A COMMON FRAMEWORK FOR THE EXTRACTION OF LINES AND EDGES

Andreas Busch
Institut für Angewandte Geodäsie
Frankfurt am Main
Germany
e-mail: busch@ifag.de

Commission III, Working Group 2

KEY WORDS: Statistics, Vision, Extraction, Mapping, Edge, Pattern Recognition, Robust Estimation, Line

ABSTRACT

In this paper models for extracting lines and edges, i.e. linear features, from digital images are presented. The models are based on a common mathematical approach. In this connection importance is attached to the automation of the recognition of lines and edges. A threshold that is required by the models is estimated by a robust method, i.e. a method that is not sensitive to outliers and that requires no assumption about the statistical distribution of the data. Nodes, i.e. junctions or crossings, and ends of the linear features are recognized and analysed to improve the results there and to find continuations of objects. If we extract both, lines and edges, we are able to find pairs of edges that bound one line object. This yields a complete description of a line and a segmentation of the lines in the image.
An example using satellite image data of SPOT shows that the spectral signatures are suitable for an object-related classification, and that it is possible to distinguish different objects, e.g. rivers and autobahns, and to extract them by knowledge-based techniques.

KURZFASSUNG

In diesem Beitrag werden Modelle für die Extraktion von Kanten und Linien, also linienhafter Merkmale, aus digitalen Bildern vorgestellt, die auf einem gemeinsamen mathematischen Ansatz beruhen. Dabei steht die Automatisierung der Erkennung von Kanten und Linien im Vordergrund. So wird ein Schwellwert, den die Modelle benötigen, mit Hilfe eines robusten, d.h. gegenüber Ausreißern unempfindlichen Verfahrens, anhand der Bilddaten geschätzt, ohne daß irgendwelche Annahmen über die statistische Verteilung der Daten erforderlich sind. Knoten, also Verzweigungen oder Kreuzungen, und Enden der linienhaften Merkmale werden erkannt und analysiert, um an diesen Stellen die Ergebnisse zu verbessern und Fortsetzungen von Objekten zu erkennen. Werden sowohl Kanten als auch Linien extrahiert, ist es möglich, Kantenpaare zuzuordnen, die ein Linienobjekt begrenzen, was zur vollständigen Beschreibung einer Linie und zu einer Segmentierung der Linien im Bild führt.
Ein Beispiel mit Satellitenbilddaten von SPOT zeigt, daß spektrale Signaturen für eine objektbezogene Klassifizierung geeignet sind und so Objektarten, z.B. Flüsse und Autobahnen, unterschieden und wissensbasiert extrahiert werden können.

1. INTRODUCTION

1.1 Background

Our approach for extracting linear features from digital images is based on the conceptual distinction of lines and edges. These basic terms are best explained by an example. For instance, we realize objects like roads and rivers in satellite imagery as lines, which are bounded by two edges. Since lines are formed by two edges, they are more complex objects than edges. To both, lines and edges, we refer as linear features or linear objects.

Owing to their different complexity line and edge detection have developed separately. A large variety of edge detectors, mostly based on linear filtering techniques, is known from image processing. They range from classical methods like Roberts, Sobel, or Prewitt gradient filters (see e.g. Haralick and Shapiro 1992/93, vol.1, p.337) to sophisticated methods like Canny (1986) and Deriche (1990) edge detectors. However, line detection is usually done by line following (e.g. Grün 1994). The disadvantage of these techniques is that they are often semi-automatic since a starting point for the algorithm is given by an operator. On the other hand it is favourable that interpretation of an object is directly done by the operator when selecting a starting point. Other methods for finding lines in digital images are related to digital filtering because they scan the whole image deciding for each pixel whether it is a line pixel or not (Busch 1994). A combination of both approaches for line detection by using line following for improving the results of the second method is promising.

The goal of this work is the derivation of a common framework for line and edge extraction. A common model for both linear features is advantageous since it leads to consistent results and allows linking of lines and edges that correspond to one object. It is possible to process some tasks, e.g. threshold estimation, for both features in the same way.

If we stick to the subdivision of computer vision in three levels, namely image processing (low-level vision), pattern recognition (mid-level vision), and image understanding (high-level vision), the main part of this paper belongs to the mid-level of computer vision. The approach is suitable for delivering salient objects, eliminating spurious details, and for computing useful attributes of the objects. For all features and attributes statistical measures of quality or uncertainty can be derived from the image data. They are important for a valuable and complete data flow to the next level of computer vision, i.e. image understanding. A major aspect of this paper is the automation of linear feature extraction. We attach importance to the fact that control parameters of the algorithm are to a great extent independent of the image data, i.e. they are standardized like, e.g. a significance level.

1.2 Underlying models

We use generic models for lines and edges which have a common mathematical background. The edge model is a third order polynomial function that is fitted to the grey levels in an image window. It is known as facet model (Haralick et al. 1983, Haralick 1984). The polynomial represents the grey levels as a function of the row and column coordinates in an image window and takes the form

$$\begin{aligned} g(x,y) &= k_0 \\ &+ k_1 x + k_2 y \\ &+ k_3 x^2 + k_4 xy + k_5 y^2 \\ &+ k_6 x^3 + k_7 x^2 y + k_8 xy^2 + k_9 y^3 \ . \end{aligned} \tag{1}$$

The coefficients k_i are determined by a least squares fit of the polynomial in the image window. From this we have derived our line model which is a second order polynomial function (Busch 1993, 1994)

$$\begin{aligned} g(x,y) &= k_0 \\ &+ k_1 x + k_2 y \\ &+ k_3 x^2 + k_4 xy + k_5 y^2 \ . \end{aligned} \tag{2}$$

The polynomial model offers great flexibility because it can easily be used with arbitrary window sizes, and because the grey levels in the image window can be weighted according to different models, e.g. as in Box or Gaussian filters. Simple classical gradient filters, like Sobel or Prewitt, are included as special cases. Additionally, the redundancy of the polynomials least squares fitting in an image window allows control and self-diagnosis of the algorithm by means of statistical testing.

The decision whether a pixel is an edge pixel or a line pixel is made from the first and second order derivatives of the polynomial functions and their principal directions. For extracting edges we calculate the intersecting polynomial of (1) which falls in the direction of the gradient vector. The centre pixel of the image window is classified as an edge pixel if the maximal absolute value of the polynomial's first derivative is located inside the pixel and differs significantly from zero. Line pixels are recognized using the intersecting parabola of (2) which falls in the direction of maximal curvature. A pixel is a line pixel if there is a zero crossing of the parabola's first derivative, i.e. if the extremum of the parabola falls inside the pixel and if the parabola's curvature is sufficiently large. By this procedure we obtain line and edge positions with sub-pixel resolution.

2. DISCRIMINATING NOISE AND REAL DETAILS

The models introduced in Section 1.2 require a decision about the significance of the absolute value of the polynomial's first or second derivative since we want to know whether it is different from zero in order to discriminate real details from effects that are due to noise. We can do this by hypothesis testing which is done individually for every image window, by a single threshold that is applied to the whole image, or by a combination of both methods.

2.1 Hypothesis testing

A classical method for checking significance is hypothesis testing. Since it is based on the assumption of normally distributed observations it may lead to unsatisfactory results when applied to image data where this condition is often violated (Busch 1994). Whenever hypothesis testing is applied to images it must be checked whether it is sufficient to assume that the data are normally — or whatever else might be prerequisite — distributed.

2.2 Robust estimation

We want to estimate a threshold that allows to separate artefacts and spurious details in the image from salient lines and edges. Thus, what we need is related to a noise estimate, but it is different from thresholding techniques for binarizing images (Sahoo et al. 1988).

The basic idea of the method is to produce an image with a lot of noisy linear features and to estimate the threshold from the noise then. In the first step we apply the method of Section 1.2 using zero as threshold. Hence, the decision about the line or edge attribute is made using only the location of the extremum of the polynomial's derivative ignoring the derivative's amount. To estimate a threshold from the mass of pseudo features produced by this, we start with the consideration that linear features in digital images are formed by long chains of pixels. Thus, single, i.e. isolated pixels classified as lines or edges are due to noise mostly and are suitable items for deriving a noise estimate. So we collect the derivatives of all single edge or line pixels and take their median as a robust estimate for the typical derivative of a noisy linear feature. As we are using isolated feature pixels our approach is different to the one of Venkatesh and Rosin (1995) who consider edge continuity and edge curves to determine a threshold. To eliminate most of the single pixels we must use a threshold which is larger than the median, since the median eliminates half of the noise. We see this from its definition:

$$\text{Median M :} \quad \int_{-\infty}^{\mathsf{M}} \mathsf{p}(c_s)\, \mathsf{d}c_s = 0.5 \tag{3}$$

Here c_s denotes the parameter which is the basis of the decision. This is the maximal absolute value of the polynomial's first derivative in case of edge pixels and the curvature of the parabola in case of line pixels. The index s reminds of the fact that the probability density function p represents single linear feature pixels.

By generalizing (3) we find thresholds corresponding to convenient significance levels

$$\begin{aligned} \mathsf{T}_{10} &= \mathsf{P}_{90} : \int_{-\infty}^{\mathsf{P}_{90}} \mathsf{p}(c_s)\, \mathsf{d}c_s = 0.90 \\ \mathsf{T}_{5} &= \mathsf{P}_{95} : \int_{-\infty}^{\mathsf{P}_{95}} \mathsf{p}(c_s)\, \mathsf{d}c_s = 0.95 \\ \mathsf{T}_{1} &= \mathsf{P}_{99} : \int_{-\infty}^{\mathsf{P}_{99}} \mathsf{p}(c_s)\, \mathsf{d}c_s = 0.99 \ . \end{aligned} \tag{4}$$

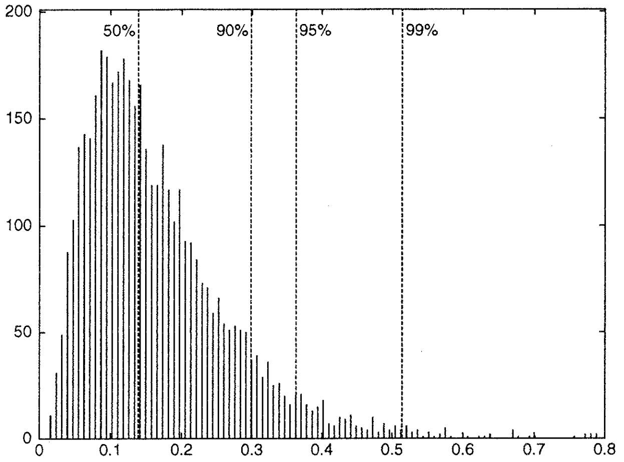

Figure 1: Typical distribution of the gradient/curvature of single linear feature pixels. It has been obtained by dividing the range of the parameter into 100 intervals of equal width and by counting the number of occurences in each interval. Additionally, the median and the thresholds corresponding to the significance levels 90%, 95%, and 99% are shown.

Figure 1 shows an example of the discrete distribution of the derivative of single pixels that are classified as linear features. Typically, some thousand single pixels are found when applying the method to images of 1000 × 1000 pixels. So we have a sample that is large enough to obtain a statistically well-founded estimate for the threshold. Generally, the thresholds (4) range between the second and sixth multiple of the median. The advantage of this estimation is that we do not need any assumption about the underlying distributions, neither for the distribution of the observation, i.e. the grey values, nor for the distribution of the isolated feature pixels. Because the thresholds (4) were developed from the median (3) they have properties in common with the median. Especially, the estimated thresholds are not sensitive to a certain percentage of outliers, i.e. they are robust. This estimation technique has proven to be very powerful with a variety of digital images of any resolution (satellite, aerial, close range, and microscopic images) which have been tested by the author.

2.3 Less sensitive hypothesis testing

The methods presented in Sections 2.1 and 2.2 have different properties. While hypothesis testing is based on a local decision that is made individually for every image window, the robust estimation leads to one global threshold for all pixels. Tests with lots of different images have shown that hypothesis testing tends to give some more details than the robust estimate, but also more noise. The robust estimate delivers all details that rise above the estimated noise level.

It is possible to combine both methods by using a threshold that excludes small values from hypothesis testing and that rejects them anyway. This may be interpreted as a less sensitive hypothesis test (Koch 1985, Koch 1990, p.88). The threshold must be distinctly smaller than those given in (4). We recommend to use values smaller than the median. Although the threshold can be measured as a fraction of the median, it is an additional control parameter of the algorithm that complicates automatic processing.

3. FROM PIXELS TO OBJECTS

The results of the methods presented in Section 2 are given as an iconic representation, i.e. they are images again. Additionally, we have line or edge positions with sub-pixel resolution. To improve the results further steps of processing are necessary. In detail these are a skeletonization process and the detection of end and node pixels (Schickler 1992, Busch 1994).

Since the extracted lines and edges are sometimes wider than one pixel we thin them by a skeletonization process that leads to linear features of a width of one pixel. We use a skeletonization method designed specially for linear features. It is based on the line and edge model, makes use of their direction and works carefully to preserve the topology of the line or edge network. Behind this is the concept of evading decisions that are made better at a later stage of computer vision.

All pixels belonging to linear features are classified as ends, nodes, i.e. sites where linear features join or cross, or simple members of a line or an edge. A grouping process links line or edge pixels to objects which connect node and/or end pixels. Closed chains of pixels without any end or node pixel are recognized, too. By this we have a vector representation of the linear features.

3.1 Analysis of nodes and ends

We want to take a closer look at nodes and ends now since they are known to be the crucial point of linear feature extraction. So the models of Section 1.2 may fail at nodes because the structure there does not correspond to the line or edge model. This leads to gaps in the linear features that have to be closed. Besides these spurious gaps and ends we have real ends of the linear features visible in the image which typically occur if objects overlap each other. Thus, we have to analyse nodes and ends to enhance the extraction process there.

When examining nodes and ends we are looking for end pixels and other nodes nearby. The prospect is to find items for closing gaps and to unite linear features. The number and the length of the converging linear features are helpful criteria for measuring the significance and importance of a node. Nodes are classified as crossings, i.e. four linear features are meeting, and branches or junctions, i.e. three linear features converge. The topology of the node and the direction of the meeting of linear features allow to find features which are each other's continuation. If there is an even number of incoming linear features, we have unique correspondence of opposite features. Additional information comes from the direction of the linear features. Sometimes there may be also pseudonodes, i.e. nodes where two linear features meet. They occur because the careful skeletonization algorithm avoids thinning if the structure is ambiguous. We analyse the pseudonodes to eliminate them by joining the incoming linear objects or to classify them as corners which are recognized using the direction of the linear features.

3.2 Combining lines and edges

If we extract both, lines and edges, we are able to find pairs of edges that correspond to a line object, i.e. bound one line. This is much easier than trying to find parallel edges without knowing the line object. Besides the geometric accuracy of the line position benefits from this, since — due to the symmetric parabolic model (2) — the line position is

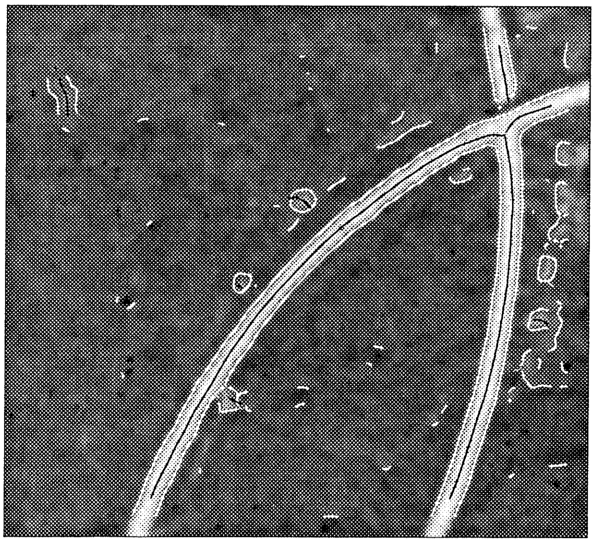

Figure 2: Detail of a KWR 1000 scene showing two roads with extracted lines and edges, 235 × 213 pixels, ground resolution ~ 2m.

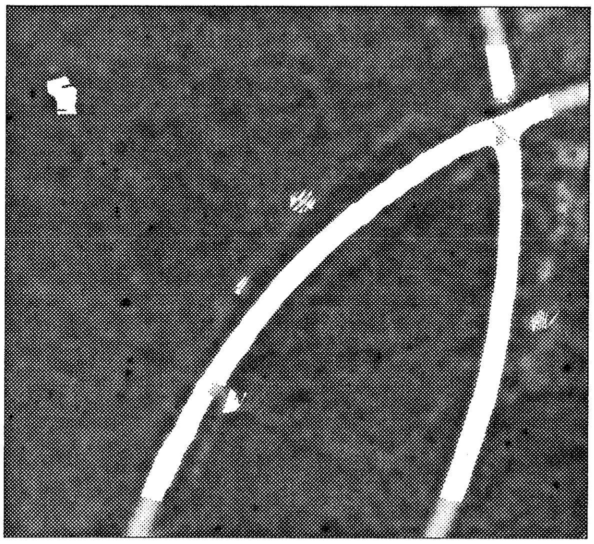

Figure 3: Resulting segmentation

affected by different grey levels on the left and right side of the line. So we can use the edge positions instead which are more exact. Criteria for evaluating correspondence are the neighbourhood, the constancy of the line width, and the direction of the linear features. After that we have a geometric description of the line including its width. We use this for the segmentation of the lines in an image, too. An example based on data of the Russian KWR 1000 sensor is shown in Figure 2 and Figure 3.

4. ROAD, RAILWAY, OR RIVER?

The method described so far is part of low and mid-level computer vision since no knowledge about the real objects depicted in the image has been incorporated. So it may be used to find lines and edges in arbitrary digital images of any resolution.

We want to apply the method to satellite images to extract objects that are relevant to cartography, e.g. roads and rivers. Although we know that there are limits due to the spatial resolution of operational sensors like SPOT and TM, so that only major roads and autobahns can be detected (McKeown 1994), we think that our example is instructive and shows the capability of the methods. Extracting lines from a SPOT or TM satellite imagery results in lots of objects that are not interpreted, i.e. all that we know about them is that they fit to our line model of Section 1.2. Besides the objects we are interested in, there is a large variety of other ones, like open strips in a wood, long and narrow fields, or long buildings.

In our example we want to use two kinds of knowledge for discriminating objects: knowledge about the width of the objects and knowledge about their spectral characteristics. Since the result of the line extraction depends on the size of the image window used for the least squares fit of the polynomial (2), we are able to select lines of different width. So it is not possible to detect lines of large width with a small window, while a large window is not sensitive to narrow lines due to smoothing. For using spectral characteristics we take advantage of the fact that the extracted lines are skeletonized as mentioned in Section 3. Hence, we have a representation of their middle axis containing only few mixed pixels which constitute the crucial point in multispectral classification. Therefore, the detected lines are a good starting point for an object-related multispectral classification. In our example the knowledge comes from training areas that have been marked interactively by an operator and that consist of detected line pixels only. But it is possible to represent the knowlegde about the spectral characteristics of roads and rivers in a knowledge base. Additionally, unsupervised classifiers (e.g. Schulz and Wende 1994) allow further improvement and automization.

The example is based on SPOT XS data (Figure 4). For extracting the river Main that is flowing from the upper left corner to the right side of the image we have applied the line extraction technique described in Section 1.2 to band 1. We have used a window size of 15 × 15 which is suitable for the width of the river that varies from 6 to 11 pixels. The significance level for the robust estimation method of Section 2.2 has been set to 10%. Figure 5 shows the result. The small part of the river Rhine in the lower left corner of the image has not been detected because of if its width of more than 25 pixels. This demonstrates that the line model (2) allows to dinstinguish lines of different width. For all pixels depicted in Figure 5 we have gathered the spectral information from the three bands so that multispectral classification has been applied to these pixels only. The result of the classification (Figure 7, bold line) illustrates that it has been possible to select the river from the other linear features.

To find roads we have analysed the SPOT XS image (Figure 4) setting the window size and the significance level to 5 × 5 and 10%, respectively. The three bands have been processed independently. Figure 6 combines the results by a logical "OR" operator and shows all pixels that have been recognized as line pixels in any of the three bands. This procedure is different to the one used for the river because the width of the roads is close to the spatial resolution of the SPOT XS sensor. Thus, we have needed information from the three bands, whereas in case of the river it has been sufficient to analyse one band. In Figure 7 we see the result of the classification together with the extracted river. It demonstrates that it has been possible to recognize the autobahns

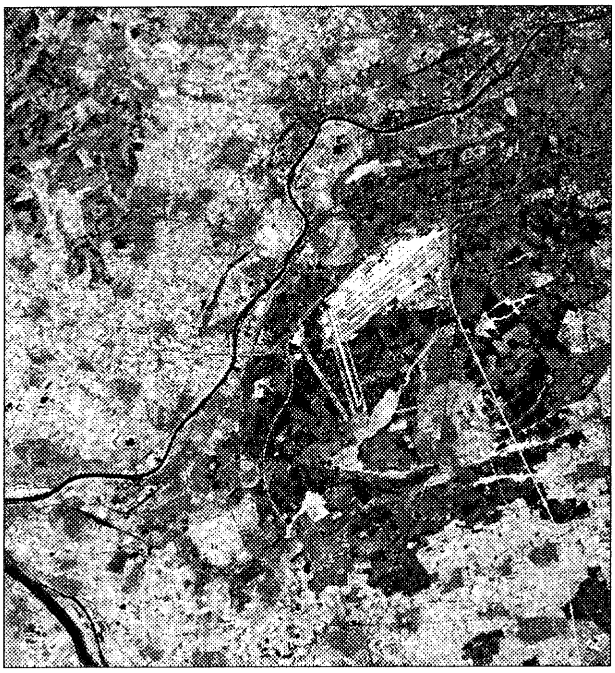

Figure 4: Part of a SPOT XS scene, ground resolution 20m × 20m, 1200 × 1300 pixels, composite of the three bands.

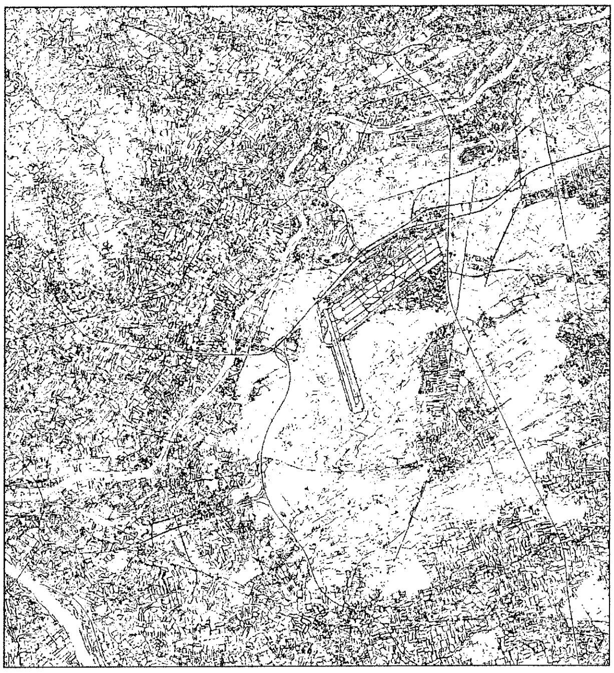

Figure 6: Lines extracted from the SPOT XS data (Fig. 4) using a window size of 5 × 5 pixels.

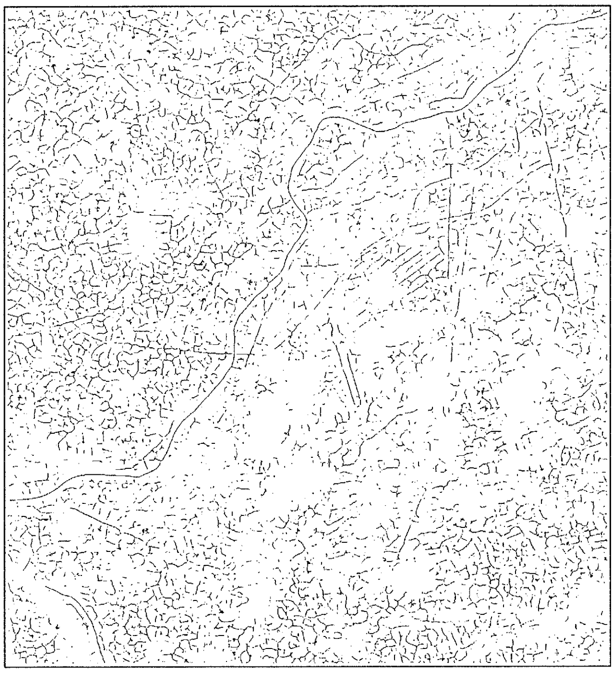

Figure 5: Lines extracted from band 1 of the SPOT XS data (Fig. 4) using a window size of 15 × 15 pixels.

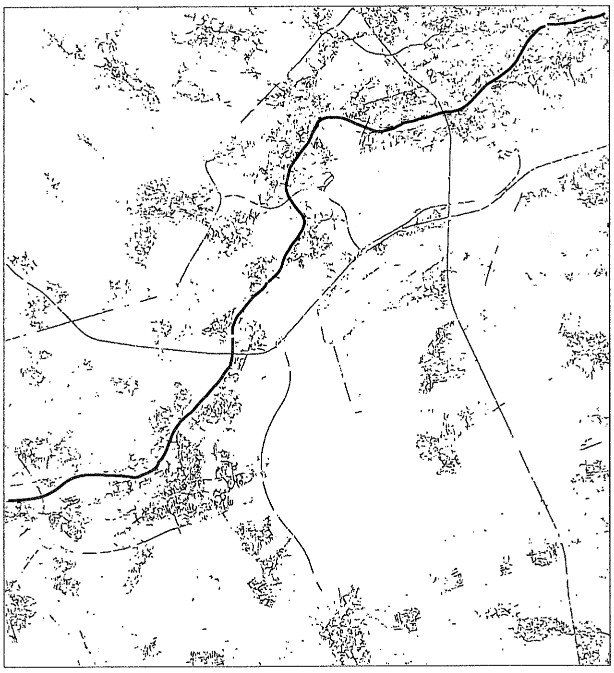

Figure 7: Result of selecting water pixels from all line pixels of Fig. 5 and road pixels from all line pixels of Fig. 6 by multispectral classification. The result has been vectorized using the sub-pixel positions of the lines and the methods mentioned in Section 3. The bold line represents the river. All other lines are roads.

almost completely. Additionally, there are small fragments of roads inside cities.

The use of multispectral images is not suitable for distinguishing all kinds of lines in an image, e.g. it will be impossible to recognize railways from SPOT or TM data in this way. It is necessary to make use of the knowledge that we find in existing maps or Geographic Information Systems and to combine it with the image using matching techniques.

5. CONCLUSIONS AND OUTLOOK

The results of this paper show that the presented models allow to automatically extract lines and edges from digital images. The extraction of lines as well as edges leads to the segmentation of the lines in the image. Even in cases where the width of lines is close to the limit of the resolution of the image, it was possible to find lines. Due to the spatial resolution of the operational remote sensing sensors SPOT and TM feature extraction from these images typically leads to a lot of spurious details. We face this fact by two different means. One is the robust estimation of a threshold that enables us to select features rising above the noise level. The other is the use of spectral characteristics of linear features which showed that it is possible to distinguish roads and rivers among a large variety of other lines, which is demonstrated by the example.

Further work on the presented methods aims at incorporating more knowledge. For instance, the use of knowledge about autobahn junctions or knowledge about the radii of curves of railways and roads is promising. Sophisticated line trackers (Grün and Li 1994) will improve gap closing and make the detection and analysis of nodes more robust. This enhancement of the results especially at intersections or junctions is important since they are the frame for matching the extracted objects with the objects in an existing Geographic Information System.

In this paper the methods have been applied to satellite images only. But it is possible to make use of them for road extraction from aerial images (Li et al. 1992, Ruskoné et al. 1994a, 1994b), too. The link to those methods is an image pyramid which starts at coarse resolution improving the results in subsequent steps by making use of the approximations achieved earlier.

REFERENCES

Busch, A., 1993. Ein Verfahren zur Erkennung linienhafter Objekte in digitalen Bildern. In: Pöppl, S.J., Handels, H. (Eds.). Mustererkennung 1993. Springer-Verlag, Berlin, pp. 616–623.

Busch, A., 1994. Fast recognition of lines in digital images without user-supplied parameters. In: Ebner, H., Heipke, C., Eder, K. (Eds.), International Archives of Photogrammetry and Remote Sensing, Vol.30, Part 3/1, SPIE – The International Society for Optical Engineering, Washington, pp. 91–97.

Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8, pp. 679–698.

Deriche, J., 1990. Fast algorithms for low-level vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, pp. 78–87.

Grün, A., Li, H., 1994. Semi-automatic road extraction by dynamic programming. In: Ebner, H., Heipke, C., Eder, K. (Eds.), International Archives of Photogrammetry and Remote Sensing, Vol.30, Part 3/1, SPIE – The International Society for Optical Engineering, Washington, pp. 324–332.

Haralick, R.M., Watson, L.T., Laffey, T.J., 1983. The topographic primal sketch. Int. J. Robotics Research 2, pp. 50–72. Reprint in: Kasturi, R., Jain, R.C. (Eds.) 1991. Computer Vision: Principles. IEEE Computer Society Press, Los Alamitos, pp. 195–217.

Haralick, R.M., 1984. Digital step edges from zero crossing of second directional derivatives. IEEE Transactions on Pattern Analysis and Machine Intelligence 6, pp. 58–68.

Haralick, R.M., Shapiro, L.G., 1992/93. Computer and Robot Vision, Vol. 1 and 2. Addison-Wesley, Reading.

Koch, K.R., 1985. Ein statistisches Auswerteverfahren für Deformationsmessungen. Allgemeine Vermessungs-Nachrichten 92, pp. 97–108.

Koch, K.R., 1990. Bayesian Inference with Geodetic Applications. Springer-Verlag, Berlin.

Li, S.Z., Kittler, J., Petrou, M., 1992. Matching and recognition of road networks from aerial images. In: Sandini, G. (Ed.). Computer Vision – ECCV'92, Second European Conference on Computer Vision, Santa Margherita Ligure, Italy, Proceedings. Springer-Verlag, Berlin, pp. 857–861.

McKeown, D.M., 1994. Top ten lessons learned in automated cartography. Paper presented at ISPRS Comm. III Symposium, Munich, Germany, Sept. 5.–9. 1994.

Ruskoné, R., Airault, S., Jamet, O., 1994. Road network interpretation: A topological hypothesis driven system. In: Ebner, H., Heipke, C., Eder, K. (Eds.), International Archives of Photogrammetry and Remote Sensing, Vol.30, Part 3/2, SPIE – The International Society for Optical Engineering, Washington, pp. 711–717.

Ruskoné, R., Airault, S., Jamet, O., 1994. A road extraction system using the connectivity properties of the network. Zeitschrift für Photogrammetrie und Fernerkundung 62, pp. 174–180.

Sahoo, P.K., Soltani, S., Wong, A.K.C., 1988. A survey of thresholding techniques. Computer Vision, Graphics, and Image Processing 41, pp. 233–260.

Schickler, W., 1992. Merkmalsextraktion für Meßaufgaben in der digitalen Photogrammetrie. Ein statistisches Auswerteverfahren für Deformationsmessungen. Zeitschrift für Photogrammetrie und Fernerkundung 60, pp. 107–116.

Schulz, B.S., Wende, C., 1994. Vollautomatische, hochdifferenzierte, pixelweise Klassifizierung multispektraler Bilder – eine neue Methodik –. In: List, F.K. (Ed.), Vorträge 13. Wissenschaftlich-Technische Jahrestagung der DGPF, Geoinformation durch Fernerkundung. Publikationen der Deutschen Gesellschaft für Photogrammetrie und Fernerkundung, Band 2, Berlin, pp. 193–197.

Venkatesh, S., Rosin, P.L., 1995. Dynamic threshold determination by local and global edge evaluation. Graphical Models and Image Processing 57, pp. 146–160.