LINEAR FEATURE EXTRACTION AND ROAD RECOGNITION FROM LARGE SCALE DIGITAL AERIAL IMAGES

Mathias J.P.M. Lemmens, Erik W. Brüel, Frank Fennis
Delft University of Technology, Faculty of Geodesy,
Institute of Photogrammetry and Remote Sensing,
Thijsseweg 11, 2629 JA Delft, The Netherlands
Comm. VII/1

Abstract: The aim is to analyze images using shape rather than multispectral and texture information. Image analysis is a two-stage process: segmentation and pattern recognition. Shape analysis requires linear features segmentation, which contains two steps: edge detection and line-following. Edge detection can be carried out by gradient operators. There are several techniques of line-following, e.g. search near an approximate location, relaxation and dynamic programming. We developed relaxation and dynamic programming techniques for linear feature extraction. Combined with several smoothing methods and edge detectors, especially the geometric accuracy of line extraction is considered. A large scale aerial image is used as test data. The image is scanned in blue, green and red, pixelsize 80 micron. Using general knowledge about roads, pattern recognition techniques are applied based on shape measures to extract roads in urban areas. Dynamic programming is superior to relaxation. The results of road recognition is qualitatively indicated.

Introduction

The main objective of the present paper is the extraction of object information from spatial digital imagery on the basis of shape. Two advanced segmentation-by-boundary-detection techniques: edge relaxation and dynamic programming as well as their geometric accuracy are specially emphasized. By spatial digital images we understand EM-images of the earth, taken from airplanes or satellites. Compared with the common digital images spatial digital images contain a tremendous number of objects, which are moreover, extremely complex ordered. So, the application of an object model, commonly applied in computer vision, is very difficult to realize.

The present investigation regards road recognition in large scale aerial photographs, although no operational road recognition system is described. The manner of approach of road recognition is strongly affected by (Fischler et. al, 1981):

- The scale of the image (i.e. resolution);
- The degree of occlusion by clouds, intervening objects, and so on;
- The degree of density of linear detail (i.e. urban versus rural scene).

A large scale true colour aerial photograph of an urban area is chosen. The illumination conditions are good and the roads are only in a minor degree occluded by shadows and intervening objects.

Information extraction about real world objects from images involves four steps: (1) image formation, (2) preprocessing, (3) analysis and (4) presentation. A detailed description of these stages is given in Lemmens (1988). Remote sensing applies mainly multispectral (MS) classification as pattern recognition technique, althoughin texture analysis earns recently interest too.

Grey value and shape are the principal basic elements of image analysis (Lemmens, 1987, 1988). Combination of grey values leads to colour, i.e. multispectral (MS) information. Spatial repetition of grey values leads to texture. So, actually, grey value is used and shape hardly. This paper undertakes some cautious shape recognition steps. A true color aerial photograph, size 23 x 23 cm and scale 1:3500, of an urban area is selected as testdata and drumscanned in three bands (blue, green and red) with pixelsize 80 micron and 8 bits quantization representing 256 grey values. MS classification only poorly discriminates man made objects and our testdata affirms that. Although shape is the principal information the MS data showed to be a very usefull aid in both segmentation and object recognition. Roads were chosen because they have very striking and distinguishable properties. To be recognizable an object must have a certain minimum size in image space. For roads a minimum width of 6 pixels was taken. Roads with a minimum width of 2 meter in object space should be identifiable. So, an image scale 1:3500 and a pixelsize of 80 micron is chosen. An urban area is selected because man made objects heavely challenges the capabilities of MS classification. The computations were carried out on 128x128 and 512x512 subimages. After the first experiments the following problems could be denoted:

- roads and buildings are spectrally hardly distinguishable;
- shadows, cars and the overhanging leafs of trees cover roads;
- urban roads in modern quarters of town end often in car parks.

The photograph is also manually digitized using an analytical plotter (Planicomp-C100), with a measuring accuracy of $\pm$ 5 mircon. The digital map is stored as a 2-D vector format data base; (fig. 1 shows a plot) and used as reference.

Before analysis can start some preprocessing is necessary. Radiometric restoration isn't found necessary. Because a drumscanner may introduce geometric distortions a geometric restoration by an affine transformation is carried out:

$$x = a_0 + a_1X + a_2Y$$

$$y = b_0 + b_1X + b_2Y$$

with:

(x, y):	coordinates of the digital image
(X, Y):	reference coordinates
(a_i, b_i):	i = 0, 1, 2: transformation parameters.

Noise reduction and diminuation of texture is performed, using non-linear smoothing filters. The following filters are applied: (1) conditional average filter, (2) median filter and (3) edge preserving filter.They are treated in detail in section 3. Because they may

affect edge location, they form part of the precision analysis. Conditional averaging is applied in the process of road recognition. Also several edge detectors are considered: (1) normal gradient, (2) Prewitt operator and (3) Sobel operator. Their masks are given in fig. 2. The precision of edge relaxation and dynamic programming are compared in section 3. Road recognition is performed with edge relaxation and qualitatively indicated in section 5. Smoothing and boundary detection are based on the approach of (Prager, 1980). The above is summarized in fig. 3.

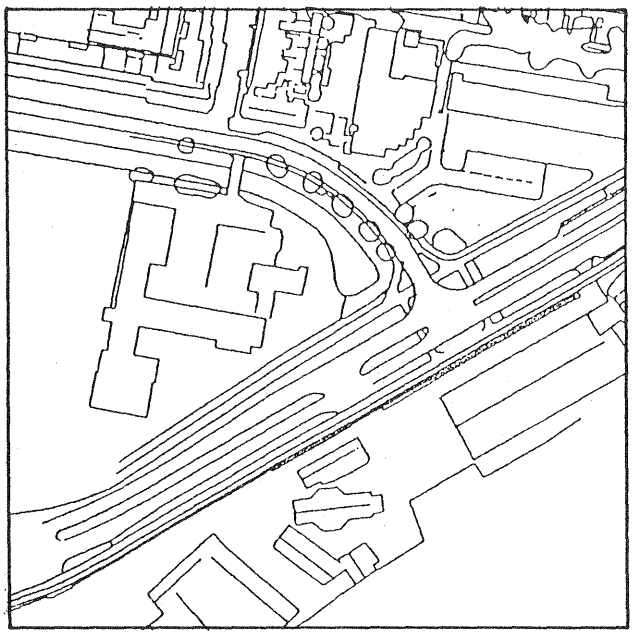

Fig. 1

2. Road recognition in low resolution images: a review

Line objects, like roads, rivers and railroads, express the principal structure of an area. Since the time, remote sensing data became available in digital format, it is tried to extract 'line-like' features automatically. From the functional, e.g. dictionary, description of a road physical and geometric properties can be deduced (Bajcsy and Tavakoli, 1976):

- a road has a smooth surface, without valleys and holes;
- the ground and surface of a road must be firm;
- roads, generally, don't end in the middle of nowhere, but are interconnected as a network;
- there is a limit on the steepness, width and curvature of a road.

These properties have to be transformed to visual ones to achieve a road model:

- spectral properties of a road correspond to those of concrete, asphalt and rock;
- a road has a limited width, limited curvature, and limited steepness.
- the length of a road must exceed a certain minimum value.

In the present investigation condition 2 and 3 are mainly employed. Research, carried out in the past, is mainly limited to small-scale images, mostly from satellite sources. The developed techniques extremely rely on the narrowness of line objects at small scales. Gradient (-like) operators are used, very similar to edge detectors. In general, road recognition can be divided into the following broad stages:

- application of a gradient-like operator (strip detector) to segment the image;
- elimination of small segments;
- connection of road regions;
- when road axes are demanded, thinning of the roads to one pixel thick lines.

For small scale images the gradient-like operator may be considered as a strip detector. It is the basic operator. The more advanced the less problems in later stages. Actually, later stages perform a reparation. A simple strip detector requires many reparations.

Bajcsy and Tavakoli (1976) use a grey value thresholding on Landsat MSS Land 2, assuming that roads are built from concrete and rock. The method relies strongly on spectral information, causing multiple errors and therefore requiring active reparation stages. Sijmons (1987) applies a median filter. The original image g is median filtered, leading to g_{rm}; g_{rm} is subtracted from the original image , giving $g' = g - g_{rm}$. Actually, g' contains noise, but, because lines of one pixel thickness are smoothed by the median filter, also linear features are detected. A threshold is employed to binarize g'.

In (Fischler et. al., 1981) the homogeneity in grey value along a potential road track and the contrast of this potential track with the adjacent terrain is measured and assigned to one value. With masks the four principal directions (row, column and the two diagonals) are evaluated. The authors note some significant weakenesses of the operator. It is sensitive to:

- road orientation, because the operator examines only the four principal directions;
- raster quantization;
- sharp changes in road direction;
- certain types of contrast between road and terrain.

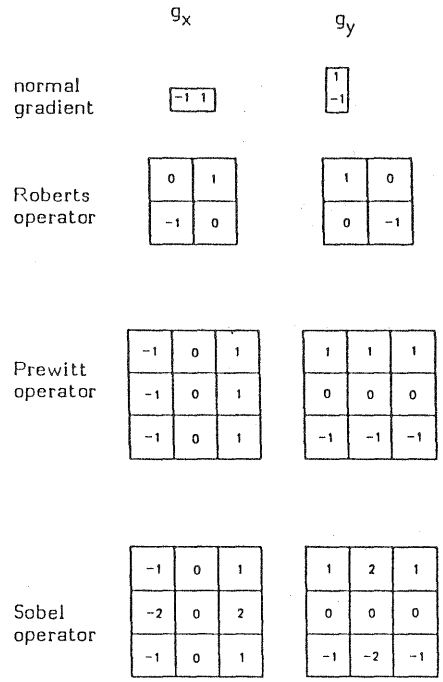

Fig. 2

The basic operator of Groch (1982) evaluates the shape of grey value profiles at locations, where the presence of a line-like structure is likely. Initial values of the shape of the profile, i.e., width of the valley or crest and depth of the valley or height of the crest, dx_0 and dg_0, respectively (see fig. 4) are given a priori, and depend on the kind of linear feature one wants to

detect. To avoid human interference also the starting places are determined automatically with the S(tart)-operator. The image is covered with a coarse raster. Along each rasterline the basic operator evaluates the grey values. If a point on the profile corresponds to the specifications (dx_o, dg_o) it becomes candidate point and centre of two concentric circles (see fig. 5). The grey values in between the circles are evaluated with the basic operator. A candidate point is accepted when it forms together with accepted points in between the circles a collinearity. The direction of the line is used as initial direction of the road.

Fischler et. al. (1981) apply a dynamic programming to link the road elements. The dynamic programming approach is very similar to our boundary detection approach.

Groch (1982) integrates entirely the process of search and connection. Starting from the initial road element, profiles across the lengthening of the initial direction are evaluated. The best point is chosen, the road is enlarged, a new local direction is computed and so on. When no point on the profile is accepted the operator goes back. When, after a repeated back stepping, still no road element is found, the area in the

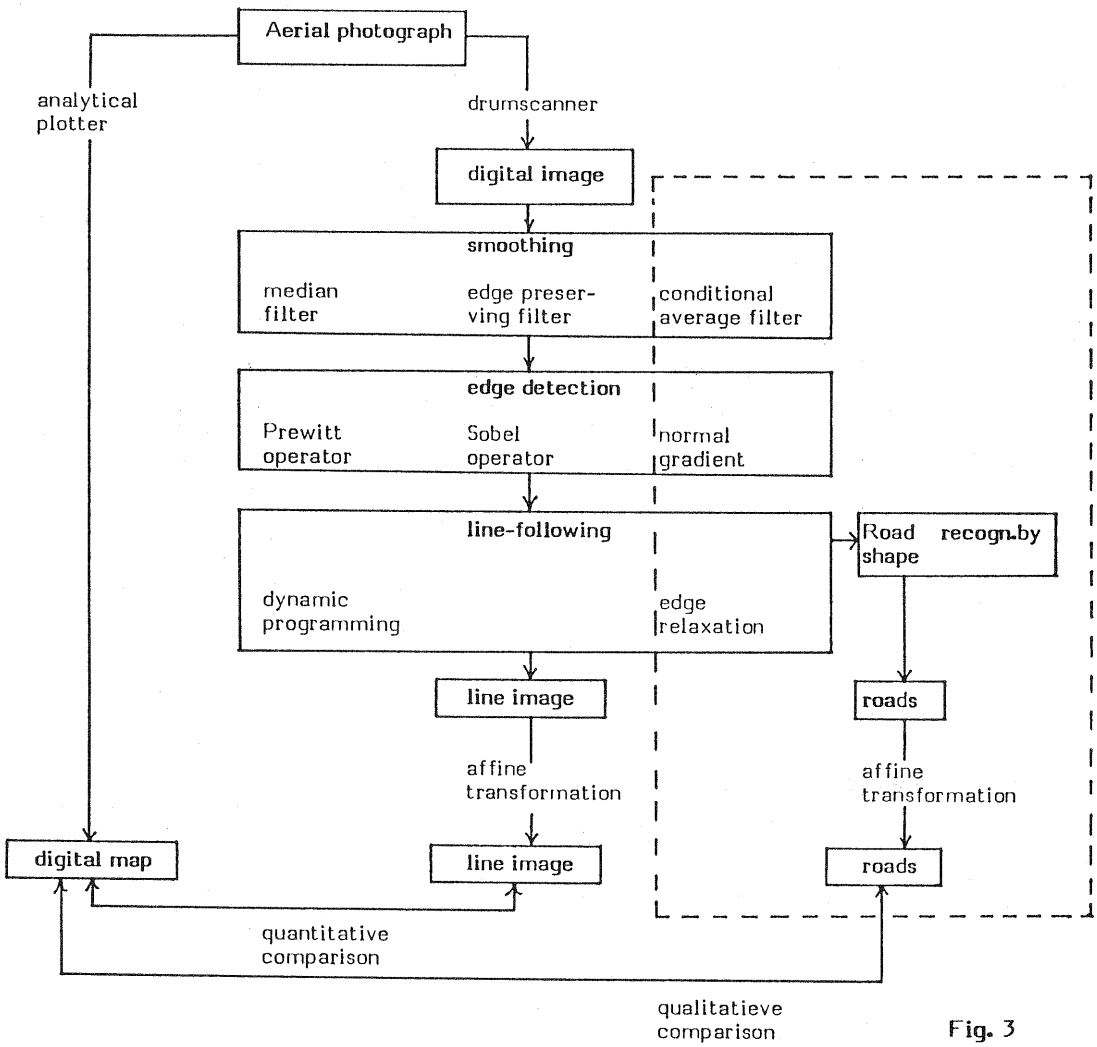

Fig. 3

To link separated road segments (Bajcsy and Tavakoli, 1976) use a 'road grower', which connects neighbouring road regions which have minimal difference between their directionalities ('directional proximity'). Because there will still remain gaps between roads, the next step is to connect those road segments that minimize the distance between disconnected pieces of road, regardless of directionality ('distance proximity').

Sijmons (1987), eliminates road segments smaller than four pixels. Disconnected road elements are connected by evaluation of the topological structure in 3 x 3 windows of the binary image. The drawback of the approaches of Bajcsy and Tavakoli (1976) and Sijmons (1987) is that the road elements are linked in the binary image. Both edge strength and edge direction are dropped in a early stage and just topology is used. An additional limitation of the operator $g' = g - g_m$ is the one pixel thickness condition of road elements introducing a strong dependency on scale and feature type.

lengthening of the local direction is covered by a rectangle of fine-drawn parallel profiles enabling an analysis in great detail to bridge gaps. This operator is actually a road connector. The approach has great flexibility with regard to changing circumstances, like width and grey value of the line feature. Its great advantage is that no information is prematurely dropped.

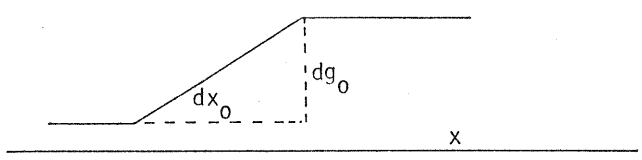

Fig. 4

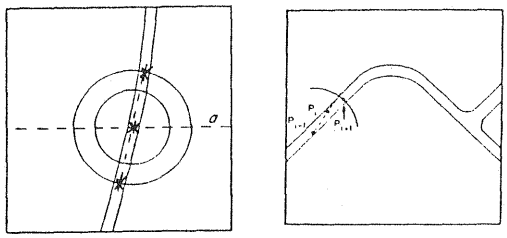

Fig. 5

3. Non-linear smoothing

Previous to segmentation, the image is enhanced by smoothing to reduce noise and texture. Smoothing is low-pass filtering and its most simple appearance is a 3 x 3 averaging. The mask of this, often called, box filter is shown in table 1[a]

The grey values may be weighted, often the Gaussian is used. For a 3 x 3 neighbourhood its mask is given in table 1[b]. Gaussian filters can be derived by repeated convolution of 2 x 2 box filters. These filters are linear, thus they smooth also edges. Non-linear filtering avoids edge flattening. We consider: (1) conditional average filter, (2) median filter and (3) edge-preserving smoothing filter. They didn't show significant differences with regard to precision. For road recognition conditional averaging is chosen because of its computational efficiency. Edge-preserving smoothing is especially developed to prevent edges for smoothing. It has also the capability to reduce texture considerably. The method, however, is computationally expensive.

$\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$	$\frac{1}{16}\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix} = \frac{1}{4}\begin{bmatrix}1&1\\1&1\end{bmatrix} * \frac{1}{4}\begin{bmatrix}1&1\\1&1\end{bmatrix}$
3x3 unweighted average filter	3x3 Gaussian filter built from convolution of two 2x2 unweighted averaging filters
a	b

Table 1

Conditional average filter. The new grey value of each pixel is the unweighted average of the grey value of the window. But if the difference between the pixel and a neighbouring pixel exceeds a threshold T the neighbouring pixel is excluded. The grey value G_0 of pixel P_0, with neighbourhood pixels P_i, i = 1, ..,n (see fig. 6. for a 3 x 3 neighbourhood), having grey values G_i, becomes after filtering:

$$\bar{G}_0 = \frac{1}{n} \sum_{G_i \in S} G_i$$

where:

$$S = \{ G_i \mid |G_i - G_0| < T \}_{i = 0,..,n}$$

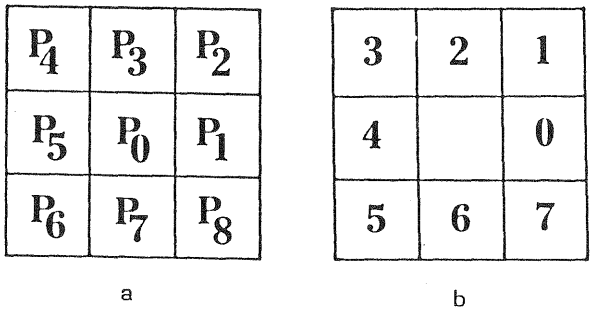

Fig. 6

The filter has the following effects:

- Small amounts of noise (noise < T) are smoothed in homogeneous regions.
- When G_0 lies on a boundary and the grey values at both sides differ more than T, the boundary isn't blurred. So, the filter preserves edges with a gradient larger than T.
- If the edge has a very slowly ascending across profile, i.e there are several gradient pixels, the filter averages roughly as many pixels above as beneath the actual pixel value. So, the noise is reduced but the gradient is maintained.
- Texture elements, which differ little in grey value, are eliminated.
- Spike or impulse noise doesn't affect the result of averaging.
- Spike noise itself isn't eliminated, when it occurs G_0 should be replaced by the average of the neighbourhood:

$$\bar{G}_0 = \frac{1}{n} \sum_{i=1}^{n} G_i$$

Median filters assign to each pixel the median grey value of the window. In practice often a 3 x 3 window is chosen, since the filter requires a time-consuming sorting procedure. The median filter is thoroughly used as non-linear smoothing filter. Its proporties are theoretical well understood (Gallagher and Wise, 1981). The filter reduces noise, without affecting edges. But, lines of one pixel thickness vanish, because, e.g., in a 3 x 3 window, a line contains only 3 pixels and they will loose competition with the other 6. On this property the application as linear feature extraction technique in (Sijmons, 1987) is founded.

Edge preserving smoothing searches the most homogeneous neighbourhood of each pixel and assigns to it the average grey value of that neighbourhood. The homogeneity is expressed in terms of variance. When the pixel under consideration lies on an edge there will be, when moving away, directions where the variance is low, i.e., the pixel belongs to that region, and directions with high variances. The principal notion is to rotate with an interval of $\Delta\phi$ (e.g. 45^0), an elongated mask around the pixel and to compute the variance of the grey values in the bar. The average of the grey values $\bar{g}_\phi(i,j)$ of the bar with the smallest variance σ^2_{min};

$$\sigma^2_{min} = \min\left(\sigma^2_{g_\phi}(i,j)\right);$$

$$\phi = r \cdot \Delta\phi \quad ; r = 1,..R; \quad R \cdot \Delta\phi = 360^0$$

is assigned to the pixel.

$$\sigma^2_{g_\phi}(i,j) = \frac{1}{n-1} \sum_{k \in V} \sum_{l \in W} \left(g(i+k,j+l) - \bar{g}_\phi(i,j)\right)^2$$

$$\bar{g}_{\phi}(i,j) = \frac{1}{n} \sum_{k \in V} \sum_{l \in W} (g(i+k, j+l))$$

where n is the number of pixels in the bar and V and W depend on ϕ and the size of the neighbourhood. A theoretically plain implementation results in a pixel cut off by the bar. i.e. each grey value should be weighted by the pixel area belonging to the bar. However, this is computationally costly. We have implemented a computationally less hostile method, although still rather time consuming, by considering only entire pixels and evaluating only four directions: $\phi = 45^{o}$, 135^{o}, 225^{o} and 315^{o} in a 5 x 5 neighbourhood. Since ϕ is referenced to the vertical axis of the grid, the resulting bars correspond to the 3 x 3 right-upper, right-under, left-under and left-upper quadrant respectively (see fig. 7). V and W become now:

	$\phi = 45^{o}$	$\phi = 135^{o}$	$\phi = 225^{o}$	$\phi = 315^{o}$
V	| 0, -2 |	| 0, -2 |	|-2, 0 |	| -2, 0 |
W	| -2, 0|	| 0, -2 |	| 0, -2 |	| -2, 0 |

4. Segmentation

The image is segmented by boundary detection. A boundary separates two regions which differ in grey value or texture. Edges are the individual features forming the boundary. They are detected by gradient operators (high-pass filtering). Commonly 3 x 3 masks are applied. The common edge detectors trace abrupt changes in grey values, but no changes in texture. Texture is a repeated spatial structure in grey value and causes problems to gradient operators. With combined texture measures and gradient operators, texture boundaries can be detected. But texture measures are time consuming. Therefore texture is reduced by smoothing. Examples of edge detectors are shown in fig. 2. The strength or magnitude M of the gradient

$$M = (g_x^2 + g_y^2)^{\frac{1}{2}}$$

indicates the probability of a pixel to lie on a boundary. The local direction of the boundary is computed from:

$$\phi = a \tan (g_y/g_x) + \tfrac{1}{2}\pi .$$

Besides edge detection, boundary detection involves also line- or border-following, i.e. connection of edges. The most rigorous way to determine whether a pixel belongs to an edge or not, is thresholding the edge magnitude. All pixels with a magnitude above the threshold are edges, all others are not. So, thresholding is a little differentiated method of binarisation. More refined techniques use neighbourhood information. They are based on probabilistic relaxation or on minimal cost patch search (graph search) combined with dynamic programming. It should be noted that graph search combines actually edge determination and line-following, although an initial low threshold is set to limit search space.

Several line-following techniques exists. They are reviewed in Rosenfeld & Kak, (1982 vol. 2, pp. 219-228.) Often lines are represented by a chain-code. A discrete arc can be defined by the coordinates of a starting point and a sequence of moves around the arc. In moving from the one pixel to the other on a rectangular tessellation only 8 directions are possible. They are coded from 0 to 7. The chaincode or Freeman code, originally developed for the guidance of computer controlled plotting, is commonly applied and shown in fig. 6[b]. After coding each series of edges is represented by a starting point and a sequence of codes, each code refering to the direction of an edge with respect to the foregoing edge in the sequence.

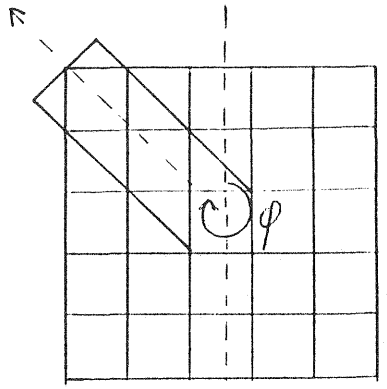

Fig. 7

Relaxation and dynamic programming are often applied in digital image processing. Relaxation has found its way in the detection of corresponding points in image sequences by feature matching. A classical paper is (Barnard and Thompson, 1981). Dynamic programming has been recently incorporated in photogrammetric stereo matching, using epipolar geometry, by (Bernard et. al. 1986). Edge relaxation and graph search are further elaborated in the sequel.

4.1 Edge relaxation

Edge relaxation utilizes neighbouring edges to weaken or to strengthen the probability that the edge belongs to a boundary. The method is greatly facilitated by constraining edges to lie between pixels (see fig. 8), causing boundaries to be covered entirely by horizontal and vertical edges (Prager, 1980). The simplest mask leading to horizontal and vertical edges between pixels is the normal gradient (see for mask, fig. 2) Experiments show that this simple mask gives the best overall results (Prager, 1980). The normal gradient will cause that wide edges will give multiple parallel indications of the same edge. In a sequence of parallel adjacent edges all but the strongest edges are eliminated by a non-maximum suppression. Next the total grey value difference between the shoulder and the toe of the gradient is assigned to the remaining edge, to achieve a more representative edge strength. Note that this approach corresponds to the hierarchical organization assumption of Marr (1982) which states that the spatial organisation of a surface's reflectance function is often generated by a number of different processes, each operating at a different scale.

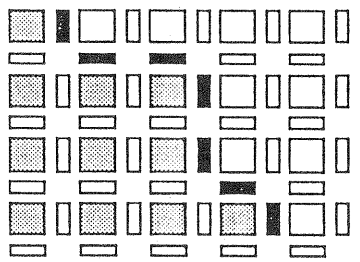

Fig. 8 The normal gradient detects horizontal and vertical edges. The edges are located between the pixels

Contrary to dynamic programming, relaxation is a stochastic approach. For a general discussion refer to (Rosenfeld and Kak, 1982, vol. 2, p. 154). To describe the boundary detection algorithm the representation of Prager (1980) summarized in fig. 9 is followed using a modified notation.

- The edge under consideration is represented by an open rectangle and denoted by r;
- A neighbouring edge position with the presents of an edge is shown as a black rectangle;
- A neighbouring edge position with no edge present is represented by a dotted line;
- A neighbouring edge position with an uncertain edge is represented by a solid line.

Edge position r is flanked at each side by 3 edge positions. They are indicated at the left side by L_1, L_2 and L_3 and at the right side by R_1, R_2 and R_3 (see fig. 10). In the sequel just the left side is considered.

Fig. 9

Each L_i, i = 1, 2, 3 may be either an edge or not. So, 0, 1, 2 or 3 edge positions are possible edges, leading to four vertex types q_j, j = 0, .., 3 (see Fig. 11). The edge strength expresses the probability of an edge position to be an edge: P(r), $P(L_i)$ and $P(R_i)$, i = 1, 2, 3. Depending on $P(L_i)$ and $P(R_i)$ -vertex type- P(r) is readjusted. Without loss of generality, $P(L_i)$ is reordered to a descending sequence:

$P_1, P_2, P_3 : P_1 > P_2 > P_3$.

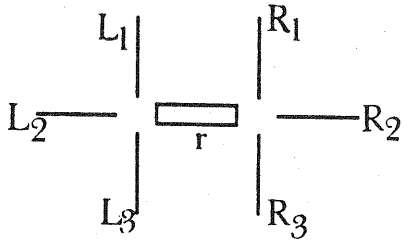

Fig.10

Assuming mutual independency of the edges, the probabilities $P(q_i)$ of the four vertex types q_j, j = 1,..,4, follows from probability theory:

$$P(q_0) = (1 - P_1)(1 - P_2)(1 - P_3)$$
$$P(q_1) = P_1(1 - P_2)(1 - P_3)$$
$$P(q_2) = P_1P_2(1 - P_3)$$
$$P(q_3) = P_1P_2P_3$$

The vertex type q_i with the largest probability $P(q_i) = \max(P(q_0), P(q_1), P(q_2), P(q_4))$ is chosen as the actual vertex configuration. There are some refinements necessary. Implementation without more ado, causes that when P_1, P_2 and P_3 are all small, but P_1 relatively large with respect to P_2 and P_3, a q_1 configuration may stay undetected. Therefore the non-edge probabilities are not related to 1 but to P_0, with $P_0 = \max(P_1, P_2, P_3)$, giving:

$$P(q_0) = (P_0 - P_1)(P_0 - P_2)(P_0 - P_3)$$
$$P(q_1) = P_1(P_0 - P_2)(P_0 - P_3)$$
$$P(q_2) = P_1P_2(P_0 - P_3)$$
$$P(q_3) = P_1P_2P_3$$

However, in this way an actual q_0 vertex type can be recognized as q_1, since P_1 may be large only with respect to P_2 and P_3, but not in absolute sense. So, P_0, has to exceed a certain threshold P_{min}, leading to the final definition: $P_0 = \max(P_1, P_2, P_3, P_{min})$. (In the current implementation P_{min} is set to 0.1). At each end of r, 4 vertex types q_i means at both ends 16 vertex types. Because of symmetry only 10 vertex types have to be considered. Examples of vertex types are sketched in fig. 12. For instance, vertex type q_0 - q_0 designates an edge position without neighbouring edges. So, r is probably due to noise and P(r) has to be decreased. Vertex type q_0 - q_1 may indicate that r marks the beginning or the end of a line, but this is not sure since r may be also the result of noise So, P(r) is neither augmented nor decreased. Vertex type q_1 - q_1 shows that it is most likely that r is part of a line; P(r) has to be increased. Each vertex type can be evaluated in this way. The result is shown in fig. 13 and table 2.

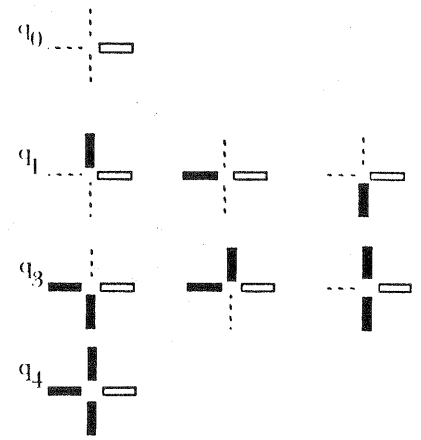

Fig. 11

The probability in iteration step k+1 is readjusted according to:

$$-: \quad P^{k+1}(r) = \max(0, P^k(r) - c)$$
$$0: \quad P^{k+1}(r) = P^k(r)$$
$$+: \quad P^{k+1}(r) = \min(1, P^k(r) + c)$$

where c is a constant; 0.2 in the current implementation. A large c gives fast convergence, i.e. a small neighbourhood is involved. A small c causes the opposite.

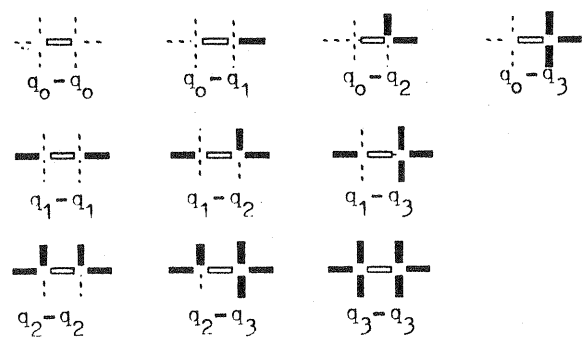

Fig. 12 Vertex types in edge relaxation

The above can be summarized in the following algorithm:

1 Search candidate edges by grey value thresholding and assign to them an initial probability $P^0(r)$ proportional to the strength of the gradient;
2 For k=1 to a steady situation (thus each candidate edge has either reached probability 0 or 1):
  3 Determine the vertex type;
  4 Readjust $P^k(r)$ on the ground of the vertex type.

The relaxation procedure drives the edge probability to 0 or 1. Next the edges with a high probability are connected by binary line-following using 4 connectedness, (i.e., each pixel has 4 neighbours).

Vertex type	q_0	q_1	q_2	q_3
q_0	-	0	-	-
q_1		+	+	+
q_2			0	0
q_3				0

- : P(r) is decreased;
0 : P(r) unchanged;
+ : P(r) is augmented.

Table 2

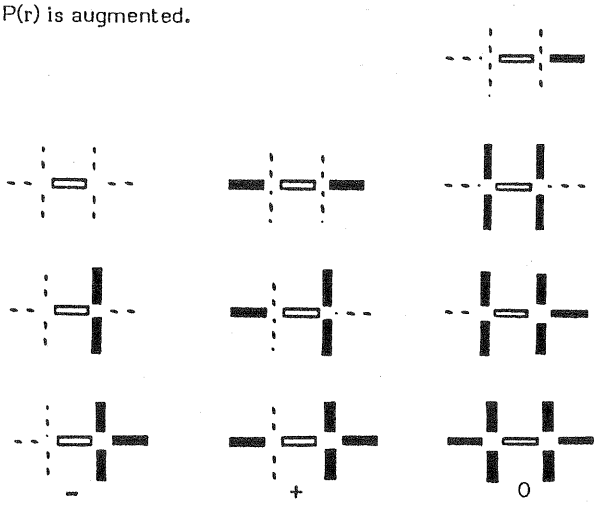

Fig. 13 Effect of the vertex type on the edge propability (- negative, 0 don't care, + positive)

4.2 Graph searching with dynamic programming

A graph consists of a set of nodes and arcs between nodes. An edge detector results in the magnitude image M_i and the direction image ϕ_i, where ϕ_i may be interpreted as nodes in a graph, each with costs M_i. Which path is the optimum one, i.e. associated with the least costs?

Dynamic programming is a computational technique for solving optimization problems in a sequential way. It is originated in optimizing route planning. There are several paths (policies) to go from A to N (see fig. 14). With each path costs are associated. The purpose is to find the path with the minimum of total costs. It isn't necessary to evaluate all possible paths between A and N (fig. 14) sequentially, because the theorem of optimality may be applied which states that an optimal policy must contain only optimal subpolicies.

To evaluate the paths an evaluation or costs function is necessary. M_i and ϕ_i are plausible parameters to build the cost function, arranged as a weighted sum of high cumulative M_i and low cumulative $\Delta\phi_i = |\phi_{i+1} - \phi_i|$. For an arc of length n between the stages s and t, the cost function C_{st} becomes:

$$C_{st} = \sum_{i=s}^{t} M_i + \alpha \sum_{i=s}^{t} \Delta\phi_i$$

with α a negative factor between -1 and 0, weighting the influence of the curvature relative against the edge strength. Applying the theorem of optimality, the cost function can be optimized in stages to find the optimal path f_i^t:

$$f_i^0 = 0$$
$$f_i^1 = \max (C_{01} + f_i^0)$$
$$f_i^t = \max (C_{st} + f_i^s)$$

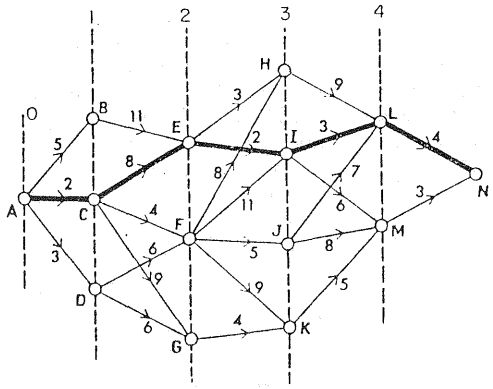

Fig. 14 Several paths to go from A to N, each associated with a cost. With dynamic programming the optimum path can be detected.

A more comprehensive discussion can be found in (Ballard and Brown, 1982). To diminish the search space dimension M_i is thresholded: $M_i > T_m$. Pixels become part of the graph, along which the optimal paths are searched, when they fulfil the following conditions:

$$M_i > T_m; \quad \Delta\phi_i \leq T_\phi;$$

$$|\theta_{ij} - \phi_i| \leq T_\theta; \quad |\theta_{ij} - \phi_j| \leq T_\theta$$

with θ_{ij} the direction of pixel i to pixel j; T_m, T_ϕ and T_θ thresholds. The general structure of the algorithm is given by:

1 Until the search space is empty, search max (M_i) the starting points of a new boundary.
2 Set $T_\phi = 90^0$ and $T_\theta = 90^0$, corresponding with 5 directions in 8-connective neighbourhood, and build the graph for n elements, with n= 4;
3 Compute the most optimal path in the one direction and remove the evaluated path from the search space;
4 Compute the most optimal path in the other direction and remove the evaluated path from the search space;
5 Go to 1.

The results of edge relaxation and dynamic programming on a 128 x 128 test image are shown in fig. 15 and fig. 16, respectively.

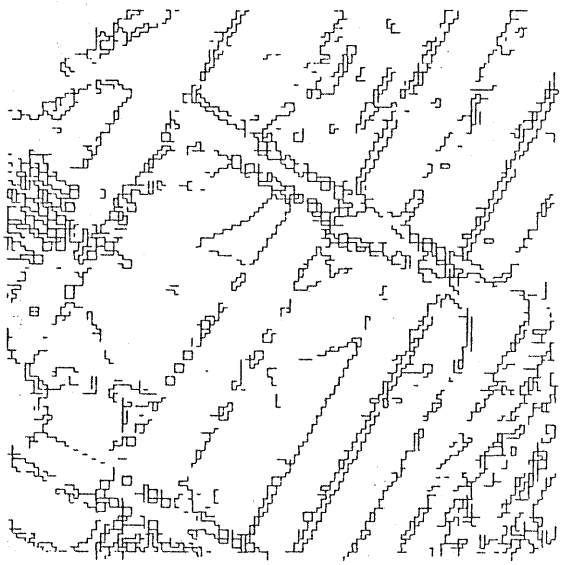

Fig. 15 128x128 test image after edge relaxation.

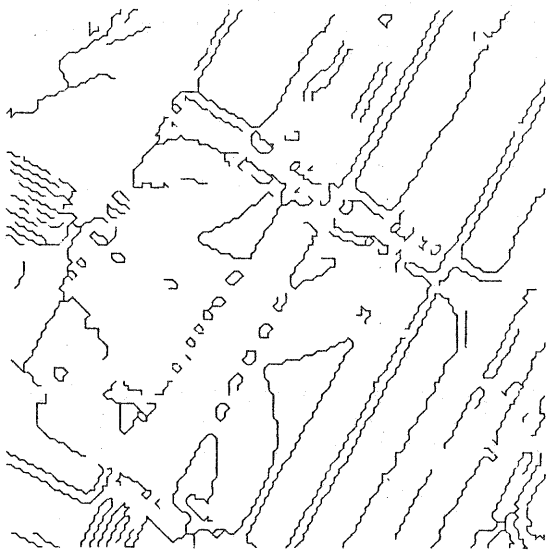

Fig. 16 128x128 test image after graph searching using dynamic programming.

5. Road recognition

Road recognition is performed using shape. Recognition by shape requires an object model. This is easy to realize in the branch of computer vision, called robot vision, because the product uniquely defines the model. But at the surface of the earth there is nearly an infinity of shapes. This complexity has strongly obstructed application of shape recognition.

The results of segmentation are elementary regions, for which now basic properties can be computed, e.g.:

1: average grey level in the individual spectral bands;
2: area size (A);
3: perimeter (P);
4: length (L);
5: width (W);
6: centre of gravity.

(2) - (5) define shape and orientation of the region.

5.1 Region descriptors

Some shape descriptors are:

Elongatedness $E_1 = A / W^2$; $E_2 = L / W$

Non compartness value C: $C = P^2 / A$

In the present investigation E_2 in combination with an eigen value analysis is used. The set of n pixels belonging to a region, may be considererd as a scatter diagram of 2 variables, each observed n times. The variables are the (x, y) co-ordinates. Suppose that C is the covariance matrix:

$$C = \begin{pmatrix} C_{xx} & C_{xy} \\ C_{xy} & C_{yy} \end{pmatrix}$$

with variances:

$$C_{xx} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2; \qquad C_{yy} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2$$

and covariance:

$$C_{xy} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

with $\bar{x}$ and $\bar{y}$ the coordinates of the centre of gravity (i.e., mean):

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i; \qquad \bar{y} = \frac{1}{n} \sum_{i=p}^{n} y_i.$$

By an eigen value transformation the eigen space can be computed. The eigen values λ_1 and λ_2 define the 1st and 2nd principal axes, which correspond to the long and short axes of the error ellips, i.e. the shape of the distribution. The eigen value is computed from:

$$\begin{vmatrix} C_{xx} - \lambda & C_{xy} \\ C_{xy} & C_{yy} - \lambda \end{vmatrix} = 0$$

λ_1 and λ_2 define the shape, not the real dimensions. So it isn't necessary to compute them from the above quadratic form since a measure of the flattening of the ellipsoid, i.e. elongatedness, is, e.g.: $E = (\lambda_1 / \lambda_2)^{\frac{1}{2}}$.

The flattening can also be given in a roundness measure R:

$$R = 1 - \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2 = \frac{4\ \lambda_1\ \lambda_2}{(\lambda_1 + \lambda_2)^2}$$

$0 \leq R \leq 1$. When R is 1 the shape of the region is a circle. When R reached 0, the shape is very elongated.

Because: $\lambda_1 \lambda_2 = C_{xx} C_{yy} - C^2_{xy}$

and $\lambda_1 + \lambda_2 = C_{xx} + C_{yy}$

$$R = \frac{4 \det C}{(\mathrm{tr}\ C)^2}$$

So, it isn't necessary to compute λ_1 and λ_2 explicitely. Because one wants often absolute measures for length and width it is appropriate to know $E = (\lambda_1 / \lambda_2)^{\frac{1}{2}}$ since, once the length is computed the width can be derived from the shape. E can be derived from R, using:

$$E = \sqrt{\frac{1 + \sqrt{(1 - R)}}{1 - \sqrt{(1 - R)}}}$$

The derivation can be found in appendix I. The direction ϕ of the 1st principal axis is given by:

$$\phi = \tfrac{1}{2}\ \mathrm{atan}\ \frac{2\ C_{xy}}{C_{xx} - C_{yy}}$$

5.2 Region junction

Edge relaxation is carried out on all three bands. The resulting edge image is the sum of the three bands. This may cause that regions become rather small, since weak edges, which exceed only in one band the threshold, lead also to boundaries. The mean spectral properties of adjacent regions are compared and the regions are joined if the maximum of the differences of three bands k.

$$\max\ (\ \Delta\ b^k_{ij}) = \max\ (|\ B^k_i - B^k_j|\ )$$

doesn't exceed a threshold T_b, i.e. $\max(\Delta\ b^k_{ij}) < T_b$. In the present investigation T_b is set to 20. The spectral response of the new region

$$\bar{B}^k_i = B^k_i \cup B^k_j$$

is computed from a weighted average:

$$\bar{B}^k_i = \frac{n_i\ B^k_i + n_j\ B^k_j}{n_i + n_j}$$

with n_i and n_j the number of pixels of region i and j respectively. There are small regions with spectral characteristics which differ entirely from all adjacent regions. When it is enclosed by only one region it is prefered to remove this
enclave. If they are less than 20 pixels they become the spectral properties of their enclosing region.

5.3 Road detection

Now we arrive at the point that the elementary regions have to be classified into road regions and background regions on ground of their shape. Due to intervening objects segmentation divides a road into several elongated regions. When a region satisfies a couple of conditions it is promoted to candidate road region. These conditions are:

- The width w should be larger than 2m, but smaller than 12m, e.g. $2 < w < 12$m, or, in pixel format, $6 < w < 40$ pixels.
- The elongatedness L/w, with L the length of the region and w the width, should exceed 3; e.g. $L/w > 3$.

To each region i, which fulfils the above conditions, a probability P_i is assigned: $P_i = 2L_i/N$. N is a normalization factor roughly corresponding to the linear image size, in the present investigation N = 500. The candidate regions form the starting points for further search, in which the following road properties are involved:

- roads form an interconnected network, e.g. roads don't suddenly end;
- the surface of a road is made of asphalt, e.g. all road segments have the same spectral properties
- the local curvature has an upper bound.

The above properties are translated into the following geometric and spectral conditions:

1 The shortest distance ΔL_{ij} bridging the gap between two candidate regions i and j is:

$$\Delta\ L_{ij} = \min(T_d,\ \tfrac{1}{2}(L_i + L_j).$$

In the current computations T_d is chosen such that the distance never exceeds 25m, sufficient to bridge small intervening objects, i.e. $T_d = 80$ pixels.

2 The directions ϕ_i and ϕ_j of the 1st principal axes coincide with each other

$$\Delta\phi = |\ \phi_i - \phi_j\ | < T_\phi$$

In the current computations T_ϕ is set to 0.3 rad.

3 The differences of the average grey values in the three spectral bands don't exceed a threshold:

$$\Delta\ B^k_{ij} = |\ B^k_i - B^k_j\ | < T^k_B$$

where k refers to the spectral bands. In the current computations T^k_B was set to 20.

Each time regions fulfil one of the above conditions their probability is augmented by:

$$P_i = 2L_j/N.$$

Note that L_j refers to the neighbour region. Condition 1 fulfils the neighbourhood relationship. Only neighbouring regions which fulfil this condition, are examined. The probability of straight neigbouring road parts is augmented both by condition 2 and 3. The probability of neighbouring road segments in curves is augmented by condition 3.

Once all regions are examined, a simple threshold is applied on the probabilities to define roads and background. The gaps between the road segments are filled by a directional dilation, i.e. the road segments are expanded in the direction of the first principal axis. The above road recognition procedure has shown good results.

6. Geometric accuracy analysis

The precision of the two boundary detection methods in combination with the smoothing filters and edge detectors are examined with respect to an accurate reference map. The reference lines are shown in fig. 17. A visual impression of the quality of edge relaxation and dynamic programming is given in fig. 18 and fig. 19, respectively. The results are satisfying. For quantitative description three measures are developed.

6.1 Precision measures

Three precision measures will be considered. For the present investigation just measure 3 showed to be appropriate.

Measure 1. The most appropriate measure is based on the comparison of vectors since in vector format the spatial information will be stored. Vectorization, i.e. raster-vector conversion, is a problem apart. We apply the following vectorization process: a straight sequence of edges is adjusted by a straight (regression) line, i.e. the parameters (a, b) of the line equation $x = ay + b$, are computed by least squares. Crossing lines are designated manually and their intersection is computed. Suppose that (x^i, y^i) are the planar co-ordinates of the image vertices and (x_0^i, y_0^i) the coordinates of the corresponding map vectors, than measure 1 (m_1) is given by:

$$m_i = \frac{1}{n} \sum_{i=1}^{n} \sqrt{(x_0^i - x^i)^2 + (y_0^i - y^i)^2}$$

with n the number vertices in the image. Due to small number of number of vertices this measure isn't suitable for the the present investigation.

Measure 2 is inspired by computer graphics, where one is faced with vector-raster conversion for display purposes. The notion is to overlay the line image with the reference map converted to raster .The similarity of pixels is a measure for the precision, i.e.

$$m_2 = \frac{100\ \%}{n} \sum_{i=1}^{n} | L_0^i - L^i |$$

with L_0^i the pixels of the reference line and L^i those of the image line, both in binary format, i.e. the line pixels have value 1 and the background pixels 0. The lack is that small deviations in line position leads to bad m_2.

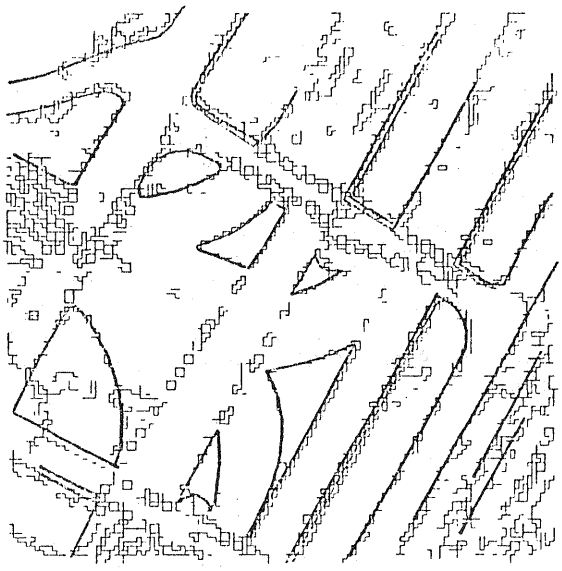

Fig. 18 Location of the edge relaxation lines relatively to the reference lines.

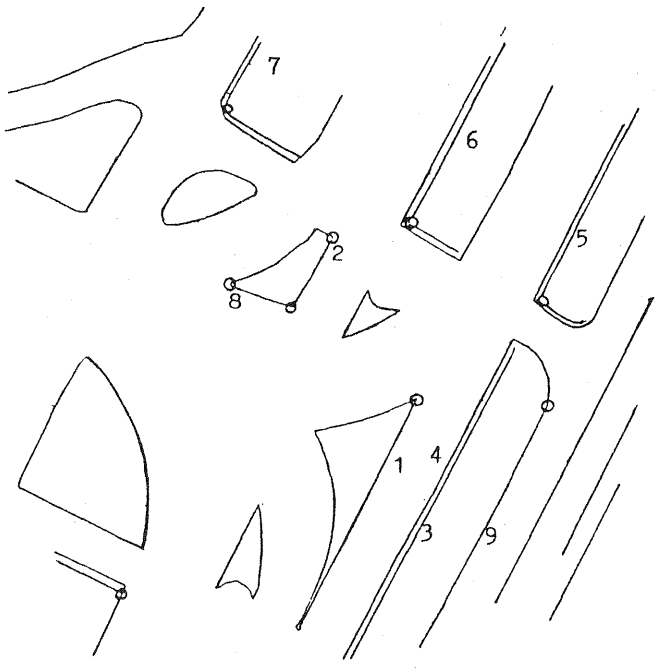

Fig. 17 Chosen reference lines in the 128x128 test image for geometric precision analysis of edge relaxation and dynamic programming.

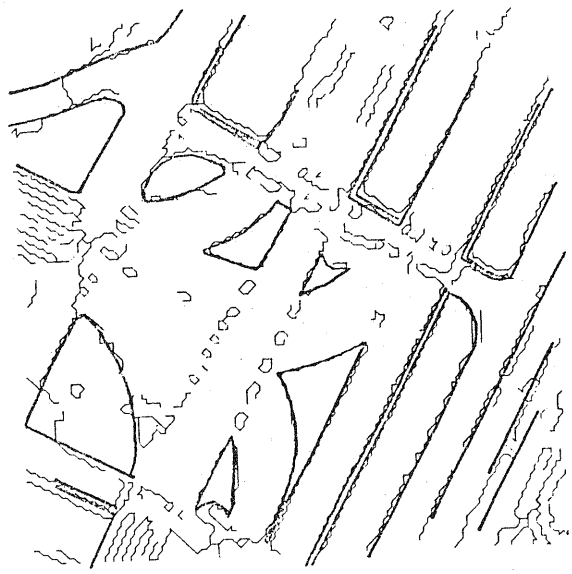

Fig. 19 Location of the dynamic programming lines relatively to the reference lines.

Measure 3 computes the mean perpendicular distance of the edges to the reference line. Suppose that (x^l, y^l) is the position of an edge and d_iL the perpendicular distance of (x^l, y^l) to the reference line, m_3 is now defined by:

$$m_3 = \frac{1}{n} \sum_{i=1}^{n} d_i L$$

with n the number of pixels part of the line.

6.2 Results

Nor the kind of smoothing filter nor the kind of edge detection affect the precision significantly. This invariance is probably caused by the excellent image quality. Computational efficiency requires the least time consuming conditional averaging and normal gradient. Table 3 shows the precision for edge relaxation and dynamic programming ordered per line. In case of edge relaxation the listed figures, refer to the same smoothing filter and edge detector as is applied for road recognition, i.e. conditional average filter and normal gradient. In case of dynamic programming the figures refer to the conditional average filter and the Prewitt operator. As explained, measure 1 is skipped. Because measure 2 isn't very reliable only from measure 3 conclusions are drawn. Our limited test set shows that dynamic programming gives better results. Only for line 4 and 9 edge relaxation appears to be superior.

	measure 2 (deviation in %)		measure 3 (deviation in pixels)	
line	edgerel.	dyn. prog.	edgerel.	dyn. prog.
1	89	53	0.80	0.64
2	29	23	0.65	0.45
3	10	84	0.55	0.51
4	17	16	0.36	0.46
5	7	65	0.39	0.32
6	22	24	0.42	0.37
7	11	39	0.46	0.39
8	20	25	0.50	0.42
9	25	59	0.47	0.54

Table 3

7 Conclusions

Two techniques to segment digital aerial images by boundary detection are presented. Road recognition is performed, using shape. The segmentation methods are based on probabilistic relaxation and dynamic programming. Special emphasis is placed on geometric accuracy. Separate road regions are linked using adjacency and spectral conditions. Through roads are detected with good results. Intervening features, like cars and shadows, cause rather frayed road borders. They should be detected too. Shadows are hard to identify because they don't have unique spectral characteristics, e.g., houses throw darker shadows then trees. Adding the infrared band will augment the shadow information.

Object recognition can further be improved by a probabilistic relaxation scheme. At present, a simple threshold on the probability is employed. However, using neighbouring regions, the probability of each region can be readjusted iteratively. Multispectral information isn't used in absolute sense, i.e., multispectral classification, but only relatively, i.e. to trace whether two or more regions are part of the same road. However, multispectral classification is able to determine a priori the non-road regions, which reduces considerably search space. Incorporation of a spatial database, e.g. a digital map or Geographic Information System (GIS), gives benefit. An outline for interfacing imagery with GIS data is expounded in (Lemmens, 1987; 1988).

Our testdata point out that dynamic programming is preferable to edge relaxation. Dynamic programming is computationally less expensive and brings out a better visual representation. Their geometric accuracy is investigated using sharp lines. Dynamic programming shows a better accuracy. The weights of the edge direction in the dynamic costs function isn't significant. So, the edge directions may be ignored. This is, probably, effected by the high image quality, which may also cause the little difference in effect of the smoothing filters and edge detectors. The mean perpendicular distance (measure 3) is appropriate. Vertex comparison (measure 1) should be an adequate measure, too, but, the few well defined vertices in the present investigation forced us to omit it. A shift or rotation leads to a small similarity measure for measure 2, so, it isn't suitable.

Analysis is carried out on the original spatial resolution level, i.e., the pixel size remains unchanged. Multiresolution approaches have proved to be very useful in many digital image processing applications and corresponds to the theory of human vision (c.f., Marr and Hildreth, 1980; Marr, 1979). The human visual system first takes a general view, next the scene part of interest is examined closely. It fits also the good geodetic sense to work from the large to the small. The present paper doesn't describe an operational road recognition system. It is shown that objects can be recognized using shape information deduced from general knowledge about the scene.

Acknowledgement: The authors wish to thank Prof. I.T. Young and the members of his research group of the faculty of Applied physics of the Delft University of Technology for useful suggestions, criticism and helpful assistance. Computations were carried out at the image processing system of the Institute of Pattern Recognition of the above faculty and at the VAX 750 of the faculty of Geodesy. Support by KLM Aerocarto, Den Haag, is greatfully acknowledged.

References

- **Ballard, D.H., Brown, C.M., 1982,** Computer Vision, Prentice Hall.
- **Bajcsy, R., Tavakoli, M., 1976,** Computer recognition of roads from satellite pictures, IEEE Trans. on Systems, Man and Cybernetics, vol. 6, no. 9, pp. 623-637.
- **Barnard, S.T., Thompson, W.B., 1981,** Disparity analysis of images, IEEE Trans. on PAMI, vol. 2, pp. 333-340.
- **Benard, M., Boutaleb, A.K., Köbl, O., Penis, C., 1986,** Automatic stereophotogrammetry: implementation and comparison of classical correlation methods and dynamic programming based techniques, Int. Arch. of Photogr., vol. 26-III, Rovaniemi, 1986, (see also Interlaken 1987).
- **Fischler, M.A., Tenenbaum, J.M., Wolf, H.C., 1981,** Detection of roads and linear structures in low-resolution aerial imaging using a multisource knowledge integration technique, Computer Graphics and Image Processing, vol. 15, pp. 201-233.
- **Gallagher, N.C., Wise, G.L., 1981,** A theoretical analysis of the properties of median filters, IEEE Trans. on ASSP, vol. 29, no. 6, pp. 1136-1141.
- **Groch, W.D., 1982,** Extraction of line shaped objects from aerial images using a special operator to analyze the profiles of functions, Computer Graphics and Image Processing, vol. 18, pp. 347-358.
- **Kauffman, A., 1967,** Graphs, dynamic programming and finite games, Academic Press.
- **Lemmens, M.J.P.M., 1987,** Towards a model for (semi-) automatic analysis of digital images, Joint colloquium of the ISPRS working groups II/1, II/2 and II/6, Londen, Sept. 21-23, pp. 15.
- **Lemmens, M.J.P.M., 1988,** GIS-digital image interaction, Archives of the 16th Congress of the ISPRS, Comm. IV, Kyoto, Japan.
- **Marr, D., 1979,** Vision, Freeman and Compagny, San Francisco.
- **Marr, D. Hildreth, E., 1980,** Theory of edge detection, Proc. R. Soc. London, B207, pp. 187-217.
- **Prager, J.M., 1980,** Extracting and labeling boundary segments in natural scenes, IEEE Trans. on PAMI, vol. 2, no. 1, pp. 16-26.
- **Rosenfeld and Kak, 1982,** Digital picture processing, academic press.
- **Sijmons, K., 1987,** Computer-assisted detection of linear features from digital remote sensing data, ITC-Journal, nr. 1, pp. 23-31.

Appendix I

Derivation of E from R:

R is a roundness measure of the error ellips, defined by

$$R = 1 - \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \right)^2 \qquad (I.1)$$

with λ_1 the largest and λ_2 the smallest eigen value. E is an elongatedness measure:

$$E = (\lambda_1 / \lambda_2)^{\frac{1}{2}} \quad \text{or:} \quad \lambda_1 = \lambda_2 E^2 \qquad (I.2)$$

Inserting (I.2) in (I.1) yields:

$$R = 1 - \left(\frac{\lambda_2 E^2 - \lambda_2}{\lambda_2 E^2 + \lambda_2} \right)^2$$

$$= 1 - \left(\frac{E^2 - \lambda_1}{E^2 + \lambda_1} \right)^2 \qquad (I.3)$$

From (I.3) it follows:

$$\frac{E^2 - 1}{E^2 + 1} = (1 - R)^{\frac{1}{2}} \qquad (I.4)$$

or:

$$E = \left(\frac{1 + (1 - R)^{\frac{1}{2}}}{1 - (1 - R)^{\frac{1}{2}}} \right)^{\frac{1}{2}} \qquad (I.5)$$