SCALE CHARACTERISTICS OF LOCAL AUTOCOVARIANCES FOR TEXTURE SEGMENTATION

Annett Faber, Wolfgang Förstner
Institute of Photogrammetry, University Bonn, Nußallee 15, D-53115 Bonn, Germany, email: (annett,wf)@ipb.uni-bonn.de

KEY WORDS: texture representation, Laplacian pyramid, spectral decomposition, eigenvalues, urban structure.

ABSTRACT

This paper describes research on the extraction of urban structures from aerial images or high resolution satellite scenes like the German MOMS02 sensor. We aim at a separation of neighboring textured regions important for describing different urban structures. It has been shown, that grey level segmentation alone is not sufficient to solve this problem. If we want to use texture additionally, the development of a suitable representation of texture images is required. We use the scale characteristics of the local autocovariance function, called SCAF, of the possibly multiband image function. The final result of the process are texture edges.

1 INTRODUCTION

Texture is one of the most fundamental and at the same time most interesting characteristics of visible surfaces in the human perception process. Therefore, in pattern recognition the analysis of textures is very important. Numerous techniques for texture analysis have been proposed. They can be mainly categorized as (Haralick and Shapiro, 1992):

1. *Texture classification*: For a given textured region, decide, to which one out of a finite number of classes the region belongs.
2. *Texture synthesis*: For a given textured region, determine a description or a model.
3. *Texture segmentation*: In a given image, which contains many textured regions, determine the boundaries between these regions.

The procedures developed for the solution of these problems can be subdivided into three categories: structural approaches, statistical approaches and filterbased approaches (J.-P. de Beuaville and Langlais, 1994, Reed and du Buf, 1993, Shao and Förstner, 1994).

Structural approaches assume that textures contain detectable primitive elements which generate the texture in a regular or irregular manner, following certain connection rules. Such approaches use e. g. so-called texture grammars (Carlucci, 1976) or texture elements (Julez and Bergen, 1983).

Due to the almost unlimited number of possible unit patterns and the complexity of the rules, these procedures have shown lower success in texture analysis than the following two approaches.

Statistical approaches use statistical characteristics, derivable from the images. These characteristics are often sufficient for texture classification and segmentation, without needing generation rules for the textures. They can further be separated into approaches which use

- descriptions derivable directly from the images such as variance, entropy or other values obtained from the local pixel neighborhood, as e. g. co-occurrences (Haralick, 1979) and
- model based descriptions, such as autoregressive models, Markov or Gibbs random fields (Andrey and Tarroux, 1996, Derin and Cole, 1986).

Statistical approaches usually refer to the image grid, thus have difficulties in handling scale space properties.

Filter based approaches assume that the image function can be described locally by its amplitude spectra. Gabor wavelets have been the first choice together with linear and nonlinear post processing steps to achieve multi scale features in the different channels (A. C. Bovik and Geisler, 1990, Bigün and du Buf, 1992, Malik and Perona, 1990). This class of approaches is motivated by their similarity to the human visual system.

The advantage is that the filter responses for basic geometrical transformations are predictable and the filters work equally for natural scenes of different texture types. However, there is no general approach for the selection of a suitable filter bank and for the linkage of different image channels.

2 MOTIVATION

We are interested in analyzing satellite and aerial images especially for extracting urban structures. Therefore, we need powerful techniques for image segmentation in the presence of natural textures. Grey level or color segmentation results often suffer from over or under segmentation. Texture segmentation can reduce the amount of over segmentation, mostly without the risk of under-segmentation.

Our first experiences with Gabor wavelets based on (Malik and Perona, 1990) demonstrated the feasibility of a filter based approach for texture edge extraction (Shao and Förstner, 1994). Our approach used the edge detection scheme in our feature extraction program FEX (Fuchs, 1998), cf. Figure 1c). It exploits its ability to handle multichannel images by taking the filter responses as a multichannel image as input.

Due to the heavy algorithmic complexity, we develop a new filter based scheme for deriving texture edges again using the edge extraction scheme of our feature extraction

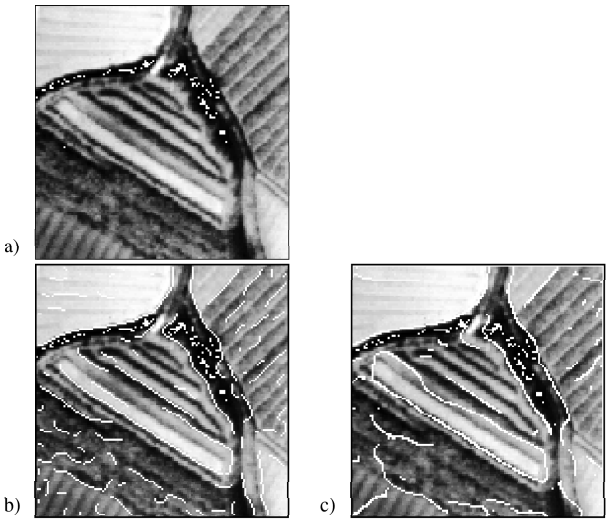

Figure 1: *a) Aerial image b) Edge extraction result. Grey level edges obtained with FEX. Observe the large number of small edges in the textured areas. c) Texture edges from (Shao and Förstner, 1994).*

program *FEX* but with a newly developed filter bank. The resulting multichannel image represents the Scale Characteristics of the local Autocovariance Function (SCAF) (cf. Fig. 3). Thus, we apply FEX on a suitable representation of texture instead of a grey level or color image for obtaining texture edges.

We do not aim at a a complete representation of the textures. However, we achieve a separation of neighboring textured areas, that is sufficiently good for the interpretation of aerial images.

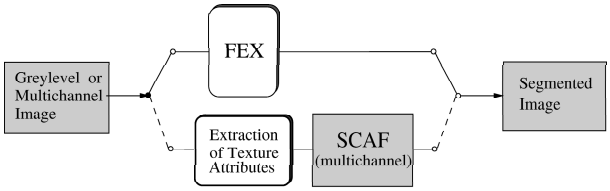

Figure 2: *Extension of FEX by the scale characteristics of the local autocovariance function (SCAF).*

3 OVERVIEW

We want to give an overview of the individual steps of our texture edge extraction scheme (cf. Fig.3).

First, we derive the scale characteristics of the local autocovariance function, (cf. section 4). This is the basic step of our approach. These characteristics are derived in two steps:

- the strength, direction and anisotropy of the texture are derived from the square gradient of the image function (4.2). They characterize the form of the local autocorrelation function. This way we obtain three features of the image texture at the highest resolution.
- the spatial frequency of these features is then determined using a Laplacian pyramid (4.3).

After a normalization step, we combine the pyramid levels for all three texture features in a multichannel image (section 4.4).

In the second step, we extract the texture edge (section5) using the multichannel scheme known from our feature extraction.

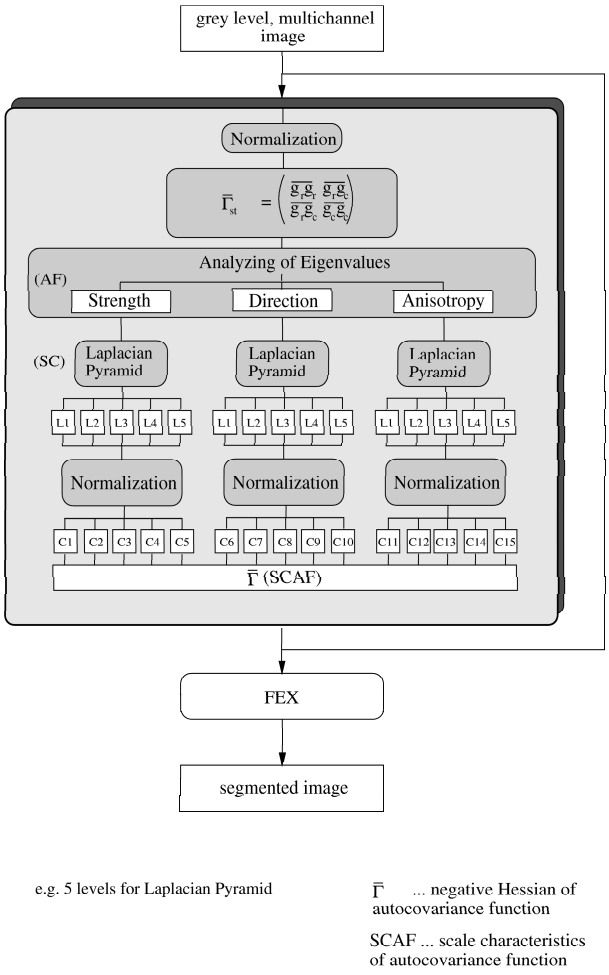

Figure 3: *Process to obtain SCAF (scale characteristics of the local autocovariance function) and the texture segmentation from a grey level or multichannel image using FEX.*

4 SCALE CHARACTERISTICS OF LOCAL AUTOCOVARIANCE FUNCTION

This section explains in detail the steps for derivation of the scale characteristics of the local autocovariance function.

4.1 Stochastic image model

To characterize the textures we have used the following image model.

Starting from a fully partitioned image ($I = \bigcup_{i=1}^{m} S_i$) into m segments S_i, we assume that the ideal image function within the segments is a weak stationary process $\mathbf{f}_i(r,c) \sim N(\mu_i, \mathbf{C_i})$, where $\mu_i = \mathbf{const}$ and $\mathbf{C_i}$ is the covariance matrix of the image pixels within the segments (r rows, c columns). We assume the covariance matrix to be representable by a p. d. covariance function $C(\Delta r, \Delta c)$, thus

fixing the second moments of the distribution. The covariance function can be assumed to be a decaying function characterized by its value at zero $C_{f_i}(0,0) = \sigma_{f_i}^2$ being the variance of $\mathbf{f}$ and the curvature $C''_{f_i}(0,0) = \mathbf{H}_{f_i}$ at zero being the Hessian of the covariance function.

The real image function $g(r,c) = f(r,c) + n(r,c)$ is composed of the true image signal $f(r,c)$ and the noise $n(r,c)$, which is assumed to be white, $n \sim N(0, \sigma_n^2)$.

The separation of the segments is based not only on their expected value μ_i, as in classical segmentation procedures, but also on their covariance structure.

For the characterization of the covariance function we use the negative Hessian $(-\mathbf{H}_g)$ of the autocovariance function of g. Due to the moment theorem (Papoulis, 1984), it is identical to the covariance matrix $\overline{\mathbf{\Gamma}g}$ of the gradient $\nabla_s g$ of the image function g over a window G_t. The filter kernels $h(x)$ of the gradient $\nabla_s g$ and the window G_t have width s and t (Fuchs, 1998). For the estimation of the negative Hessian of the available image function g we thus obtain:

$$\overline{\mathbf{\Gamma}_{st}} = G_t * (\nabla_s g \, \nabla_s g^T) = -\mathbf{H}_g \qquad (1)$$

$$= \begin{pmatrix} \hat{\sigma}^2_{g_r} & \hat{\sigma}_{g_r g_c} \\ \hat{\sigma}_{g_r g_c} & \hat{\sigma}^2_{g_c} \end{pmatrix}$$

$$= \begin{pmatrix} \overline{g_r^2} & \overline{g_r g_c} \\ \overline{g_r g_c} & \overline{g_c^2} \end{pmatrix}$$

We need to specify two scale parameters:

1. Differentiation scale s: The image resolution in terms of the sharpness of the edges defines the scale for the differentiation kernel. E. g. the 3×3-Sobel kernels correspond to the 3×3-Binomial kernel having scale $s = 1/\sqrt{2}$.
2. Integration kernel t: The width of the expected edges motivates the window size of the integration kernel. Observe, that the edge extraction does not distinguish between step and bar edges, thus also the expected width of bar edges can be used for specifying the integration kernel t.

4.2 Texture parameters

The core of the representation is the characterization of the local autocorrelation function, represented by the squared gradient $\overline{\Gamma g}$ of the autocovariance function. By analyzing the eigenvalues λ_k of the matrix $\overline{\Gamma g}$, we obtain three *local* features. These are (Förstner, 1991):

- the **strength** a (amplitude) of the texture. It represents the local variation of the intensity function and is measured using the variance of gradients, being the trace of the squared gradient: $\overline{\Gamma g}$:

$$a = \mathrm{tr}\ \overline{\Gamma g} = \overline{g_r^2} + \overline{g_c^2} = \lambda_1 + \lambda_2$$

- the **direction** φ or the orientation of the texture. The squared gradient is large in the direction of large intensity variations, but small in directions with small intensity variations, which in case of directed textures is identical to the subjective perception of the orientation of the texture. Instead of determining Gabor filters with different orientations, thus sampling the radial variation of the gradient, we directly determine the direction of smallest variation, being the direction φ of the eigenvector of $\overline{\Gamma g}$ corresponding to the smallest eigenvalue:

$$\varphi = \frac{1}{2} \arctan \frac{2\overline{g_r g_c}}{\overline{g_r^2} - \overline{g_c^2}}$$

  The orientation lies in the range $0 \le \varphi < \pi$, thus at pure edges the information on the direction of the gradient is lost.

- the **anisotropy** q (quality of direction) of the texture. It indicates the angular variation of texture and edge characteristics. The anisotropy can be measured by the ratio v of eigenvalues λ_1 and λ_2 or, equivalently by the form factor:

$$q = \frac{4 \det \overline{\Gamma g}}{\mathrm{tr}^2\, \overline{\Gamma g}} = 1 - \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2$$

  Isotropic, i. e. non-oriented, texture is characterized by $q = 1$, whereas anisotropic, i. e. oriented, texture is characterized by $q \ll 1$.

These three characteristic texture features are quite informative as can be seen in Fig. 4 and Fig. 5[1]. In the figures the strength is coded from low values (white) to large values (black), the direction is coded from $\phi = -\pi/2$ (white) to $\phi = \pi/2$ (black) and the anisotropy is coded from $q = 0$ (white) to $q = 1$ (black).

As the squared gradient of the given image function g instead of the true image function f is used, we can expect the properties of the texture to be recoverable only in images with low noise. An information preserving filtering cf. (Förstner, 1991, Weidner, 1994) is useful as preprocessing step to reduce noise without smoothing the edges.

4.3 Laplacian pyramid

The texture parameters discussed above describe texture properties of a small image region, defined by the scale parameters s and t of the Gaussian windows in eq. (1).

However, textures contain interesting characteristics: they are hierarchically structured. The texture parameters vary within the local neighborhood. This gives rise to a scale space analysis, which determines the variation of the texture features depending on the size of the local neighborhood.

To use the complete information, contained in the image signal, we perform a spectral decomposition of our texture parameters.

To obtain a spatial scale decomposition, we use a Laplacian image pyramid (Burt and Adelson, 1983) applied to all three texture features. It separates the different spectral bands of the texture image features. The basis of the generation of an image $l_i(r,c)$ of the Laplacian pyramid is the difference formation between two subsequent levels i and $i-1$ of the *Gaussian pyramid* $g_i(r,c)$

$$l_i(r,c) = g_i(r,c) - g_{i-1}(r,c)$$

[1] The used image "Avenches" belongs to the data set in www.goed.ethz.ch/p02/projects/AMOBE/index.html

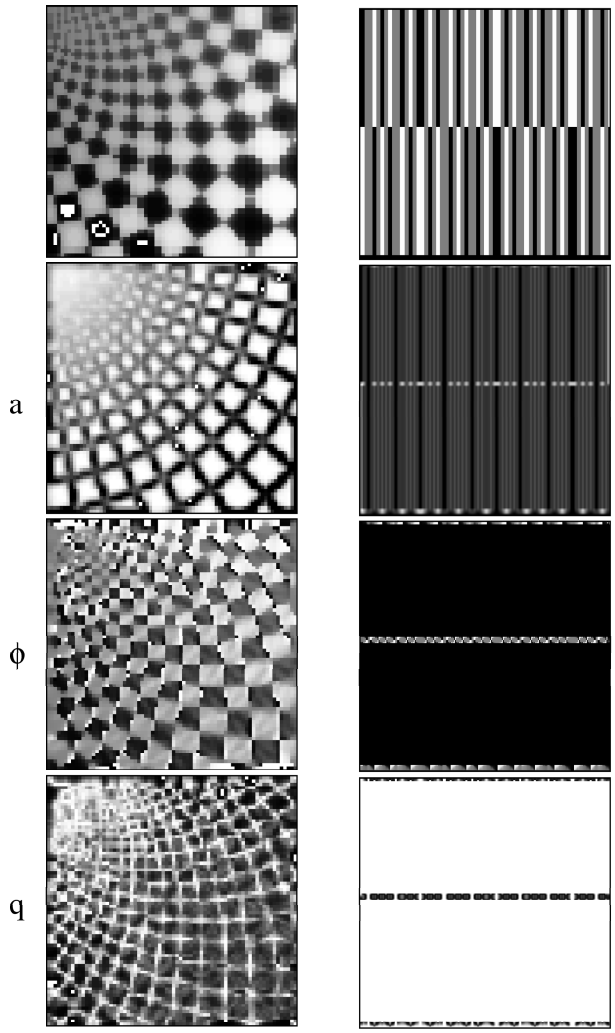

Figure 4: **Synthetical scenes:***From top to bottom: (1) synthetic images, left "Rosette", right "bar codes"; (2) strength a, (3) direction ϕ and (4) anisotropy q. All three texture measures computed using 3×3-Sobel as differentiation kernel, with $s = 0.7$, and integration kernel $t = 0.5$.*

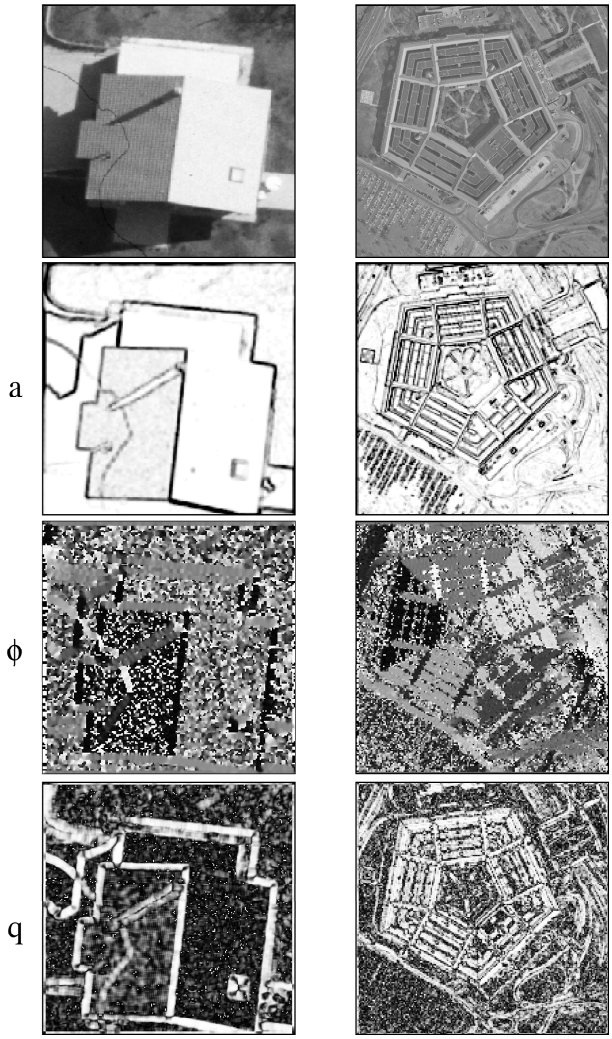

Figure 5: **Natural scenes:**.*From top to bottom: (1) synthetic images, left "Avenches", right "Pentagon"; (2) strength a, (3) direction ϕ and (4) anisotropy q. All three texture measures computed using 3×3-Sobel as differentiation kernel, with $s = 0.7$, and integration kernel $t = 0.5$*

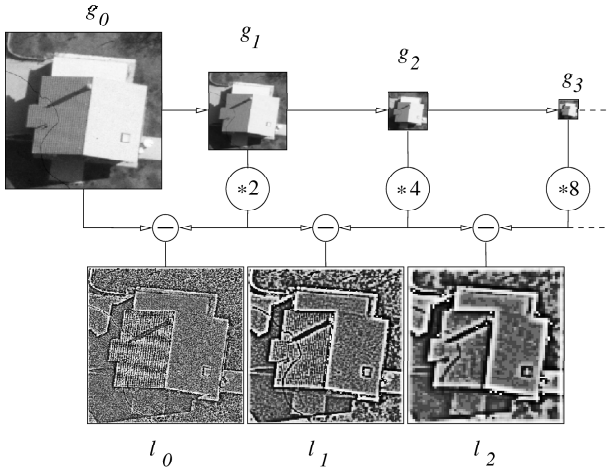

Figure 6: *Scheme for generating the Laplacian box. Upper row: generated Gaussian pyramid, lower row: generated Laplacian pyramid.*

The resampling factor k_i for the Gaussian pyramid levels i generated before is $k_i = 2^i$. Because we are interested in an uniform size of all generated levels of the Laplacian pyramid, our resampling factor differs to the factor used in (Burt and Adelson, 1983). Due to the uniform size of the generated images in the different levels, we call this stack *Laplacian box*.

4.4 Normalization for generation a multichannel image

In section 4.3 we discussed the use of the scale space in order to obtain a rich image description. Our next step is to specify this predicate. In the segmentation process we use all generated resolution levels in parallel. Combining all levels l_i $(0 \leq i < N)$ of the Laplacian box of all texture parameters, we get a multichannel image, thus we actually stack the three Laplacian boxes.

In order to be able to fuse the different channels of the Laplacian box, we need to normalize these channels.

For the normalization, we use the expected noise behavior of the filter kernels of the Laplace box, which we determine by analyzing the impulse response, based on the linearity of the generation process:

If a filter $h(r,c)$ is applied to an image $g(r,c)$ with white noise $n(r,c) \sim N(0,\sigma_n^2)$, the noise variance $\sigma_{g'}^2$ of the resulting image $g'(r,c) = h(r,c) * n(r,c)$ is given by:

$$\sigma_{g'}^2 = \sum_r \sum_c h^2(r,c)\, \sigma_n^2$$

Therefore the influence factor of the filter operation is the total of the squares of the filter coefficients. This corresponds to the proposal of (Ballard and Rao, 1994) who take the total energy of the filters. For our specific case, the analysis of the impulse response of the levels l_i of the Laplacian box, generated using the binomial mask B_4 (Jähne, 1989), we get the normalization factor f_i:

$$f_i = \frac{\sigma_{g_i'}^2}{\sigma_{n_i}^2} = \sum_{r,c} h_i^2(r,c)$$

The normalization factors only depend on the used filter mask. Fig. 7 shows all channels of the feature space for the texture edge extraction.

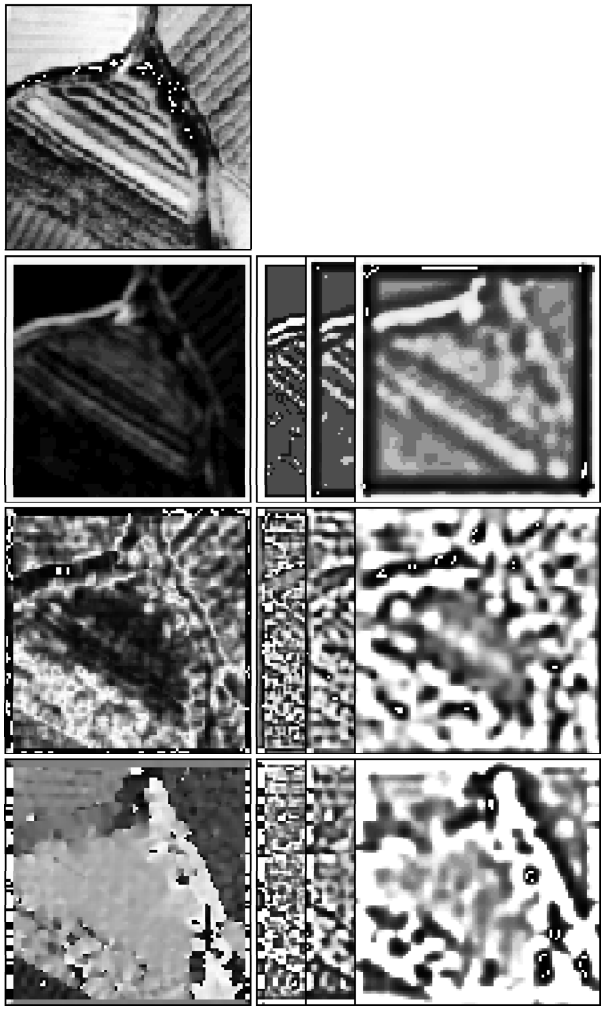

Figure 7: *Feature space for texture edge extraction. From top to down the aerial image with strength, anisotropy and direction of the texture respectively, and from left to right the levels of the Laplacian box for each feature.*

5 TEXTURE EDGE EXTRACTION

The final task is the extraction of texture edges.

5.1 Edge detection

We use the feature extraction program FEX to extract the texture edges. This program analyzes the local autocovariance function of a multichannel image g using the negative Hessian $\overline{\mathbf{\Gamma} g}$, in our specific case $\overline{\mathbf{\Gamma}(SCAF)}$. Using FEX for edge detection, results in texture edges. These edges separate neighboring textured areas depend on the user-selectable parameters of FEX (resolution scale, scale for lines and a significance level for internal statistical tests).

Alltogether, we need to specify five parameters:

1. The differentiation scale s_1, needed for determining the texture properties at the highest resolution.

2. The integration scale t_1, needed for determining the texture properties at the highest resolution.
3. The number n_l of the used pyramid levels.
4. The differentiation scale s_2, needed for determining the integrated squared gradient of the texture features.
5. The integration scale t_2, needed for determining the integrated squared gradient of the texture features.

5.2 Choice of scale

One of the main and up to now unsolved problem is the choice of suitable scales for texture edge extraction. This is a general problem and to our knowledge has not been solved satisfactory. The reason is that textures may appear at very different scales. Though these scales may be identified by some automatic means it is not clear whether these scales correspond to textures or macro structures which the analyzing module wants to resolve. E. g. in the case of the tile row 2 column 4 in Fig. 8 two interpretations are possible: Either the tile is part of a textured region, where the texture is quite regular and has long wave lengths, or the tile may be interpreted as a composition of some few homogeneous regions separated by lines.

That means, the user or the calling routine has to decide which levels of the Laplacian box should be used to obtain the required results. This strongly depends on the application. This may be a severe problem in case of textures of very different scale.

In our work this step was done interactively.

6 RESULTS

This section demonstrates the properties of the new texture edge extraction scheme.

In all cases we compare the result of the texture edge extraction with the grey level edge extraction from FEX.

The basic problem is the proper choice of the scale parameters.

For the first investigations presented below, we used the Sobel operators as differentiation kernel for the determination of the texture parameters, thus fixed $s_1 = 1/\sqrt{2}$.

6.1 Results from synthetic data

First, we present our results of the texture edge extraction for synthetic data. The scales were selected such that the feature extraction could rely on the intensity differences between the tiles. Obviously, this was quite successful in this case. The texture edges, however, are a bit cleaner.

6.2 Results from natural scenes

In this section, we show the potential of our algorithm for texture edge extraction of natural scenes. The results are shown in Fig. 9.

In order to show the difference between grey level and texture edge extraction, and not to obtain intensity edges alone, the integration scale t is chosen larger in the texture edge extraction scheme. This is reasonable, as we want to group several furrows into one field. Obviously, this reasoning leads to quite satisfying results. Due to our special implementation, not all texture edges at the image borders are captured.

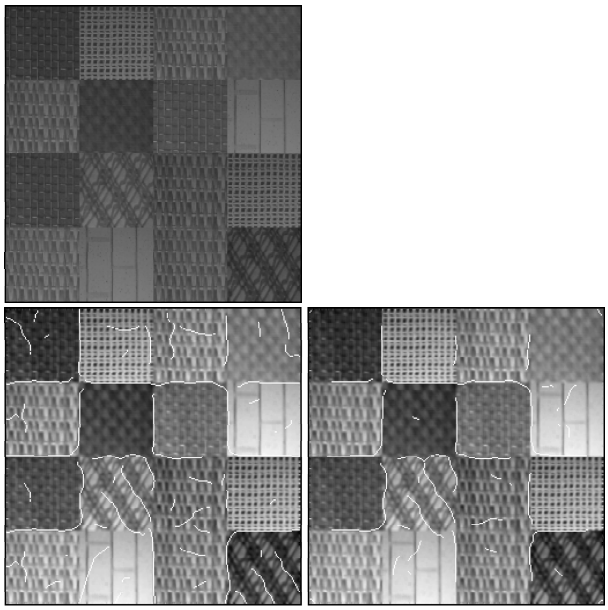

Figure 8: *Results of texture edge extraction: Upper row) collage of Brodatz-textures; lower row left) using intensity alone ($s = 5.0$, $t = 5.0$); right) using the Laplacian box ($s_1 = 0.7, t_1 = 6.0, n_l = 3, s_2 = 5.0, t_2 = 5.0$).*

6.3 Results from non-textured images

To achieve an improved method for image segmentation we have to ensure, that our technique provides good results not only for textured images, but also for non-textured images. Therefore, we applied our approach to some non-textured images too. The results are shown in Fig. 10. The result is satisfactory. The spurious texture edges in the background can be explained, as *no* thresholding is performed, in contrast to the procedure for grey level edge extraction.

7 SUMMARY AND CONCLUSION

This paper presented a filter based approach for texture edge extraction using the scale characteristics of the local autocovariance function.

The approach was implemented and tested on synthetic and natural scenes and shows some promising results. One of the main problems is the choice of suitable scales for texture edge extraction. In our experiments this step was done interactively.

To perform a qualitative evaluation, we have to compare our algorithm with other approaches, as shown (Shao and Förstner, 1994). Also more detailed aspects need to be analyzed, e. g. the effect of the higher levels of the Laplacian box on texture edge extraction and the ability to extract edges at boundaries between textured and non-textured regions.

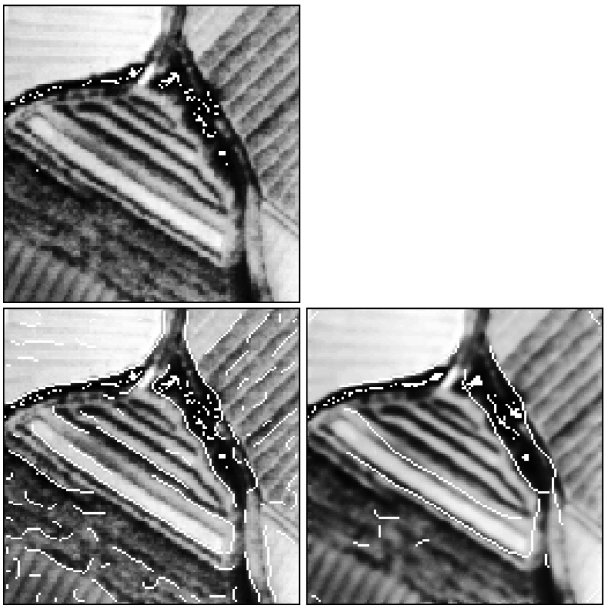

Figure 9: *Results of edge extraction: Upper row) aerial image, lower row left) using intensity alone ($s = 2.0, t = 2.0$), right) using the Laplacian box ($s_1 = 0.7, t_1 = 2.0, n_l = 3, s_2 = 4.0, t_2 = 3.0$).*

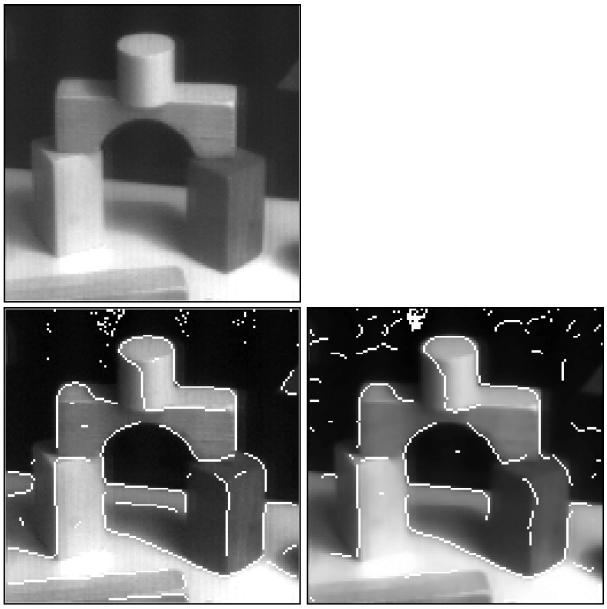

Figure 10: *Results of edge extraction: Upper row) image "toys", lower row left) using intensity alone ($s = 2.0, t = 2.0$), right) using level the Laplacian box ($s_1 = 0.7, t_1 = 2.0, n_l = 2, s_2 = 2.0, t_2 = 2.0$).*

REFERENCES

A. C. Bovik, M. C. and Geisler, W. S., 1990. Multichannel texture analysis using localized spatial filters. IEEE Transaction on PAMI 12, pp. 55–73.

Andrey, P. and Tarroux, P., 1996. Unsupervised texture segmentation using selectionist relaxation. Proc. of ECCV, pp. 483–491.

Ballard, D. H. and Rao, R. P. N., 1994. Seeing behind occlusions. In: Lecture Notes in Computer Scienes, Vol. 800, Springer Verlag, Heidelberg, pp. 274–285.

Bigün, J. and du Buf, J. M. H., 1992. Geometric image primitives by complex moments in Gabor space and the application to texture segmentation. In: Proceedings of Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Press, pp. 648–649.

Burt, P. J. and Adelson, E. H., 1983. The Laplacian pyramid as a compact image code. IEEE Transactions on Communications COM-31(4), pp. 532–540.

Carlucci, L. A., 1976. A formal system for texture langagues. Pattern Recognition 4(1), pp. 53–72.

de Beuaville, J.-P., Bi, D. and Langlais, L., 1994. Texture segmentation using grey level rank-vectors. Graphics and Vision 3(4), pp. 667–674.

Derin, H. and Cole, W. S., 1986. Segmentation of textured images using Gibbs random fields. Computer Vision, Graphics, and Image Processing 35 (1), pp. 72–98.

Förstner, W., 1991. Statistische Verfahren für die automatische Bildanalyse und ihre Bewertung bei der Objekterkennung und -vermessung. München. DGK bei der Bayerischen Akademie der Wissenschaften, Reihe C, Heft 370.

Fuchs, C., 1998. Extraktion polymorpher Bildstrukturen und ihre topologische und geometrische Gruppierung. München. DGK bei der Bayerischen Akademie der Wissenschaften, Reihe C, Heft 502.

Haralick, R., 1979. Statistical and structural approaches to texture. In: Proceedings of IEEE, Vol. 67 No. 5, pp. 786–804.

Haralick, R. M. and Shapiro, L. G., 1992. Computer and Robot Vision, Volume I/II. 1st edition, Addison-Wesley Publishing Company.

Jähne, B., 1989. Digitale Bildverarbeitung. Springer.

Julez, B. and Bergen, J. R., 1983. Textons, the fundamental elements in preattentive vision and perception of textures. Bell SystemTechnology Journal 62(6), pp. 1619–1645.

Malik, J. and Perona, P., 1990. Preattentive texture discrimination with early vision mechanism. Journal of Optical Society of America, pp. 923–932.

Papoulis, A., 1984. Probability, Random Variables, and Stochastic Processes. Electrical Engineering, 2nd edition, McGraw-Hill.

Reed, T. R. and du Buf, J. M. H., 1993. A review of recent texture segmentation and feature extraction techniques. CVGIP: Image Understanding 57(3), pp. 359–372.

Shao, J. and Förstner, W., 1994. Gabor wavelets for texture edge extraction. In: Proceedings of Symposium on Spatial Information from Digital Photogrammetry and Computer Vision, ISPRS Commission III, Munich, pp. 745–751.

Weidner, U., 1994. Information preserving surface restoration and feature extraction for digital elevation models. In: Proceedings of Symposium on Spatial Information from Digital Photogrammetry and Computer Vision, ISPRS Comm. III, Munich, pp. 908–915.