Texture segmentation using a class of narrowband filters,” *IEEE CH2396-0*, 1987.

[5] A. K. Jain, F. Farrokhnia, “Unsupervised texture segmentation using Gabor filters,” *Pattern Recognition*, vol. 24, no. 12, pp. 1167-1186, 1991.

[6] P. Brodatz, “Textures: A Photographic Album for Artists and Desingers,” *Dover*, New York, 1966.

[7] O. Pichler, A. Teuner, B. Hosticka, “An unsupervised texture segmentation algorithm with feature space reduction and knowledge feedback,” *IEEE Trans. Image Proc*, vol. 7, no. 1, 1998.

[8] T. P. Weldon, W. E. Higgins, D. F. Dunn, “Gabor filter design for multiple texture segmentation.”

[9] N. W. Campbell, B. T. Thomas, “Automatic selection of Gabor filters for pixel classification,” University of Bristol, UK.

[10] D. A. Clausi, “Texture Segmentation of SAR Sea Ice Imagery,” PhD Thesis, University of Waterloo, Waterloo, Ontario, Canada, 1996.