A MODEL FOR TEXTURE EDGES

E. Triendl and T. Henderson

DFVLR (Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt)
Institut für Nachrichtentechnik, D-8031 Oberpfaffenhofen, Post Weßling
W.Germany

The exact location of boundaries between natural textures is computed by an optimizing edge model applied to a set of texture parameters. Texture parameters are measures computed on a certain neighborhood of each pixel. This filter type operation produces a set of pictures, one for each parameter, and each picture has a blurred appearance due to the averaging properties of texture operators. An edge model taking this smearing effect into account is then applied to this set of texture parameter pictures (or multichannel picture whose channels correspond to texture parameter images). Edge vectors in the resulting multi-channel edge image are then grouped both across channels and spatially to give the location of texture boundaries. An experiment is described applying this process to an image composed of Brodatz textures.

Texture Edge Detection

A local operator that detects texture edges at their precise locations is described. It works on textures definable by the following (recursive) "definition":

> A texture is a picture in which certain local fixed aperture operators produce a constant value, the texture parameter.

A texture operator may be thought of as a spatial filter whose output is averaged over its aperture. Thus, texture edges will be blurred in the texture parameter image. If n texture operators are applied to a picture, the result is an n channel picture with one channel per texture operator. This blurring of edges is similar of the blurring of edges in remotely sensed images due to the image sensor aperture. An optimizing edge model has been proposed by Triendl[1] to solve that problem, and the same approach is used here.

The edge results from each texture parameter image must be combined in some way since a given texture parameter may not distinguish between certain textures. Moreover, spatial grouping is performed to produce more continuous edges. The texture edge detection algorithm has three steps.

1. Compute texture parameters and store as n-channel picture.
2. Apply edge model (which takes into account the blurring by the aperture of texture parameters to all channels separately).
3. Group individual edge elements to get texture boundary (channel grouping followed by local spatial grouping).

Texture Parameters

Twenty-two texture operators are applied to each pixel in the image in Figure 1. This image is composed of 6 juxtaposed textures taken from Brodatz[2] (the textures are: loose burlap, beach pebbles, fur, wood grain, brick wall, and oriental rattan). The gray level mean and standard deviation is the same for all six textures. The texture operators are local operations which use high frequency content and directional histograms of 2 by 2 and 4 by 4 vector gradients averaged in an 11 by 11 neighborhood. The filtered image is reduced by a factor of 10 to produce a one pixel ramp across step edges. The first four principal components of the 22 channel output of the texture operators will be used as the texture parameter image (see Figure 2).

Edges

The edge model assumes a blurred ideal step edge exists between two textures in the texture parameter image. In the case of texture parameter images, the gray level at a pixel is no longer the radiation intensity at the pixel, but rather the result of a texture operator applied to some neighborhood of the pixel. The optimizing edge detector maximizes the correlation between an assumed edge appearance and a small window of the image. This gives the required position, angle and edge quality.

The optimizing edge detector is applied to each channel of the texture parameter image. Figure 3 shows the edge elements of the 4 texture parameters channels. Each pixel of the multi-channel image has an edge descriptor triple associated with

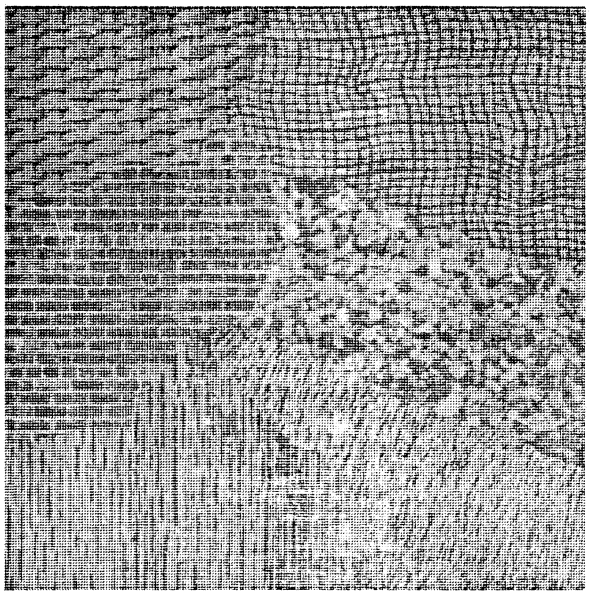

Figure 1. Original Image

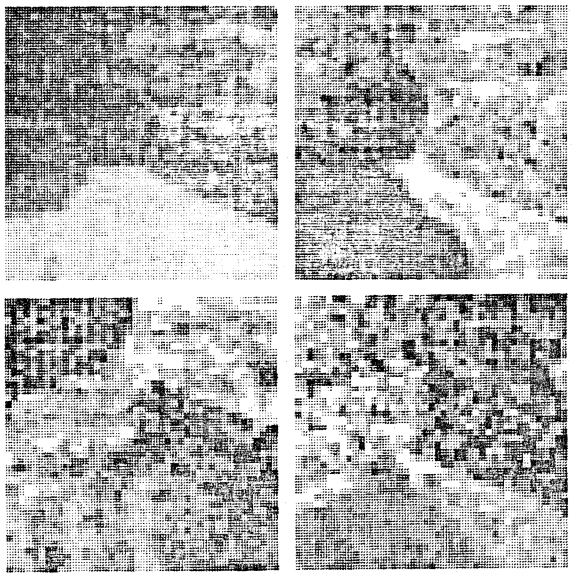

Figure 2. Texture Parameter Image (4 channels)

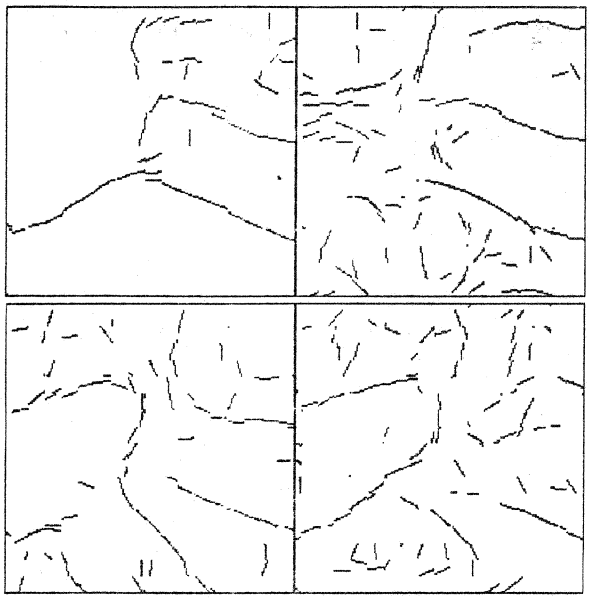

Figure 3. Edges in the Texture Parameter Image

it, namely, (PHI, RAD, COR) which precisely locates an edge with respect to the pixel and also gives the correlation value, COR, of the edge.

Edge Grouping

Given the edge descriptors, two types of grouping can be performed: 1) channel grouping and 2) spatial grouping. Edges detected in different channels, but in the same spatial location, will be grouped if their angles are within a specified threshold, and their quality is high enough. Let (PHI, RAD, COR) be associated with pixel P and (PHI', RAD', COR') associated with pixel P' with P and P' in different channels, but in the same spatial location. If $|RAD-RAD'| <$ threshold, then these triples are mapped into a single new triple in the output image. Each element of the new triple is a weighted sum of the original two triples, e.g.,

$$RAD'' = C*RAD + D*RAD'$$

where C = COR/(COR+COR') and D = COR'/(COR+COR'). This weighted sum gives higher weight to the more certain edge. Of course, the weighting function for angles takes the periodic nature of angles into account. This grouping method will therefore compact similar edges into a single edge, but will leave crossing edges (from different channels) unchanged. Since each pixel in the original image maps to some enlarged neighborhood, several distinct edges may pass through this larger neighborhood. Figure 4 shows the channel grouping image.

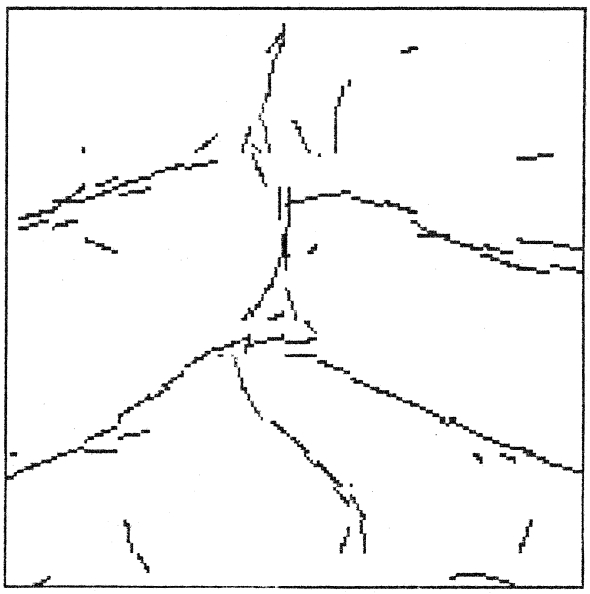

Figure 4. Result of Channel Grouping

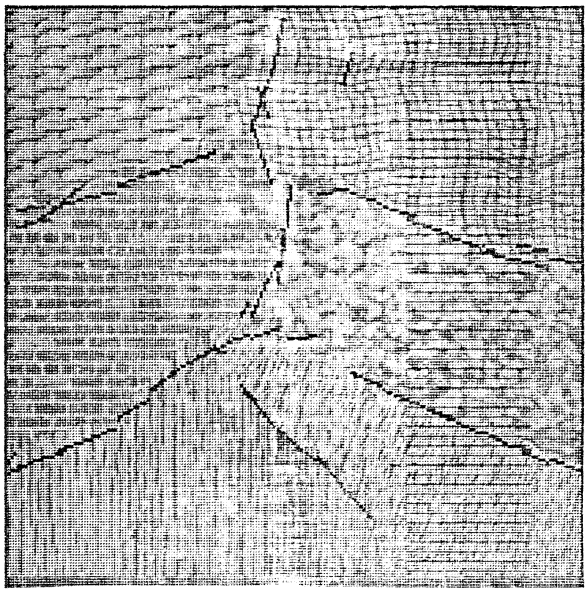

Figure 5. Final Edges (overlayed on original image)

Similarly, spatial grouping is done on the image produced by the channel grouping. The edge element at each pixel is spatially grouped with the other edge responses in a 3 by 3 neighborhood. Figure 5 shows these edges superimposed on the original texture image, and it can be seen that the edges do indeed follow the texture boundaries very well in general. Since no corner detector is involved, the edges disappear in the vicinity of texture corners. The spurious response in the upper right texture is due to a distortion of the loose burlap in that texture; however, a dark stain in the wood grain pattern (lower left) did not affect the detector.

Discussion

A method for detecting texture edges has been defined, and experimental results using the method have been presented. The edge model assumes a step edge; however, it can be modified to account for both lines and ramp edges. This would not alter the basic technique.

The choice of texture parameters is significant only in that the parameter set must be chosen so that each neighboring texture pair produces different values for some texture parameter. Thus, if two textures are indistinguishable by any texture parameter, then no edge will be detected between the textures. The size of the texture operator aperture must be chosen so that the texture parameters produced actually reflect the textures and not components of the individual textures.

The edge triples produced at each pixel provide a basis which could be used by a symbolic system. It must be noted that this edge description is at the texture boundary level. The edge model could also be applied to the original intensity image to produce intensity edge descriptors. These could then be used as input to texture parameter operators or for more complicated texture analysis methods, e.g., as edge segments in Marr's[3] primal sketch or as elements in a generalized cooccurrence matrix (See Davis et al[4]).

We would like to thank the members of the DIBIAS group for their support. This work was also supported by a grant from the Deutsche Forschungs-Gemeinschaft.

References

1. E. Triendl, "How to Get the Edge into the Map," IJCPR, Kyoto, Japan, pp. 946-950, 1978.

2. P. Brodatz, Textures, Dover Publications, N.Y., 1966.

3. D. Marr, "Early Processing of Visual Information. "MIT AL Memo No. 340, December, 1975.

4. L. Davis, S. Johns and J. Aggarwal, "Texture Analysis Using Generalized Cooccurrence Matrices, "IEEE Trans. PAMI, pp. 251-258, 1979.